# PIC16F630/676 Data Sheet

## 14-Pin, Flash-Based 8-Bit CMOS Microcontrollers

**Note the following details of the code protection feature on Microchip devices:**

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights.

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
ISO/TS 16949:2002

**Trademarks**

The Microchip name and logo, the Microchip logo, dsPIC, KEELOQ, KEELOQ logo, MPLAB, PIC, PICmicro, PICSTART, PIC$^{32}$ logo, rfPIC and UNI/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

FilterLab, Hampshire, HI-TECH C, Linear Active Thermistor, MXDEV, MXLAB, SEEVAL and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, Application Maestro, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, dsSPEAK, ECAN, ECONOMONITOR, FanSense, HI-TIDE, In-Circuit Serial Programming, ICSP, Mindi, MiWi, MPASM, MPLAB Certified logo, MPLIB, MPLINK, mTouch, Octopus, Omniscient Code Generation, PICC, PICC-18, PICDEM, PICDEM.net, PICkit, PICtail, REAL ICE, rfLAB, Select Mode, Total Endurance, TSHARC, UniWinDriver, WiperLock and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

All other trademarks mentioned herein are property of their respective companies.

© 2010, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

Printed on recycled paper.

ISBN: 978-1-60932-173-4

*Microchip received ISO/TS-16949:2002 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.*

MICROCHIP

# PIC16F630/676

## 14-Pin, Flash-Based 8-Bit CMOS Microcontroller

### High-Performance RISC CPU:

- Only 35 Instructions to Learn
  - All single-cycle instructions except branches
- Operating Speed:
  - DC – 20 MHz oscillator/clock input
  - DC – 200 ns instruction cycle
- Interrupt Capability
- 8-level Deep Hardware Stack
- Direct, Indirect, and Relative Addressing modes

### Special Microcontroller Features:

- Internal and External Oscillator Options
  - Precision Internal 4 MHz oscillator factory calibrated to ±1%
  - External Oscillator support for crystals and resonators
  - 5 μs wake-up from Sleep, 3.0V, typical
- Power-Saving Sleep mode
- Wide Operating Voltage Range – 2.0V to 5.5V
- Industrial and Extended Temperature Range
- Low-Power Power-on Reset (POR)
- Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Brown-out Detect (BOD)
- Watchdog Timer (WDT) with Independent Oscillator for Reliable Operation
- Multiplexed $\overline{\text{MCLR}}$/Input-pin
- Interrupt-on-Pin Change
- Individual Programmable Weak Pull-ups
- Programmable Code Protection
- High Endurance Flash/EEPROM Cell
  - 100,000 write Flash endurance
  - 1,000,000 write EEPROM endurance
  - Flash/data EEPROM retention: > 40 years

### Low-Power Features:

- Standby Current:
  - 1 nA @ 2.0V, typical
- Operating Current:
  - 8.5 μA @ 32 kHz, 2.0V, typical
  - 100 μA @ 1 MHz, 2.0V, typical
- Watchdog Timer Current
  - 300 nA @ 2.0V, typical
- Timer1 Oscillator Current:
  - 4 μA @ 32 kHz, 2.0V, typical

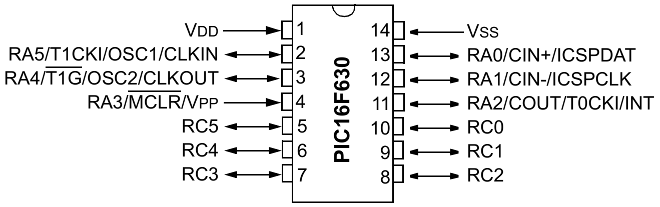
### Peripheral Features:

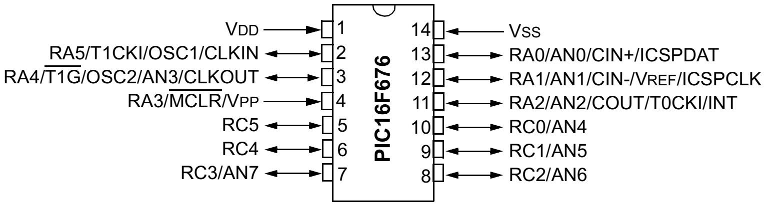
- 12 I/O Pins with Individual Direction Control
- High Current Sink/Source for Direct LED Drive
- Analog Comparator module with:
  - One analog comparator
  - Programmable on-chip comparator voltage reference (CVREF) module
  - Programmable input multiplexing from device inputs
  - Comparator output is externally accessible
- Analog-to-Digital Converter module (PIC16F676):
  - 10-bit resolution
  - Programmable 8-channel input
  - Voltage reference input
- Timer0: 8-bit Timer/Counter with 8-bit Programmable Prescaler
- Enhanced Timer1:
  - 16-bit timer/counter with prescaler
  - External Gate Input mode
  - Option to use OSC1 and OSC2 in LP mode as Timer1 oscillator, if INTOSC mode selected
- In-Circuit Serial Programming™ (ICSP™) via two pins

| Device | Program Memory | Data Memory | | I/O | 10-bit A/D (ch) | Comparators | Timers 8/16-bit |
|---|---|---|---|---|---|---|---|
| | Flash (words) | SRAM (bytes) | EEPROM (bytes) | | | | |
| PIC16F630 | 1024 | 64 | 128 | 12 | – | 1 | 1/1 |
| PIC16F676 | 1024 | 64 | 128 | 12 | 8 | 1 | 1/1 |

## Pin Diagrams

**14-pin PDIP, SOIC, TSSOP**





## Table of Contents

## TO OUR VALUED CUSTOMERS

It is our intention to provide our valued customers with the best documentation possible to ensure successful use of your Microchip products. To this end, we will continue to improve our publications to better suit your needs. Our publications will be refined and enhanced as new volumes and updates are introduced.

If you have any questions or comments regarding this publication, please contact the Marketing Communications Department via E-mail at **docerrors@microchip.com** or fax the **Reader Response Form** in the back of this data sheet to (480) 792-4150. We welcome your feedback.

### Most Current Data Sheet

To obtain the most up-to-date version of this data sheet, please register at our Worldwide Web site at:

http://www.microchip.com

You can determine the version of a data sheet by examining its literature number found on the bottom outside corner of any page. The last character of the literature number is the version number, (e.g., DS30000A is version A of document DS30000).

### Errata

An errata sheet, describing minor operational differences from the data sheet and recommended workarounds, may exist for current devices. As device/documentation issues become known to us, we will publish an errata sheet. The errata will specify the revision of silicon and revision of document to which it applies.

To determine if an errata sheet exists for a particular device, please check with one of the following:

- Microchip's Worldwide Web site; http://www.microchip.com
- Your local Microchip sales office (see last page)

When contacting a sales office, please specify which device, revision of silicon and data sheet (include literature number) you are using.

### Customer Notification System

Register on our web site at **www.microchip.com** to receive the most current information on all of our products.

**NOTES:**

## 1.0 DEVICE OVERVIEW

This document contains device specific information for the PIC16F630/676. Additional information may be found in the PIC® Mid-Range Reference Manual (DS33023), which may be obtained from your local Microchip Sales Representative or downloaded from the Microchip web site. The Reference Manual should be considered a complementary document to this Data Sheet and is highly recommended reading for a better understanding of the device architecture and operation of the peripheral modules.

The PIC16F630 and PIC16F676 devices are covered by this Data Sheet. They are identical, except the PIC16F676 has a 10-bit A/D converter. They come in 14-pin PDIP, SOIC and TSSOP packages. Figure 1-1 shows a block diagram of the PIC16F630/676 devices. Table 1-1 shows the pinout description.

**FIGURE 1-1: PIC16F630/676 BLOCK DIAGRAM**

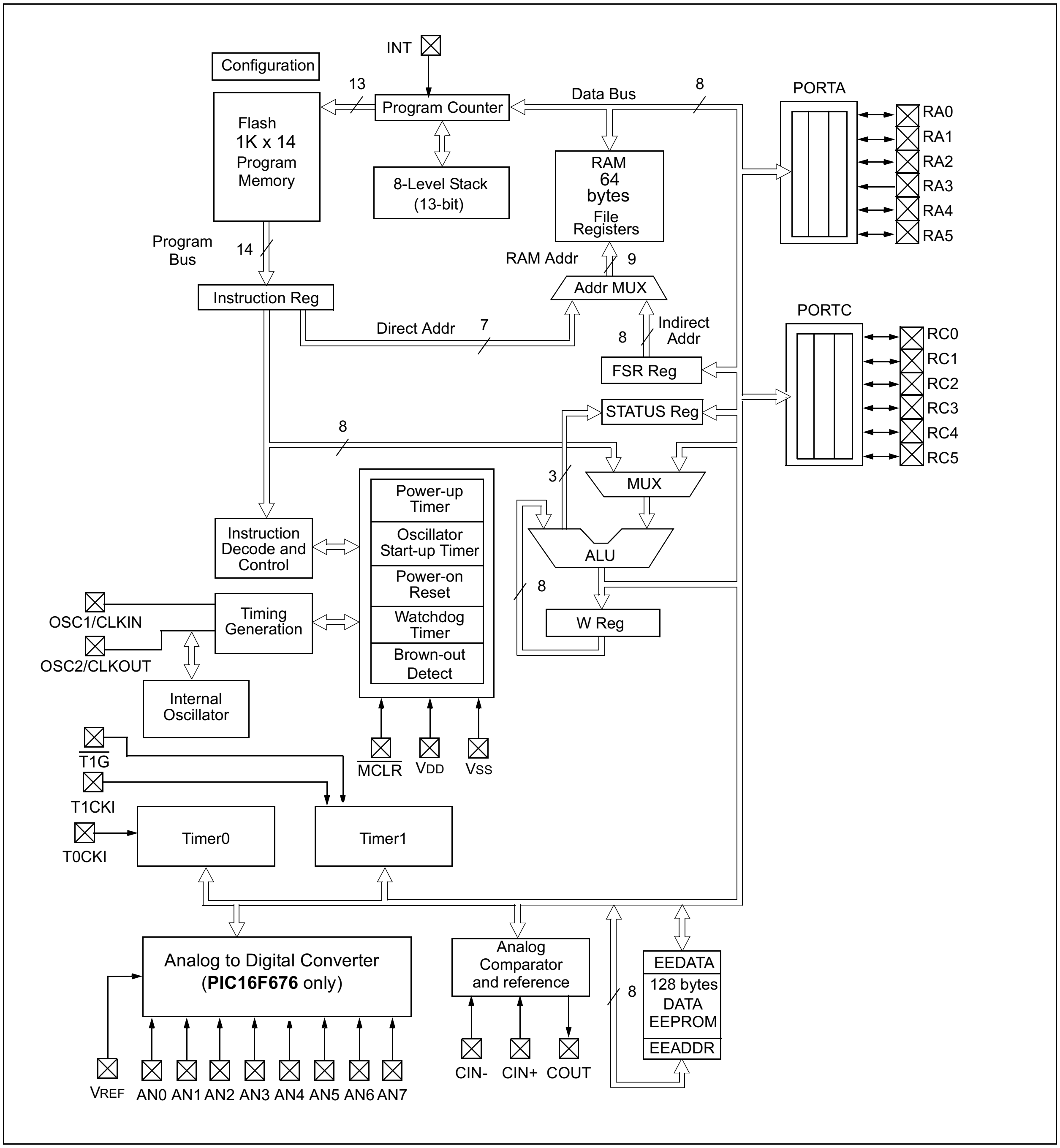


**TABLE 1-1: PIC16F630/676 PINOUT DESCRIPTION**

| Name | Function | Input Type | Output Type | Description |
|---|---|---|---|---|
| RA0/AN0/CIN+/ICSPDAT | RA0 | TTL | CMOS | Bidirectional I/O w/ programmable pull-up and interrupt-on-change. |
| | AN0 | AN | — | A/D Channel 0 input. |
| | CIN+ | AN | | Comparator input. |
| | ICSPDAT | TTL | CMOS | Serial Programming Data I/O. |
| RA1/AN1/CIN-/VREF/ICSPCLK | RA1 | TTL | CMOS | Bidirectional I/O w/ programmable pull-up and interrupt-on-change. |
| | AN1 | AN | — | A/D Channel 1 input. |
| | CIN- | AN | — | Comparator input. |
| | VREF | AN | — | External Voltage reference. |
| | ICSPCLK | ST | — | Serial Programming Clock. |
| RA2/AN2/COUT/T0CKI/INT | RA2 | ST | CMOS | Bidirectional I/O w/ programmable pull-up and interrupt-on-change. |
| | AN2 | AN | — | A/D Channel 2 input. |
| | COUT | — | CMOS | Comparator output. |
| | T0CKI | ST | — | Timer0 clock input. |
| | INT | ST | — | External Interrupt. |
| RA3/$\overline{\text{MCLR}}$/VPP | RA3 | TTL | — | Input port with interrupt-on-change. |
| | $\overline{\text{MCLR}}$ | ST | — | Master Clear. |
| | VPP | HV | — | Programming voltage. |
| RA4/$\overline{\text{T1G}}$/AN3/OSC2/CLKOUT | RA4 | TTL | CMOS | Bidirectional I/O w/ programmable pull-up and interrupt-on-change. |
| | $\overline{\text{T1G}}$ | ST | — | Timer1 gate. |
| | AN3 | AN3 | — | A/D Channel 3 input. |
| | OSC2 | — | XTAL | Crystal/Resonator. |
| | CLKOUT | — | CMOS | FOSC/4 output. |
| RA5/T1CKI/OSC1/CLKIN | RA5 | TTL | CMOS | Bidirectional I/O w/ programmable pull-up and interrupt-on-change. |
| | T1CKI | ST | — | Timer1 clock. |
| | OSC1 | XTAL | — | Crystal/Resonator. |
| | CLKIN | ST | — | External clock input/RC oscillator connection. |
| RC0/AN4 | RC0 | TTL | CMOS | Bidirectional I/O. |
| | AN4 | AN4 | — | A/D Channel 4 input. |
| RC1/AN5 | RC1 | TTL | CMOS | Bidirectional I/O. |
| | AN5 | AN5 | — | A/D Channel 5 input. |
| RC2/AN6 | RC2 | TTL | CMOS | Bidirectional I/O. |
| | AN6 | AN6 | — | A/D Channel 6 input. |
| RC3/AN7 | RC3 | TTL | CMOS | Bidirectional I/O. |
| | AN7 | AN7 | — | A/D Channel 7 input. |
| RC4 | RC4 | TTL | CMOS | Bidirectional I/O. |
| RC5 | RC5 | TTL | CMOS | Bidirectional I/O. |
| VSS | VSS | Power | — | Ground reference. |
| VDD | VDD | Power | — | Positive supply. |

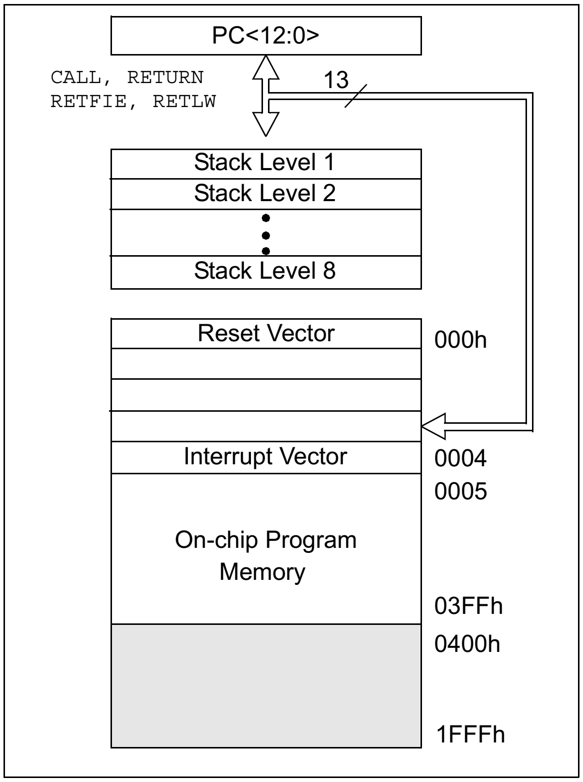
**Legend:** Shade = PIC16F676 only
TTL = TTL input buffer
ST = Schmitt Trigger input buffer

# 2.0 MEMORY ORGANIZATION

## 2.1 Program Memory Organization

The PIC16F630/676 devices have a 13-bit program counter capable of addressing an 8K x 14 program memory space. Only the first 1K x 14 (0000h-03FFh) for the PIC16F630/676 devices is physically implemented. Accessing a location above these boundaries will cause a wrap around within the first 1K x 14 space. The Reset vector is at 0000h and the interrupt vector is at 0004h (see Figure 2-1).

**FIGURE 2-1: PROGRAM MEMORY MAP AND STACK FOR THE PIC16F630/676**



## 2.2 Data Memory Organization

The data memory (see Figure 2-2) is partitioned into two banks, which contain the General Purpose Registers and the Special Function Registers. The Special Function Registers are located in the first 32 locations of each bank. Register locations 20h-5Fh are General Purpose Registers, implemented as static RAM and are mapped across both banks. All other RAM is unimplemented and returns '`0`' when read. RP0 (STATUS<5>) is the bank select bit.

- RP0 = `0` Bank 0 is selected
- RP0 = `1` Bank 1 is selected

**Note:** The IRP and RP1 bits STATUS<7:6> are reserved and should always be maintained as '`0`'s.

### 2.2.1 GENERAL PURPOSE REGISTER FILE

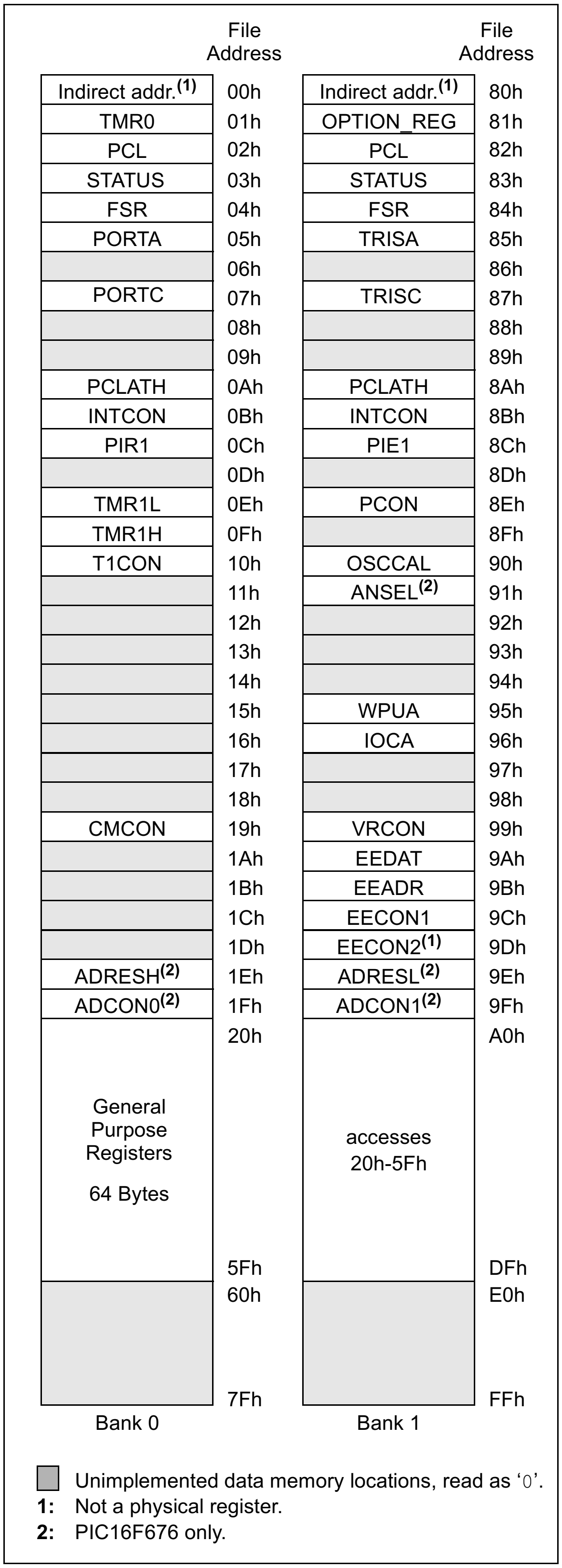
The register file is organized as 64 x 8 in the PIC16F630/676 devices. Each register is accessed, either directly or indirectly, through the File Select Register FSR (see **Section 2.4 "Indirect Addressing, INDF and FSR Registers"**).

### 2.2.2 SPECIAL FUNCTION REGISTERS

The Special Function Registers are registers used by the CPU and peripheral functions for controlling the desired operation of the device (see Table 2-1). These registers are static RAM.

The special registers can be classified into two sets: core and peripheral. The Special Function Registers associated with the “core” are described in this section. Those related to the operation of the peripheral features are described in the section of that peripheral feature.

**FIGURE 2-2: DATA MEMORY MAP OF THE PIC16F630/676**



| Bank 0 | File Address | Bank 1 | File Address |
|---|---|---|---|
| Indirect addr.(1) | 00h | Indirect addr.(1) | 80h |
| TMR0 | 01h | OPTION_REG | 81h |
| PCL | 02h | PCL | 82h |
| STATUS | 03h | STATUS | 83h |
| FSR | 04h | FSR | 84h |
| PORTA | 05h | TRISA | 85h |
| ▨ | 06h | ▨ | 86h |
| PORTC | 07h | TRISC | 87h |
| ▨ | 08h | ▨ | 88h |
| ▨ | 09h | ▨ | 89h |
| PCLATH | 0Ah | PCLATH | 8Ah |
| INTCON | 0Bh | INTCON | 8Bh |
| PIR1 | 0Ch | PIE1 | 8Ch |
| ▨ | 0Dh | ▨ | 8Dh |
| TMR1L | 0Eh | PCON | 8Eh |
| TMR1H | 0Fh | ▨ | 8Fh |
| T1CON | 10h | OSCCAL | 90h |
| ▨ | 11h | ANSEL(2) | 91h |
| ▨ | 12h | ▨ | 92h |
| ▨ | 13h | ▨ | 93h |
| ▨ | 14h | ▨ | 94h |
| ▨ | 15h | WPUA | 95h |
| ▨ | 16h | IOCA | 96h |
| ▨ | 17h | ▨ | 97h |
| ▨ | 18h | ▨ | 98h |
| CMCON | 19h | VRCON | 99h |
| ▨ | 1Ah | EEDAT | 9Ah |
| ▨ | 1Bh | EEADR | 9Bh |
| ▨ | 1Ch | EECON1 | 9Ch |
| ▨ | 1Dh | EECON2(1) | 9Dh |
| ADRESH(2) | 1Eh | ADRESL(2) | 9Eh |
| ADCON0(2) | 1Fh | ADCON1(2) | 9Fh |
| General Purpose Registers<br>64 Bytes | 20h<br>5Fh | accesses 20h-5Fh | A0h<br>DFh |
| ▨ | 60h<br>7Fh | ▨ | E0h<br>FFh |

▨ Unimplemented data memory locations, read as ‘0’.

**1:** Not a physical register.

**2:** PIC16F676 only.

### TABLE 2-1: PIC16F630/676 SPECIAL REGISTERS SUMMARY BANK 0

| Addr | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR, BOD | Page |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Bank 0 | | | | | | | | | | | |
| 00h | INDF | Addressing this location uses contents of FSR to address data memory (not a physical register) | | | | | | | | `xxxx xxxx` | 20,63 |
| 01h | TMR0 | Timer0 Module's Register | | | | | | | | `xxxx xxxx` | 31 |
| 02h | PCL | Program Counter's (PC) Least Significant Byte | | | | | | | | `0000 0000` | 19 |
| 03h | STATUS | IRP[(2)] | RP1[(2)] | RP0 | $\overline{\text{TO}}$ | $\overline{\text{PD}}$ | Z | DC | C | `0001 1xxx` | 13 |
| 04h | FSR | Indirect data memory Address Pointer | | | | | | | | `xxxx xxxx` | 20 |
| 05h | PORTA | — | — | I/O Control Registers | | | | | | `--xx xxxx` | 21 |
| 06h | — | Unimplemented | | | | | | | | — | — |
| 07h | PORTC | — | — | I/O Control Registers | | | | | | `--xx xxxx` | 28 |
| 08h | — | Unimplemented | | | | | | | | — | — |
| 09h | — | Unimplemented | | | | | | | | — | — |
| 0Ah | PCLATH | — | — | — | Write buffer for upper 5 bits of program counter | | | | | `---0 0000` | 19 |
| 0Bh | INTCON | GIE | PEIE | T0IE | INTE | RAIE | T0IF | INTF | RAIF | `0000 0000` | 15 |
| 0Ch | PIR1 | EEIF | ADIF | — | — | CMIF | — | — | TMR1IF | `00-- 0--0` | 17 |
| 0Dh | — | Unimplemented | | | | | | | | — | — |
| 0Eh | TMR1L | Holding register for the Least Significant Byte of the 16-bit TMR1 | | | | | | | | `xxxx xxxx` | 34 |
| 0Fh | TMR1H | Holding register for the Most Significant Byte of the 16-bit TMR1 | | | | | | | | `xxxx xxxx` | 34 |
| 10h | T1CON | — | T1GE | T1CKPS1 | T1CKPS0 | T1OSCEN | $\overline{\text{T1SYNC}}$ | TMR1CS | TMR1ON | `-000 0000` | 36 |
| 11h | — | Unimplemented | | | | | | | | — | — |
| 12h | — | Unimplemented | | | | | | | | — | — |
| 13h | — | Unimplemented | | | | | | | | — | — |
| 14h | — | Unimplemented | | | | | | | | — | — |
| 15h | — | Unimplemented | | | | | | | | — | — |
| 16h | — | Unimplemented | | | | | | | | — | — |
| 17h | — | Unimplemented | | | | | | | | — | — |
| 18h | — | Unimplemented | | | | | | | | — | — |
| 19h | CMCON | — | COUT | — | CINV | CIS | CM2 | CM1 | CM0 | `-0-0 0000` | 39 |
| 1Ah | — | Unimplemented | | | | | | | | — | — |
| 1Bh | — | Unimplemented | | | | | | | | — | — |
| 1Ch | — | Unimplemented | | | | | | | | — | — |
| 1Dh | — | Unimplemented | | | | | | | | — | — |
| 1Eh | ADRESH[(3)] | Most Significant 8 bits of the left shifted A/D result or 2 bits of right shifted result | | | | | | | | `xxxx xxxx` | 46 |
| 1Fh | ADCON0[(3)] | ADFM | VCFG | — | CHS2 | CHS1 | CHS0 | GO/$\overline{\text{DONE}}$ | ADON | `00-0 0000` | 47,63 |

**Legend:** – = Unimplemented locations read as '`0`', `u` = unchanged, `x` = unknown, `q` = value depends on condition shaded = unimplemented

**Note 1:** Other (non Power-up) Resets include $\overline{\text{MCLR}}$ Reset, Brown-out Detect and Watchdog Timer Reset during normal operation.

**2:** IRP and RP1 bits are reserved, always maintain these bits clear.

**3:** PIC16F676 only.

### TABLE 2-2: PIC16F630/676 SPECIAL FUNCTION REGISTERS SUMMARY BANK 1

| Addr | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR, BOD | Page |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Bank 1 | | | | | | | | | | | |
| 80h | INDF | Addressing this location uses contents of FSR to address data memory (not a physical register) | | | | | | | | `xxxx xxxx` | `20,63` |
| 81h | OPTION_REG | $\overline{\text{RAPU}}$ | INTEDG | T0CS | T0SE | PSA | PS2 | PS1 | PS0 | `1111 1111` | `14,32` |
| 82h | PCL | Program Counter's (PC) Least Significant Byte | | | | | | | | `0000 0000` | `19` |
| 83h | STATUS | IRP[(2)] | RP1[(2)] | RP0 | $\overline{\text{TO}}$ | $\overline{\text{PD}}$ | Z | DC | C | `0001 1xxx` | `13` |
| 84h | FSR | Indirect data memory Address Pointer | | | | | | | | `xxxx xxxx` | `20` |
| 85h | TRISA | — | — | TRISA5 | TRISA4 | TRISA3 | TRISA2 | TRISA1 | TRISA0 | `--11 1111` | `21` |
| 86h | — | Unimplemented | | | | | | | | — | — |
| 87h | TRISC | — | — | TRISC5 | TRISC4 | TRISC3 | TRISC2 | TRISC1 | TRISC0 | `--11 1111` | — |
| 88h | — | Unimplemented | | | | | | | | — | — |
| 89h | — | Unimplemented | | | | | | | | — | — |
| 8Ah | PCLATH | — | — | — | Write buffer for upper 5 bits of program counter | | | | | `---0 0000` | `19` |
| 8Bh | INTCON | GIE | PEIE | T0IE | INTE | RAIE | T0IF | INTF | RAIF | `0000 0000` | `15` |
| 8Ch | PIE1 | EEIE | ADIE | — | — | CMIE | — | — | TMR1IE | `00-- 0--0` | `16` |
| 8Dh | — | Unimplemented | | | | | | | | — | — |
| 8Eh | PCON | — | — | — | — | — | — | $\overline{\text{POR}}$ | $\overline{\text{BOD}}$ | `---- --qq` | `18` |
| 8Fh | — | | | | | | | | | — | |
| 90h | OSCCAL | CAL5 | CAL4 | CAL3 | CAL2 | CAL1 | CAL0 | — | — | `1000 00--` | `18` |
| 91h | ANSEL[(3)] | ANS7 | ANS6 | ANS5 | ANS4 | ANS3 | ANS2 | ANS1 | ANS0 | `1111 1111` | `48` |
| 92h | — | Unimplemented | | | | | | | | — | — |
| 93h | — | Unimplemented | | | | | | | | — | — |
| 94h | — | Unimplemented | | | | | | | | — | — |
| 95h | WPUA | — | — | WPUA5 | WPUA4 | — | WPUA2 | WPUA1 | WPUA0 | `--11 -111` | `22` |
| 96h | IOCA | — | — | IOCA5 | IOCA4 | IOCA3 | IOCA2 | IOCA1 | IOCA0 | `--00 0000` | `23` |
| 97h | — | Unimplemented | | | | | | | | — | — |
| 98h | — | Unimplemented | | | | | | | | — | — |
| 99h | VRCON | VREN | — | VRR | — | VR3 | VR2 | VR1 | VR0 | `0-0- 0000` | `44` |
| 9Ah | EEDAT | EEPROM data register | | | | | | | | `0000 0000` | `51` |
| 9Bh | EEADR | — | EEPROM address register | | | | | | | `0000 0000` | `51` |
| 9Ch | EECON1 | — | — | — | — | WRERR | WREN | WR | RD | `---- x000` | `52` |
| 9Dh | EECON2 | EEPROM control register 2 (not a physical register) | | | | | | | | `---- ----` | `51` |
| 9Eh | ADRESL[(3)] | Least Significant 2 bits of the left shifted result or 8 bits of the right shifted result | | | | | | | | `xxxx xxxx` | `46` |
| 9Fh | ADCON1[(3)] | — | ADCS2 | ADCS1 | ADCS0 | — | — | — | — | `-000 ----` | `47,63` |

**Legend:** – = Unimplemented locations read as '`0`', `u` = unchanged, `x` = unknown, `q` = value depends on condition, shaded = unimplemented

**Note 1:** Other (non Power-up) Resets include $\overline{\text{MCLR}}$ Reset, Brown-out Detect and Watchdog Timer Reset during normal operation.

**2:** IRP and RP1 bits are reserved, always maintain these bits clear.

**3:** PIC16F676 only.

### 2.2.2.1 STATUS Register

The STATUS register, shown in Register 2-1, contains:

- the arithmetic status of the ALU
- the Reset status
- the bank select bits for data memory (SRAM)

The STATUS register can be the destination for any instruction, like any other register. If the STATUS register is the destination for an instruction that affects the Z, DC or C bits, then the write to these three bits is disabled. These bits are set or cleared according to the device logic. Furthermore, the $\overline{\text{TO}}$ and $\overline{\text{PD}}$ bits are not writable. Therefore, the result of an instruction with the STATUS register as destination may be different than intended.

For example, `CLRF STATUS` will clear the upper three bits and set the Z bit. This leaves the STATUS register as `000u u1uu` (where `u` = unchanged).

It is recommended, therefore, that only `BCF`, `BSF`, `SWAPF` and `MOVWF` instructions are used to alter the STATUS register, because these instructions do not affect any Status bits. For other instructions not affecting any Status bits, see **Section 10.0 "Instruction Set Summary"**.

**Note 1:** Bits IRP and RP1 (STATUS<7:6>) are not used by the PIC16F630/676 and should be maintained as clear. Use of these bits is not recommended, since this may affect upward compatibility with future products.

**2:** The C and DC bits operate as a $\overline{\text{Borrow}}$ and Digit $\overline{\text{Borrow}}$ out bit, respectively, in subtraction. See the `SUBLW` and `SUBWF` instructions for examples.

**REGISTER 2-1: STATUS — STATUS REGISTER (ADDRESS: 03h OR 83h)**

| Reserved | Reserved | R/W-0 | R-1 | R-1 | R/W-x | R/W-x | R/W-x |
|---|---|---|---|---|---|---|---|
| IRP | RP1 | RP0 | $\overline{\text{TO}}$ | $\overline{\text{PD}}$ | Z | DC | C |
| bit 7 | | | | | | | bit 0 |

bit 7 **IRP:** This bit is reserved and should be maintained as '`0`'

bit 6 **RP1:** This bit is reserved and should be maintained as '`0`'

bit 5 **RP0:** Register Bank Select bit (used for direct addressing)
`1` = Bank 1 (80h-FFh)
`0` = Bank 0 (00h-7Fh)

bit 4 **$\overline{\text{TO}}$:** Time-out bit
`1` = After power-up, `CLRWDT` instruction, or `SLEEP` instruction
`0` = A WDT time-out occurred

bit 3 **$\overline{\text{PD}}$**: Power-Down bit
`1` = After power-up or by the `CLRWDT` instruction
`0` = By execution of the `SLEEP` instruction

bit 2 **Z**: Zero bit
`1` = The result of an arithmetic or logic operation is zero
`0` = The result of an arithmetic or logic operation is not zero

bit 1 **DC**: Digit carry/$\overline{\text{borrow}}$ bit (`ADDWF`, `ADDLW`, `SUBLW`, `SUBWF` instructions)
For borrow, the polarity is reversed.
`1` = A carry-out from the 4th low order bit of the result occurred
`0` = No carry-out from the 4th low order bit of the result

bit 0 **C**: Carry/$\overline{\text{borrow}}$ bit (`ADDWF`, `ADDLW`, `SUBLW`, `SUBWF` instructions)
`1` = A carry-out from the Most Significant bit of the result occurred
`0` = No carry-out from the Most Significant bit of the result occurred

**Note:** For $\overline{\text{borrow}}$ the polarity is reversed. A subtraction is executed by adding the two's complement of the second operand. For rotate (`RRF`, `RLF`) instructions, this bit is loaded with either the high or low order bit of the source register.

**Legend:**

| | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' | |
| - n = Value at POR | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |

### 2.2.2.2 OPTION Register

The OPTION register is a readable and writable register, which contains various control bits to configure:

- TMR0/WDT prescaler
- External RA2/INT interrupt
- TMR0
- Weak pull-ups on PORTA

**Note:** To achieve a 1:1 prescaler assignment for TMR0, assign the prescaler to the WDT by setting PSA bit to '`1`' (OPTION<3>). See **Section 4.4 "Prescaler"**.

### REGISTER 2-2: OPTION_REG — OPTION REGISTER (ADDRESS: 81h)

| R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 |
|---|---|---|---|---|---|---|---|
| $\overline{\text{RAPU}}$ | INTEDG | T0CS | T0SE | PSA | PS2 | PS1 | PS0 |
| bit 7 | | | | | | | bit 0 |

bit 7 **$\overline{\text{RAPU}}$:** PORTA Pull-up Enable bit
`1` = PORTA pull-ups are disabled
`0` = PORTA pull-ups are enabled by individual PORT latch values

bit 6 **INTEDG:** Interrupt Edge Select bit
`1` = Interrupt on rising edge of RA2/INT pin
`0` = Interrupt on falling edge of RA2/INT pin

bit 5 **T0CS:** TMR0 Clock Source Select bit
`1` = Transition on RA2/T0CKI pin
`0` = Internal instruction cycle clock (CLKOUT)

bit 4 **T0SE:** TMR0 Source Edge Select bit
`1` = Increment on high-to-low transition on RA2/T0CKI pin
`0` = Increment on low-to-high transition on RA2/T0CKI pin

bit 3 **PSA:** Prescaler Assignment bit
`1` = Prescaler is assigned to the WDT
`0` = Prescaler is assigned to the Timer0 module

bit 2-0 **PS2:PS0:** Prescaler Rate Select bits

| Bit Value | TMR0 Rate | WDT Rate |
|---|---|---|
| `000` | 1 : 2 | 1 : 1 |
| `001` | 1 : 4 | 1 : 2 |
| `010` | 1 : 8 | 1 : 4 |
| `011` | 1 : 16 | 1 : 8 |
| `100` | 1 : 32 | 1 : 16 |
| `101` | 1 : 64 | 1 : 32 |
| `110` | 1 : 128 | 1 : 64 |
| `111` | 1 : 256 | 1 : 128 |

| **Legend:** | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' | |
| - n = Value at POR | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |

### 2.2.2.3 INTCON Register

The INTCON register is a readable and writable register, which contains the various enable and flag bits for TMR0 register overflow, PORTA change and external RA2/INT pin interrupts.

**Note:** Interrupt flag bits are set when an interrupt condition occurs, regardless of the state of its corresponding enable bit or the global enable bit, GIE (INTCON<7>). User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

**REGISTER 2-3: INTCON — INTERRUPT CONTROL REGISTER (ADDRESS: 0Bh OR 8Bh)**

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| GIE | PEIE | T0IE | INTE | RAIE | T0IF | INTF | RAIF |
| bit 7 | | | | | | | bit 0 |

bit 7 **GIE:** Global Interrupt Enable bit
`1` = Enables all unmasked interrupts
`0` = Disables all interrupts

bit 6 **PEIE:** Peripheral Interrupt Enable bit
`1` = Enables all unmasked peripheral interrupts
`0` = Disables all peripheral interrupts

bit 5 **T0IE:** TMR0 Overflow Interrupt Enable bit
`1` = Enables the TMR0 interrupt
`0` = Disables the TMR0 interrupt

bit 4 **INTE:** RA2/INT External Interrupt Enable bit
`1` = Enables the RA2/INT external interrupt
`0` = Disables the RA2/INT external interrupt

bit 3 **RAIE:** Port Change Interrupt Enable bit**(1)**
`1` = Enables the PORTA change interrupt
`0` = Disables the PORTA change interrupt

bit 2 **T0IF:** TMR0 Overflow Interrupt Flag bit**(2)**
`1` = TMR0 register has overflowed (must be cleared in software)
`0` = TMR0 register did not overflow

bit 1 **INTF:** RA2/INT External Interrupt Flag bit
`1` = The RA2/INT external interrupt occurred (must be cleared in software)
`0` = The RA2/INT external interrupt did not occur

bit 0 **RAIF:** Port Change Interrupt Flag bit
`1` = When at least one of the PORTA <5:0> pins changed state (must be cleared in software)
`0` = None of the PORTA <5:0> pins have changed state

**Note 1:** IOCA register must also be enabled.

**2:** T0IF bit is set when Timer0 rolls over. Timer0 is unchanged on Reset and should be initialized before clearing T0IF bit.

| **Legend:** | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as ‘0’ | |
| - n = Value at POR | ‘1’ = Bit is set | ‘0’ = Bit is cleared | x = Bit is unknown |

### 2.2.2.4 PIE1 Register

The PIE1 register contains the interrupt enable bits, as shown in Register 2-4.

| **Note:** | Bit PEIE (INTCON<6>) must be set to enable any peripheral interrupt. |
|---|---|

**REGISTER 2-4: PIE1 — PERIPHERAL INTERRUPT ENABLE REGISTER 1 (ADDRESS: 8Ch)**

| R/W-0 | R/W-0 | U-0 | U-0 | R/W-0 | U-0 | U-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| EEIE | ADIE | — | — | CMIE | — | — | TMR1IE |
| bit 7 | | | | | | | bit 0 |

| | |
|---|---|
| bit 7 | **EEIE:** EE Write Complete Interrupt Enable bit<br>`1` = Enables the EE write complete interrupt<br>`0` = Disables the EE write complete interrupt |
| bit 6 | **ADIE:** A/D Converter Interrupt Enable bit (PIC16F676 only)<br>`1` = Enables the A/D converter interrupt<br>`0` = Disables the A/D converter interrupt |
| bit 5-4 | **Unimplemented:** Read as ‘`0`’ |
| bit 3 | **CMIE:** Comparator Interrupt Enable bit<br>`1` = Enables the comparator interrupt<br>`0` = Disables the comparator interrupt |
| bit 2-1 | **Unimplemented:** Read as ‘`0`’ |
| bit 0 | **TMR1IE:** TMR1 Overflow Interrupt Enable bit<br>`1` = Enables the TMR1 overflow interrupt<br>`0` = Disables the TMR1 overflow interrupt |

| **Legend:** | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as ‘0’ | |
| - n = Value at POR | ‘1’ = Bit is set | ‘0’ = Bit is cleared | x = Bit is unknown |

### 2.2.2.5 PIR1 Register

The PIR1 register contains the interrupt flag bits, as shown in Register 2-5.

| **Note:** | Interrupt flag bits are set when an interrupt condition occurs, regardless of the state of its corresponding enable bit or the global enable bit, GIE (INTCON<7>). User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt. |
|---|---|

**REGISTER 2-5: PIR1 — PERIPHERAL INTERRUPT REGISTER 1 (ADDRESS: 0Ch)**

| R/W-0 | R/W-0 | U-0 | U-0 | R/W-0 | U-0 | U-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| EEIF | ADIF | — | — | CMIF | — | — | TMR1IF |
| bit 7 | | | | | | | bit 0 |

| | |
|---|---|
| bit 7 | **EEIF:** EEPROM Write Operation Interrupt Flag bit<br>`1` = The write operation completed (must be cleared in software)<br>`0` = The write operation has not completed or has not been started |
| bit 6 | **ADIF:** A/D Converter Interrupt Flag bit (PIC16F676 only)<br>`1` = The A/D conversion is complete (must be cleared in software)<br>`0` = The A/D conversion is not complete |
| bit 5-4 | **Unimplemented**: Read as ‘`0`’ |
| bit 3 | **CMIF**: Comparator Interrupt Flag bit<br>`1` = Comparator input has changed (must be cleared in software)<br>`0` = Comparator input has not changed |
| bit 2-1 | **Unimplemented**: Read as ‘`0`’ |
| bit 0 | **TMR1IF**: TMR1 Overflow Interrupt Flag bit<br>`1` = TMR1 register overflowed (must be cleared in software)<br>`0` = TMR1 register did not overflow |

| **Legend:** | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as ‘0’ | |
| - n = Value at POR | ‘1’ = Bit is set | ‘0’ = Bit is cleared | x = Bit is unknown |

### 2.2.2.6 PCON Register

The Power Control (PCON) register contains flag bits to differentiate between a:

- Power-on Reset (POR)
- Brown-out Detect (BOD)
- Watchdog Timer Reset (WDT)
- External $\overline{\text{MCLR}}$ Reset

The PCON Register bits are shown in Register 2-6.

**REGISTER 2-6: PCON — POWER CONTROL REGISTER (ADDRESS: 8Eh)**

| U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | R/W-0 | R/W-x |
|---|---|---|---|---|---|---|---|
| — | — | — | — | — | — | $\overline{\text{POR}}$ | $\overline{\text{BOD}}$ |
| bit 7 | | | | | | | bit 0 |

bit 7-2 **Unimplemented:** Read as ‘`0`’

bit 1 **$\overline{\text{POR}}$:** Power-on Reset Status bit
`1` = No Power-on Reset occurred
`0` = A Power-on Reset occurred (must be set in software after a Power-on Reset occurs)

bit 0 **$\overline{\text{BOD}}$:** Brown-out Detect Status bit
`1` = No Brown-out Detect occurred
`0` = A Brown-out Detect occurred (must be set in software after a Brown-out Detect occurs)

**Legend:**
R = Readable bit W = Writable bit U = Unimplemented bit, read as ‘0’
- n = Value at POR ‘1’ = Bit is set ‘0’ = Bit is cleared x = Bit is unknown

### 2.2.2.7 OSCCAL Register

The Oscillator Calibration register (OSCCAL) is used to calibrate the internal 4 MHz oscillator. It contains 6 bits to adjust the frequency up or down to achieve 4 MHz.

The OSCCAL register bits are shown in Register 2-7.

**REGISTER 2-7: OSCCAL—INTERNAL OSCILLATOR CALIBRATION REGISTER (ADDRESS: 90h)**

| R/W-1 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | U-0 | U-0 |
|---|---|---|---|---|---|---|---|
| CAL5 | CAL4 | CAL3 | CAL2 | CAL1 | CAL0 | — | — |
| bit 7 | | | | | | | bit 0 |

bit 7-2 **CAL5:CAL0:** 6-bit Signed Oscillator Calibration bits
`111111` = Maximum frequency
`100000` = Center frequency
`000000` = Minimum frequency

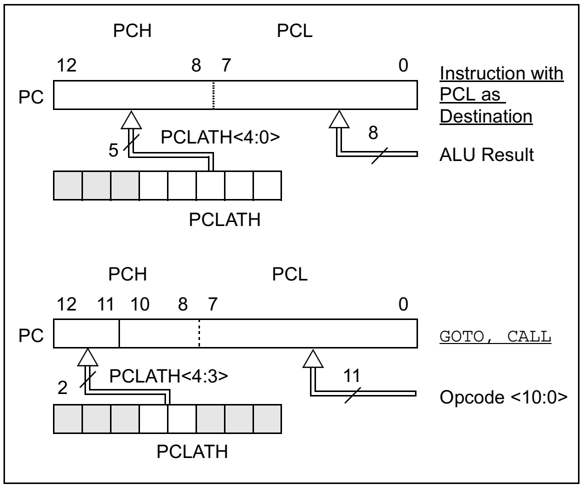
bit 1-0 **Unimplemented:** Read as ‘`0`’

**Legend:**
R = Readable bit W = Writable bit U = Unimplemented bit, read as ‘0’
- n = Value at POR ‘1’ = Bit is set ‘0’ = Bit is cleared x = Bit is unknown

## 2.3 PCL and PCLATH

The program counter (PC) is 13-bits wide. The low byte comes from the PCL register, which is a readable and writable register. The high byte (PC<12:8>) is not directly readable or writable and comes from PCLATH. On any Reset, the PC is cleared. Figure 2-3 shows the two situations for the loading of the PC. The upper example in Figure 2-3 shows how the PC is loaded on a write to PCL (PCLATH<4:0> → PCH). The lower example in Figure 2-3 shows how the PC is loaded during a `CALL` or `GOTO` instruction (PCLATH<4:3> → PCH).

**FIGURE 2-3: LOADING OF PC IN DIFFERENT SITUATIONS**



### 2.3.1 COMPUTED `GOTO`

A computed `GOTO` is accomplished by adding an offset to the program counter (`ADDWF PCL`). When performing a table read using a computed `GOTO` method, care should be exercised if the table location crosses a PCL memory boundary (each 256-byte block). Refer to the Application Note *"Implementing a Table Read"* (AN556).

### 2.3.2 STACK

The PIC16F630/676 family has an 8-level x 13-bit wide hardware stack (see Figure 2-1). The stack space is not part of either program or data space and the Stack Pointer is not readable or writable. The PC is PUSHed onto the stack when a `CALL` instruction is executed or an interrupt causes a branch. The stack is POPed in the event of a `RETURN`, `RETLW` or a `RETFIE` instruction execution. PCLATH is not affected by a PUSH or POP operation.

The stack operates as a circular buffer. This means that after the stack has been PUSHed eight times, the ninth push overwrites the value that was stored from the first push. The tenth push overwrites the second push (and so on).

**Note 1:** There are no Status bits to indicate Stack Overflow or Stack Underflow conditions.

**2:** There are no instructions/mnemonics called PUSH or POP. These are actions that occur from the execution of the `CALL`, `RETURN`, `RETLW` and `RETFIE` instructions or the vectoring to an interrupt address.

## 2.4 Indirect Addressing, INDF and FSR Registers

The INDF register is not a physical register. Addressing the INDF register will cause indirect addressing.

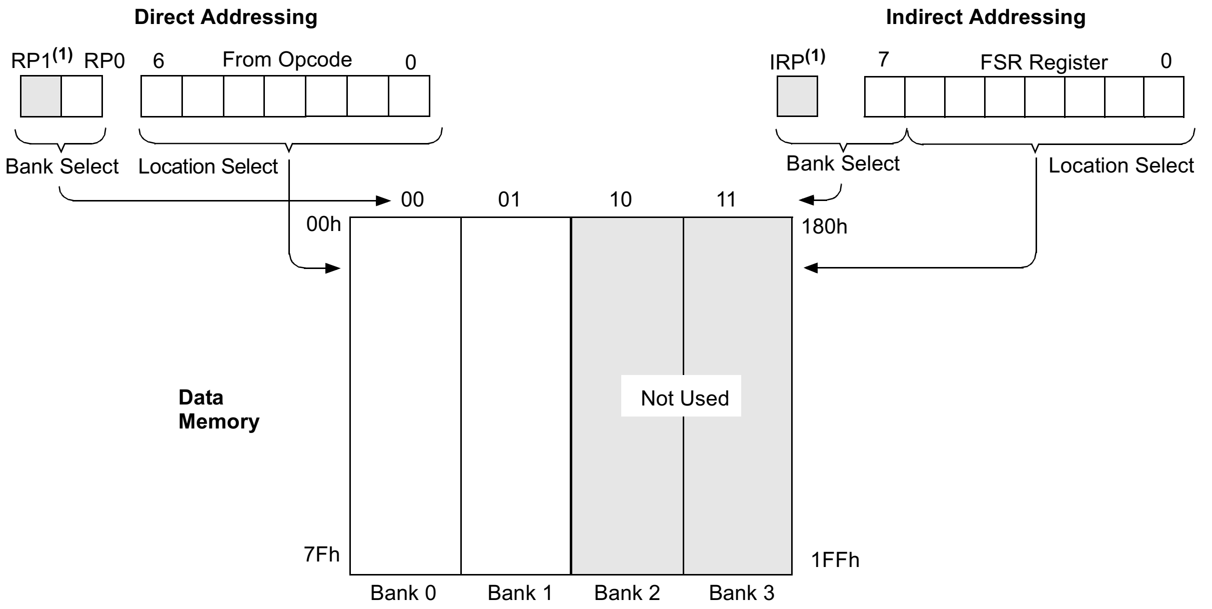
Indirect addressing is possible by using the INDF register. Any instruction using the INDF register actually accesses data pointed to by the File Select Register (FSR). Reading INDF itself indirectly will produce 00h. Writing to the INDF register indirectly results in a no operation (although Status bits may be affected). An effective 9-bit address is obtained by concatenating the 8-bit FSR register and the IRP bit (STATUS<7>), as shown in Figure 2-4.

A simple program to clear RAM location 20h-2Fh using indirect addressing is shown in Example 2-1.

**EXAMPLE 2-1: INDIRECT ADDRESSING**

```
         MOVLW   0x20   ;initialize pointer
         MOVWF   FSR    ;to RAM
NEXT     CLRF    INDF   ;clear INDF register
         INCF    FSR    ;inc pointer
         BTFSS   FSR,4  ;all done?
         GOTO    NEXT   ;no clear next
CONTINUE                ;yes continue
```

**FIGURE 2-4: DIRECT/INDIRECT ADDRESSING PIC16F630/676**



For memory map detail see Figure 2-2.

**Note 1:** The RP1 and IRP bits are reserved; always maintain these bits clear.

## 3.0 PORTS A AND C

There are as many as twelve general purpose I/O pins available. Depending on which peripherals are enabled, some or all of the pins may not be available as general purpose I/O. In general, when a peripheral is enabled, the associated pin may not be used as a general purpose I/O pin.

**Note:** Additional information on I/O ports may be found in the PIC® Mid-Range Reference Manual, (DS33023)

## 3.1 PORTA and the TRISA Registers

PORTA is an 6-bit wide, bidirectional port. The corresponding data direction register is TRISA. Setting a TRISA bit (= `1`) will make the corresponding PORTA pin an input (i.e., put the corresponding output driver in a High-Impedance mode). Clearing a TRISA bit (= `0`) will make the corresponding PORTA pin an output (i.e., put the contents of the output latch on the selected pin). The exception is RA3, which is input only and its TRIS bit will always read as ‘`1`’. Example 3-1 shows how to initialize PORTA.

Reading the PORTA register reads the status of the pins, whereas writing to it will write to the PORT latch. All write operations are read-modify-write operations. Therefore, a write to a port implies that the port pins are read, this value is modified and then written to the PORT data latch. RA3 reads ‘`0`’ when MCLREN = `1`.

The TRISA register controls the direction of the PORTA pins, even when they are being used as analog inputs. The user must ensure the bits in the TRISA register are maintained set when using them as analog inputs. I/O pins configured as analog input always read ‘`0`’.

**Note:** The ANSEL (91h) and CMCON (19h) registers must be initialized to configure an analog channel as a digital input. Pins configured as analog inputs will read ‘`0`’. The ANSEL register is defined for the PIC16F676.

**EXAMPLE 3-1: INITIALIZING PORTA**

```
BCF     STATUS,RP0      ;Bank 0
CLRF    PORTA           ;Init PORTA
MOVLW   05h             ;Set RA<2:0> to
MOVWF   CMCON           ;digital I/O
BSF     STATUS,RP0      ;Bank 1
CLRF    ANSEL           ;digital I/O
MOVLW   0Ch             ;Set RA<3:2> as inputs
MOVWF   TRISA           ;and set RA<5:4,1:0>
                        ;as outputs
BCF     STATUS,RP0      ;Bank 0
```

## 3.2 Additional Pin Functions

Every PORTA pin on the PIC16F630/676 has an interrupt-on-change option and every PORTA pin, except RA3, has a weak pull-up option. The next two sections describe these functions.

### 3.2.1 WEAK PULL-UP

Each of the PORTA pins, except RA3, has an individually configurable weak internal pull-up. Control bits WPUAx enable or disable each pull-up. Refer to Register 3-3. Each weak pull-up is automatically turned off when the port pin is configured as an output. The pull-ups are disabled on a Power-on Reset by the $\overline{\text{RAPU}}$ bit (OPTION<7>).

**REGISTER 3-1: PORTA — PORTA REGISTER (ADDRESS: 05h)**

| U-0 | U-0 | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x |
|---|---|---|---|---|---|---|---|
| — | — | RA5 | RA4 | RA3 | RA2 | RA1 | RA0 |
| bit 7 | | | | | | | bit 0 |

bit 7-6: **Unimplemented**: Read as ‘`0`’

bit 5-0: **PORTA<5:0>**: PORTA I/O pin bits
`1` = Port pin is >$V_{IH}$
`0` = Port pin is <$V_{IL}$

| **Legend:** | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as ‘0’ | |
| - n = Value at POR | ‘`1`’ = Bit is set | ‘`0`’ = Bit is cleared | x = Bit is unknown |

**REGISTER 3-2: TRISA — PORTA TRI-STATE REGISTER (ADDRESS: 85h)**

| U-0 | U-0 | R/W-x | R/W-x | R-1 | R/W-x | R/W-x | R/W-x |
|---|---|---|---|---|---|---|---|
| — | — | TRISA5 | TRISA4 | TRISA3 | TRISA2 | TRISA1 | TRISA0 |
| bit 7 | | | | | | | bit 0 |

bit 7-6: **Unimplemented**: Read as '`0`'

bit 5-0: **TRISA<5:0>**: PORTA Tri-State Control bits
`1` = PORTA pin configured as an input (tri-stated)
`0` = PORTA pin configured as an output

**Note:** TRISA<3> always reads `1`.

| **Legend:** | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '`0`' | |
| - n = Value at POR | '`1`' = Bit is set | '`0`' = Bit is cleared | `x` = Bit is unknown |

**REGISTER 3-3: WPUA — WEAK PULL-UP REGISTER (ADDRESS: 95h)**

| U-0 | U-0 | R/W-1 | R/W-1 | U-0 | R/W-1 | R/W-1 | R/W-1 |
|---|---|---|---|---|---|---|---|
| — | — | WPUA5 | WPUA4 | — | WPUA2 | WPUA1 | WPUA0 |
| bit 7 | | | | | | | bit 0 |

bit 7-6 **Unimplemented:** Read as '`0`'

bit 5-4 **WPUA<5:4>:** Weak Pull-up Register bits
`1` = Pull-up enabled
`0` = Pull-up disabled

bit 3 **Unimplemented:** Read as '`0`'

bit 2-0 **WPUA<2:0>:** Weak Pull-up Register bits
`1` = Pull-up enabled
`0` = Pull-up disabled

**Note 1:** Global $\overline{\text{RAPU}}$ must be enabled for individual pull-ups to be enabled.
**2:** The weak pull-up device is automatically disabled if the pin is in Output mode (TRISA = `0`).

| **Legend:** | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' | |
| - n = Value at POR | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |

### 3.2.2 INTERRUPT-ON-CHANGE

Each of the PORTA pins is individually configurable as an interrupt-on-change pin. Control bits IOCAx enable or disable the interrupt function for each pin. Refer to Register 3-4. The interrupt-on-change is disabled on a Power-on Reset.

For enabled interrupt-on-change pins, the values are compared with the old value latched on the last read of PORTA. The 'mismatch' outputs of the last read are OR'd together to set, the PORTA Change Interrupt flag bit (RAIF) in the INTCON register.

This interrupt can wake the device from Sleep. The user, in the Interrupt Service Routine, can clear the interrupt in the following manner:

a) Any read or write of PORTA. This will end the mismatch condition.
b) Clear the flag bit RAIF.

A mismatch condition will continue to set flag bit RAIF. Reading PORTA will end the mismatch condition and allow flag bit RAIF to be cleared.

**Note:** If a change on the I/O pin should occur when the read operation is being executed (start of the Q2 cycle), then the RAIF interrupt flag may not get set.

### REGISTER 3-4: IOCA — INTERRUPT-ON-CHANGE PORTA REGISTER (ADDRESS: 96h)

| U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | — | IOCA5 | IOCA4 | IOCA3 | IOCA2 | IOCA1 | IOCA0 |
| bit 7 | | | | | | | bit 0 |

bit 7-6 **Unimplemented:** Read as '`0`'

bit 5-0 **IOCA<5:0>:** Interrupt-on-Change PORTA Control bits
`1` = Interrupt-on-change enabled
`0` = Interrupt-on-change disabled

**Note:** Global Interrupt Enable (GIE) must be enabled for individual interrupts to be recognized.

| **Legend:** | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '`0`' | |
| - n = Value at POR | '`1`' = Bit is set | '`0`' = Bit is cleared | x = Bit is unknown |

### 3.2.3 PIN DESCRIPTIONS AND DIAGRAMS

Each PORTA pin is multiplexed with other functions. The pins and their combined functions are briefly described here. For specific information about individual functions such as the comparator or the A/D, refer to the appropriate section in this Data Sheet.

#### 3.2.3.1 RA0/AN0/CIN+

Figure 3-1 shows the diagram for this pin. The RA0 pin is configurable to function as one of the following:

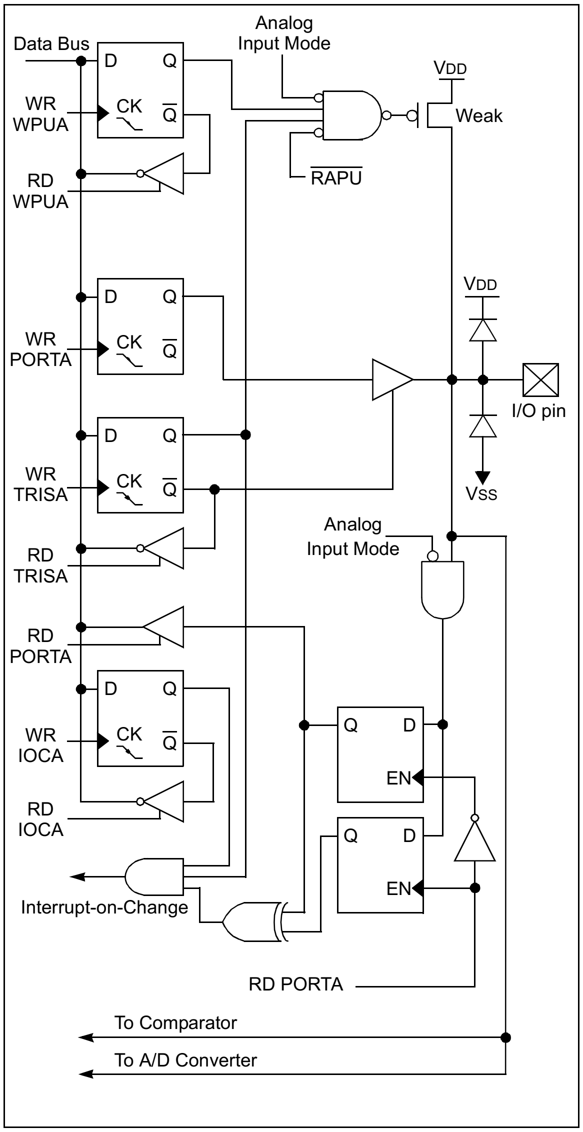
- a general purpose I/O
- an analog input for the A/D (PIC16F676 only)
- an analog input to the comparator

#### 3.2.3.2 RA1/AN1/CIN-/VREF

Figure 3-1 shows the diagram for this pin. The RA1 pin is configurable to function as one of the following:

- as a general purpose I/O
- an analog input for the A/D (PIC16F676 only)
- an analog input to the comparator
- a voltage reference input for the A/D (PIC16F676 only)

**FIGURE 3-1: BLOCK DIAGRAM OF RA0 AND RA1 PINS**

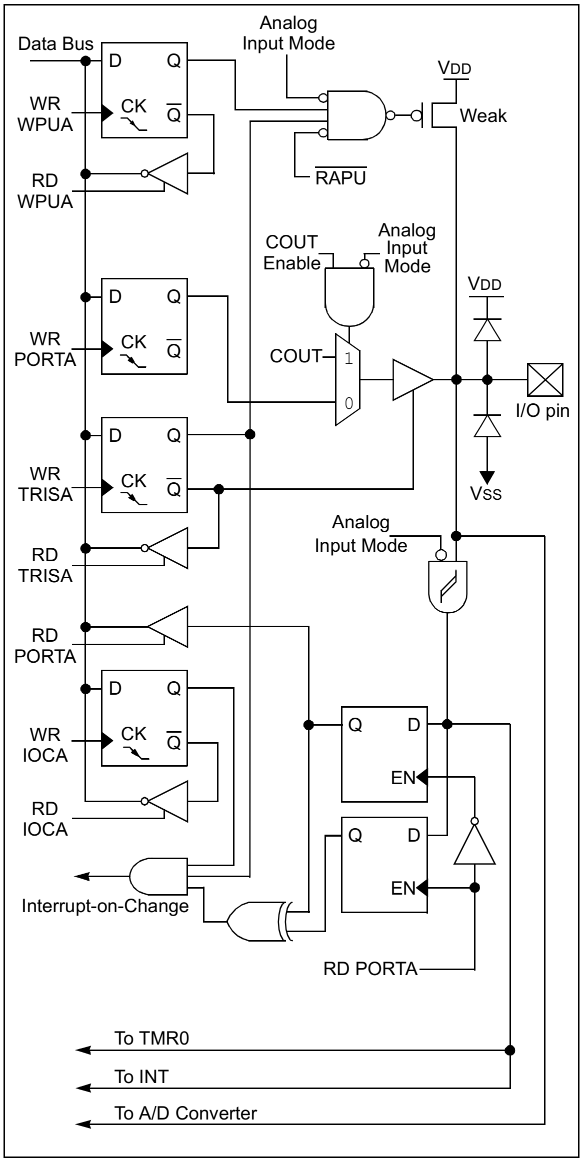


### 3.2.3.3 RA2/AN2/T0CKI/INT/COUT

Figure 3-2 shows the diagram for this pin. The RA2 pin is configurable to function as one of the following:

- a general purpose I/O
- an analog input for the A/D (PIC16F676 only)
- a digital output from the comparator
- the clock input for TMR0
- an external edge triggered interrupt

**FIGURE 3-2: BLOCK DIAGRAM OF RA2**

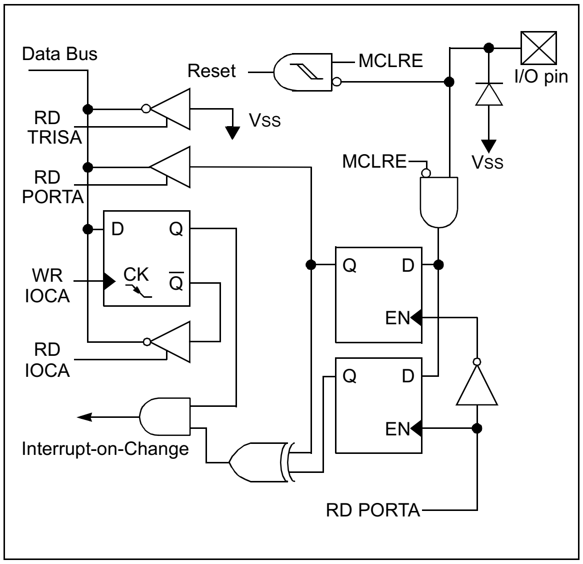


### 3.2.3.4 RA3/$\overline{\text{MCLR}}$/VPP

Figure 3-3 shows the diagram for this pin. The RA3 pin is configurable to function as one of the following:

- a general purpose input
- as Master Clear Reset

**FIGURE 3-3: BLOCK DIAGRAM OF RA3**

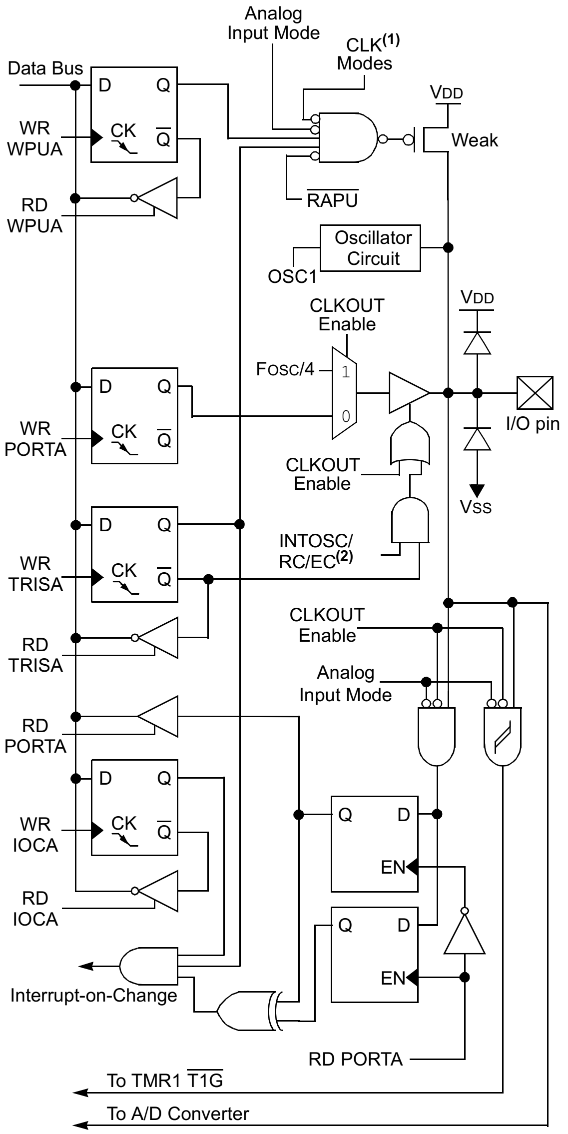


#### 3.2.3.5 RA4/AN3/$\overline{\text{T1G}}$/OSC2/CLKOUT

Figure 3-4 shows the diagram for this pin. The RA4 pin is configurable to function as one of the following:

- a general purpose I/O
- an analog input for the A/D (PIC16F676 only)
- a TMR1 gate input
- a crystal/resonator connection
- a clock output

**FIGURE 3-4: BLOCK DIAGRAM OF RA4**



**Note 1:** CLK modes are XT, HS, LP, LPTMR1 and CLKOUT Enable.

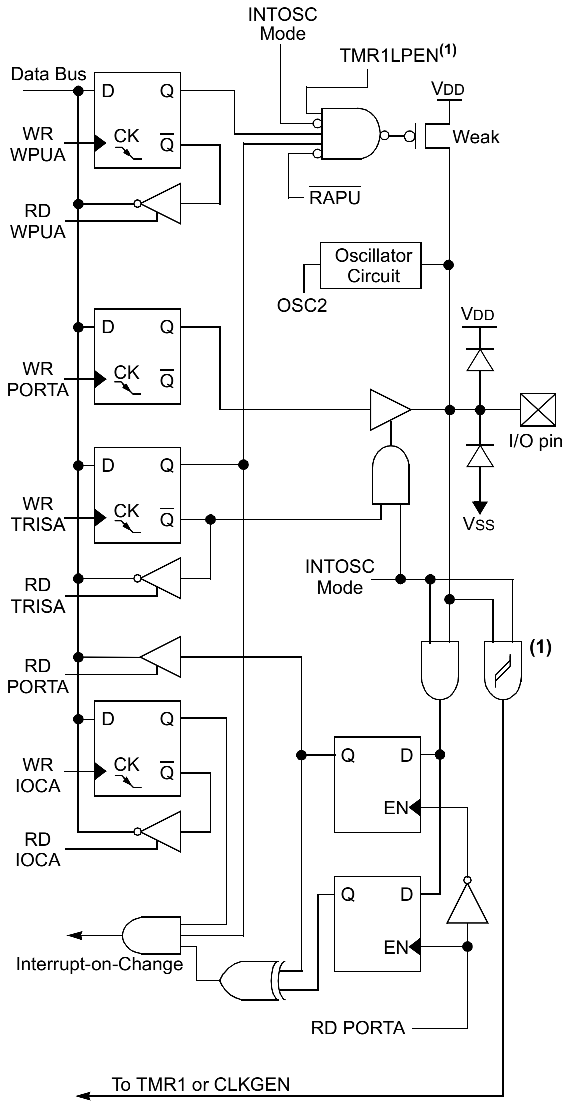
**2:** With CLKOUT option.

#### 3.2.3.6 RA5/T1CKI/OSC1/CLKIN

Figure 3-5 shows the diagram for this pin. The RA5 pin is configurable to function as one of the following:

- a general purpose I/O
- a TMR1 clock input
- a crystal/resonator connection
- a clock input

**FIGURE 3-5: BLOCK DIAGRAM OF RA5**



**Note 1:** Timer1 LP Oscillator enabled.

**2:** When using Timer1 with LP oscillator, the Schmitt Trigger is by-passed.

### TABLE 3-1: SUMMARY OF REGISTERS ASSOCIATED WITH PORTA

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR, BOD | Value on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 05h | PORTA | — | — | RA5 | RA4 | RA3 | RA2 | RA1 | RA0 | `--xx xxxx` | `--uu uuuu` |
| 0Bh/8Bh | INTCON | GIE | PEIE | T0IE | INTE | RAIE | T0IF | INTF | RAIF | `0000 0000` | `0000 000u` |
| 19h | CMCON | — | COUT | — | CINV | CIS | CM2 | CM1 | CM0 | `-0-0 0000` | `-0-0 0000` |
| 81h | OPTION_REG | $\overline{\text{RAPU}}$ | INTEDG | T0CS | T0SE | PSA | PS2 | PS1 | PS0 | `1111 1111` | `1111 1111` |
| 85h | TRISA | — | — | TRISA5 | TRISA4 | TRISA3 | TRISA2 | TRISA1 | TRISA0 | `--11 1111` | `--11 1111` |
| 91h | ANSEL(1) | ANS7 | ANS6 | ANS5 | ANS4 | ANS3 | ANS2 | ANS1 | ANS0 | `1111 1111` | `1111 1111` |
| 95h | WPUA | — | — | WPUA5 | WPUA4 | — | WPUA2 | WPUA1 | WPUA0 | `--11 -111` | `--11 -111` |
| 96h | IOCA | — | — | IOCA5 | IOCA4 | IOCA3 | IOCA2 | IOCA1 | IOCA0 | `--00 0000` | `--00 0000` |

**Note 1:** PIC16F676 only.

**Legend:** `x` = unknown, `u` = unchanged, `-` = unimplemented locations read as ‘`0`’. Shaded cells are not used by PORTA.

## 3.3 PORTC

PORTC is a general purpose I/O port consisting of 6 bidirectional pins. The pins can be configured for either digital I/O or analog input to A/D converter. For specific information about individual functions such as the comparator or the A/D, refer to the appropriate section in this Data Sheet.

**Note:** The ANSEL register (91h) must be clear to configure an analog channel as a digital input. Pins configured as analog inputs will read '0'. The ANSEL register is defined for the PIC16F676.

**EXAMPLE 3-2: INITIALIZING PORTC**

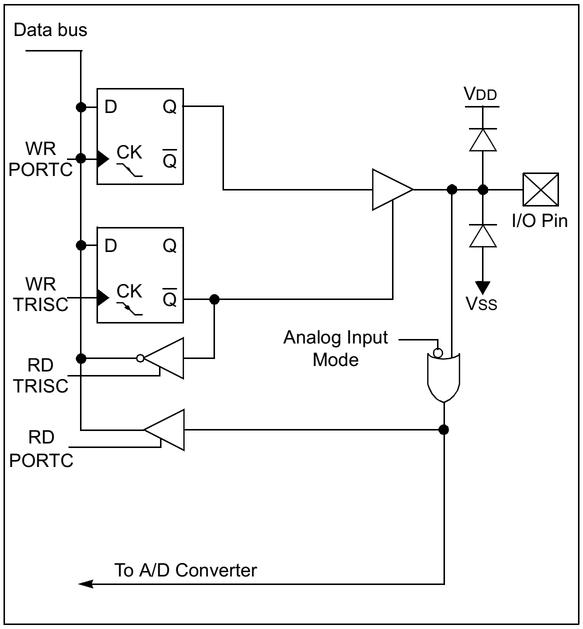
```
BCF     STATUS,RP0     ;Bank 0
CLRF    PORTC          ;Init PORTC
BSF     STATUS,RP0     ;Bank 1
CLRF    ANSEL          ;digital I/O
MOVLW   0Ch            ;Set RC<3:2> as inputs
MOVWF   TRISC          ;and set RC<5:4,1:0>
                       ;as outputs
BCF     STATUS,RP0     ;Bank 0
```

### 3.3.1 RC0/AN4, RC1/AN5, RC2/AN6, RC3/AN7

The RC0/RC1/RC2/RC3 pins are configurable to function as one of the following:

- a general purpose I/O
- an analog input for the A/D Converter (PIC16F676 only)

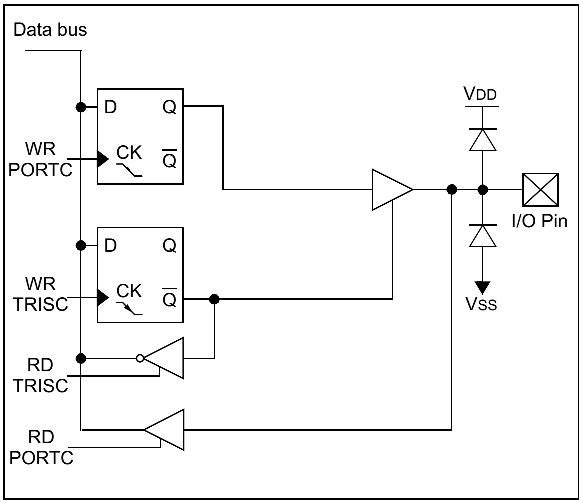
**FIGURE 3-6: BLOCK DIAGRAM OF RC0/RC1/RC2/RC3 PINs**



### 3.3.2 RC4 AND RC5

The RC4 and RC5 pins are configurable to function as a general purpose I/Os.

**FIGURE 3-7: BLOCK DIAGRAM OF RC4 AND RC5 PINS**



### REGISTER 3-5: PORTC — PORTC REGISTER (ADDRESS: 07h)

| U-0 | U-0 | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x |
|---|---|---|---|---|---|---|---|
| — | — | RC5 | RC4 | RC3 | RC2 | RC1 | RC0 |
| bit 7 | | | | | | | bit 0 |

bit 7-6: **Unimplemented**: Read as ‘`0`’

bit 5-0: **PORTC<5:0>**: General Purpose I/O pin bits
`1` = Port pin is >$V_{IH}$
`0` = Port pin is <$V_{IL}$

| **Legend:** | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as ‘`0`’ | |
| - n = Value at POR | ‘`1`’ = Bit is set | ‘`0`’ = Bit is cleared | x = Bit is unknown |

### REGISTER 3-6: TRISC — PORTC TRI-STATE REGISTER (ADDRESS: 87h)

| U-0 | U-0 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 |
|---|---|---|---|---|---|---|---|
| — | — | TRISC5 | TRISC4 | TRISC3 | TRISC2 | TRISC1 | TRISC0 |
| bit 7 | | | | | | | bit 0 |

bit 7-6: **Unimplemented**: Read as ‘0’

bit 5-0: **TRISC<5:0>**: PORTC Tri-State Control bits
`1` = PORTC pin configured as an input (tri-stated)
`0` = PORTC pin configured as an output

| **Legend:** | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as ‘`0`’ | |
| - n = Value at POR | ‘`1`’ = Bit is set | ‘`0`’ = Bit is cleared | x = Bit is unknown |

### TABLE 3-2: SUMMARY OF REGISTERS ASSOCIATED WITH PORTC

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR, BOD | Value on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 07h | PORTC | — | — | RC5 | RC4 | RC3 | RC2 | RC1 | RC0 | `--xx xxxx` | `--uu uuuu` |
| 87h | TRISC | — | — | TRISC5 | TRISC4 | TRISC3 | TRISC2 | TRISC1 | TRISC0 | `--11 1111` | `--11 1111` |
| 91h | ANSEL[(1)] | ANS7 | ANS6 | ANS5 | ANS4 | ANS3 | ANS2 | ANS1 | ANS0 | `1111 1111` | `1111 1111` |

**Note 1:** PIC16F676 only.

**Legend:** `x` = unknown, `u` = unchanged, `-` = unimplemented locations read as ‘`0`’. Shaded cells are not used by PORTC.

NOTES:

## 4.0 TIMER0 MODULE

The Timer0 module timer/counter has the following features:

- 8-bit timer/counter
- Readable and writable
- 8-bit software programmable prescaler
- Internal or external clock select
- Interrupt on overflow from FFh to 00h
- Edge select for external clock

Figure 4-1 is a block diagram of the Timer0 module and the prescaler shared with the WDT.

**Note:** Additional information on the Timer0 module is available in the PIC® Mid-Range Reference Manual, (DS33023).

### 4.1 Timer0 Operation

Timer mode is selected by clearing the T0CS bit (OPTION_REG<5>). In Timer mode, the Timer0 module will increment every instruction cycle (without prescaler). If TMR0 is written, the increment is inhibited for the following two instruction cycles. The user can work around this by writing an adjusted value to the TMR0 register.

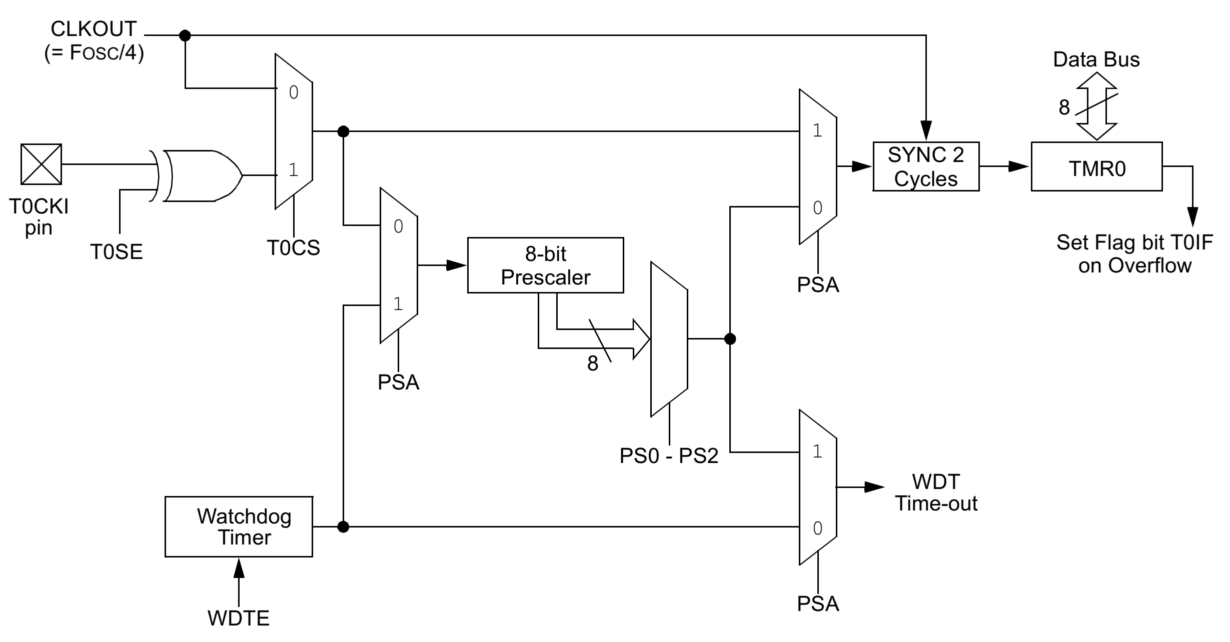
Counter mode is selected by setting the T0CS bit (OPTION_REG<5>). In this mode, the Timer0 module will increment either on every rising or falling edge of pin RA2/T0CKI. The incrementing edge is determined by the source edge (T0SE) control bit (OPTION_REG<4>). Clearing the T0SE bit selects the rising edge.

**Note:** Counter mode has specific external clock requirements. Additional information on these requirements is available in the PIC® Mid-Range Reference Manual, (DS33023).

### 4.2 Timer0 Interrupt

A Timer0 interrupt is generated when the TMR0 register timer/counter overflows from FFh to 00h. This overflow sets the T0IF bit. The interrupt can be masked by clearing the T0IE bit (INTCON<5>). The T0IF bit (INTCON<2>) must be cleared in software by the Timer0 module Interrupt Service Routine before re-enabling this interrupt. The Timer0 interrupt cannot wake the processor from Sleep since the timer is shut-off during Sleep.

**FIGURE 4-1: BLOCK DIAGRAM OF THE TIMER0/WDT PRESCALER**



**Note 1:** T0SE, T0CS, PSA, PS0-PS2 are bits in the OPTION register.

## 4.3 Using Timer0 with an External Clock

When no prescaler is used, the external clock input is the same as the prescaler output. The synchronization of T0CKI, with the internal phase clocks, is accomplished by sampling the prescaler output on the Q2 and Q4 cycles of the internal phase clocks. Therefore, it is necessary for T0CKI to be high for at least 2TOSC (and a small RC delay of 20 ns) and low for at least 2TOSC (and a small RC delay of 20 ns). Refer to the electrical specification of the desired device.

**Note:** The ANSEL (91h) and CMCON (19h) registers must be initialized to configure an analog channel as a digital input. Pins configured as analog inputs will read '`0`'. The ANSEL register is defined for the PIC16F676.

**REGISTER 4-1: OPTION_REG — OPTION REGISTER (ADDRESS: 81h)**

| R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 |
|---|---|---|---|---|---|---|---|
| $\overline{\text{RAPU}}$ | INTEDG | T0CS | T0SE | PSA | PS2 | PS1 | PS0 |
| bit 7 | | | | | | | bit 0 |

bit 7 **$\overline{\text{RAPU}}$:** PORTA Pull-up Enable bit
`1` = PORTA pull-ups are disabled
`0` = PORTA pull-ups are enabled by individual PORT latch values

bit 6 **INTEDG:** Interrupt Edge Select bit
`1` = Interrupt on rising edge of RA2/INT pin
`0` = Interrupt on falling edge of RA2/INT pin

bit 5 **T0CS:** TMR0 Clock Source Select bit
`1` = Transition on RA2/T0CKI pin
`0` = Internal instruction cycle clock (CLKOUT)

bit 4 **T0SE:** TMR0 Source Edge Select bit
`1` = Increment on high-to-low transition on RA2/T0CKI pin
`0` = Increment on low-to-high transition on RA2/T0CKI pin

bit 3 **PSA:** Prescaler Assignment bit
`1` = Prescaler is assigned to the WDT
`0` = Prescaler is assigned to the Timer0 module

bit 2-0 **PS2:PS0:** Prescaler Rate Select bits

| Bit Value | TMR0 Rate | WDT Rate |
|---|---|---|
| `000` | 1 : 2 | 1 : 1 |
| `001` | 1 : 4 | 1 : 2 |
| `010` | 1 : 8 | 1 : 4 |
| `011` | 1 : 16 | 1 : 8 |
| `100` | 1 : 32 | 1 : 16 |
| `101` | 1 : 64 | 1 : 32 |
| `110` | 1 : 128 | 1 : 64 |
| `111` | 1 : 256 | 1 : 128 |

| | | | |
|---|---|---|---|
| **Legend:** | | | |
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' | |
| - n = Value at POR | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |

## 4.4 Prescaler

An 8-bit counter is available as a prescaler for the Timer0 module, or as a postscaler for the Watchdog Timer. For simplicity, this counter will be referred to as “prescaler” throughout this Data Sheet. The prescaler assignment is controlled in software by the control bit PSA (OPTION_REG<3>). Clearing the PSA bit will assign the prescaler to Timer0. Prescale values are selectable via the PS2:PS0 bits (OPTION_REG<2:0>).

The prescaler is not readable or writable. When assigned to the Timer0 module, all instructions writing to the TMR0 register (e.g., `CLRF 1,  MOVWF 1, BSF 1, x....etc.`) will clear the prescaler. When assigned to WDT, a `CLRWDT` instruction will clear the prescaler along with the Watchdog Timer.

### 4.4.1 SWITCHING PRESCALER ASSIGNMENT

The prescaler assignment is fully under software control (i.e., it can be changed “on the fly” during program execution). To avoid an unintended device Reset, the following instruction sequence (Example 4-1) must be executed when changing the prescaler assignment from Timer0 to WDT.

**EXAMPLE 4-1: CHANGING PRESCALER (TIMER0→WDT)**

```
BCF     STATUS,RP0   ;Bank 0
CLRWDT               ;Clear WDT
CLRF    TMR0         ;Clear TMR0 and
                     ; prescaler
BSF     STATUS,RP0   ;Bank 1

MOVLW   b’00101111’  ;Required if desired
MOVWF   OPTION_REG   ; PS2:PS0 is
CLRWDT               ; 000 or 001
                     ;
MOVLW   b’00101xxx’  ;Set postscaler to
MOVWF   OPTION_REG   ; desired WDT rate
BCF     STATUS,RP0   ;Bank 0
```

To change prescaler from the WDT to the TMR0 module, use the sequence shown in Example 4-2. This precaution must be taken even if the WDT is disabled.

**EXAMPLE 4-2: CHANGING PRESCALER (WDT→TIMER0)**

```
CLRWDT               ;Clear WDT and
                     ; postscaler
BSF     STATUS,RP0   ;Bank 1

MOVLW   b’xxxx0xxx’  ;Select TMR0,
                     ; prescale, and
                     ; clock source
MOVWF   OPTION_REG   ;
BCF     STATUS,RP0   ;Bank 0
```

**TABLE 4-1: REGISTERS ASSOCIATED WITH TIMER0**

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR, BOD | Value on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 01h | TMR0 | Timer0 Module Register | | | | | | | | `xxxx xxxx` | `uuuu uuuu` |
| 0Bh/8Bh | INTCON | GIE | PEIE | T0IE | INTE | RAIE | T0IF | INTF | RAIF | `0000 0000` | `0000 000u` |
| 81h | OPTION_REG | $\overline{\text{RAPU}}$ | INTEDG | T0CS | T0SE | PSA | PS2 | PS1 | PS0 | `1111 1111` | `1111 1111` |
| 85h | TRISA | — | — | TRISA5 | TRISA4 | TRISA3 | TRISA2 | TRISA1 | TRISA0 | `--11 1111` | `--11 1111` |

**Legend:** – = Unimplemented locations, read as ‘0’, u = unchanged, x = unknown.
Shaded cells are not used by the Timer0 module.

## 5.0 TIMER1 MODULE WITH GATE CONTROL

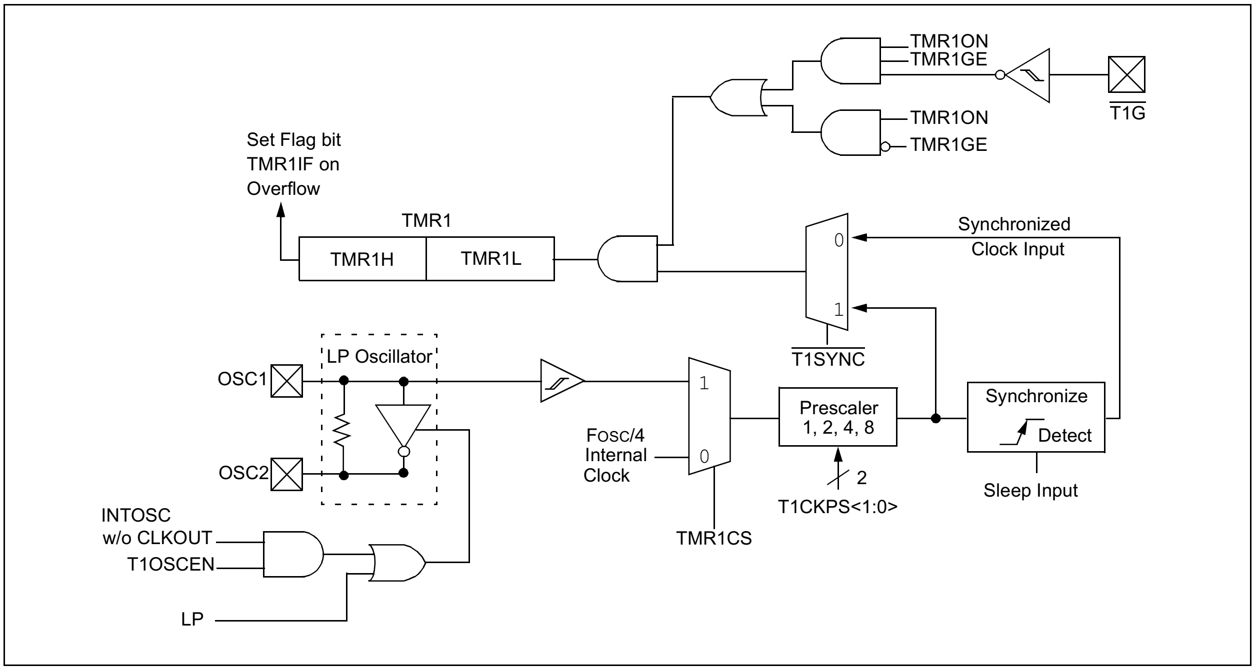
The PIC16F630/676 devices have a 16-bit timer. Figure 5-1 shows the basic block diagram of the Timer1 module. Timer1 has the following features:

- 16-bit timer/counter (TMR1H:TMR1L)
- Readable and writable
- Internal or external clock selection
- Synchronous or asynchronous operation
- Interrupt on overflow from FFFFh to 0000h
- Wake-up upon overflow (Asynchronous mode)
- Optional external enable input ($\overline{\text{T1G}}$)
- Optional LP oscillator

The Timer1 Control register (T1CON), shown in Register 5-1, is used to enable/disable Timer1 and select the various features of the Timer1 module.

| **Note:** | Additional information on timer modules is available in the PIC® Mid-Range Reference Manual, (DS33023). |
|---|---|

**FIGURE 5-1: TIMER1 BLOCK DIAGRAM**



## 5.1 Timer1 Modes of Operation

Timer1 can operate in one of three modes:

- 16-bit timer with prescaler
- 16-bit synchronous counter
- 16-bit asynchronous counter

In Timer mode, Timer1 is incremented on every instruction cycle. In Counter mode, Timer1 is incremented on the rising edge of the external clock input T1CKI. In addition, the Counter mode clock can be synchronized to the microcontroller system clock or run asynchronously.

In counter and timer modules, the counter/timer clock can be gated by the $\overline{\text{T1G}}$ input.

If an external clock oscillator is needed (and the microcontroller is using the INTOSC w/o CLKOUT), Timer1 can use the LP oscillator as a clock source.

| **Note:** | In Counter mode, a falling edge must be registered by the counter prior to the first incrementing rising edge. |
|---|---|

## 5.2 Timer1 Interrupt

The Timer1 register pair (TMR1H:TMR1L) increments to FFFFh and rolls over to 0000h. When Timer1 rolls over, the Timer1 interrupt flag bit (PIR1<0>) is set. To enable the interrupt on rollover, you must set these bits:

- Timer1 interrupt Enable bit (PIE1<0>)
- PEIE bit (INTCON<6>)
- GIE bit (INTCON<7>).

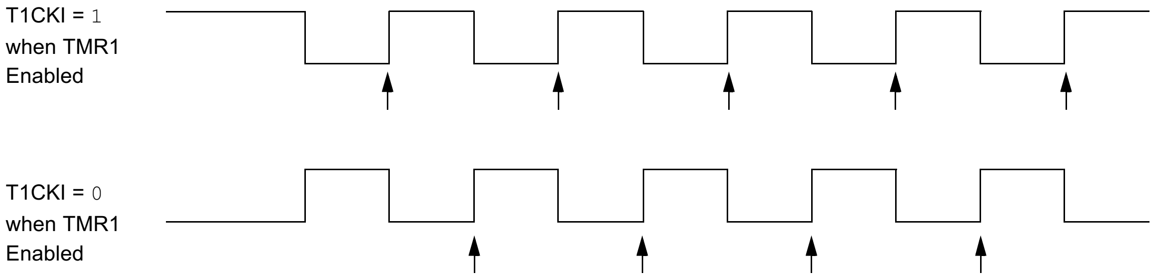
The interrupt is cleared by clearing the TMR1IF in the Interrupt Service Routine.

| **Note:** | The TMR1H:TTMR1L register pair and the TMR1IF bit should be cleared before enabling interrupts. |
|---|---|

## 5.3 Timer1 Prescaler

Timer1 has four prescaler options allowing 1, 2, 4, or 8 divisions of the clock input. The T1CKPS bits (T1CON<5:4>) control the prescale counter. The prescale counter is not directly readable or writable; however, the prescaler counter is cleared upon a write to TMR1H or TMR1L.

**FIGURE 5-2: TIMER1 INCREMENTING EDGE**



**Note 1:** Arrows indicate counter increments.

**2:** In Counter mode, a falling edge must be registered by the counter prior to the first incrementing rising edge of the clock.

### REGISTER 5-1: T1CON — TIMER1 CONTROL REGISTER (ADDRESS: 10h)

| U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | TMR1GE | T1CKPS1 | T1CKPS0 | T1OSCEN | $\overline{\text{T1SYNC}}$ | TMR1CS | TMR1ON |
| bit 7 | | | | | | | bit 0 |

bit 7 **Unimplemented:** Read as ‘0’

bit 6 **TMR1GE:** Timer1 Gate Enable bit
<u>If TMR1ON = `0`:</u>
This bit is ignored
<u>If TMR1ON = `1`:</u>
`1` = Timer1 is on if $\overline{\text{T1G}}$ pin is low
`0` = Timer1 is on

bit 5-4 **T1CKPS1:T1CKPS0:** Timer1 Input Clock Prescale Select bits
`11` = 1:8 Prescale Value
`10` = 1:4 Prescale Value
`01` = 1:2 Prescale Value
`00` = 1:1 Prescale Value

bit 3 **T1OSCEN:** LP Oscillator Enable Control bit
<u>If INTOSC without CLKOUT oscillator is active:</u>
`1` = LP oscillator is enabled for Timer1 clock
`0` = LP oscillator is off
<u>Else:</u>
This bit is ignored

bit 2 **$\overline{\text{T1SYNC}}$:** Timer1 External Clock Input Synchronization Control bit
<u>TMR1CS = `1`:</u>
`1` = Do not synchronize external clock input
`0` = Synchronize external clock input
<u>TMR1CS = `0`:</u>
This bit is ignored. Timer1 uses the internal clock.

bit 1 **TMR1CS:** Timer1 Clock Source Select bit
`1` = External clock from T1OSO/T1CKI pin (on the rising edge)
`0` = Internal clock (FOSC/4)

bit 0 **TMR1ON:** Timer1 On bit
`1` = Enables Timer1
`0` = Stops Timer1

| **Legend:** | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as ‘0’ | |
| - n = Value at POR | ‘1’ = Bit is set | ‘0’ = Bit is cleared | x = Bit is unknown |

## 5.4 Timer1 Operation in Asynchronous Counter Mode

If control bit $\overline{\text{T1SYNC}}$ (T1CON<2>) is set, the external clock input is not synchronized. The timer continues to increment asynchronous to the internal phase clocks. The timer will continue to run during Sleep and can generate an interrupt on overflow, which will wake-up the processor. However, special precautions in software are needed to read/write the timer (Section 5.4.1).

| **Note:** | The ANSEL (91h) and CMCON (19h) registers must be initialized to configure an analog channel as a digital input. Pins configured as analog inputs will read '`0`'. The ANSEL register is defined for the PIC16F676. |
|---|---|

### 5.4.1 READING AND WRITING TIMER1 IN ASYNCHRONOUS COUNTER MODE

Reading TMR1H or TMR1L, while the timer is running from an external asynchronous clock, will ensure a valid read (taken care of in hardware). However, the user should keep in mind that reading the 16-bit timer in two 8-bit values itself, poses certain problems, since the timer may overflow between the reads.

For writes, it is recommended that the user simply stop the timer and write the desired values. A write contention may occur by writing to the timer registers, while the register is incrementing. This may produce an unpredictable value in the timer register.

Reading the 16-bit value requires some care. Examples 12-2 and 12-3 in the PIC® Mid-Range MCU Family Reference Manual (DS33023) show how to read and write Timer1 when it is running in Asynchronous mode.

## 5.5 Timer1 Oscillator

A crystal oscillator circuit is built-in between pins OSC1 (input) and OSC2 (amplifier output). It is enabled by setting control bit T1OSCEN (T1CON<3>). The oscillator is a low power oscillator rated up to 32 kHz. It will continue to run during Sleep. It is primarily intended for a 32 kHz crystal. Table 9-2 shows the capacitor selection for the Timer1 oscillator.

The Timer1 oscillator is shared with the system LP oscillator. Thus, Timer1 can use this mode only when the system clock is derived from the internal oscillator. As with the system LP oscillator, the user must provide a software time delay to ensure proper oscillator start-up.

TRISA5 and TRISA4 bits are set when the Timer1 oscillator is enabled. RA5 and RA4 read as '`0`' and TRISA5 and TRISA4 bits read as '`1`'.

| **Note:** | The oscillator requires a start-up and stabilization time before use. Thus, T1OSCEN should be set and a suitable delay observed prior to enabling Timer1. |
|---|---|

## 5.6 Timer1 Operation During Sleep

Timer1 can only operate during Sleep when setup in Asynchronous Counter mode. In this mode, an external crystal or clock source can be used to increment the counter. To setup the timer to wake the device:

- Timer1 must be on (T1CON<0>)
- TMR1IE bit (PIE1<0>) must be set
- PEIE bit (INTCON<6>) must be set

The device will wake-up on an overflow. If the GIE bit (INTCON<7>) is set, the device will wake-up and jump to the Interrupt Service Routine on an overflow.

**TABLE 5-1: REGISTERS ASSOCIATED WITH TIMER1 AS A TIMER/COUNTER**

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR, BOD | Value on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0Bh/8Bh | INTCON | GIE | PEIE | T0IE | INTE | RAIE | T0IF | INTF | RAIF | `0000 0000` | `0000 000u` |
| 0Ch | PIR1 | EEIF | ADIF | — | — | CMIF | — | — | TMR1IF | `00-- 0--0` | `00-- 0--0` |
| 0Eh | TMR1L | Holding Register for the Least Significant Byte of the 16-bit TMR1 Register | | | | | | | | `xxxx xxxx` | `uuuu uuuu` |
| 0Fh | TMR1H | Holding Register for the Most Significant Byte of the 16-bit TMR1 Register | | | | | | | | `xxxx xxxx` | `uuuu uuuu` |
| 10h | T1CON | — | TMR1GE | T1CKPS1 | T1CKPS0 | T1OSCEN | $\overline{\text{T1SYNC}}$ | TMR1CS | TMR1ON | `-000 0000` | `-uuu uuuu` |
| 8Ch | PIE1 | EEIE | ADIE | — | — | CMIE | — | — | TMR1IE | `00-- 0--0` | `00-- 0--0` |

**Legend:** `x` = unknown, `u` = unchanged, `-` = unimplemented, read as '`0`'. Shaded cells are not used by the Timer1 module.

**NOTES:**

## 6.0 COMPARATOR MODULE

The PIC16F630/676 devices have one analog comparator. The inputs to the comparator are multiplexed with the RA0 and RA1 pins. There is an on-chip Comparator Voltage Reference that can also be applied to an input of the comparator. In addition, RA2 can be configured as the comparator output. The Comparator Control Register (CMCON), shown in Register 6-1, contains the bits to control the comparator.

**REGISTER 6-1: CMCON — COMPARATOR CONTROL REGISTER (ADDRESS: 19h)**

| U-0 | R-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | COUT | — | CINV | CIS | CM2 | CM1 | CM0 |
| bit 7 | | | | | | | bit 0 |

| | |
|---|---|
| bit 7 | **Unimplemented**: Read as '0' |
| bit 6 | **COUT**: Comparator Output bit<br>When CINV = 0:<br>1 = VIN+ > VIN-<br>0 = VIN+ < VIN-<br>When CINV = 1:<br>1 = VIN+ < VIN-<br>0 = VIN+ > VIN- |
| bit 5 | **Unimplemented**: Read as '0' |
| bit 4 | **CINV**: Comparator Output Inversion bit<br>1 = Output inverted<br>0 = Output not inverted |
| bit 3 | **CIS**: Comparator Input Switch bit<br>When CM2:CM0 = 110 or 101:<br>1 = VIN- connects to CIN+<br>0 = VIN- connects to CIN- |
| bit 2-0 | **CM2:CM0**: Comparator Mode bits<br>Figure 6-2 shows the Comparator modes and CM2:CM0 bit settings |

**Legend:**

| | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' | |
| - n = Value at POR | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |

## 6.1 Comparator Operation

A single comparator is shown in Figure 6-1, along with the relationship between the analog input levels and the digital output. When the analog input at VIN+ is less than the analog input VIN-, the output of the comparator is a digital low level. When the analog input at VIN+ is greater than the analog input VIN-, the output of the comparator is a digital high level. The shaded areas of the output of the comparator in Figure 6-1 represent the uncertainty due to input offsets and response time.

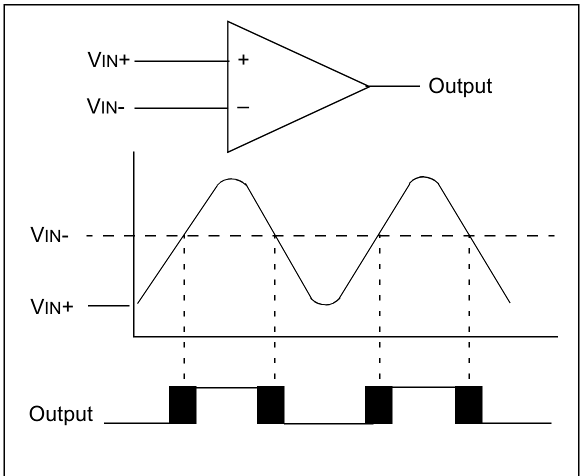
**Note:** To use CIN+ and CIN- pins as analog inputs, the appropriate bits must be programmed in the CMCON (19h) register.

The polarity of the comparator output can be inverted by setting the CINV bit (CMCON<4>). Clearing CINV results in a non-inverted output. A complete table showing the output state versus input conditions and the polarity bit is shown in Table 6-1.

**TABLE 6-1: OUTPUT STATE VS. INPUT CONDITIONS**

| Input Conditions | CINV | COUT |
|---|---|---|
| VIN- > VIN+ | `0` | `0` |
| VIN- < VIN+ | `0` | `1` |
| VIN- > VIN+ | `1` | `1` |
| VIN- < VIN+ | `1` | `0` |

**FIGURE 6-1: SINGLE COMPARATOR**



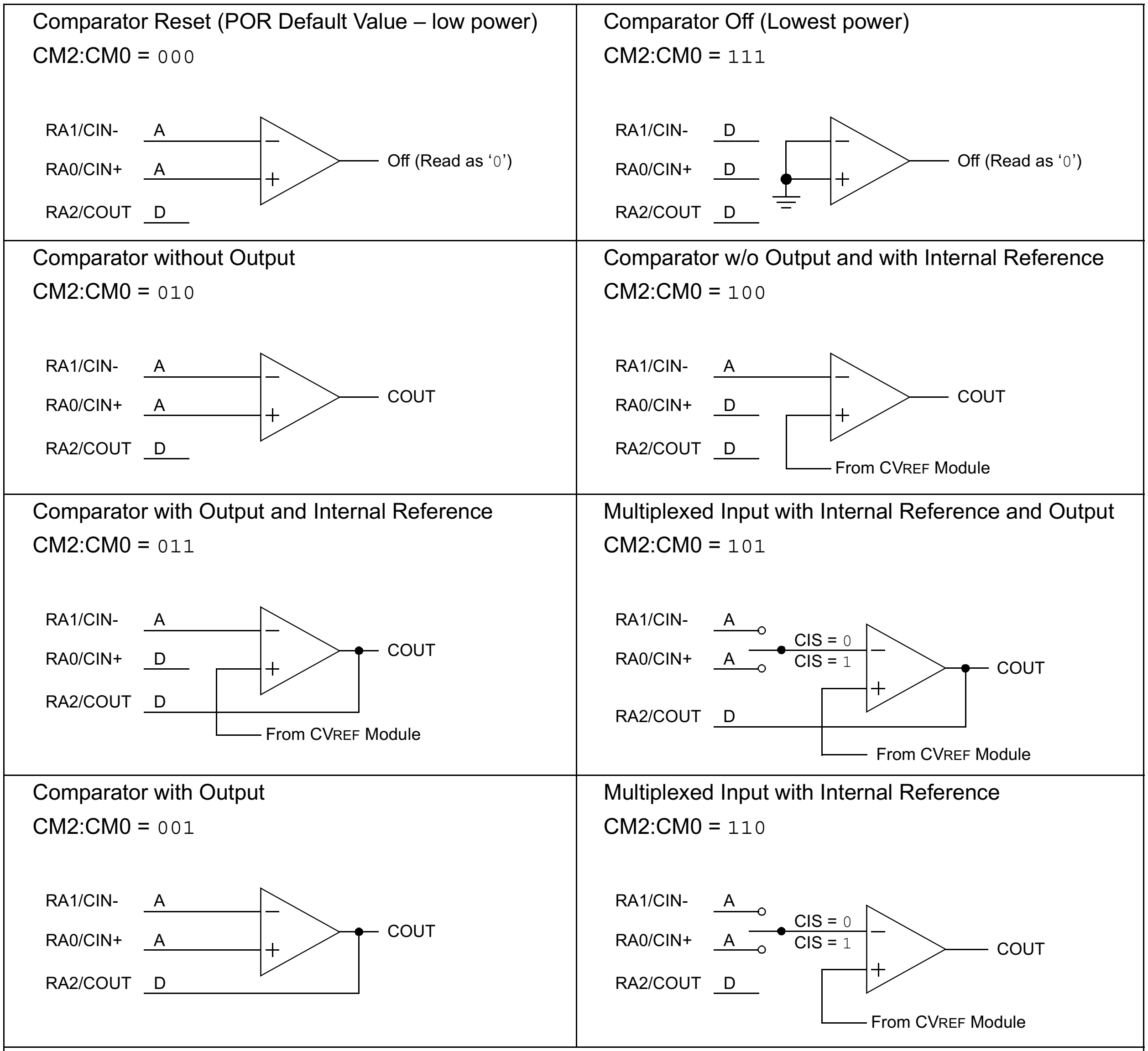
**Note:** CINV bit (CMCON<4>) is clear.

## 6.2 Comparator Configuration

There are eight modes of operation for the comparator. The CMCON register, shown in Register 6-1, is used to select the mode. Figure 6-2 shows the eight possible modes. The TRISA register controls the data direction of the comparator pins for each mode. If the Comparator mode is changed, the comparator output level may not be valid for a specified period of time. Refer to the specifications in **Section 12.0 “Electrical Specifications”**.

| **Note:** | Comparator interrupts should be disabled during a Comparator mode change. Otherwise, a false interrupt may occur. |
|---|---|

**FIGURE 6-2: COMPARATOR I/O OPERATING MODES**


Comparator Reset (POR Default Value – low power)
CM2:CM0 = `000`

Comparator Off (Lowest power)
CM2:CM0 = `111`

Comparator without Output
CM2:CM0 = `010`

Comparator w/o Output and with Internal Reference
CM2:CM0 = `100`

Comparator with Output and Internal Reference
CM2:CM0 = `011`

Multiplexed Input with Internal Reference and Output
CM2:CM0 = `101`

Comparator with Output
CM2:CM0 = `001`

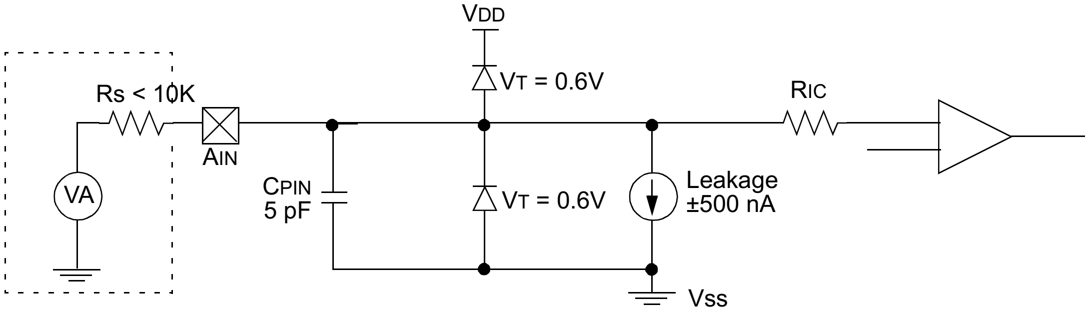
Multiplexed Input with Internal Reference
CM2:CM0 = `110`

A = Analog Input, ports always reads ‘0’
D = Digital Input
CIS = Comparator Input Switch (CMCON<3>)

## 6.3 Analog Input Connection Considerations

A simplified circuit for an analog input is shown in Figure 6-3. Since the analog pins are connected to a digital output, they have reverse biased diodes to VDD and VSS. The analog input, therefore, must be between VSS and VDD. If the input voltage deviates from this range by more than 0.6V in either direction, one of the diodes is forward biased and a latch-up may occur. A maximum source impedance of 10 kΩ is recommended for the analog sources. Any external component connected to an analog input pin, such as a capacitor or a Zener diode, should have very little leakage current.

**FIGURE 6-3: ANALOG INPUT MODE**



**Legend:**
CPIN = Input Capacitance
VT = Threshold Voltage
ILEAKAGE = Leakage Current at the pin due to Various Junctions
RIC = Interconnect Resistance
RS = Source Impedance
VA = Analog Voltage

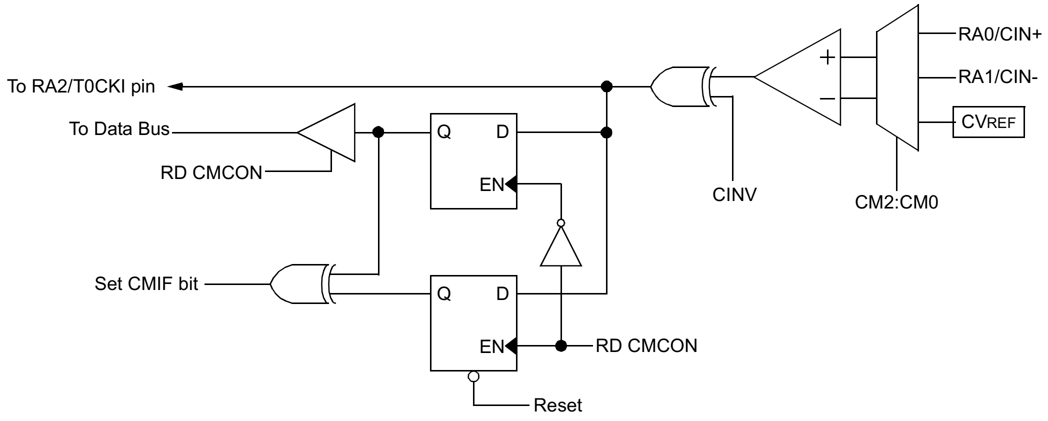
## 6.4 Comparator Output

The comparator output, COUT, is read through the CMCON register. This bit is read-only. The comparator output may also be directly output to the RA2 pin in three of the eight possible modes, as shown in Figure 6-2. When in one of these modes, the output on RA2 is asynchronous to the internal clock. Figure 6-4 shows the comparator output block diagram.

The TRISA<2> bit functions as an output enable/ disable for the RA2 pin while the comparator is in an Output mode.

> **Note 1:** When reading the PORTA register, all pins configured as analog inputs will read as ‘0’. Pins configured as digital inputs will convert an analog input according to the TTL input specification.
>
> **2:** Analog levels on any pin that is defined as a digital input, may cause the input buffer to consume more current than is specified.

**FIGURE 6-4: MODIFIED COMPARATOR OUTPUT BLOCK DIAGRAM**



## 6.5 Comparator Reference

The comparator module also allows the selection of an internally generated voltage reference for one of the comparator inputs. The internal reference signal is used for four of the eight Comparator modes. The VRCON register, Register 6-2, controls the voltage reference module shown in Figure 6-5.

### 6.5.1 CONFIGURING THE VOLTAGE REFERENCE

The voltage reference can output 32 distinct voltage levels, 16 in a high range and 16 in a low range.

The following equations determine the output voltages:

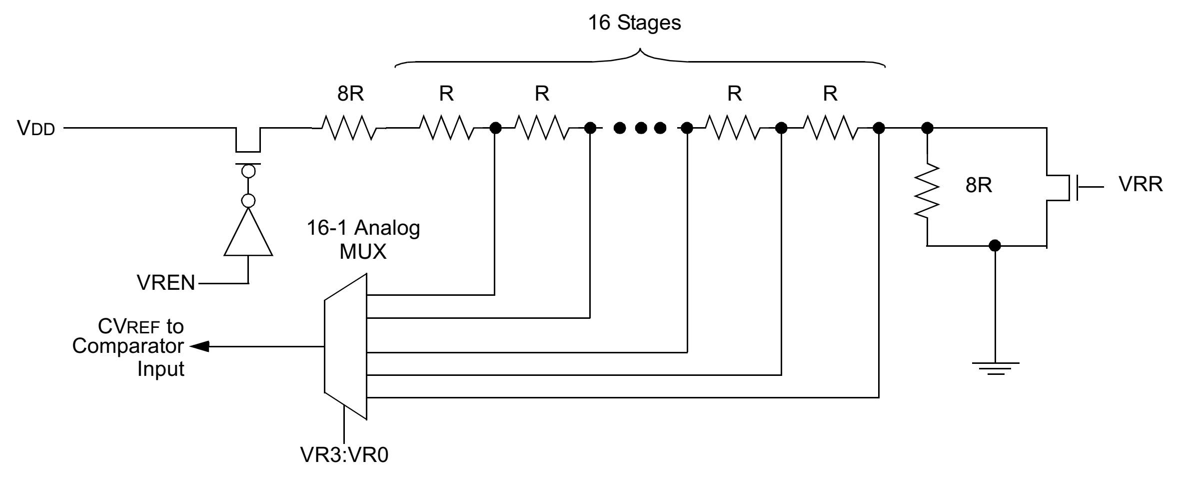
$VRR = 1$ (low range): $CV_{REF} = (VR3{:}VR0 / 24) \times V_{DD}$

$VRR = 0$ (high range): $CV_{REF} = (V_{DD} / 4) + (VR3{:}VR0 \times V_{DD} / 32)$

### 6.5.2 VOLTAGE REFERENCE ACCURACY/ERROR

The full range of $V_{SS}$ to $V_{DD}$ cannot be realized due to the construction of the module. The transistors on the top and bottom of the resistor ladder network (Figure 6-5) keep $CV_{REF}$ from approaching $V_{SS}$ or $V_{DD}$. The Voltage Reference is $V_{DD}$ derived and therefore, the $CV_{REF}$ output changes with fluctuations in $V_{DD}$. The tested absolute accuracy of the Comparator Voltage Reference can be found in **Section 12.0 "Electrical Specifications"**.

**FIGURE 6-5: COMPARATOR VOLTAGE REFERENCE BLOCK DIAGRAM**



## 6.6 Comparator Response Time

Response time is the minimum time, after selecting a new reference voltage or input source, before the comparator output is ensured to have a valid level. If the internal reference is changed, the maximum delay of the internal voltage reference must be considered when using the comparator outputs. Otherwise, the maximum delay of the comparators should be used (Table 12-7).

## 6.7 Operation During Sleep

Both the comparator and voltage reference, if enabled before entering Sleep mode, remain active during Sleep. This results in higher Sleep currents than shown in the power-down specifications. The additional current consumed by the comparator and the voltage reference is shown separately in the specifications. To minimize power consumption while in Sleep mode, turn off the comparator, CM2:CM0 = `111`, and voltage reference, VRCON<7> = `0`.

While the comparator is enabled during Sleep, an interrupt will wake-up the device. If the device wakes up from Sleep, the contents of the CMCON and VRCON registers are not affected.

## 6.8 Effects of a Reset

A device Reset forces the CMCON and VRCON registers to their Reset states. This forces the comparator module to be in the Comparator Reset mode, CM2:CM0 = `000` and the voltage reference to its off state. Thus, all potential inputs are analog inputs with the comparator and voltage reference disabled to consume the smallest current possible.

### REGISTER 6-2: VRCON — VOLTAGE REFERENCE CONTROL REGISTER (ADDRESS: 99h)

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| R/W-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| VREN | — | VRR | — | VR3 | VR2 | VR1 | VR0 |
| bit 7 | | | | | | | bit 0 |

| | |
|---|---|
| bit 7 | **VREN:** CVREF Enable bit<br>`1` = CVREF circuit powered on<br>`0` = CVREF circuit powered down, no IDD drain |
| bit 6 | **Unimplemented:** Read as '`0`' |
| bit 5 | **VRR:** CVREF Range Selection bit<br>`1` = Low range<br>`0` = High range |
| bit 4 | **Unimplemented:** Read as '`0`' |
| bit 3-0 | **VR3:VR0:** CVREF value selection bits $0 \leq VR[3:0] \leq 15$<br>When VRR = `1`: CVREF = (VR3:VR0 / 24) * VDD<br>When VRR = `0`: CVREF = VDD/4 + (VR3:VR0 / 32) * VDD |

**Legend:**

| | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' | |
| - n = Value at POR | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |

## 6.9 Comparator Interrupts

The comparator interrupt flag is set whenever there is a change in the output value of the comparator. Software will need to maintain information about the status of the output bits, as read from CMCON<6>, to determine the actual change that has occurred. The CMIF bit, PIR1<3>, is the comparator interrupt flag. This bit must be reset in software by clearing it to '`0`'. Since it is also possible to write a '`1`' to this register, a simulated interrupt may be initiated.

The CMIE bit (PIE1<3>) and the PEIE bit (INTCON<6>) must be set to enable the interrupt. In addition, the GIE bit must also be set. If any of these bits are cleared, the interrupt is not enabled, though the CMIF bit will still be set if an interrupt condition occurs.

The user, in the Interrupt Service Routine, can clear the interrupt in the following manner:

a) Any read or write of CMCON. This will end the mismatch condition.

b) Clear flag bit CMIF.

A mismatch condition will continue to set flag bit CMIF. Reading CMCON will end the mismatch condition and allow flag bit CMIF to be cleared.

| | |
|---|---|
| **Note:** | If a change in the CMCON register (COUT) should occur when a read operation is being executed (start of the Q2 cycle), then the CMIF (PIR1<3>) interrupt flag may not get set. |

**TABLE 6-2: REGISTERS ASSOCIATED WITH COMPARATOR MODULE**

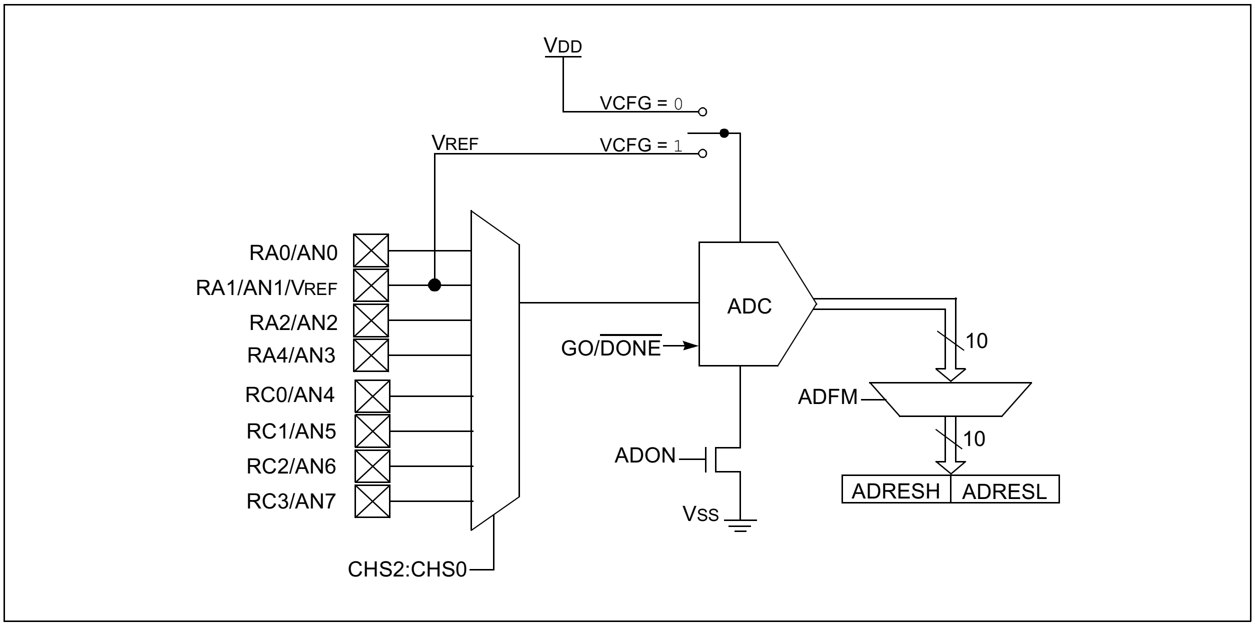
| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR, BOD | Value on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0Bh/8Bh | INTCON | GIE | PEIE | T0IE | INTE | RAIE | T0IF | INTF | RAIF | `0000 0000` | `0000 000u` |
| 0Ch | PIR1 | EEIF | ADIF | — | — | CMIF | — | — | TMR1IF | `00-- 0--0` | `00-- 0--0` |
| 19h | CMCON | — | COUT | — | CINV | CIS | CM2 | CM1 | CM0 | `-0-0 0000` | `-0-0 0000` |
| 8Ch | PIE1 | EEIE | ADIE | — | — | CMIE | — | — | TMR1IE | `00-- 0--0` | `00-- 0--0` |
| 85h | TRISA | — | — | TRISA5 | TRISA4 | TRISA3 | TRISA2 | TRISA1 | TRISA0 | `--11 1111` | `--11 1111` |
| 99h | VRCON | VREN | — | VRR | — | VR3 | VR2 | VR1 | VR0 | `0-0- 0000` | `0-0- 0000` |

**Legend:** `x` = unknown, `u` = unchanged, `-` = unimplemented, read as '`0`'. Shaded cells are not used by the comparator module.

## 7.0 ANALOG-TO-DIGITAL CONVERTER (A/D) MODULE (PIC16F676 ONLY)

The Analog-to-Digital Converter (ADC) allows conversion of an analog input signal to a 10-bit binary representation of that signal. The PIC16F676 has eight analog inputs, multiplexed into one sample and hold circuit. The output of the sample and hold is connected to the input of the converter. The converter generates a binary result via successive approximation and stores the result in a 10-bit register. The voltage reference used in the conversion is software selectable to either VDD or a voltage applied by the VREF pin. Figure 7-1 shows the block diagram of the A/D on the PIC16F676.

**FIGURE 7-1: A/D BLOCK DIAGRAM**



## 7.1 A/D Configuration and Operation

There are three registers available to control the functionality of the A/D module:

1. ADCON0 (Register 7-1)
2. ADCON1 (Register 7-2)
3. ANSEL (Register 7-3)

### 7.1.1 ANALOG PORT PINS

The ANS7:ANS0 bits (ANSEL<7:0>) and the TRISA bits control the operation of the A/D port pins. Set the corresponding TRISA bits to set the pin output driver to its high-impedance state. Likewise, set the corresponding ANS bit to disable the digital input buffer.

| **Note:** | Analog voltages on any pin that is defined as a digital input may cause the input buffer to conduct excess current. |
|---|---|

### 7.1.2 CHANNEL SELECTION

There are eight analog channels on the PIC16F676, AN0 through AN7. The CHS2:CHS0 bits (ADCON0<4:2>) control which channel is connected to the sample and hold circuit.

### 7.1.3 VOLTAGE REFERENCE

There are two options for the voltage reference to the A/D converter: either VDD is used, or an analog voltage applied to VREF is used. The VCFG bit (ADCON0<6>) controls the voltage reference selection. If VCFG is set, then the voltage on the VREF pin is the reference; otherwise, VDD is the reference.

### 7.1.4 CONVERSION CLOCK

The A/D conversion cycle requires 11 TAD. The source of the conversion clock is software selectable via the ADCS bits (ADCON1<6:4>). There are seven possible clock options:

- FOSC/2
- FOSC/4
- FOSC/8
- FOSC/16
- FOSC/32
- FOSC/64
- FRC (dedicated internal oscillator)

For correct conversion, the A/D conversion clock (1/TAD) must be selected to ensure a minimum TAD of 1.6 μs. Table 7-1 shows a few TAD calculations for selected frequencies.

**TABLE 7-1: TAD vs. DEVICE OPERATING FREQUENCIES**

| A/D Clock Source (TAD) | | Device Frequency | | | |
|---|---|---|---|---|---|
| **Operation** | **ADCS2:ADCS0** | **20 MHz** | **5 MHz** | **4 MHz** | **1.25 MHz** |
| 2 TOSC | `000` | 100 ns(2) | 400 ns(2) | 500 ns(2) | 1.6 µs |
| 4 TOSC | `100` | 200 ns(2) | 800 ns(2) | 1.0 µs(2) | 3.2 µs |
| 8 TOSC | `001` | 400 ns(2) | 1.6 µs | 2.0 µs | 6.4 µs |
| 16 TOSC | `101` | 800 ns(2) | 3.2 µs | 4.0 µs | 12.8 µs(3) |
| 32 TOSC | `010` | 1.6 µs | 6.4 µs | 8.0 µs(3) | 25.6 µs(3) |
| 64 TOSC | `110` | 3.2 µs | 12.8 µs(3) | 16.0 µs(3) | 51.2 µs(3) |
| A/D RC | `x11` | 2 - 6 µs(1,4) | 2 - 6 µs(1,4) | 2 - 6 µs(1,4) | 2 - 6 µs(1,4) |

**Legend:** Shaded cells are outside of recommended range.

**Note 1:** The A/D RC source has a typical TAD time of 4 µs for VDD > 3.0V.

**2:** These values violate the minimum required TAD time.

**3:** For faster conversion times, the selection of another clock source is recommended.

**4:** When the device frequency is greater than 1 MHz, the A/D RC clock source is only recommended if the conversion will be performed during Sleep.

### 7.1.5 STARTING A CONVERSION

The A/D conversion is initiated by setting the GO/$\overline{\text{DONE}}$ bit (ADCON0<1>). When the conversion is complete, the A/D module:

- Clears the GO/$\overline{\text{DONE}}$ bit
- Sets the ADIF flag (PIR1<6>)
- Generates an interrupt (if enabled)

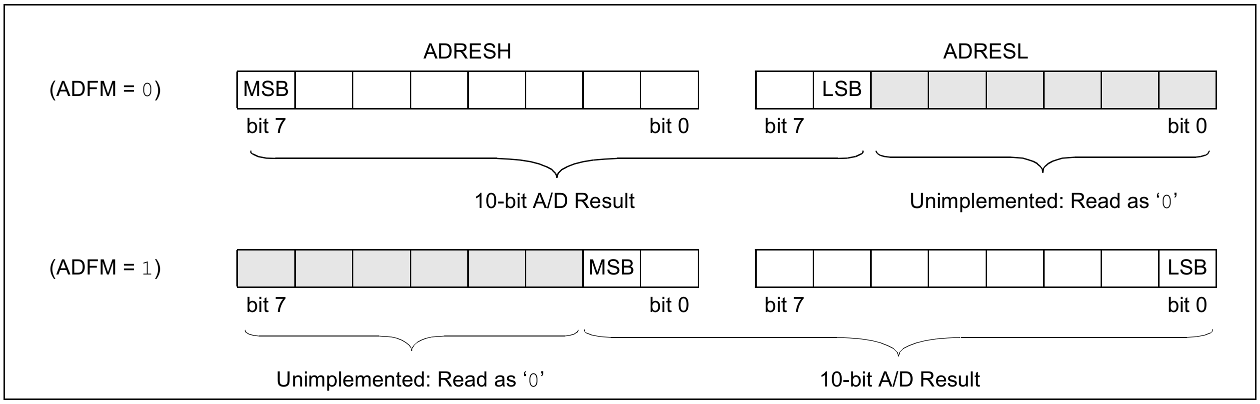
If the conversion must be aborted, the GO/$\overline{\text{DONE}}$ bit can be cleared in software. The ADRESH:ADRESL registers will not be updated with the partially complete A/D conversion sample. Instead, the ADRESH:ADRESL registers will retain the value of the previous conversion. After an aborted conversion, a 2 TAD delay is required before another acquisition can be initiated. Following the delay, an input acquisition is automatically started on the selected channel.

| | |
|---|---|
| **Note:** | The GO/$\overline{\text{DONE}}$ bit should not be set in the same instruction that turns on the A/D. |

### 7.1.6 CONVERSION OUTPUT

The A/D conversion can be supplied in two formats: left or right shifted. The ADFM bit (ADCON0<7>) controls the output format. Figure 7-2 shows the output formats.

**FIGURE 7-2: 10-BIT A/D RESULT FORMAT**



### REGISTER 7-1: ADCON0 — A/D CONTROL REGISTER (ADDRESS: 1Fh)

| R/W-0 | R/W-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| ADFM | VCFG | — | CHS2 | CHS1 | CHS0 | GO/$\overline{\text{DONE}}$ | ADON |
| bit 7 | | | | | | | bit 0 |

bit 7 **ADFM:** A/D Result Formed Select bit
`1` = Right justified
`0` = Left justified

bit 6 **VCFG:** Voltage Reference bit
`1` = VREF pin
`0` = VDD

bit 5 **Unimplemented:** Read as zero

bit 4-2 **CHS2:CHS0:** Analog Channel Select bits
`000` = Channel 00 (AN0)
`001` = Channel 01 (AN1)
`010` = Channel 02 (AN2)
`011` = Channel 03 (AN3)
`100` = Channel 04 (AN4)
`101` = Channel 05 (AN5)
`110` = Channel 06 (AN6)
`111` = Channel 07 (AN7)

bit 1 **GO/$\overline{\text{DONE}}$:** A/D Conversion Status bit
`1` = A/D conversion cycle in progress. Setting this bit starts an A/D conversion cycle.
This bit is automatically cleared by hardware when the A/D conversion has completed.
`0` = A/D conversion completed/not in progress

bit 0 **ADON:** A/D Conversion Status bit
`1` = A/D converter module is operating
`0` = A/D converter is shut-off and consumes no operating current

| **Legend:** | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as ‘0’ | |
| - n = Value at POR | ‘1’ = Bit is set | ‘0’ = Bit is cleared | x = Bit is unknown |

### REGISTER 7-2: ADCON1 — A/D CONTROL REGISTER 1 (ADRESS: 9Fh)

| U-0 | R/W-0 | R/W-0 | R/W-0 | U-0 | U-0 | U-0 | U-0 |
|---|---|---|---|---|---|---|---|
| — | ADCS2 | ADCS1 | ADCS0 | — | — | — | — |
| bit 7 | | | | | | | bit 0 |

bit 7: **Unimplemented:** Read as ‘`0`’

bit 6-4: **ADCS<2:0>:** A/D Conversion Clock Select bits
`000` = FOSC/2
`001` = FOSC/8
`010` = FOSC/32
`x11` = FRC (clock derived from a dedicated internal oscillator = 500 kHz max)
`100` = FOSC/4
`101` = FOSC/16
`110` = FOSC/64

bit 3-0: **Unimplemented:** Read as ‘`0`’

| **Legend:** | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as ‘0’ | |
| - n = Value at POR | ‘1’ = Bit is set | ‘0’ = Bit is cleared | x = Bit is unknown |

**REGISTER 7-3: ANSEL — ANALOG SELECT REGISTER (ADRESS: 91h) (PIC16F676 ONLY)**

| R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 |
|---|---|---|---|---|---|---|---|
| ANS7 | ANS6 | ANS5 | ANS4 | ANS3 | ANS2 | ANS1 | ANS0 |
| bit 7 | | | | | | | bit 0 |

bit 7-0: **ANS<7:0>**: Analog Select between analog or digital function on pins AN<7:0>, respectively.
`1` = Analog input. Pin is assigned as analog input.(1)
`0` = Digital I/O. Pin is assigned to port or special function.

**Note 1:** Setting a pin to an analog input automatically disables the digital input circuitry, weak pull-ups, and interrupt-on-change if available. The corresponding TRIS bit must be set to Input mode in order to allow external control of the voltage on the pin.

| **Legend:** | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as ‘0’ | |
| - n = Value at POR | ‘1’ = Bit is set | ‘0’ = Bit is cleared | x = Bit is unknown |

## 7.2 A/D Acquisition Requirements

For the A/D converter to meet its specified accuracy, the charge holding capacitor (CHOLD) must be allowed to fully charge to the input channel voltage level. The analog input model is shown in Figure 7-3. The source impedance (RS) and the internal sampling switch (RSS) impedance directly affect the time required to charge the capacitor CHOLD. The sampling switch (RSS) impedance varies over the device voltage (VDD), see Figure 7-3. **The maximum recommended impedance for analog sources is 10 kΩ.** As the impedance is decreased, the acquisition time may be decreased. After the analog input channel is selected (changed), this acquisition must be done before the conversion can be started.

To calculate the minimum acquisition time, Equation 7-1 may be used. This equation assumes that 1/2 LSb error is used (1024 steps for the A/D). The 1/2 LSb error is the maximum error allowed for the A/D to meet its specified resolution.

To calculate the minimum acquisition time, TACQ, see the PIC® Mid-Range Reference Manual (DS33023).

**EQUATION 7-1: ACQUISITION TIME**

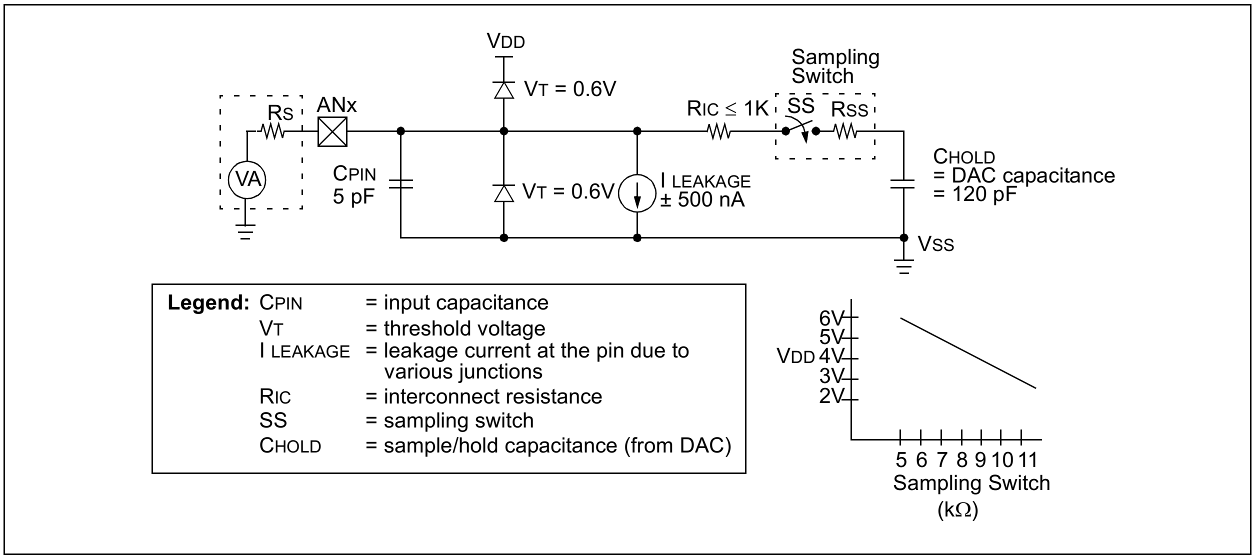
$$
\begin{aligned}
T_{ACQ} &= \text{Amplifier Settling Time} + \\
&\quad \text{Hold Capacitor Charging Time} + \\
&\quad \text{Temperature Coefficient} \\
&= T_{AMP} + T_C + T_{COFF} \\
&= 2\mu s + T_C + [(\text{Temperature} -25^\circ C)(0.05\mu s/^\circ C)] \\
T_C &= C_{HOLD}\,(R_{IC} + R_{SS} + R_S)\ \text{In}(1/2047) \\
&= -\,120pF\,(1k\Omega + 7k\Omega + 10k\Omega)\ \text{In}(0.0004885) \\
&= 16.47\mu s \\
T_{ACQ} &= 2\mu s + 16.47\mu s + [(50^\circ C -25^\circ C)(0.05\mu s/^\circ C) \\
&= 19.72\mu s
\end{aligned}
$$

**Note 1:** The reference voltage (VREF) has no effect on the equation, since it cancels itself out.

**2:** The charge holding capacitor (CHOLD) is not discharged after each conversion.

**3:** The maximum recommended impedance for analog sources is 10 kΩ. This is required to meet the pin leakage specification.

**FIGURE 7-3: ANALOG INPUT MODEL**



**Legend:**
CPIN = input capacitance
VT = threshold voltage
I LEAKAGE = leakage current at the pin due to various junctions
RIC = interconnect resistance
SS = sampling switch
CHOLD = sample/hold capacitance (from DAC)

## 7.3 A/D Operation During Sleep

The A/D converter module can operate during Sleep. This requires the A/D clock source to be set to the internal oscillator. When the RC clock source is selected, the A/D waits one instruction before starting the conversion. This allows the `SLEEP` instruction to be executed, thus eliminating much of the switching noise from the conversion. When the conversion is complete, the GO/$\overline{\mathrm{DONE}}$ bit is cleared, and the result is loaded into the ADRESH:ADRESL registers. If the A/D interrupt is enabled, the device awakens from Sleep. If the A/D interrupt is not enabled, the A/D module is turned off, although the ADON bit remains set.

When the A/D clock source is something other than RC, a `SLEEP` instruction causes the present conversion to be aborted, and the A/D module is turned off. The ADON bit remains set.

## 7.4 Effects of Reset

A device Reset forces all registers to their Reset state. Thus, the A/D module is turned off and any pending conversion is aborted. The ADRESH:ADRESL registers are unchanged.

**TABLE 7-2: SUMMARY OF A/D REGISTERS**

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR, BOD | Value on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 05h | PORTA | — | — | PORTA5 | PORTA4 | PORTA3 | PORTA2 | PORTA1 | PORTA0 | `--xx xxxx` | `--uu uuuu` |
| 07h | PORTC | — | — | PORTC5 | PORTC4 | PORTC3 | PORTC2 | PORTC1 | PORTC0 | `--xx xxxx` | `--uu uuuu` |
| 0Bh, 8Bh | INTCON | GIE | PEIE | T0IE | INTE | RAIE | T0IF | INTF | RAIF | `0000 0000` | `0000 000u` |
| 0Ch | PIR1 | EEIF | ADIF | — | — | CMIF | — | — | TMR1IF | `00-- 0--0` | `00-- 0--0` |
| 1Eh | ADRESH | Most Significant 8 bits of the Left Shifted A/D result or 2 bits of the Right Shifted Result | | | | | | | | `xxxx xxxx` | `uuuu uuuu` |
| 1Fh | ADCON0 | ADFM | VCFG | — | CHS2 | CHS1 | CHS0 | GO | ADON | `00-0 0000` | `00-0 0000` |
| 85h | TRISA | — | — | TRISA5 | TRISA4 | TRISA3 | TRISA2 | TRISA1 | TRISA0 | `--11 1111` | `--11 1111` |
| 87h | TRISC | — | — | TRISC5 | TRISC4 | TRISC3 | TRISC2 | TRISC1 | TRISC0 | `--11 1111` | `--11 1111` |
| 8Ch | PIE1 | EEIE | ADIE | — | — | CMIE | — | — | TMR1IE | `00-- 0--0` | `00-- 0--0` |
| 91h | ANSEL | ANS7 | ANS6 | ANS5 | ANS4 | ANS3 | ANS2 | ANS1 | ANS0 | `1111 1111` | `1111 1111` |
| 9Eh | ADRESL | Least Significant 2 bits of the Left Shifted A/D Result or 8 bits of the Right Shifted Result | | | | | | | | `xxxx xxxx` | `uuuu uuuu` |
| 9Fh | ADCON1 | — | ADCS2 | ADCS1 | ADCS0 | — | — | — | — | `-000 ----` | `-000 ----` |

**Legend:** `x` = unknown, `u` = unchanged, `-` = unimplemented read as '`0`'. Shaded cells are not used for A/D converter module.

## 8.0 DATA EEPROM MEMORY

The EEPROM data memory is readable and writable during normal operation (full VDD range). This memory is not directly mapped in the register file space. Instead, it is indirectly addressed through the Special Function Registers. There are four SFRs used to read and write this memory:

- EECON1
- EECON2 (not a physically implemented register)
- EEDATA
- EEADR

EEDATA holds the 8-bit data for read/write, and EEADR holds the address of the EEPROM location being accessed. PIC16F630/676 devices have 128 bytes of data EEPROM with an address range from 0h to 7Fh.

The EEPROM data memory allows byte read and write. A byte write automatically erases the location and writes the new data (erase before write). The EEPROM data memory is rated for high erase/write cycles. The write time is controlled by an on-chip timer. The write time will vary with voltage and temperature as well as from chip to chip. Please refer to AC Specifications for exact limits.

When the data memory is code-protected, the CPU may continue to read and write the data EEPROM memory. The device programmer can no longer access this memory.

Additional information on the data EEPROM is available in the PIC® Mid-Range Reference Manual, (DS33023).

**REGISTER 8-1: EEDAT — EEPROM DATA REGISTER (ADDRESS: 9Ah)**

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| EEDAT7 | EEDAT6 | EEDAT5 | EEDAT4 | EEDAT3 | EEDAT2 | EEDAT1 | EEDAT0 |
| bit 7 | | | | | | | bit 0 |

bit 7-0 **EEDATn**: Byte value to write to or read from data EEPROM

| **Legend:** | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as ‘0’ | |
| - n = Value at POR | ’1’ = Bit is set | ’0’ = Bit is cleared | x = Bit is unknown |

**REGISTER 8-2: EEADR — EEPROM ADDRESS REGISTER (ADDRESS: 9Bh)**

| U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | EADR6 | EADR5 | EADR4 | EADR3 | EADR2 | EADR1 | EADR0 |
| bit 7 | | | | | | | bit 0 |

bit 7 **Unimplemented**: Should be set to ‘0’

bit 6-0 **EEADR**: Specifies one of 128 locations for EEPROM Read/Write Operation

| **Legend:** | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as ‘0’ | |
| - n = Value at POR | ’1’ = Bit is set | ’0’ = Bit is cleared | x = Bit is unknown |

## 8.1 EEADR

The EEADR register can address up to a maximum of 128 bytes of data EEPROM. Only seven of the eight bits in the register (EEADR<6:0>) are required. The MSb (bit 7) is ignored.

The upper bit should always be '`0`' to remain upward compatible with devices that have more data EEPROM memory.

## 8.2 EECON1 AND EECON2 REGISTERS

EECON1 is the control register with four low order bits physically implemented. The upper four bits are non-implemented and read as '`0`'s.

Control bits RD and WR initiate read and write, respectively. These bits cannot be cleared, only set, in software. They are cleared in hardware at completion of the read or write operation. The inability to clear the WR bit in software prevents the accidental, premature termination of a write operation.

The WREN bit, when set, will allow a write operation. On power-up, the WREN bit is clear. The WRERR bit is set when a write operation is interrupted by a $\overline{\text{MCLR}}$ Reset, or a WDT Time-out Reset during normal operation. In these situations, following Reset, the user can check the WRERR bit, clear it, and rewrite the location. The data and address will be cleared, therefore, the EEDATA and EEADR registers will need to be re-initialized.

The Interrupt flag bit EEIF in the PIR1 register is set when the write is complete. This bit must be cleared in software.

EECON2 is not a physical register. Reading EECON2 will read all '`0`'s. The EECON2 register is used exclusively in the data EEPROM write sequence.

### REGISTER 8-3: EECON1 — EEPROM CONTROL REGISTER (ADDRESS: 9Ch)

| U-0 | U-0 | U-0 | U-0 | R/W-x | R/W-0 | R/S-0 | R/S-0 |
|---|---|---|---|---|---|---|---|
| — | — | — | — | WRERR | WREN | WR | RD |
| bit 7 | | | | | | | bit 0 |

bit 7-4 **Unimplemented:** Read as '`0`'

bit 3 **WRERR:** EEPROM Error Flag bit

`1` = A write operation is prematurely terminated (any $\overline{\text{MCLR}}$ Reset, any WDT Reset during normal operation or BOD detect)
`0` = The write operation completed

bit 2 **WREN:** EEPROM Write Enable bit

`1` = Allows write cycles
`0` = Inhibits write to the data EEPROM

bit 1 **WR**: Write Control bit

`1` = Initiates a write cycle (The bit is cleared by hardware once write is complete. The WR bit can only be set, not cleared, in software.)
`0` = Write cycle to the data EEPROM is complete

bit 0 **RD**: Read Control bit

`1` = Initiates an EEPROM read (Read takes one cycle. RD is cleared in hardware. The RD bit can only be set, not cleared, in software.)
`0` = Does not initiate an EEPROM read

| | | | |
|---|---|---|---|
| **Legend:** | | | |
| S = Bit can only be set | | | |
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' | |
| - n = Value at POR | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |

## 8.3 READING THE EEPROM DATA MEMORY

To read a data memory location, the user must write the address to the EEADR register and then set control bit RD (EECON1<0>), as shown in Example 8-1. The data is available in the very next cycle in the EEDATA register. Therefore, it can be read in the next instruction. EEDATA holds this value until another read, or until it is written to by the user (during a write operation).

**EXAMPLE 8-1: DATA EEPROM READ**

```
BSF    STATUS,RP0     ;Bank 1
MOVLW  CONFIG_ADDR    ;
MOVWF  EEADR          ;Address to read
BSF    EECON1,RD      ;EE Read
MOVF   EEDATA,W       ;Move data to W
```

## 8.4 WRITING TO THE EEPROM DATA MEMORY

To write an EEPROM data location, the user must first write the address to the EEADR register and the data to the EEDATA register. Then the user must follow a specific sequence to initiate the write for each byte, as shown in Example 8-2.

**EXAMPLE 8-2: DATA EEPROM WRITE**

Required Sequence

```
BSF    STATUS,RP0     ;Bank 1
BSF    EECON1,WREN    ;Enable write
BCF    INTCON,GIE     ;Disable INTs
MOVLW  55h            ;Unlock write
MOVWF  EECON2         ;
MOVLW  AAh            ;
MOVWF  EECON2         ;
BSF    EECON1,WR      ;Start the write
BSF    INTCON,GIE     ;Enable INTS
```

The write will not initiate if the above sequence is not exactly followed (write 55h to EECON2, write AAh to EECON2, then set WR bit) for each byte. We strongly recommend that interrupts be disabled during this code segment. A cycle count is executed during the required sequence. Any number that is not equal to the required cycles to execute the required sequence will prevent the data from being written into the EEPROM.

Additionally, the WREN bit in EECON1 must be set to enable write. This mechanism prevents accidental writes to data EEPROM due to errant (unexpected) code execution (i.e., lost programs). The user should keep the WREN bit clear at all times, except when updating EEPROM. The WREN bit is not cleared by hardware.

After a write sequence has been initiated, clearing the WREN bit will not affect this write cycle. The WR bit will be inhibited from being set unless the WREN bit is set.

At the completion of the write cycle, the WR bit is cleared in hardware and the EE Write Complete Interrupt Flag bit (EEIF) is set. The user can either enable this interrupt or poll this bit. The EEIF bit (PIR<7>) register must be cleared by software.

## 8.5 WRITE VERIFY

Depending on the application, good programming practice may dictate that the value written to the data EEPROM should be verified (see Example 8-3) to the desired value to be written.

**EXAMPLE 8-3: WRITE VERIFY**

```
BCF    STATUS,RP0    ;Bank 0
:                    ;Any code
BSF    STATUS,RP0    ;Bank 1 READ
MOVF   EEDATA,W      ;EEDATA not changed
                     ;from previous write
BSF    EECON1,RD     ;YES, Read the
                     ;value written
XORWF  EEDATA,W
BTFSS  STATUS,Z      ;Is data the same
GOTO   WRITE_ERR     ;No, handle error
:                    ;Yes, continue
```

### 8.5.1 USING THE DATA EEPROM

The data EEPROM is a high-endurance, byte addressable array that has been optimized for the storage of frequently changing information (e.g., program variables or other data that are updated often). Frequently changing values will typically be updated more often than specifications D120 or D120A. If this is not the case, an array refresh must be performed. For this reason, variables that change infrequently (such as constants, IDs, calibration, etc.) should be stored in Flash program memory.

## 8.6 PROTECTION AGAINST SPURIOUS WRITE

There are conditions when the user may not want to write to the data EEPROM memory. To protect against spurious EEPROM writes, various mechanisms have been built in. On power-up, WREN is cleared. Also, the Power-up Timer (72 ms duration) prevents EEPROM write.

The write initiate sequence and the WREN bit together help prevent an accidental write during:

- brown-out
- power glitch
- software malfunction

## 8.7 DATA EEPROM OPERATION DURING CODE-PROTECT

Data memory can be code-protected by programming the CPD bit to ‘`0`’.

When the data memory is code-protected, the CPU is able to read and write data to the data EEPROM. It is recommended to code-protect the program memory when code protecting data memory. This prevents anyone from programming zeroes over the existing code (which will execute as `NOP`s) to reach an added routine, programmed in unused program memory, which outputs the contents of data memory. Programming unused locations to ‘`0`’ will also help prevent data memory code protection from becoming breached.

**TABLE 8-1: REGISTERS/BITS ASSOCIATED WITH DATA EEPROM**

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR, BOD | Value on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0Ch | PIR1 | EEIF | ADIF | — | — | CMIF | — | — | TMR1IF | `00-- 0--0` | `00-- 0--0` |
| 9Ah | EEDATA | EEPROM Data Register | | | | | | | | `0000 0000` | `0000 0000` |
| 9Bh | EEADR | — | EEPROM Address Register | | | | | | | `-000 0000` | `-000 0000` |
| 9Ch | EECON1 | — | — | — | — | WRERR | WREN | WR | RD | `---- x000` | `---- q000` |
| 9Dh | EECON2[(1)] | EEPROM Control Register 2 | | | | | | | | `---- ----` | `---- ----` |

**Legend:** `x` = unknown, `u` = unchanged, – = unimplemented read as ‘`0`’, `q` = value depends upon condition. Shaded cells are not used by the data EEPROM module.

**Note 1:** EECON2 is not a physical register.

## 9.0 SPECIAL FEATURES OF THE CPU

Certain special circuits that deal with the needs of real time applications are what sets a microcontroller apart from other processors. The PIC16F630/676 family has a host of such features intended to:

- maximize system reliability
- minimize cost through elimination of external components
- provide power-saving operating modes and offer code protection

These features are:

- Oscillator selection
- Reset
  - Power-on Reset (POR)
  - Power-up Timer (PWRT)
  - Oscillator Start-up Timer (OST)
  - Brown-out Detect (BOD)
- Interrupts
- Watchdog Timer (WDT)
- Sleep
- Code protection
- ID Locations
- In-Circuit Serial Programming™

The PIC16F630/676 has a Watchdog Timer that is controlled by Configuration bits. It runs off its own RC oscillator for added reliability. There are two timers that offer necessary delays on power-up. One is the Oscillator Start-up Timer (OST), intended to keep the chip in Reset until the crystal oscillator is stable. The other is the Power-up Timer (PWRT), which provides a fixed delay of 72 ms (nominal) on power-up only, designed to keep the part in Reset while the power supply stabilizes. There is also circuitry to reset the device if a brown-out occurs, which can provide at least a 72 ms Reset. With these three functions on-chip, most applications need no external Reset circuitry.

The Sleep mode is designed to offer a very low current Power-down mode. The user can wake-up from Sleep through:

- External Reset
- Watchdog Timer wake-up
- An interrupt

Several oscillator options are also made available to allow the part to fit the application. The INTOSC option saves system cost while the LP crystal option saves power. A set of Configuration bits are used to select various options (see Register 9-1).

## 9.1 Configuration Bits

The Configuration bits can be programmed (read as ‘`0`’), or left unprogrammed (read as ‘`1`’) to select various device configurations, as shown in Register 9-1. These bits are mapped in program memory location 2007h.

> **Note:** Address 2007h is beyond the user program memory space. It belongs to the special configuration memory space (2000h-3FFFh), which can be accessed only during programming. See PIC16F630/676 Programming Specification for more information.

### REGISTER 9-1: CONFIG — CONFIGURATION WORD (ADDRESS: 2007h)

| R/P-1 | R/P-1 | U-0 | U-0 | U-0 | R/P-1 | R/P-1 | R/P-1 | R/P-1 | R/P-1 | R/P-1 | R/P-1 | R/P-1 | R/P-1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| BG1 | BG0 | — | — | — | $\overline{\text{CPD}}$ | $\overline{\text{CP}}$ | BODEN | MCLRE | $\overline{\text{PWRTE}}$ | WDTE | F0SC2 | F0SC1 | F0SC0 |

bit 13 bit 0

bit 13-12 **BG1:BG0:** Bandgap Calibration bits for BOD and POR voltage[(1)]
`00` = Lowest bandgap voltage
`11` = Highest bandgap voltage

bit 11-9 **Unimplemented**: Read as ‘0’

bit 8 **$\overline{\text{CPD}}$:** Data Code Protection bit[(2)]
`1` = Data memory code protection is disabled
`0` = Data memory code protection is enabled

bit 7 **$\overline{\text{CP}}$**: Code Protection bit[(3)]
`1` = Program Memory code protection is disabled
`0` = Program Memory code protection is enabled

bit 6 **BODEN**: Brown-out Detect Enable bit[(4)]
`1` = BOD enabled
`0` = BOD disabled

bit 5 **MCLRE**: RA3/$\overline{\text{MCLR}}$ pin function select bit[(5)]
`1` = RA3/$\overline{\text{MCLR}}$ pin function is $\overline{\text{MCLR}}$
`0` = RA3/$\overline{\text{MCLR}}$ pin function is digital I/O, $\overline{\text{MCLR}}$ internally tied to $V_{DD}$

bit 4 **$\overline{\text{PWRTE}}$**: Power-up Timer Enable bit
`1` = PWRT disabled
`0` = PWRT enabled

bit 3 **WDTE**: Watchdog Timer Enable bit
`1` = WDT enabled
`0` = WDT disabled

bit 2-0 **FOSC2:FOSC0**: Oscillator Selection bits
`111` = RC oscillator: CLKOUT function on RA4/OSC2/CLKOUT pin, RC on RA5/OSC1/CLKIN
`110` = RC oscillator: I/O function on RA4/OSC2/CLKOUT pin, RC on RA5/OSC1/CLKIN
`101` = INTOSC oscillator: CLKOUT function on RA4/OSC2/CLKOUT pin, I/O function on RA5/OSC1/CLKIN
`100` = INTOSC oscillator: I/O function on RA4/OSC2/CLKOUT pin, I/O function on RA5/OSC1/CLKIN
`011` = EC: I/O function on RA4/OSC2/CLKOUT pin, CLKIN on RA5/OSC1/CLKIN
`010` = HS oscillator: High speed crystal/resonator on RA4/OSC2/CLKOUT and RA5/OSC1/CLKIN
`001` = XT oscillator: Crystal/resonator on RA4/OSC2/CLKOUT and RA5/OSC1/CLKIN
`000` = LP oscillator: Low power crystal on RA4/OSC2/CLKOUT and RA5/OSC1/CLKIN

**Note 1:** The Bandgap Calibration bits are factory programmed and must be read and saved prior to erasing the device as specified in the PIC16F630/676 Programming Specification. These bits are reflected in an export of the Configuration Word. Microchip Development Tools maintain all calibration bits to factory settings.
**2:** The entire data EEPROM will be erased when the code protection is turned off.
**3:** The entire program memory will be erased, including OSCCAL value, when the code protection is turned off.
**4:** Enabling Brown-out Detect does not automatically enable Power-up Timer.
**5:** When $\overline{\text{MCLR}}$ is asserted in INTOSC or RC mode, the internal clock oscillator is disabled.

**Legend:**
P = Programmed using ICSP™
R = Readable bit W = Writable bit U = Unimplemented bit, read as ‘0’
-n = Value at POR 1 = bit is set 0 = bit is cleared x = bit is unknown

## 9.2 Oscillator Configurations

### 9.2.1 OSCILLATOR TYPES

The PIC16F630/676 can be operated in eight different Oscillator Option modes. The user can program three Configuration bits (FOSC2 through FOSC0) to select one of these eight modes:

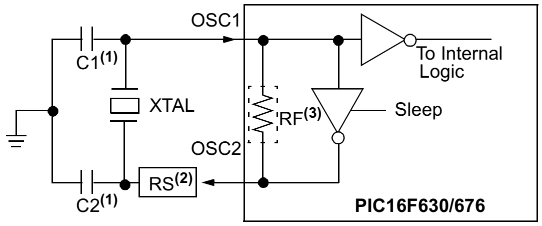
| | |
|---|---|
| • LP | Low-Power Crystal |
| • XT | Crystal/Resonator |
| • HS | High Speed Crystal/Resonator |
| • RC | External Resistor/Capacitor (2 modes) |
| • INTOSC | Internal Oscillator (2 modes) |
| • EC | External Clock In |

**Note:** Additional information on oscillator configurations is available in the PIC® Mid-Range Reference Manual, (DS33023).

### 9.2.2 CRYSTAL OSCILLATOR / CERAMIC RESONATORS

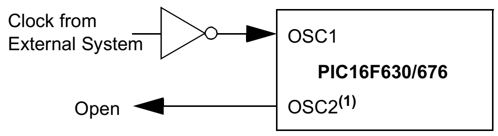
In XT, LP or HS modes a crystal or ceramic resonator is connected to the OSC1 and OSC2 pins to establish oscillation (see Figure 9-1). The PIC16F630/676 oscillator design requires the use of a parallel cut crystal. Use of a series cut crystal may yield a frequency outside of the crystal manufacturers specifications. When in XT, LP or HS modes, the device can have an external clock source to drive the OSC1 pin (see Figure 9-2).

**FIGURE 9-1: CRYSTAL OPERATION (OR CERAMIC RESONATOR) (HS, XT OR LP OSC CONFIGURATION)**



**Note 1:** See Table 9-1 and Table 9-2 for recommended values of C1 and C2.
**2:** A series resistor may be required for AT strip cut crystals.
**3:** RF varies with the Oscillator mode selected (Approx. value = 10 MΩ).

**FIGURE 9-2: EXTERNAL CLOCK INPUT OPERATION (HS, XT, EC, OR LP OSC CONFIGURATION)**



**Note 1:** Functions as RA4 in EC Osc mode.

**TABLE 9-1: CAPACITOR SELECTION FOR CERAMIC RESONATORS**

| Ranges Characterized: | | | |
|---|---|---|---|
| **Mode** | **Freq** | **OSC1(C1)** | **OSC2(C2)** |
| XT | 455 kHz<br>2.0 MHz<br>4.0 MHz | 68-100 pF<br>15-68 pF<br>15-68 pF | 68-100 pF<br>15-68 pF<br>15-68 pF |
| HS | 8.0 MHz<br>16.0 MHz | 10-68 pF<br>10-22 pF | 10-68 pF<br>10-22 pF |

**Note 1:** Higher capacitance increases the stability of the oscillator but also increases the start-up time. These values are for design guidance only. Since each resonator has its own characteristics, the user should consult the resonator manufacturer for appropriate values of external components.

**TABLE 9-2: CAPACITOR SELECTION FOR CRYSTAL OSCILLATOR**

| Mode | Freq | OSC1(C1) | OSC2(C2) |
|---|---|---|---|
| LP | 32 kHz | 68-100 pF | 68-100 pF |
| XT | 100 kHz<br>2 MHz<br>4 MHz | 68-150 pF<br>15-30 pF<br>15-30 pF | 150-200 pF<br>15-30 pF<br>15-30 pF |
| HS | 8 MHz<br>10 MHz<br>20 MHz | 15-30 pF<br>15-30 pF<br>15-30 pF | 15-30 pF<br>15-30 pF<br>15-30 pF |

**Note 1:** Higher capacitance increases the stability of the oscillator but also increases the start-up time. These values are for design guidance only. Rs may be required in HS mode as well as XT mode to avoid overdriving crystals with low drive level specification. Since each crystal has its own characteristics, the user should consult the crystal manufacturer for appropriate values of external components.

### 9.2.3 EXTERNAL CLOCK IN

For applications where a clock is already available elsewhere, users may directly drive the PIC16F630/676 provided that this external clock source meets the AC/DC timing requirements listed in **Section 12.0 "Electrical Specifications"**. Figure 9-2 shows how an external clock circuit should be configured.

### 9.2.4 RC OSCILLATOR

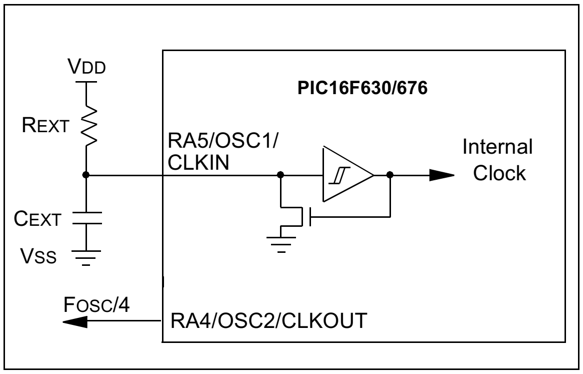
For applications where precise timing is not a requirement, the RC oscillator option is available. The operation and functionality of the RC oscillator is dependent upon a number of variables. The RC oscillator frequency is a function of:

- Supply voltage
- Resistor (REXT) and capacitor (CEXT) values
- Operating temperature

The oscillator frequency will vary from unit to unit due to normal process parameter variation. The difference in lead frame capacitance between package types will also affect the oscillation frequency, especially for low CEXT values. The user also needs to account for the tolerance of the external R and C components. Figure 9-3 shows how the R/C combination is connected.

Two options are available for this Oscillator mode which allow RA4 to be used as a general purpose I/O or to output FOSC/4.

**FIGURE 9-3: RC OSCILLATOR MODE**



### 9.2.5 INTERNAL 4 MHz OSCILLATOR

When calibrated, the internal oscillator provides a fixed 4 MHz (nominal) system clock. See Electrical Specifications, **Section 12.0 "Electrical Specifications"**, for information on variation over voltage and temperature.

Two options are available for this Oscillator mode which allow RA4 to be used as a general purpose I/O or to output FOSC/4.

#### 9.2.5.1 Calibrating the Internal Oscillator

A calibration instruction is programmed into the last location of program memory. This instruction is a `RETLW XX`, where the literal is the calibration value. The literal is placed in the OSCCAL register to set the calibration of the internal oscillator. Example 9-1 demonstrates how to calibrate the internal oscillator. For best operation, decouple (with capacitance) VDD and VSS as close to the device as possible.

**Note:** Erasing the device will also erase the pre-programmed internal calibration value for the internal oscillator. The calibration value must be saved prior to erasing part as specified in the PIC16F630/676 Programming specification. Microchip Development Tools maintain all calibration bits to factory settings.

**EXAMPLE 9-1: CALIBRATING THE INTERNAL OSCILLATOR**

```
BSF     STATUS, RP0    ;Bank 1
CALL    3FFh           ;Get the cal value
MOVWF   OSCCAL         ;Calibrate
BCF     STATUS, RP0    ;Bank 0
```

### 9.2.6 CLKOUT

The PIC16F630/676 devices can be configured to provide a clock out signal in the INTOSC and RC Oscillator modes. When configured, the oscillator frequency divided by four (FOSC/4) is output on the RA4/OSC2/CLKOUT pin. FOSC/4 can be used for test purposes or to synchronize other logic.

## 9.3 Reset

The PIC16F630/676 differentiates between various kinds of Reset:

a) Power-on Reset (POR)
b) WDT Reset during normal operation
c) WDT Reset during Sleep
d) $\overline{\text{MCLR}}$ Reset during normal operation
e) $\overline{\text{MCLR}}$ Reset during Sleep
f) Brown-out Detect (BOD)

Some registers are not affected in any Reset condition; their status is unknown on POR and unchanged in any other Reset. Most other registers are reset to a "Reset state" on:

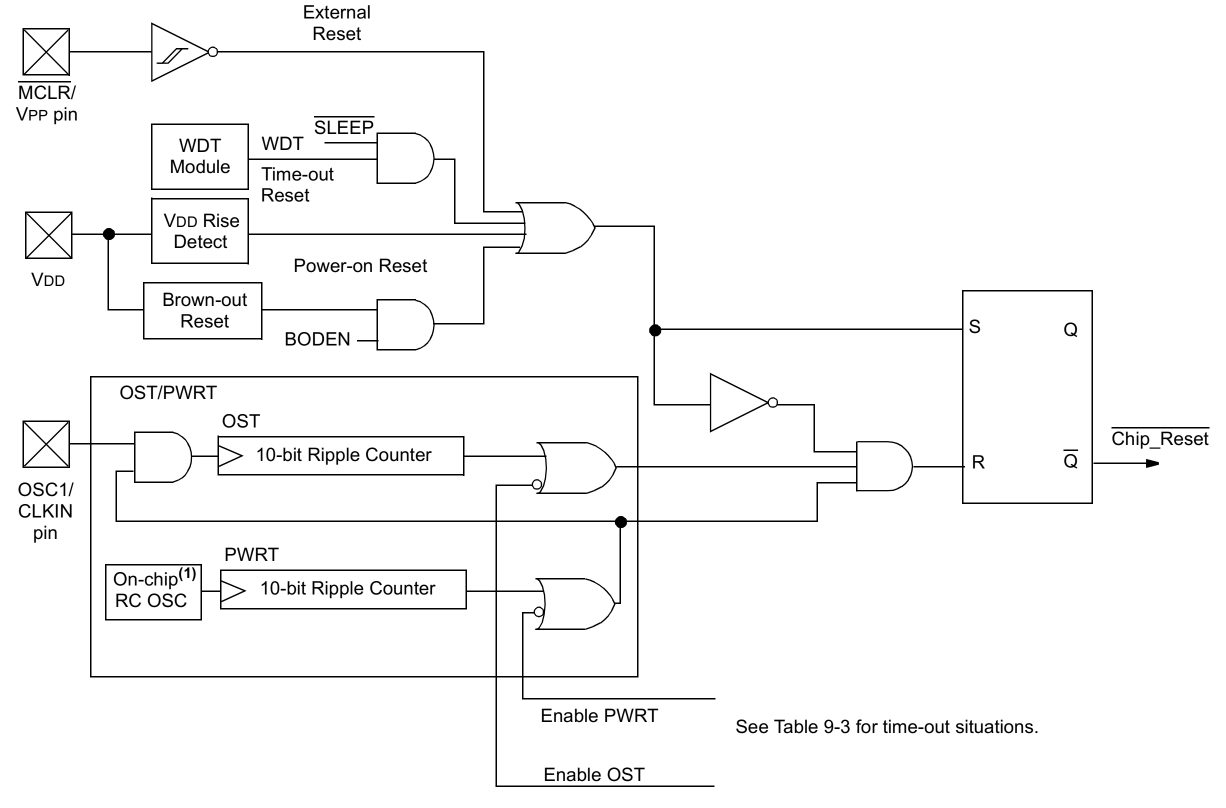
- Power-on Reset
- $\overline{\text{MCLR}}$ Reset
- WDT Reset
- WDT Reset during Sleep
- Brown-out Detect (BOD)

They are not affected by a WDT wake-up, since this is viewed as the resumption of normal operation. $\overline{\text{TO}}$ and $\overline{\text{PD}}$ bits are set or cleared differently in different Reset situations as indicated in Table 9-4. These bits are used in software to determine the nature of the Reset. See Table 9-7 for a full description of Reset states of all registers.

A simplified block diagram of the On-Chip Reset Circuit is shown in Figure 9-4.

The $\overline{\text{MCLR}}$ Reset path has a noise filter to detect and ignore small pulses. See Table 12-4 in Electrical Specifications Section for pulse-width specification.

**FIGURE 9-4: SIMPLIFIED BLOCK DIAGRAM OF ON-CHIP RESET CIRCUIT**



See Table 9-3 for time-out situations.

**Note 1:** This is a separate oscillator from the INTOSC/EC oscillator.

### 9.3.1 MCLR

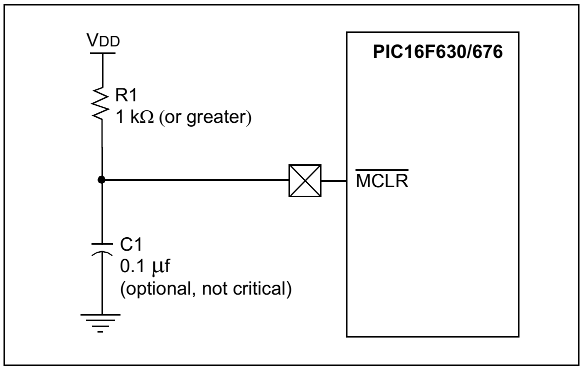
PIC16F630/676 devices have a noise filter in the MCLR Reset path. The filter will detect and ignore small pulses.

It should be noted that a WDT Reset does not drive MCLR pin low.

The behavior of the ESD protection on the MCLR pin has been altered from previous devices of this family. Voltages applied to the pin that exceed its specification can result in both MCLR Resets and excessive current beyond the device specification during the ESD event. For this reason, Microchip recommends that the MCLR pin no longer be tied directly to VDD. The use of an RC network, as shown in Figure 9-5, is suggested.

An internal MCLR option is enabled by setting the MCLRE bit in the Configuration Word. When enabled, MCLR is internally tied to VDD. No internal pull-up option is available for the MCLR pin.

**FIGURE 9-5: RECOMMENDED MCLR CIRCUIT**



### 9.3.2 POWER-ON RESET (POR)

The on-chip POR circuit holds the chip in Reset until VDD has reached a high enough level for proper operation. To take advantage of the POR, simply tie the MCLR pin through a resistor to VDD. This will eliminate external RC components usually needed to create Power-on Reset. A maximum rise time for VDD is required. See **Section 12.0 “Electrical Specifications”** for details. If the BOD is enabled, the maximum rise time specification does not apply. The BOD circuitry will keep the device in Reset until VDD reaches VBOD (see **Section 9.3.5 “Brown-out Detect (BOD)”**).

| **Note:** | The POR circuit does not produce an internal Reset when VDD declines. |
|---|---|

When the device starts normal operation (exits the Reset condition), device operating parameters (i.e., voltage, frequency, temperature, etc.) must be met to ensure operation. If these conditions are not met, the device must be held in Reset until the operating conditions are met.

For additional information, refer to Application Note *AN607 “Power-up Trouble Shooting.”*

### 9.3.3 POWER-UP TIMER (PWRT)

The Power-up Timer provides a fixed 72 ms (nominal) time-out on power-up only, from POR or Brown-out Detect. The Power-up Timer operates on an internal RC oscillator. The chip is kept in Reset as long as PWRT is active. The PWRT delay allows the VDD to rise to an acceptable level. A Configuration bit, PWRTE can disable (if set) or enable (if cleared or programmed) the Power-up Timer. The Power-up Timer should always be enabled when Brown-out Detect is enabled.

The Power-up Time delay will vary from chip to chip and due to:

- VDD variation
- Temperature variation
- Process variation.

See DC parameters for details (**Section 12.0 “Electrical Specifications”**).

### 9.3.4 OSCILLATOR START-UP TIMER (OST)

The Oscillator Start-up Timer (OST) provides a 1024 oscillator cycle (from OSC1 input) delay after the PWRT delay is over. This ensures that the crystal oscillator or resonator has started and stabilized.

The OST time-out is invoked only for XT, LP and HS modes and only on Power-on Reset or wake-up from Sleep.

### 9.3.5 BROWN-OUT DETECT (BOD)

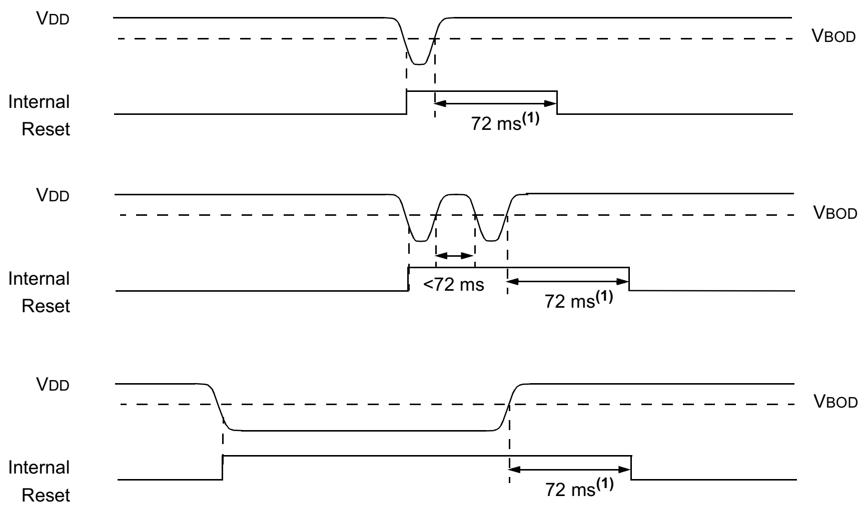
The PIC16F630/676 members have on-chip Brown-out Detect circuitry. A Configuration bit, BODEN, can disable (if clear/programmed) or enable (if set) the Brown-out Detect circuitry. If VDD falls below VBOD for greater than parameter (TBOD) in Table 12-4 (see **Section 12.0 "Electrical Specifications"**), the Brown-out situation will reset the device. This will occur regardless of VDD slew-rate. A Reset is not guaranteed to occur if VDD falls below VBOD for less than parameter (TBOD).

On any Reset (Power-on, Brown-out Detect, Watchdog, etc.), the chip will remain in Reset until VDD rises above BVDD (see Figure 9-6). The Power-up Timer will now be invoked, if enabled, and will keep the chip in Reset an additional 72 ms.

**Note:** A Brown-out Detect does not enable the Power-up Timer if the $\overline{\text{PWRTE}}$ bit in the Configuration Word is set.

If VDD drops below BVDD while the Power-up Timer is running, the chip will go back into a Brown-out Detect and the Power-up Timer will be re-initialized. Once VDD rises above BVDD, the Power-up Timer will execute a 72 ms Reset.

**FIGURE 9-6: BROWN-OUT SITUATIONS**



**Note 1:** 72 ms delay only if $\overline{\text{PWRTE}}$ bit is programmed to '`0`'.

### 9.3.6 TIME-OUT SEQUENCE

On power-up, the time-out sequence is as follows: first, PWRT time-out is invoked after POR has expired. Then, OST is activated. The total time-out will vary based on oscillator configuration and $\overline{\text{PWRTE}}$ bit status. For example, in EC mode with $\overline{\text{PWRTE}}$ bit erased (PWRT disabled), there will be no time-out at all. Figure 9-7, Figure 9-8 and Figure 9-9 depict time-out sequences.

Since the time-outs occur from the POR pulse, if $\overline{\text{MCLR}}$ is kept low long enough, the time-outs will expire. Then bringing $\overline{\text{MCLR}}$ high will begin execution immediately (see Figure 9-8). This is useful for testing purposes or to synchronize more than one PIC16F630/676 device operating in parallel.

Table 9-6 shows the Reset conditions for some special registers, while Table 9-7 shows the Reset conditions for all the registers.

### 9.3.7 POWER CONTROL (PCON) STATUS REGISTER

The power CONTROL/STATUS register, PCON (address 8Eh) has two bits.

Bit 0 is $\overline{\text{BOD}}$ (Brown-out). $\overline{\text{BOD}}$ is unknown on Power-on Reset. It must then be set by the user and checked on subsequent Resets to see if $\overline{\text{BOD}}$ = `0`, indicating that a brown-out has occurred. The $\overline{\text{BOD}}$ Status bit is a "don't care" and is not necessarily predictable if the brown-out circuit is disabled (by setting BODEN bit = `0` in the Configuration Word).

Bit 1 is $\overline{\text{POR}}$ (Power-on Reset). It is a '`0`' on Power-on Reset and unaffected otherwise. The user must write a '`1`' to this bit following a Power-on Reset. On a subsequent Reset, if $\overline{\text{POR}}$ is '`0`', it will indicate that a Power-on Reset must have occurred (i.e., VDD may have gone too low).

### TABLE 9-3: TIME-OUT IN VARIOUS SITUATIONS

| Oscillator Configuration | Power-up | | Brown-out Detect | | Wake-up from Sleep |
|---|---|---|---|---|---|
| | $\overline{\text{PWRTE}}$ = 0 | $\overline{\text{PWRTE}}$ = 1 | $\overline{\text{PWRTE}}$ = 0 | $\overline{\text{PWRTE}}$ = 1 | |
| XT, HS, LP | TPWRT +<br>1024•TOSC | 1024•TOSC | TPWRT +<br>1024•TOSC | 1024•TOSC | 1024•TOSC |
| RC, EC, INTOSC | TPWRT | — | TPWRT | — | — |

### TABLE 9-4: STATUS/PCON BITS AND THEIR SIGNIFICANCE

| $\overline{\text{POR}}$ | $\overline{\text{BOD}}$ | $\overline{\text{TO}}$ | $\overline{\text{PD}}$ | |
|---|---|---|---|---|
| `0` | `u` | `1` | `1` | Power-on Reset |
| `1` | `0` | `1` | `1` | Brown-out Detect |
| `u` | `u` | `0` | `u` | WDT Reset |
| `u` | `u` | `0` | `0` | WDT Wake-up |
| `u` | `u` | `u` | `u` | $\overline{\text{MCLR}}$ Reset during normal operation |
| `u` | `u` | `1` | `0` | $\overline{\text{MCLR}}$ Reset during Sleep |

Legend: u = unchanged, x = unknown

### TABLE 9-5: SUMMARY OF REGISTERS ASSOCIATED WITH BROWN-OUT

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR, BOD | Value on all other Resets[(1)] |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 03h | STATUS | IRP | RP1 | RPO | $\overline{\text{TO}}$ | $\overline{\text{PD}}$ | Z | DC | C | `0001 1xxx` | `000q quuu` |
| 8Eh | PCON | — | — | — | — | — | — | $\overline{\text{POR}}$ | $\overline{\text{BOD}}$ | `---- --0x` | `---- --uq` |

**Legend:** `u` = unchanged, `x` = unknown, `-` = unimplemented bit, reads as '`0`', `q` = value depends on condition.

**Note 1:** Other (non Power-up) Resets include $\overline{\text{MCLR}}$ Reset, Brown-out Detect and Watchdog Timer Reset during normal operation.

### TABLE 9-6: INITIALIZATION CONDITION FOR SPECIAL REGISTERS

| Condition | Program Counter | STATUS Register | PCON Register |
|---|---|---|---|
| Power-on Reset | 000h | `0001 1xxx` | `---- --0x` |
| $\overline{\text{MCLR}}$ Reset during normal operation | 000h | `000u uuuu` | `---- --uu` |
| $\overline{\text{MCLR}}$ Reset during Sleep | 000h | `0001 0uuu` | `---- --uu` |
| WDT Reset | 000h | `0000 uuuu` | `---- --uu` |
| WDT Wake-up | PC + 1 | `uuu0 0uuu` | `---- --uu` |
| Brown-out Detect | 000h | `0001 1uuu` | `---- --10` |
| Interrupt Wake-up from Sleep | PC + 1[(1)] | `uuu1 0uuu` | `---- --uu` |

**Legend:** `u` = unchanged, `x` = unknown, `-` = unimplemented bit, reads as '`0`'.

**Note 1:** When the wake-up is due to an interrupt and global enable bit GIE is set, the PC is loaded with the interrupt vector (0004h) after execution of PC + 1.

**TABLE 9-7: INITIALIZATION CONDITION FOR REGISTERS**

| Register | Address | Power-on Reset | • $\overline{\text{MCLR}}$ Reset<br>• WDT Reset<br>• Brown-out Detect(1) | • Wake-up from Sleep through interrupt<br>• Wake-up from Sleep through WDT time-out |
|---|---|---|---|---|
| W | — | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| INDF | 00h/80h | — | — | — |
| TMR0 | 01h | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| PCL | 02h/82h | 0000 0000 | 0000 0000 | PC + 1(3) |
| STATUS | 03h/83h | 0001 1xxx | 000q quuu(4) | uuuq quuu(4) |
| FSR | 04h/84h | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| PORTA | 05h | --xx xxxx | --uu uuuu | --uu uuuu |
| PORTC | 07h | --xx xxxx | --uu uuuu | --uu uuuu |
| PCLATH | 0Ah/8Ah | ---0 0000 | ---0 0000 | ---u uuuu |
| INTCON | 0Bh/8Bh | 0000 0000 | 0000 000u | uuuu uuqq(2) |
| PIR1 | 0Ch | 00-- 0--0 | 00-- 0--0 | qq-- q--q(2,5) |
| T1CON | 10h | -000 0000 | -uuu uuuu | -uuu uuuu |
| CMCON | 19h | -0-0 0000 | -0-0 0000 | -u-u uuuu |
| ADRESH | 1Eh | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| ADCON0 | 1Fh | 00-0 0000 | 00-0 0000 | uu-u uuuu |
| OPTION_REG | 81h | 1111 1111 | 1111 1111 | uuuu uuuu |
| TRISA | 85h | --11 1111 | --11 1111 | --uu uuuu |
| TRISC | 87h | --11 1111 | --11 1111 | --uu uuuu |
| PIE1 | 8Ch | 00-- 0--0 | 00-- 0--0 | uu-- u--u |
| PCON | 8Eh | ---- --0x | ---- --uu(1,6) | ---- --uu |
| OSCCAL | 90h | 1000 00-- | 1000 00-- | uuuu uu-- |
| ANSEL | 91h | 1111 1111 | 1111 1111 | uuuu uuuu |
| WPUA | 95h | --11 -111 | --11 -111 | uuuu uuuu |
| IOCA | 96h | --00 0000 | --00 0000 | --uu uuuu |
| VRCON | 99h | 0-0- 0000 | 0-0- 0000 | u-u- uuuu |
| EEDATA | 9Ah | 0000 0000 | 0000 0000 | uuuu uuuu |
| EEADR | 9Bh | -000 0000 | -000 0000 | -uuu uuuu |
| EECON1 | 9Ch | ---- x000 | ---- q000 | ---- uuuu |
| EECON2 | 9Dh | ---- ---- | ---- ---- | ---- ---- |
| ADRESL | 9Eh | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| ADCON1 | 9Fh | -000 ---- | -000 ---- | -uuu ---- |

**Legend:** u = unchanged, x = unknown, - = unimplemented bit, reads as ‘0’, q = value depends on condition.

**Note 1:** If VDD goes too low, Power-on Reset will be activated and registers will be affected differently.

**2:** One or more bits in INTCON and/or PIR1 will be affected (to cause wake-up).

**3:** When the wake-up is due to an interrupt and the GIE bit is set, the PC is loaded with the interrupt vector (0004h).

**4:** See Table 9-6 for Reset value for specific condition.

**5:** If wake-up was due to data EEPROM write completing, bit 7 = 1; A/D conversion completing, bit 6 = 1; Comparator input changing, bit 3 = 1; or Timer1 rolling over, bit 0 = 1. All other interrupts generating a wake-up will cause these bits to = u.

**6:** If Reset was due to brown-out, then bit 0 = 0. All other Resets will cause bit 0 = u.

**FIGURE 9-7: TIME-OUT SEQUENCE ON POWER-UP ($\overline{MCLR}$ NOT TIED TO $V_{DD}$): CASE 1**

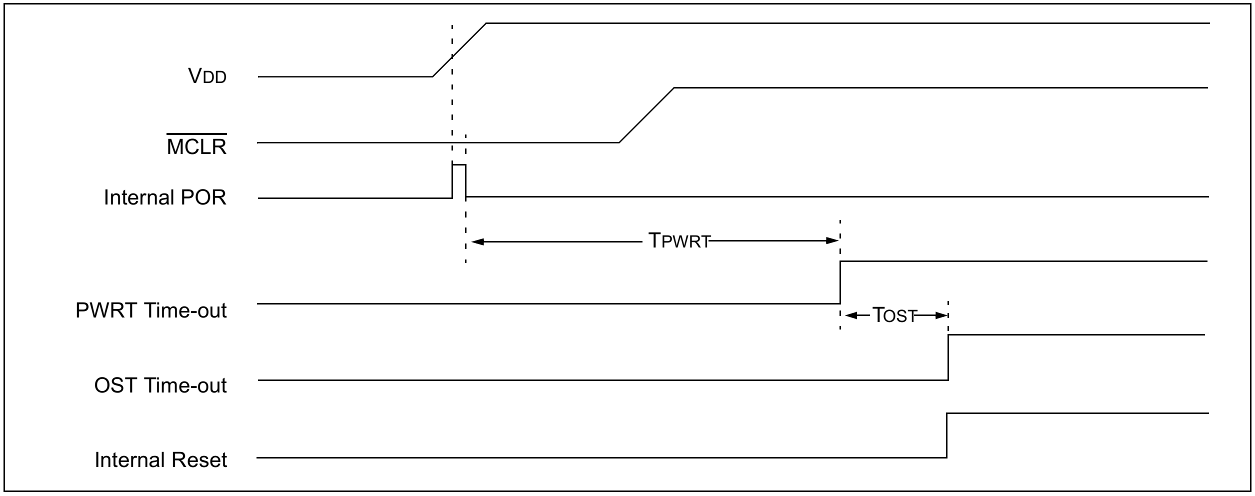


**FIGURE 9-8: TIME-OUT SEQUENCE ON POWER-UP ($\overline{MCLR}$ NOT TIED TO $V_{DD}$): CASE 2**

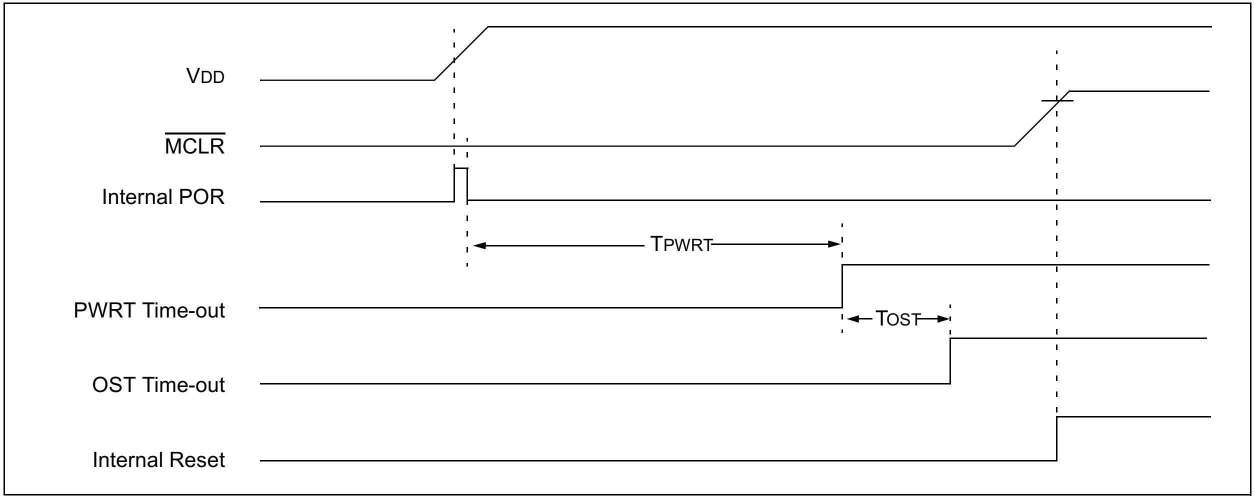
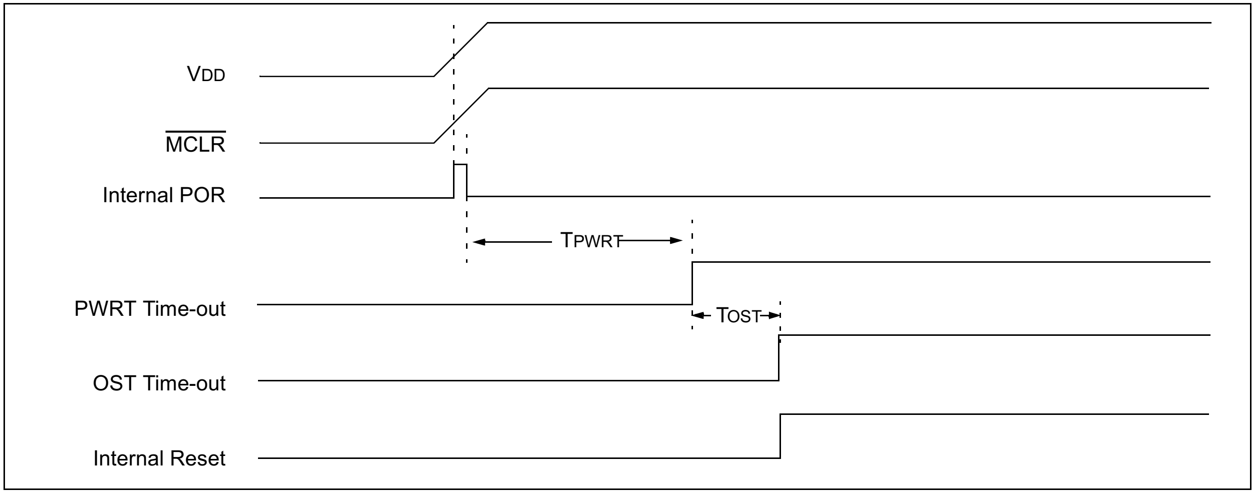


**FIGURE 9-9: TIME-OUT SEQUENCE ON POWER-UP ($\overline{MCLR}$ TIED TO $V_{DD}$)**



## 9.4 Interrupts

The PIC16F630/676 has 7 sources of interrupt:

- External Interrupt RA2/INT
- TMR0 Overflow Interrupt
- PORTA Change Interrupts
- Comparator Interrupt
- A/D Interrupt (PIC16F676 only)
- TMR1 Overflow Interrupt
- EEPROM Data Write Interrupt

The Interrupt Control register (INTCON) and Peripheral Interrupt register (PIR) record individual interrupt requests in flag bits. The INTCON register also has individual and Global Interrupt Enable bits.

A Global Interrupt Enable bit, GIE (INTCON<7>) enables (if set) all unmasked interrupts, or disables (if cleared) all interrupts. Individual interrupts can be disabled through their corresponding enable bits in INTCON register and PIE register. GIE is cleared on Reset.

The return from interrupt instruction, `RETFIE`, exits interrupt routine, as well as sets the GIE bit, which re-enables unmasked interrupts.

The following interrupt flags are contained in the INTCON register:

- INT pin interrupt
- PORTA change interrupt
- TMR0 overflow interrupt

The peripheral interrupt flags are contained in the special register PIR1. The corresponding interrupt enable bit is contained in Special Register PIE1.

The following interrupt flags are contained in the PIR register:

- EEPROM data write interrupt
- A/D interrupt
- Comparator interrupt
- Timer1 overflow interrupt

When an interrupt is serviced:

- The GIE is cleared to disable any further interrupt
- The return address is pushed onto the stack
- The PC is loaded with 0004h

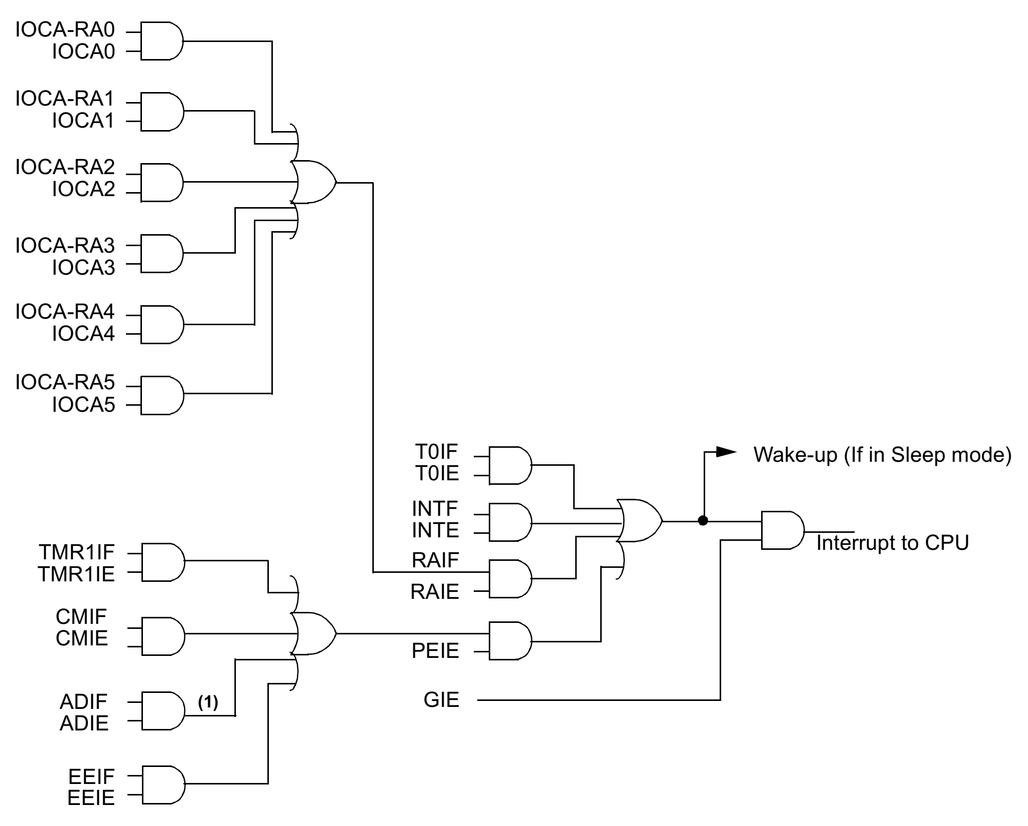
Once in the Interrupt Service Routine, the source(s) of the interrupt can be determined by polling the interrupt flag bits. The interrupt flag bit(s) must be cleared in software before re-enabling interrupts to avoid RA2/INT recursive interrupts.

For external interrupt events, such as the INT pin, or PORTA change interrupt, the interrupt latency will be three or four instruction cycles. The exact latency depends upon when the interrupt event occurs (see Figure 9-11). The latency is the same for one or two-cycle instructions. Once in the Interrupt Service Routine, the source(s) of the interrupt can be determined by polling the interrupt flag bits. The interrupt flag bit(s) must be cleared in software before re-enabling interrupts to avoid multiple interrupt requests.

**Note 1:** Individual interrupt flag bits are set, regardless of the status of their corresponding mask bit or the GIE bit.

**2:** When an instruction that clears the GIE bit is executed, any interrupts that were pending for execution in the next cycle are ignored. The interrupts which were ignored are still pending to be serviced when the GIE bit is set again.

**FIGURE 9-10: INTERRUPT LOGIC**



**Note 1:** PIC16F676 only.

### 9.4.1 RA2/INT INTERRUPT

External interrupt on RA2/INT pin is edge-triggered; either rising if INTEDG bit (OPTION<6>) is set, or falling, if INTEDG bit is clear. When a valid edge appears on the RA2/INT pin, the INTF bit (INTCON<1>) is set. This interrupt can be disabled by clearing the INTE control bit (INTCON<4>). The INTF bit must be cleared in software in the Interrupt Service Routine before re-enabling this interrupt. The RA2/INT interrupt can wake-up the processor from Sleep if the INTE bit was set prior to going into Sleep. The status of the GIE bit decides whether or not the processor branches to the interrupt vector following wake-up. See **Section 9.7 “Power-Down Mode (Sleep)”** for details on Sleep and Figure 9-13 for timing of wake-up from Sleep through RA2/INT interrupt.

**Note:** The ANSEL (91h) and CMCON (19h) registers must be initialized to configure an analog channel as a digital input. Pins configured as analog inputs will read ‘`0`’. The ANSEL register is defined for the PIC16F676.

### 9.4.2 TMR0 INTERRUPT

An overflow (FFh → 00h) in the TMR0 register will set the T0IF (INTCON<2>) bit. The interrupt can be enabled/disabled by setting/clearing T0IE (INTCON<5>) bit. For operation of the Timer0 module, see **Section 4.0 “Timer0 Module”**.

### 9.4.3 PORTA INTERRUPT

An input change on PORTA change sets the RAIF (INTCON<0>) bit. The interrupt can be enabled/disabled by setting/clearing the RAIE (INTCON<3>) bit. Plus individual pins can be configured through the IOCA register.

**Note:** If a change on the I/O pin should occur when the read operation is being executed (start of the Q2 cycle), then the RAIF interrupt flag may not get set.

### 9.4.4 COMPARATOR INTERRUPT

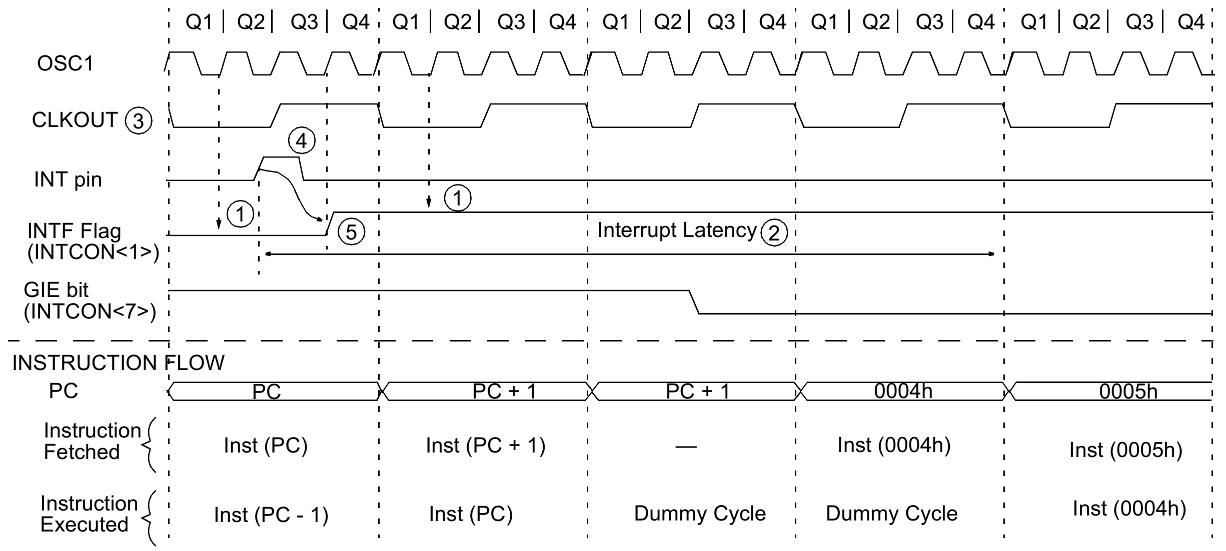
See **Section 6.9 “Comparator Interrupts”** for description of comparator interrupt.

### 9.4.5 A/D CONVERTER INTERRUPT

After a conversion is complete, the ADIF flag (PIR<6>) is set. The interrupt can be enabled/disabled by setting or clearing ADIE (PIE<6>).

See **Section 7.0 “Analog-to-Digital Converter (A/D) Module (PIC16F676 only)”** for operation of the A/D converter interrupt.

**FIGURE 9-11: INT PIN INTERRUPT TIMING**



**Note**
1: INTF flag is sampled here (every Q1).
2: Asynchronous interrupt latency = 3-4 TCY. Synchronous latency = 3 TCY, where TCY = instruction cycle time. Latency is the same whether Inst (PC) is a single cycle or a 2-cycle instruction.
3: CLKOUT is available only in RC Oscillator mode.
4: For minimum width of INT pulse, refer to AC specs.
5: INTF is enabled to be set any time during the Q4-Q1 cycles.

**TABLE 9-8: SUMMARY OF INTERRUPT REGISTERS**

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR, BOD | Value on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0Bh, 8Bh | INTCON | GIE | PEIE | T0IE | INTE | RAIE | T0IF | INTF | RAIF | `0000 0000` | `0000 000u` |
| 0Ch | PIR1 | EEIF | ADIF | — | — | CMIF | — | — | TMR1IF | `00-- 0--0` | `00-- 0--0` |
| 8Ch | PIE1 | EEIE | ADIE | — | — | CMIE | — | — | TMR1IE | `00-- 0--0` | `00-- 0--0` |

**Legend:** x = unknown, u = unchanged, - = unimplemented read as ‘0’, q = value depends upon condition.
Shaded cells are not used by the Interrupt module.

## 9.5 Context Saving During Interrupts

During an interrupt, only the return PC value is saved on the stack. Typically, users may wish to save key registers during an interrupt (e.g., W register and STATUS register). This must be implemented in software.

Example 9-2 stores and restores the STATUS and W registers. The user register, W_TEMP, must be defined in both banks and must be defined at the same offset from the bank base address (i.e., W_TEMP is defined at 0x20 in Bank 0 and it must also be defined at 0xA0 in Bank 1). The user register, STATUS_TEMP, must be defined in Bank 0. The Example 9-2:

- Stores the W register
- Stores the STATUS register in Bank 0
- Executes the ISR code
- Restores the Status (and bank select bit register)
- Restores the W register

**EXAMPLE 9-2: SAVING THE STATUS AND W REGISTERS IN RAM**

```
MOVWF  W_TEMP       ;copy W to temp register,
                     could be in either bank
SWAPF  STATUS,W     ;swap status to be saved into W
BCF    STATUS,RP0   ;change to bank 0 regardless of
                     current bank
MOVWF  STATUS_TEMP  ;save status to bank 0 register
    :
    :(ISR)
    :
SWAPF  STATUS_TEMP,W;swap STATUS_TEMP register into
                     W, sets bank to original state
MOVWF  STATUS       ;move W into STATUS register
SWAPF  W_TEMP,F     ;swap W_TEMP
SWAPF  W_TEMP,W     ;swap W_TEMP into W
```

## 9.6 Watchdog Timer (WDT)

The Watchdog Timer is a free running, on-chip RC oscillator, which requires no external components. This RC oscillator is separate from the external RC oscillator of the CLKIN pin. That means that the WDT will run, even if the clock on the OSC1 and OSC2 pins of the device has been stopped (for example, by execution of a `SLEEP` instruction). During normal operation, a WDT time-out generates a device Reset. If the device is in Sleep mode, a WDT time-out causes the device to wake-up and continue with normal operation. The WDT can be permanently disabled by programming the Configuration bit WDTE as clear (**Section 9.1 “Configuration Bits”**).

### 9.6.1 WDT PERIOD

The WDT has a nominal time-out period of 18 ms, (with no prescaler). The time-out periods vary with temperature, VDD and process variations from part to part (see DC specs). If longer time-out periods are desired, a prescaler with a division ratio of up to 1:128 can be assigned to the WDT under software control by writing to the OPTION register. Thus, time-out periods up to 2.3 seconds can be realized.

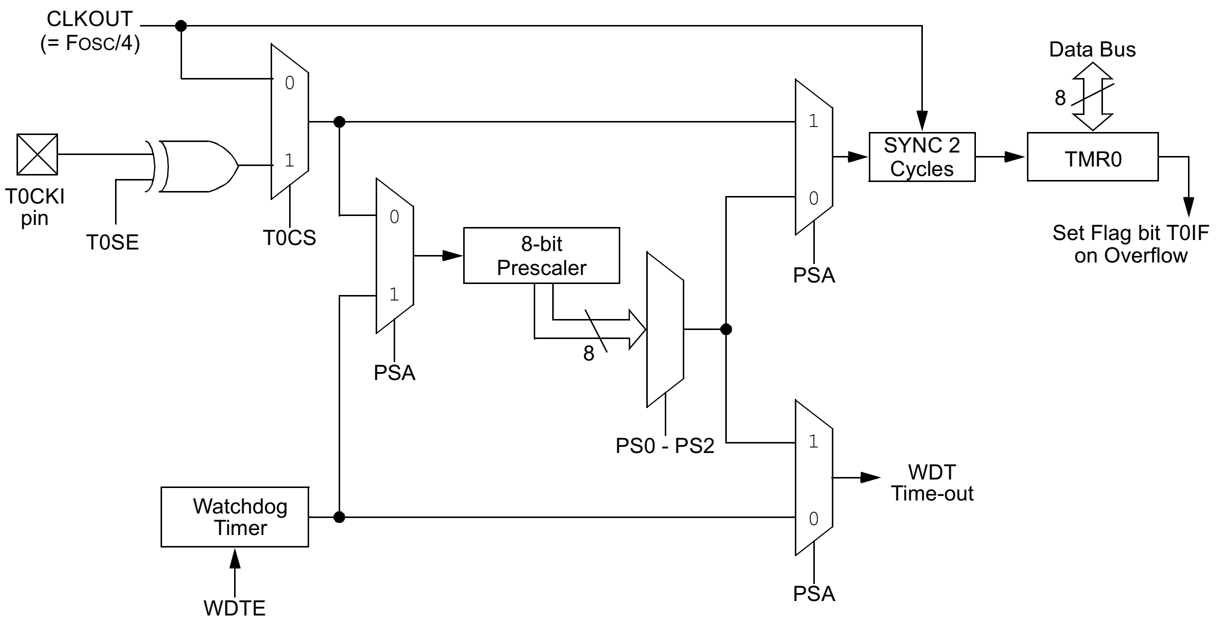
The `CLRWDT` and `SLEEP` instructions clear the WDT and the prescaler, if assigned to the WDT, and prevent it from timing out and generating a device Reset.

The $\overline{\text{TO}}$ bit in the STATUS register will be cleared upon a Watchdog Timer time-out.

### 9.6.2 WDT PROGRAMMING CONSIDERATIONS

It should also be taken in account that under worst-case conditions (i.e., VDD = Min., Temperature = Max., Max. WDT prescaler) it may take several seconds before a WDT time-out occurs.

### FIGURE 9-12: WATCHDOG TIMER BLOCK DIAGRAM



**Note 1:** T0SE, T0CS, PSA, PS0-PS2 are bits in the OPTION register.

### TABLE 9-9: SUMMARY OF WATCHDOG TIMER REGISTERS

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR, BOD | Value on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 81h | OPTION_REG | $\overline{\text{RAPU}}$ | INTEDG | T0CS | T0SE | PSA | PS2 | PS1 | PS0 | `1111 1111` | `1111 1111` |
| 2007h | Config. bits | $\overline{\text{CP}}$ | BODEN | MCLRE | $\overline{\text{PWRTE}}$ | WDTE | F0SC2 | F0SC1 | F0SC0 | `uuuu uuuu` | `uuuu uuuu` |

**Legend:** `u` = Unchanged, shaded cells are not used by the Watchdog Timer.

## 9.7 Power-Down Mode (Sleep)

The Power-down mode is entered by executing a `SLEEP` instruction.

If the Watchdog Timer is enabled:

- WDT will be cleared but keeps running
- PD bit in the STATUS register is cleared
- TO bit is set
- Oscillator driver is turned off
- I/O ports maintain the status they had before `SLEEP` was executed (driving high, low, or high-impedance).

For lowest current consumption in this mode, all I/O pins should be either at VDD, or VSS, with no external circuitry drawing current from the I/O pin and the comparators and CVREF should be disabled. I/O pins that are high-impedance inputs should be pulled high or low externally to avoid switching currents caused by floating inputs. The T0CKI input should also be at VDD or VSS for lowest current consumption. The contribution from on-chip pull-ups on PORTA should be considered.

The MCLR pin must be at a logic high level (VIHMC).

| **Note:** | It should be noted that a Reset generated by a WDT time-out does not drive MCLR pin low. |
|---|---|

### 9.7.1 WAKE-UP FROM SLEEP

The device can wake-up from Sleep through one of the following events:

1. External Reset input on MCLR pin
2. Watchdog Timer Wake-up (if WDT was enabled)
3. Interrupt from RA2/INT pin, PORTA change, or a peripheral interrupt.

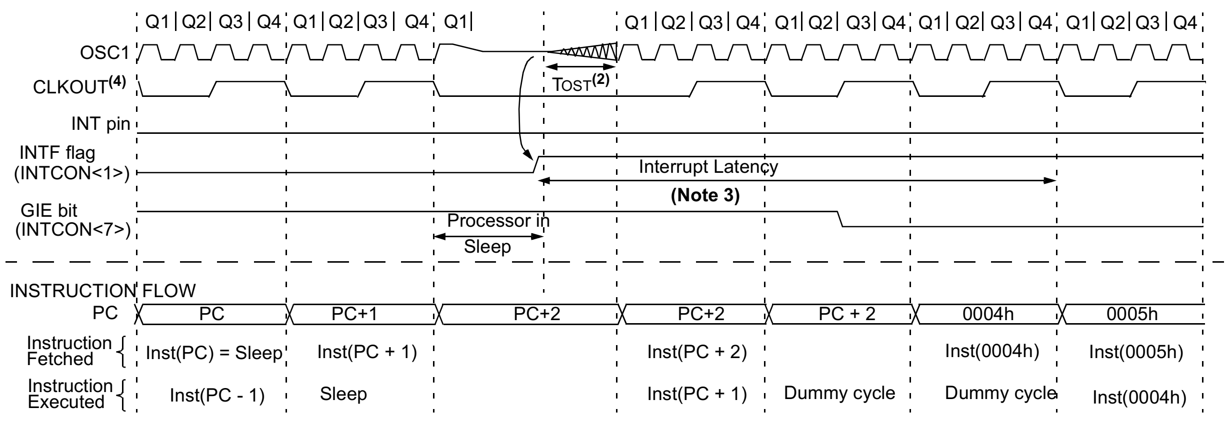
The first event will cause a device Reset. The two latter events are considered a continuation of program execution. The TO and PD bits in the STATUS register can be used to determine the cause of device Reset. The PD bit, which is set on power-up, is cleared when Sleep is invoked. TO bit is cleared if WDT Wake-up occurred.

When the `SLEEP` instruction is being executed, the next instruction (PC + 1) is pre-fetched. For the device to wake-up through an interrupt event, the corresponding interrupt enable bit must be set (enabled). Wake-up is regardless of the state of the GIE bit. If the GIE bit is clear (disabled), the device continues execution at the instruction after the `SLEEP` instruction. If the GIE bit is set (enabled), the device executes the instruction after the `SLEEP` instruction, then branches to the interrupt address (0004h). In cases where the execution of the instruction following `SLEEP` is not desirable, the user should have an `NOP` after the `SLEEP` instruction.

| **Note:** | If the global interrupts are disabled (GIE is cleared), but any interrupt source has both its interrupt enable bit and the corresponding interrupt flag bits set, the device will immediately wake-up from Sleep. The `SLEEP` instruction is completely executed. |
|---|---|

The WDT is cleared when the device wakes up from Sleep, regardless of the source of wake-up.

**FIGURE 9-13: WAKE-UP FROM SLEEP THROUGH INTERRUPT**



**Note 1:** XT, HS or LP Oscillator mode assumed.

**2:** TOST = 1024TOSC (drawing not to scale). Approximately 1 μs delay for RC Oscillator mode. See Section 12 for wake-up from Sleep delay in INTOSC mode.

**3:** GIE = `1` assumed. In this case after wake-up, the processor jumps to the interrupt routine. If GIE = `0`, execution will continue in-line.

**4:** CLKOUT is not available in XT, HS, LP or EC Osc modes, but shown here for timing reference.

## 9.8 Code Protection

If the code protection bit(s) have not been programmed, the on-chip program memory can be read out for verification purposes.

**Note:** The entire data EEPROM and Flash program memory will be erased when the code protection is turned off. The INTOSC calibration data is also erased. See PIC16F630/676 Programming Specification for more information.

## 9.9 ID Locations

Four memory locations (2000h-2003h) are designated as ID locations where the user can store checksum or other code identification numbers. These locations are not accessible during normal execution but are readable and writable during Program/Verify. Only the Least Significant 7 bits of the ID locations are used.

## 9.10 In-Circuit Serial Programming

The PIC16F630/676 microcontrollers can be serially programmed while in the end application circuit. This is simply done with two lines for clock and data, and three other lines for:

- power
- ground
- programming voltage

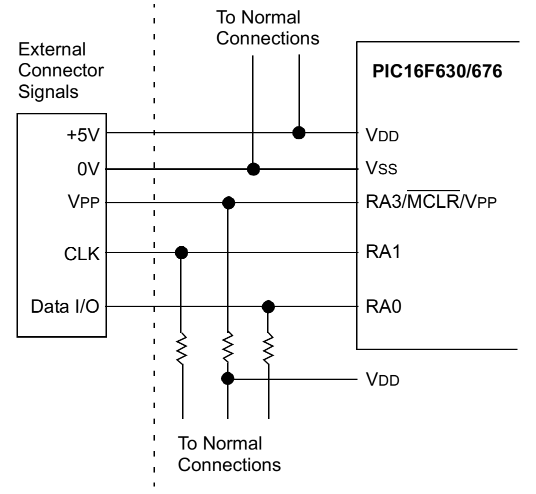
This allows customers to manufacture boards with unprogrammed devices and then program the microcontroller just before shipping the product. This also allows the most recent firmware or a custom firmware to be programmed.

The device is placed into a Program/Verify mode by holding the RA0 and RA1 pins low, while raising the $\overline{\text{MCLR}}$ (VPP) pin from VIL to VIHH (see Programming Specification). RA0 becomes the programming data and RA1 becomes the programming clock. Both RA0 and RA1 are Schmitt Trigger inputs in this mode.

After Reset, to place the device into Programming/Verify mode, the program counter (PC) is at location 00h. A 6-bit command is then supplied to the device. Depending on the command, 14 bits of program data are then supplied to or from the device, depending on whether the command was a load or a read. For complete details of serial programming, please refer to the PIC16F630/676 Programming Specification.

A typical In-Circuit Serial Programming connection is shown in Figure 9-14.

**FIGURE 9-14: TYPICAL IN-CIRCUIT SERIAL PROGRAMMING CONNECTION**



## 9.11 In-Circuit Debugger

Since in-circuit debugging requires the loss of clock, data and $\overline{\text{MCLR}}$ pins, MPLAB® ICD 2 development with an 14-pin device is not practical. A special 20-pin PIC16F676-ICD device is used with MPLAB ICD 2 to provide separate clock, data and $\overline{\text{MCLR}}$ pins and frees all normally available pins to the user.

This special ICD device is mounted on the top of the header and its signals are routed to the MPLAB ICD 2 connector. On the bottom of the header is an 14-pin socket that plugs into the user's target via the 14-pin stand-off connector.

When the $\overline{\text{ICD}}$ pin on the PIC16F676-ICD device is held low, the In-Circuit Debugger functionality is enabled. This function allows simple debugging functions when used with MPLAB ICD 2. When the microcontroller has this feature enabled, some of the resources are not available for general use. Table 9-10 shows which features are consumed by the background debugger:

**TABLE 9-10: DEBUGGER RESOURCES**

| | |
|---|---|
| I/O pins | ICDCLK, ICDDATA |
| Stack | 1 level |
| Program Memory | Address 0h must be `NOP`<br>300h-3FEh |

For more information, see 14-Pin MPLAB ICD 2 Header Information Sheet (DS51292) available on Microchip's web site (www.microchip.com).

**NOTES:**

## 10.0 INSTRUCTION SET SUMMARY

The PIC16F630/676 instruction set is highly orthogonal and is comprised of three basic categories:

- **Byte-oriented** operations
- **Bit-oriented** operations
- **Literal and control** operations

Each PIC16 instruction is a 14-bit word divided into an **opcode,** which specifies the instruction type, and one or more **operands,** which further specify the operation of the instruction. The formats for each of the categories is presented in Figure 10-1, while the various opcode fields are summarized in Table 10-1.

Table 10-2 lists the instructions recognized by the MPASM™ assembler. A complete description of each instruction is also available in the PIC® Mid-Range Reference Manual (DS33023).

For **byte-oriented** instructions, ‘f’ represents a file register designator and ‘d’ represents a destination designator. The file register designator specifies which file register is to be used by the instruction.

The destination designator specifies where the result of the operation is to be placed. If ‘d’ is zero, the result is placed in the W register. If ‘d’ is one, the result is placed in the file register specified in the instruction.

For **bit-oriented** instructions, ‘b’ represents a bit field designator, which selects the bit affected by the operation, while ‘f’ represents the address of the file in which the bit is located.

For **literal and control** operations, ‘**k**’ represents an 8-bit or 11-bit constant, or literal value

One instruction cycle consists of four oscillator periods; for an oscillator frequency of 4 MHz, this gives a normal instruction execution time of 1 μs. All instructions are executed within a single instruction cycle, unless a conditional test is true, or the program counter is changed as a result of an instruction. When this occurs, the execution takes two instruction cycles, with the second cycle executed as a `NOP`.

| **Note:** | To maintain upward compatibility with future products, do not use the `OPTION` and `TRIS` instructions. |
|---|---|

All instruction examples use the format ‘0xhh’ to represent a hexadecimal number, where ‘h’ signifies a hexadecimal digit.

## 10.1 READ-MODIFY-WRITE OPERATIONS

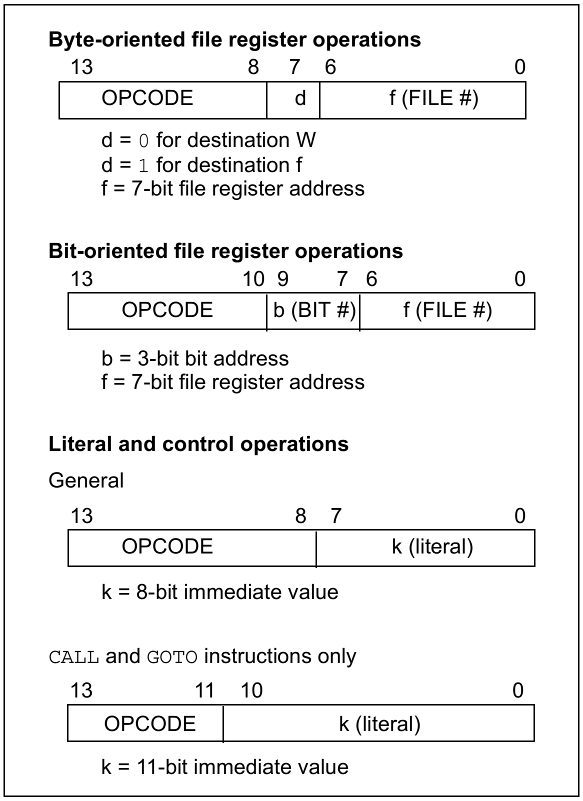
Any instruction that specifies a file register as part of the instruction performs a Read-Modify-Write (R-M-W) operation. The register is read, the data is modified, and the result is stored according to either the instruction, or the destination designator ‘d’. A read operation is performed on a register even if the instruction writes to that register.

For example, a `CLRF PORTA` instruction will read PORTA, clear all the data bits, then write the result back to PORTA. This example would have the unintended result of clearing the condition that set the RAIF flag.

### TABLE 10-1: OPCODE FIELD DESCRIPTIONS

| Field | Description |
|---|---|
| f | Register file address (0x00 to 0x7F) |
| W | Working register (accumulator) |
| b | Bit address within an 8-bit file register |
| k | Literal field, constant data or label |
| x | Don’t care location (= `0` or `1`).<br>The assembler will generate code with x = `0`.<br>It is the recommended form of use for compatibility with all Microchip software tools. |
| d | Destination select; d = `0`: store result in W,<br>d = `1`: store result in file register f.<br>Default is d = `1`. |
| PC | Program Counter |
| TO | Time-out bit |
| PD | Power-down bit |

### FIGURE 10-1: GENERAL FORMAT FOR INSTRUCTIONS



### TABLE 10-2: PIC16F630/676 INSTRUCTION SET

| Mnemonic, Operands | | Description | Cycles | 14-Bit Opcode | | | | Status Affected | Notes |
|---|---|---|---|---|---|---|---|---|---|
| | | | | MSb | | | LSb | | |
| **BYTE-ORIENTED FILE REGISTER OPERATIONS** | | | | | | | | | |
| ADDWF | f, d | Add W and f | 1 | `00` | `0111` | `dfff` | `ffff` | C,DC,Z | **1,2** |
| ANDWF | f, d | AND W with f | 1 | `00` | `0101` | `dfff` | `ffff` | Z | **1,2** |
| CLRF | f | Clear f | 1 | `00` | `0001` | `lfff` | `ffff` | Z | **2** |
| CLRW | - | Clear W | 1 | `00` | `0001` | `0xxx` | `xxxx` | Z | |
| COMF | f, d | Complement f | 1 | `00` | `1001` | `dfff` | `ffff` | Z | **1,2** |
| DECF | f, d | Decrement f | 1 | `00` | `0011` | `dfff` | `ffff` | Z | **1,2** |
| DECFSZ | f, d | Decrement f, Skip if 0 | 1(2) | `00` | `1011` | `dfff` | `ffff` | | **1,2,3** |
| INCF | f, d | Increment f | 1 | `00` | `1010` | `dfff` | `ffff` | Z | **1,2** |
| INCFSZ | f, d | Increment f, Skip if 0 | 1(2) | `00` | `1111` | `dfff` | `ffff` | | **1,2,3** |
| IORWF | f, d | Inclusive OR W with f | 1 | `00` | `0100` | `dfff` | `ffff` | Z | **1,2** |
| MOVF | f, d | Move f | 1 | `00` | `1000` | `dfff` | `ffff` | Z | **1,2** |
| MOVWF | f | Move W to f | 1 | `00` | `0000` | `lfff` | `ffff` | | |
| NOP | - | No Operation | 1 | `00` | `0000` | `0xx0` | `0000` | | |
| RLF | f, d | Rotate Left f through Carry | 1 | `00` | `1101` | `dfff` | `ffff` | C | **1,2** |
| RRF | f, d | Rotate Right f through Carry | 1 | `00` | `1100` | `dfff` | `ffff` | C | **1,2** |
| SUBWF | f, d | Subtract W from f | 1 | `00` | `0010` | `dfff` | `ffff` | C,DC,Z | **1,2** |
| SWAPF | f, d | Swap nibbles in f | 1 | `00` | `1110` | `dfff` | `ffff` | | **1,2** |
| XORWF | f, d | Exclusive OR W with f | 1 | `00` | `0110` | `dfff` | `ffff` | Z | **1,2** |
| **BIT-ORIENTED FILE REGISTER OPERATIONS** | | | | | | | | | |
| BCF | f, b | Bit Clear f | 1 | `01` | `00bb` | `bfff` | `ffff` | | **1,2** |
| BSF | f, b | Bit Set f | 1 | `01` | `01bb` | `bfff` | `ffff` | | **1,2** |
| BTFSC | f, b | Bit Test f, Skip if Clear | 1 (2) | `01` | `10bb` | `bfff` | `ffff` | | **3** |
| BTFSS | f, b | Bit Test f, Skip if Set | 1 (2) | `01` | `11bb` | `bfff` | `ffff` | | **3** |
| **LITERAL AND CONTROL OPERATIONS** | | | | | | | | | |
| ADDLW | k | Add literal and W | 1 | `11` | `111x` | `kkkk` | `kkkk` | C,DC,Z | |
| ANDLW | k | AND literal with W | 1 | `11` | `1001` | `kkkk` | `kkkk` | Z | |
| CALL | k | Call subroutine | 2 | `10` | `0kkk` | `kkkk` | `kkkk` | | |
| CLRWDT | - | Clear Watchdog Timer | 1 | `00` | `0000` | `0110` | `0100` | $\overline{TO}$,$\overline{PD}$ | |
| GOTO | k | Go to address | 2 | `10` | `1kkk` | `kkkk` | `kkkk` | | |
| IORLW | k | Inclusive OR literal with W | 1 | `11` | `1000` | `kkkk` | `kkkk` | Z | |
| MOVLW | k | Move literal to W | 1 | `11` | `00xx` | `kkkk` | `kkkk` | | |
| RETFIE | - | Return from interrupt | 2 | `00` | `0000` | `0000` | `1001` | | |
| RETLW | k | Return with literal in W | 2 | `11` | `01xx` | `kkkk` | `kkkk` | | |
| RETURN | - | Return from Subroutine | 2 | `00` | `0000` | `0000` | `1000` | | |
| SLEEP | - | Go into Standby mode | 1 | `00` | `0000` | `0110` | `0011` | $\overline{TO}$,$\overline{PD}$ | |
| SUBLW | k | Subtract W from literal | 1 | `11` | `110x` | `kkkk` | `kkkk` | C,DC,Z | |
| XORLW | k | Exclusive OR literal with W | 1 | `11` | `1010` | `kkkk` | `kkkk` | Z | |

**Note 1:** When an I/O register is modified as a function of itself (e.g., `MOVF PORTA, 1`), the value used will be that value present on the pins themselves. For example, if the data latch is '`1`' for a pin configured as input and is driven low by an external device, the data will be written back with a '`0`'.

**2:** If this instruction is executed on the TMR0 register (and, where applicable, d = `1`), the prescaler will be cleared if assigned to the Timer0 module.

**3:** If Program Counter (PC) is modified, or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a `NOP`.

| | |
|---|---|
| **Note:** | Additional information on the mid-range instruction set is available in the PIC® Mid-Range MCU Family Reference Manual (DS33023). |

## 10.2 Instruction Descriptions

| **ADDLW** | **Add Literal and W** |
|---|---|
| Syntax: | [*label*] ADDLW k |
| Operands: | $0 \le k \le 255$ |
| Operation: | $(W) + k \rightarrow (W)$ |
| Status Affected: | C, DC, Z |
| Description: | The contents of the W register are added to the eight-bit literal ‘k’ and the result is placed in the W register. |

| **ADDWF** | **Add W and f** |
|---|---|
| Syntax: | [*label*] ADDWF f,d |
| Operands: | $0 \le f \le 127$<br>$d \in [0,1]$ |
| Operation: | $(W) + (f) \rightarrow$ (destination) |
| Status Affected: | C, DC, Z |
| Description: | Add the contents of the W register with register ‘f’. If ‘d’ is `0`, the result is stored in the W register. If ‘d’ is `1`, the result is stored back in register ‘f’. |

| **ANDLW** | **AND Literal with W** |
|---|---|
| Syntax: | [*label*] ANDLW k |
| Operands: | $0 \le k \le 255$ |
| Operation: | (W) .AND. (k) $\rightarrow$ (W) |
| Status Affected: | Z |
| Description: | The contents of W register are AND’ed with the eight-bit literal ‘k’. The result is placed in the W register. |

| **ANDWF** | **AND W with f** |
|---|---|
| Syntax: | [*label*] ANDWF f,d |
| Operands: | $0 \le f \le 127$<br>$d \in [0,1]$ |
| Operation: | (W) .AND. (f) $\rightarrow$ (destination) |
| Status Affected: | Z |
| Description: | AND the W register with register ‘f’. If ‘d’ is `0`, the result is stored in the W register. If ‘d’ is `1`, the result is stored back in register ‘f’. |

| **BCF** | **Bit Clear f** |
|---|---|
| Syntax: | [*label*] BCF f,b |
| Operands: | $0 \le f \le 127$<br>$0 \le b \le 7$ |
| Operation: | $0 \rightarrow$ (f<b>) |
| Status Affected: | None |
| Description: | Bit ‘b’ in register ‘f’ is cleared. |

| **BSF** | **Bit Set f** |
|---|---|
| Syntax: | [*label*] BSF f,b |
| Operands: | $0 \le f \le 127$<br>$0 \le b \le 7$ |
| Operation: | $1 \rightarrow$ (f<b>) |
| Status Affected: | None |
| Description: | Bit ‘b’ in register ‘f’ is set. |

| **BTFSS** | **Bit Test f, Skip if Set** |
|---|---|
| Syntax: | [*label*] BTFSS f,b |
| Operands: | $0 \le f \le 127$<br>$0 \le b < 7$ |
| Operation: | skip if (f<b>) = 1 |
| Status Affected: | None |
| Description: | If bit ‘b’ in register ‘f’ is ‘`0`’, the next instruction is executed.<br>If bit ‘b’ is ‘`1`’, then the next instruction is discarded and a `NOP` is executed instead, making this a 2-cycle instruction. |

| **BTFSC** | **Bit Test, Skip if Clear** |
|---|---|
| Syntax: | [*label*] BTFSC f,b |
| Operands: | $0 \le f \le 127$<br>$0 \le b \le 7$ |
| Operation: | skip if (f<b>) = 0 |
| Status Affected: | None |
| Description: | If bit ‘b’ in register ‘f’ is ‘`1`’, the next instruction is executed.<br>If bit ‘b’, in register ‘f’, is ‘`0`’, the next instruction is discarded, and a `NOP` is executed instead, making this a 2-cycle instruction. |

| **CALL** | **Call Subroutine** |
|---|---|
| Syntax: | [ *label* ] CALL k |
| Operands: | $0 \le k \le 2047$ |
| Operation: | (PC)+ 1→ TOS,<br>k → PC<10:0>,<br>(PCLATH<4:3>) → PC<12:11> |
| Status Affected: | None |
| Description: | Call Subroutine. First, return address (PC + 1) is pushed onto the stack. The eleven-bit immediate address is loaded into PC bits <10:0>. The upper bits of the PC are loaded from PCLATH. `CALL` is a two-cycle instruction. |

| **CLRF** | **Clear f** |
|---|---|
| Syntax: | [*label*] CLRF f |
| Operands: | $0 \le f \le 127$ |
| Operation: | 00h → (f)<br>1 → Z |
| Status Affected: | Z |
| Description: | The contents of register ‘f’ are cleared and the Z bit is set. |

| **CLRW** | **Clear W** |
|---|---|
| Syntax: | [ *label* ] CLRW |
| Operands: | None |
| Operation: | 00h → (W)<br>1 → Z |
| Status Affected: | Z |
| Description: | W register is cleared. Zero bit (Z) is set. |

| **CLRWDT** | **Clear Watchdog Timer** |
|---|---|
| Syntax: | [ *label* ] CLRWDT |
| Operands: | None |
| Operation: | 00h → WDT<br>0 → WDT prescaler,<br>1 → $\overline{\text{TO}}$<br>1 → $\overline{\text{PD}}$ |
| Status Affected: | $\overline{\text{TO}}$, $\overline{\text{PD}}$ |
| Description: | `CLRWDT` instruction resets the Watchdog Timer. It also resets the prescaler of the WDT.<br>Status bits $\overline{\text{TO}}$ and $\overline{\text{PD}}$ are set. |

| **COMF** | **Complement f** |
|---|---|
| Syntax: | [ *label* ] COMF f,d |
| Operands: | $0 \le f \le 127$<br>$d \in [0,1]$ |
| Operation: | $(\bar{f}) \rightarrow$ (destination) |
| Status Affected: | Z |
| Description: | The contents of register ‘f’ are complemented. If ‘d’ is `0`, the result is stored in W. If ‘d’ is `1`, the result is stored back in register ‘f’. |

| **DECF** | **Decrement f** |
|---|---|
| Syntax: | [*label*] DECF f,d |
| Operands: | $0 \le f \le 127$<br>$d \in [0,1]$ |
| Operation: | (f) - 1 → (destination) |
| Status Affected: | Z |
| Description: | Decrement register ‘f’. If ‘d’ is `0`, the result is stored in the W register. If ‘d’ is `1`, the result is stored back in register ‘f’. |

| **DECFSZ** | **Decrement f, Skip if 0** |
|---|---|
| Syntax: | [ *label* ] DECFSZ f,d |
| Operands: | $0 \leq f \leq 127$<br>$d \in [0,1]$ |
| Operation: | (f) - 1 → (destination);<br>skip if result = 0 |
| Status Affected: | None |
| Description: | The contents of register ‘f’ are decremented. If ‘d’ is `0`, the result is placed in the W register. If ‘d’ is `1`, the result is placed back in register ‘f’.<br>If the result is `1`, the next instruction is executed. If the result is `0`, then a `NOP` is executed instead, making it a 2-cycle instruction. |

| **GOTO** | **Unconditional Branch** |
|---|---|
| Syntax: | [ *label* ] GOTO k |
| Operands: | $0 \leq k \leq 2047$ |
| Operation: | k → PC<10:0><br>PCLATH<4:3> → PC<12:11> |
| Status Affected: | None |
| Description: | `GOTO` is an unconditional branch. The eleven-bit immediate value is loaded into PC bits <10:0>. The upper bits of PC are loaded from PCLATH<4:3>. `GOTO` is a two-cycle instruction. |

| **INCF** | **Increment f** |
|---|---|
| Syntax: | [ *label* ] INCF f,d |
| Operands: | $0 \leq f \leq 127$<br>$d \in [0,1]$ |
| Operation: | (f) + 1 → (destination) |
| Status Affected: | Z |
| Description: | The contents of register ‘f’ are incremented. If ‘d’ is `0`, the result is placed in the W register. If ‘d’ is `1`, the result is placed back in register ‘f’. |

| **INCFSZ** | **Increment f, Skip if 0** |
|---|---|
| Syntax: | [ *label* ] INCFSZ f,d |
| Operands: | $0 \leq f \leq 127$<br>$d \in [0,1]$ |
| Operation: | (f) + 1 → (destination),<br>skip if result = 0 |
| Status Affected: | None |
| Description: | The contents of register ‘f’ are incremented. If ‘d’ is `0`, the result is placed in the W register. If ‘d’ is `1`, the result is placed back in register ‘f’.<br>If the result is `1`, the next instruction is executed. If the result is `0`, a `NOP` is executed instead, making it a 2-cycle instruction. |

| **IORLW** | **Inclusive OR Literal with W** |
|---|---|
| Syntax: | [ *label* ] IORLW k |
| Operands: | $0 \leq k \leq 255$ |
| Operation: | (W) .OR. k → (W) |
| Status Affected: | Z |
| Description: | The contents of the W register are OR’ed with the eight-bit literal ‘k’. The result is placed in the W register. |

| **IORWF** | **Inclusive OR W with f** |
|---|---|
| Syntax: | [ *label* ] IORWF f,d |
| Operands: | $0 \leq f \leq 127$<br>$d \in [0,1]$ |
| Operation: | (W) .OR. (f) → (destination) |
| Status Affected: | Z |
| Description: | Inclusive OR the W register with register ‘f’. If ‘d’ is `0`, the result is placed in the W register. If ‘d’ is `1`, the result is placed back in register ‘f’. |

| **MOVF** | **Move f** |
|---|---|
| Syntax: | [ *label* ] MOVF f,d |
| Operands: | $0 \leq f \leq 127$<br>$d \in [0,1]$ |
| Operation: | (f) → (destination) |
| Status Affected: | Z |
| Description: | The contents of register f are moved to a destination dependant upon the status of d. If d = `0`, destination is W register. If d = `1`, the destination is file register f itself. d = `1` is useful to test a file register, since status flag Z is affected. |

| **MOVLW** | **Move Literal to W** |
|---|---|
| Syntax: | [ *label* ] MOVLW k |
| Operands: | $0 \leq k \leq 255$ |
| Operation: | k → (W) |
| Status Affected: | None |
| Description: | The eight-bit literal ‘k’ is loaded into W register. The don’t cares will assemble as `0`’s. |

| **MOVWF** | **Move W to f** |
|---|---|
| Syntax: | [ *label* ] MOVWF f |
| Operands: | $0 \leq f \leq 127$ |
| Operation: | (W) → (f) |
| Status Affected: | None |
| Description: | Move data from W register to register ‘f’. |

| **NOP** | **No Operation** |
|---|---|
| Syntax: | [ *label* ] NOP |
| Operands: | None |
| Operation: | No operation |
| Status Affected: | None |
| Description: | No operation. |

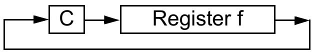
| **RETFIE** | **Return from Interrupt** |
|---|---|
| Syntax: | [ *label* ] RETFIE |
| Operands: | None |
| Operation: | TOS → PC,<br>1 → GIE |
| Status Affected: | None |

| **RETLW** | **Return with Literal in W** |
|---|---|
| Syntax: | [ *label* ] RETLW k |
| Operands: | $0 \leq k \leq 255$ |
| Operation: | k → (W);<br>TOS → PC |
| Status Affected: | None |
| Description: | The W register is loaded with the eight-bit literal ‘k’. The program counter is loaded from the top of the stack (the return address). This is a two-cycle instruction. |

| **RLF** | **Rotate Left f through Carry** |
|---|---|
| Syntax: | [ *label* ]   RLF   f,d |
| Operands: | 0 ≤ f ≤ 127<br>d ∈ [0,1] |
| Operation: | See description below |
| Status Affected: | C |
| Description: | The contents of register ‘f’ are rotated one bit to the left through the Carry Flag. If ‘d’ is `0`, the result is placed in the W register. If ‘d’ is `1`, the result is stored back in register ‘f’. |

| **RETURN** | **Return from Subroutine** |
|---|---|
| Syntax: | [ *label* ]   RETURN |
| Operands: | None |
| Operation: | TOS → PC |
| Status Affected: | None |
| Description: | Return from subroutine. The stack is POPed and the top of the stack (TOS) is loaded into the program counter. This is a two-cycle instruction. |

| **RRF** | **Rotate Right f through Carry** |
|---|---|
| Syntax: | [ *label* ]   RRF   f,d |
| Operands: | 0 ≤ f ≤ 127<br>d ∈ [0,1] |
| Operation: | See description below |
| Status Affected: | C |
| Description: | The contents of register ‘f’ are rotated one bit to the right through the Carry Flag. If ‘d’ is `0`, the result is placed in the W register. If ‘d’ is `1`, the result is placed back in register ‘f’. |



| **SLEEP** | |
|---|---|
| Syntax: | [ *label* ]   SLEEP |
| Operands: | None |
| Operation: | 00h → WDT,<br>0 → WDT prescaler,<br>1 → $\overline{TO}$,<br>0 → $\overline{PD}$ |
| Status Affected: | $\overline{TO}$, $\overline{PD}$ |
| Description: | The power-down Status bit, $\overline{PD}$ is cleared. Time-out Status bit, $\overline{TO}$ is set. Watchdog Timer and its prescaler are cleared.<br>The processor is put into Sleep mode with the oscillator stopped. |

| **SUBLW** | **Subtract W from Literal** |
|---|---|
| Syntax: | [ *label* ]   SUBLW   k |
| Operands: | 0 ≤ k ≤ 255 |
| Operation: | k - (W) → (W) |
| Status Affected: | C, DC, Z |
| Description: | The W register is subtracted (2’s complement method) from the eight-bit literal ‘k’. The result is placed in the W register. |

| **SUBWF** | **Subtract W from f** |
|---|---|
| Syntax: | [ *label* ]   SUBWF   f,d |
| Operands: | 0 ≤ f ≤ 127<br>d ∈ [0,1] |
| Operation: | (f) - (W) → (destination) |
| Status Affected: | C, DC, Z |
| Description: | Subtract (2’s complement method) W register from register ‘f’. If ‘d’ is `0`, the result is stored in the W register. If ‘d’ is `1`, the result is stored back in register ‘f’. |

| **SWAPF** | **Swap Nibbles in f** |
|---|---|
| Syntax: | [ *label* ]   SWAPF f,d |
| Operands: | $0 \leq f \leq 127$<br>$d \in [0,1]$ |
| Operation: | (f<3:0>) → (destination<7:4>),<br>(f<7:4>) → (destination<3:0>) |
| Status Affected: | None |
| Description: | The upper and lower nibbles of register ‘f’ are exchanged. If ‘d’ is `0`, the result is placed in the W register. If ‘d’ is `1`, the result is placed in register ‘f’. |

| **XORLW** | **Exclusive OR Literal with W** |
|---|---|
| Syntax: | [*label*]   XORLW   k |
| Operands: | $0 \leq k \leq 255$ |
| Operation: | (W) .XOR. k → (W) |
| Status Affected: | Z |
| Description: | The contents of the W register are XOR’ed with the eight-bit literal ‘k’. The result is placed in the W register. |

| **XORWF** | **Exclusive OR W with f** |
|---|---|
| Syntax: | [*label*]   XORWF   f,d |
| Operands: | $0 \leq f \leq 127$<br>$d \in [0,1]$ |
| Operation: | (W) .XOR. (f) → (destination) |
| Status Affected: | Z |
| Description: | Exclusive OR the contents of the W register with register ‘f’. If ‘d’ is `0`, the result is stored in the W register. If ‘d’ is `1`, the result is stored back in register ‘f’. |

## 11.0 DEVELOPMENT SUPPORT

The PIC® microcontrollers and dsPIC® digital signal controllers are supported with a full range of software and hardware development tools:

- Integrated Development Environment
  - MPLAB® IDE Software
- Compilers/Assemblers/Linkers
  - MPLAB C Compiler for Various Device Families
  - HI-TECH C for Various Device Families
  - MPASM™ Assembler
  - MPLINK™ Object Linker/ MPLIB™ Object Librarian
  - MPLAB Assembler/Linker/Librarian for Various Device Families
- Simulators
  - MPLAB SIM Software Simulator
- Emulators
  - MPLAB REAL ICE™ In-Circuit Emulator
- In-Circuit Debuggers
  - MPLAB ICD 3
  - PICkit™ 3 Debug Express
- Device Programmers
  - PICkit™ 2 Programmer
  - MPLAB PM3 Device Programmer
- Low-Cost Demonstration/Development Boards, Evaluation Kits, and Starter Kits

### 11.1 MPLAB Integrated Development Environment Software

The MPLAB IDE software brings an ease of software development previously unseen in the 8/16/32-bit microcontroller market. The MPLAB IDE is a Windows® operating system-based application that contains:

- A single graphical interface to all debugging tools
  - Simulator
  - Programmer (sold separately)
  - In-Circuit Emulator (sold separately)
  - In-Circuit Debugger (sold separately)
- A full-featured editor with color-coded context
- A multiple project manager
- Customizable data windows with direct edit of contents
- High-level source code debugging
- Mouse over variable inspection
- Drag and drop variables from source to watch windows
- Extensive on-line help
- Integration of select third party tools, such as IAR C Compilers

The MPLAB IDE allows you to:

- Edit your source files (either C or assembly)
- One-touch compile or assemble, and download to emulator and simulator tools (automatically updates all project information)
- Debug using:
  - Source files (C or assembly)
  - Mixed C and assembly
  - Machine code

MPLAB IDE supports multiple debugging tools in a single development paradigm, from the cost-effective simulators, through low-cost in-circuit debuggers, to full-featured emulators. This eliminates the learning curve when upgrading to tools with increased flexibility and power.

## 11.2 MPLAB C Compilers for Various Device Families

The MPLAB C Compiler code development systems are complete ANSI C compilers for Microchip's PIC18, PIC24 and PIC32 families of microcontrollers and the dsPIC30 and dsPIC33 families of digital signal controllers. These compilers provide powerful integration capabilities, superior code optimization and ease of use.

For easy source level debugging, the compilers provide symbol information that is optimized to the MPLAB IDE debugger.

## 11.3 HI-TECH C for Various Device Families

The HI-TECH C Compiler code development systems are complete ANSI C compilers for Microchip's PIC family of microcontrollers and the dsPIC family of digital signal controllers. These compilers provide powerful integration capabilities, omniscient code generation and ease of use.

For easy source level debugging, the compilers provide symbol information that is optimized to the MPLAB IDE debugger.

The compilers include a macro assembler, linker, preprocessor, and one-step driver, and can run on multiple platforms.

## 11.4 MPASM Assembler

The MPASM Assembler is a full-featured, universal macro assembler for PIC10/12/16/18 MCUs.

The MPASM Assembler generates relocatable object files for the MPLINK Object Linker, Intel® standard HEX files, MAP files to detail memory usage and symbol reference, absolute LST files that contain source lines and generated machine code and COFF files for debugging.

The MPASM Assembler features include:

- Integration into MPLAB IDE projects
- User-defined macros to streamline assembly code
- Conditional assembly for multi-purpose source files
- Directives that allow complete control over the assembly process

## 11.5 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK Object Linker combines relocatable objects created by the MPASM Assembler and the MPLAB C18 C Compiler. It can link relocatable objects from precompiled libraries, using directives from a linker script.

The MPLIB Object Librarian manages the creation and modification of library files of precompiled code. When a routine from a library is called from a source file, only the modules that contain that routine will be linked in with the application. This allows large libraries to be used efficiently in many different applications.

The object linker/library features include:

- Efficient linking of single libraries instead of many smaller files
- Enhanced code maintainability by grouping related modules together
- Flexible creation of libraries with easy module listing, replacement, deletion and extraction

## 11.6 MPLAB Assembler, Linker and Librarian for Various Device Families

MPLAB Assembler produces relocatable machine code from symbolic assembly language for PIC24, PIC32 and dsPIC devices. MPLAB C Compiler uses the assembler to produce its object file. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. Notable features of the assembler include:

- Support for the entire device instruction set
- Support for fixed-point and floating-point data
- Command line interface
- Rich directive set
- Flexible macro language
- MPLAB IDE compatibility

## 11.7 MPLAB SIM Software Simulator

The MPLAB SIM Software Simulator allows code development in a PC-hosted environment by simulating the PIC MCUs and dsPIC® DSCs on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a comprehensive stimulus controller. Registers can be logged to files for further run-time analysis. The trace buffer and logic analyzer display extend the power of the simulator to record and track program execution, actions on I/O, most peripherals and internal registers.

The MPLAB SIM Software Simulator fully supports symbolic debugging using the MPLAB C Compilers, and the MPASM and MPLAB Assemblers. The software simulator offers the flexibility to develop and debug code outside of the hardware laboratory environment, making it an excellent, economical software development tool.

## 11.8 MPLAB REAL ICE In-Circuit Emulator System

MPLAB REAL ICE In-Circuit Emulator System is Microchip's next generation high-speed emulator for Microchip Flash DSC and MCU devices. It debugs and programs PIC® Flash MCUs and dsPIC® Flash DSCs with the easy-to-use, powerful graphical user interface of the MPLAB Integrated Development Environment (IDE), included with each kit.

The emulator is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with either a connector compatible with in-circuit debugger systems (RJ11) or with the new high-speed, noise tolerant, Low-Voltage Differential Signal (LVDS) interconnection (CAT5).

The emulator is field upgradable through future firmware downloads in MPLAB IDE. In upcoming releases of MPLAB IDE, new devices will be supported, and new features will be added. MPLAB REAL ICE offers significant advantages over competitive emulators including low-cost, full-speed emulation, run-time variable watches, trace analysis, complex breakpoints, a ruggedized probe interface and long (up to three meters) interconnection cables.

## 11.9 MPLAB ICD 3 In-Circuit Debugger System

MPLAB ICD 3 In-Circuit Debugger System is Microchip's most cost effective high-speed hardware debugger/programmer for Microchip Flash Digital Signal Controller (DSC) and microcontroller (MCU) devices. It debugs and programs PIC® Flash microcontrollers and dsPIC® DSCs with the powerful, yet easy-to-use graphical user interface of MPLAB Integrated Development Environment (IDE).

The MPLAB ICD 3 In-Circuit Debugger probe is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with a connector compatible with the MPLAB ICD 2 or MPLAB REAL ICE systems (RJ-11). MPLAB ICD 3 supports all MPLAB ICD 2 headers.

## 11.10 PICkit 3 In-Circuit Debugger/ Programmer and PICkit 3 Debug Express

The MPLAB PICkit 3 allows debugging and programming of PIC® and dsPIC® Flash microcontrollers at a most affordable price point using the powerful graphical user interface of the MPLAB Integrated Development Environment (IDE). The MPLAB PICkit 3 is connected to the design engineer's PC using a full speed USB interface and can be connected to the target via an Microchip debug (RJ-11) connector (compatible with MPLAB ICD 3 and MPLAB REAL ICE). The connector uses two device I/O pins and the reset line to implement in-circuit debugging and In-Circuit Serial Programming™.

The PICkit 3 Debug Express include the PICkit 3, demo board and microcontroller, hookup cables and CDROM with user's guide, lessons, tutorial, compiler and MPLAB IDE software.

## 11.11 PICkit 2 Development Programmer/Debugger and PICkit 2 Debug Express

The PICkit™ 2 Development Programmer/Debugger is a low-cost development tool with an easy to use interface for programming and debugging Microchip's Flash families of microcontrollers. The full featured Windows® programming interface supports baseline (PIC10F, PIC12F5xx, PIC16F5xx), midrange (PIC12F6xx, PIC16F), PIC18F, PIC24, dsPIC30, dsPIC33, and PIC32 families of 8-bit, 16-bit, and 32-bit microcontrollers, and many Microchip Serial EEPROM products. With Microchip's powerful MPLAB Integrated Development Environment (IDE) the PICkit™ 2 enables in-circuit debugging on most PIC® microcontrollers. In-Circuit-Debugging runs, halts and single steps the program while the PIC microcontroller is embedded in the application. When halted at a breakpoint, the file registers can be examined and modified.

The PICkit 2 Debug Express include the PICkit 2, demo board and microcontroller, hookup cables and CDROM with user's guide, lessons, tutorial, compiler and MPLAB IDE software.

## 11.12 MPLAB PM3 Device Programmer

The MPLAB PM3 Device Programmer is a universal, CE compliant device programmer with programmable voltage verification at VDDMIN and VDDMAX for maximum reliability. It features a large LCD display (128 x 64) for menus and error messages and a modular, detachable socket assembly to support various package types. The ICSP™ cable assembly is included as a standard item. In Stand-Alone mode, the MPLAB PM3 Device Programmer can read, verify and program PIC devices without a PC connection. It can also set code protection in this mode. The MPLAB PM3 connects to the host PC via an RS-232 or USB cable. The MPLAB PM3 has high-speed communications and optimized algorithms for quick programming of large memory devices and incorporates an MMC card for file storage and data applications.

## 11.13 Demonstration/Development Boards, Evaluation Kits, and Starter Kits

A wide variety of demonstration, development and evaluation boards for various PIC MCUs and dsPIC DSCs allows quick application development on fully functional systems. Most boards include prototyping areas for adding custom circuitry and provide application firmware and source code for examination and modification.

The boards support a variety of features, including LEDs, temperature sensors, switches, speakers, RS-232 interfaces, LCD displays, potentiometers and additional EEPROM memory.

The demonstration and development boards can be used in teaching environments, for prototyping custom circuits and for learning about various microcontroller applications.

In addition to the PICDEM™ and dsPICDEM™ demonstration/development board series of circuits, Microchip has a line of evaluation kits and demonstration software for analog filter design, KEELOQ® security ICs, CAN, IrDA®, PowerSmart battery management, SEEVAL® evaluation system, Sigma-Delta ADC, flow rate sensing, plus many more.

Also available are starter kits that contain everything needed to experience the specified device. This usually includes a single application and debug capability, all on one board.

Check the Microchip web page (www.microchip.com) for the complete list of demonstration, development and evaluation kits.

## 12.0 ELECTRICAL SPECIFICATIONS

**Absolute Maximum Ratings†**

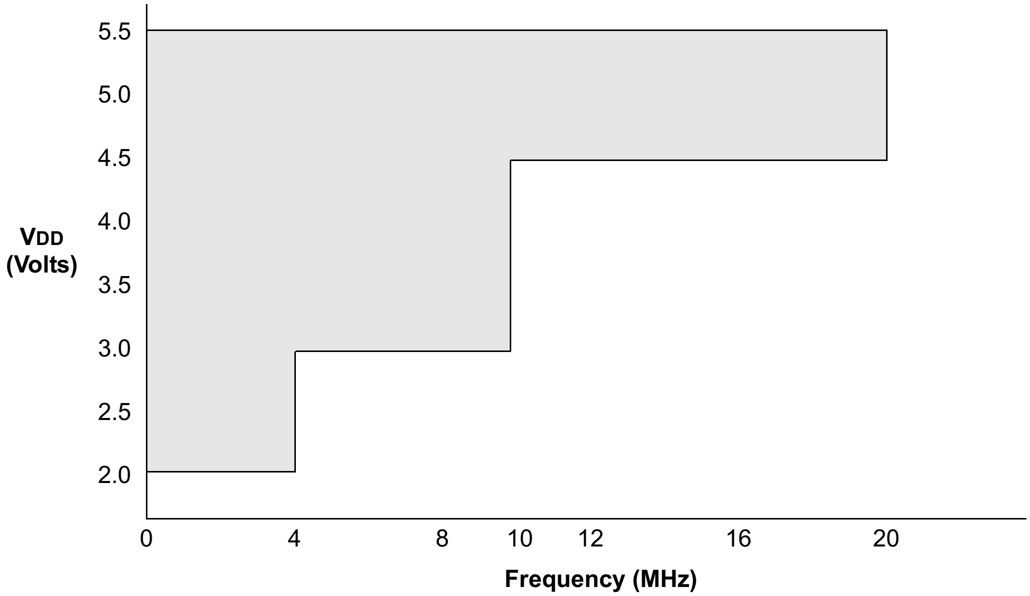
| | |
|---|---|
| Ambient temperature under bias | -40 to +125°C |
| Storage temperature | -65°C to +150°C |
| Voltage on $V_{DD}$ with respect to $V_{SS}$ | -0.3 to +6.5V |
| Voltage on $\overline{MCLR}$ with respect to Vss | -0.3 to +13.5V |
| Voltage on all other pins with respect to $V_{SS}$ | -0.3V to ($V_{DD}$ + 0.3V) |
| Total power dissipation(1) | 800 mW |
| Maximum current out of $V_{SS}$ pin | 300 mA |
| Maximum current into $V_{DD}$ pin | 250 mA |
| Input clamp current, $I_{IK}$ ($V_I < 0$ or $V_I > V_{DD}$) | ± 20 mA |
| Output clamp current, $I_{OK}$ (Vo < 0 or Vo >$V_{DD}$) | ± 20 mA |
| Maximum output current sunk by any I/O pin | 25 mA |
| Maximum output current sourced by any I/O pin | 25 mA |
| Maximum current sunk by PORTA and PORTC (combined) | 200 mA |
| Maximum current sourced PORTA and PORTC (combined) | 200 mA |

**Note 1:** Power dissipation is calculated as follows: $P_{DIS} = V_{DD} \times \{I_{DD} - \sum I_{OH}\} + \sum \{(V_{DD} - V_{OH}) \times I_{OH}\} + \sum (V_{Ol} \times I_{OL})$.

**† NOTICE**: Stresses above those listed under 'Absolute Maximum Ratings' may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

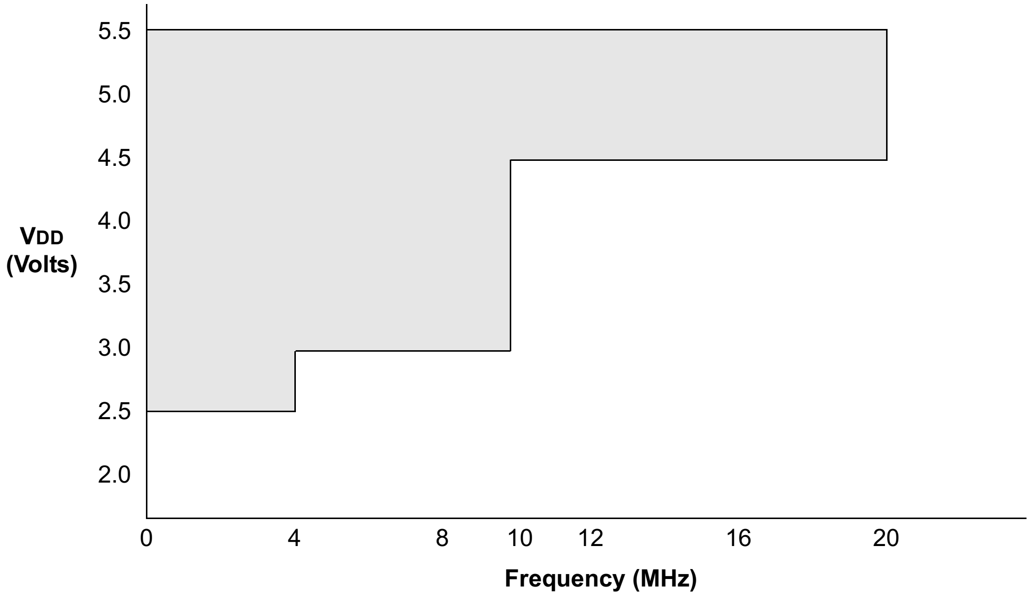
**Note:** Voltage spikes below $V_{SS}$ at the $\overline{MCLR}$ pin, inducing currents greater than 80 mA, may cause latch-up. Thus, a series resistor of 50-100 Ω should be used when applying a "low" level to the $\overline{MCLR}$ pin, rather than pulling this pin directly to $V_{SS}$.

**FIGURE 12-1: PIC16F630/676 WITH A/D DISABLED VOLTAGE-FREQUENCY GRAPH, -40°C ≤ TA ≤ +125°C**



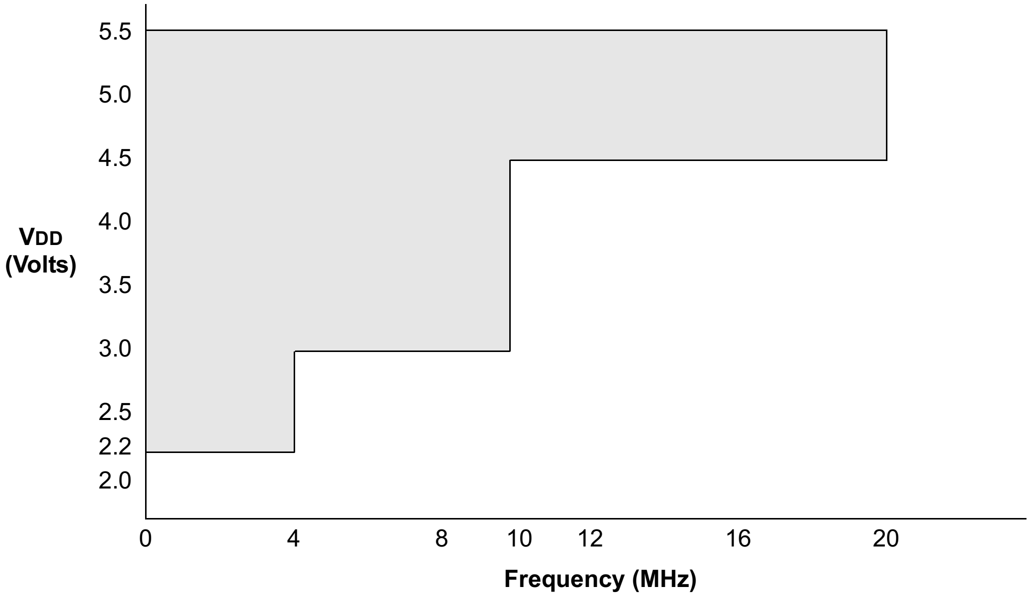
**Note 1:** The shaded region indicates the permissible combinations of voltage and frequency.

**FIGURE 12-2: PIC16F676 WITH A/D ENABLED VOLTAGE-FREQUENCY GRAPH, -40°C ≤ TA ≤ +125°C**



**Note 1:** The shaded region indicates the permissible combinations of voltage and frequency.

**FIGURE 12-3: PIC16F676 WITH A/D ENABLED VOLTAGE-FREQUENCY GRAPH, 0°C ≤ TA ≤ +125°C**



**Note 1:** The shaded region indicates the permissible combinations of voltage and frequency.

## 12.1 DC Characteristics: PIC16F630/676-I (Industrial), PIC16F630/676-E (Extended)

| **DC CHARACTERISTICS** | | | **Standard Operating Conditions (unless otherwise stated)**<br>Operating temperature -40°C ≤ TA ≤ +85°C for industrial<br>-40°C ≤ TA ≤ +125°C for extended | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Sym** | **Characteristic** | **Min** | **Typ†** | **Max** | **Units** | **Conditions** |
| | VDD | **Supply Voltage** | | | | | FOSC < = 4 MHz: |
| D001 | | | 2.0 | — | 5.5 | V | PIC16F630/676 with A/D off |
| D001A | | | 2.2 | — | 5.5 | V | PIC16F676 with A/D on, 0°C to +125°C |
| D001B | | | 2.5 | — | 5.5 | V | PIC16F676 with A/D on, -40°C to +125°C |
| D001C | | | 3.0 | — | 5.5 | V | 4 MHz < FOSC < = 10 MHz |
| D001D | | | 4.5 | — | 5.5 | V | |
| D002 | VDR | **RAM Data Retention Voltage**[(1)] | 1.5* | — | — | V | Device in Sleep mode |
| D003 | VPOR | **VDD Start Voltage** to ensure internal Power-on Reset signal | — | VSS | — | V | See section on Power-on Reset for details |
| D004 | SVDD | **VDD Rise Rate** to ensure internal Power-on Reset signal | 0.05* | — | — | V/ms | See section on Power-on Reset for details |
| D005 | VBOD | | — | 2.1 | — | V | |

* These parameters are characterized but not tested.

† Data in “Typ” column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** This is the limit to which VDD can be lowered in Sleep mode without losing RAM data.

## 12.2 DC Characteristics: PIC16F630/676-I (Industrial)

| | | **Standard Operating Conditions (unless otherwise stated)**<br>Operating temperature -40°C ≤ TA ≤ +85°C for industrial | | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Device Characteristics** | **Min** | **Typ†** | **Max** | **Units** | **Conditions** | |
| | | | | | | **VDD** | **Note** |
| D010 | **Supply Current (IDD)** | — | 9 | 16 | μA | 2.0 | FOSC = 32 kHz<br>LP Oscillator Mode |
| | | — | 18 | 28 | μA | 3.0 | |
| | | — | 35 | 54 | μA | 5.0 | |
| D011 | | — | 110 | 150 | μA | 2.0 | FOSC = 1 MHz<br>XT Oscillator Mode |
| | | — | 190 | 280 | μA | 3.0 | |
| | | — | 330 | 450 | μA | 5.0 | |
| D012 | | — | 220 | 280 | μA | 2.0 | FOSC = 4 MHz<br>XT Oscillator Mode |
| | | — | 370 | 650 | μA | 3.0 | |
| | | — | 0.6 | 1.4 | mA | 5.0 | |
| D013 | | — | 70 | 110 | μA | 2.0 | FOSC = 1 MHz<br>EC Oscillator Mode |
| | | — | 140 | 250 | μA | 3.0 | |
| | | — | 260 | 390 | μA | 5.0 | |
| D014 | | — | 180 | 250 | μA | 2.0 | FOSC = 4 MHz<br>EC Oscillator Mode |
| | | — | 320 | 470 | μA | 3.0 | |
| | | — | 580 | 850 | μA | 5.0 | |
| D015 | | — | 340 | 450 | μA | 2.0 | FOSC = 4 MHz<br>INTOSC Mode |
| | | — | 500 | 780 | μA | 3.0 | |
| | | — | 0.8 | 1.1 | mA | 5.0 | |
| D016 | | — | 180 | 250 | μA | 2.0 | FOSC = 4 MHz<br>EXTRC Mode |
| | | — | 320 | 450 | μA | 3.0 | |
| | | — | 580 | 800 | μA | 5.0 | |
| D017 | | — | 2.1 | 2.95 | mA | 4.5 | FOSC = 20 MHz<br>HS Oscillator Mode |
| | | — | 2.4 | 3.0 | mA | 5.0 | |

† Data in “Typ” column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** The test conditions for all IDD measurements in Active Operation mode are: OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to VDD; $\overline{\text{MCLR}}$ = VDD; WDT disabled.

**2:** The supply current is mainly a function of the operating voltage and frequency. Other factors such as I/O pin loading and switching rate, oscillator type, internal code execution pattern, and temperature also have an impact on the current consumption.

## 12.3 DC Characteristics: PIC16F630/676-I (Industrial)

| | | **Standard Operating Conditions (unless otherwise stated)**<br>Operating temperature -40°C ≤ TA ≤ +85°C for industrial | | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Device Characteristics** | **Min** | **Typ†** | **Max** | **Units** | **Conditions** | |
| | | | | | | **VDD** | **Note** |
| D020 | Power-down Base Current (IPD) | — | 0.99 | 700 | nA | 2.0 | WDT, BOD, Comparators, VREF, and T1OSC disabled |
| | | — | 1.2 | 770 | nA | 3.0 | |
| | | — | 2.9 | 995 | nA | 5.0 | |
| D021 | | — | 0.3 | 1.5 | μA | 2.0 | WDT Current[(1)] |
| | | — | 1.8 | 3.5 | μA | 3.0 | |
| | | — | 8.4 | 17 | μA | 5.0 | |
| D022 | | — | 58 | 70 | μA | 3.0 | BOD Current[(1)] |
| | | — | 109 | 130 | μA | 5.0 | |
| D023 | | — | 3.3 | 6.5 | μA | 2.0 | Comparator Current[(1)] |
| | | — | 6.1 | 8.5 | μA | 3.0 | |
| | | — | 11.5 | 16 | μA | 5.0 | |
| D024 | | — | 58 | 70 | μA | 2.0 | CVREF Current[(1)] |
| | | — | 85 | 100 | μA | 3.0 | |
| | | — | 138 | 160 | μA | 5.0 | |
| D025 | | — | 4.0 | 6.5 | μA | 2.0 | T1 OSC Current[(1)] |
| | | — | 4.6 | 7.0 | μA | 3.0 | |
| | | — | 6.0 | 10.5 | μA | 5.0 | |
| D026 | | — | 1.2 | 755 | nA | 3.0 | A/D Current[(1)] |
| | | — | 0.0022 | 1.0 | μA | 5.0 | |

† Data in “Typ” column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** The peripheral current is the sum of the base IDD or IPD and the additional current consumed when this peripheral is enabled. The peripheral Δ current can be determined by subtracting the base IDD or IPD current from this limit. Max values should be used when calculating total current consumption.

**2:** The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in high-impedance state and tied to VDD.

## 12.4 DC Characteristics: PIC16F630/676-E (Extended)

| | | **Standard Operating Conditions (unless otherwise stated)**<br>Operating temperature -40°C ≤ TA ≤ +125°C for extended | | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Device Characteristics** | **Min** | **Typ†** | **Max** | **Units** | **Conditions** | |
| | | | | | | **VDD** | **Note** |
| D010E | **Supply Current (IDD)** | — | 9 | 16 | μA | 2.0 | FOSC = 32 kHz<br>LP Oscillator Mode |
| | | — | 18 | 28 | μA | 3.0 | |
| | | — | 35 | 54 | μA | 5.0 | |
| D011E | | — | 110 | 150 | μA | 2.0 | FOSC = 1 MHz<br>XT Oscillator Mode |
| | | — | 190 | 280 | μA | 3.0 | |
| | | — | 330 | 450 | μA | 5.0 | |
| D012E | | — | 220 | 280 | μA | 2.0 | FOSC = 4 MHz<br>XT Oscillator Mode |
| | | — | 370 | 650 | μA | 3.0 | |
| | | — | 0.6 | 1.4 | mA | 5.0 | |
| D013E | | — | 70 | 110 | μA | 2.0 | FOSC = 1 MHz<br>EC Oscillator Mode |
| | | — | 140 | 250 | μA | 3.0 | |
| | | — | 260 | 390 | μA | 5.0 | |
| D014E | | — | 180 | 250 | μA | 2.0 | FOSC = 4 MHz<br>EC Oscillator Mode |
| | | — | 320 | 470 | μA | 3.0 | |
| | | — | 580 | 850 | μA | 5.0 | |
| D015E | | — | 340 | 450 | μA | 2.0 | FOSC = 4 MHz<br>INTOSC Mode |
| | | — | 500 | 780 | μA | 3.0 | |
| | | — | 0.8 | 1.1 | mA | 5.0 | |
| D016E | | — | 180 | 250 | μA | 2.0 | FOSC = 4 MHz<br>EXTRC Mode |
| | | — | 320 | 450 | μA | 3.0 | |
| | | — | 580 | 800 | μA | 5.0 | |
| D017E | | — | 2.1 | 2.95 | mA | 4.5 | FOSC = 20 MHz<br>HS Oscillator Mode |
| | | — | 2.4 | 3.0 | mA | 5.0 | |

† Data in “Typ” column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** The test conditions for all IDD measurements in Active Operation mode are: OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to VDD; $\overline{\text{MCLR}}$ = VDD; WDT disabled.

**2:** The supply current is mainly a function of the operating voltage and frequency. Other factors such as I/O pin loading and switching rate, oscillator type, internal code execution pattern, and temperature also have an impact on the current consumption.

## 12.5 DC Characteristics: PIC16F630/676-E (Extended)

| | | **Standard Operating Conditions (unless otherwise stated)**<br>Operating temperature -40°C ≤ TA ≤ +125°C for extended | | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Device Characteristics** | **Min** | **Typ†** | **Max** | **Units** | **Conditions** | |
| | | | | | | **VDD** | **Note** |
| D020E | Power-down Base Current (IPD) | — | 0.00099 | 3.5 | μA | 2.0 | WDT, BOD, Comparators, VREF, and T1OSC disabled |
| | | — | 0.0012 | 4.0 | μA | 3.0 | |
| | | — | 0.0029 | 8.0 | μA | 5.0 | |
| D021E | | — | 0.3 | 6.0 | μA | 2.0 | WDT Current(1) |
| | | — | 1.8 | 9.0 | μA | 3.0 | |
| | | — | 8.4 | 20 | μA | 5.0 | |
| D022E | | — | 58 | 70 | μA | 3.0 | BOD Current(1) |
| | | — | 109 | 130 | μA | 5.0 | |
| D023E | | — | 3.3 | 10 | μA | 2.0 | Comparator Current(1) |
| | | — | 6.1 | 13 | μA | 3.0 | |
| | | — | 11.5 | 24 | μA | 5.0 | |
| D024E | | — | 58 | 70 | μA | 2.0 | CVREF Current(1) |
| | | — | 85 | 100 | μA | 3.0 | |
| | | — | 138 | 165 | μA | 5.0 | |
| D025E | | — | 4.0 | 10 | μA | 2.0 | T1 OSC Current(1) |
| | | — | 4.6 | 12 | μA | 3.0 | |
| | | — | 6.0 | 20 | μA | 5.0 | |
| D026E | | — | 0.0012 | 6.0 | μA | 3.0 | A/D Current(1) |
| | | — | 0.0022 | 8.5 | μA | 5.0 | |

† Data in "Typ" column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** The peripheral current is the sum of the base IDD or IPD and the additional current consumed when this peripheral is enabled. The peripheral ∆ current can be determined by subtracting the base IDD or IPD current from this limit. Max values should be used when calculating total current consumption.

**2:** The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in high-impedance state and tied to VDD.

## 12.6 DC Characteristics: PIC16F630/676-I (Industrial), PIC16F630/676-E (Extended)

| **DC CHARACTERISTICS** | | | **Standard Operating Conditions (unless otherwise stated)**<br>Operating temperature -40°C ≤ TA ≤ +85°C for industrial<br>-40°C ≤ TA ≤ +125°C for extended | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Sym** | **Characteristic** | **Min** | **Typ†** | **Max** | **Units** | **Conditions** |
| | | **Input Low Voltage** | | | | | |
| | VIL | I/O ports | | | | | |
| D030 | | with TTL buffer | VSS | — | 0.8 | V | 4.5V ≤ VDD ≤ 5.5V |
| D030A | | | VSS | — | 0.15 VDD | V | Otherwise |
| D031 | | with Schmitt Trigger buffer | VSS | — | 0.2 VDD | V | Entire range |
| D032 | | $\overline{\text{MCLR}}$, OSC1 (RC mode) | VSS | — | 0.2 VDD | V | |
| D033 | | OSC1 (XT and LP modes) | VSS | — | 0.3 | V | **(Note 1)** |
| D033A | | OSC1 (HS mode) | VSS | — | 0.3 VDD | V | **(Note 1)** |
| | | **Input High Voltage** | | | | | |
| | VIH | I/O ports | | — | | | |
| D040 | | with TTL buffer | 2.0 | — | VDD | V | 4.5V ≤ VDD ≤ 5.5V |
| D040A | | | (0.25 VDD+0.8) | — | VDD | V | otherwise |
| D041 | | with Schmitt Trigger buffer | 0.8 VDD | — | VDD | | entire range |
| D042 | | $\overline{\text{MCLR}}$ | 0.8 VDD | — | VDD | V | |
| D043 | | OSC1 (XT and LP modes) | 1.6 | — | VDD | V | **(Note 1)** |
| D043A | | OSC1 (HS mode) | 0.7 VDD | — | VDD | V | **(Note 1)** |
| D043B | | OSC1 (RC mode) | 0.9 VDD | — | VDD | V | |
| D070 | IPUR | **PORTA Weak Pull-up Current** | 50* | 250 | 400* | μA | VDD = 5.0V, VPIN = VSS |
| | | **Input Leakage Current$^{(3)}$** | | | | | |
| D060 | IIL | I/O ports | — | ± 0.1 | ± 1 | μA | VSS ≤ VPIN ≤ VDD,<br>Pin at high-impedance |
| D060A | | Analog inputs | — | ± 0.1 | ± 1 | μA | VSS ≤ VPIN ≤ VDD |
| D060B | | VREF | — | ± 0.1 | ± 1 | μA | VSS ≤ VPIN ≤ VDD |
| D061 | | $\overline{\text{MCLR}}^{(2)}$ | — | ± 0.1 | ± 5 | μA | VSS ≤ VPIN ≤ VDD |
| D063 | | OSC1 | — | ± 0.1 | ± 5 | μA | VSS ≤ VPIN ≤ VDD, XT, HS and LP osc configuration |
| | | **Output Low Voltage** | | | | | |
| D080 | VOL | I/O ports | — | — | 0.6 | V | IOL = 8.5 mA, VDD = 4.5V (Ind.) |
| D083 | | OSC2/CLKOUT (RC mode) | — | — | 0.6 | V | IOL = 1.6 mA, VDD = 4.5V (Ind.)<br>IOL = 1.2 mA, VDD = 4.5V (Ext.) |
| | | **Output High Voltage** | | | | | |
| D090 | VOH | I/O ports | VDD - 0.7 | — | — | V | IOH = -3.0 mA, VDD = 4.5V (Ind.) |
| D092 | | OSC2/CLKOUT (RC mode) | VDD - 0.7 | — | — | V | IOH = -1.3 mA, VDD = 4.5V (Ind.)<br>IOH = -1.0 mA, VDD = 4.5V (Ext.) |

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** In RC oscillator configuration, the OSC1/CLKIN pin is a Schmitt Trigger input. It is not recommended to use an external clock in RC mode.

**2:** The leakage current on the $\overline{\text{MCLR}}$ pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.

**3:** Negative current is defined as current sourced by the pin.

## 12.7 DC Characteristics: PIC16F630/676-I (Industrial), PIC16F630/676-E (Extended) (Cont.)

| **DC CHARACTERISTICS** | | | **Standard Operating Conditions (unless otherwise stated)**<br>Operating temperature -40°C ≤ TA ≤ +85°C for industrial<br>-40°C ≤ TA ≤ +125°C for extended | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Sym** | **Characteristic** | **Min** | **Typ†** | **Max** | **Units** | **Conditions** |
| | | **Capacitive Loading Specs on Output Pins** | | | | | |
| D100 | COSC2 | OSC2 pin | — | — | 15* | pF | In XT, HS and LP modes when external clock is used to drive OSC1 |
| D101 | CIO | All I/O pins | — | — | 50* | pF | |
| | | **Data EEPROM Memory** | | | | | |
| D120 | ED | Byte Endurance | 100K | 1M | — | E/W | -40°C ≤ TA ≤ +85°C |
| D120A | ED | Byte Endurance | 10K | 100K | — | E/W | +85°C ≤ TA ≤ +125°C |
| D121 | VDRW | VDD for Read/Write | VMIN | — | 5.5 | V | Using EECON to read/write<br>VMIN = Minimum operating voltage |
| D122 | TDEW | Erase/Write cycle time | — | 5 | 6 | ms | |
| D123 | TRETD | Characteristic Retention | 40 | — | — | Year | Provided no other specifications are violated |
| D124 | TREF | Number of Total Erase/Write Cycles before Refresh(1) | 1M | 10M | — | E/W | -40°C ≤ TA ≤ +85°C |
| | | **Program Flash Memory** | | | | | |
| D130 | EP | Cell Endurance | 10K | 100K | — | E/W | -40°C ≤ TA ≤ +85°C |
| D130A | ED | Cell Endurance | 1K | 10K | — | E/W | +85°C ≤ TA ≤ +125°C |
| D131 | VPR | VDD for Read | VMIN | — | 5.5 | V | VMIN = Minimum operating voltage |
| D132 | VPEW | VDD for Erase/Write | 4.5 | — | 5.5 | V | |
| D133 | TPEW | Erase/Write cycle time | — | 2 | 2.5 | ms | |
| D134 | TRETD | Characteristic Retention | 40 | — | — | Year | Provided no other specifications are violated |

* These parameters are characterized but not tested.

† Data in 'Typ' column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** See Section 8.5.1 for additional information.

## 12.8 TIMING PARAMETER SYMBOLOGY

The timing parameter symbols have been created with one of the following formats:

1. TppS2ppS
2. TppS

| | | | |
|---|---|---|---|
| **T** | | | |
| F | Frequency | T | Time |

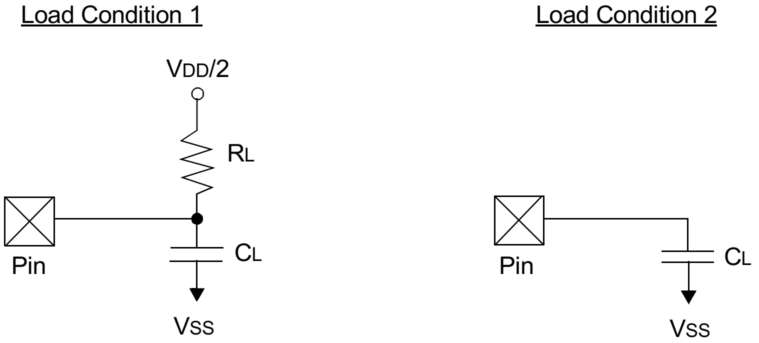
Lowercase letters (pp) and their meanings:

| | | | |
|---|---|---|---|
| **pp** | | | |
| cc | CCP1 | osc | OSC1 |
| ck | CLKOUT | rd | $\overline{\text{RD}}$ |
| cs | $\overline{\text{CS}}$ | rw | $\overline{\text{RD}}$ or $\overline{\text{WR}}$ |
| di | SDI | sc | SCK |
| do | SDO | ss | $\overline{\text{SS}}$ |
| dt | Data in | t0 | T0CKI |
| io | I/O port | t1 | T1CKI |
| mc | $\overline{\text{MCLR}}$ | wr | $\overline{\text{WR}}$ |

Uppercase letters and their meanings:

| | | | |
|---|---|---|---|
| **S** | | | |
| F | Fall | P | Period |
| H | High | R | Rise |
| I | Invalid (High-impedance) | V | Valid |
| L | Low | Z | High-impedance |

**FIGURE 12-4: LOAD CONDITIONS**



**Legend:**

$R_L$ = 464Ω

$C_L$ = 50 pF for all pins

15 pF for OSC2 output

## 12.9 AC CHARACTERISTICS: PIC16F630/676 (INDUSTRIAL, EXTENDED)

**FIGURE 12-5: EXTERNAL CLOCK TIMING**

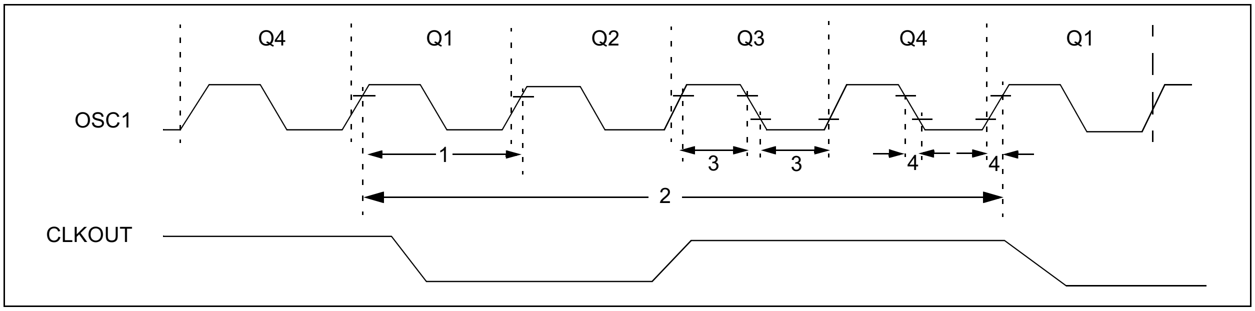


**TABLE 12-1: EXTERNAL CLOCK TIMING REQUIREMENTS**

| Param No. | Sym | Characteristic | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|
| | FOSC | External CLKIN Frequency[(1)] | DC | — | 37 | kHz | LP Osc mode |
| | | | DC | — | 4 | MHz | XT mode |
| | | | DC | — | 20 | MHz | HS mode |
| | | | DC | — | 20 | MHz | EC mode |
| | | Oscillator Frequency[(1)] | 5 | — | 37 | kHz | LP Osc mode |
| | | | — | 4 | — | MHz | INTOSC mode |
| | | | DC | — | 4 | MHz | RC Osc mode |
| | | | 0.1 | — | 4 | MHz | XT Osc mode |
| | | | 1 | — | 20 | MHz | HS Osc mode |
| 1 | TOSC | External CLKIN Period[(1)] | 27 | — | ∞ | μs | LP Osc mode |
| | | | 50 | — | ∞ | ns | HS Osc mode |
| | | | 50 | — | ∞ | ns | EC Osc mode |
| | | | 250 | — | ∞ | ns | XT Osc mode |
| | | Oscillator Period[(1)] | 27 | | 200 | μs | LP Osc mode |
| | | | — | 250 | — | ns | INTOSC mode |
| | | | 250 | — | — | ns | RC Osc mode |
| | | | 250 | — | 10,000 | ns | XT Osc mode |
| | | | 50 | — | 1,000 | ns | HS Osc mode |
| 2 | TCY | Instruction Cycle Time[(1)] | 200 | TCY | DC | ns | TCY = 4/FOSC |
| 3 | TosL, | External CLKIN (OSC1) High | 2* | — | — | μs | LP oscillator, TOSC L/H duty cycle |
| | TosH | External CLKIN Low | 20* | — | — | ns | HS oscillator, TOSC L/H duty cycle |
| | | | 100 * | — | — | ns | XT oscillator, TOSC L/H duty cycle |
| 4 | TosR, | External CLKIN Rise | — | — | 50* | ns | LP oscillator |
| | TosF | External CLKIN Fall | — | — | 25* | ns | XT oscillator |
| | | | — | — | 15* | ns | HS oscillator |

* These parameters are characterized but not tested.

† Data in “Typ” column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** Instruction cycle period (TCY) equals four times the input oscillator time-base period. All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at “min” values with an external clock applied to OSC1 pin. When an external clock input is used, the “max” cycle time limit is “DC” (no clock) for all devices.

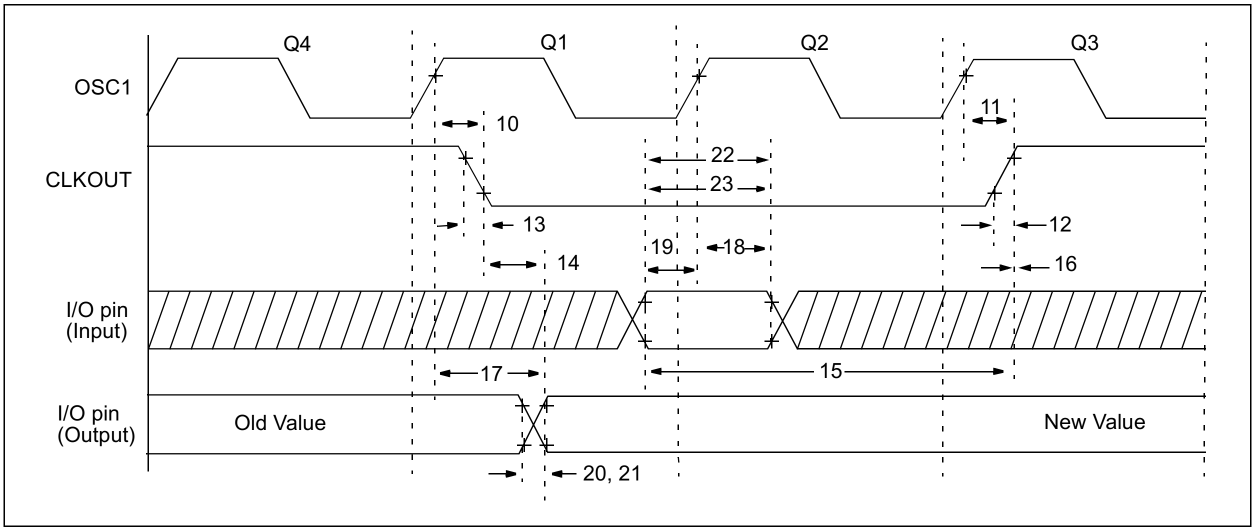
### TABLE 12-2: PRECISION INTERNAL OSCILLATOR PARAMETERS

| Param No. | Sym | Characteristic | Freq Tolerance | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|---|
| F10 | FOSC | Internal Calibrated INTOSC Frequency | ±1 | 3.96 | 4.00 | 4.04 | MHz | $V_{DD}$ = 3.5V, 25°C |
| | | | ±2 | 3.92 | 4.00 | 4.08 | MHz | 2.5V ≤ $V_{DD}$ ≤ 5.5V<br>0°C ≤ $T_A$ ≤ +85°C |
| | | | ±5 | 3.80 | 4.00 | 4.20 | MHz | 2.0V ≤ $V_{DD}$ ≤ 5.5V<br>-40°C ≤ $T_A$ ≤ +85°C (IND)<br>-40°C ≤ $T_A$ ≤ +125°C (EXT) |
| F14 | TIOSC ST | Oscillator Wake-up from Sleep start-up time* | — | — | 6 | 8 | μs | $V_{DD}$ = 2.0V, -40°C to +85°C |
| | | | — | — | 4 | 6 | μs | $V_{DD}$ = 3.0V, -40°C to +85°C |
| | | | — | — | 3 | 5 | μs | $V_{DD}$ = 5.0V, -40°C to +85°C |

* These parameters are characterized but not tested.

† Data in “Typ” column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**FIGURE 12-6: CLKOUT AND I/O TIMING**



### TABLE 12-3: CLKOUT AND I/O TIMING REQUIREMENTS

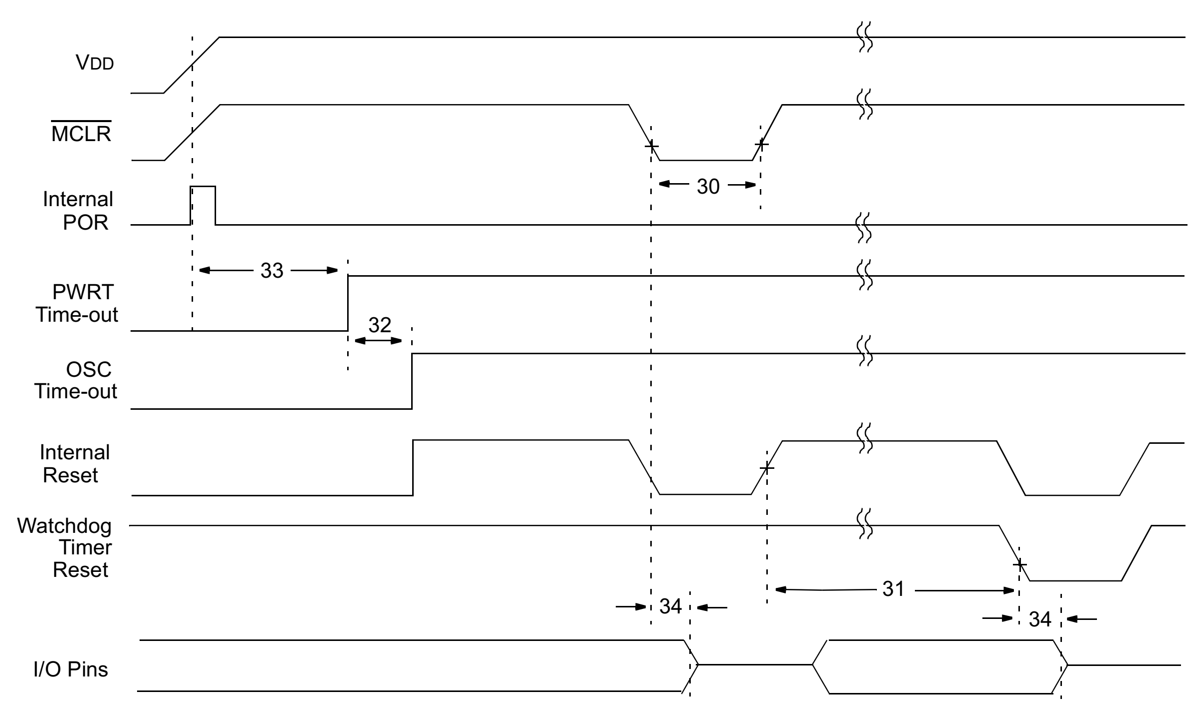
| Param No. | Sym | Characteristic | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|
| 10 | TosH2ckL | OSC1↑ to CLOUT↓ | — | 75 | 200 | ns | **(Note 1)** |
| 11 | TosH2ckH | OSC1↑ to CLOUT↑ | — | 75 | 200 | ns | **(Note 1)** |
| 12 | TckR | CLKOUT rise time | — | 35 | 100 | ns | **(Note 1)** |
| 13 | TckF | CLKOUT fall time | — | 35 | 100 | ns | **(Note 1)** |
| 14 | TckL2ioV | CLKOUT↓ to Port out valid | — | — | 20 | ns | **(Note 1)** |
| 15 | TioV2ckH | Port in valid before CLKOUT↑ | TOSC + 200 ns | — | — | ns | **(Note 1)** |
| 16 | TckH2ioI | Port in hold after CLKOUT↑ | 0 | — | — | ns | **(Note 1)** |
| 17 | TosH2ioV | OSC1↑ (Q1 cycle) to Port out valid | — | 50 | 150 * | ns | |
| | | | — | — | 300 | ns | |
| 18 | TosH2ioI | OSC1↑ (Q2 cycle) to Port input invalid (I/O in hold time) | 100 | — | — | ns | |
| 19 | TioV2osH | Port input valid to OSC1↑ (I/O in setup time) | 0 | — | — | ns | |
| 20 | TioR | Port output rise time | — | 10 | 40 | ns | |
| 21 | TioF | Port output fall time | — | 10 | 40 | ns | |
| 22 | Tinp | INT pin high or low time | 25 | — | — | ns | |
| 23 | Trbp | PORTA change INT high or low time | TCY | — | — | ns | |

* These parameters are characterized but not tested.

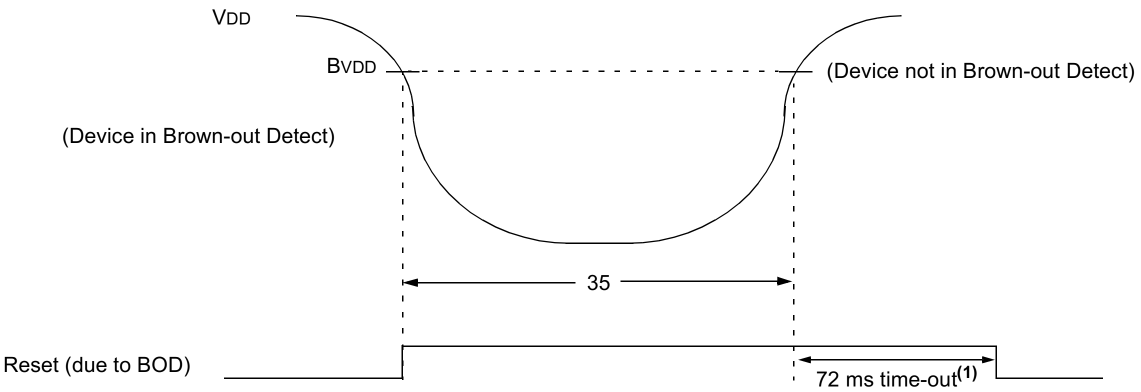
† Data in "Typ" column is at 5.0V, 25°C unless otherwise stated.

**Note 1:** Measurements are taken in RC mode where CLKOUT output is 4xTOSC.

### FIGURE 12-7: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING



### FIGURE 12-8: BROWN-OUT DETECT TIMING AND CHARACTERISTICS



**Note 1:** 72 ms delay only if $\overline{\text{PWRTE}}$ bit in Configuration Word is programmed to '0'.

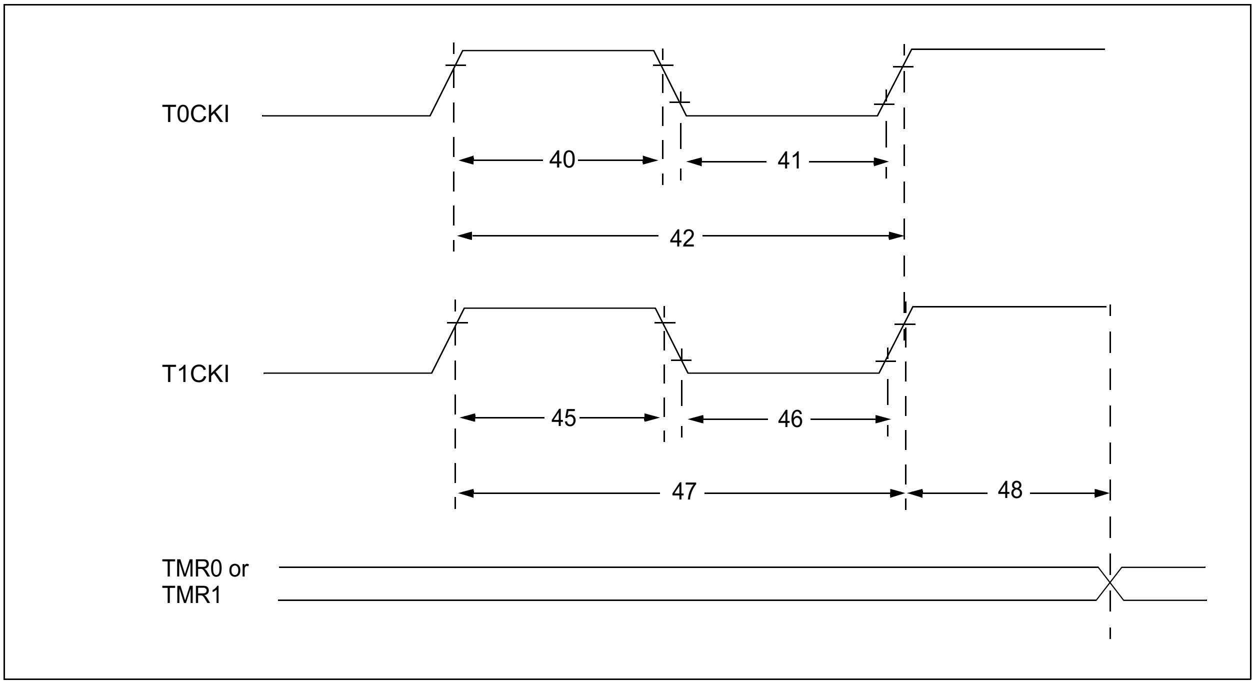
### TABLE 12-4: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER, POWER-UP TIMER, AND BROWN-OUT DETECT REQUIREMENTS

| Param No. | Sym | Characteristic | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|
| 30 | TMCL | $\overline{\text{MCLR}}$ Pulse Width (low) | 2<br>11 | —<br>18 | —<br>24 | μs<br>ms | VDD = 5V, -40°C to +85°C<br>Extended temperature |
| 31 | TWDT | Watchdog Timer Time-out Period (No Prescaler) | 10<br>10 | 17<br>17 | 25<br>30 | ms<br>ms | VDD = 5V, -40°C to +85°C<br>Extended temperature |
| 32 | TOST | Oscillation Start-up Timer Period | — | 1024TOSC | — | — | TOSC = OSC1 period |
| 33* | TPWRT | Power-up Timer Period | 28*<br>TBD | 72<br>TBD | 132*<br>TBD | ms<br>ms | VDD = 5V, -40°C to +85°C<br>Extended Temperature |
| 34 | TIOZ | I/O High-impedance from $\overline{\text{MCLR}}$ Low or Watchdog Timer Reset | — | — | 2.0 | μs | |
| | BVDD | Brown-out Detect Voltage | 2.025 | — | 2.175 | V | |
| | | Brown-out Hysteresis | TBD | — | — | — | |
| 35 | TBOD | Brown-out Detect Pulse Width | 100* | — | — | μs | VDD ≤ BVDD (D005) |

* These parameters are characterized but not tested.

† Data in “Typ” column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

### FIGURE 12-9: TIMER0 AND TIMER1 EXTERNAL CLOCK TIMINGS



### TABLE 12-5: TIMER0 AND TIMER1 EXTERNAL CLOCK REQUIREMENTS

| Param No. | Sym | Characteristic | | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|---|
| 40* | Tt0H | T0CKI High Pulse Width | No Prescaler | 0.5 TCY + 20 | — | — | ns | |
| | | | With Prescaler | 10 | — | — | ns | |
| 41* | Tt0L | T0CKI Low Pulse Width | No Prescaler | 0.5 TCY + 20 | — | — | ns | |
| | | | With Prescaler | 10 | — | — | ns | |
| 42* | Tt0P | T0CKI Period | | Greater of:<br>20 or $\frac{T_{CY} + 40}{N}$ | — | — | ns | N = prescale value (2, 4, ..., 256) |
| 45* | Tt1H | T1CKI High Time | Synchronous, No Prescaler | 0.5 TCY + 20 | — | — | ns | |
| | | | Synchronous, with Prescaler | 15 | — | — | ns | |
| | | | Asynchronous | 30 | — | — | ns | |
| 46* | Tt1L | T1CKI Low Time | Synchronous, No Prescaler | 0.5 TCY + 20 | — | — | ns | |
| | | | Synchronous, with Prescaler | 15 | — | — | ns | |
| | | | Asynchronous | 30 | — | — | ns | |
| 47* | Tt1P | T1CKI Input Period | Synchronous | Greater of:<br>30 or $\frac{T_{CY} + 40}{N}$ | — | — | ns | N = prescale value (1, 2, 4, 8) |
| | | | Asynchronous | 60 | — | — | ns | |
| | Ft1 | Timer1 oscillator input frequency range (oscillator enabled by setting bit T1OSCEN) | | DC | — | 200* | kHz | |
| 48 | TCKEZtmr1 | Delay from external clock edge to timer increment | | 2 TOSC* | — | 7 TOSC* | — | |

* These parameters are characterized but not tested.

† Data in “Typ” column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

### TABLE 12-6: COMPARATOR SPECIFICATIONS

| **Comparator Specifications** | | **Standard Operating Conditions**<br>-40°C to +125°C (unless otherwise stated) | | | | |
|---|---|---|---|---|---|---|
| **Sym** | **Characteristics** | **Min** | **Typ** | **Max** | **Units** | **Comments** |
| VOS | Input Offset Voltage | — | ± 5.0 | ± 10 | mV | |
| VCM | Input Common Mode Voltage | 0 | — | VDD - 1.5 | V | |
| CMRR | Common Mode Rejection Ratio | +55* | — | — | db | |
| TRT | Response Time(1) | — | 150 | 400* | ns | |
| TMC2COV | Comparator Mode Change to Output Valid | — | — | 10* | μs | |

* These parameters are characterized but not tested.

**Note 1:** Response time measured with one comparator input at (VDD - 1.5)/2 while the other input transitions from VSS to VDD - 1.5V.

### TABLE 12-7: COMPARATOR VOLTAGE REFERENCE SPECIFICATIONS

| **Voltage Reference Specifications** | | **Standard Operating Conditions**<br>-40°C to +125°C (unless otherwise stated) | | | | |
|---|---|---|---|---|---|---|
| **Sym** | **Characteristics** | **Min** | **Typ** | **Max** | **Units** | **Comments** |
| | Resolution | —<br>— | VDD/24*<br>VDD/32 | —<br>— | LSb<br>LSb | Low Range (VRR = `1`)<br>High Range (VRR = `0`) |
| | Absolute Accuracy | —<br>— | —<br>— | ± 1/2*<br>± 1/2* | LSb<br>LSb | Low Range (VRR = `1`)<br>High Range (VRR = `0`) |
| | Unit Resistor Value (R) | — | 2k* | — | Ω | |
| | Settling Time(1) | — | — | 10* | μs | |

* These parameters are characterized but not tested.

**Note 1:** Settling time measured while VRR = `1` and VR<3:0> transitions from `0000` to `1111`.

**TABLE 12-8: PIC16F676 A/D CONVERTER CHARACTERISTICS**:

| Param No. | Sym | Characteristic | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|
| A01 | NR | Resolution | — | — | 10 bits | bit | |
| A02 | EABS | Total Absolute Error* | — | — | ±1 | LSb | VREF = 5.0V |
| A03 | EIL | Integral Error | — | — | ±1 | LSb | VREF = 5.0V |
| A04 | EDL | Differential Error | — | — | ±1 | LSb | No missing codes to 10 bits<br>VREF = 5.0V |
| A05 | EFS | Full Scale Range | 2.2* | — | 5.5* | V | |
| A06 | EOFF | Offset Error | — | — | ±1 | LSb | VREF = 5.0V |
| A07 | EGN | Gain Error | — | — | ±1 | LSb | VREF = 5.0V |
| A10 | — | Monotonicity | — | guaranteed(3) | — | — | VSS ≤ VAIN ≤ VREF+ |
| A20<br>A20A | VREF | Reference Voltage | 2.0<br>2.5 | — | —<br>VDD + 0.3 | V | <br>Absolute minimum to ensure 10-bit accuracy |
| A21 | VREF | Reference V High (VDD or VREF) | VSS | — | VDD | V | |
| A25 | VAIN | Analog Input Voltage | VSS | — | VREF | V | |
| A30 | ZAIN | Recommended Impedance of Analog Voltage Source | — | — | 10 | kΩ | |
| A50 | IREF | VREF Input Current(2) | 10<br>— | —<br>— | 1000<br>10 | μA<br>μA | During VAIN acquisition. Based on differential of VHOLD to VAIN.<br>During A/D conversion cycle. |

* These parameters are characterized but not tested.

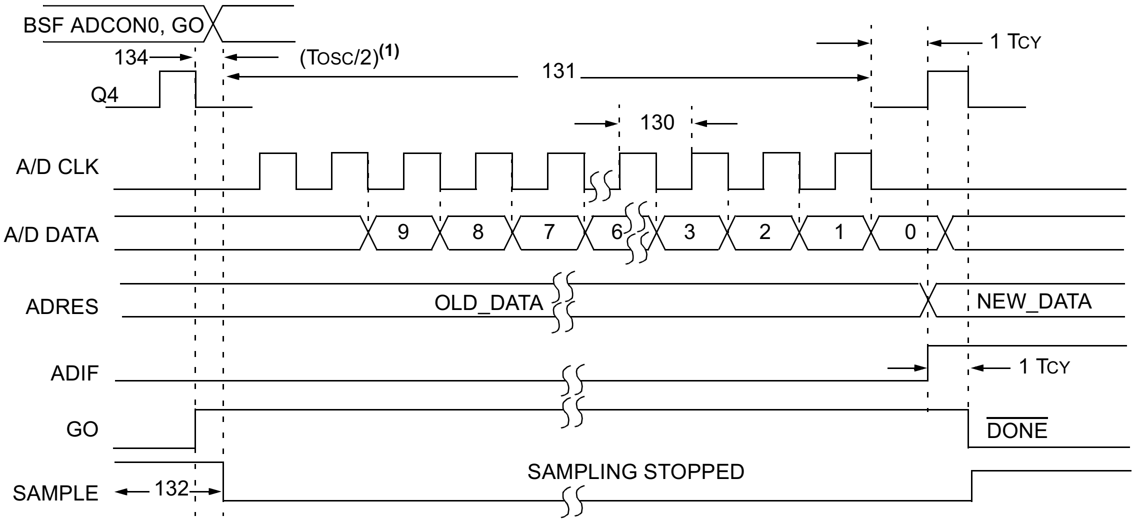
† Data in "Typ" column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** When A/D is off, it will not consume any current other than leakage current. The power-down current spec includes any such leakage from the A/D module.

**2:** VREF current is from External VREF or VDD pin, whichever is selected as reference input.

**3:** The A/D conversion result never decreases with an increase in the input voltage and has no missing codes.

**FIGURE 12-10: PIC16F676 A/D CONVERSION TIMING (NORMAL MODE)**



**Note 1:** If the A/D clock source is selected as RC, a time of TCY is added before the A/D clock starts. This allows the `SLEEP` instruction to be executed.

**TABLE 12-9: PIC16F676 A/D CONVERSION REQUIREMENTS**

| Param No. | Sym | Characteristic | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|
| 130 | TAD | A/D Clock Period | 1.6 | — | — | µs | TOSC based, VREF ≥ 3.0V |
| | | | 3.0* | — | — | µs | TOSC based, VREF full range |
| 130 | TAD | A/D Internal RC Oscillator Period | | | | | ADCS<1:0> = `11` (RC mode) |
| | | | 3.0* | 6.0 | 9.0* | µs | At VDD = 2.5V |
| | | | 2.0* | 4.0 | 6.0* | µs | At VDD = 5.0V |
| 131 | TCNV | Conversion Time (not including Acquisition Time)$^{(1)}$ | — | 11 | — | TAD | Set GO bit to new data in A/D result register |
| 132 | TACQ | Acquisition Time | **(Note 2)** | 11.5 | — | µs | |
| | | | 5* | — | — | µs | The minimum time is the amplifier settling time. This may be used if the “new” input voltage has not changed by more than 1 LSb (i.e., 4.1 mV @ 4.096V) from the last sampled voltage (as stored on CHOLD). |
| 134 | TGO | Q4 to A/D Clock Start | — | TOSC/2 | — | — | If the A/D clock source is selected as RC, a time of TCY is added before the A/D clock starts. This allows the `SLEEP` instruction to be executed. |

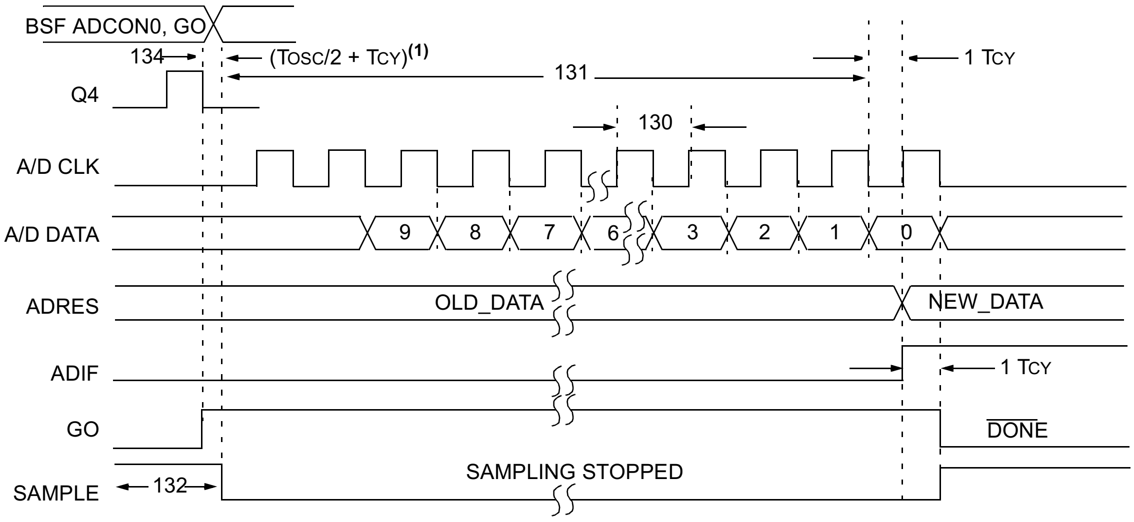
* These parameters are characterized but not tested.

† Data in “Typ” column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** ADRES register may be read on the following TCY cycle.

**2:** See Table 7-1 for minimum conditions.

**FIGURE 12-11: PIC16F676 A/D CONVERSION TIMING (SLEEP MODE)**



**Note 1:** If the A/D clock source is selected as RC, a time of TCY is added before the A/D clock starts. This allows the `SLEEP` instruction to be executed.

**TABLE 12-10: PIC16F676 A/D CONVERSION REQUIREMENTS (SLEEP MODE)**

| Param No. | Sym | Characteristic | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|
| 130 | TAD | A/D Clock Period | 1.6 | — | — | µs | VREF ≥ 3.0V |
| | | | 3.0* | — | — | µs | VREF full range |
| 130 | TAD | A/D Internal RC Oscillator Period | | | | | ADCS<1:0> = `11` (RC mode) |
| | | | 3.0* | 6.0 | 9.0* | µs | At VDD = 2.5V |
| | | | 2.0* | 4.0 | 6.0* | µs | At VDD = 5.0V |
| 131 | TCNV | Conversion Time (not including Acquisition Time)(1) | — | 11 | — | TAD | |
| 132 | TACQ | Acquisition Time | **(Note 2)** | 11.5 | — | µs | |
| | | | 5* | — | — | µs | The minimum time is the amplifier settling time. This may be used if the “new” input voltage has not changed by more than 1 LSb (i.e., 4.1 mV @ 4.096V) from the last sampled voltage (as stored on CHOLD). |
| 134 | TGO | Q4 to A/D Clock Start | — | TOSC/2 + TCY | — | — | If the A/D clock source is selected as RC, a time of TCY is added before the A/D clock starts. This allows the `SLEEP` instruction to be executed. |

* These parameters are characterized but not tested.

† Data in “Typ” column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** ADRES register may be read on the following TCY cycle.

**2:** See Table 7-1 for minimum conditions.

**NOTES:**

## 13.0 DC AND AC CHARACTERISTICS GRAPHS AND TABLES

The graphs and tables provided in this section are for **design guidance** and are **not tested**.

In some graphs or tables, the data presented are **outside specified operating range** (i.e., outside specified VDD range). This is for **information only** and devices are ensured to operate properly only within the specified range.

The data presented in this section is a **statistical summary** of data collected on units from different lots over a period of time and matrix samples. “Typical” represents the mean of the distribution at 25°C. “Max” or “min” represents (mean + 3σ) or (mean - 3σ) respectively, where σ is standard deviation, over the whole temperature range.

**FIGURE 13-1: TYPICAL IPD vs. VDD OVER TEMP (-40°C TO +25°C)**

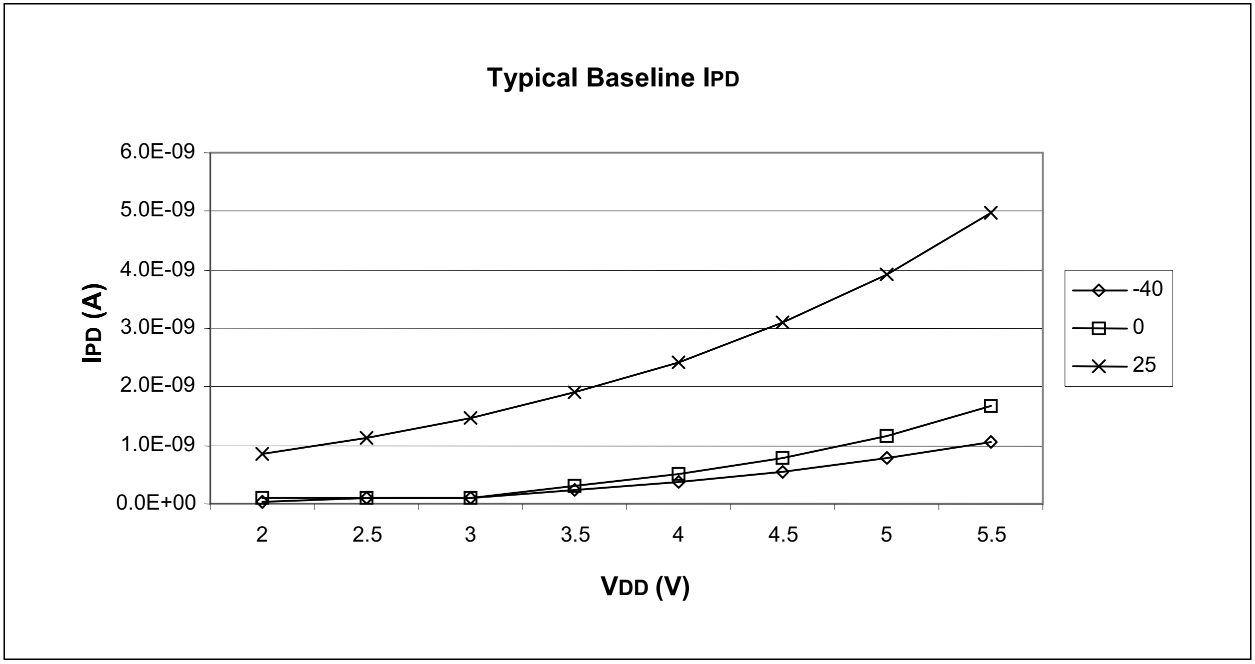


**FIGURE 13-2: TYPICAL IPD vs. VDD OVER TEMP (+85°C)**

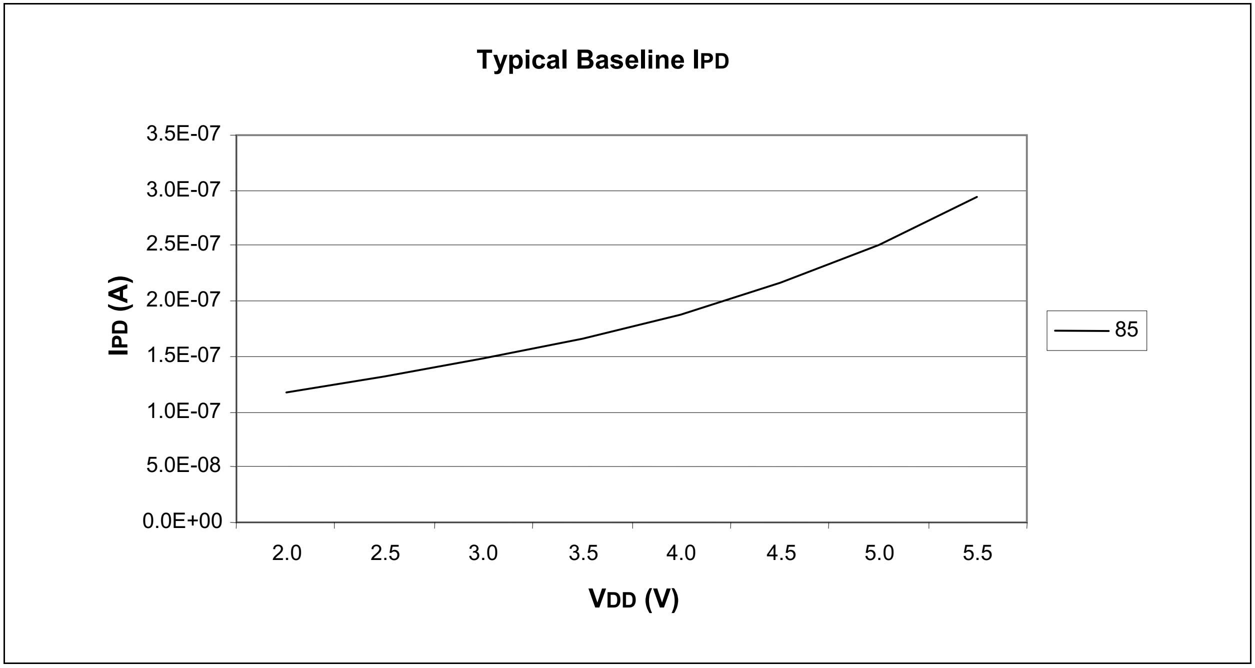


**FIGURE 13-3: TYPICAL IPD vs. VDD OVER TEMP (+125°C)**

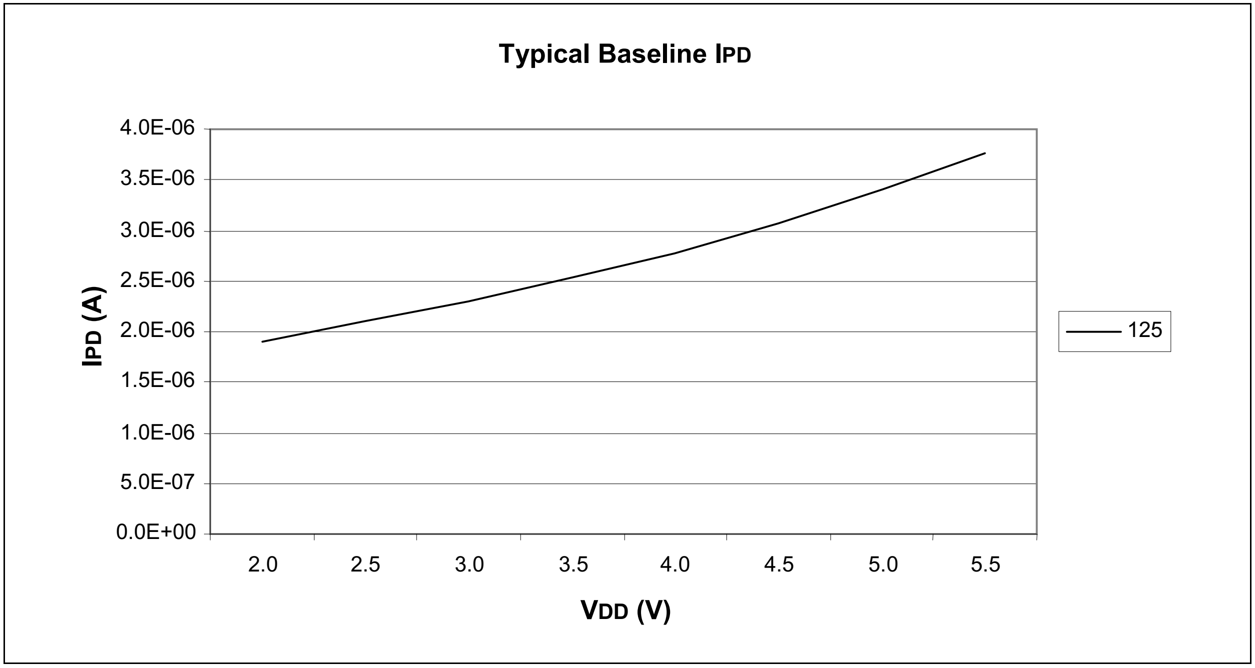


**FIGURE 13-4: MAXIMUM IPD vs. VDD OVER TEMP (-40°C TO +25°C)**

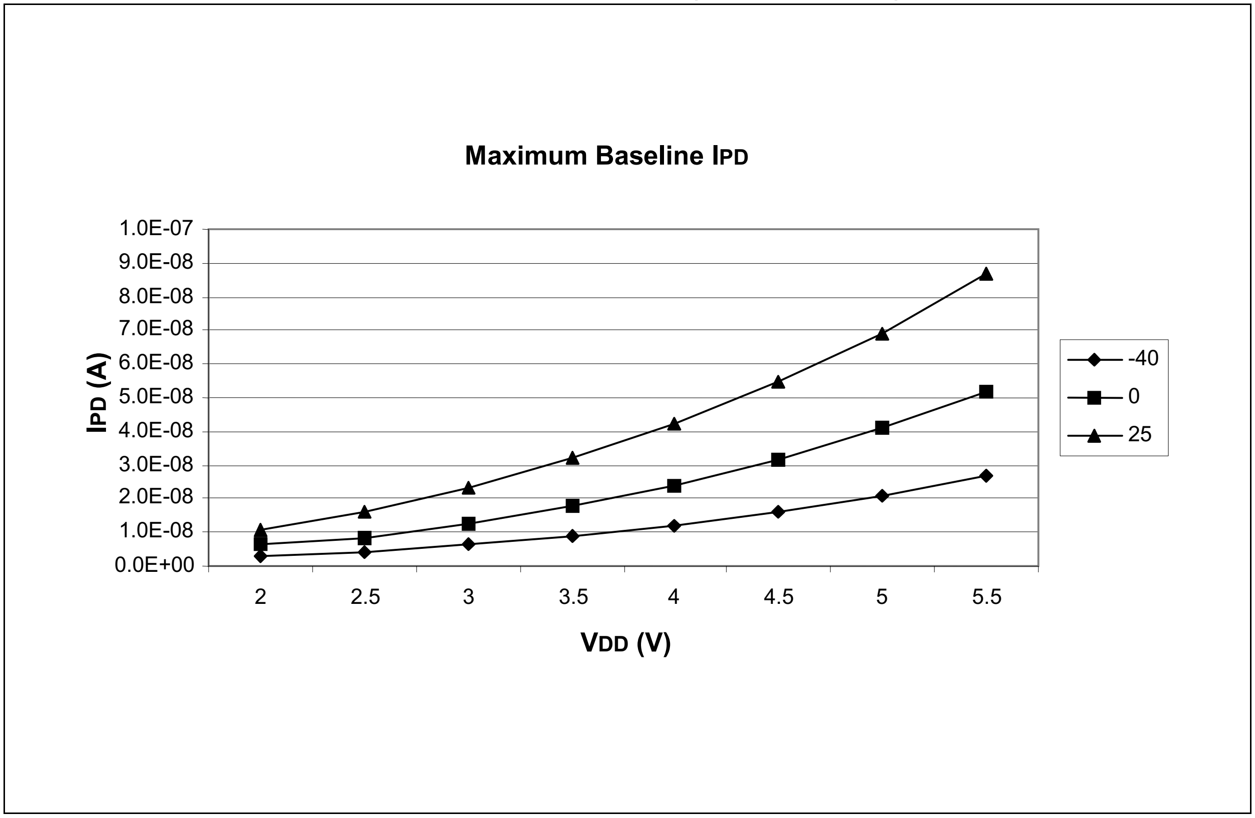


**FIGURE 13-5: MAXIMUM IPD vs. VDD OVER TEMP (+85°C)**

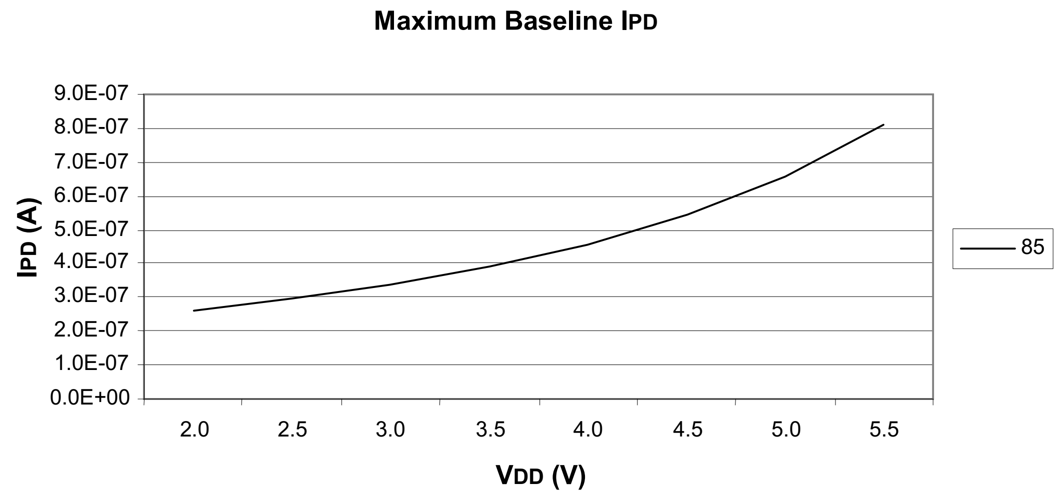


**FIGURE 13-6: MAXIMUM IPD vs. VDD OVER TEMP (+125°C)**

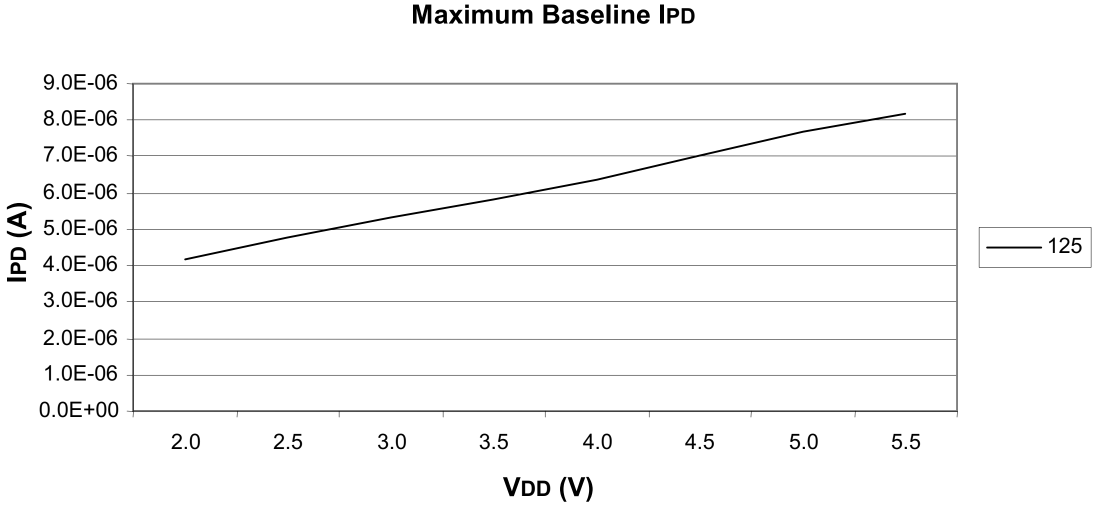


**FIGURE 13-7: TYPICAL IPD WITH BOD ENABLED vs. VDD OVER TEMP (-40°C TO +125°C)**

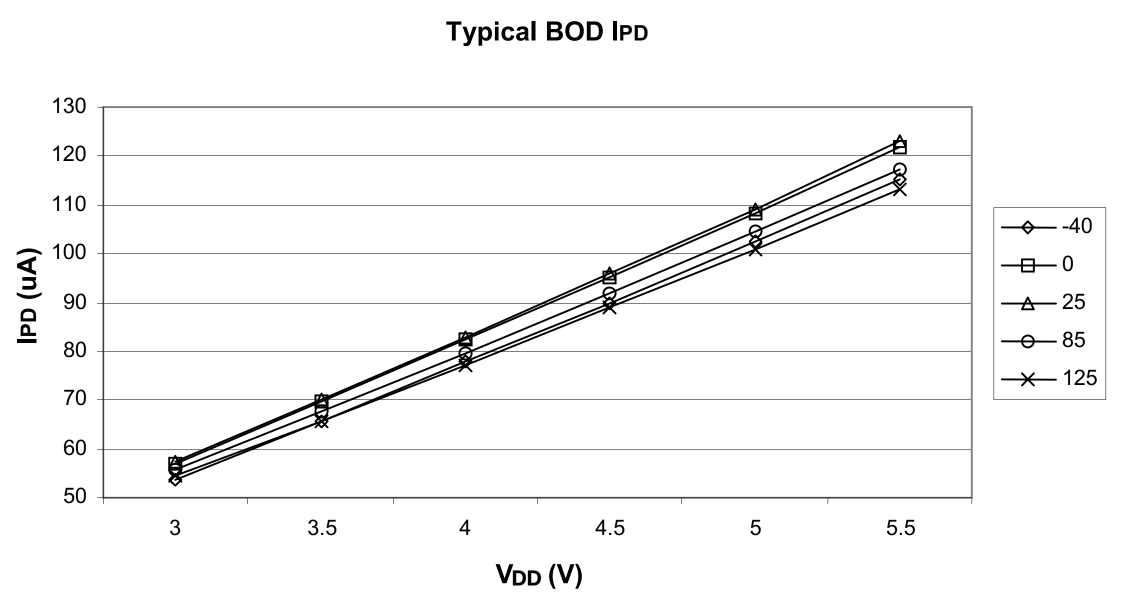


**FIGURE 13-8: TYPICAL IPD WITH CMP ENABLED vs. VDD OVER TEMP (-40°C TO +125°C)**

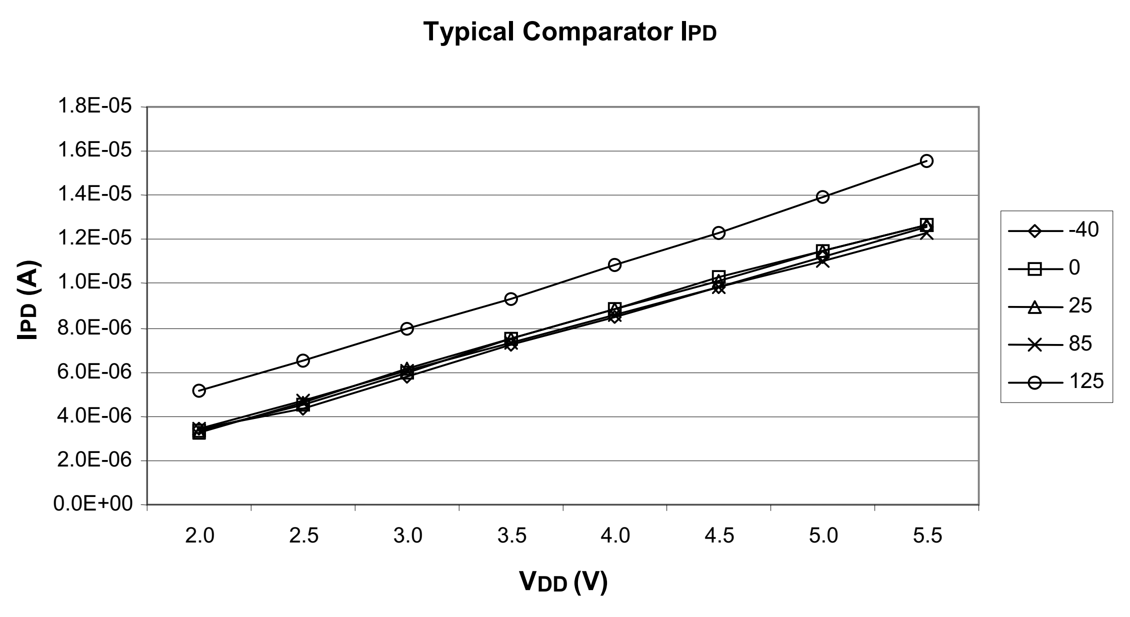


**FIGURE 13-9: TYPICAL IPD WITH A/D ENABLED vs. VDD OVER TEMP (-40°C TO +25°C)**

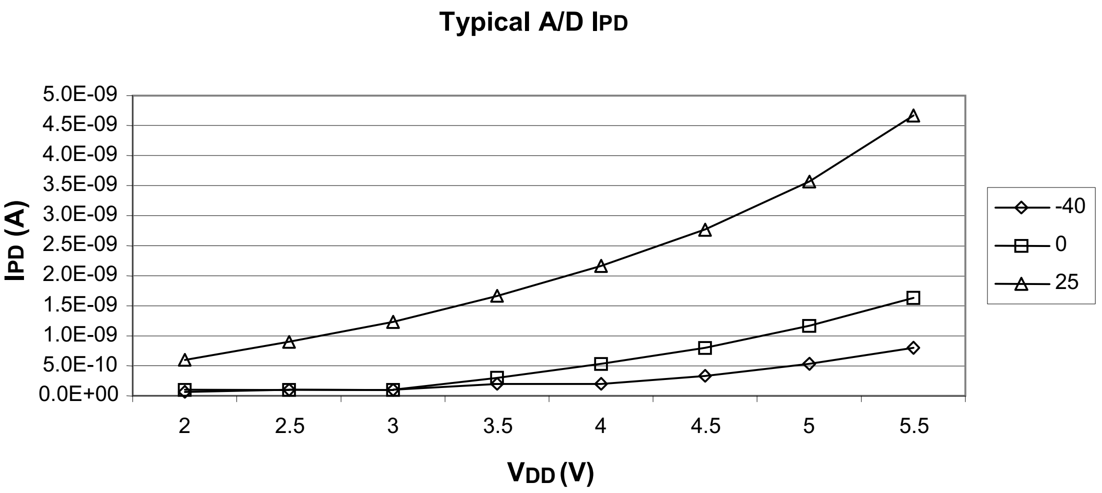


**FIGURE 13-10: TYPICAL IPD WITH A/D ENABLED vs. VDD OVER TEMP (+85°C)**

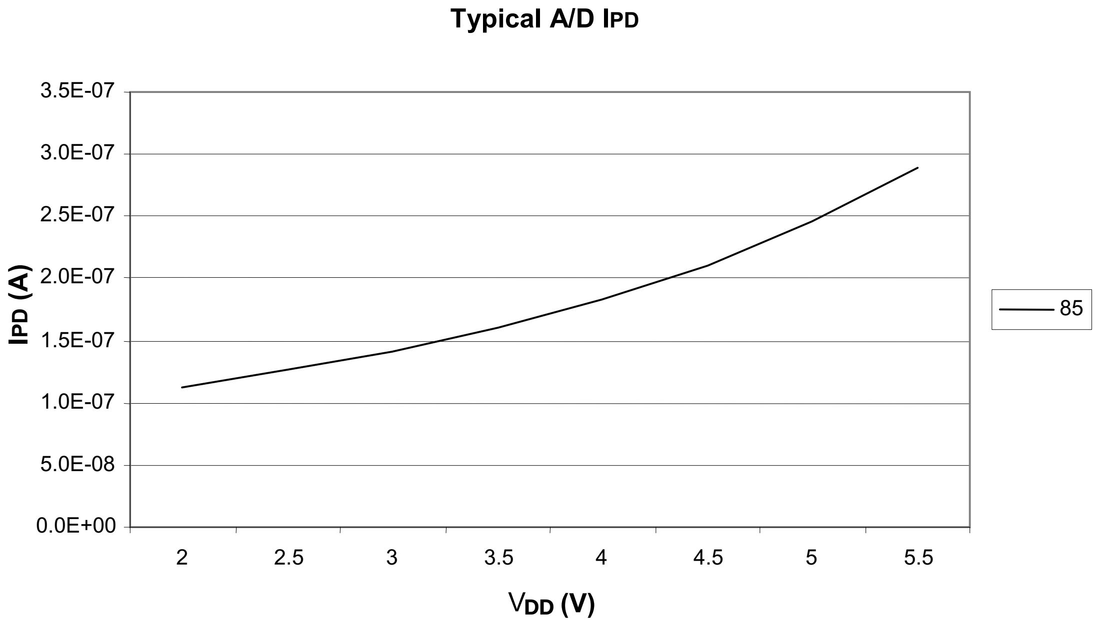


**FIGURE 13-11: TYPICAL IPD WITH A/D ENABLED vs. VDD OVER TEMP (+125°C)**

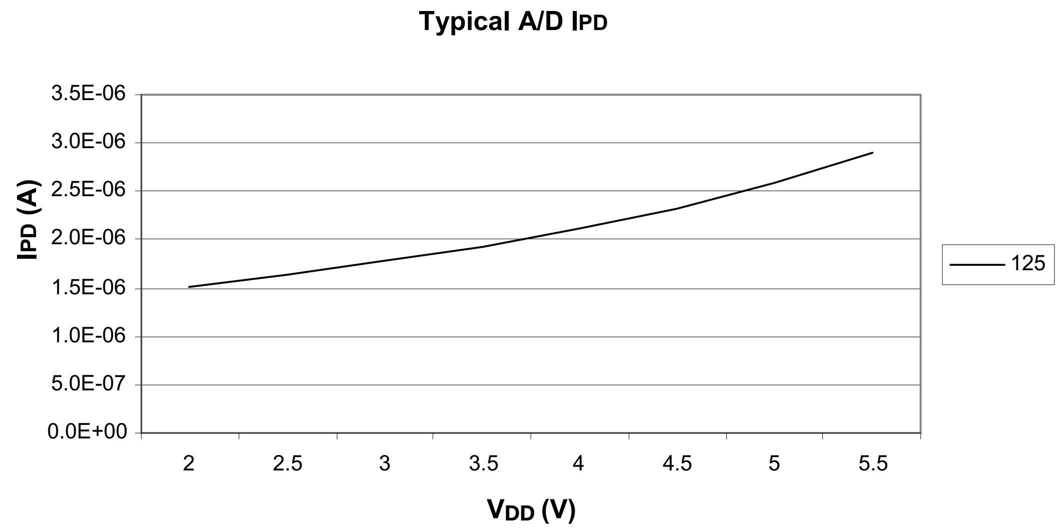


**FIGURE 13-12: TYPICAL IPD WITH T1 OSC ENABLED vs. VDD OVER TEMP (-40°C TO +125°C), 32 KHZ, C1 AND C2=50 pF)**

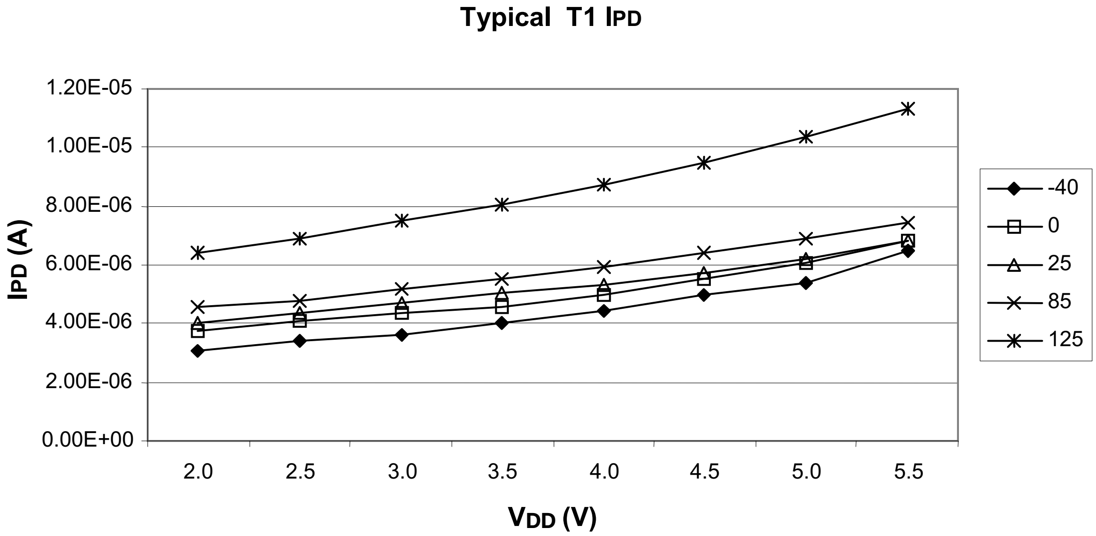


**FIGURE 13-13: TYPICAL IPD WITH CVREF ENABLED vs. VDD OVER TEMP (-40°C TO +125°C)**

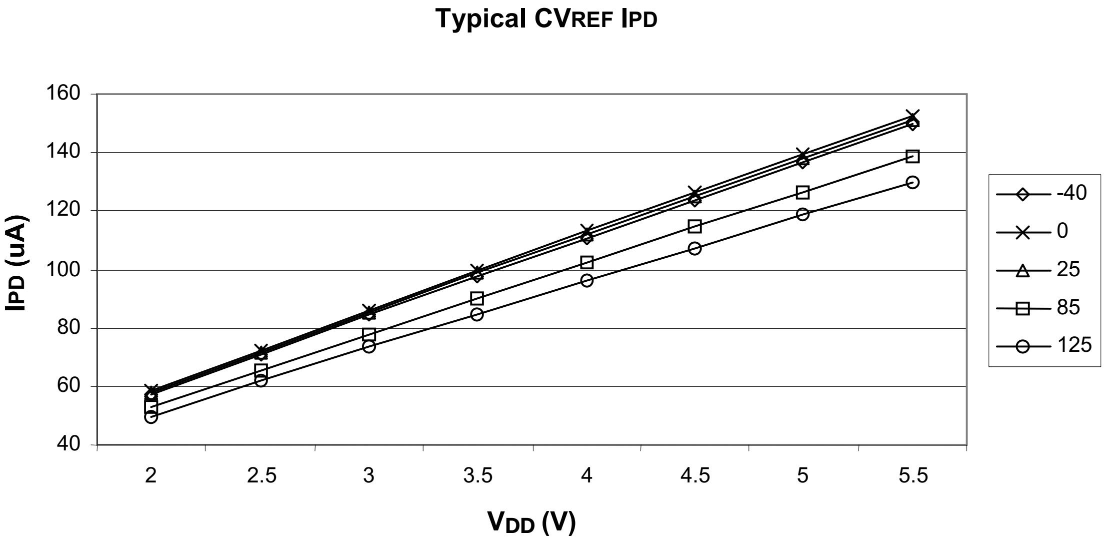


**FIGURE 13-14: TYPICAL IPD WITH WDT ENABLED vs. VDD OVER TEMP (-40°C TO +125°C)**

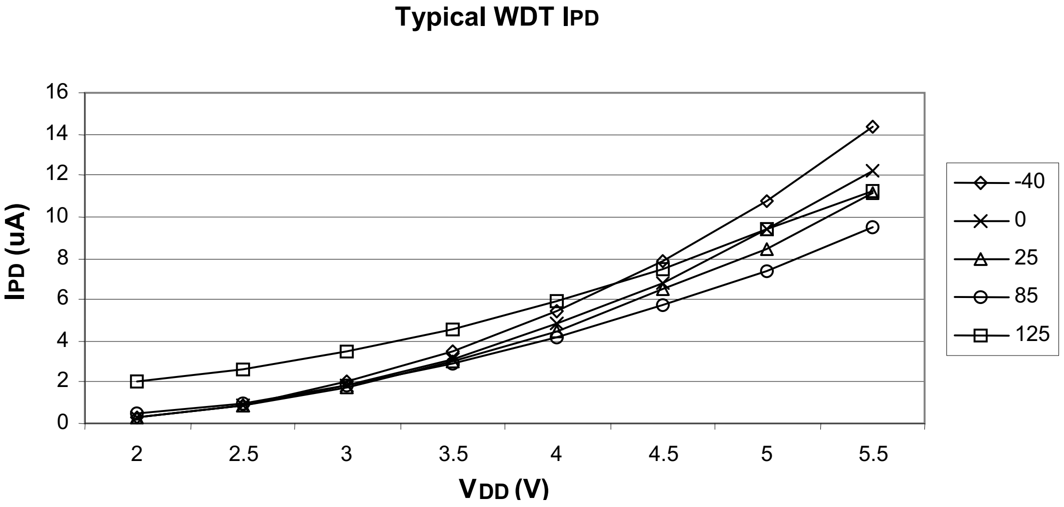


**FIGURE 13-15: MAXIMUM AND MINIMUM INTOSC FREQ vs. TEMPERATURE WITH 0.1µF AND 0.01µF DECOUPLING (VDD = 3.5V)**

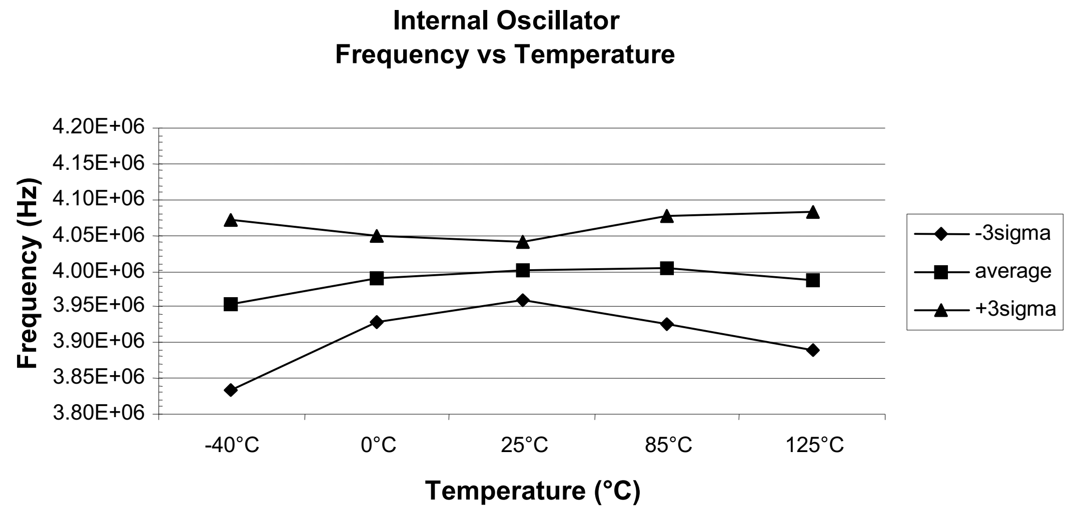


**FIGURE 13-16: MAXIMUM AND MINIMUM INTOSC FREQ vs. VDD WITH 0.1µF AND 0.01µF DECOUPLING (+25°C)**

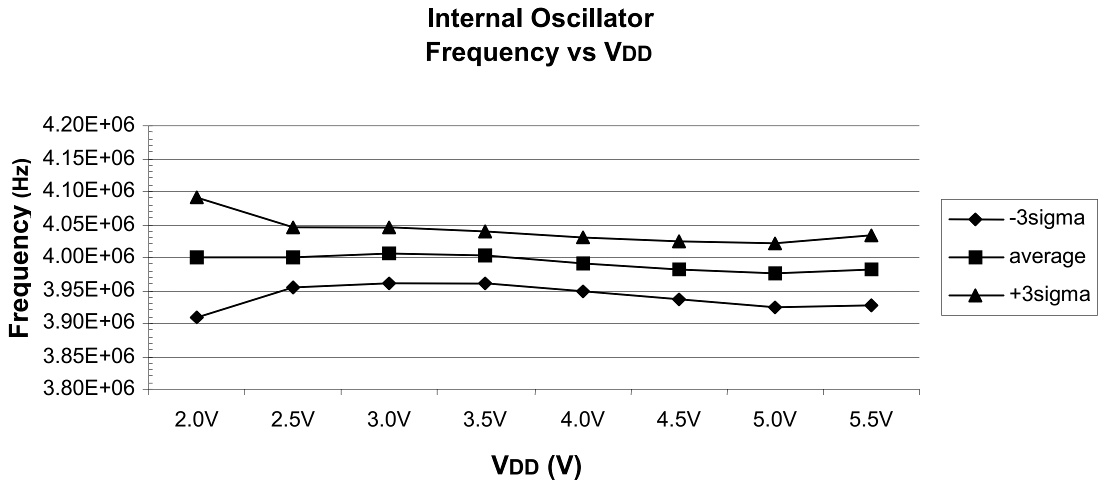
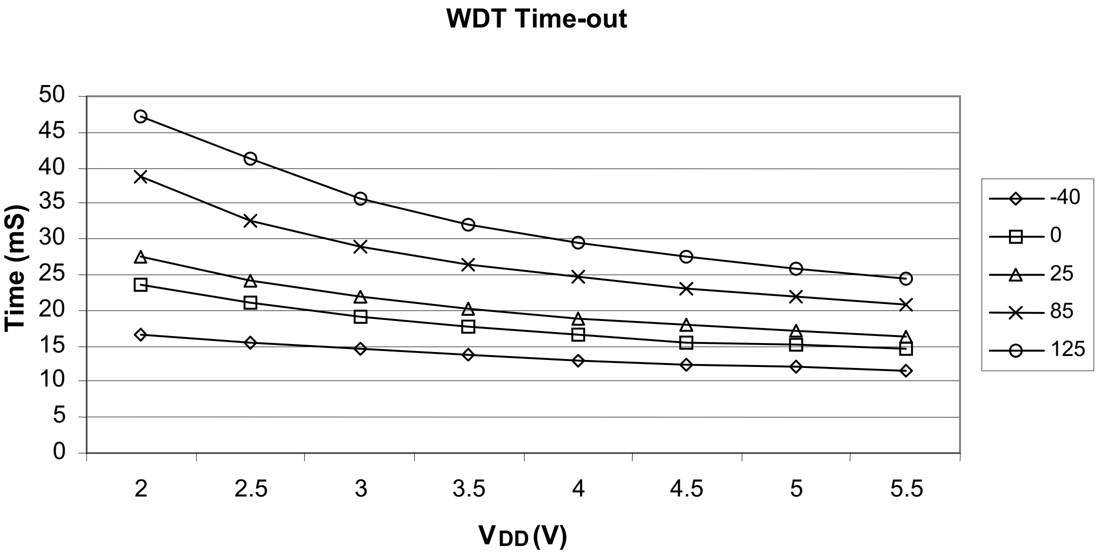


**FIGURE 13-17: TYPICAL WDT PERIOD vs. VDD (-40°C TO +125°C)**



**NOTES:**

# 14.0 PACKAGING INFORMATION

## 14.1 Package Marking Information

| | Example |
|---|---|
| 14-Lead PDIP (Skinny DIP)<br>XXXXXXXXXXXXXX<br>XXXXXXXXXXXXXX<br>○ YYWWNNN | 16F630-I e3<br>○ 1015/017 |
| 14-Lead SOIC<br>XXXXXXXXXXX<br>XXXXXXXXXXX<br>○ YYWWNNN | 16F630-E e3<br>○ 1015/017 |
| 14-Lead TSSOP<br>XXXXXXXX<br>YYWW<br>○ NNN | 16F630 e3<br>1015<br>○ 017 |

| **Legend:** | | |
|---|---|---|
| | XX...X | Customer-specific information |
| | Y | Year code (last digit of calendar year) |
| | YY | Year code (last 2 digits of calendar year) |
| | WW | Week code (week of January 1 is week ‘01’) |
| | NNN | Alphanumeric traceability code |
| | e3 | Pb-free JEDEC designator for Matte Tin (Sn) |
| | * | This package is Pb-free. The Pb-free JEDEC designator (e3) can be found on the outer packaging for this package. |

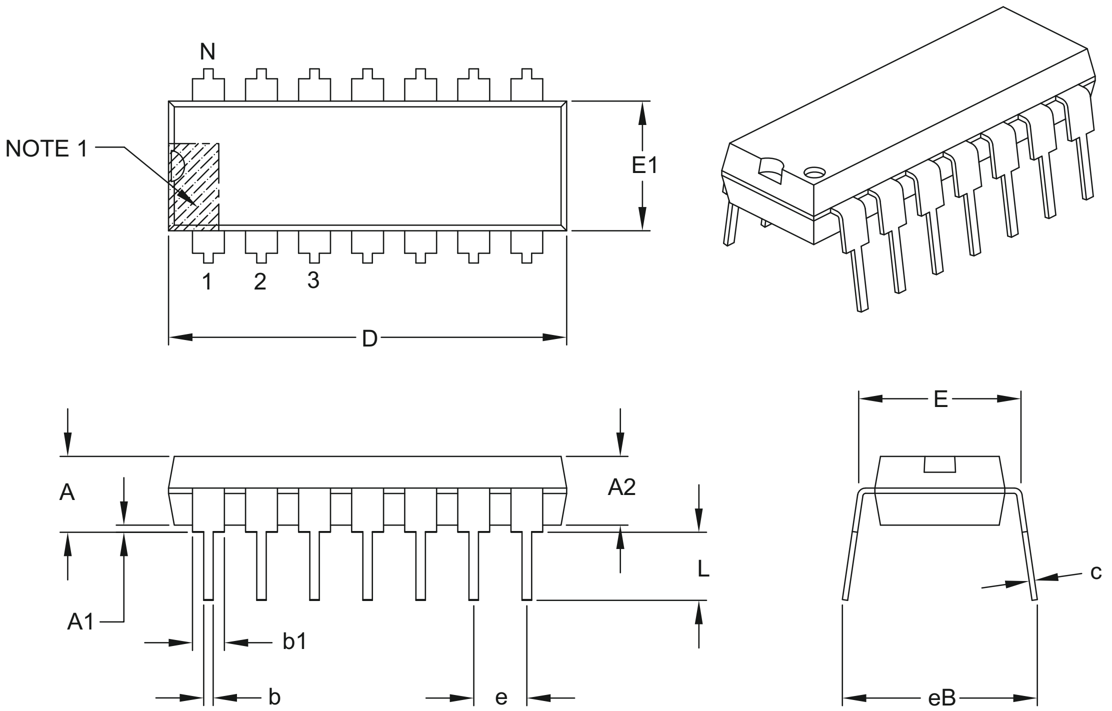
**Note**: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line, thus limiting the number of available characters for customer-specific information.

## 14.2 Package Details

The following sections give the technical details of the packages.

### 14-Lead Plastic Dual In-Line (P) – 300 mil Body [PDIP]

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



| Units | | INCHES | | |
|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX |
| Number of Pins | N | 14 | | |
| Pitch | e | .100 BSC | | |
| Top to Seating Plane | A | – | – | .210 |
| Molded Package Thickness | A2 | .115 | .130 | .195 |
| Base to Seating Plane | A1 | .015 | – | – |
| Shoulder to Shoulder Width | E | .290 | .310 | .325 |
| Molded Package Width | E1 | .240 | .250 | .280 |
| Overall Length | D | .735 | .750 | .775 |
| Tip to Seating Plane | L | .115 | .130 | .150 |
| Lead Thickness | c | .008 | .010 | .015 |
| Upper Lead Width | b1 | .045 | .060 | .070 |
| Lower Lead Width | b | .014 | .018 | .022 |
| Overall Row Spacing § | eB | – | – | .430 |

**Notes:**

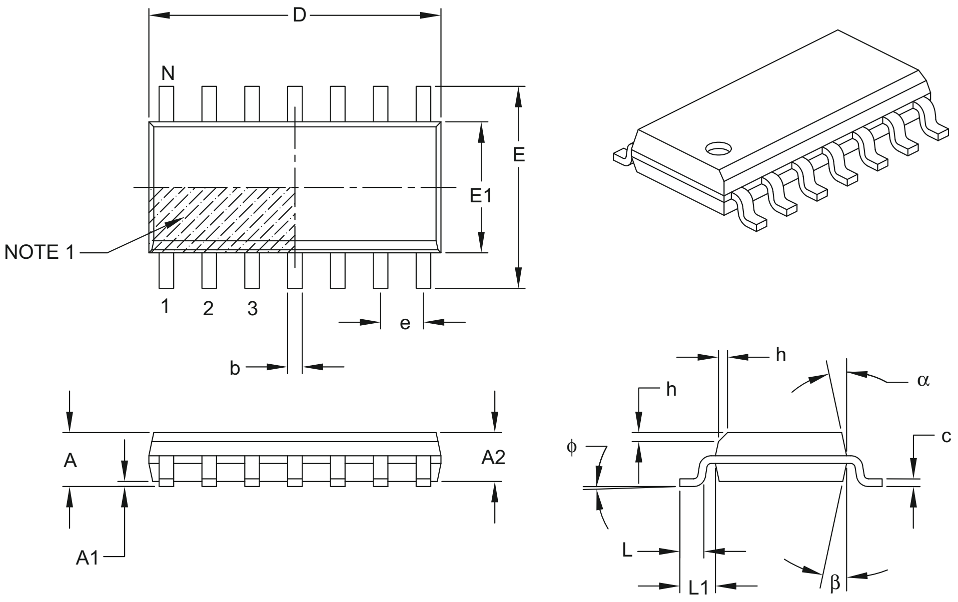
1. Pin 1 visual index feature may vary, but must be located with the hatched area.
2. § Significant Characteristic.
3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.
4. Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing C04-005B

## 14-Lead Plastic Small Outline (SL) – Narrow, 3.90 mm Body [SOIC]

| **Note:** | For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging |
|---|---|



| Units | | MILLIMETERS | | |
|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX |
| Number of Pins | N | 14 | | |
| Pitch | e | 1.27 BSC | | |
| Overall Height | A | – | – | 1.75 |
| Molded Package Thickness | A2 | 1.25 | – | – |
| Standoff § | A1 | 0.10 | – | 0.25 |
| Overall Width | E | 6.00 BSC | | |
| Molded Package Width | E1 | 3.90 BSC | | |
| Overall Length | D | 8.65 BSC | | |
| Chamfer (optional) | h | 0.25 | – | 0.50 |
| Foot Length | L | 0.40 | – | 1.27 |
| Footprint | L1 | 1.04 REF | | |
| Foot Angle | ϕ | 0° | – | 8° |
| Lead Thickness | c | 0.17 | – | 0.25 |
| Lead Width | b | 0.31 | – | 0.51 |
| Mold Draft Angle Top | α | 5° | – | 15° |
| Mold Draft Angle Bottom | β | 5° | – | 15° |

**Notes:**

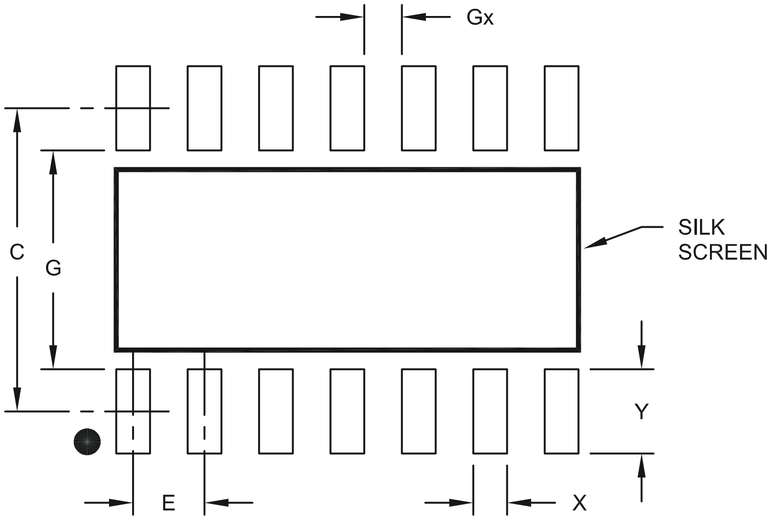
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. § Significant Characteristic.
3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.15 mm per side.
4. Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-065B

## 14-Lead Plastic Small Outline (SL) - Narrow, 3.90 mm Body [SOIC]

| **Note:** | For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging |
|---|---|



RECOMMENDED LAND PATTERN

| Units | | MILLIMETERS | | |
|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX |
| Contact Pitch | E | 1.27 BSC | | |
| Contact Pad Spacing | C | | 5.40 | |
| Contact Pad Width | X | | | 0.60 |
| Contact Pad Length | Y | | | 1.50 |
| Distance Between Pads | Gx | 0.67 | | |
| Distance Between Pads | G | 3.90 | | |

Notes:

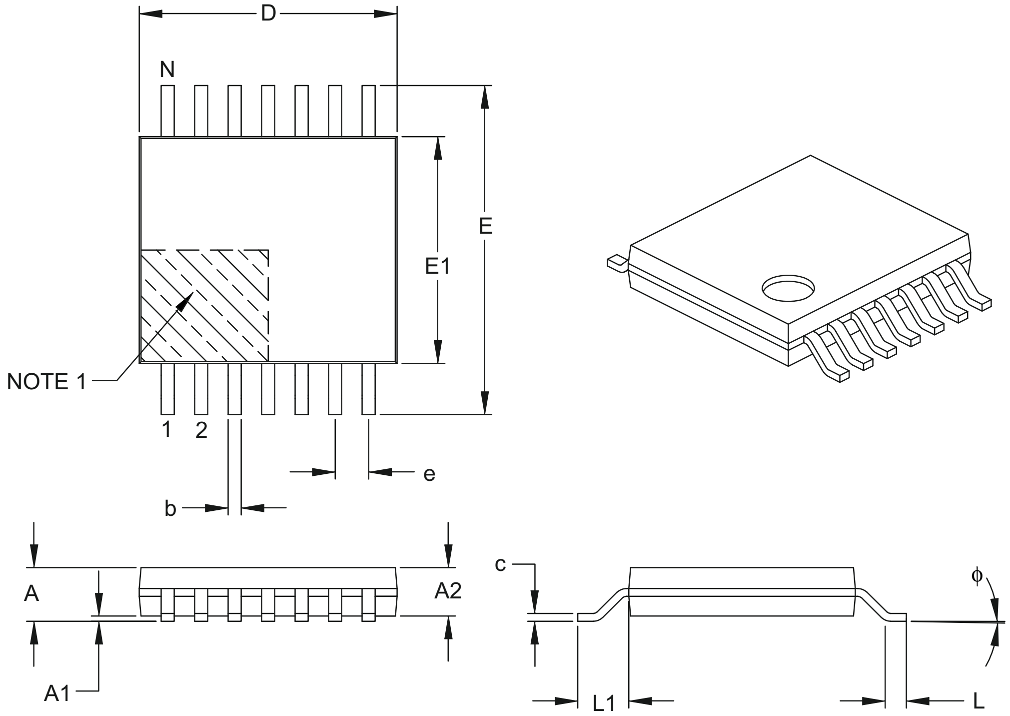
1. Dimensioning and tolerancing per ASME Y14.5M

   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2065A

## 14-Lead Plastic Thin Shrink Small Outline (ST) – 4.4 mm Body [TSSOP]

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



| | | MILLIMETERS | | |
|---|---|---|---|---|
| | Units | MILLIMETERS | | |
| | Dimension Limits | MIN | NOM | MAX |
| Number of Pins | N | 14 | | |
| Pitch | e | 0.65 BSC | | |
| Overall Height | A | – | – | 1.20 |
| Molded Package Thickness | A2 | 0.80 | 1.00 | 1.05 |
| Standoff | A1 | 0.05 | – | 0.15 |
| Overall Width | E | 6.40 BSC | | |
| Molded Package Width | E1 | 4.30 | 4.40 | 4.50 |
| Molded Package Length | D | 4.90 | 5.00 | 5.10 |
| Foot Length | L | 0.45 | 0.60 | 0.75 |
| Footprint | L1 | 1.00 REF | | |
| Foot Angle | φ | 0° | – | 8° |
| Lead Thickness | c | 0.09 | – | 0.20 |
| Lead Width | b | 0.19 | – | 0.30 |

**Notes:**

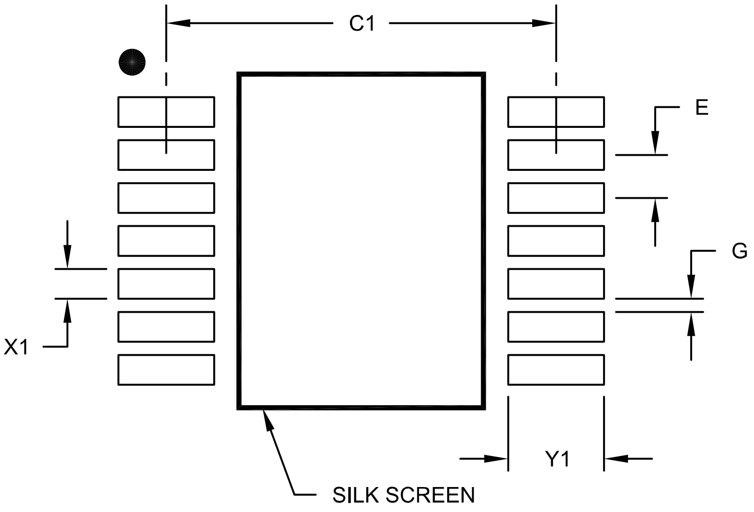
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.15 mm per side.
3. Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-087B

### 14-Lead Plastic Thin Shrink Small Outline (ST) - 4.4 mm Body [TSSOP]

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



RECOMMENDED LAND PATTERN

| | | Units | MILLIMETERS | |
|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX |
| Contact Pitch | E | 0.65 BSC | | |
| Contact Pad Spacing | C1 | | 5.90 | |
| Contact Pad Width (X14) | X1 | | | 0.45 |
| Contact Pad Length (X14) | Y1 | | | 1.45 |
| Distance Between Pads | G | 0.20 | | |

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2087A

## APPENDIX A: DATA SHEET REVISION HISTORY

### Revision A

This is a new data sheet.

### Revision B

Added characterization graphs.

Updated specifications.

Added notes to indicate Microchip programmers maintain all calibration bits to factory settings and the PIC16F676 ANSEL register must be initialized to configure pins as digital I/O.

### Revision C

### Revision D

Updated Package Drawings; Replaced PICmicro with PIC.

### Revision E (03/2007)

Replaced Package Drawings (Rev. AM); Replaced Development Support Section.

### Revision F (05/2010)

Replaced Package Drawings (Rev. BD); Replaced Development Support Section.

## APPENDIX B: DEVICE DIFFERENCES

The differences between the PIC16F630/676 devices listed in this data sheet are shown in Table B-1.

**TABLE B-1: DEVICE DIFFERENCES**

| Feature | PIC16F630 | PIC16F676 |
|---|---|---|
| A/D | No | Yes |

## APPENDIX C: DEVICE MIGRATIONS

This section is intended to describe the functional and electrical specification differences when migrating between functionally similar devices (such as from a PIC16C74A to a PIC16C74B).

**Not Applicable**

## APPENDIX D: MIGRATING FROM OTHER PIC® DEVICES

This discusses some of the issues in migrating from other PIC devices to the PIC16F6XX family of devices.

### D.1 PIC12C67X to PIC12F6XX

**TABLE 1: FEATURE COMPARISON**

| Feature | PIC12C67X | PIC16F6XX |
|---|---|---|
| Max Operating Speed | 10 MHz | 20 MHz |
| Max Program Memory | 2048 bytes | 1024 bytes |
| A/D Resolution | 8-bit | 10-bit |
| Data EEPROM | 16 bytes | 64 bytes |
| Oscillator Modes | 5 | 8 |
| Brown-out Detect | N | Y |
| Internal Pull-ups | RA0/1/3 | RA0/1/2/4/5 |
| Interrupt-on-change | RA0/1/3 | RA0/1/2/3/4/5 |
| Comparator | N | Y |

**Note:** This device has been designed to perform to the parameters of its data sheet. It has been tested to an electrical specification designed to determine its conformance with these parameters. Due to process differences in the manufacture of this device, this device may have different performance characteristics than its earlier version. These differences may cause this device to perform differently in your application than the earlier version of this device.

## INDEX

NOTES:

## THE MICROCHIP WEB SITE

Microchip provides online support via our WWW site at www.microchip.com. This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user’s guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

## CUSTOMER CHANGE NOTIFICATION SERVICE

Microchip’s customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip web site at www.microchip.com, click on Customer Change Notification and follow the registration instructions.

## CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support
- Development Systems Information Line

Customers should contact their distributor, representative or field application engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

**Technical support is available through the web site at: http://support.microchip.com**

## READER RESPONSE

It is our intention to provide you with the best documentation possible to ensure successful use of your Microchip product. If you wish to provide your comments on organization, clarity, subject matter, and ways in which our documentation can better serve you, please FAX your comments to the Technical Publications Manager at (480) 792-4150.

Please list the following information, and use this outline to provide us with your comments about this document.

To: Technical Publications Manager Total Pages Sent ________

RE: Reader Response

From: Name ____________________________________________

Company ____________________________________________

Address ____________________________________________

City / State / ZIP / Country ____________________________________________

Telephone: (_______) _________ - _________ FAX: (______) _________ - _________

Application (optional):

Would you like a reply?___Y ___N

Device: PIC16F630/676 Literature Number: DS40039F

Questions:

1. What are the best features of this document?

____________________________________________

____________________________________________

2. How does this document meet your hardware and software development needs?

____________________________________________

____________________________________________

3. Do you find the organization of this document easy to follow? If not, why?

____________________________________________

____________________________________________

4. What additions to the document do you think would enhance the structure and subject?

____________________________________________

____________________________________________

5. What deletions from the document could be made without affecting the overall usefulness?

____________________________________________

____________________________________________

6. Is there any incorrect or misleading information (what and where)?

____________________________________________

____________________________________________

7. How would you improve this document?

____________________________________________

____________________________________________

## PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

| PART NO. | X | /XX | XXX |
|---|---|---|---|
| Device | Temperature Range | Package | Pattern |

| | | | |
|---|---|---|---|
| **Device:** | : Standard $V_{DD}$ range<br>T: (Tape and Reel) | | |
| **Temperature Range:** | I<br>E | =<br>= | -40°C to +85°C<br>-40°C to +125°C |
| **Package:** | P<br>SL<br>ST | =<br>=<br>= | PDIP<br>SOIC (Gull wing, 3.90 mm body)<br>TSSOP(4.4 mm) |
| **Pattern:** | 3-Digit Pattern Code for QTP (blank otherwise) | | |

**Examples:**

a) PIC16F630 – E/P 301 = Extended Temp., PDIP package, 20 MHz, QTP pattern #301

b) PIC16F676 – I/SL = Industrial Temp., SOIC package, 20 MHz

* JW Devices are UV erasable and can be programmed to any device configuration. JW Devices meet the electrical requirement of each oscillator type.

# WORLDWIDE SALES AND SERVICE

## AMERICAS

**Corporate Office**
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://support.microchip.com
Web Address:
www.microchip.com

**Atlanta**
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

**Boston**
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

**Chicago**
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

**Cleveland**
Independence, OH
Tel: 216-447-0464
Fax: 216-447-0643

**Dallas**
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

**Detroit**
Farmington Hills, MI
Tel: 248-538-2250
Fax: 248-538-2260

**Kokomo**
Kokomo, IN
Tel: 765-864-8360
Fax: 765-864-8387

**Los Angeles**
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

**Santa Clara**
Santa Clara, CA
Tel: 408-961-6444
Fax: 408-961-6445

**Toronto**
Mississauga, Ontario,
Canada
Tel: 905-673-0699
Fax: 905-673-6509

## ASIA/PACIFIC

**Asia Pacific Office**
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon
Hong Kong
Tel: 852-2401-1200
Fax: 852-2401-3431

**Australia - Sydney**
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

**China - Beijing**
Tel: 86-10-8528-2100
Fax: 86-10-8528-2104

**China - Chengdu**
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

**China - Chongqing**
Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

**China - Hong Kong SAR**
Tel: 852-2401-1200
Fax: 852-2401-3431

**China - Nanjing**
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

**China - Qingdao**
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

**China - Shanghai**
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

**China - Shenyang**
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

**China - Shenzhen**
Tel: 86-755-8203-2660
Fax: 86-755-8203-1760

**China - Wuhan**
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

**China - Xian**
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

**China - Xiamen**
Tel: 86-592-2388138
Fax: 86-592-2388130

**China - Zhuhai**
Tel: 86-756-3210040
Fax: 86-756-3210049

## ASIA/PACIFIC

**India - Bangalore**
Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

**India - New Delhi**
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

**India - Pune**
Tel: 91-20-2566-1512
Fax: 91-20-2566-1513

**Japan - Yokohama**
Tel: 81-45-471- 6166
Fax: 81-45-471-6122

**Korea - Daegu**
Tel: 82-53-744-4301
Fax: 82-53-744-4302

**Korea - Seoul**
Tel: 82-2-554-7200
Fax: 82-2-558-5932 or
82-2-558-5934

**Malaysia - Kuala Lumpur**
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

**Malaysia - Penang**
Tel: 60-4-227-8870
Fax: 60-4-227-4068

**Philippines - Manila**
Tel: 63-2-634-9065
Fax: 63-2-634-9069

**Singapore**
Tel: 65-6334-8870
Fax: 65-6334-8850

**Taiwan - Hsin Chu**
Tel: 886-3-6578-300
Fax: 886-3-6578-370

**Taiwan - Kaohsiung**
Tel: 886-7-536-4818
Fax: 886-7-536-4803

**Taiwan - Taipei**
Tel: 886-2-2500-6610
Fax: 886-2-2508-0102

**Thailand - Bangkok**
Tel: 66-2-694-1351
Fax: 66-2-694-1350

## EUROPE

**Austria - Wels**
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

**Denmark - Copenhagen**
Tel: 45-4450-2828
Fax: 45-4485-2829

**France - Paris**
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

**Germany - Munich**
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

**Italy - Milan**
Tel: 39-0331-742611
Fax: 39-0331-466781

**Netherlands - Drunen**
Tel: 31-416-690399
Fax: 31-416-690340

**Spain - Madrid**
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

**UK - Wokingham**
Tel: 44-118-921-5869
Fax: 44-118-921-5820

01/05/10