GENERAL DESCRIPTION

The XR17V352[1] (V352) is a single chip 2-channel PCI Express (PCIe) UART (Universal Asynchronous Receiver and Transmitter), optimized for higher performance and lower power. The V352 serves as a single lane PCIe bridge to 2 indepedent enhanced 16550 compatible UARTs. The V352 is compliant to PCIe 2.0 Gen 1 (2.5GT/s).

In addition to the UART channels, the V352 has 16 multi-purpose I/Os (MPIOs), a 16-bit general purpose counter/timer and a global interrupt status register to optimize interrupt servicing.

Each UART of the V352 has many enhanced features such as the 256-bytes TX and RX FIFOs, programmable Fractional Baud Rate Generator, Automatic Hardware or Software Flow Control, Auto RS-485 Half-Duplex Direction Control, programmable TX and RX FIFO Trigger Levels, TX and RX FIFO Level Counters, infrared mode, and data rates up to 25Mbps. The V352 is available in a 113-pin FPBGA package (9 x 9 mm).

NOTE 1: *Covered by U.S. Patents #5,649,122, #6,754,839, #6,865,626 and #6,947,999*

APPLICATIONS

- Next generation Point-of-Sale Systems
- Remote Access Servers
- Storage Network Management
- Factory Automation and Process Control
- Multi-port RS-232/RS-422/RS-485 Cards

FEATURES

- Single 3.3V power supply
- Internal buck regulator for 1.2V core
- PCIe 2.0 Gen 1 compliant
- x1 Link, dual simplex, 2.5Gbps in each direction
- EEPROM interface for configuration
- Data read/write burst operation
- Global interrupt status register for both UARTs
- Up to 25 Mbps serial data rate
- 16 multi-purpose inputs/outputs (MPIOs)
- 16-bit general purpose timer/counter
- Sleep mode with wake-up Indicator
- Two independent UART channels controlled with
  - 16550 compatible register Set
  - 256-byte TX and RX FIFOs
  - Programmable TX and RX Trigger Levels
  - TX/RX FIFO Level Counters
  - Fractional baud rate generator
  - Automatic RTS/CTS or DTR/DSR hardware flow control with programmable hysteresis
  - Automatic Xon/Xoff software flow control
  - RS-485 half duplex direction control output with programmable turn-around delay
  - Multi-drop with Auto Address Detection
  - Infrared (IrDA 1.1) data encoder/decoder
- Software compatible to XR17C15x, XR17D15x, XR17V25x PCI UARTs

FIGURE 1. BLOCK DIAGRAM OF THE XR17V352

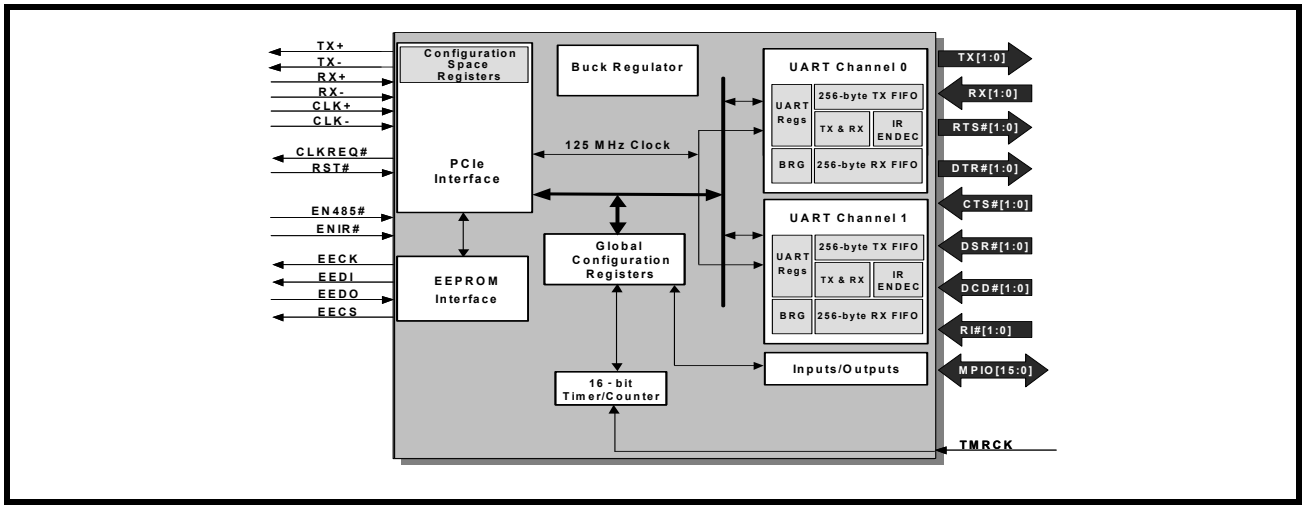
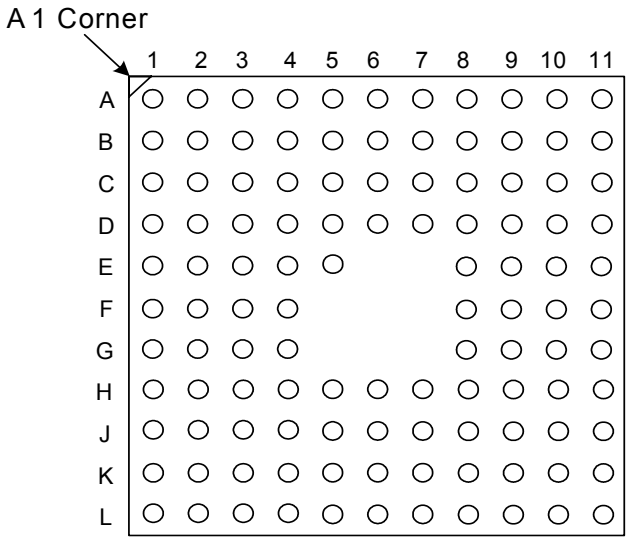


FIGURE 2. 113-FPBGA PINOUT



Transparent Top View

	1	2	3	4	5	6	7	8	9	10	11
A	NC	MPIO0	NC	GND	NC	TMRCK	TEST2	GND	LX	LX	NC
B	MPIO3	MPIO2	NC	NC	NC	ENIR#	TEST1	GND	VCC33	VCC33	VCC33
C	MPIO6	MPIO4	MPIO1	NC	NC	EN485#	FB	GND	VCC33	ENABLE	NC
D	GND	TEST0	MPIO5	GND	VCC33	GND	VCC12	GND	PWRGD	TEST3	GND
E	RX+	RX-	GND	CLK+	GND			VCC33	TEST5	TEST4#	RI1#
F	GND	GND	REXT	CLK-				GND	CD1#	DSR1#	DTR1#
G	TX+	TX-	GND	GND				VCC12	RTS1#	RX1	CTS1#
H	GND	VCC12	VCC33	GND	VCC12	GND	VCC33	GND	RI0#	TX1	GND
J	CLKREQ#	PERST#	MPIO9	MPIO13	RESET#	TMS	EECK	EEDI	RX0	DSR0#	CD0#
K	MPIO7	MPIO8	MPIO11	MPIO14	NC	TCK	TDO	EECS	TX0	CTS0#	DTR0#
L	NC	MPIO10	MPIO12	GND	MPIO15	TRST#	TDI	GND	EEDO	RTS0#	NC

ORDERING INFORMATION

PART NUMBER	PACKAGE	OPERATING TEMPERATURE RANGE	DEVICE STATUS
XR17V352IB113-F	113-FPBGA	-40°C to +85°C	In Development

PIN DESCRIPTIONS

NAME	PIN #	TYPE	DESCRIPTION
PCIe SIGNALS			
CLK+ CLK-	E4 F4	I I	PCIe reference clock input.
TX+ TX-	G1 G2	O O	PCIe differential TX outputs
RX+ RX-	E1 E2	I I	PCIe differential RX inputs
CLKREQ#	J1	O	PCIe edge connector clock request
PERST#	J2	I	PCIe edge connector reset
REXT	F3		Connect a 191 ohm 1% resistor to GND. This is used for PCIe PHY calibration.
MODEM OR SERIAL I/O INTERFACE			
TX0	K9	O	UART channel 0 Transmit Data or infrared transmit data.
RX0	J9	I	UART channel 0 Receive Data or infrared receive data. Normal RXD input idles at HIGH condition. The infrared pulses can be inverted internally prior to decoding by setting FCTR bit [4].
RTS0#	L10	O	UART channel 0 Request to Send or general purpose output (active LOW).
CTS0#	K10	I	UART channel 0 Clear to Send or general purpose input (active LOW).
DTR0#	K11	O	UART channel 0 Data Terminal Ready or general purpose output (active LOW).
DSR0#	J10	I	UART channel 0 Data Set Ready or general purpose input (active LOW).
CD0#	J11	I	UART channel 0 Carrier Detect or general purpose input (active LOW).
RI0#	H9	I	UART channel 0 Ring Indicator or general purpose input (active LOW).
TX1	H10	O	UART channel 1 Transmit Data or infrared transmit data.
RX1	G10	I	UART channel 1 Receive Data or infrared receive data. Normal RXD input idles at HIGH condition. The infrared pulses can be inverted prior to decoding by setting FCTR bit [4].
RTS1#	G9	O	UART channel 1 Request to Send or general purpose output (active LOW).
CTS1#	G11	I	UART channel 1 Clear to Send or general purpose input (active LOW).
DTR1#	F11	O	UART channel 1 Data Terminal Ready or general purpose output (active LOW).
DSR1#	F10	I	UART channel 1 Data Set Ready or general purpose input (active LOW).
CD1#	F9	I	UART channel 1 Carrier Detect or general purpose input (active LOW).
RI1#	E11	I	UART channel 1 Ring Indicator or general purpose input (active LOW).

PIN DESCRIPTIONS

NAME	PIN #	TYPE	DESCRIPTION
MPIO SIGNALS			
MPIO0	A2	I/O	Multi-purpose input/output 0. The function of this pin is defined thru the Configuration Register MPIOSEL, MPIOLVL, MPIOINV, MPIO3T and MPIOINT
MPIO1	C3	I/O	Multi-purpose input/output 1. The function of this pin is defined thru the Configuration Register MPIOSEL, MPIOLVL, MPIOINV, MPIO3T and MPIOINT.
MPIO2	B2	I/O	Multi-purpose input/output 2. The function of this pin is defined thru the Configuration Register MPIOSEL, MPIOLVL, MPIOINV, MPIO3T and MPIOINT.
MPIO3	B1	I/O	Multi-purpose input/output 3. The function of this pin is defined thru the Configuration Register MPIOSEL, MPIOLVL, MPIOINV, MPIO3T and MPIOINT.
MPIO4	C2	I/O	Multi-purpose input/output 4. The function of this pin is defined thru the Configuration Register MPIOSEL, MPIOLVL, MPIOINV, MPIO3T and MPIOINT.
MPIO5	D3	I/O	Multi-purpose input/output 5. The function of this pin is defined thru the Configuration Register MPIOSEL, MPIOLVL, MPIOINV, MPIO3T and MPIOINT.
MPIO6	C1	I/O	Multi-purpose input/output 6. The function of this pin is defined thru the Configuration Register MPIOSEL, MPIOLVL, MPIOINV, MPIO3T and MPIOINT.
MPIO7	K1	I/O	Multi-purpose input/output 7. The function of this pin is defined thru the Configuration Register MPIOSEL, MPIOLVL, MPIOINV, MPIO3T and MPIOINT.
MPIO8	K2	I/O	Multi-purpose input/output 8. The function of this pin is defined thru the Configuration Register MPIOSEL, MPIOLVL, MPIOINV, MPIO3T and MPIOINT
MPIO9	J3	I/O	Multi-purpose input/output 9. The function of this pin is defined thru the Configuration Register MPIOSEL, MPIOLVL, MPIOINV, MPIO3T and MPIOINT.
MPIO10	L2	I/O	Multi-purpose input/output 10. The function of this pin is defined thru the Configuration Register MPIOSEL, MPIOLVL, MPIOINV, MPIO3T and MPIOINT.
MPIO11	K3	I/O	Multi-purpose input/output 11. The function of this pin is defined thru the Configuration Register MPIOSEL, MPIOLVL, MPIOINV, MPIO3T and MPIOINT.
MPIO12	L3	I/O	Multi-purpose input/output 12. The function of this pin is defined thru the Configuration Register MPIOSEL, MPIOLVL, MPIOINV, MPIO3T and MPIOINT.
MPIO13	J4	I/O	Multi-purpose input/output 13. The function of this pin is defined thru the Configuration Register MPIOSEL, MPIOLVL, MPIOINV, MPIO3T and MPIOINT.
MPIO14	K4	I/O	Multi-purpose input/output 14. The function of this pin is defined thru the Configuration Register MPIOSEL, MPIOLVL, MPIOINV, MPIO3T and MPIOINT.
MPIO15	L5	I/O	Multi-purpose input/output 15. The function of this pin is defined thru the Configuration Register MPIOSEL, MPIOLVL, MPIOINV, MPIO3T and MPIOINT.
EEPROM SIGNALS			
EECK	J7	O	Serial clock to EEPROM. An internal clock of CLK divide by 256 is used for reading the vendor and sub-vendor ID during power up or reset. However, it can be manually clocked thru the Configuration Register REGB.
EECS	K8	O	Chip select to a EEPROM device like 93C46. It is manually selectable thru the Configuration Register REGB. Requires a pull-up 4.7K ohm resistor for external sensing of EEPROM during power up.

PIN DESCRIPTIONS

NAME	PIN #	TYPE	DESCRIPTION
EEDI	J8	O	Write data to EEPROM device. It is manually accessible thru the Configuration Register REGB.
EEDO	L9	I	Read data from EEPROM device. It is manually accessible thru the Configuration Register REGB.
JTAG SIGNALS			
TRST#	L6	I	JTAG Test Reset. This signal is active LOW.
TCK	K6	I	JTAG Test Clock
TMS	J6	I	JTAG Test Mode Select
TDI	L7	I	JTAG Data Input
TDO	K7	O	JTAG Data Output
BUCK REGULATOR SIGNALS			
ENABLE	C10	I	Connect to VCC to enable buck regulator. Connect to GND to disable buck regulator.
LX LX	A9 A10	O O	Connect these two signals together to external 4.7uH inductor.
FB	C7	I	Connect this signal to other end of external 4.7uH inductor. 47uF capacitor to GND is also required on this pin.
PWRGD	D9	O	Indicates that 1.2V core has been powered up.
ANCILLARY SIGNALS			
RESET#	J5	I	System reset (active low). In normal operation, this signal should be HIGH.
TMRCK	A6	I	16-bit timer/counter external clock input.
EN485#	C6	I	Auto RS-485 mode enable (active low). This pin is sampled during power up, following a hardware reset (RST#) or soft reset (register RESET). It can be used to start up both UARTs in the Auto RS-485 Half-Duplex Direction control mode. The sampled logic state is transferred to FCTR bit-5 in the UART channel.
ENIR#	B6	I	Infrared mode enable (active low). This pin is sampled during power up, following a hardware reset (RST#) or soft-reset (register RESET). It can be used to start up both UARTs in the infrared mode. The sampled logic state is transferred to MCR bit-6 in the UART.
TEST0 TEST1 TEST2 TEST3	D2 B7 A7 D10	I I I I	Factory Test Modes. For normal operation, connect to GND.
TEST4#	E10	I	Factory Test Mode 4. For normal operation, connect to VCC.
TEST5	E9	I/O	Factory Test I/O. For normal operation, connect to pull-down resistor.
VCC33	D5, E8, H7	Pwr	3.3V I/O power supply.
VCC33A	H3	Pwr	3.3V analog PHY power supply. A ferrite bead is recommended on this pin.
VCC33P	B9, C9	Pwr	3.3V power supply voltage for output stage of buck regulator.

PIN DESCRIPTIONS

NAME	PIN #	TYPE	DESCRIPTION
VCC33B	B10, B11	Pwr	3.3V power supply for the analog blocks of the buck regulator.
VCC12	D7, G8, H5	Pwr	1.2V core power supply. A ferrite bead is recommended on these pins.
VCC12A	H2	Pwr	1.2V analog PHY power supply. A ferrite bead is recommended on this pin.
GND	A4, A8, B8, C8, D1, D4, D6, D8, D11, E3, E5, F1, F2, F8, G3, G4, H1, H4, H6, H8, H11, L4, L8	Pwr	Power supply common, ground.
NC	A1, A3, A5, A11, B3, B4, B5, C4, C5, C11, K5, L1, L11	-	No internal connection.

NOTE: *Pin type: I=Input, O=Output, IO= Input/output, OD=Output Open Drain, OT=Output Tristate, IS=Input Schmitt Trigger.*

FUNCTIONAL DESCRIPTION

The XR17V352 (V352) integrates the functions of two independent enhanced 16550 UARTs, a general purpose 16-bit timer/counter, and 16 multi-purpose I/Os (MPIOs). Each UART channel has its own 16550 UART compatible configuration register set for individual channel control, status and data transfer. The device configuration registers include a set of four consecutive interrupt source registers that provides interrupt status for both UARTs, timer/counter, MPIOs and a sleep wake-up indicator. Additionally, each UART channel has 256-byte of transmit and receive FIFOs, automatic RTS/CTS or DTR/DSR hardware flow control, automatic XON/XOFF, special character flow control, programmable transmit and receive FIFO trigger levels, infrared encoder/decoder (IrDA ver. 1.1), and a programmable fractional baud rate generator with a prescaler of divide by 1 or 4, and a data rate up to 25 Mbps with the 4X sampling rate.

PCIe Interface and Data Transfers

This is the host interface and it meets the PCIe base specifications revision 2.0 Gen 1. The V352 also supports data read/write burst operations so the 256-byte TX or RX FIFO can be loaded or unloaded in a single transaction greatly increasing the overall system performance.

Local Bus Configuration Space Registers

A set of local bus configuration space register is provided. These registers provide the PCI vendor ID, device ID, sub-vendor ID, product model number, resources and capabilities which is collected by the host during the auto configuration phase that follows immediately after a power up or system reset/reboot. After the host has sorted out all devices on the bus, it defines and download the operating conditions to the cards. One of the definitions is the base address loaded into the Base Address Register (BAR) where the card will be operating in the PCI local bus memory space. All this is described in more detail in **“Section 1.1, PCI LOCAL BUS CONFIGURATION SPACE REGISTERS” on page 8**.

EEPROM Interface

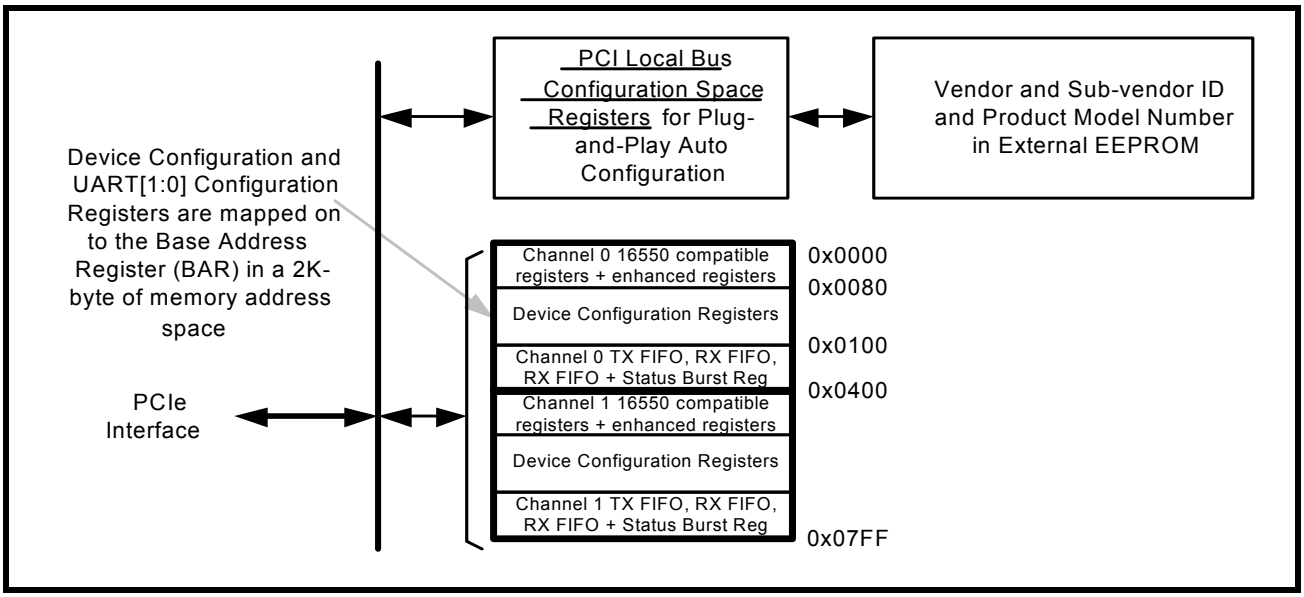
An external 93C46 EEPROM is used to store words of information such as PCI Vendor ID, PCI Device ID, Class Code, etc. Details of this information can be found in **“Section 1.2, EEPROM Interface” on page 12**. This information is only used with the plug-and-play auto configuration of the PCI local bus. These data provide automatic hardware installation onto the PCI bus. The EEPROM interface consists of 4 signals, EEDI, EEDO, EECS, and EECK. The EEPROM is not needed when auto configuration is not required in the application. However, if your design requires non-volatile memory for other purpose, it is possible to store and retrieve data on the EEPROM through a special PCI device configuration register. See application note DAN112 for its programming details.

1.0 XR17V352 INTERNAL REGISTERS

The XR17V352 UART register set is very similar to the previous generation PCI UARTs. This makes the V352 software compatible with the previous generation PCI UARTs. Minimal changes are needed to the software driver of an existing Exar PCI UART driver so that it can be used with the V352 PCIe UART.

There are three different sets of registers as shown in Figure 3. The ***PCI Local Bus Configuration Space Registers*** is needed for plug-and-play auto-configuration. This auto-configuration feature makes installation very easy into a PCI system and it is part of the PCI local bus specification. The second register set is the ***Device Configuration Registers*** that are also accessible directly from the PCI bus for programming general operating conditions of the device and monitoring the status of various functions common to both channels. These functions include both channel UARTs' interrupt control and status, 16-bit general purpose timer control and status, multipurpose inputs/outputs control and status, sleep mode, soft-reset, and device identification and revision. And lastly, each UART channel has its own set of internal ***UART Configuration Registers*** for its own operation control and status reporting. Both sets of channel registers are embedded inside the device configuration registers space, which provides faster access. The second and third set of registers are mapped into 2K of the PCI bus memory address space. The following paragraphs describe all 3 sets of registers in detail.

FIGURE 3. THE XR17V352 REGISTER SETS



1.1 PCI LOCAL BUS CONFIGURATION SPACE REGISTERS

The PCI local bus configuration space registers are responsible for setting up the device's operating environment in the PCI local bus. The pre-defined operating parameters of the device is read by the PCI bus plug-and-play auto-configuration manager in the operating system. After the PCI bus has collected all data from every device/card on the bus, it defines and downloads the memory mapping information to each device/card about their individual operation memory address location and conditions. The operating memory mapped address location is downloaded into the Base Address Register (BAR) register, located at an address offset of 0x10 in the configuration space. Custom modification of certain registers is possible by using an external 93C46 EEPROM. The EEPROM contains the device vendor and sub-vendor data, along with 6 other words of information (see **"Section 1.2, EEPROM Interface" on page 12**) required by the auto-configuration setup.

TABLE 1: PCI LOCAL BUS CONFIGURATION SPACE REGISTERS

ADDRESS OFFSET	BITS	TYPE	DESCRIPTION	RESET VALUE (HEX OR BINARY)
0x00	31:16	EWR	Device ID	0x0352
	15:0	EWR	Vendor ID (Exar) specified by PCISIG	0x13A8
0x04	31 30	RWC RWC	Parity error detected. Cleared by writing a logic 1. System error detected. Cleared by writing a logic 1.	0b 0b
	29:28	RO	Unused	00b
	27	RO	Target Abort.	0b
	26:25	RO	DEVSEL# timing.	00b
	24	RO	Unemployments bus master error reporting bit	0b
	23	RO	Fast back to back transactions are supported	0b
	22	RO	Reserved Status bit	0b
	21	RO	66MHz capable	0b
	20	RO	Capabilities List	1b
	19:16	RO	Reserved Status bits	0000b
	15:11, 9,7, 5, 4, 3, 2	RO	Command bits (reserved)	0x0000
	10	RWR	This bit disables the device from asserting INTx#. logic 1 = disable assertion of INTx# and logic 0 = enables assertion of INTx#	0b
	8	RWR	SERR# driver enable. logic 1=enable driver and 0=disable driver	0b
	6	RWR	Parity error enable. logic 1=respond to parity error and 0=ignore	0b
	1	RWR	Command controls a device's response to mem space accesses: 0=disable mem space accesses, 1=enable mem space accesses	0b
	0	RO	Device's response to I/O space accesses is disabled. (0 = disable I/O space accesses)	0b
0x08	31:8	EWR	Class Code (Default is 'Simple 550 Communication Controller')	0x070002
	7:0	RO	Revision ID (Exar device revision number)	Current Rev. value
0x0C	31:24	RO	BIST (Built-in Self Test)	0x00
	23:16	RO	Header Type (a single function device with one BAR)	0x00
	15:8	RO	Unimplemented Latency Timer (needed only for bus master)	0x00
	7:0	RO	Unimplemented Cache Line Size	0x00
0x10	31:11	RWR	Memory Base Address Register (BAR0)	0x00000
	10:0	RO	Claims an 2K address space for the memory mapped UARTs including the UARTs on the expansion interface.	0x0000
0x14	31:0	RWR	Unimplemented Base Address Register (returns zeros)	0x00000000

TABLE 1: PCI LOCAL BUS CONFIGURATION SPACE REGISTERS

ADDRESS OFFSET	BITS	TYPE	DESCRIPTION	RESET VALUE (HEX OR BINARY)
0x18h	31:0	RO	Unimplemented Base Address Register (returns zeros)	0x00000000
0x1C	31:0	RO	Unimplemented Base Address Register (returns zeros)	0x00000000
0x20	31:0	RO	Unimplemented Base Address Register (returns zeros)	0x00000000
0x24	31:0	RO	Unimplemented Base Address Register (returns zeros)	0x00000000
0x28	31:0	RO	Reserved	0x00000000
0x2C	31:16	EWR	Subsystem ID (write from external EEPROM by customer)	0x0000
	15:0	EWR	Subsystem Vendor ID (write from external EEPROM by customer)	0x0000
0x30	31:0	RO	Expansion ROM Base Address (Unimplemented)	0x00000000
0x34	31:8	RO	Reserved (returns zeros)	0x000000
	7:0	RO	Capability Pointer	0x50
0x38	31:0	RO	Reserved (returns zeros)	0x00000000
0x3C	31:24	RO	Unimplemented MAXLAT	0x00
	23:16	RO	Unimplemented MINGNT	0x00
	15:8	RO	Interrupt Pin, use INTA#.	0x01
	7:0	RWR	Interrupt Line.	0xXX
0x40	31:0	RO	Not implemented or not applicable (return zeros)	0x00000000
0x44	31:0	RO	CSR	0x02106160
0x48	31:0	RO	Not implemented or not applicable (return zeros)	0x00000000
0x4C	31:0	RO	Not implemented or not applicable (return zeros)	0x00000000
0x50	31:16	RO	64-bit address capable	0x0080
	15:8	RO	Next Capability Pointer	0x78
	7:0	RO	MSI Capable Capability ID	0x05
0x54-0x67	31:0	RO	Not implemented or not applicable (return zeros)	0x00000000
0x68	31:0	RO	Not implemented or not applicable	0x0000xxxx
0x6C-0x77	31:0	RO	Not implemented or not applicable (return zeros)	0x00000000
0x78	31:16	RO	PME# support (PME# can be asserted from D3hot and D0) PCI Power Management 1.2	0x4803
	15:8	RO	Next Capability Pointer	0x80
	7:0	RO	Power Management Capability ID	0x01
0x7C	31:0	RO	No soft reset when transitioning from D3hot to D0 state	0x00000008

TABLE 1: PCI LOCAL BUS CONFIGURATION SPACE REGISTERS

ADDRESS OFFSET	BITS	TYPE	DESCRIPTION	RESET VALUE (HEX OR BINARY)
0x80	31:16	RO	PCI Express 2.0 capable endpoint, Interrupt Message Number 1	0x0202
	15:8	RO	Next Capability Pointer	0x00
	7:0	RO	PCI Express Capability ID	0x10
0x84	31:16	RO	Not implemented or not applicable (return zeros)	0x0000
	15:8	RO	Role-Based Error Reporting	0x80
	7:0	RO	256 bytes max payload size	0x01
0x88	31:16	RW	Not implemented or not applicable (return zeros)	0x0000
	15:8	RW	512 bytes max read request, Enable No Snoop	0x28
	7:0	RW	256 bytes max TLP payload size	0x10
0x8C	31:24	RO	Port Number	0x01
	23:22	RO	Not implemented or not applicable (return zeros)	00b
	21:18	RO	Not implemented or not applicable (return zeros)	0000b
	17:15	RO	L1 Exit Latency < 1 us	000b
	14:12	RO	L0s Exit Latency < 64 ns	000b
	11:10	RO	Active State Power Management (ASPM) Support L0s and L1 Supported	11b
	9:4	RO	x1 max Link Width	000001b
	3:0	RO	2.5GT/s Link speed supported	0001b
0x90	31:21	RO	Not implemented or not applicable (return zeros)	00000000000b
	20	RO	Data Link Layer Active Reporting capable	1b
	19	RO	Surprise Down Error Reporting not supported	0b
	18	RO	Reference clock must not be removed.	0b
	17:15	RO	L1 Exit Latency - 2 us to less than 4 us	010b
	14:10	RO	Not implemented or not applicable (return zeros)	00000b
	9:4	RO	x1 negotiated Link Width	000001b
	3:0	RO	Current Link Speed is 2.5GT/s	0001b
0x94	31:0	RO	PCIe Capability Offset 0x14 - Slot Capabilities Register	0x00040000
0x98-0xAF	31:0	RO	Not implemented or not applicable (return zeros)	0x00000000
0xB0	31:0	RO	PCIe Capability Offset 0x30 - Link Status2/Control2	0x00010001
0xB4-0xFF	31:0	RO	Not implemented or not applicable (return zeros)	0x00000000
0x100	31:0	RO	VC Resource Capability Register	0x00010002

TABLE 1: PCI LOCAL BUS CONFIGURATION SPACE REGISTERS

ADDRESS OFFSET	BITS	TYPE	DESCRIPTION	RESET VALUE (HEX OR BINARY)
0x104-0x113	31:0	RO	Not implemented or not applicable (return zeros)	0x00000000
0x114	31:0	RO	VC Offset 0x4	0x8000000FF

NOTE: EWR=Read/Write from external EEPROM. RWR=Read/Write. RO= Read Only. RWC=Read/Write-Clear.

1.2 EEPROM Interface

The V352 provides an interface to an Electrically Erasable Programmable Read Only Memory (EEPROM). The EEPROM must be a 93C46-like device, with its memory configured as 16-bit words. This interface is provided in order to program the registers in the PCI Configuration Space of the PCI UART during power-up. The EEPROM must be organized into address/data pairs. The first word of the pair is the address and the second word is the data. Table 2 below shows the format of the 16-bit address:

TABLE 2: EEPROM ADDRESS BIT DEFINITIONS

BIT(S)	DEFINITION
15	Parity Bit - Odd parity over entire address/data pair If there is a parity error, it will be reported in bit-3 of the REGB register in the Device Configuration Registers (offset 0x08E).
14	Final Address If 1, this will be the last data to be read. If 0, there will be more data to be read after this.
13:8	Reserved - Bits must be '0'
7:0	Target Address - See Table 3

Table 3 shows the Target Addresses available for programming into bits 7:0 of the 16-bit address word. All other Target Addresses are reserved and must not be used.

TABLE 3: TARGET ADDRESS FOR EEPROM VALUES

TARGET ADDRESS	DATA	EXAR DEFAULT
0x00	Vendor ID	0x13A8
0x01	Device ID	0x0352
0x02	Class Code [7:0] lower 8-bits are reserved	0x0200
0x03	Class Code [23:8]	0x0700
0x04	Subsystem Vendor ID	0x0000
0x05	Subsystem ID	0x0000

The second 16-bit word of the address/data pair is the data. The default values are shown in Table 3. The address/data pairs can be in any order. Only the contents which need to be changed from the Exar defaults need to be included in the EEPROM.

1.3 Device Internal Register Sets

The ***Device Configuration Registers*** and the two individual ***UART Configuration Registers*** of the V352 occupy 2K of PCI bus memory address space. These addresses are offset onto the basic memory address, a value loaded into the Memory Base Address Register (BAR) in the PCI local bus configuration register set. The UART Configuration Registers are mapped into 2 address blocks where each UART channel occupies 1024 bytes memory space for its own registers that include the 16550 compatible registers. The Device Configuration Registers are accessible from both UART channels. However, not all bits can be controlled by both channels. The UART channel can only control the 8XMODE, 4XMODE, RESET and SLEEP register bits that apply to that particular channel. For example, this prevents channel 0 from accidentally resetting channel 1.

All these registers can be accessed in 8, 16, 24 or 32 bits width depending on the starting address given by the host at the beginning of the bus cycle. Transmit and receive data may be loaded or unloaded in 8, 16, 24 or 32 bits format in special locations given in the **Table 4** below. Every time a read or write operation is made to the transmit or receive register, its FIFO data pointer is automatically bumped to the next sequential data location either in byte, word or DWORD. One special case applies to the receive data unloading when reading the receive data together with its LSR register content. The host must read them in 16 or 32 bits format in order to maintain integrity of the data byte with its associated error flags. These special registers are further discussed in **"Section 2.1, FIFO DATA LOADING AND UNLOADING IN 32-BIT FORMAT" on page 28**.

TABLE 4: XR17V352 UART AND DEVICE CONFIGURATION REGISTERS

OFFSET ADDRESS	MEMORY SPACE	READ/WRITE	COMMENT
0x0000 - 0x000F	UART channel 0 Regs	(Table 12 & Table 13)	First 8 regs are 16550 compatible
0x0010 - 0x007F	Reserved		
0x0080 - 0x009A	DEVICE CONFIGURATION REGISTERS	(Table 5)	
0x009B - 0x00FF	Reserved		
0x0100 - 0x01FF	UART 0 – Read FIFO	Read-Only	256 bytes of RX FIFO data
0x0100 - 0x01FF	UART 0 – Write FIFO	Write-Only	256 bytes of TX FIFO data
0x0200 - 0x03FF	UART 0 – Read FIFO with errors	Read-Only	256 bytes of RX FIFO data + LSR
0x0400 - 0x040F	UART channel 1 Regs	(Table 12 & Table 13)	First 8 regs are 16550 compatible
0x0410 - 0x047F	Reserved		
0x0480 - 0x049A	DEVICE CONFIGURATION REGISTERS	(Table 5)	
0x049B - 0x04FF	Reserved		
0x0500 - 0x05FF	UART 1 – Read FIFO	Read-Only	256 bytes of RX FIFO data
0x0500 - 0x05FF	UART 1 – Write FIFO	Write-Only	256 bytes of TX FIFO data
0x0600 - 0x07FF	UART 1 – Read FIFO with errors	Read-Only	256 bytes of RX FIFO data + LSR

1.4 Device Configuration Registers

The Device Configuration Registers provide easy programming of general operating parameters to the V352 and for monitoring the status of various functions. These registers control or report on both channel UARTs functions that include interrupt control and status, 16-bit general purpose timer control and status, multipurpose inputs/outputs control and status, sleep mode control, soft-reset control, and device identification and revision, and others. Tables 5 and 6 below show these registers in BYTE and DWORD alignment. Each of these registers is described in detail in the following paragraphs.

TABLE 5: DEVICE CONFIGURATION REGISTERS SHOWN IN BYTE ALIGNMENT

ADDRESS [A7:A0]	REGISTER	READ/WRITE COMMENT	RESET STATE
Ox080	INT0 [7:0]	Read-only Interrupt [7:0]	Bits [7:0] = 0x00
Ox081	INT1 [15:8]	Read-only	Bits [7:0] = 0x00
Ox082	INT2 [23:16]	Read-only	Bits [7:0] = 0x00
Ox083	INT3 [31:24]	Read-only	Bits [7:0] = 0x00
Ox084	TIMERCNTL	Read/Write Timer Control	Bits [7:0] = 0x00
Ox085	REGA	Reserved	Bits [7:0] = 0x00
Ox086	TIMERLSB	Read/Write Timer LSB	Bits [7:0]= 0x00
Ox087	TIMERMSB	Read/Write Timer MSB	Bits [7:0]= 0x00
		Individual UART channels can only control the bit pertaining to that channel in the registers at address offset 0x088-0x08B.	
Ox088	8XMODE	Read/Write	Bits [7:0] = 0x00
Ox089	4XMODE	Read/Write	Bits [7:0] = 0x00
Ox08A	RESET	Write-only Self clear bits after executing Reset	Bits [7:0] = 0x00
Ox08B	SLEEP	Read/Write Sleep mode	Bits [7:0]= 0x00
Ox08C	DREV	Read-only Device revision	Bits [7:0] = Current Rev.
Ox08D	DVID	Read-only Device identification	Bits [7:0] = 0x82
Ox08E	REGB	Read/Write EEPROM control	Bits [7:0] = 0x00
Ox08F	MPIOINT[7:0]	Read/Write MPIO[7:0] interrupt mask	Bits [7:0] = 0x00
Ox090	MPIOLVL[7:0]	Read/Write MPIO[7:0] level control	Bits [7:0] = 0x00
Ox091	MPIO3T[7:0]	Read/Write MPIO[7:0] output control	Bits [7:0] = 0x00
Ox092	MPIOINV[7:0]	Read/Write MPIO[7:0] input polarity select	Bits [7:0] = 0x00
Ox093	MPIOSEL[7:0]	Read/Write MPIO[7:0] select	Bits [7:0] = 0xFF
0x094	MPIOOD[7:0]	Read/Write MPIO[7:0] open-drain output control	Bits [7:0] = 0x00
Ox095	MPIOINT[15:8]	Read/Write MPIO[15:8] interrupt mask	Bits [15:8] = 0x00
Ox096	MPIOLVL[15:8]	Read/Write MPIO[15:8] level control	Bits [15:8] = 0x00
Ox097	MPIO3T[15:8]	Read/Write MPIO[15:8] output control	Bits [15:8] = 0x00

TABLE 5: DEVICE CONFIGURATION REGISTERS SHOWN IN BYTE ALIGNMENT

ADDRESS [A7:A0]	REGISTER	READ/WRITE COMMENT	RESET STATE
Ox098	MPIOINV[15:8]	Read/Write MPIO[15:8] input polarity select	Bits [15:8] = 0x00
Ox099	MPIOSEL[15:8]	Read/Write MPIO[15:8] select	Bits [15:8] = 0xFF
0x09A	MPIOOD[15:8]	Read/Write MPIO[15:8] open-drain output control	Bits [15:8] = 0x00
0x09B	Reserved		0x00

TABLE 6: DEVICE CONFIGURATION REGISTERS SHOWN IN DWORD ALIGNMENT

ADDRESS	REGISTER	BYTE 3 [31:24]	BYTE 2 [23:16]	BYTE 1 [15:8]	BYTE 0 [7:0]
0x0080-0x0083	INTERRUPT (read-only)	INT3	INT2	INT1	INT0
0x0084-0x0087	TIMER (read/write)	TIMERMSB	TIMERLSB	Reserved	TIMERCNTL
0x0088-0x008B	ANCILLARY1 (read/write)	SLEEP	RESET	4XMODE	8XMODE
0x008C-0x008F	ANCILLARY2 (read-only)	MPIOINT[7:0]	REGB	DVID	DREV
0x0090-0x0093	MPIO1 (read/write)	MPIOSEL[7:0]	MPIOINV[7:0]	MPIO3T[7:0]	MPIOLVL[7:0]
0x0094-0x0097	MPIO2 (read/write)	MPIO3T[7:0]	MPIOLVL[15:8]	MPIOINT[15:8]	MPIOOD[7:0]
0x0098-0x009B	MPIO3 (read/write)	Reserved	MPIOOD[15:8]	MPIOSEL[15:8]	MPIOINV[15:8]

1.4.1 The Global Interrupt Registers - INT0, INT1, INT2 and INT3

The XR17V352 has a 32-bit wide register [INT0, INT1, INT2 and INT3] to provide interrupt information and supports two interrupt schemes. The first scheme is an 4-bit indicator representing both channels with each bit representing each channel from 0 to 1. This permits the interrupt service routine to quickly determine which UART channels need servicing so that it can go to the appropriate UART channel interrupt service routines. INT0 bit [0] represents the interrupt status for UART channel 0 when its transmitter, receiver, line status, or modem port status requires service. Other bits in the INT0 register provide indication for the other channels with bit [1] representing UART channel 1 respectively.

The second scheme provides detail about the source of the interrupts for each UART channel. All the interrupts are encoded into a 3-bit code. This 3-bit code represents 7 interrupts corresponding to individual UART's transmitter, receiver, line status, modem port status. INT1, INT2 and INT3 registers provide the 24-bit interrupt status for both channels. bits [10:8] representing channel 0 and bits [13:11] representing channel 1 respectively. All other bits are reserved. Both channel interrupts status are available with a single DWORD read operation. This feature allows the host another method to quickly service the interrupts, thus reducing the service interval and host bandwidth requirement.

Note that the interrupts reported in this register is specific to each UART channel. If there is a global interrupt such as the wake-up interrupt, timer/counter interrupt or MPIO interrupt, they would be reported in the 3-bit code for channel 0 in INT1.

GLOBAL INTERRUPT REGISTER (DWORD) [default 0x00-00-00-00]

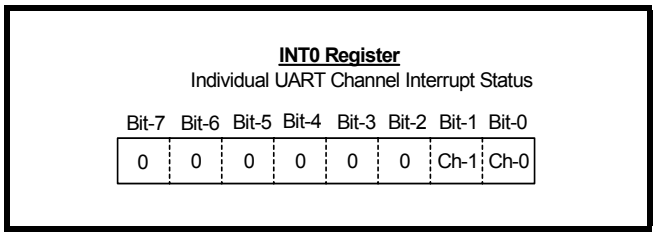
INT3 [31:24]	INT2 [23:16]	INT1 [15:8]	INT0 [7:0]

All bits start up zero. A special interrupt condition is generated by the V352 upon awakening from sleep after both channels were put to sleep mode earlier. This wake-up interrupt is cleared by a read to the INT0 register. Figure 4 shows the 4-byte interrupt register and its make up.

INT0 [7:0] Channel Interrupt Indicator

Each bit gives an indication of the channel that has requested for service. Bit [0] represents channel 0 and bit [1] indicates channel 1. All other bits are reserved. Logic 1 indicates the channel N [1:0] has called for service. The interrupt bit clears after reading the appropriate register of the interrupting channel register, see Interrupt Clearing section.

The INT0 register provides individual status for each channel



INT3, INT2 and INT1 [31:8] 3-bit Channel Interrupt Encoding

Each channel's interrupt is encoded into 3 bits for receive, transmit, and status. Bits [10:8] represent channel 0 and go up to channel 1 with bits [13:11]. The 3-bit encoding and their priority order are shown below in **Table 7**. The wake-up interrupt, timer/counter interrupt and MPIO interrupt are only reported in channel 0 of INT1 (bits[10:8]). These interrupts are not reported in any other location.

FIGURE 4. THE GLOBAL INTERRUPT REGISTER, INT0, INT1, INT2 AND INT3

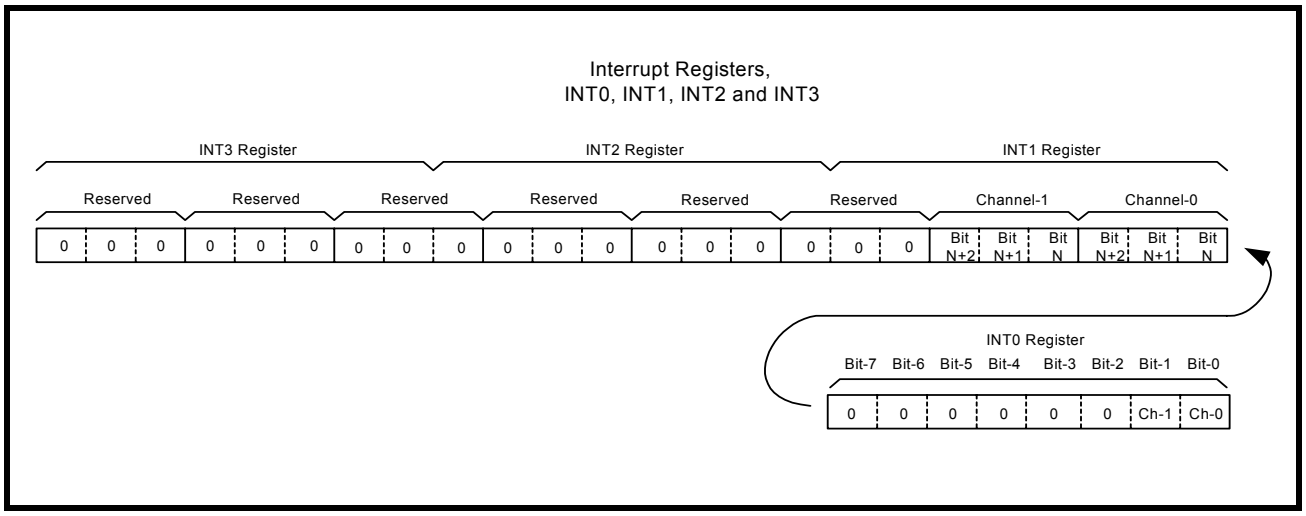


TABLE 7: UART CHANNEL [1:0] INTERRUPT SOURCE ENCODING

PRIORITY	BIT[N+2]	BIT[N+1]	BIT[N]	INTERRUPT SOURCE(S)
x	0	0	0	None or wake-up indicator (wake-up indicator is reported in channel 0 only)
1	0	0	1	RXRDY and RX Line Status (logic OR of LSR[4:1])
2	0	1	0	RXRDY Time-out
3	0	1	1	TXRDY, THR or TSR (auto RS485 mode) empty
4	1	0	0	MSR, RTS/CTS or DTR/DSR delta or Xoff/Xon det. or special char. detected
5	1	0	1	Reserved.
6	1	1	0	MPIO pin(s). Reported in channel 0 only.
7	1	1	1	Timer/Counter. Reported in channel 0 only.

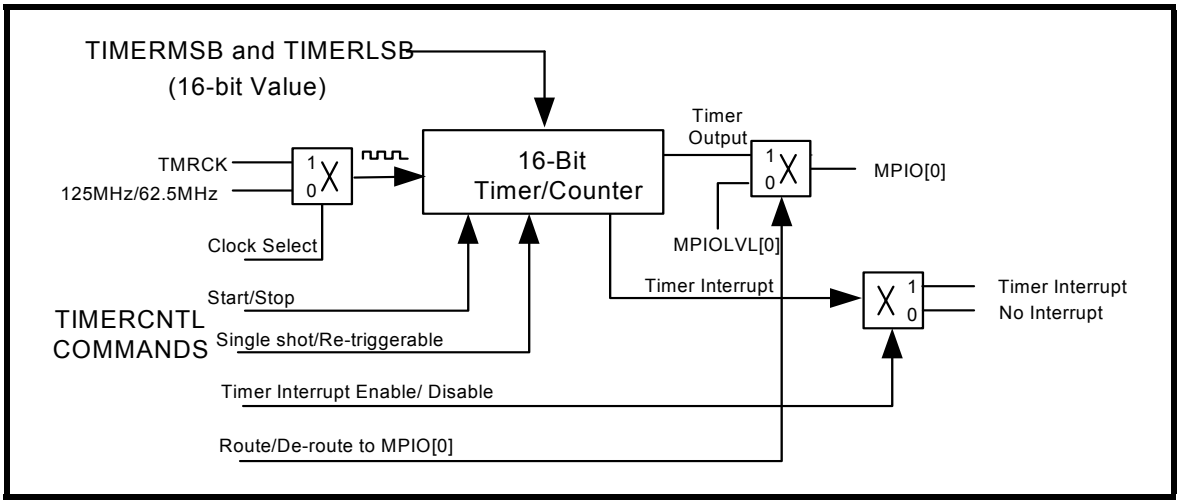
TABLE 8: UART CHANNEL [1:0] INTERRUPT CLEARING

Wake-up Indicator is cleared by reading the INT0 register.
RXRDY and RXRDY Time-out is cleared by reading data in the RX FIFO.
RX Line Status interrupt clears after reading the LSR register that is in the UART channel register set.
TXRDY interrupt clears after reading ISR register that is in the UART channel register set.
Modem Status Register interrupt clears after reading MSR register that is in the UART channel register set.
RTS/CTS or DTR/DSR delta interrupt clears after reading MSR register that is in the UART channel register set.
Xoff/Xon delta and special character detect interrupt clears after reading the ISR register that is in the UART channel register set.
TIMER Time-out interrupt clears after reading the TIMERCNTL register that is in the Device Configuration register set.
MPIO interrupt clears after reading the MPIOLVL register that is in the Device Configuration register set.

1.4.2 General Purpose 16-bit Timer/Counter [TIMERMSB, TIMELSB, TIMER, TIMECNTL] (DEFAULT 0xXX-XX-00-00)

The XR17V352 has a general purpose 16-bit timer/counter. The internal 125MHz clock or the external clock at the TMRCK input pin can be selected as the clock source for the timer/counter. The timer can be set to be a single-shot for a one-time event or re-triggerable for a periodic signal. An interrupt may be generated when the timer times out and will show up as a Channel 0 interrupt (see Table 7). It is controlled through 4 configuration registers [TIMERCNTL, TIMER, TIMELSB, TIMERMSB]. The TIMERCNTL register provides the Timer commands such as start/stop, as shown in Table 9 below. The time-out output of the Timer can also be optionally routed to the MPIO[0] pin. The block diagram of the Timer/Counter circuit is shown below:

FIGURE 5. TIMER/COUNTER CIRCUIT

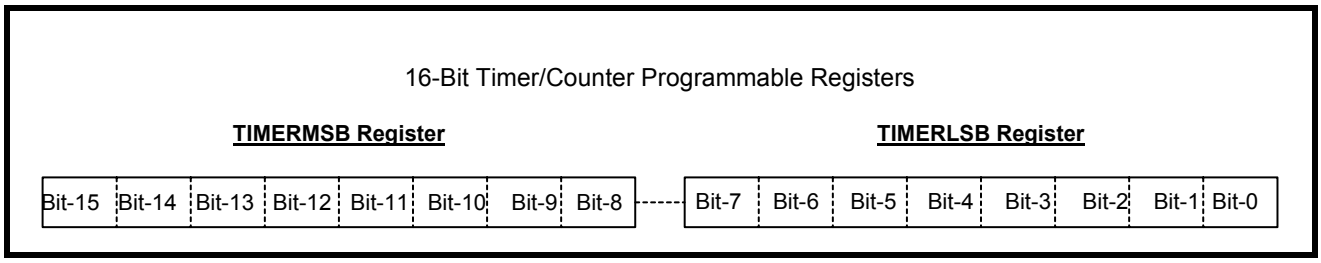


TIMERMSB [31:24] and TIMERLSB [23:16] registers

The concatentaion of the 8-bit registers TIMERMSB and TIMERLSB forms a 16-bit value which decides the time-out period of the Timer, per the following equation:

Timer output frequency = Timer input clock / 16-bit Timer value

The least-significant bit of the timer is being bit [0] of the TIMERLSB with most-significant-bit being bit [7] in TIMERMSB. Notice that these registers do not hold the current counter value when read. Default value is zero (timer disabled) upon powerup and reset. The 'Reset Timer' command does not have any effect on this register.



REGA [15:8] Register

Reserved.

TIMERCNTL [7:0] Register

The bits [3:0] of this register are used to issue commands. The commands are self-clearing, so reading this register does not show the last written command. Reading this register returns a value of 0x01 when the Timer interrupt is enabled and there is a pending Timer interrupt. It returns a value of 0x00 at all other times. The default settings of the Timer, upon power-up, a hardware reset or upon the issue of a 'Timer Reset' command are:

- Timer Interrupt Disabled
- Re-triggerable mode selected
- Internal 125MHz clock selected as clock source
- Timer output not routed to MPIO[0]
- Timer stopped

TABLE 9: TIMER CONTROL REGISTERS

TIMERCNTL [7:4]	Reserved
TIMERCNTL [3:0]	These bits are used to invoke a series of commands that control the function of the Timer/Counter. The commands 1100 to 1111 are reserved. 0001: Enable Timer Interrupt 0010: Disable Timer Interrupt 0011: Select One-shot mode 0100: Select Re-triggerable mode 0101: Select Internal 125MHz clock as clock input for the Timer 0110: Select External Clock input through the TMRCK pin for the Timer 0111: Route Timer output to MPIO[0] pin 1000: De-route Timer output from MPIO[0] 1001: Start Timer 1010: Stop Timer 1011: Reset Timer

TIMER OPERATION

The following paragraphs describe the operation of the 16-bit Timer/Counter. The following conventions will be used in this discussion:

- 'N' is the 16-bit value programmed in the TIMER MSB, LSB registers
- P +Q = N, where 'P' and 'Q' are approximately half of 'N'.
- If N is even, P = Q = N/2.
- If N is odd, P = (N – 1)/2 and Q = (N + 1)/2.
- 'N' can take any value from 0x0002 to 0xFFFF.

Timer Operation in One-Shot Mode:

In the one-shot mode, the Timer output will stay HIGH when started (default state) and will continue to stay HIGH until it times out (reaches the terminal count of 'N' clocks), at which time it will become LOW and stay LOW. If the Timer is re-started before the Timer times out, the counter is reset and the Timer will wait for another time-out period before setting its output LOW (See **Figure 6**). If the Timer times out, re-starting the Timer does not have any effect and a 'Stop Timer' command needs to be issued first which will set the Timer output to its default HIGH state. The Timer must be programmed while it is stopped since the following operations are blocked after the Timer has been started:

- Any write to TIMER MSB, LSB registers
- Issue of any command other than 'Start Timer', 'Stop Timer' and 'Reset Timer'

Timer Operation in Re-triggerable Mode:

In the re-triggerable mode, when the Timer is started, the Timer output will stay HIGH until it reaches half of the terminal count N (= P clocks) and toggle LOW and stay LOW for a similar amount of time (Q clocks). The above step will keep repeating until the Timer is stopped at which time the output will become HIGH (default state). See **Figure 6**. Also, after the Timer is started, re-starting the Timer does not have any effect in re-triggerable mode. The Timer must be programmed while it is stopped since the following operations are blocked when the Timer is running:

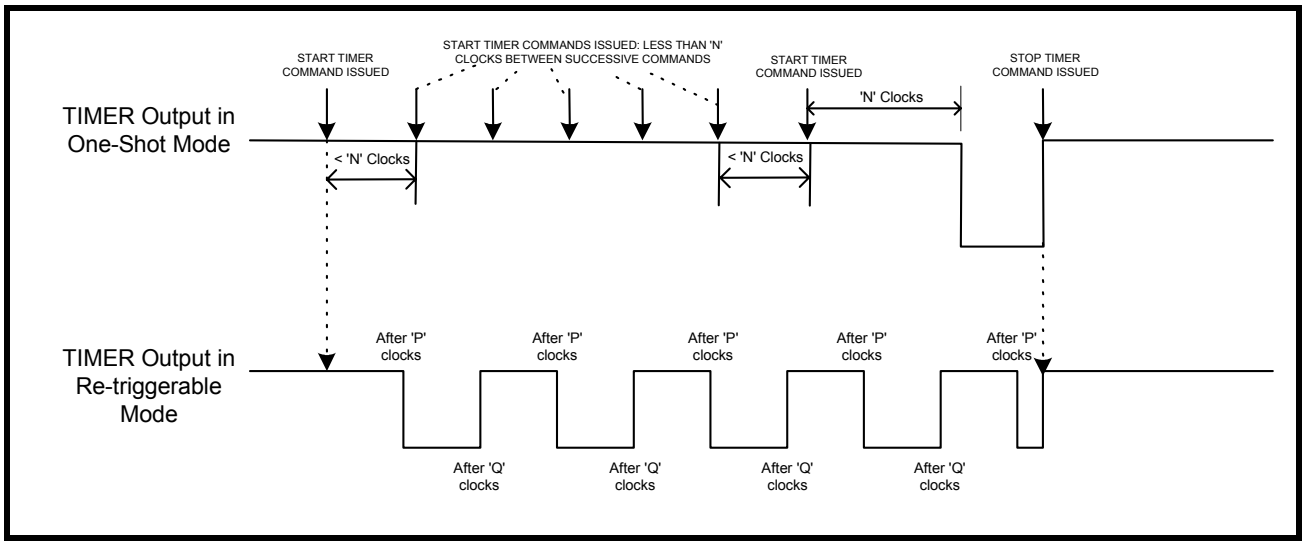
- Any write to TIMER MSB, LSB registers
- Issue of any command other than 'Stop Timer' and 'Reset Timer' ('Start Timer' is not allowed)

Routing the Timer Output to MPIO[0] Pin:

MPIO[0] pin is by default (on power up or reset, for example) an input. However, whenever the Timer output is routed to MPIO[0] pin,

- MPIO[0] will be automatically selected as an output
- MPIO[0] will become HIGH (the default state of Timer output)
- All MPIO control registers (MPIOLVL, MPIOSEL etc) lose control over MPIO[0] and get the control back only when the Timer output is de-routed from MPIO[0].

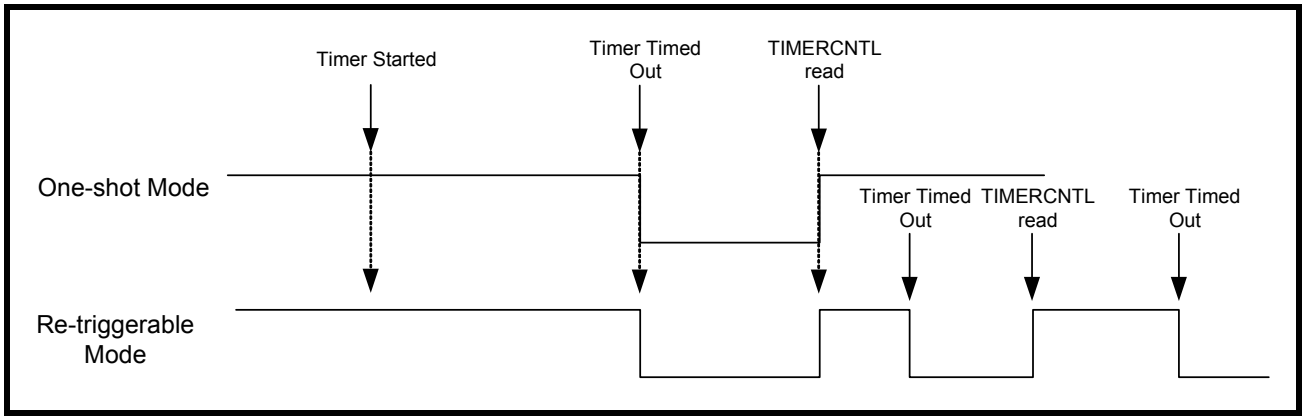
FIGURE 6. TIMER OUTPUT IN ONE-SHOT AND RE-TRIGGERABLE MODES



Timer Interrupt

In the one-shot mode, the Timer will issue an interrupt upon timing out which is 'N' clocks after the Timer is started. In the re-triggerable mode, the Timer will keep issuing an interrupt every 'N' clocks which is on every rising edge of the Timer output. The Timer interrupt can be cleared by reading the TIMERCNTL register or when a Timer Reset command is issued which brings the Timer back to its default settings. The TIMERCNTL will read a value of 0x01 when the Timer interrupt is enabled and there is a pending interrupt. It reads a value of 0x00 at all other times. Stopping the Timer does not clear the interrupt and neither does subsequent re-starting.

FIGURE 7. INTERRUPT OUTPUT (ACTIVE LOW) IN ONE-SHOT AND RE-TRIGGERABLE MODES



1.4.3 8XMODE [7:0] (default 0x00)

Each bit selects 8X or 16X sampling rate for that UART channel. The 8XMODE register is accessible from the Device Configuration Registers in all UART channels but the UART channel can only control the bit for that channel. For example, bit [0] is for channel 0 and can only be controlled by channel 0. All other bits are read-only in channel 0. Logic 0 (default) selects normal 16X sampling (and 4XMODE = 0x00) with logic one selects

8X sampling rate. Transmit and receive data rates will double by selecting 8X. If using the 4XMODE, the corresponding bit in this register should be logic 0

8XMODE Register
Individual UART Channel 8X Clock Mode Enable

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
0	0	0	0	0	0	Ch-1	Ch-0

1.4.4 4XMODE [15:8] (default 0x00)

Each bit selects 4X or 16X sampling rate for that UART channel. The 4XMODE register is accessible from the Device Configuration Registers in all UART channels but the UART channel can only control the bit for that channel. For example, bit [0] is channel 0 and can only be controlled by channel 0. All other bits are read-only in channel 0. Logic 0 (default) selects normal 16X sampling (and 8XMODE = 0x00) with logic one selects 4X sampling rate. Transmit and receive data rates will quadruple by selecting 4X. If using the 8XMODE, the corresponding bit in this register should be logic 0

4XMODE Register
Individual UART Channel 4X Clock Mode Enable

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
0	0	0	0	0	0	Ch-1	Ch-0

RESET [23:16] (default 0x00)

The 8-bit RESET register provides the software with the ability to reset the UART(s) when there is a need. The RESET register is accessible from the Device Configuration Registers in all UART channels but the UART channel can only control the bit for that channel. For example, writing 0xFF to the RESET register in channel 0 will only reset channel 0. Each bit is self-clearing after it is written a logic 1 to perform a reset to that channel. All registers in that channel will be reset to the default condition, see **Table 20** for details. .

RESET Register
Individual UART Channel Reset Enable

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
0	0	0	0	0	0	Ch-1	Ch-0

1.4.5 SLEEP [31:24] (default 0x00)

SLEEP Register
Individual UART Channel Sleep Enable

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
0	0	0	0	0	0	Ch-1	Ch-0

The 8-bit SLEEP register enables each UART separately to enter Sleep mode. The SLEEP register is accessible from the Device Configuration Registers in all UART channels but the UART channel can only control the bit for that channel. For example, writing 0xFF to the SLEEP register in channel 0 will only enable the sleep mode for channel 0.

Sleep mode reduces power consumption when the system needs to put the UART(s) to idle. The UART enters sleep mode when the following conditions are satisfied after the sleep mode is enabled (Logic 0 (default) is to disable and logic 1 is to enable sleep mode):

- Transmitter and Receiver are empty (LSR[6]=1, LSR[0]=0)
- RX pin is idling at a HIGH in normal mode or a LOW in infrared mode
- The modem inputs (CTS#, DSR#, CD# and RI#) are steady at either HIGH or LOW (MSR bits [3:0] = 0x0)

The V352 is awakened by any of the following events occurring at any of the 2 UART channels:

- A receive data start bit transition (HIGH to LOW in normal mode or from LOW to HIGH in infrared mode)
- A data byte is loaded into the transmitter
- A change of logic state on any of the modem inputs so that any of the delta bits (MSR bits[3:0]) is set (RI# delta bit is only set on the rising edge)

A receive data start bit transition will not wake up the UART if the Multidrop mode is disabled (DLD[6] = 0) and the receiver is disabled (MSR[2] = 1, MSR[0] = 0).

A special interrupt is generated with an indication of no pending interrupt. The V352 will return to sleep mode automatically after all interrupting conditions have been serviced and cleared. It will stay in the sleep mode of operation until it is disabled by resetting the SLEEP register bits.

1.4.6 Device Identification and Revision

There are two internal registers that provide device identification and revision, DVID and DREV registers. The 8-bit content in the DVID register provides device identification. A return value of 0x82 from this register indicates the device is a XR17V352. The DREV register returns an 8-bit value of 0x01 for revision A with 0x02 equals to revision B and so on. This information is very useful to the software driver for identifying which device it is communicating with and to keep up with revision changes.

DVID [15:8]

Device identification for the type of UART. The Device ID of the XR17V352 is 0x82.

DREV [7:0]

Revision number of the XR17V352. A 0x01 represents "revision-A" with 0x02 for rev-B and so on.

REGB [23:16] (default 0x00)

REGB register provides a control for simultaneous write to both UARTs configuration register or individually. This is very useful for device initialization in the power up and reset routines. Also, the register provides a facility to interface to the non-volatile memory device such as a 93C46 EEPROM. In embedded applications, the user can use this facility to store proprietary data in an external EEPROM.

1.4.7 REGB Register

REGB[16](Read/Write)	Logic 0 (default) write to each UART configuration registers individually.
	Logic 1 enables simultaneous write to both UARTs configuration register.
REGB[17](Read/Write)	Logic 0 (default) - wake-up interrupt is generated when UART exits sleep mode.
	Logic 1 - No wake-up interrupt is generated when UART exits sleep mode.

1.4.7 REGB Register

REGB[18](Read/Write	Logic 0 (default) - Global interrupt enable. Interrupts to PCI host are enabled.
	Logic 1 - Global interrupt disable. Interrupts to PCI host are disabled.
REGB[19](Read-Only)	Logic 0 - EEPROM load is valid.
	Logic 1 - EEPROM load error caused by one of the following conditions: EEPROM not attached, final bit not found, parity error detected.
REGB[20] (Write-Only)	Control the EECK, clock, output on the EEPROM interface.
REGB[21] (Write-Only)	Control the EECS, chips select, output to the EEPROM device.
REGB[22] (Write-Only)	EEDI data input. Write data to the EEPROM device.
REGB[23] (Read-Only)	EEDO data output. Read data from the EEPROM device.

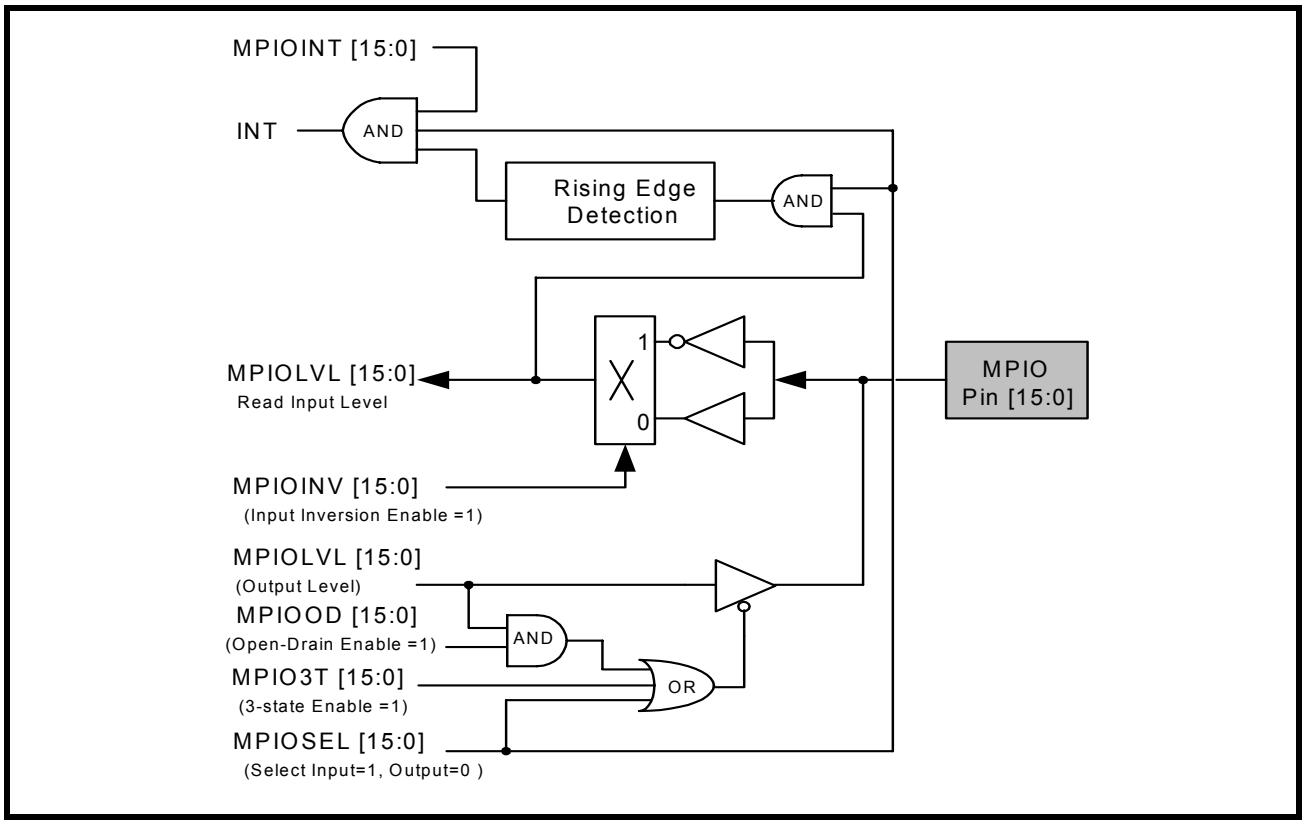
1.4.8 Multi-Purpose Inputs and Outputs

The V352 provides 16 multi-purpose inputs/outputs MPIO[15:0] for general use. Each pin can be programmed to be an input or output function. The input logic state can be set for normal or inverted level, and optionally set to generate an interrupt. The outputs can be set to be normal HIGH or LOW state, 3-state, or open drain. Their functions and definitions are programmed through 6 registers: MPIOINT, MPIOLVL, MPIO3T, MPIOINV, MPIOSEL, and MPIOOD. If all 16 pins are set for inputs, all 16 interrupts would be ORed together. The ORed interrupt is reported in the channel 0 UART interrupt status, see Interrupt Status Register. The pins may also be programmed to be outputs and to the 3-state condition for signal sharing. The MPIO[0] pin can be programmed to show the Timer output. When it is programmed to be the Timer output, all the above 5 registers lose control over the MPIO[0] pin. For details on Timer output, please see **“Section 1.4.2, General Purpose 16-bit Timer/ Counter [TIMERMSB, TIMELSB, TIMER, TIMECNTL] (default 0xXX-XX-00-00)” on page 18**.

1.4.9 MPIO REGISTERS

There are 2 sets of 6 registers that select, control and monitor the 16 multipurpose inputs and outputs. Figure 8 shows the internal circuitry.

FIGURE 8. MULTIPURPOSE INPUT/OUTPUT INTERNAL CIRCUIT



MPIOINT [15:0] (default 0x00)

The MPIOINT register enables the multipurpose input pin interrupt. If an MPIO pin is selected by MPIOSEL as an input, then it can be selected to generate an interrupt. MPIOINT bit[0] enables input pin MPIO0 for interrupt, and bit [7] enables input pin 7. No interrupt is enable if the pin is selected to be an output. The interrupt is edge sensing and determined by MPIOINV and MPIOLVL registers. The MPIO interrupt clears after a read to register MPIOLVL. The combination of MPIOLVL and MPIOINV determines the interrupt being active LOW or active HIGH. Logic 0 (default) disables the pin's interrupt and logic 1 enables it.

MPIOINT Register
Multipurpose Input/Output Interrupt Enable

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
MPIO7	MPIO6	MPIO5	MPIO4	MPIO3	MPIO2	MPIO1	MPIO0

MPIOLVL [15:0] (default 0x00)

The MPIOLVL register controls the output pins and provides the input level status for the input pins. The status of the input pin(s) is read on this register and output pins are controlled on this register. A logic 0 (default) sets the output to LOW and a logic 1 sets the output pin to HIGH. The MPIO interrupt will clear upon reading this register.

MPIOLVL Register
Multipurpose Output Level Control

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
MPIO7	MPIO6	MPIO5	MPIO4	MPIO3	MPIO2	MPIO1	MPIO0

MPIO3T [15:0] (default 0x00)

The MPIO outputs can be tri-stated by the MPIO3T register. A logic 0 (default) sets the output to active level per register MPIOBIT settling, a logic 1 sets the output pin to tri-state.

MPIO3T Register
Multipurpose Output 3-state Enable

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
MPIO7	MPIO6	MPIO5	MPIO4	MPIO3	MPIO2	MPIO1	MPIO0

MPIOINV [15:0] (default 0x00)

The MPIO inputs can be inverted by the MPIOINV register. A logic 0 (default) does not invert the input pin logic. A logic 1 inverts the input logic level.

MPIOINV Register
Multipurpose Input Signal Inversion Enable

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
MPIO7	MPIO6	MPIO5	MPIO4	MPIO3	MPIO2	MPIO1	MPIO0

MPIOSEL [15:0](default 0xFF)

The MPIOSEL register defines the MPIOs as either an input or output. A logic 1 (default) defines the pin for input and a logic 0 for output.

MPIOSEL Register
Multipurpose Input/Output Selection

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
MPIO7	MPIO6	MPIO5	MPIO4	MPIO3	MPIO2	MPIO1	MPIO0

MPIOOD [15:0] (default 0x00)

The MPIO outputs can behave as an open-drain output by the MPIOOD register. When the MPIOOD register is a logic 0 (default), the MPIO is not an open-drain output. A logic 1 enables the MPIO as an open-drain output. This register has no effect, when the MPIO is an input.

MPIOOD Register
Multipurpose Open-Drain Output Enable

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
MPIO7	MPIO6	MPIO5	MPIO4	MPIO3	MPIO2	MPIO1	MPIO0

2.0 TRANSMIT AND RECEIVE DATA

There are two methods to load transmit data and unload receive data from each UART channel. First, there is a transmit data register and receive data register for each UART channel as shown in Table 4 set to ease programming. These registers support 8, 16, 24 and 32 bits wide format. In the 32-bit format, it increases the data transfer rate on the PCI bus. Additionally, a special register location provides receive data byte with its associated error flags. This is a 16-bit or 32-bit read operation where the Line Status Register (LSR) content in the UART channel register is paired along with the data byte. This operation further facilitates data unloading with the error flags without having to read the LSR register separately. Furthermore, the XR17V352 supports PCI burst mode for read/write operation of up to 256 bytes of data.

The second method is through each UART channel's transmit holding register (THR) and receive holding register (RHR). The THR and RHR registers are 16550 compatible so their access is limited to 8-bit format. The software driver must separately read the LSR content for the associated error flags before reading the data byte.

2.1 FIFO DATA LOADING AND UNLOADING IN 32-BIT FORMAT

The XR17V352 supports PCI Burst Read and PCI Burst Write transactions anywhere in the mapped memory region (except reserved areas). In addition, to utilize this feature fully, the device provides a separate memory location (apart from the individual channel's register set) where the RX and the TX FIFO can be read from/written to, as shown in Table 4. The following is an extract from the table showing the memory locations that support burst transactions:

Channel N: (for channels 0 through 1) where M = 4N + 1.

RX FIFO	:	0xM00 - 0xMFF (256 bytes)
TX FIFO	:	0xM00 - 0xMFF (256 bytes)
RX FIFO + status	:	0x(M+1)0 - 0x(M+2)FF (256 bytes data + 256 bytes status)

For example, the locations for channel 1 are:

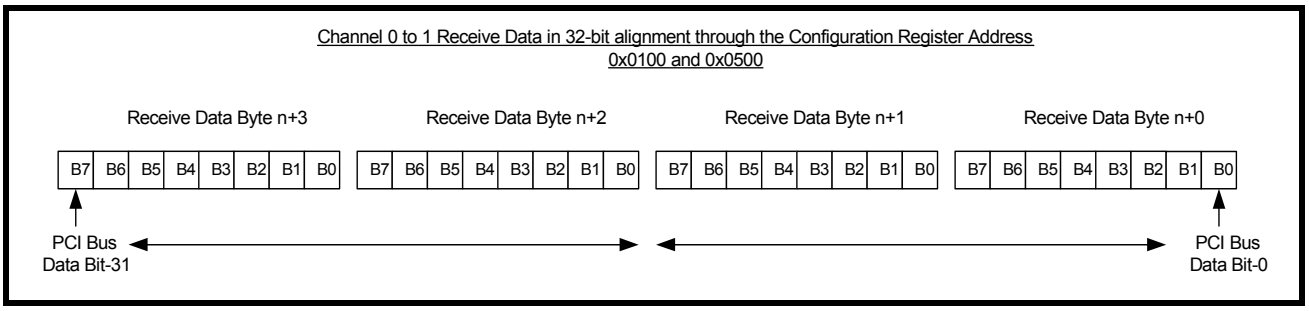
Channel 1:

RX FIFO	:	0x0500 - 0x05FF (256 bytes)
TX FIFO	:	0x0500 - 0x05FF (256 bytes)
RX FIFO + status	:	0x0600 - 0x07FF (256 bytes data + 256 bytes status)

2.1.1 Normal Rx FIFO Data Unloading at locations 0x100 and 0x500

The RX FIFO data (up to the maximum 256 bytes) can be read out in a single burst 32-bit read operation (maximum 16 DWORD reads) at memory locations 0x100 (channel 0) and 0x500 (channel 1). This operation is at least 16 times faster than reading the data in 256 separate 8-bit memory reads of RHR register (0x000 for channel 0 and 0x400 for channel 1).

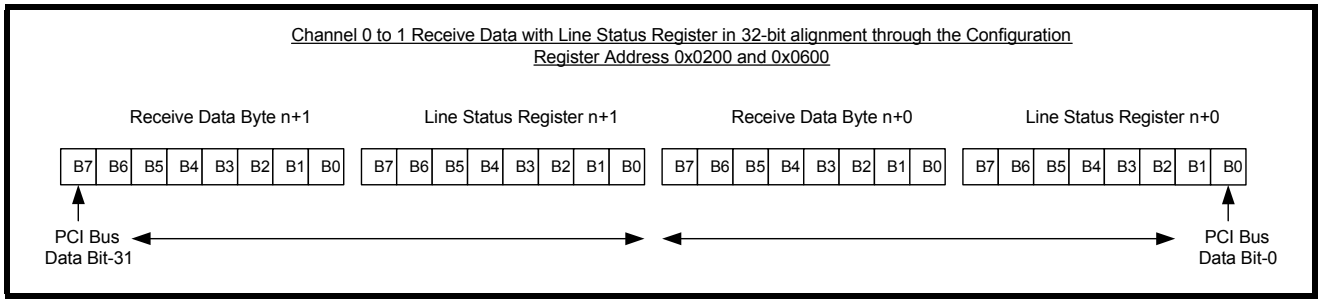
READ RX FIFO, WITH NO ERRORS	BYTE 3	BYTE 2	BYTE 1	BYTE 0
Read n+0 to n+3	FIFO Data n+3	FIFO Data n+2	FIFO Data n+1	FIFO Data n+0
Read n+4 to n+7	FIFO Data n+7	FIFO Data n+6	FIFO Data n+5	FIFO Data n+4
Etc.				



2.1.2 Special Rx FIFO Data Unloading at locations 0x0200 and 0x0600

The XR17V352 also provides the same RX FIFO data along with the LSR status information of each byte side-by-side, at locations 0x0200 (channel 0) and 0x0600 (channel 1). The entire RX data along with the status can be downloaded in a single PCI Burst Read operation of 32 DWORD reads. The Status and Data bytes must be read in 16 or 32 bits format to maintain data integrity. The following tables show this clearly.

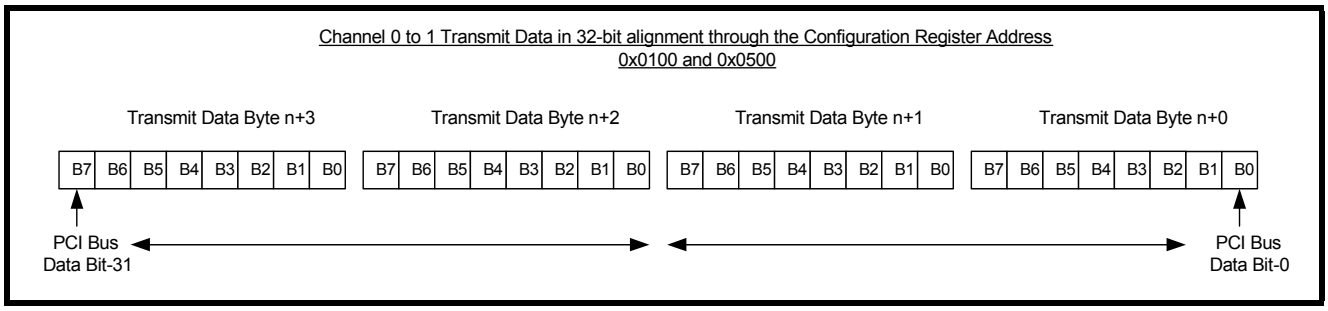
READ RX FIFO, WITH LSR ERRORS	BYTE 3	BYTE 2	BYTE 1	BYTE 0
Read n+0 to n+1	FIFO Data n+1	LSR n+1	FIFO Data n+0	LSR n+0
Read n+2 to n+3	FIFO Data n+3	LSR n+3	FIFO Data n+2	LSR n+2
Etc				



2.1.3 Tx FIFO Data Loading at locations 0x100 and 0x500

The TX FIFO data (up to the maximum 256 bytes) can be loaded in a single burst 32-bit write operation (maximum 16 DWORD writes) at memory locations 0x0100 (channel 0) and 0x0500 (channel 1).

WRITE TX FIFO	BYTE 3	BYTE 2	BYTE 1	BYTE 0
Write n+0 to n+3	FIFO Data n+3	FIFO Data n+2	FIFO Data n+1	FIFO Data n+0
Write n+4 to n+7	FIFO Data n+7	FIFO Data n+6	FIFO Data n+5	FIFO Data n+4
Etc.				



2.2 FIFO DATA LOADING AND UNLOADING THROUGH THE UART CHANNEL REGISTERS, THR AND RHR IN 8-BIT FORMAT

The THR and RHR register address for channel 0 to channel 1 is shown in **Table 10** below. The THR and RHR for each channel 0 to 1 are located sequentially at address 0x0000 and 0x0200. Transmit data byte is loaded to the THR when writing to that address and receive data is unloaded from the RHR register when reading that address. Both THR and RHR registers are 16C550 compatible in 8-bit format, so each bus operation can only write or read in bytes.

TABLE 10: TRANSMIT AND RECEIVE DATA REGISTER IN BYTE FORMAT, 16C550 COMPATIBLE

THR and RHR Address Locations For CH0 to CH1 (16C550 Compatible)

CH0 0x0000 Read THR	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
CH0 0x0000 Write RHR	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
CH1 0x4000 Read THR	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
CH1 0x4000 Write RHR	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0

3.0 UART

There are 2 UARTs channel [1:0] in the V352. Each has its own 256-byte of transmit and receive FIFO, a set of 16550 compatible control and status registers, and a baud rate generator for individual channel data rate setting. Eight additional registers per UART were added for the EXAR enhanced features.

3.1 Programmable Baud Rate Generator with Fractional Divisor

Each UART has its own Baud Rate Generator (BRG) with a prescaler for the transmitter and receiver. The prescaler is controlled by a software bit in the MCR register. The MCR register bit [7] sets the prescaler to divide the internal 125MHz clock by 1 or 4. The output of the prescaler clocks to the BRG. The BRG further divides this clock by a programmable divisor between 1 and (2^{16} - 0.0625) in increments of 0.0625 (1/16) to obtain a 16X, 8X or 4X sampling clock of the serial data rate. The sampling clock is used by the transmitter for data bit shifting and receiver for data sampling.

The BRG divisor (DLL, DLM and DLD registers) defaults to 1 (DLL = 0x01, DLM = 0x00, DLD = 0x00). The DLL and DLM registers provide the integer part of the divisor and the DLD register provides the fractional part of the divisor. Only the four lower bits of the DLD are implemented and they are used to select a value from 0 (for setting 0000) to 0.9375 or 15/16 (for setting 1111). Programming the Baud Rate Generator Registers DLL, DLM and DLD provides the capability for selecting the operating data rate. **Table 11** shows the divisor for some standard and non-standard data rates when using the internal 125MHz clock at 16X clock rate. If the pre-scaler is used (MCR bit [7] = 1), the output data rate will be 4 times less than that shown in **Table 11**. At 8X sampling rate, these data rates would double. At 4X sampling rate, these data rates would quadruple. Also, when using 8X or 4X sampling mode, note that the bit-time will have a jitter (+/- 1/16) whenever the DLD is an odd number. For data rates not listed in **Table 11**, the divisor value can be calculated with the following equation(s):

Required Divisor (decimal) = (125MHz clock frequency / prescaler) / (serial data rate x 16),
WITH 8XMODE =0 AND 4XMODE = 0

Required Divisor (decimal) = (125MHz clock frequency / prescaler / (serial data rate x 8),
WITH 8XMODE = 1 AND 4XMODE = 0

Required Divisor (decimal) = (125MHz clock frequency / prescaler / (serial data rate x 4),
WITH 8XMODE = 0 AND 4XMODE = 1

The closest divisor that is obtainable in the V352 can be calculated using the following formula:

ROUND((Required Divisor - TRUNC (Required Divisor))*16)/16 + TRUNC (Required Divisor), where

DLM = TRUNC(Required Divisor) >> 8
DLL = TRUNC (Required Divisor) & 0xFF
DLD = ROUND ((Required Divisor-TRUNC(Required Divisor))*16)

In the formulas above, please note that:

TRUNC (N) = Integer Part of N. For example, TRUNC (5.6) = 5.

ROUND (N) = N rounded towards the closest integer. For example, ROUND (7.3) = 7 and ROUND (9.9) = 10.

FIGURE 9. BAUD RATE GENERATOR

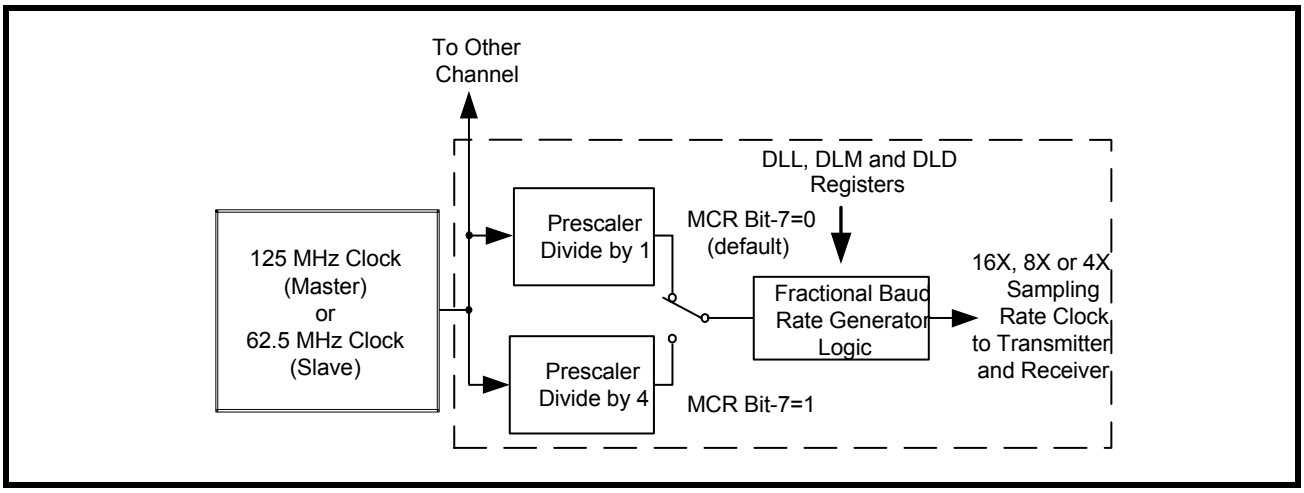


TABLE 11: TYPICAL DATA RATES WITH INTERNAL 125MHZ CLOCK AT 16X SAMPLING

REQUIRED OUTPUT DATA RATE	DIVISOR FOR 16x Clock (Decimal)	DIVISOR OBTAINABLE IN V352	DLM PROGRAM VALUE (HEX)	DLL PROGRAM VALUE (HEX)	DLD PROGRAM VALUE (HEX))	DATA ERROR RATE (%)
2400	3255.21	3255 3/16	0C	B7	3	0
4800	1627.60	1627 9/16	06	5B	9	0
9600	813.80	813 12/16	03	2D	C	0.01
10000	781.25	781 4/16	03	0D	4	0
19200	406.90	406 14/16	01	96	E	0.01
25000	312.5	312 8/16	01	38	8	0
28800	271.27	271 4/16	01	0F	4	0.01
38400	203.45	203 7/16	00	CB	7	0.01
50000	156.25	156 4/16	00	9C	4	0
57600	135.63	135 10/16	00	87	A	0.01
75000	104.17	104 2/16	00	68	2	0.04
100000	78.125	78 2/16	00	4E	2	0
115200	67.82	67 13/16	00	43	D	0.01
153600	50.86	50 13/16	00	32	D	0.10
200000	39.06	39 1/16	00	27	1	0
225000	34.72	34 11/16	00	22	B	0.10
230400	33.91	33 14/16	00	21	E	0.10
250000	31.25	31 4/16	00	1F	4	0
300000	26.04	26	00	1A	0	0.16
400000	19.53	19 8/16	00	13	8	0.16
460800	16.95	16 15/16	00	10	F	0.10
500000	15.625	15 10/16	00	0F	A	0
576000	13.56	13 9/16	00	0D	9	0.01
750000	10.42	10 6/16	00	0A	6	0.40
921600	8.48	8 7/16	00	08	7	0.47
1000000	7.81	7 13/16	00	07	D	0
1152000	6.78	6 12/16	00	06	C	0.47

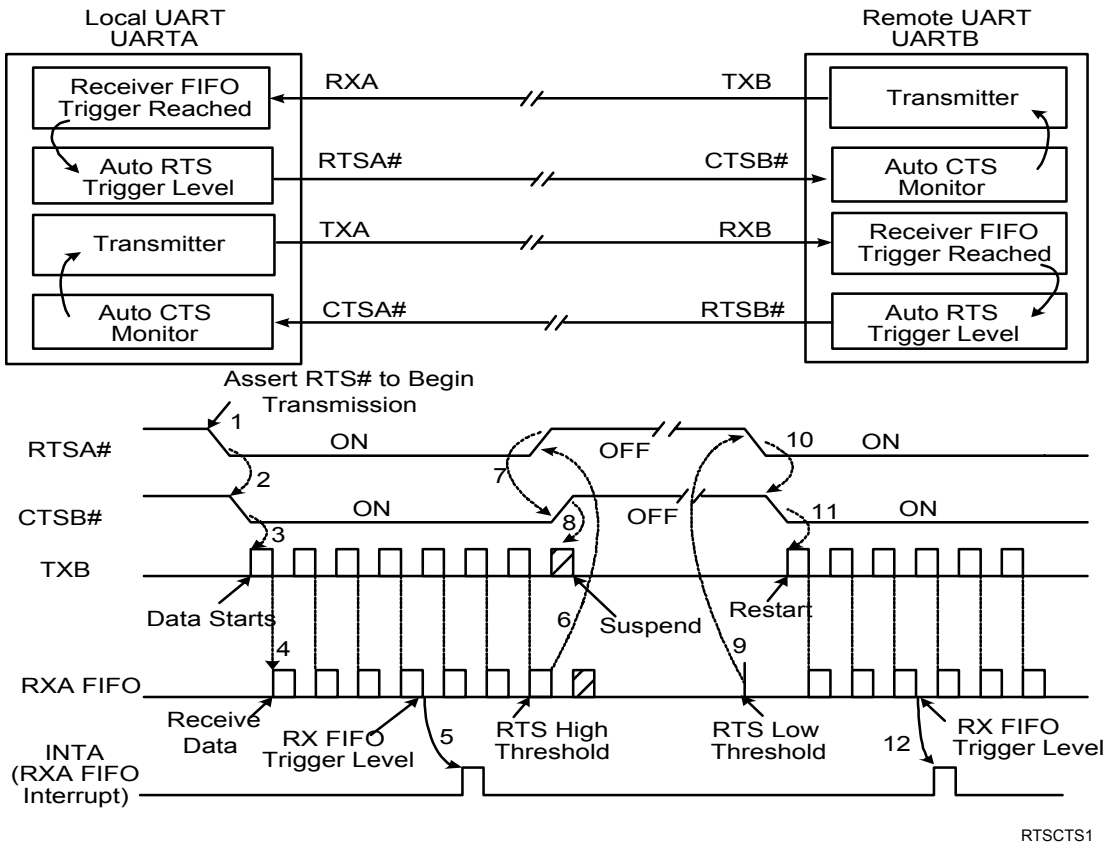
3.2 Automatic Hardware (RTS/CTS or DTR/DSR) Flow Control Operation

Automatic hardware or RTS/DTR and CTS/DSR flow control is used to prevent data overrun to the local receiver FIFO and remote receiver FIFO. The RTS#/DTR# output pin is used to request remote unit to suspend/restart data transmission while the CTS#/DSR# input pin is monitored to suspend/restart local transmitter. The auto RTS/DTR and auto CTS/DSR flow control features are individually selected to fit specific application requirement and enabled through EFR bit[7:6] and MCR bit [2] for either RTS/CTS or DTR/DSR control signals. The auto RTS/DTR function must be started by asserting RTS/DTR# output pin (MCR bit [0] or bit [1] to logic 1) after it is enabled. **Figure 10** below explains how it works.

Two interrupts associated with RTS/DTR and CTS/DSR flow control have been added to give indication when RTS/DTR# pin or CTS/DSR# pin is de-asserted during operation. The RTS/DTR and CTS/DSR interrupts must be first enabled by EFR bit [4], and then enabled individually by IER bits [7:6], and chosen with MCR bit [2].

Automatic hardware flow control is selected by setting bits [7 (CTS): 6 (RTS)] of the EFR register to logic 1. If CTS# pin transitions from LOW to HIGH indicating a flow control request, ISR bit [5] will be set to logic 1, (if enabled via IER bit [7:6]), and the UART will suspend TX transmissions as soon as the stop bit of the character in process is shifted out. Transmission is resumed after the CTS# input returns to LOW, indicating more data may be sent.

FIGURE 10. AUTO RTS/DTR AND CTS/DSR FLOW CONTROL OPERATION



The local UART (UARTA) starts data transfer by asserting -RTSA# (1). RTSA# is normally connected to CTSB# (2) of remote UART (UARTB). CTSB# allows its transmitter to send data (3). TXB data arrives and fills UARTA receive FIFO (4). When RXA data fills up to its receive FIFO trigger level, UARTA activates its RXA data ready interrupt (5) and continues to receive and put data into its FIFO. If interrupt service latency is long and data is not being unloaded, UARTA monitors its receive data fill level to match the upper threshold of RTS delay and de-assert RTSA# (6). CTSB# follows (7) and request UARTB transmitter to suspend data transfer. UARTB stops or finishes sending the data bits in its transmit shift register (8). When receive FIFO data in UARTA is unloaded to match the lower threshold of RTS delay (9), UARTA re-asserts RTSA# (10), CTSB# recognizes the change (11) and restarts its transmitter and data flow again until next receive FIFO trigger (12). This same event applies to the reverse direction when UARTA sends data to UARTB with RTSB# and CTSA# controlling the data flow.

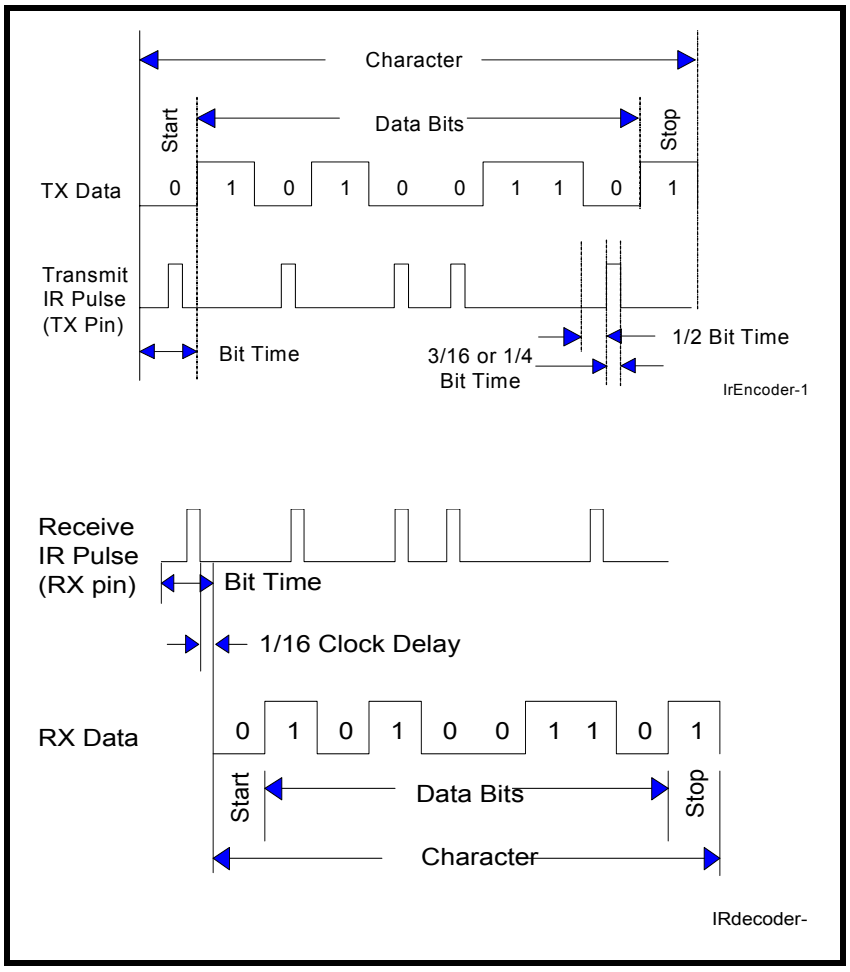
3.3 Infrared Mode

Each UART in the V352 includes the infrared encoder and decoder compatible to the IrDA (Infrared Data Association) version 1.1. The input pin ENIR conveniently activates both UART channels to start up in the infrared mode. This global control pin enables the MCR bit [6] function in every UART channel register. After power up or a reset, the software can overwrite MCR bit [6] if so desired. ENIR and MCR bit [6] also disable its receiver while the transmitter is sending data. This prevents the echoed data from going to the receiver. The global activation ENIR pin prevents the infrared emitter from turning on and drawing large amount of current while the system is starting up. When the infrared feature is enabled, the transmit data outputs, TX[1:0], would idle LOW. Likewise, the RX [1:0] inputs assume a LOW idle level.

The infrared encoder sends out a 3/16 of a bit wide pulse for each “0” bit in the transmit data stream. This signal encoding reduces the on-time of the infrared LED, hence reduces the power consumption. See **Figure 11** below. Typical max data rate for the infrared encoder with a 3/16 of a bit wide pulse is 115.2 kbps. For data rates above 115.2 kbps and up to1.152 Mbps, Fast IR mode can be enabled via DLD bit-4 for a 1/4 of bit wide pulse. For exact 3/16 or 1/4 of a bit wide pulse, the 16X sampling rate should be used and DLD[3:0] = ’0000’. The IR pulse width can vary if DLD[3:0] is not ’0000’.

The infrared decoder receives the input pulse from the infrared sensing diode on RX pin. Each time the decoder senses a light pulse, it returns a "0" to the data bit stream. The RX input signal may be inverted prior delivered to the input of the decoder via internal register setting. This option supports active LOW instead of normal active HIGH pulse from some infrared modules on the market.

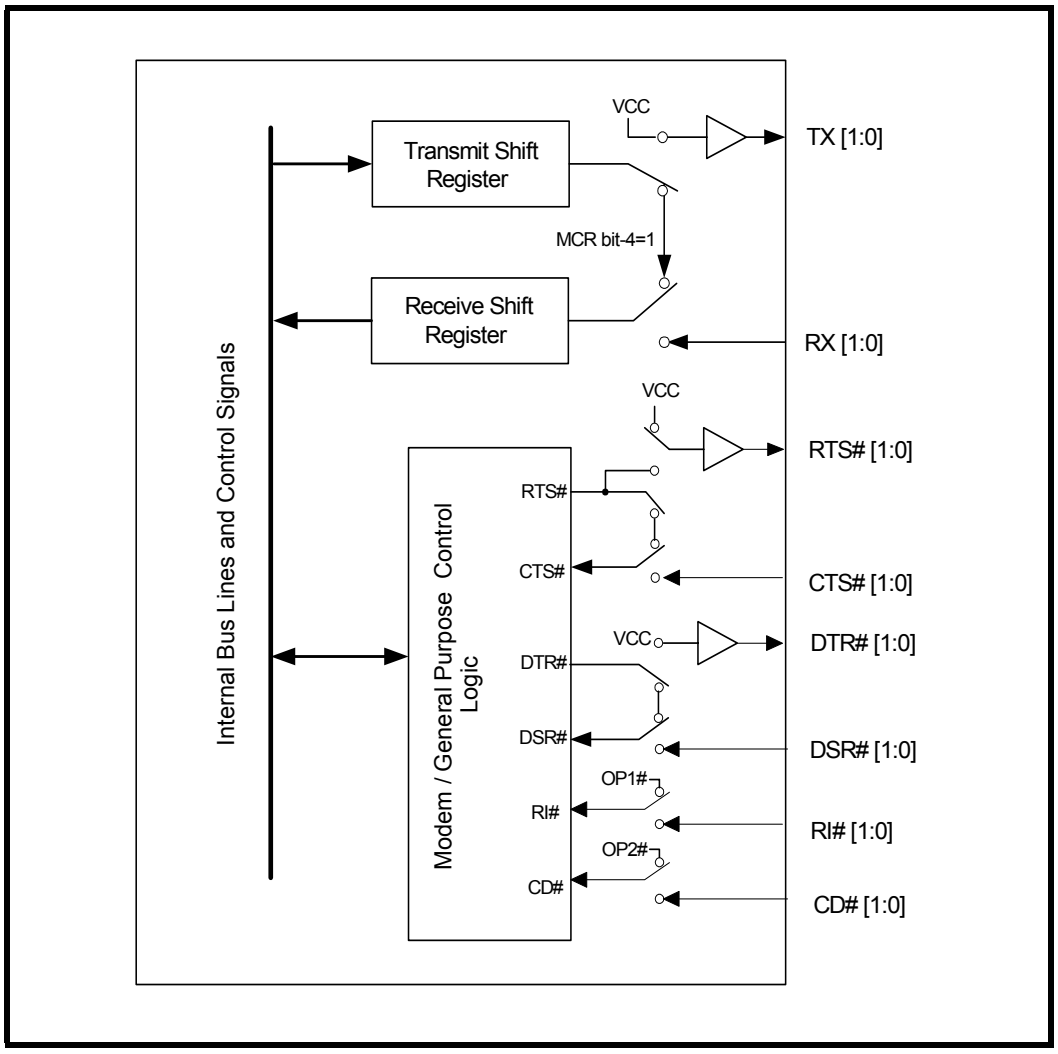
FIGURE 11. INFRARED TRANSMIT DATA ENCODING AND RECEIVE DATA DECODING



3.4 Internal Loopback

Each UART channel provides an internal loopback capability for system diagnostic. The internal loopback mode is enabled by setting MCR register bit [4] to a logic 1. All regular UART functions operate normally. Figure 12 shows how the modem port signals are re-configured. Transmit data from the transmit shift register output is internally routed to the receive shift register input allowing the system to receive the same data that it was sending. The TX pin is held at HIGH or mark condition while RTS# and DTR# are de-asserted. The CTS#, DSR#, CD# and RI# inputs are ignored.

FIGURE 12. INTERNAL LOOP BACK



3.5 UART CHANNEL CONFIGURATION REGISTERS

Address lines A0 to A3 select the 16 registers in each channel. The first 8 registers are 16550 compatible with EXAR enhanced feature registers located on the upper 8 addresses.

TABLE 12: UART CHANNEL CONFIGURATION REGISTERS

ADDRESS	REGISTER	READ/WRITE	COMMENTS
A3 A2 A1 A0			
16550 COMPATIBLE			
0 0 0 0	RHR - Receive Holding Register THR - Transmit Holding Register	Read-only Write-only	LCR[7] = 0
0 0 0 0	DLL - Divisor LSB	Read/Write	LCR[7] = 1
0 0 0 1	DLM - Divisor MSB	Read/Write	LCR[7] = 1
0 0 1 0	DLD - Divisor Fractional	Read/Write	LCR[7] = 1
0 0 0 1	IER - Interrupt Enable Register	Read/Write	LCR[7] = 0
0 0 1 0	ISR - Interrupt Status Register FCR - FIFO Control Register	Read-only Write-only	LCR[7] = 0
0 0 1 1	LCR - Line Control Register	Read/Write	
0 1 0 0	MCR - Modem Control Register	Read/Write	
0 1 0 1	LSR - Line Status Register	Read-only	
0 1 1 0	MSR - Modem Status Register - Auto RS485 Delay	Read-only Write-only	 EFR bit-4 = 1
0 1 1 1	SPR - Scratch Pad Register	Read/Write	
ENHANCED REGISTER			
1 0 0 0	FCTR - Feature Control Register	Read/Write	
1 0 0 1	EFR - Enhanced Function Register	Read/Write	
1 0 1 0	TXCNT - Transmit FIFO Level Counter TXTRG - Transmit FIFO Trigger Level	Read-only Write-only	
1 0 1 1	RXCNT - Receive FIFO Level Counter RXTRG - Receive FIFO Trigger Level	Read-only Write-only	
1 1 0 0	Xoff-1 - Xoff Character 1 Xchar	Write-only Read-only	Xon,Xoff Rcvd. Flags
1 1 0 1	Xoff-2 - Xoff Character 2	Write-only	
1 1 1 0	Xon-1 - Xon Character 1	Write-only	
1 1 1 1	Xon-2 - Xon Character 2	Write-only	

TABLE 13: UART CHANNEL CONFIGURATION REGISTERS DESCRIPTION. SHADED BITS ARE ENABLED BY EFR BIT-4.

ADDRESS A3-A0	REG NAME	READ/ WRITE	BIT [7]	BIT [6]	BIT [5]	BIT [4]	BIT [3]	BIT [2]	BIT[1]	BIT [0]	COMMENT
0 0 0 0	RHR	R	BIT [7]	BIT [6]	Bit [5]	Bit [4]	Bit [3]	Bit [2]	Bit [1]	Bit [0]	LCR[7]=0
0 0 0 0	THR	W	BIT [7]	BIT [6]	Bit [5]	Bit [4]	Bit [3]	Bit [2]	Bit [1]	Bit [0]	LCR[7]=0
0 0 0 0	DLL	R/W	BIT [7]	BIT [6]	Bit [5]	Bit [4]	Bit [3]	Bit [2]	Bit [1]	Bit [0]	LCR[7]=1
0 0 0 1	DLM	R/W	BIT [7]	BIT [6]	Bit [5]	Bit [4]	Bit [3]	Bit [2]	Bit [1]	Bit [0]	LCR[7]=1
0 0 1 0	DLD	R/W	Invert RS485 Polarity	Multi-drop mode	XON/ XOFF parity check	Fast IR mode	Bit [3]	Bit [2]	Bit [1]	Bit [0]	LCR[7]=1
0 0 0 1	IER	R/W	0/ CTS/ DSR# Int. Enable	0/ RTS/ DTR# Int. Enable	0/ Xon/ Xoff/Sp. Char. Int. Enable	0	Modem Status Int. Enable	RX Line Status Int. Enable	TX Empty Int. Enable	RX Data Int. Enable	LCR[7]=0
0 0 1 0	ISR	R	FIFOs Enable	FIFOs Enable	0/ Delta-Flow Cntl	0/ Xoff/special char	INT Source Bit [3]	INT Source Bit [2]	INT Source Bit [1]	INT Source Bit [0]	LCR[7]=0
0 0 1 0	FCR	W	RXFIFO Trigger	RXFIFO Trigger	0/ TXFIFO Trigger	0/ TX FIFO Trigger	DMA Mode	TX FIFO Reset	RX FIFO Reset	FIFOs Enable	LCR[7]=0
0 0 1 1	LCR	R/W	Divisor Enable	Set TX Break	Set Parity	Even Parity	Parity Enable	Stop Bits	Word Length Bit [1]	Word Length Bit [0]	
0 1 0 0	MCR	R/W	0/ BRG Prescaler	0/ IR Enable	0/ XonAny	Internal Loopback Enable	(OP2)[1] TX char Immediate	(OP1)[1] RTS/ DTR Flow Sel	RTS# Pin Control	DTR# Pin Control	
0 1 0 1	LSR	R	RX FIFO Error	TSR Empty	THR Empty	RX Break	RX Framing Error	RX Parity Error	RX Overrun	RX Data Ready	
0 1 1 0	MSR	R	CD	RI	DSR	CTS	Delta CD#	Delta RI#	Delta DSR#	Delta CTS#	
	MSR	W	RS485 DLY[3]	RS485 DLY[2]	RS485 DLY[1]	RS485 DLY[0]	Disable TX	Disable RX	Disable TX mode	Disable RX mode	
0 1 1 1	SPR	R/W	Bit [7]	Bit [6]	Bit [5]	Bit [4]	Bit [3]	Bit [2]	Bit [1]	Bit [0]	User Data
1 0 0 0	FCTR	R/W	TRG Table Bit [1]	TRG Table Bit [0]	Auto RS485 Enable	Invert IR RX Input	RTS/ DTR Hyst Bit [3]	RTS/ DTR Hyst Bit [2]	RTS/ DTR Hyst Bit [1]	RTS/ DTR Hyst Bit [0]	

TABLE 13: UART CHANNEL CONFIGURATION REGISTERS DESCRIPTION. SHADED BITS ARE ENABLED BY EFR BIT-4.

ADDRESS A3-A0	REG NAME	READ/ WRITE	BIT [7]	BIT [6]	BIT [5]	BIT [4]	BIT [3]	BIT [2]	BIT[1]	BIT [0]	COMMENT
1 0 0 1	EFR	R/W	Auto CTS/ DSR Enable	Auto RTS/ DTR Enable	Special Char Select	Enable IER [7:5], ISR [5:4], FCR[5:4], MCR[7:5], MSR	Soft-ware Flow Cntl Bit [3]	Soft-ware Flow Cntl Bit [2]	Soft-ware Flow Cntl Bit [1]	Software Flow Cntl Bit [0]	
1 0 1 0	TXCNT	R	Bit [7]	Bit [6]	Bit [5]	Bit [4]	Bit [3]	Bit [2]	Bit [1]	Bit [0]	
1 0 1 0	TXTRG	W	Bit [7]	Bit [6]	Bit [5]	Bit [4]	Bit [3]	Bit [2]	Bit [1]	Bit [0]	
1 0 1 1	RXCNT	R	Bit [7]	Bit [6]	Bit [5]	Bit [4]	Bit [3]	Bit [2]	Bit [1]	Bit [0]	
1 0 1 1	RXTRG	W	Bit [7]	Bit [6]	Bit [5]	Bit [4]	Bit [3]	Bit [2]	Bit [1]	Bit [0]	
1 1 0 0	XCHAR	R	0	0	0	0	TX Xon Indicator	TX Xoff Indicator	Xon Det. Indicator	Xoff Det. Indicator	Self clear after read
1 1 0 0	XOFF1	W	Bit [7]	Bit [6]	Bit [5]	Bit [4]	Bit [3]	Bit-2	Bit [1]	Bit [0]	
1 1 0 1	XOFF2	W	Bit [7]	Bit [6]	Bit [5]	Bit [4]	Bit [3]	Bit-2	Bit [1]	Bit [0]	
1 1 1 0	XON1	W	Bit [7]	Bit [6]	Bit [5]	Bit [4]	Bit [3]	Bit-2	Bit [1]	Bit [0]	
1 1 1 1	XON2	W	Bit [7]	Bit [6]	Bit [5]	Bit [4]	Bit [3]	Bit-2	Bit [1]	Bit [0]	

NOTE: *MCR bits [3:2] (OP1 and OP2 outputs) are not available in the XR17V352. They are present for 16C550 compatibility during Internal loopback, see* **Figure 12**.

3.6 Transmitter

The transmitter section comprises of a 256 bytes of FIFO, a byte-wide Transmit Holding Register (THR) and an 8-bit Transmit Shift Register (TSR). THR receives a data byte from the host (non-FIFO mode) or a data byte from the FIFO when the FIFO is enabled by FCR bit [0]. TSR shifts out every data bit with the 16X or 8X internal clock. A bit time is 16 or 8 clock periods. The transmitter sends the start bit followed by the number of data bits, inserts the proper parity bit if enable, and adds the stop bit(s). The status of the THR and TSR are reported in the Line Status Register (LSR bit [6:5]).

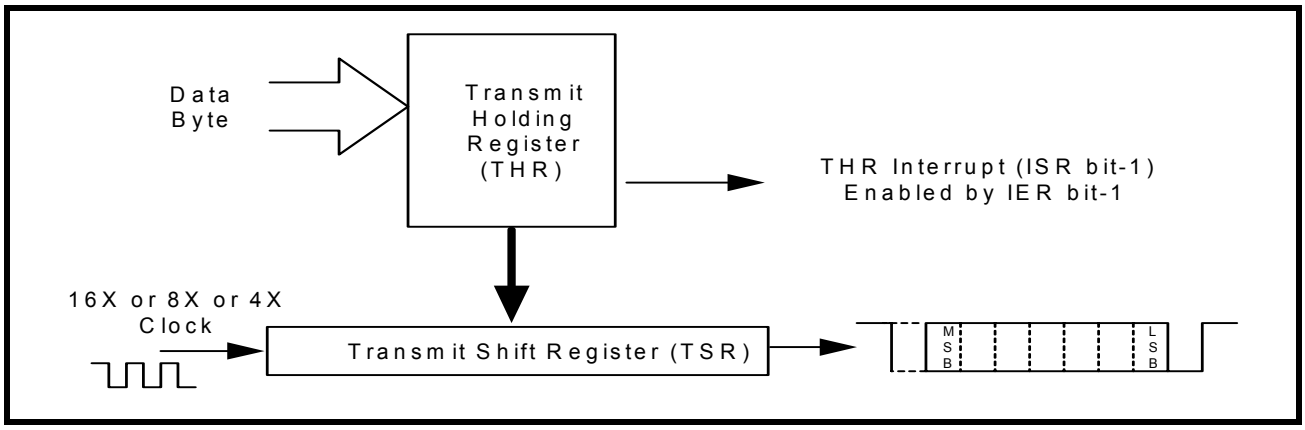
3.6.1 Transmit Holding Register (THR)

The transmit holding register is an 8-bit register providing a data interface to the host processor. The host writes transmit data byte to the THR to be converted into a serial data stream including start-bit, data bits, parity-bit and stop-bit(s). The least-significant-bit (bit [0]) becomes first data bit to go out. The THR is also the input register to the transmit FIFO of 256 bytes when FIFO operation is enabled by FCR bit[0]. A THR empty interrupt can be generated when it is enabled in IER bit [1].

3.6.2 Transmitter Operation in non-FIFO Mode

The host loads transmit data to THR one character at a time. The THR empty flag (LSR bit [5]) is set when the data byte is transferred to TSR. THR flag can generate a transmit empty interrupt (ISR bit [1]) when it is enabled by IER bit [1]. The TSR flag (LSR bit [6]) is set when TSR becomes completely empty.

FIGURE 13. TRANSMITTER OPERATION IN NON-FIFO MODE



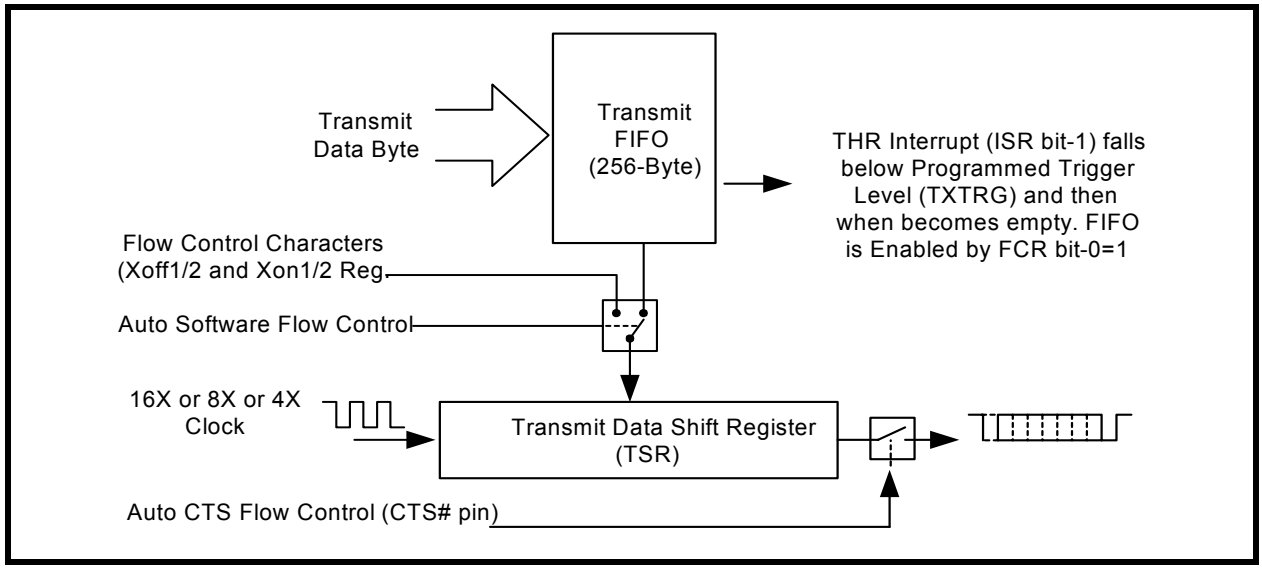
3.6.3 Transmitter Operation in FIFO Mode

The host may fill the transmit FIFO with up to 256 bytes of transmit data. The THR empty flag (LSR bit [5]) is set whenever the FIFO is empty. The THR empty flag can generate a transmit empty interrupt (ISR bit [1]) when the amount of data in the FIFO falls below its programmed trigger level (see TXTRG register). The transmit empty interrupt is enabled by IER bit [1]. The TSR flag (LSR bit [6]) is set when TSR becomes completely empty. Furthermore, with the RS485 half-duplex direction control enabled (FCTR bit [5]=1) the source of the transmit empty interrupt changes to TSR empty instead of THR empty. This is to ensure the RTS# output is not changed until the last stop bit of the last character is shifted out.

3.6.4 Auto RS485 Operation

The auto RS485 half-duplex direction control changes the behavior of the transmitter when enabled by FCTR bit [5]. It de-asserts RTS# or DTR# after a specified delay indicated in MSR[7:4] following the last stop bit of the last character that has been transmitted. This helps in turning around the transceiver to receive the remote station's response. The delay optimizes the time needed for the last transmission to reach the farthest station on a long cable network before switching off the line driver. This delay prevents undesirable line signal disturbance that causes signal degradation. It also changes the transmitter empty interrupt to TSR empty instead of THR empty.

FIGURE 14. TRANSMITTER OPERATION IN FIFO AND FLOW CONTROL MODE

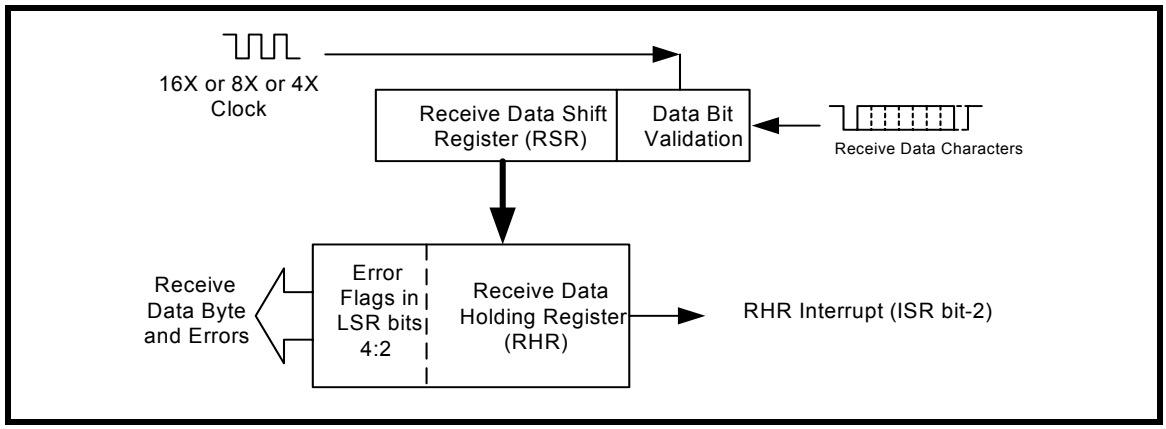


3.7 Receiver

The receiver section contains an 8-bit Receive Shift Register (RSR) and Receive Holding Register (RHR). The RSR uses the 16X, 8X or 4X clock for timing. It verifies and validates every bit on the incoming character in the middle of each data bit. On the falling edge of a start or false start bit, an internal receiver counter starts counting at the 16X, 8X or 4X clock rate. After 8 or 4 or 2 clocks the start bit period should be at the center of the start bit. At this time the start bit is sampled and if it is still a logic 0 it is validated. Evaluating the start bit in this manner prevents the receiver from assembling a false character. The rest of the data bits and stop bits are sampled and validated in this same manner to prevent false framing. If there were any error(s), they are reported in the LSR register bits [4:1]. Upon unloading the receive data byte from RHR, the receive FIFO pointer is bumped and the error flags are immediately updated to reflect the status of the data byte in RHR register. RHR can generate a receive data ready interrupt upon receiving a character or delay until it reaches the FIFO trigger level. Furthermore, data delivery to the host is guaranteed by a receive data ready time-out function when receive data does not reach the receive FIFO trigger level. This time-out delay is 4 word lengths as defined by LCR bits [1:0] plus 12 bits time. The RHR interrupt is enabled by IER bit [0].

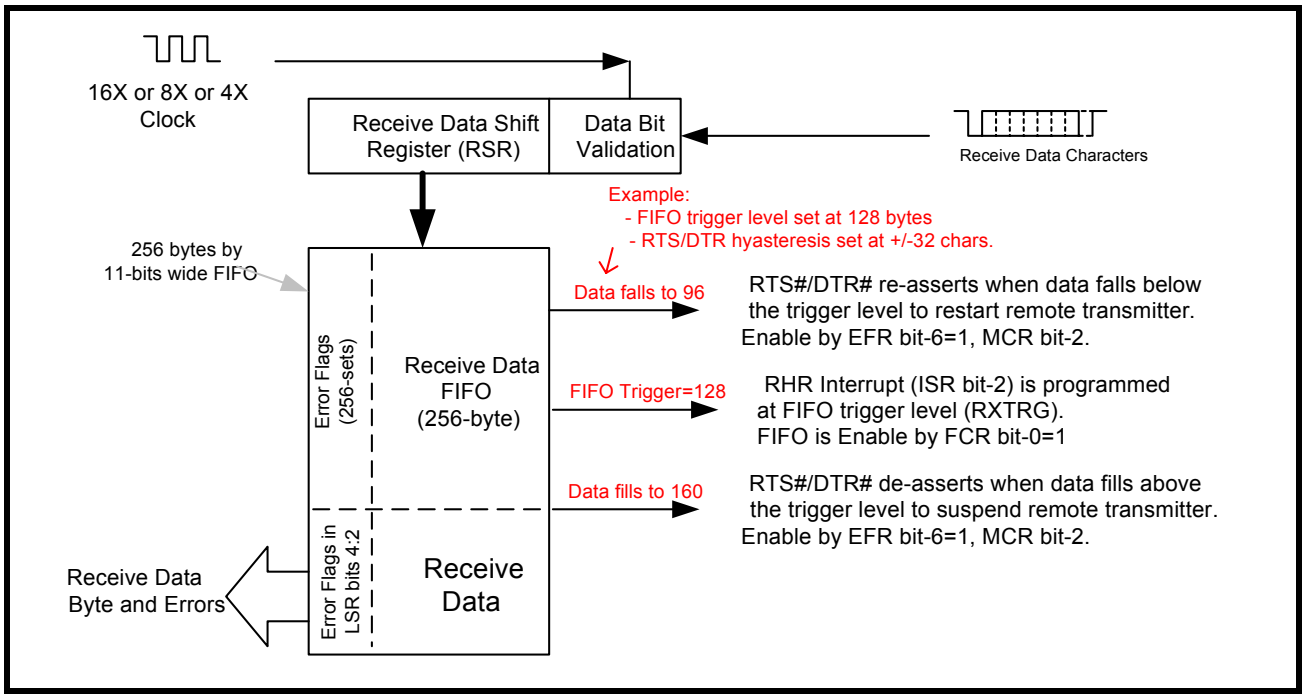
3.7.1 Receiver Operation in non-FIFO Mode

FIGURE 15. RECEIVER OPERATION IN NON-FIFO MODE



3.7.2 Receiver Operation with FIFO

FIGURE 16. RECEIVER OPERATION IN FIFO AND FLOW CONTROL MODE



3.7.3 Normal Multidrop (9-bit) Mode

Normal multidrop mode is enabled when DLD[6] = 1 and EFR[5] = 0 (Special Character Detect disabled). The receiver is set to Force Parity 0 (LCR[5:3] = '111') in order to detect address bytes.

With the receiver initially disabled (MSR[2] = 1), it ignores all the data bytes (parity bit = 0) until an address byte is received (parity bit = 1). This address byte will cause the UART to set the parity error. The UART will generate an LSR interrupt and place the address byte in the RX FIFO. The software then examines the byte and enables the receiver if the address matches its slave address, otherwise, it does not enable the receiver.

If the receiver has been enabled, the receiver will receive the subsequent data. If an address byte is received, it will generate an LSR interrupt. The software again examines the byte and if the address matches its slave address, it does not have to do anything. If the address does not match its slave address, then the receiver should be disabled.

3.7.4 Auto Address Detection Mode

Auto address detection mode is enabled when DLD[6] = 1 and EFR bit-5 = 1 (Special Character Detect enabled). The receiver is set to Force Parity 0 (LCR[5:3] = '111') in order to detect address bytes. The desired slave address will need to be written into the XOFF2 register. The receiver will monitor all incoming address bytes and compare with the programmed character in the XOFF2 register. If the received byte is a data byte or an address byte that does not match the programmed character in the XOFF2 register, the receiver will discard the data. Upon receiving an address byte that matches the XOFF2 character, the receiver will be automatically enabled if not already enabled, and the address character is pushed into the RX FIFO along with the parity bit (in place of the parity error bit). The receiver also generates an LSR interrupt. The receiver will then receive the subsequent data. If another address byte is received and this address does not match the programmed XOFF2 character, then the receiver will automatically be disabled and all subsequent data is ignored until there is another address byte match with XOFF2.

4.0 UART CONFIGURATION REGISTERS

4.1 Receive Holding Register (RHR) - Read only

SEE"RECEIVER" ON PAGE 42.

4.2 Transmit Holding Register (THR) - Write only

SEE"TRANSMITTER" ON PAGE 40.

4.3 Baud Rate Generator Divisors (DLM, DLL and DLD)

DLM[7:0], DLL[7:0] and DLD[3:0]

The Baud Rate Generator (BRG) generates the data rate for the transmitter and receiver. The rate is programmed through registers DLM, DLL and DLD which are only accessible when LCR bit [7] is set to logic 1. Refer to **"Section 3.1, Programmable Baud Rate Generator with Fractional Divisor" on page 31** for more details.

DLD[7]: RS-485 Polarity

- Logic 0 = The Auto RS-485 Half-duplex direction control pin will be HIGH for TX and LOW for RX.
- Logic 1 = The Auto RS-485 Half-duplex direction control pin will be LOW for TX and HIGH for RX.

DLD[6]: Multi-drop Mode

- Logic 0 = Normal mode.
- Logic 1 = Enable Multi-drop mode.

DLD[5]: XON/XOFF Parity Check

- Logic 0 = XON/XOFF characters are valid flow control characters even if they have parity errors.
- Logic 1 = XON/XOFF characters are not valid flow control characters if they have parity errors.

DLD[4]: Fast IR Mode

- Logic 0 = If IR mode is enabled, IR pulsewidth will be 3/16th of bit time.
- Logic 1 = If IR mode is enabled, IR pulsewidth will be 1/4th of bit time.

4.4 Interrupt Enable Register (IER) - Read/Write

The Interrupt Enable Register (IER) masks the interrupts from receive data ready, transmit empty, line status and modem status registers. These interrupts are reported in the Interrupt Status Register (ISR) and also encoded in INT (INT0-INT3) register in the Device Configuration Registers.

4.4.1 IER versus Receive FIFO Interrupt Mode Operation

When the receive FIFO (FCR bit [0] = logic 1) and receive interrupts (IER bit [0] = logic 1) are enabled, the RHR interrupts (see ISR bits [4:3]) status will reflect the following:

A. The receive data available interrupts are issued to the host when the FIFO has reached the programmed trigger level. It will be cleared when the FIFO drops below the programmed trigger level.

B. FIFO level will be reflected in the ISR register when the FIFO trigger level is reached. Both the ISR register status bit and the interrupt will be cleared when the FIFO drops below the trigger level.

C. The receive data ready bit (LSR bit [0]) is set as soon as a character is transferred from the shift register to the receive FIFO. It is reset when the FIFO is empty.

4.4.2 IER versus Receive/Transmit FIFO Polled Mode Operation

When FCR bit [0] equals a logic 1 for FIFO enable; resetting IER bits [3:0] enables the XR16V352 in the FIFO polled mode of operation. Since the receiver and transmitter have separate bits in the LSR either can be used in the polled mode by selecting respective transmit or receive control bit(s).

A. LSR BIT-0 indicates there is data in RHR (non-FIFO mode) or RX FIFO (FIFO mode).

B. LSR BIT-1 indicates an overrun error has occurred and that data in the FIFO may not be valid.

C. LSR BIT 2-4 provides the type of receive data errors encountered for the data byte in RHR, if any.

D. LSR BIT-5 indicates THR (non-FIFO mode) or TX FIFO (FIFO mode) is empty.

E. LSR BIT-6 indicates when both the transmit FIFO and TSR are empty.

F. LSR BIT-7 indicates a data error in at least one character in the RX FIFO.

IER[7]: CTS# Input Interrupt Enable (requires EFR bit [4]=1)

- Logic 0 = Disable the CTS# interrupt (default).
- Logic 1 = Enable the CTS# interrupt. The UART issues an interrupt when CTS# pin makes a transition from LOW to HIGH.

IER[6]: RTS# Output Interrupt Enable (requires EFR bit [4]=1)

- Logic 0 = Disable the RTS# interrupt (default).
- Logic 1 = Enable the RTS# interrupt. The UART issues an interrupt when RTS# pin makes a transition from LOW to HIGH.

IER[5]: Xoff Interrupt Enable (requires EFR bit [4]=1)

- Logic 0 = Disable the software flow control, receive Xoff interrupt (default).
- Logic 1 = Enable the software flow control, receive Xoff interrupt. See Software Flow Control section for details.

IER[4]: Reserved

IER[3]: Modem Status Interrupt Enable

The Modem Status Register interrupt is issued whenever any of the delta bits of the MSR register (bits [3:0]) is set.

- Logic 0 = Disable the modem status register interrupt (default).
- Logic 1 = Enable the modem status register interrupt.

IER[2]: Receive Line Status Interrupt Enable

An Overrun error, Framing error, Parity error or detection of a Break character will result in an LSR interrupt. The V352 will issue an LSR interrupt immediately after receiving a character with an error. It will again re-issue the interrupt (if the first one has been cleared by reading the LSR register) when the character with the error is on the top of the FIFO, meaning the next one to be read out of the FIFO.

For example, let's consider an incoming data stream of 0x55, 0xAA, etc. and that the character 0xAA has a Parity error associated with it. Let's assume that the character 0x55 has not been read out of the FIFO yet. The V352v352 will issue an interrupt as soon as the stop bit of the character 0xAA is received. The LSR register will have only the FIFO error bit (bit [7]) set and none of the other error bits (bits [4:1]) will be set, since the byte on the top of the FIFO is 0x55 which does not have any errors associated with it. When this byte has been read out, the V352 will issue another LSR interrupt and this time the LSR register will show the Parity bit (bit [2]) set.

- Logic 0 = Disable the receiver line status interrupt (default).
- Logic 1 = Enable the receiver line status interrupt.

IER[1]: TX Ready Interrupt Enable

In non-FIFO mode, a TX interrupt is issued whenever the THR is empty. In the FIFO mode, an interrupt is issued twice: once when the number of bytes in the TX FIFO falls below the programmed trigger level and again when the TX FIFO becomes empty. When autoRS485 mode is enabled (FCTR bit [5] = 1), the second interrupt is delayed until the transmitter (both the TX FIFO and the TX Shift Register) is empty.

- Logic 0= Disable Transmit Ready Interrupt (default).
- Logic 1 = Enable Transmit Ready Interrupt.

IER[0]: RX Interrupt Enable

The receive data ready interrupt will be issued when RHR has a data character in the non-FIFO mode or when the receive FIFO has reached the programmed trigger level in the FIFO mode.

- Logic 0 = Disable the receive data ready interrupt (default).
- Logic 1 = Enable the receiver data ready interrupt.

4.5 Interrupt Status Register (ISR) - Read Only

The UART provides multiple levels of prioritized interrupts to minimize external software interaction. The Interrupt Status Register (ISR) provides the user with six interrupt status bits. Performing a read cycle on the ISR will give the user the current highest pending interrupt level to be serviced, others queue up for next service. No other interrupts are acknowledged until the pending interrupt is serviced. The Interrupt Source Table, Table 14, shows the data values (bit [5:0]) for the six prioritized interrupt levels and the interrupt sources associated with each of these interrupt levels.

4.5.1 Interrupt Generation:

- LSR is by any of the LSR bits [4:1]. See IER bit [2] description on the previous page.
- RXRDY is by RX trigger level.
- RXRDY Time-out is by a 4-char plus 12 bits delay timer.
- TXRDY is by TX trigger level or TX FIFO empty (or transmitter empty in auto RS-485 control).
- MSR is by any of the MSR bits [3:0].
- Receive Xoff/Xon/Special character is by detection of a Xoff, Xon or Special character.
- CTS#/DSR# is when its transmitter toggles the input pin (from LOW to HIGH) during auto CTS/DSR flow control enabled by EFR bit [7] and selection on MCR bit [2].
- RTS#/DTR# is when its receiver toggles the output pin (from LOW to HIGH) during auto RTS/DTR flow control enabled by EFR bit [6] and selection on MCR bit [2].
- Wake-up indicator is when the UART wakes up from the sleep mode.

4.5.2 Interrupt Clearing:

- LSR interrupt is cleared by a read to the LSR register.
- RXRDY interrupt is cleared by reading data until FIFO falls below the trigger level.
- RXRDY Time-out interrupt is cleared by reading RHR.
- TXRDY interrupt is cleared by a read to the ISR register or writing to THR.
- MSR interrupt is cleared by a read to the MSR register.
- Xoff/Xon interrupt is cleared by reading ISR.
- Special character interrupt is cleared by a read to ISR.
- RTS#/DTR# and CTS#/DSR# status change interrupts are cleared by a read to the MSR register.

- Wake-up indicator is cleared by a read to the INT0 register.

TABLE 14: INTERRUPT SOURCE AND PRIORITY LEVEL

PRIORITY LEVEL	ISR REGISTER STATUS BITS						SOURCE OF THE INTERRUPT
	BIT [5]	BIT [4]	BIT [3]	BIT [2]	BIT [1]	BIT [0]	
1	0	0	0	1	1	0	LSR (Receiver Line Status Register)
2	0	0	0	1	0	0	RXRDY (Received Data Ready)
3	0	0	1	1	0	0	RXRDY (Receive Data Time-out)
4	0	0	0	0	1	0	TXRDY (Transmitter Holding Register Empty)
5	0	0	0	0	0	0	MSR (Modem Status Register)
6	0	1	0	0	0	0	RXRDY (Received Xon/Xoff or Special character)
7	1	0	0	0	0	0	CTS#/DSR#, RTS#/DTR# change of state
X	0	0	0	0	0	1	None (default)

ISR[7:6]: FIFO Enable Status

These bits are set to a logic 0 when the FIFOs are disabled. They are set to a logic 1 when the FIFOs are enabled.

ISR[5:1]: Interrupt Status

These bits indicate the source for a pending interrupt at interrupt priority levels (See Table 14). See "Section 4.5.1, Interrupt Generation:" on page 46 and "Section 4.5.2, Interrupt Clearing:" on page 46 for details.

ISR[0]: Interrupt Status

- Logic 0 = An interrupt is pending and the ISR contents may be used as a pointer to the appropriate interrupt service routine.
- Logic 1 = No interrupt pending. (default condition)

4.6 FIFO Control Register (FCR) - Write Only

This register is used to enable the FIFOs, clear the FIFOs, set the transmit/receive FIFO trigger levels, and select the DMA mode. The DMA, and FIFO modes are defined as follows:

FCR[7:6]: Receive FIFO Trigger Select

(logic 0 = default, RX trigger level =1)

The FCTR bits [5:4] are associated with these 2 bits. These 2 bits are used to set the trigger level for the receive FIFO. The UART will issue a receive interrupt when the number of the characters in the FIFO crosses the trigger level. Table 15 shows the complete selections. Note that the receiver and the transmitter cannot use different trigger tables. Whichever selection is made last applies to both the RX and TX side.

FCR[5:4]: Transmit FIFO Trigger Select (requires EFR bit [4]=1)

(logic 0 = default, TX trigger level = 1)

The FCTR bits [7:6] are associated with these 2 bits by selecting one of the four tables. The 4 user selectable trigger levels in 4 tables are supported for compatibility reasons. These 2 bits set the trigger level for the transmit FIFO interrupt. The UART will issue a transmit interrupt when the number of characters in the FIFO falls below the selected trigger level, or when it gets empty in case that the FIFO did not get filled over the trigger level on last re-load. Table 15 below shows the selections.

FCR[3]: DMA Mode Select

This bit has no effect since TXRDY and RXRDY pins are not available in this device. It is provided for legacy software compatibility.

- Logic 0 = Set DMA to mode 0 (default).
- Logic 1 = Set DMA to mode 1.

FCR[2]: TX FIFO Reset

This bit is only active when FCR bit [0] is active.

- Logic 0= No transmit FIFO reset (default).
- Logic 1 = Reset the transmit FIFO pointers and FIFO level counter logic (the transmit shift register is not cleared or altered). This bit will return to a logic 0 after resetting the FIFO.

FCR[1]: RX FIFO Reset

This bit is only active when FCR bit [0] is active.

- Logic 0 = No receive FIFO reset (default).
- Logic 1 = Reset the receive FIFO pointers and FIFO level counter logic (the receive shift register is not cleared or altered). This bit will return to a logic 0 after resetting the FIFO.

FCR[0]: TX and RX FIFO Enable

- Logic 0 = Disable the transmit and receive FIFO (default).
- Logic 1 = Enable the transmit and receive FIFOs. This bit must be set to logic 1 when other FCR bits are written or they will not be programmed.

TABLE 15: TRANSMIT AND RECEIVE FIFO TRIGGER TABLE AND LEVEL SELECTION

TRIGGER TABLE	FCTR BIT [7]	FCTR BIT [6]	FCR BIT [7]	FCR BIT [6]	FCR BIT [5]	FCR BIT [4]	RECEIVE TRIGGER LEVEL	TRANSMIT TRIGGER LEVEL	COMPATIBILITY
Table-A	0	0			0	0		1 (default)	16C550, 16C2550, 16C2552, 16C554, 16C580, 16L580
			0	0			1 (default)		
			0	1			4		
			1	0			8		
			1	1			14		
Table-B	0	1			0	0		16	16C650A, 16L651
					0	1		8	
					1	0		24	
					1	1		30	
			0	0			8		
			0	1			16		
			1	0			24		
			1	1			28		

TABLE 15: TRANSMIT AND RECEIVE FIFO TRIGGER TABLE AND LEVEL SELECTION

TRIGGER TABLE	FCTR BIT [7]	FCTR BIT [6]	FCR BIT [7]	FCR BIT [6]	FCR BIT [5]	FCR BIT [4]	RECEIVE TRIGGER LEVEL	TRANSMIT TRIGGER LEVEL	COMPATIBILITY
Table-C	1	0			0	0		8	16C654
					0	1		16	
					1	0		32	
					1	1		56	
			0	0			8		
			0	1			16		
			1	0			56		
			1	1			60		
Table-D	1	1	X	X	X	X	Programmable via RXTRG register	Programmable via TXTRG register	16L2752, 16L2750, 16C2852, 16C850, 16C854, 16C864

4.7 Line Control Register (LCR) - Read/Write

The Line Control Register is used to specify the asynchronous data communication format. The word or character length, the number of stop bits, and the parity are selected by writing the appropriate bits in this register.

LCR[7]: Baud Rate Divisors Enable

Baud rate generator divisor (DLL, DLM, DLD) enable.

- Logic 0 = Data registers are selected (default).
- Logic 1 = Divisor latch registers (DLL, DLM and DLD) are selected.

LCR[6]: Transmit Break Enable

When enabled the Break control bit causes a break condition to be transmitted (the TX output is forced to a “space", LOW, state). This condition remains until disabled by setting LCR bit [6] to a logic 0.

- Logic 0 = No TX break condition. (default)
- Logic 1 = Forces the transmitter output (TX) to a “space”, LOW, for alerting the remote receiver of a line break condition.

LCR[5]: TX and RX Parity Select

If the parity bit is enabled, LCR bit [5] selects the forced parity format.

- LCR bit [5] = logic 0, parity is not forced (default).
- LCR bit [5] = logic 1 and LCR bit [4] = logic 0, parity bit is forced to a logical 1for the transmit and receive data.
- LCR bit [5] = logic 1 and LCR bit [4] = logic 1, parity bit is forced to a logical 0 for the transmit and receive data.

TABLE 16: PARITY PROGRAMMING

LCR BIT [5]	LCR BIT [4]	LCR BIT [3]	PARITY SELECTION
X	X	0	No parity
0	0	1	Odd parity
0	1	1	Even parity
1	0	1	Force parity to mark, “1”
1	1	1	Forced parity to space, “0”

LCR[4]: TX and RX Parity Select

If the parity bit is enabled with LCR bit [3] set to a logic 1, LCR bit [4] selects the even or odd parity format.

- Logic 0 = ODD Parity is generated by forcing an odd number of logic 1’s in the transmitted character. The receiver must be programmed to check the same format (default).
- Logic 1 = EVEN Parity is generated by forcing an even the number of logic 1’s in the transmitted character. The receiver must be programmed to check the same format.

LCR[3]: TX and RX Parity Select

Parity or no parity can be selected via this bit. The parity bit is a simple way used in communications for data integrity check. See Table 16 above for parity selection summary.

- Logic 0 = No parity.
- Logic 1 = A parity bit is generated during the transmission while the receiver checks for parity error of the data character received.

LCR[2]: TX and RX Stop-bit Length Select

The length of stop bit is specified by this bit in conjunction with the programmed word length.

BIT [2]	WORD LENGTH	STOP BIT LENGTH (BIT TIME(S))
0	5,6,7,8	1 (default)
1	5	1-1/2
1	6,7,8	2

LCR[1:0]: TX and RX Word Length Select

These two bits specify the word length to be transmitted or received.

BIT [1]	BIT [0]	WORD LENGTH
0	0	5 (default)
0	1	6
1	0	7
1	1	8

4.8 Modem Control Register (MCR) - Read/Write

The MCR register is used for controlling the modem interface signals or general purpose inputs/outputs.

MCR[7]: Clock Prescaler Select (requires EFR bit [4]=1)

- Logic 0 = Divide by one. The internal 125MHz clock is fed directly to the Programmable Baud Rate Generator without further modification, i.e., divide by one (default).
- Logic 1 = Divide by four. The prescaler divides the internal 125MHz clock by 4 and feeds it to the Programmable Baud Rate Generator, hence, data rates become one forth.

MCR[6]: Infrared Encoder/Decoder Enable (requires EFR bit [4]=1)

The state of this bit depends on the sampled logic level of pin ENIR during power up, following a hardware reset (rising edge of RST# input). Afterward user can override this bit for desired operation.

- Logic 0 = Enable the standard modem receive and transmit character interface.
- Logic 1 = Enable infrared IrDA receive and transmit inputs/outputs. While in this mode, the TX/RX output/input are routed to the infrared encoder/decoder. The data input and output levels will conform to the IrDA infrared interface requirement. As such, while in this mode the infrared TX output will be a LOW during idle data conditions. FCTR bit [4] may be selected to invert the RX input signal level going to the decoder for infrared modules that provide rather an inverted output. For exact 3/16 or 1/4 bit wide pulse, the 16X sampling rate must be used and DLD[3:0] = '0000'. If DLD[3:0] is not '0000', the pulse width can vary.

MCR[5]: Xon-Any Enable (requires EFR bit [4]=1)

- Logic 0 = Disable Xon-Any function (default).
- Logic 1 = Enable Xon-Any function. In this mode any RX character received will enable Xon, resume data transmission.

MCR[4]: Internal Loopback Enable

- Logic 1 = Disable loopback mode (default).
- Logic 1 = Enable local loopback mode, see loopback section and **Figure 12**.

MCR[3]: Send Char Immediate (OP2 in Local Loopback Mode)

This bit is used to transmit a character immediately irrespective of the bytes currently in the transmit FIFO. The data byte must be loaded into the transmit holding register (THR) immediately following the write to this bit (to set it to a '1'). In other words, no other register must be accessed between setting this bit and writing to the THR. The loaded byte will be transmitted ahead of all the bytes in the TX FIFO, immediately after the character currently being shifted out of the transmit shift register is sent out. The existing line parameters (parity, stop bits) will be used when composing the character. This bit is self clearing, therefore, must be set before sending a custom character each time. Please note that the Transmitter must be enabled for this function (MSR[3] = 0). Also, if software flow control is enabled, the software flow control characters (Xon, Xoff) have higher priority and will get shifted out before the custom byte is transmitted.

- Logic 0 = Send Char Immediate disabled (default).
- Logic 1 = Send Char Immediate enabled.

In Local Loopback Mode (MCR[4] = 1), this bit acts as the legacy OP2 output and controls the CD bit in the MSR register as shown in **Figure 12**. Please make sure that this bit is a '0' when exiting the Local Loopback Mode.

MCR[2]: DTR# or RTS# for Auto Flow Control (OP1 in Local Loopback Mode)

DTR# or RTS# auto hardware flow control select. This bit is in effect only when auto RTS/DTR is enabled by EFR bit [6]. DTR# selection is associated with DSR# and RTS# is with CTS#.

- Logic 0 = Uses RTS# and CTS# pins for auto hardware flow control.
- Logic 1 = Uses DTR# and DSR# pins for auto hardware flow control.

In Local Loopback mode (MCR[4] = 1), this bit acts as the legacy OP1 output and controls the RI bit in the MSR register, as shown in **Figure 12**.

MCR[1]: RTS# Output

The RTS# pin may be used for automatic hardware flow control by enabled by EFR bit [6] and MCR bit [2]=0. If the modem interface is not used, this output may be used for general purpose.

- Logic 0 = Force RTS# output to a HIGH (default).
- Logic 1= Force RTS# output to LOW.

MCR[0]: DTR# Output

The DTR# pin may be used for automatic hardware flow control enabled by EFR bit [6] and MCR bit [2]=1. If the modem interface is not used, this output may be used for general purpose.

- Logic 0 = Force DTR# output to a HIGH (default).
- Logic 1 = Force DTR# output to a LOW.

4.9 Line Status Register (LSR) - Read Only

This register provides the status of data transfers between the UART and the host. If IER bit [2] is set to a logic 1, an LSR interrupt will be generated immediately when any character in the RX FIFO has an error (parity, framing, overrun, break).

LSR[7]: Receive FIFO Data Error Flag

- Logic 0 = No FIFO error (default).
- Logic 1 = An indicator for the sum of all error bits in the RX FIFO. At least one parity error, framing error or break indication is in the FIFO data. This bit clears when there are no more errors in the FIFO.

LSR[6]: Transmitter Empty Flag

This bit is the Transmitter Empty indicator. This bit is set to a logic 1 whenever both the transmit FIFO (or THR, in non-FIFO mode) and the transmit shift register (TSR) are both empty. It is set to logic 0 whenever either the TX FIFO or TSR contains a data character.

LSR[5]: Transmit FIFO Empty Flag

This bit is the Transmit FIFO Empty indicator. This bit indicates that the transmitter is ready to accept a new character for transmission. This bit is set to a logic HIGH when the last data byte is transferred from the transmit FIFO to the transmit shift register. The bit is reset to logic 0 as soon as a data byte is loaded into the transmit FIFO. In the non-FIFO mode this bit is set when the transmit holding register (THR) is empty; it is cleared when at a byte is written to the THR.

LSR[4]: Receive Break Flag

- Logic 0 = No break condition (default).
- Logic 1 = The receiver received a break signal (RX was LOW for one character frame time). In the FIFO mode, only one break character is loaded into the FIFO. The break indication remains until the RX input returns to the idle condition, “mark” or HIGH.

LSR[3]: Receive Data Framing Error Flag

- Logic 0 = No framing error (default).
- Logic 1 = Framing error. The receive character did not have a valid stop bit(s). This error is associated with the character available for reading in RHR.

LSR[2]: Receive Data Parity Error Flag

- Logic 0 = No parity error (default).
- Logic 1 = Parity error. The receive character in RHR (top of the FIFO) does not have correct parity information and is suspect. This error is associated with the character available for reading in RHR.

LSR[1]: Receiver Overrun Flag

- Logic 0 = No overrun error (default).
- Logic 1 = Overrun error. A data overrun error condition occurred in the receive shift register. This happens when additional data arrives while the FIFO is full. In this case the previous data in the receive shift register is overwritten. Note that under this condition the data byte in the receive shift register is not transferred into the FIFO, therefore the data in the FIFO is not corrupted by the error.

LSR[0]: Receive Data Ready Indicator

- Logic 0 = No data in receive holding register or FIFO (default).
- Logic 1 = Data has been received and is saved in the receive holding register or FIFO.

4.10 Modem Status Register (MSR) - Read Only

This register provides the current state of the modem interface signals, or other peripheral device that the UART is connected. Lower four bits of this register are used to indicate the changed information. These bits are set to a logic 1 whenever a signal from the modem changes state. These bits may be used as general purpose inputs/outputs when they are not used with modem signals.

MSR[7]: CD Input Status

Normally this bit is the complement of the CD# input. In the loopback mode this bit is equivalent to bit [3] in the MCR register. The CD# input may be used as a general purpose input when the modem interface is not used.

MSR[6]: RI Input Status

Normally this bit is the complement of the RI# input. In the loopback mode this bit is equivalent to bit [2] in the MCR register. The RI# input may be used as a general purpose input when the modem interface is not used.

MSR[5]: DSR Input Status

DSR# pin may function as automatic hardware flow control signal input if it is enabled and selected by Auto CTS/DSR bit (EFR bit [6]=1) and RTS/DTR flow control select bit (MCR bit [2]=1). Auto CTS/DSR flow control allows starting and stopping of local data transmissions based on the modem DSR# signal. A HIGH on the DSR# pin will stop UART transmitter as soon as the current character has finished transmission, and a LOW will resume data transmission. Normally MSR bit [5] is the complement of the DSR# input. However in the loopback mode, this bit is equivalent to the DTR# bit in the MCR register. The DSR# input may be used as a general purpose input when the modem interface is not used.

MSR[4]: CTS Input Status

CTS# pin may function as automatic hardware flow control signal input if it is enabled and selected by Auto CTS/DSR bit (EFR bit [6]=1) and RTS/DTR flow control select bit (MCR bit [2]=0). Auto CTS/DSR flow control allows starting and stopping of local data transmissions based on the modem CTS# signal. A HIGH on the CTS# pin will stop UART transmitter as soon as the current character has finished transmission, and a LOW will resume data transmission. Normally MSR bit [4] is the complement of the CTS# input. However in the loopback mode, this bit is equivalent to the RTS# bit in the MCR register. The CTS# input may be used as a general purpose input when the modem interface is not used.

MSR[3]: Delta CD# Input Flag

- Logic 0 = No change on CD# input (default).
- Logic 1 = Indicates that the CD# input has changed state since the last time it was monitored. A modem status interrupt will be generated if MSR interrupt is enabled (IER bit [3]).

MSR[2]: Delta RI# Input Flag

- Logic 0 = No change on RI# input (default).
- Logic 1 = The RI# input has changed from a LOW to a HIGH, ending of the ringing signal. A modem status interrupt will be generated if MSR interrupt is enabled (IER bit [3]).

MSR[1]: Delta DSR# Input Flag

- Logic 0 = No change on DSR# input (default).
- Logic 1 = The DSR# input has changed state since the last time it was monitored. A modem status interrupt will be generated if MSR interrupt is enabled (IER bit [3]).

MSR[0]: Delta CTS# Input Flag

- Logic 0 = No change on CTS# input (default).
- Logic 1 = The CTS# input has changed state since the last time it was monitored. A modem status interrupt will be generated if MSR interrupt is enabled (IER bit [3]).

4.11 Modem Status Register (MSR) - Write Only

The upper four bits [7:4] of this register set the delay in number of bits time for the auto RS-485 turn around from transmit to receive.

MSR [7:4]: Auto RS485 Turn-Around Delay (requires EFR bit [4]=1)

When Auto RS485 feature is enabled (FCTR bit [5]=1) and RTS#/DTR# output is connected to the enable input of a RS-485 transceiver. These 4 bits select from 0 to 15 bit-time delay after the end of the last stop-bit of the last transmitted character. This delay controls when to change the state of RTS#/DTR# output. This delay is very useful in long-cable networks. Table 17 shows the selection. The bits are enabled by EFR bit-4.

TABLE 17: AUTO RS485 HALF-DUPLEX DIRECTION CONTROL DELAY FROM TRANSMIT-TO-RECEIVE

MSR[7]	MSR[6]	MSR[5]	MSR[4]	DELAY IN DATA BIT(S) TIME
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
9	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
1	1	0	0	12
1	1	0	1	13
1	1	1	0	14
1	1	1	1	15

MSR [3]: Transmitter Disable

This bit can be used to disable the transmitter by halting the Transmit Shift Register (TSR). When this bit is set to a logic 1, the bytes already in the FIFO will not be sent out. Also, any more data loaded into the FIFO will stay in the FIFO and will not be sent out. When this bit is set to a logic 0, the bytes currently in the TX FIFO will be sent out. Please note that setting this bit to a logic 1 stops any character from going out. Also, this bit must be a logic 0 for the Send Char Immediate function (see MCR[3]).

- Logic 0 = Enable Transmitter (default).
- Logic 1 = Disable Transmitter.

MSR [2]: Receiver Disable

This bit can be used to disable the receiver by halting the Receive Shift Register (RSR). When this bit is set to a logic 1, the receiver will operate in one of the following ways:

- If a character is being received at the time of setting this bit, that character will be correctly received. No more characters will be received.
- If the receiver is idle at the time of setting this bit, no more characters will be received.

The receiver can be enabled and will start receiving characters by resetting this bit to a logic 0. The receiver will operate in one of the following ways:

- If the receiver is idle (RX pin is HIGH) at the time of setting this bit, the next character will be received normally. It is recommended that the receiver be idle when resetting this bit to a logic 0.
- If the receiver is not idle (RX pin is toggling) at the time of setting this bit, the RX FIFO will be filled with unknown data.

Any data that is in the RX FIFO can be read out at any time whether the receiver is disabled or not.

- Logic 0 = Enable Receiver (default).
- Logic 1 = Disable Receiver.

MSR [1]: Transmitter Disable Modes

This bit is only applicable when MSR[3] = 1.

- Logic 0 = No xon/xoff software flow control characters will be transmitted when the transmitter is disabled. If there is a pending xon/xoff character to be sent while the transmitter is disabled, it will be transmitted. No additional xon/xoff characters will be sent.
- Logic 1 = Xon/xoff software flow control characters will be transmitted even though the transmitter is disabled.

MSR[0]: Receiver Disable Modes

This is only applicable when MSR[2] = 1.

- Logic 0 = All RX data and xon/xoff flow control characters are ignored.
- Logic 1 = All RX data is ignored. Xon/xoff flow control characters are detected and acted upon.

4.12 SCRATCH PAD REGISTER (SPR) - Read/Write

This is a 8-bit general purpose register for the user to store temporary data. The content of this register is preserved during sleep mode but becomes 0xFF (default) after a reset or a power off-on cycle.

4.13 FEATURE CONTROL REGISTER (FCTR) - Read/Write

This register controls the UART enhanced functions that are not available on ST16C554 or ST16C654.

FCTR[7:6]: TX and RX FIFO Trigger Table Select

These 2 bits select the transmit and receive FIFO trigger level table A, B, C or D. When table A, B, or C is selected the auto RTS flow control trigger level is set to "next FIFO trigger level" for compatibility to ST16C550 and ST16C650 series. RTS/DTR# triggers on the next level of the RX FIFO trigger level, in another word, one FIFO level above and one FIFO level below. See in **Table 15** for complete selection with FCR bit [5:4] and FCTR bits [7:6], i.e. if Table C is used on the receiver with RX FIFO trigger level set to 56 bytes, RTS/DTR# output will de-assert at 60 and re-assert at 16.

FCTR[5]: Auto RS485 Enable

Auto RS485 half duplex control enable/disable. RTS# or DTR# can be selected as the control output via MCR bit-2. Note that this feature has precedence over the Auto RTS/DTR flow control feature (EFR bit-6). Therefore, the Auto RTS/DTR flow control feature will not have any effect when the Auto RS485 Half-Duplex Direction Control feature is enabled.

- Logic 0 = Standard ST16C550 mode. Transmitter generates an interrupt when transmit holding register (THR) becomes empty. Transmit Shift Register (TSR) may still be shifting data bit out.
- Logic 1 = Enable Auto RS485 half duplex direction control. RTS#/DTR# output changes from HIGH to LOW when finished sending the last stop bit of the last character out of the TSR register. It changes from LOW to HIGH when a data byte is loaded into the THR or transmit FIFO. The change to HIGH occurs prior sending the start-bit. It also changes the transmitter interrupt from transmit holding to transmit shift register (TSR) empty. If software flow control is enabled, the RTS#/DTR# output will not change if the TX FIFO is empty and the RX FIFO level generates an XON or XOFF character to be transmitted.

FCTR[4]: Infrared RX Input Logic Select

- Logic 0 = Select RX input as active HIGH encoded IrDA data, normal, (default).
- Logic 1 = Select RX input as active LOW encoded IrDA data, inverted.

FCTR [3:0] - Auto RTS/DTR Flow Control Hysteresis Select

These bits select the auto RTS/DTR flow control hysteresis and only valid when TX and RX Trigger Table-D is selected (FCTR bit [7:6] are set to logic 1). The RTS/DTR hysteresis is referenced to the RX FIFO trigger level. After reset, these bits are set to logic 0 selecting the next FIFO trigger level for hardware flow control. Table 18 below shows the 16 selectable hysteresis levels.

TABLE 18: 16 SELECTABLE HYSTERESIS LEVELS WHEN TRIGGER TABLE-D IS SELECTED

FCTR BIT [3]	FCTR BIT [2]	FCTR BIT [1]	FCTR BIT [0]	RTS/DTR HYSTERESIS (CHARACTERS)
0	0	0	0	0
0	0	0	1	+/- 4
0	0	1	0	+/- 6
0	0	1	1	+/- 8
0	1	0	0	+/- 8
0	1	0	1	+/- 16
0	1	1	0	+/- 24
0	1	1	1	+/- 32
1	1	0	0	+/- 12
1	1	0	1	+/- 20
1	1	1	0	+/- 28
1	1	1	1	+/- 36
1	0	0	0	+/- 40
1	0	0	1	+/- 44
1	0	1	0	+/- 48
1	0	1	1	+/- 52

4.14 Enhanced Feature Register (EFR) - Read/Write

Enhanced features are enabled or disabled using this register. Bits [3:0] provide single or dual consecutive character software flow control selection (see Table 19). When the Xon1 and Xon2 and Xoff1 and Xoff2 modes are selected, the double 8-bit words are concatenated into two sequential characters. Caution: note that whenever changing the TX or RX flow control bits, always reset all bits back to logic 0 (disable) before programming a new setting.

EFR[7]: Auto CTS Flow Control Enable

Automatic CTS or DSR Flow Control.

- Logic 0 = Automatic CTS/DSR flow control is disabled (default).
- Logic 1 = Enable Automatic CTS/DSR flow control. Transmission stops when CTS/DSR# pin de-asserts (HIGH). Transmission resumes when CTS/DSR# pin is asserted (LOW). The selection for CTS# or DSR# is through MCR bit [2].

EFR[6]: Auto RTS or DTR Flow Control Enable

RTS#/DTR# output may be used for hardware flow control by setting EFR bit [6] to logic 1. When Auto RTS/DTR is selected, an interrupt will be generated when the receive FIFO is filled to the programmed trigger level and RTS/DTR# will de-assert (HIGH) at the next upper trigger or selected hysteresis level. RTS/DTR# will re-assert (LOW) when FIFO data falls below the next lower trigger or selected hysteresis level (see FCTR bits 4-7). The RTS# or DTR# output must be asserted (LOW) before the auto RTS/DTR can take effect. The selection for RTS# or DTR# is through MCR bit [2]. RTS/DTR# pin will function as a general purpose output when hardware flow control is disabled.

- Logic 0 = Automatic RTS/DTR flow control is disabled (default).
- Logic 1 = Enable Automatic RTS/DTR flow control.

EFR[5]: Special Character Detect Enable

- Logic 0 = Special Character Detect Disabled (default).
- Logic 1 = Special Character Detect Enabled. The UART compares each incoming receive character with data in Xoff-2 register. If a match exists, the received data will be transferred to FIFO and ISR bit [4] will be set to indicate detection of the special character. bit [0] corresponds with the LSB bit for the receive character. If flow control is set for comparing Xon1, Xoff1 (EFR [1:0]=10) then flow control and special character work normally. However, if flow control is set for comparing Xon2, Xoff2 (EFR[1:0]=01) then flow control works normally, but Xoff2 will not go to the FIFO, and will generate an Xoff interrupt and a special character interrupt.

EFR[4]: Enhanced Function Bits Enable

Enhanced function control bit. This bit enables the enhanced functions in IER bits [7:5], ISR bits [5:4], FCR bits [5:4], MCR bits [7:5] and MSR [7:0] bits to be modified. After modifying any enhanced bits, EFR bit [4] can be set to a logic 0 to latch the new values. This feature prevents legacy software from altering or overwriting the enhanced functions once set. Normally, it is recommended to leave it enabled.

- Logic 0 = Disable write access to the enhanced function bits: IER bits [7:5], ISR bits [5:4], FCR bits [5:4], MCR bits [7:5] and MSR [7:0] bits. After a reset, all these bits are set to a logic 0 to be compatible with ST16C550 mode (default).
- Logic 1 = Enables write access to the enhanced function bits: IER bits [7:5], ISR bits [5:4], FCR bits [5:4], MCR bits [7:5] and MSR [7:0] bits.

EFR[3:0]: Software Flow Control Select

Combinations of software flow control can be selected by programming these bits, as shown in Table 19.

TABLE 19: SOFTWARE FLOW CONTROL FUNCTIONS

EFR BIT [3]	EFR BIT [2]	EFR BIT [1]	EFR BIT [0]	TRANSMIT AND RECEIVE SOFTWARE FLOW CONTROL
0	0	0	0	No TX and RX flow control (default and reset)
0	0	X	X	No transmit flow control
1	0	X	X	Transmit Xon1, Xoff1
0	1	X	X	Transmit Xon2, Xoff2
1	1	X	X	Transmit Xon1 and Xon2, Xoff1 and Xoff2
X	X	0	0	No receive flow control
X	X	1	0	Receiver compares Xon1, Xoff1
X	X	0	1	Receiver compares Xon2, Xoff2
1	0	1	1	Transmit Xon1, Xoff1 Receiver compares Xon1 or Xon2, Xoff1 or Xoff2
0	1	1	1	Transmit Xon2, Xoff2 Receiver compares Xon1 or Xon2, Xoff1 or Xoff2
1	1	1	1	Transmit Xon1 and Xon2, Xoff1 and Xoff2 Receiver compares Xon1 and Xon2, Xoff1 and Xoff2
0	0	1	1	No transmit flow control Receiver compares Xon1 and Xon2, Xoff1 and Xoff2

Software flow control can not be used when the Auto RS-485 Half-Duplex Direction Control feature is enabled (FCTR[5]=1). With this feature enabled, the RTS#/DTR# output controls the direction of the half-duplex RS-485 transceiver. The RTS#/DTR# output changes the direction of the half-duplex transceiver to the transmit mode when data is being transmitted from the UART on the TX output. However, the RTS#/DTR# output will remain in the receive direction if the TX FIFO is empty and the RX FIFO triggers an XON or XOFF character to be transmitted.

4.15 TXCNT[7:0]: Transmit FIFO Level Counter - Read Only

Transmit FIFO level byte count from 0x00 (0 bytes) to 0xFF (255 or 256 bytes). This 8-bit register gives an indication of the number of characters in the transmit FIFO. The FIFO level Byte count register is read only. The user can take advantage of the FIFO level byte counter for faster data loading to the transmit FIFO, which reduces CPU bandwidth requirements.

4.16 TXTRG [7:0]: Transmit FIFO Trigger Level - Write Only

An 8-bit value written to this register sets the TX FIFO trigger level from 0x00 (zero) to 0xFF (255). The TX FIFO trigger level generates an interrupt whenever the data level in the transmit FIFO falls below this preset trigger level.

4.17 RXCNT[7:0]: Receive FIFO Level Counter - Read Only

Receive FIFO level byte count from 0x00 (0 bytes) to 0xFF (255 or 256 bytes). It gives an indication of the number of characters in the receive FIFO. The FIFO level byte count register is read only. The user can take advantage of the FIFO level byte counter for faster data unloading from the receiver FIFO, which reduces CPU bandwidth requirements.

4.18 RXTRG[7:0]: Receive FIFO Trigger Level - Write Only

An 8-bit value written to this register, sets the RX FIFO trigger level from 0x00 (zero) to 0xFF (255). The RX FIFO trigger level generates an interrupt whenever the receive FIFO level rises to this preset trigger level.

4.19 XOFF1, XOFF2, XON1 AND XON2 REGISTERS - Write Only

These registers are used to program the Xoff1, Xoff2, Xon1 and Xon2 control characters respectively.

4.20 XCHAR REGISTER - Read Only

This register gives the status of the last sent control character (Xon or Xoff) and the last received control character (Xon or Xoff). This register will be reset to 0x00 if, at anytime, the Software Flow Control is disabled.

XCHAR [7:4]: Reserved

XCHAR [3]: Transmit Xon Indicator

If the last transmitted control character was a Xon character or characters (Xon1, Xon2), this bit will be set to a logic 1. This bit will clear after the read.

XCHAR [2]: Transmit Xoff Indicator

If the last transmitted control character was a Xoff character or characters (Xoff1, Xoff2), this bit will be set to a logic 1. This bit will clear after the read.

XCHAR [1]: Xon Detect Indicator

If the last received control character was a Xon character, Xon characters (Xon1, Xon2) or an Xon-Any character, this bit will be set to a logic 1. This bit will clear after the read.

XCHAR [0]: Xoff Detect Indicator

If the last received control character was a Xoff character or characters (Xoff1, Xoff2), this bit will be set to a logic 1. This bit will clear after the read.

TABLE 20: UART RESET CONDITIONS

REGISTERS	RESET STATE
DLL	Bits [7:0] = 0x01
DLM	Bits [7:0] = 0x00
DLD	Bits [7:0] = 0x00
RHR	Bits [7:0] = 0xXX
THR	Bits [7:0] = 0xXX
IER	Bits [7:0] = 0x00
FCR	Bits [7:0] = 0x00
ISR	Bits [7:0] = 0x01
LCR	Bits [7:0] = 0x00
MCR	Bits [7:0] = 0x00
LSR	Bits [7:0] = 0x60
MSR	Bits [3:0] = logic 0 Bits [7:4] = logic levels of the inputs
SPR	Bits [7:0] = 0xFF
FCTR	Bits [7:0] = 0x00
EFR	Bits [7:0] = 0x00
TXCNT	Bits [7:0] = 0x00
TXTRG	Bits [7:0] = 0x00
RXCNT	Bits [7:0] = 0x00
RXTRG	Bits [7:0] = 0x00
XCHAR	Bits [7:0] = 0x00
XON1	Bits [7:0] = 0x00
XON2	Bits [7:0] = 0x00
XOFF1	Bits [7:0] = 0x00
XOFF2	Bits [7:0] = 0x00

I/O SIGNALS	RESET STATE
TX[1:0]	HIGH
IRTX[1:0]	LOW
RTS#[1:0]	HIGH
DTR#[1:0]	HIGH
EECK	LOW
EECS	LOW
EEDI	LOW

ABSOLUTE MAXIMUM RATINGS

Power Supply Range	3.6 Volts
Voltage at Any Pin	-0.5 to VCC+0.5V
Operating Temperature	-40° to +85° C
Storage Temperature	-65° to +150° C
Package Dissipation	500 mW
Thermal Resistance (113-FPBGA)	theta-ja = 30.3° C/W, theta-jc = 24.4° C/W

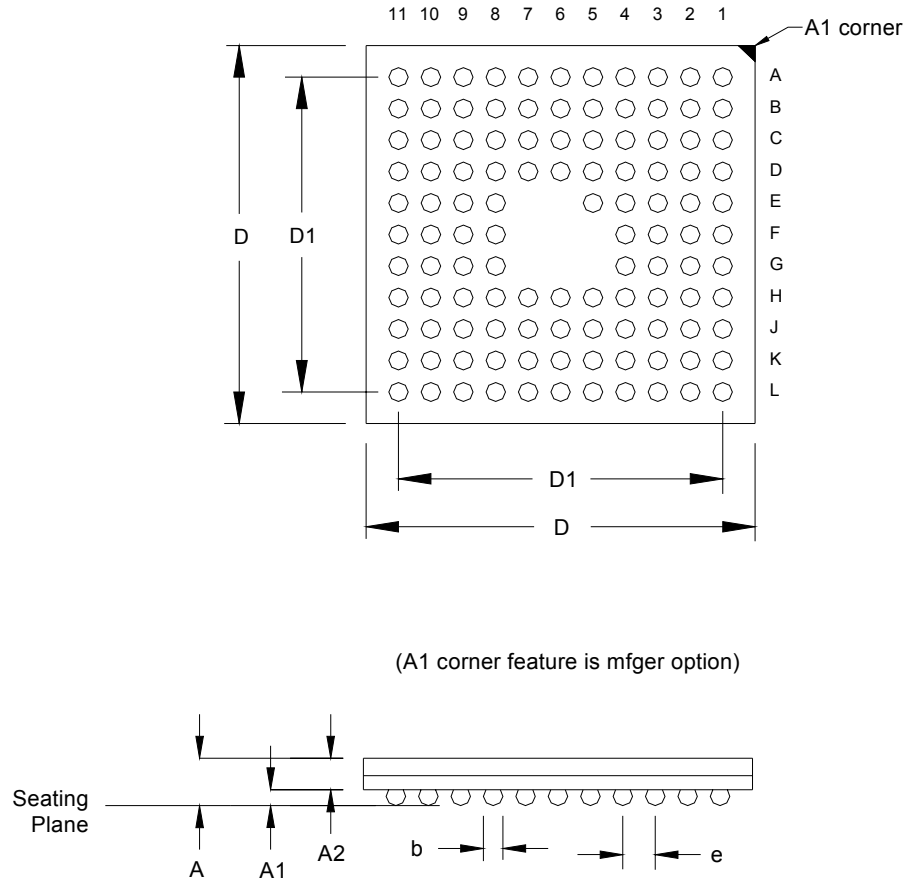
ELECTRICAL CHARACTERISTICS

DC ELECTRICAL CHARACTERISTICS

TA=-40° TO +85°C (INDUSTRIAL GRADE) SUPPLY VOLTAGE, VCC = 3.3V

SYMBOL	PARAMETER	MIN	MAX	UNITS	CONDITION	NOTES
V_{IL}	Input Low Voltage	-0.3	0.6	V		
V_{IH}	Input High Voltage	2.4	VCC33	V		
V_{OL}	Output Low Voltage		0.4	V	I_{OL} = 6 mA	
V_{OH}	Output High Voltage	2.4		V	I_{OH} = -4mA	
I_{CC}	Power Supply Current		70	mA		Total for all VCC33 power supplies

PACKAGE DIMENSIONS (113-FPBGA)



NOTE: *Note: The control dimension is the millimeter column*

SYMBOL	INCHES		MILLIMETERS	
	MIN	MAX	MIN	MAX
A	0.048	0.060	1.23	1.53
A1	0.011	0.015	0.28	0.38
A2	0.037	0.045	0.95	1.15
D	0.350	0.358	8.90	9.10
D1	0.315 BSC		8.00 BSC	
b	0.018	0.022	0.45	0.55
e	0.031 BSC		0.80 BSC	

REVISION HISTORY

DATE	REVISION	DESCRIPTION
March 2009	P1.0.0	Preliminary Datasheet.
July 2009	P1.0.1	Added preliminary DC Electrical Characteristics. Clarified VCC33 and VCC12 pin descriptions.
December 2009	Rev 1.0.0	Final datasheet. Updated DC Electrical specs.
September 2010	Rev 1.0.1	Corrected the polarity of "TRST#" signal.
April 2011	Rev 1.0.2	Removed the "MSI-X Capable Capability ID" in the Table 1.

Datasheet April 2011.

Send your UART technical inquiry with technical details to hotline: uarttechsupport@exar.com.