PIC16F684
Data Sheet

14-Pin Flash-Based, 8-Bit CMOS Microcontrollers with nanoWatt Technology
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.

Trademark

The Microchip name and logo, the Microchip logo, Accuron, dsPIC, KeeLog, microID, MPLAB, PIC, PICmicro, PICSTART, PRO MATE, PowerSmart, rfPIC and SmartShunt are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

AmpLab, FilterLab, Migratable Memory, MXDEV, MXLAB, SEEVAL, SmartSensor 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, dsPICDEM, dsPICDEM.net, dsPICworks, ECAN, ECONOMONITOR, FanSense, FlexROM, fuzzyLAB, In-Circuit Serial Programming, IGSP, ICEPIC, Linear Active Thermistor, MPASM, MPLIB, MPLINK, MPSIM, PICkit, PICDEM, PICDEM.net, PICLAB, PICtail, PowerCal, PowerInfo, PowerMate, PowerTool, REAL ICE, rfLAB, rfPICDEM, Select Mode, Smart Serial, SmartTel, Total Endurance, UNI/O, 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.

© 2006, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

Printed on recycled paper.
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:
- Precision Internal Oscillator:
  - Factory calibrated to ±1%, typical
  - Software selectable frequency range of
    8 MHz to 125 kHz
  - Software tunable
  - Two-Speed Start-up mode
  - Crystal fail detect for critical applications
  - Clock mode switching during operation for
    power savings
- Software Selectable 31 kHz Internal Oscillator
- Power-Saving Sleep mode
- Wide operating voltage range (2.0V-5.5V)
- Industrial and Extended Temperature range
- Power-on Reset (POR)
- Power-up Timer (PWRT) and Oscillator Start-up
  Timer (OST)
- Brown-out Reset (BOR) with software control
  option
- Enhanced low-current Watchdog Timer (WDT)
  with on-chip oscillator (software selectable
  nominal 268 seconds with full prescaler) with
  software enable
- Multiplexed Master Clear with pull-up/input pin
- 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:
  - 50 nA @ 2.0V, typical
- Operating Current:
  - 11 µA @ 32 kHz, 2.0V, typical
  - 220 µA @ 4 MHz, 2.0V, typical
- Watchdog Timer Current:
  - 1 µA @ 2.0V, typical

Peripheral Features:
- 12 I/O pins with individual direction control:
  - High current source/sink for direct LED drive
  - Interrupt-on-change pin
  - Individually programmable weak pull-ups
  - Ultra Low-Power Wake-Up (ULPWU)
- Analog Comparator module with:
  - Two analog comparators
  - Programmable on-chip voltage reference
    (CVref) module (% of VDD)
  - Comparator inputs and outputs externally
    accessible
- A/D Converter:
  - 10-bit resolution and 8 channels
- Timer0: 8-bit timer/counter with 8-bit
  programmable prescaler
- Enhanced Timer1:
  - 16-bit timer/counter with prescaler
  - External Timer1 Gate (count enable)
  - Option to use OSC1 and OSC2 in LP mode
    as Timer1 oscillator if INTOSC mode
    selected
- Timer2: 8-bit timer/counter with 8-bit period
  register, prescaler and postscaler
- Enhanced Capture, Compare, PWM module:
  - 16-bit Capture, max resolution 12.5 ns
  - Compare, max resolution 200 ns
  - 10-bit PWM with 1, 2 or 4 output channels,
    programmable “dead time”, max frequency
    20 kHz
- In-Circuit Serial Programming™ (ICSP™) via two
  pins
### TABLE 1: DUAL IN-LINE PIN SUMMARY

<table>
<thead>
<tr>
<th>I/O</th>
<th>Pin</th>
<th>Analog</th>
<th>Comparators</th>
<th>Timer</th>
<th>CCP</th>
<th>Interrupts</th>
<th>Pull-ups</th>
<th>Basic</th>
</tr>
</thead>
<tbody>
<tr>
<td>RA0</td>
<td>13</td>
<td>AN0</td>
<td>C1IN+</td>
<td>—</td>
<td>—</td>
<td>IOC</td>
<td>Y</td>
<td>ICSPDAT/ULPWU</td>
</tr>
<tr>
<td>RA1</td>
<td>12</td>
<td>AN1/REF</td>
<td>C1IN-</td>
<td>—</td>
<td>—</td>
<td>IOC</td>
<td>Y</td>
<td>ICSPCLK</td>
</tr>
<tr>
<td>RA2</td>
<td>11</td>
<td>AN2</td>
<td>C1OUT</td>
<td>T0CKI</td>
<td>—</td>
<td>INT/IOC</td>
<td>Y</td>
<td>—</td>
</tr>
<tr>
<td>RA3(1)</td>
<td>4</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>IOC</td>
<td>Y(2)</td>
<td>MCLR/VPp</td>
</tr>
<tr>
<td>RA4</td>
<td>3</td>
<td>AN3</td>
<td>—</td>
<td>TTG</td>
<td>—</td>
<td>IOC</td>
<td>Y</td>
<td>OSC2/CLKOUT</td>
</tr>
<tr>
<td>RA5</td>
<td>2</td>
<td>—</td>
<td>—</td>
<td>T1CKI</td>
<td>—</td>
<td>IOC</td>
<td>Y</td>
<td>OSC1/CLKIN</td>
</tr>
<tr>
<td>RC0</td>
<td>10</td>
<td>AN4</td>
<td>C2IN+</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>RC1</td>
<td>9</td>
<td>AN5</td>
<td>C2IN-</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>RC2</td>
<td>8</td>
<td>AN6</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>P1D</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>RC3</td>
<td>7</td>
<td>AN7</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>P1C</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>RC4</td>
<td>6</td>
<td>—</td>
<td>C2OUT</td>
<td>—</td>
<td>—</td>
<td>P1B</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>RC5</td>
<td>5</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>CCP1/P1A</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>—</td>
<td>1</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>VDD</td>
</tr>
<tr>
<td>—</td>
<td>14</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>VSS</td>
</tr>
</tbody>
</table>

**Note 1:** Input only.

**2:** Only when pin is configured for external MCLR.
### TABLE 2: QFN PIN SUMMARY

<table>
<thead>
<tr>
<th>I/O</th>
<th>Pin</th>
<th>Analog</th>
<th>Comparators</th>
<th>Timers</th>
<th>CCP</th>
<th>Interrupts</th>
<th>Pull-ups</th>
<th>Basic</th>
</tr>
</thead>
<tbody>
<tr>
<td>RA0</td>
<td>12</td>
<td>AN0</td>
<td>C1IN+</td>
<td>—</td>
<td>—</td>
<td>IOC</td>
<td>Y</td>
<td>ICSPDAT/ULPWU</td>
</tr>
<tr>
<td>RA1</td>
<td>11</td>
<td>AN1/VREF</td>
<td>C1IN-</td>
<td>—</td>
<td>—</td>
<td>IOC</td>
<td>Y</td>
<td>ICSPCLK</td>
</tr>
<tr>
<td>RA2</td>
<td>10</td>
<td>AN2</td>
<td>C1OUT</td>
<td>T0CKI</td>
<td>—</td>
<td>INT/IOC</td>
<td>Y</td>
<td>—</td>
</tr>
<tr>
<td>RA3(1)</td>
<td>3</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>IOC</td>
<td>Y(2)</td>
<td>MCLR/VPP</td>
</tr>
<tr>
<td>RA4</td>
<td>2</td>
<td>AN3</td>
<td>TTG</td>
<td>—</td>
<td>IOC</td>
<td>Y</td>
<td>OSC2/CLKOUT</td>
<td>—</td>
</tr>
<tr>
<td>RA5</td>
<td>1</td>
<td>—</td>
<td>T1CKI</td>
<td>—</td>
<td>IOC</td>
<td>Y</td>
<td>OSC1/CLKIN</td>
<td>—</td>
</tr>
<tr>
<td>RC0</td>
<td>9</td>
<td>AN4</td>
<td>C2IN+</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>RC1</td>
<td>8</td>
<td>AN5</td>
<td>C2IN-</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>RC2</td>
<td>7</td>
<td>AN6</td>
<td>—</td>
<td>—</td>
<td>P1D</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>RC3</td>
<td>6</td>
<td>AN7</td>
<td>—</td>
<td>—</td>
<td>P1C</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>RC4</td>
<td>5</td>
<td>—</td>
<td>C2OUT</td>
<td>—</td>
<td>P1B</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>RC5</td>
<td>4</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>CCP1/P1A</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>—</td>
<td>16</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>VDD</td>
<td>—</td>
</tr>
<tr>
<td>—</td>
<td>13</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>VSS</td>
<td>—</td>
</tr>
</tbody>
</table>

**Note 1:** Input only.

**Note 2:** Only when pin is configured for external MCLR.
Table of Contents

1.0 Device Overview ........................................................................................................ 5
2.0 Memory Organization ................................................................................................. 7
3.0 Oscillator Module (With Fail-Safe Clock Monitor) .................................................... 19
4.0 I/O Ports .................................................................................................................... 31
5.0 Timer0 Module .......................................................................................................... 43
6.0 Timer1 Module with Gate Control ............................................................................ 47
7.0 Timer2 Module .......................................................................................................... 53
8.0 Comparator Module .................................................................................................. 55
9.0 Analog-to-Digital Converter (ADC) Module ............................................................. 65
10.0 Data EEPROM Memory ........................................................................................... 75
11.0 Enhanced Capture/Compare/PWM (With Auto-Shutdown and Dead Band) Module ................................................................................................................ 79
12.0 Special Features of the CPU ................................................................................... 97
13.0 Instruction Set Summary .......................................................................................... 115
14.0 Development Support .............................................................................................. 125
15.0 Electrical Specifications .......................................................................................... 129
16.0 DC and AC Characteristics Graphs and Tables ....................................................... 151
17.0 Packaging Information ............................................................................................ 173
Appendix A: Data Sheet Revision History ...................................................................... 179
Appendix B: Migrating from other PICmicro® Devices .................................................. 179
Index ................................................................................................................................ 181
The Microchip Web Site ................................................................................................. 187
Customer Change Notification Service ......................................................................... 187
Customer Support ......................................................................................................... 187
Reader Response ............................................................................................................. 188
Product Identification System ......................................................................................... 189

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.
1.0 DEVICE OVERVIEW

The PIC16F684 is covered by this data sheet. It is available in 14-pin PDIP, SOIC, TSSOP and 16-pin QFN packages. Figure 1-1 shows a block diagram of the PIC16F684 device. Table 1-1 shows the pinout description.

FIGURE 1-1: PIC16F684 BLOCK DIAGRAM

[Block diagram of PIC16F684 device]
<table>
<thead>
<tr>
<th>Name</th>
<th>Function</th>
<th>Input Type</th>
<th>Output Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RA0/AN0/C1IN+/ICSPDAT/ULPWU</td>
<td>RA0</td>
<td>TTL</td>
<td>CMOS</td>
<td>PORTA I/O with prog. pull-up and interrupt-on-change</td>
</tr>
<tr>
<td>AN0</td>
<td>AN</td>
<td>—</td>
<td>—</td>
<td>— A/D Channel 0 input</td>
</tr>
<tr>
<td>C1IN+</td>
<td>AN</td>
<td>—</td>
<td>—</td>
<td>— Comparator 1 non-inverting input</td>
</tr>
<tr>
<td>ICSPDAT</td>
<td>TTL</td>
<td>CMOS</td>
<td>—</td>
<td>— Serial Programming Data I/O</td>
</tr>
<tr>
<td>ULPWU</td>
<td>AN</td>
<td>—</td>
<td>—</td>
<td>— Ultra-Low-Power Wake-Up input</td>
</tr>
<tr>
<td>RA1/AN1/C1IN-/VREF/ICSPCLK</td>
<td>RA1</td>
<td>TTL</td>
<td>CMOS</td>
<td>PORTA I/O with prog. pull-up and interrupt-on-change</td>
</tr>
<tr>
<td>AN1</td>
<td>AN</td>
<td>—</td>
<td>—</td>
<td>— A/D Channel 1 input</td>
</tr>
<tr>
<td>C1IN-</td>
<td>AN</td>
<td>—</td>
<td>—</td>
<td>— Comparator 1 inverting input</td>
</tr>
<tr>
<td>VREF</td>
<td>AN</td>
<td>—</td>
<td>—</td>
<td>— External Voltage Reference for A/D</td>
</tr>
<tr>
<td>ICSPCLK</td>
<td>ST</td>
<td>—</td>
<td>—</td>
<td>— Serial Programming Clock</td>
</tr>
<tr>
<td>RA2/AN2/T0CKI/INT/C1OUT</td>
<td>RA2</td>
<td>ST</td>
<td>CMOS</td>
<td>PORTA I/O with prog. pull-up and interrupt-on-change</td>
</tr>
<tr>
<td>AN2</td>
<td>AN</td>
<td>—</td>
<td>—</td>
<td>— A/D Channel 2 input</td>
</tr>
<tr>
<td>T0CKI</td>
<td>ST</td>
<td>—</td>
<td>—</td>
<td>— Timer0 clock input</td>
</tr>
<tr>
<td>INT</td>
<td>ST</td>
<td>—</td>
<td>—</td>
<td>— External Interrupt</td>
</tr>
<tr>
<td>C1OUT</td>
<td>—</td>
<td>CMOS</td>
<td>—</td>
<td>— Comparator 1 output</td>
</tr>
<tr>
<td>RA3/MCLR/VPP</td>
<td>RA3</td>
<td>TTL</td>
<td>—</td>
<td>— PORTA input with interrupt-on-change</td>
</tr>
<tr>
<td>MCLR</td>
<td>ST</td>
<td>—</td>
<td>—</td>
<td>— Master Clear w/internal pull-up</td>
</tr>
<tr>
<td>VPP</td>
<td>HV</td>
<td>—</td>
<td>—</td>
<td>— Programming voltage</td>
</tr>
<tr>
<td>RA4/AN3/T1G/OSC2/CLKOUT</td>
<td>RA4</td>
<td>TTL</td>
<td>CMOS</td>
<td>PORTA I/O with prog. pull-up and interrupt-on-change</td>
</tr>
<tr>
<td>AN3</td>
<td>AN</td>
<td>—</td>
<td>—</td>
<td>— A/D Channel 3 input</td>
</tr>
<tr>
<td>T1G</td>
<td>ST</td>
<td>—</td>
<td>—</td>
<td>— Timer1 gate (count enable)</td>
</tr>
<tr>
<td>OSC2</td>
<td>—</td>
<td>XTAL</td>
<td>—</td>
<td>— Crystal/Resonator</td>
</tr>
<tr>
<td>CLKOUT</td>
<td>—</td>
<td>CMOS</td>
<td>—</td>
<td>— Fosc/4 output</td>
</tr>
<tr>
<td>RA5/T1CKI/OSC1/CLKIN</td>
<td>RA5</td>
<td>TTL</td>
<td>CMOS</td>
<td>PORTA I/O with prog. pull-up and interrupt-on-change</td>
</tr>
<tr>
<td>T1CKI</td>
<td>ST</td>
<td>—</td>
<td>—</td>
<td>— Timer1 clock</td>
</tr>
<tr>
<td>OSC1</td>
<td>XTAL</td>
<td>—</td>
<td>—</td>
<td>— Crystal/Resonator</td>
</tr>
<tr>
<td>CLKIN</td>
<td>ST</td>
<td>—</td>
<td>—</td>
<td>— External clock input/RC oscillator connection</td>
</tr>
<tr>
<td>RC0/AN4/C2IN+</td>
<td>RC0</td>
<td>TTL</td>
<td>CMOS</td>
<td>PORTC I/O</td>
</tr>
<tr>
<td>AN4</td>
<td>AN</td>
<td>—</td>
<td>—</td>
<td>— A/D Channel 4 input</td>
</tr>
<tr>
<td>C2IN+</td>
<td>AN</td>
<td>—</td>
<td>—</td>
<td>— Comparator 2 non-inverting input</td>
</tr>
<tr>
<td>RC1/AN5/C2IN-</td>
<td>RC1</td>
<td>TTL</td>
<td>CMOS</td>
<td>PORTC I/O</td>
</tr>
<tr>
<td>AN5</td>
<td>AN</td>
<td>—</td>
<td>—</td>
<td>— A/D Channel 5 input</td>
</tr>
<tr>
<td>C2IN-</td>
<td>AN</td>
<td>—</td>
<td>—</td>
<td>— Comparator 2 inverting input</td>
</tr>
<tr>
<td>RC2/AN6/P1D</td>
<td>RC2</td>
<td>TTL</td>
<td>CMOS</td>
<td>PORTC I/O</td>
</tr>
<tr>
<td>AN6</td>
<td>AN</td>
<td>—</td>
<td>—</td>
<td>— A/D Channel 6 input</td>
</tr>
<tr>
<td>P1D</td>
<td>—</td>
<td>CMOS</td>
<td>—</td>
<td>— PWM output</td>
</tr>
<tr>
<td>RC3/AN7/P1C</td>
<td>RC3</td>
<td>TTL</td>
<td>CMOS</td>
<td>PORTC I/O</td>
</tr>
<tr>
<td>AN7</td>
<td>AN</td>
<td>—</td>
<td>—</td>
<td>— A/D Channel 7 input</td>
</tr>
<tr>
<td>P1C</td>
<td>—</td>
<td>CMOS</td>
<td>—</td>
<td>— PWM output</td>
</tr>
<tr>
<td>RC4/C2OUT/P1B</td>
<td>RC4</td>
<td>TTL</td>
<td>CMOS</td>
<td>PORTC I/O</td>
</tr>
<tr>
<td>C2OUT</td>
<td>—</td>
<td>CMOS</td>
<td>—</td>
<td>— Comparator 2 output</td>
</tr>
<tr>
<td>P1B</td>
<td>—</td>
<td>CMOS</td>
<td>—</td>
<td>— PWM output</td>
</tr>
<tr>
<td>RC5/CCP1/P1A</td>
<td>RC5</td>
<td>TTL</td>
<td>CMOS</td>
<td>PORTC I/O</td>
</tr>
<tr>
<td>CCP1</td>
<td>ST</td>
<td>CMOS</td>
<td>—</td>
<td>— Capture input/Compare output</td>
</tr>
<tr>
<td>P1A</td>
<td>—</td>
<td>CMOS</td>
<td>—</td>
<td>— PWM output</td>
</tr>
<tr>
<td>VDD</td>
<td>VDD</td>
<td>Power</td>
<td>—</td>
<td>— Positive supply</td>
</tr>
<tr>
<td>VSS</td>
<td>VSS</td>
<td>Power</td>
<td>—</td>
<td>— Ground reference</td>
</tr>
</tbody>
</table>

**Legend:**
- AN = Analog input or output
- CMOS= CMOS compatible input or output
- HV = High Voltage
- ST = Schmitt Trigger input with CMOS levels
- TTL = TTL compatible input
- XTAL = Crystal
2.0 MEMORY ORGANIZATION

2.1 Program Memory Organization

The PIC16F684 has a 13-bit program counter capable of addressing an 8k x 14 program memory space. Only the first 2k x 14 (0000h-07FFh) for the PIC16F684 is physically implemented. Accessing a location above these boundaries will cause a wrap around within the first 2k 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 PIC16F684

2.2 Data Memory Organization

The data memory (see Figure 2-2) is partitioned into two banks, which contain the General Purpose Registers (GPR) and the Special Function Registers (SFR). The Special Function Registers are located in the first 32 locations of each bank. Register locations 20h-7Fh in Bank 0 and A0h-BFh in Bank 1 are General Purpose Registers, implemented as static RAM. Register locations F0h-FFh in Bank 1 point to addresses 70h-7Fh in Bank 0. All other RAM is unimplemented and returns ‘0’ when read. The RP0 bit of the STATUS register is the bank select bit.

RP0
0 → Bank 0 is selected
1 → Bank 1 is selected

Note: The IRP and RP1 bits of the STATUS register are reserved and should always be maintained as ‘0’ s.
2.2.1 GENERAL PURPOSE REGISTER FILE

The register file is organized as 128 x 8 in the PIC16F684. 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 PIC16F684

<table>
<thead>
<tr>
<th>File Address</th>
<th>File Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>Indirect Addr.(^{(1)})</td>
<td>Indirect Addr.(^{(1)})</td>
</tr>
<tr>
<td>TMR0</td>
<td>OPTION_REG</td>
</tr>
<tr>
<td>00h</td>
<td>80h</td>
</tr>
<tr>
<td>TMR1</td>
<td>PCON</td>
</tr>
<tr>
<td>01h</td>
<td>08h</td>
</tr>
<tr>
<td>STATUS</td>
<td>OSCCON</td>
</tr>
<tr>
<td>02h</td>
<td>09h</td>
</tr>
<tr>
<td>PCL</td>
<td>OSCCAL</td>
</tr>
<tr>
<td>03h</td>
<td>0Ah</td>
</tr>
<tr>
<td>FSR</td>
<td>90h</td>
</tr>
<tr>
<td>04h</td>
<td>91h</td>
</tr>
<tr>
<td>STATUS</td>
<td>92h</td>
</tr>
<tr>
<td>05h</td>
<td>93h</td>
</tr>
<tr>
<td>PORTA</td>
<td>ANSEL</td>
</tr>
<tr>
<td>06h</td>
<td>94h</td>
</tr>
<tr>
<td>TRISA</td>
<td>CCP1CON</td>
</tr>
<tr>
<td>07h</td>
<td>95h</td>
</tr>
<tr>
<td>PORTC</td>
<td>CCP2CON</td>
</tr>
<tr>
<td>08h</td>
<td>96h</td>
</tr>
<tr>
<td>TRISC</td>
<td>TMR2L</td>
</tr>
<tr>
<td>09h</td>
<td>97h</td>
</tr>
<tr>
<td>TMR2H</td>
<td>TMR2L</td>
</tr>
<tr>
<td>0Ah</td>
<td>98h</td>
</tr>
<tr>
<td>T1CON</td>
<td>TMR2H</td>
</tr>
<tr>
<td>0Bh</td>
<td>99h</td>
</tr>
<tr>
<td>T2CON</td>
<td>ANSEL</td>
</tr>
<tr>
<td>0Ch</td>
<td>9Ah</td>
</tr>
<tr>
<td>PCLATH</td>
<td>CCP0CON</td>
</tr>
<tr>
<td>0Dh</td>
<td>9Bh</td>
</tr>
<tr>
<td>INTCON</td>
<td>CCP1CON</td>
</tr>
<tr>
<td>0Eh</td>
<td>9Ch</td>
</tr>
<tr>
<td>PIE1</td>
<td>ADRESL</td>
</tr>
<tr>
<td>0Fh</td>
<td>9Dh</td>
</tr>
<tr>
<td>CMCON0</td>
<td>ADRESH</td>
</tr>
<tr>
<td>10h</td>
<td>9 Eh</td>
</tr>
<tr>
<td>CMCON1</td>
<td>ADCON0</td>
</tr>
<tr>
<td>11h</td>
<td>9Fh</td>
</tr>
<tr>
<td>12h</td>
<td>A0h</td>
</tr>
<tr>
<td>General Purpose Registers</td>
<td>General Purpose Registers</td>
</tr>
<tr>
<td>13h</td>
<td>A1h</td>
</tr>
<tr>
<td>14h</td>
<td>A2h</td>
</tr>
<tr>
<td>15h</td>
<td>A3h</td>
</tr>
<tr>
<td>16h</td>
<td>A4h</td>
</tr>
<tr>
<td>17h</td>
<td>A5h</td>
</tr>
<tr>
<td>18h</td>
<td>A6h</td>
</tr>
<tr>
<td>19h</td>
<td>A7h</td>
</tr>
<tr>
<td>1Ah</td>
<td>A8h</td>
</tr>
<tr>
<td>1Bh</td>
<td>A9h</td>
</tr>
<tr>
<td>1Ch</td>
<td>AAh</td>
</tr>
<tr>
<td>1Dh</td>
<td>ABh</td>
</tr>
<tr>
<td>1 Eh</td>
<td>ACb</td>
</tr>
<tr>
<td>1Fh</td>
<td>ADh</td>
</tr>
<tr>
<td>20h</td>
<td>AEh</td>
</tr>
<tr>
<td>6Fh</td>
<td>BFh</td>
</tr>
<tr>
<td>70h</td>
<td>Bank 0</td>
</tr>
<tr>
<td>7Fh</td>
<td>Bank 1</td>
</tr>
<tr>
<td>F0h</td>
<td>FFh</td>
</tr>
</tbody>
</table>

Note 1: Not a physical register.
**TABLE 2-1: PIC16F684 SPECIAL FUNCTION REGISTERS SUMMARY BANK 0**

<table>
<thead>
<tr>
<th>Addr</th>
<th>Name</th>
<th>Bit 7</th>
<th>Bit 6</th>
<th>Bit 5</th>
<th>Bit 4</th>
<th>Bit 3</th>
<th>Bit 2</th>
<th>Bit 1</th>
<th>Bit 0</th>
<th>Value on POR, BOR</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>00h</td>
<td>INDF</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>xxxx xxxx</td>
<td>19, 104</td>
</tr>
<tr>
<td>01h</td>
<td>TMRO</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>xxxx xxxx</td>
<td>43, 104</td>
</tr>
<tr>
<td>02h</td>
<td>PCL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0000 0000</td>
<td>19, 104</td>
</tr>
<tr>
<td>03h</td>
<td>STATUS</td>
<td>RP1(1)</td>
<td>RP0</td>
<td>TO</td>
<td>PD</td>
<td>Z</td>
<td>DC</td>
<td>C</td>
<td>0001 xxxx</td>
<td>13, 104</td>
<td></td>
</tr>
<tr>
<td>04h</td>
<td>FSR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>xxxx xxxx</td>
<td>19, 104</td>
</tr>
<tr>
<td>05h</td>
<td>PORTA(2)</td>
<td></td>
<td>RA5</td>
<td>RA4</td>
<td>RA3</td>
<td>RA2</td>
<td>RA1</td>
<td>RA0</td>
<td></td>
<td>xxx0 x000</td>
<td>31, 104</td>
</tr>
<tr>
<td>06h</td>
<td></td>
<td>Unimplemented</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>07h</td>
<td>PORTC(2)</td>
<td></td>
<td>RC5</td>
<td>RC4</td>
<td>RC3</td>
<td>RC2</td>
<td>RC1</td>
<td>RC0</td>
<td></td>
<td>xxxx 0000</td>
<td>40, 104</td>
</tr>
<tr>
<td>08h</td>
<td></td>
<td>Unimplemented</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>09h</td>
<td></td>
<td>Unimplemented</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0Ah</td>
<td>PCLATH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>---- 0 0000</td>
<td>19, 104</td>
</tr>
<tr>
<td>0Bh</td>
<td>INTCON</td>
<td>GIE</td>
<td>PEIE</td>
<td>T0IE</td>
<td>INTE</td>
<td>RAIE</td>
<td>T0IF</td>
<td>INTF</td>
<td>RAIF</td>
<td>0000 0000</td>
<td>15, 104</td>
</tr>
<tr>
<td>0Ch</td>
<td>PIR1</td>
<td>EEIF</td>
<td>ADIF</td>
<td>CCP1IF</td>
<td>C2IF</td>
<td>C1IF</td>
<td>OSIF</td>
<td>TMRIF</td>
<td>TMR1IF</td>
<td>0000 0000</td>
<td>17, 104</td>
</tr>
<tr>
<td>0Dh</td>
<td></td>
<td>Unimplemented</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 Eh</td>
<td>TMR1L</td>
<td>Holding Register for the Least Significant Byte of the 16-bit TMR1 Register</td>
<td>xxxx xxxx</td>
<td>47, 104</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0Fh</td>
<td>TMR1H</td>
<td>Holding Register for the Most Significant Byte of the 16-bit TMR1 Register</td>
<td>xxxx xxxx</td>
<td>47, 104</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10h</td>
<td>T1CON</td>
<td>T1GINV</td>
<td>TMR1GE</td>
<td>T1CKPS1</td>
<td>T1CKPS0</td>
<td>T1OSCEN</td>
<td>T1SYNC</td>
<td>TMR1CS</td>
<td>TMR1ON</td>
<td>0000 0000</td>
<td>50, 104</td>
</tr>
<tr>
<td>11h</td>
<td>TMR2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0000 0000</td>
<td>53, 104</td>
</tr>
<tr>
<td>12h</td>
<td>T2CON</td>
<td></td>
<td>TOUTPS3</td>
<td>TOUTPS2</td>
<td>TOUTPS1</td>
<td>TOUTPS0</td>
<td>TMR2ON</td>
<td>T2CKPS1</td>
<td>T2CKPS0</td>
<td>-000 0000</td>
<td>54, 104</td>
</tr>
<tr>
<td>13h</td>
<td>CCPR1L</td>
<td>Capture/Compare/PWM Register 1 Low Byte</td>
<td>xxxx xxxx</td>
<td>80, 104</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14h</td>
<td>CCPR1H</td>
<td>Capture/Compare/PWM Register 1 High Byte</td>
<td>xxxx xxxx</td>
<td>80, 104</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15h</td>
<td>CCP1CON</td>
<td>P1M1</td>
<td>P1M0</td>
<td>DC1B1</td>
<td>DC1B0</td>
<td>CCP1M3</td>
<td>CCP1M2</td>
<td>CCP1M1</td>
<td>CCP1M0</td>
<td>0000 0000</td>
<td>79, 104</td>
</tr>
<tr>
<td>16h</td>
<td>PWM1CON</td>
<td>PRSEN</td>
<td>PDC6</td>
<td>PDC5</td>
<td>PDC4</td>
<td>PDC3</td>
<td>PDC2</td>
<td>PDC1</td>
<td>PDC0</td>
<td>0000 0000</td>
<td>96, 104</td>
</tr>
<tr>
<td>17h</td>
<td>ECPAS</td>
<td>ECPASE</td>
<td>ECPAS2</td>
<td>ECPAS1</td>
<td>ECPAS0</td>
<td>PSSAC1</td>
<td>PSSAC0</td>
<td>PSSBD1</td>
<td>PSSBD0</td>
<td>0000 0000</td>
<td>93, 104</td>
</tr>
<tr>
<td>18h</td>
<td>WDTCON</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>---- 10 0000</td>
<td>111, 104</td>
</tr>
<tr>
<td>19h</td>
<td>CMCON0</td>
<td>C2OUT</td>
<td>C1OUT</td>
<td>C2INV</td>
<td>C1INV</td>
<td>CIS</td>
<td>CM2</td>
<td>CM1</td>
<td>CM0</td>
<td>0000 0000</td>
<td>61, 104</td>
</tr>
<tr>
<td>1Ah</td>
<td>CMCON1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>---- --10</td>
<td>62, 104</td>
</tr>
<tr>
<td>1Bh</td>
<td></td>
<td>Unimplemented</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1Ch</td>
<td></td>
<td>Unimplemented</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1Dh</td>
<td></td>
<td>Unimplemented</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 Eh</td>
<td>ADRESH</td>
<td>Most Significant 8 bits of the left shifted A/D result or 2 bits of right shifted result</td>
<td>xxxx xxxx</td>
<td>71, 104</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1Fh</td>
<td>ADCON0</td>
<td>ADFM</td>
<td>VCFG</td>
<td></td>
<td>CHS2</td>
<td>CHS1</td>
<td>CHS0</td>
<td>GO/DONE</td>
<td>ADON</td>
<td>00-0 0000</td>
<td>70, 104</td>
</tr>
</tbody>
</table>

**Legend:**
- = Unimplemented locations read as ‘0’, u = unchanged, x = unknown, q = value depends on condition, shaded = unimplemented

**Note:**
1. IRP and RP1 bits are reserved, always maintain these bits clear.
2. Port pins with analog functions controlled by the ANSEL register will read ‘0’ immediately after a Reset even though the data latches are either undefined (POR) or unchanged (other Resets).
<table>
<thead>
<tr>
<th>Addr</th>
<th>Name</th>
<th>Bit 7</th>
<th>Bit 6</th>
<th>Bit 5</th>
<th>Bit 4</th>
<th>Bit 3</th>
<th>Bit 2</th>
<th>Bit 1</th>
<th>Bit 0</th>
<th>Value on POR, BOR</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>80h</td>
<td>INDF</td>
<td>xxxxx</td>
<td>xxxx</td>
<td>xxxx</td>
<td>19</td>
<td>104</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>81h</td>
<td>OPTION_REG</td>
<td>RAPU</td>
<td>INTE DG</td>
<td>T0CS</td>
<td>T0SE</td>
<td>PSA</td>
<td>PS2</td>
<td>PS1</td>
<td>PS0</td>
<td>xxxxx</td>
<td>14</td>
</tr>
<tr>
<td>82h</td>
<td>PCL</td>
<td></td>
<td></td>
<td>xxxx</td>
<td>xxxx</td>
<td>19</td>
<td>104</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>84h</td>
<td>FSR</td>
<td>Indirect Data Memory Address Pointer</td>
<td>xxxxx</td>
<td>19</td>
<td>104</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>85h</td>
<td>TRISA</td>
<td>TRISA5</td>
<td>TRISA4</td>
<td>TRISA3</td>
<td>TRISA2</td>
<td>TRISA1</td>
<td>TRISA0</td>
<td></td>
<td></td>
<td>--11 1111</td>
<td>31</td>
</tr>
<tr>
<td>86h</td>
<td>—</td>
<td>Unimplemented</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>87h</td>
<td>TRISC</td>
<td>TRISC5</td>
<td>TRISC4</td>
<td>TRISC3</td>
<td>TRISC2</td>
<td>TRISC1</td>
<td>TRISC0</td>
<td></td>
<td></td>
<td>--11 1111</td>
<td>40</td>
</tr>
<tr>
<td>88h</td>
<td>—</td>
<td>Unimplemented</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>89h</td>
<td>—</td>
<td>Unimplemented</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8Ah</td>
<td>PCLATH</td>
<td>Write Buffer for upper 5 bits of Program Counter</td>
<td>---0</td>
<td>0000</td>
<td>19</td>
<td>104</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8Bh</td>
<td>INTCON</td>
<td>GIE</td>
<td>PEIE</td>
<td>TOIE</td>
<td>INTE</td>
<td>RAIE</td>
<td>T0IF</td>
<td>INTF</td>
<td>RAIF</td>
<td>0000 0000</td>
<td>15</td>
</tr>
<tr>
<td>8Ch</td>
<td>PIE1</td>
<td>EEIE</td>
<td>ADIE</td>
<td>CCP1IE</td>
<td>C2IE</td>
<td>C1IE</td>
<td>OSFIE</td>
<td>TMR2IE</td>
<td>TMR1IE</td>
<td>0000 0000</td>
<td>16</td>
</tr>
<tr>
<td>8Dh</td>
<td>—</td>
<td>Unimplemented</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8 Eh</td>
<td>PCON</td>
<td></td>
<td></td>
<td>ULPWUE</td>
<td>SBOREN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>--01 --00</td>
<td>18</td>
</tr>
<tr>
<td>8Fh</td>
<td>OSCCON</td>
<td>IRF2</td>
<td>IRFC1</td>
<td>IRFC0</td>
<td>OTS[2]</td>
<td>HTS</td>
<td>LTS</td>
<td>SCS</td>
<td>--11 0010</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>90h</td>
<td>OSCITUDE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>TUN4</td>
<td>TUN3</td>
<td>TUN2</td>
<td>TUN1</td>
<td>TUN0</td>
<td>24</td>
</tr>
<tr>
<td>91h</td>
<td>ANSEL</td>
<td>ANS7</td>
<td>ANS6</td>
<td>ANS5</td>
<td>ANS4</td>
<td>ANS3</td>
<td>ANS2</td>
<td>ANS1</td>
<td>ANS0</td>
<td>1111 1111</td>
<td>32</td>
</tr>
<tr>
<td>92h</td>
<td>PR2</td>
<td>Timer2 Module Period Register</td>
<td>1111</td>
<td>1111</td>
<td>53</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>93h</td>
<td>—</td>
<td>Unimplemented</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>94h</td>
<td>—</td>
<td>Unimplemented</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>95h</td>
<td>WPUA[3]</td>
<td></td>
<td></td>
<td>WPUA5</td>
<td>WPUA4</td>
<td>WPUA2</td>
<td>WPUA1</td>
<td>WPUA0</td>
<td></td>
<td>--11 1111</td>
<td>33</td>
</tr>
<tr>
<td>96h</td>
<td>IOCA</td>
<td></td>
<td></td>
<td>IOCA5</td>
<td>IOCA4</td>
<td>IOCA3</td>
<td>IOCA2</td>
<td>IOCA1</td>
<td>IOCA0</td>
<td>--00 0000</td>
<td>33</td>
</tr>
<tr>
<td>97h</td>
<td>—</td>
<td>Unimplemented</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>98h</td>
<td>—</td>
<td>Unimplemented</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>99h</td>
<td>VRCO N</td>
<td></td>
<td></td>
<td>VREN</td>
<td>VRR</td>
<td>VR3</td>
<td>VR2</td>
<td>VR1</td>
<td>VR0</td>
<td>0-0 0000</td>
<td>63</td>
</tr>
<tr>
<td>9Ah</td>
<td>EEAD R</td>
<td>EEDAT7</td>
<td>EEDAT6</td>
<td>EEDAT5</td>
<td>EEDAT4</td>
<td>EEDAT3</td>
<td>EEDAT2</td>
<td>EEDAT1</td>
<td>EEDAT0</td>
<td>0000 0000</td>
<td>75</td>
</tr>
<tr>
<td>9 Bh</td>
<td>EEAD R</td>
<td>EEDR7</td>
<td>EEDR6</td>
<td>EEDR5</td>
<td>EEDR4</td>
<td>EEDR3</td>
<td>EEDR2</td>
<td>EEDR1</td>
<td>EEDR0</td>
<td>0000 0000</td>
<td>75</td>
</tr>
<tr>
<td>9Ch</td>
<td>EECO N1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>WRERR</td>
<td>WREN</td>
<td>WR</td>
<td>RD</td>
<td>---- 0010</td>
<td>76</td>
</tr>
<tr>
<td>9Dh</td>
<td>EECO N2</td>
<td>EEPROM Control Register 2 (not a physical register)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>76</td>
</tr>
<tr>
<td>9 Eh</td>
<td>ADRES L</td>
<td>Least Significant 2 bits of the left shifted result or 8 bits of the right shifted result</td>
<td>xxxxx</td>
<td>xxxx</td>
<td>71</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9Fh</td>
<td>ADCON1</td>
<td></td>
<td></td>
<td>ADCS2</td>
<td>ADCS1</td>
<td>ADCS0</td>
<td></td>
<td></td>
<td></td>
<td>--00 0000</td>
<td>70</td>
</tr>
</tbody>
</table>

Legend:  
- = Unimplemented locations read as ‘0’, u = unchanged, x = unknown, q = value depends on condition, shaded = unimplemented

Note:  
1: IRP and RP1 bits are reserved, always maintain these bits clear.
2: OTS bit of the OSCCON register reset to ‘0’ with Dual Speed Start-up and LP, HS or XT selected as the oscillator.
3: RA3 pull-up is enabled when MCLRE is ‘1’ in the Configuration Word register.
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 TO and 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).

REGISTE 2-1: STATUS: STATUS REGISTER

<table>
<thead>
<tr>
<th>Reserved</th>
<th>Reserved</th>
<th>R/W-0</th>
<th>R-1</th>
<th>R-1</th>
<th>R/W-x</th>
<th>R/W-x</th>
<th>R/W-x</th>
</tr>
</thead>
<tbody>
<tr>
<td>IRP</td>
<td>RP1</td>
<td>RP0</td>
<td>TO</td>
<td>PD</td>
<td>Z</td>
<td>DC</td>
<td>C</td>
</tr>
<tr>
<td>bit 7</td>
<td>bit 6</td>
<td>bit 5</td>
<td>bit 4</td>
<td>bit 3</td>
<td>bit 2</td>
<td>bit 1</td>
<td>bit 0</td>
</tr>
</tbody>
</table>

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

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 TO: Time-out bit
1 = After power-up, CLRWD T instruction or SLEEP instruction
0 = A WDT time-out occurred
bit 3 PD: Power-down bit
1 = After power-up or by the CLRWD T 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/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/Borrow bit(1) (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 1: For 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-order or low-order bit of the source register.

Note 2: The C and DC bits operate as a Borrow and Digit Borrow out bit, respectively, in subtraction. See the SUBLW and SUBWF instructions for examples.
2.2.2.2 OPTIONS Register

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

- Timer0/WDT prescaler
- External RA2/INT interrupt
- Timer0
- Weak pull-ups on PORTA

**Note:** To achieve a 1:1 prescaler assignment for Timer0, assign the prescaler to the WDT by setting PSA bit to ‘1’ of the OPTION register. See Section 5.1.3 “Software Programmable Prescaler”.

**REGISTER 2-2: OPTION_REG: OPTION REGISTER**

<table>
<thead>
<tr>
<th>R/W-1</th>
<th>R/W-1</th>
<th>R/W-1</th>
<th>R/W-1</th>
<th>R/W-1</th>
<th>R/W-1</th>
<th>R/W-1</th>
<th>R/W-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAPU</td>
<td>INTEDG</td>
<td>T0CS</td>
<td>T0SE</td>
<td>PSA</td>
<td>PS2</td>
<td>PS1</td>
<td>PS0</td>
</tr>
</tbody>
</table>

bit 7  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: Timer0 Clock Source Select bit
  1 = Transition on RA2/T0CKI pin
  0 = Internal instruction cycle clock (Fosc/4)

bit 4  T0SE: Timer0 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  PS<2:0>: Prescaler Rate Select bits

<table>
<thead>
<tr>
<th>BIT VALUE</th>
<th>TIMER0 RATE</th>
<th>WDT RATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>1 : 2</td>
<td>1 : 1</td>
</tr>
<tr>
<td>001</td>
<td>1 : 4</td>
<td>1 : 2</td>
</tr>
<tr>
<td>010</td>
<td>1 : 8</td>
<td>1 : 4</td>
</tr>
<tr>
<td>011</td>
<td>1 : 16</td>
<td>1 : 8</td>
</tr>
<tr>
<td>100</td>
<td>1 : 32</td>
<td>1 : 16</td>
</tr>
<tr>
<td>101</td>
<td>1 : 64</td>
<td>1 : 32</td>
</tr>
<tr>
<td>110</td>
<td>1 : 128</td>
<td>1 : 64</td>
</tr>
<tr>
<td>111</td>
<td>1 : 256</td>
<td>1 : 128</td>
</tr>
</tbody>
</table>

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 of the INTCON register. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

**REGISTER 2-3:  INTCON: INTERRUPT CONTROL REGISTER**

<table>
<thead>
<tr>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>GIE</td>
<td>PEIE</td>
<td>T0IE</td>
<td>INTE</td>
<td>RAIE</td>
<td>T0IF</td>
<td>INTF</td>
<td>RAIF</td>
</tr>
</tbody>
</table>

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: Timer0 Overflow Interrupt Enable bit
1 = Enables the Timer0 interrupt
0 = Disables the Timer0 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: PORTA Change Interrupt Enable bit\(^{(1)}\)
1 = Enables the PORTA change interrupt
0 = Disables the PORTA change interrupt

bit 2  
T0IF: Timer0 Overflow Interrupt Flag bit\(^{(2)}\)
1 = Timer0 register has overflowed (must be cleared in software)
0 = Timer0 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: PORTA 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.

**Note 2:** T0IF bit is set when TMR0 rolls over. TMR0 is unchanged on Reset and should be initialized before clearing T0IF bit.
2.2.2.4 PIE1 Register

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

Note: Bit PEIE of the INTCON register must be set to enable any peripheral interrupt.

REGISTER 2-4: PIE1: PERIPHERAL INTERRUPT ENABLE REGISTER 1

<table>
<thead>
<tr>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>EEIE</td>
<td>ADIE</td>
<td>CCP1IE</td>
<td>C2IE</td>
<td>C1IE</td>
<td>OSFIE</td>
<td>TMR2IE</td>
<td>TMR1IE</td>
</tr>
</tbody>
</table>

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

bit 7    EEIE: EE Write Complete Interrupt Enable bit
         1 = Enables the EE write complete interrupt
         0 = Disables the EE write complete interrupt

bit 6    ADIE: A/D Converter (ADC) Interrupt Enable bit
         1 = Enables the ADC interrupt
         0 = Disables the ADC interrupt

bit 5    CCP1IE: CCP1 Interrupt Enable bit
         1 = Enables the CCP1 interrupt
         0 = Disables the CCP1 interrupt

bit 4    C2IE: Comparator 2 Interrupt Enable bit
         1 = Enables the Comparator 2 interrupt
         0 = Disables the Comparator 2 interrupt

bit 3    C1IE: Comparator 1 Interrupt Enable bit
         1 = Enables the Comparator 1 interrupt
         0 = Disables the Comparator 1 interrupt

bit 2    OSFIE: Oscillator Fail Interrupt Enable bit
         1 = Enables the oscillator fail interrupt
         0 = Disables the oscillator fail interrupt

bit 1    TMR2IE: Timer2 to PR2 Match Interrupt Enable bit
         1 = Enables the Timer2 to PR2 match interrupt
         0 = Disables the Timer2 to PR2 match interrupt

bit 0    TMR1IE: Timer1 Overflow Interrupt Enable bit
         1 = Enables the Timer1 overflow interrupt
         0 = Disables the Timer1 overflow interrupt
2.2.2.5 PIR1 Register

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

**REGISTER 2-5: PIR1: PERIPHERAL INTERRUPT REQUEST REGISTER 1**

<table>
<thead>
<tr>
<th>Bit 7</th>
<th>Bit 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>EEIF</td>
<td>ADIF</td>
</tr>
<tr>
<td>CCP1IF</td>
<td>C2IF</td>
</tr>
<tr>
<td>C1IF</td>
<td>OSFIF</td>
</tr>
<tr>
<td>TMR2IF</td>
<td>TMR1IF</td>
</tr>
</tbody>
</table>

**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

**bit 7 EEIF:** EEPROM Write Operation Interrupt Flag bit
- 1 = The write operation completed (must be cleared in software)
- 0 = The write operation has not completed or has not been started

**bit 6 ADIF:** A/D Interrupt Flag bit
- 1 = A/D conversion complete
- 0 = A/D conversion has not completed or has not been started

**bit 5 CCP1IF:** CCP1 Interrupt Flag bit
- Capture mode:
  - 1 = A TMR1 register capture occurred (must be cleared in software)
  - 0 = No TMR1 register capture occurred
- Compare mode:
  - 1 = A TMR1 register compare match occurred (must be cleared in software)
  - 0 = No TMR1 register compare match occurred
- PWM mode:
  - Unused in this mode

**bit 4 C2IF:** Comparator 2 Interrupt Flag bit
- 1 = Comparator 2 output has changed (must be cleared in software)
- 0 = Comparator 2 output has not changed

**bit 3 C1IF:** Comparator 1 Interrupt Flag bit
- 1 = Comparator 1 output has changed (must be cleared in software)
- 0 = Comparator 1 output has not changed

**bit 2 OSFIF:** Oscillator Fail Interrupt Flag bit
- 1 = System oscillator failed, clock input has changed to INTOSC (must be cleared in software)
- 0 = System clock operating

**bit 1 TMR2IF:** Timer2 to PR2 Match Interrupt Flag bit
- 1 = Timer2 to PR2 match occurred (must be cleared in software)
- 0 = Timer2 to PR2 match has not occurred

**bit 0 TMR1IF:** Timer1 Overflow Interrupt Flag bit
- 1 = Timer1 register overflowed (must be cleared in software)
- 0 = Timer1 has not overflowed

**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 of the INTCON register. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.
### 2.2.2.6 PCON Register

The Power Control (PCON) register (see Table 12-2) contains flag bits to differentiate between a:

- Power-on Reset (POR)
- Brown-out Reset (BOR)
- Watchdog Timer Reset (WDT)
- External MCLR Reset

The PCON register also controls the Ultra Low-Power Wake-up and software enable of the BOR.

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

#### REGISTER 2-6: PCON: POWER CONTROL REGISTER

<table>
<thead>
<tr>
<th>bit 7-6</th>
<th>bit 5-4</th>
<th>bit 3-2</th>
<th>bit 1-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>U-0</td>
<td>U-0</td>
<td>R/W-0</td>
<td>R/W-1</td>
</tr>
<tr>
<td>U-0</td>
<td>U-0</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>—</td>
<td>—</td>
<td>ULPWUE</td>
<td>SBOREN</td>
</tr>
<tr>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>—</td>
<td>—</td>
<td>POR</td>
<td>BOR</td>
</tr>
</tbody>
</table>

**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

- **bit 7-6 Unimplemented:** Read as ‘0’
- **bit 5 ULPWUE:** Ultra Low-Power Wake-Up Enable bit
  - 1 = Ultra Low-Power Wake-up enabled
  - 0 = Ultra Low-Power Wake-up disabled
- **bit 4 SBOREN:** Software BOR Enable bit\(^1\)
  - 1 = BOR enabled
  - 0 = BOR disabled
- **bit 3-2 Unimplemented:** Read as ‘0’
- **bit 1 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 BOR:** Brown-out Reset Status bit
  - 1 = No Brown-out Reset occurred
  - 0 = A Brown-out Reset occurred (must be set in software after a Power-on Reset or Brown-out Reset occurs)

**Note 1:** BOREN<1:0> = 01 in the Configuration Word register for this bit to control the BOR.
### 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](image)

### 2.3.1 MODIFYING PCL

Executing any instruction with the PCL register as the destination simultaneously causes the Program Counter PC<12:8> bits (PCH) to be replaced by the contents of the PCLATH register. This allows the entire contents of the program counter to be changed by first writing the desired upper 5 bits to the PCLATH register. Then, when the lower 8 bits are written to the PCL register, all 13 bits of the program counter will change to the values contained in the PCLATH register and those being written to the PCL register.

A computed GOTO is accomplished by adding an offset to the program counter (ADDWF PCL). Care should be exercised when jumping into a look-up table or program branch table (computed GOTO) by modifying the PCL register. Assuming that PCLATH is set to the table start address, if the table length is greater than 255 instructions or if the lower 8 bits of the memory address rolls over from 0xFF to 0x00 in the middle of the table, then PCLATH must be incremented for each address rollover that occurs between the table beginning and the target location within the table.

For more information refer to Application Note AN556, “Implementing a Table Read” (DS00556).

### 2.3.2 STACK

The PIC16F684 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 PIC 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.

**Note 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 and the IRP bit of the STATUS register, 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

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

© 2006 Microchip Technology Inc.
FIGURE 2-4: DIRECT/INDIRECT ADDRESSING PIC16F684

For memory map detail, see Figure 2-2.

**Note 1:** The RP1 and IRP bits are reserved; always maintain these bits clear.
3.0 OSCILLATOR MODULE (WITH FAIL-SAFE CLOCK MONITOR)

3.1 Overview

The Oscillator module has a wide variety of clock sources and selection features that allow it to be used in a wide range of applications while maximizing performance and minimizing power consumption. Figure 3-1 illustrates a block diagram of the Oscillator module.

Clock sources can be configured from external oscillators, quartz crystal resonators, ceramic resonators and Resistor-Capacitor (RC) circuits. In addition, the system clock source can be configured from one of two internal oscillators, with a choice of speeds selectable via software. Additional clock features include:

- Selectable system clock source between external or internal via software.
- Two-Speed Start-up mode, which minimizes latency between external oscillator start-up and code execution.
- Fail-Safe Clock Monitor (FSCM) designed to detect a failure of the external clock source (LP, XT, HS, EC or RC modes) and switch automatically to the internal oscillator.

The Oscillator module can be configured in one of eight clock modes:

1. EC – External clock with I/O on OSC2/CLKOUT.
2. LP – 32 kHz Low-Power Crystal mode.
3. XT – Medium Gain Crystal or Ceramic Resonator Oscillator mode.
4. HS – High Gain Crystal or Ceramic Resonator mode.
5. RC – External Resistor-Capacitor (RC) with Fosc/4 output on OSC2/CLKOUT.
6. RCIO – External Resistor-Capacitor (RC) with I/O on OSC2/CLKOUT.
7. INTOSC – Internal oscillator with Fosc/4 output on OSC2 and I/O on OSC1/CLKIN.
8. INTOSCIO – Internal oscillator with I/O on OSC1/CLKIN and OSC2/CLKOUT.

Clock Source modes are configured by the FOSC<2:0> bits in the Configuration Word register (CONFIG). The internal clock can be generated from two internal oscillators. The HFINTOSC is a calibrated high-frequency oscillator. The LFINTOSC is an uncalibrated low-frequency oscillator.

FIGURE 3-1: PICmicro® MCU CLOCK SOURCE BLOCK DIAGRAM
# 3.2 Oscillator Control

The Oscillator Control (OSCCON) register (Figure 3-1) controls the system clock and frequency selection options. The OSCCON register contains the following bits:

- Frequency selection bits (IRCF)
- Frequency Status bits (HTS, LTS)
- System clock control bits (OSTS, SCS)

## REGISTER 3-1: OSCCON: OSCILLATOR CONTROL REGISTER

<table>
<thead>
<tr>
<th>bit 7</th>
<th>bit 6-4</th>
<th>bit 3</th>
<th>bit 2</th>
<th>bit 1</th>
<th>bit 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>IRCF2</td>
<td>IRCF1</td>
<td>IRCF0</td>
<td>OSTS(1)</td>
<td>HTS</td>
<td>LTS</td>
</tr>
</tbody>
</table>

### 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

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

### bit 6-4
- **IRCF<2:0>: Internal Oscillator Frequency Select bits**
  - 111 = 8 MHz
  - 110 = 4 MHz (default)
  - 101 = 2 MHz
  - 100 = 1 MHz
  - 011 = 500 kHz
  - 010 = 250 kHz
  - 001 = 125 kHz
  - 000 = 31 kHz (LFINTOSC)

### bit 3
- **OSTS**: Oscillator Start-up Time-out Status bit(1)
  - 1 = Device is running from the external clock defined by FOSC<2:0> of the CONFIG register
  - 0 = Device is running from the internal oscillator (HFINTOSC or LFINTOSC)

### bit 2
- **HTS**: HFINTOSC Status bit (High Frequency – 8 MHz to 125 kHz)
  - 1 = HFINTOSC is stable
  - 0 = HFINTOSC is not stable

### bit 1
- **LTS**: LFINTOSC Stable bit (Low Frequency – 31 kHz)
  - 1 = LFINTOSC is stable
  - 0 = LFINTOSC is not stable

### bit 0
- **SCS**: System Clock Select bit
  - 1 = Internal oscillator is used for system clock
  - 0 = Clock source defined by FOSC<2:0> of the CONFIG register

### Note 1:
- Bit resets to ‘0’ with Two-Speed Start-up and LP, XT or HS selected as the Oscillator mode or Fail-Safe mode is enabled.
3.3 Clock Source Modes

Clock Source modes can be classified as external or internal.
- External Clock modes rely on external circuitry for the clock source. Examples are: Oscillator modules (EC mode), quartz crystal resonators or ceramic resonators (LP, XT and HS modes) and Resistor-Capacitor (RC) mode circuits.
- Internal clock sources are contained internally within the Oscillator module. The Oscillator module has two internal oscillators: the 8 MHz High-Frequency Internal Oscillator (HFINTOSC) and the 31 kHz Low-Frequency Internal Oscillator (LFINTOSC).

The system clock can be selected between external or internal clock sources via the System Clock Select (SCS) bit of the OSCCON register. See Section 3.6 “Clock Switching” for additional information.

3.4 External Clock Modes

3.4.1 OSCILLATOR START-UP TIMER (OST)

If the Oscillator module is configured for LP, XT or HS modes, the Oscillator Start-up Timer (OST) counts 1024 oscillations from OSC1. This occurs following a Power-on Reset (POR) and when the Power-up Timer (PWRT) has expired (if configured), or a wake-up from Sleep. During this time, the program counter does not increment and program execution is suspended. The OST ensures that the oscillator circuit, using a quartz crystal resonator or ceramic resonator, has started and is providing a stable system clock to the Oscillator module. When switching between clock sources, a delay is required to allow the new clock to stabilize. These oscillator delays are shown in Table 3-1.

In order to minimize latency between external oscillator start-up and code execution, the Two-Speed Clock Start-up mode can be selected (see Section 3.7 “Two-Speed Clock Start-up Mode”).

3.4.2 EC MODE

The External Clock (EC) mode allows an externally generated logic level as the system clock source. When operating in this mode, an external clock source is connected to the OSC1 input and the OSC2 is available for general purpose I/O. Figure 3-2 shows the pin connections for EC mode.

The Oscillator Start-up Timer (OST) is disabled when EC mode is selected. Therefore, there is no delay in operation after a Power-on Reset (POR) or wake-up from Sleep. Because the PICmicro MCU design is fully static, stopping the external clock input will have the effect of halting the device while leaving all data intact. Upon restarting the external clock, the device will resume operation as if no time had elapsed.

### TABLE 3-1: OSCILLATOR DELAY EXAMPLES

<table>
<thead>
<tr>
<th>Switch From</th>
<th>Switch To</th>
<th>Frequency</th>
<th>Oscillator Delay</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sleep/POR</td>
<td>LFINTOSC</td>
<td>31 kHz</td>
<td>Oscillator Warm-Up Delay (TWARM)</td>
</tr>
<tr>
<td></td>
<td>HFINTOSC</td>
<td>125 kHz to 8 MHz</td>
<td></td>
</tr>
<tr>
<td>Sleep/POR</td>
<td>EC, RC</td>
<td>DC – 20 MHz</td>
<td>2 cycles</td>
</tr>
<tr>
<td>LFINTOSC (31 kHz)</td>
<td>EC, RC</td>
<td>DC – 20 MHz</td>
<td>1 cycle of each</td>
</tr>
<tr>
<td>Sleep/POR</td>
<td>LP, XT, HS</td>
<td>32 kHz to 20 MHz</td>
<td>1024 Clock Cycles (OST)</td>
</tr>
<tr>
<td>LFINTOSC (31 kHz)</td>
<td>HFINTOSC</td>
<td>125 kHz to 8 MHz</td>
<td>1 μs (approx.)</td>
</tr>
</tbody>
</table>

**FIGURE 3-2: EXTERNAL CLOCK (EC) MODE OPERATION**

- Clock from Ext. System
- OSC1/CLKIN
- PICmicro® MCU
- I/O
- OSC2/CLKOUT\(^{(1)}\)

**Note 1:** Alternate pin functions are listed in the Section 1.0 “Device Overview”.
3.4.3 LP, XT, HS MODES

The LP, XT and HS modes support the use of quartz crystal resonators or ceramic resonators connected to OSC1 and OSC2 (Figure 3-3). The mode selects a low, medium or high gain setting of the internal inverter-amplifier to support various resonator types and speed.

**LP** Oscillator mode selects the lowest gain setting of the internal inverter-amplifier. LP mode current consumption is the least of the three modes. This mode is designed to drive only 32.768 kHz tuning-fork type crystals (watch crystals).

**XT** Oscillator mode selects the intermediate gain setting of the internal inverter-amplifier. XT mode current consumption is the medium of the three modes. This mode is best suited to drive resonators with a medium drive level specification.

**HS** Oscillator mode selects the highest gain setting of the internal inverter-amplifier. HS mode current consumption is the highest of the three modes. This mode is best suited for resonators that require a high drive setting.

Figure 3-3 and Figure 3-4 show typical circuits for quartz crystal and ceramic resonators, respectively.

**FIGURE 3-3:** QUARTZ CRYSTAL OPERATION (LP, XT OR HS MODE)

- **Note 1:** A series resistor (Rs) may be required for quartz crystals with low drive level.
- **Note 2:** The value of Rs varies with the Oscillator mode selected (typically between 2 MΩ to 10 MΩ).

**FIGURE 3-4:** CERAMIC RESONATOR OPERATION (XT OR HS MODE)

- **Note 1:** A series resistor (Rs) may be required for ceramic resonators with low drive level.
- **Note 2:** The value of Rs varies with the Oscillator mode selected (typically between 2 MΩ to 10 MΩ).
- **Note 3:** An additional parallel feedback resistor (Rp) may be required for proper ceramic resonator operation.

---

Note 1: Quartz crystal characteristics vary according to type, package and manufacturer. The user should consult the manufacturer data sheets for specifications and recommended application.

2: Always verify oscillator performance over the VDD and temperature range that is expected for the application.

3: For oscillator design assistance, reference the following Microchip Applications Notes:

- AN826, “Crystal Oscillator Basics and Crystal Selection for rfPIC® and PICmicro® Devices” (DS00826)
- AN849, “Basic PICmicro® Oscillator Design” (DS00849)
- AN943, “Practical PICmicro® Oscillator Analysis and Design” (DS00943)
- AN949, “Making Your Oscillator Work” (DS00949)
3.4.4 EXTERNAL RC MODES

The external Resistor-Capacitor (RC) modes support the use of an external RC circuit. This allows the designer maximum flexibility in frequency choice while keeping costs to a minimum when clock accuracy is not required. There are two modes: RC and RCIO.

In RC mode, the RC circuit connects to OSC1. OSC2/CLKOUT outputs the RC oscillator frequency divided by 4. This signal may be used to provide a clock for external circuitry, synchronization, calibration, test or other application requirements. Figure 3-5 shows the external RC mode connections.

![FIGURE 3-5: EXTERNAL RC MODES](image)

In RCIO mode, the RC circuit is connected to OSC1. OSC2 becomes an additional general purpose I/O pin. The RC oscillator frequency is a function of the supply voltage, the resistor (REXT) and capacitor (CEXT) values and the operating temperature. Other factors affecting the oscillator frequency are:

- threshold voltage variation
- component tolerances
- packaging variations in capacitance

The user also needs to take into account variation due to tolerance of external RC components used.

3.5 Internal Clock Modes

The Oscillator module has two independent, internal oscillators that can be configured or selected as the system clock source.

1. The **HFINTOSC** (High-Frequency Internal Oscillator) is factory calibrated and operates at 8 MHz. The frequency of the HFINTOSC can be user-adjusted via software using the OSCTUNE register (Register 3-2).

2. The **LFINTOSC** (Low-Frequency Internal Oscillator) is uncalibrated and operates at 31 kHz.

The system clock speed can be selected via software using the Internal Oscillator Frequency Select bits IRCF<2:0> of the OSCCON register.

The system clock can be selected between external or internal clock sources via the System Clock Selection (SCS) bit of the OSCCON register. See Section 3.6 “Clock Switching” for more information.

3.5.1 INTOSC AND INTOSCIO MODES

The INTOSC and INTOSCIO modes configure the internal oscillators as the system clock source when the device is programmed using the oscillator selection or the FOSC<2:0> bits in the Configuration Word register (CONFIG).

In **INTOSC** mode, OSC1/CLKIN is available for general purpose I/O. OSC2/CLKOUT outputs the selected internal oscillator frequency divided by 4. The CLKOUT signal may be used to provide a clock for external circuitry, synchronization, calibration, test or other application requirements.

In **INTOSCIO** mode, OSC1/CLKIN and OSC2/CLKOUT are available for general purpose I/O.

3.5.2 HFINTOSC

The High-Frequency Internal Oscillator (HFINTOSC) is a factory calibrated 8 MHz internal clock source. The frequency of the HFINTOSC can be altered via software using the OSCTUNE register (Register 3-2).

The output of the HFINTOSC connects to a postscaler and multiplexer (see Figure 3-1). One of seven frequencies can be selected via software using the IRCF<2:0> bits of the OSCCON register. See Section 3.5.4 “Frequency Select Bits (IRCF)” for more information.

The HFINTOSC is enabled by selecting any frequency between 8 MHz and 125 kHz by setting the IRCF<2:0> bits of the OSCCON register ≠ 000. Then, set the System Clock Source (SCS) bit of the OSCCON register to ‘1’ or enable Two-Speed Start-up by setting the IESO bit in the Configuration Word register (CONFIG) to ‘1’.

The HF Internal Oscillator (HTS) bit of the OSCCON register indicates whether the HFINTOSC is stable or not.
3.5.2.1 OSCTUNE Register

The HFINTOSC is factory calibrated but can be adjusted in software by writing to the OSCTUNE register (Register 3-2).

The default value of the OSCTUNE register is ‘0’. The value is a 5-bit two’s complement number.

When the OSCTUNE register is modified, the HFINTOSC frequency will begin shifting to the new frequency. Code execution continues during this shift. There is no indication that the shift has occurred.

OSCTUNE does not affect the LFINTOSC frequency. Operation of features that depend on the LFINTOSC clock source frequency, such as the Power-up Timer (PWRT), Watchdog Timer (WDT), Fail-Safe Clock Monitor (FSCM) and peripherals, are not affected by the change in frequency.

REGISTER 3-2: OSCTUNE: OSCILLATOR TUNING REGISTER

<table>
<thead>
<tr>
<th>U-0</th>
<th>U-0</th>
<th>U-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>TUN4</td>
<td>TUN3</td>
<td>TUN2</td>
<td>TUN1</td>
<td>TUN0</td>
</tr>
</tbody>
</table>

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

bit 7-5 Unimplemented: Read as ‘0’
bit 4-0 TUN<4:0>: Frequency Tuning bits
01111 = Maximum frequency
01110 =
•
•
00001 =
00000 = Oscillator module is running at the calibrated frequency.
11111 =
•
•
10000 = Minimum frequency
3.5.3 LFINTOSC

The Low-Frequency Internal Oscillator (LFINTOSC) is an uncalibrated 31 kHz internal clock source.

The output of the LFINTOSC connects to a postscaler and multiplexer (see Figure 3-1). Select 31 kHz, via software, using the IRCF<2:0> bits of the OSCCON register. See Section 3.5.4 “Frequency Select Bits (IRCF)” for more information. The LFINTOSC is also the frequency for the Power-up Timer (PWRT), Watchdog Timer (WDT) and Fail-Safe Clock Monitor (FSCM).

The LFINTOSC is enabled by selecting 31 kHz (IRCF<2:0> bits of the OSCCON register = 000) as the system clock source (SCS bit of the OSCCON register = 1), or when any of the following are enabled:

- Two-Speed Start-up IESO bit of the Configuration Word register = 1 and IRCF<2:0> bits of the OSCCON register = 000
- Power-up Timer (PWRT)
- Watchdog Timer (WDT)
- Fail-Safe Clock Monitor (FSCM)

The LF Internal Oscillator (LTS) bit of the OSCCON register indicates whether the LFINTOSC is stable or not.

3.5.4 FREQUENCY SELECT BITS (IRCF)

The output of the 8 MHz HFINTOSC and 31 kHz LFINTOSC connects to a postscaler and multiplexer (see Figure 3-1). The Internal Oscillator Frequency Select bits IRCF<2:0> of the OSCCON register select the frequency output of the internal oscillators. One of eight frequencies can be selected via software:

- 8 MHz
- 4 MHz (Default after Reset)
- 2 MHz
- 1 MHz
- 500 kHz
- 250 kHz
- 125 kHz
- 31 kHz (LFINTOSC)

Note: Following any Reset, the IRCF<2:0> bits of the OSCCON register are set to '110' and the frequency selection is set to 4 MHz. The user can modify the IRCF bits to select a different frequency.

3.5.5 HFINTOSC AND LFINTOSC CLOCK SWITCH TIMING

When switching between the LFINTOSC and the HFINTOSC, the new oscillator may already be shut down to save power (see Figure 3-6). If this is the case, there is a delay after the IRCF<2:0> bits of the OSCCON register are modified before the frequency selection takes place. The LTS and HTS bits of the OSCCON register will reflect the current active status of the LFINTOSC and HFINTOSC oscillators. The timing of a frequency selection is as follows:

1. IRCF<2:0> bits of the OSCCON register are modified.
2. If the new clock is shut down, a clock start-up delay is started.
3. Clock switch circuitry waits for a falling edge of the current clock.
4. CLKOUT is held low and the clock switch circuitry waits for a rising edge in the new clock.
5. CLKOUT is now connected with the new clock. LTS and HTS bits of the OSCCON register are updated as required.
6. Clock switch is complete.

See Figure 3-1 for more details.

If the internal oscillator speed selected is between 8 MHz and 125 kHz, there is no start-up delay before the new frequency is selected. This is because the old and new frequencies are derived from the HFINTOSC via the postscaler and multiplexer.

Start-up delay specifications are located in the oscillator tables of Section 15.0 “Electrical Specifications”.

Note: Following any Reset, the IRCF<2:0> bits of the OSCCON register are set to '110' and the frequency selection is set to 4 MHz. The user can modify the IRCF bits to select a different frequency.
FIGURE 3-6: INTERNAL OSCILLATOR SWITCH TIMING

**HFINTOSC -> LFINTOSC (FSCM and WDT disabled)**

- HFINTOSC
- LFINTOSC
- IRCF <2:0>: ≠ 0
- System Clock

**HFINTOSC -> LFINTOSC (Either FSCM or WDT enabled)**

- HFINTOSC
- LFINTOSC
- IRCF <2:0>: ≠ 0
- System Clock

**LFINTOSC -> HFINTOSC**

- LFINTOSC
- HFINTOSC
- IRCF <2:0>: = 0
- System Clock

LFINTOSC turns off unless WDT or FSCM is enabled.
3.6 Clock Switching

The system clock source can be switched between external and internal clock sources via software using the System Clock Select (SCS) bit of the OSCCON register.

3.6.1 SYSTEM CLOCK SELECT (SCS) BIT

The System Clock Select (SCS) bit of the OSCCON register selects the system clock source that is used for the CPU and peripherals.

- When the SCS bit of the OSCCON register = 0, the system clock source is determined by configuration of the FOSC<2:0> bits in the Configuration Word register (CONFIG).
- When the SCS bit of the OSCCON register = 1, the system clock source is chosen by the internal oscillator frequency selected by the IRCF<2:0> bits of the OSCCON register. After a Reset, the SCS bit of the OSCCON register is always cleared.

Note: Any automatic clock switch, which may occur from Two-Speed Start-up or Fail-Safe Clock Monitor, does not update the SCS bit of the OSCCON register. The user can monitor the OSTS bit of the OSCCON register to determine the current system clock source.

3.6.2 OSCILLATOR START-UP TIME-OUT STATUS (OSTS) BIT

The Oscillator Start-up Time-out Status (OSTS) bit of the OSCCON register indicates whether the system clock is running from the external clock source, as defined by the FOSC<2:0> bits in the Configuration Word register (CONFIG), or from the internal clock source. In particular, OSTS indicates that the Oscillator Start-up Timer (OST) has timed out for LP, XT or HS modes.

3.7 Two-Speed Clock Start-up Mode

Two-Speed Start-up mode provides additional power savings by minimizing the latency between external oscillator start-up and code execution. In applications that make heavy use of the Sleep mode, Two-Speed Start-up will remove the external oscillator start-up time from the time spent awake and can reduce the overall power consumption of the device.

This mode allows the application to wake-up from Sleep, perform a few instructions using the INTOSC as the clock source and go back to Sleep without waiting for the primary oscillator to become stable.

Note: Executing a SLEEP instruction will abort the oscillator start-up time and will cause the OSTS bit of the OSCCON register to remain clear.

When the Oscillator module is configured for LP, XT or HS modes, the Oscillator Start-up Timer (OST) is enabled (see Section 3.4.1 “Oscillator Start-up Timer (OST)”). The OST will suspend program execution until 1024 oscillations are counted. Two-Speed Start-up mode minimizes the delay in code execution by operating from the internal oscillator as the OST is counting. When the OST count reaches 1024 and the OSTS bit of the OSCCON register is set, program execution switches to the external oscillator.

3.7.1 TWO-SPEED START-UP MODE CONFIGURATION

Two-Speed Start-up mode is configured by the following settings:

- IESO (of the Configuration Word register) = 1; Internal/External Switchover bit (Two-Speed Start-up mode enabled).
- SCS (of the OSCCON register) = 0.
- FOSC<2:0> bits in the Configuration Word register (CONFIG) configured for LP, XT or HS mode.

Two-Speed Start-up mode is entered after:

- Power-on Reset (POR) and, if enabled, after Power-up Timer (PWRT) has expired, or
- Wake-up from Sleep.

If the external clock oscillator is configured to be anything other than LP, XT or HS mode, then Two-Speed Start-up is disabled. This is because the external clock oscillator does not require any stabilization time after POR or an exit from Sleep.

3.7.2 TWO-SPEED START-UP SEQUENCE

1. Wake-up from Power-on Reset or Sleep.
2. Instructions begin execution by the internal oscillator at the frequency set in the IRCF<2:0> bits of the OSCCON register.
3. OST enabled to count 1024 clock cycles.
4. OST timed out, wait for falling edge of the internal oscillator.
5. OSTS is set.
6. System clock held low until the next falling edge of new clock (LP, XT or HS mode).
7. System clock is switched to external clock source.

Note: Executing a SLEEP instruction will abort the oscillator start-up time and will cause the OSTS bit of the OSCCON register to remain clear.
3.7.3 CHECKING TWO-SPEED CLOCK STATUS

Checking the state of the OSTS bit of the OSCCON register will confirm if the microcontroller is running from the external clock source, as defined by the FOSC<2:0> bits in the Configuration Word register (CONFIG), or the internal oscillator.

FIGURE 3-7: TWO-SPEED START-UP

![Diagram showing two-speed start-up](image-url)
3.8 Fail-Safe Clock Monitor

The Fail-Safe Clock Monitor (FSCM) allows the device to continue operating should the external oscillator fail. The FSCM can detect oscillator failure any time after the Oscillator Start-up Timer (OST) has expired. The FSCM is enabled by setting the FCMEN bit in the Configuration Word register (CONFIG). The FSCM is applicable to all external oscillator modes (LP, XT, HS, EC, RC and RCIO).

3.8.1 FAIL-SAFE DETECTION

The FSCM module detects a failed oscillator by comparing the external oscillator to the FSCM sample clock. The sample clock is generated by dividing the LFINTOSC by 64. See Figure 3-8. Inside the fail detector block is a latch. The external clock sets the latch on each falling edge of the external clock. The sample clock clears the latch on each rising edge of the sample clock. A failure is detected when an entire half-cycle of the sample clock elapses before the primary clock goes low.

3.8.2 FAIL-SAFE OPERATION

When the external clock fails, the FSCM switches the device clock to an internal clock source and sets the bit flag OSFIF of the PIR1 register. Setting this flag will generate an interrupt if the OSFIE bit of the PIE1 register is also set. The device firmware can then take steps to mitigate the problems that may arise from a failed clock. The system clock will continue to be sourced from the internal clock source until the device firmware successfully restarts the external oscillator and switches back to external operation.

The internal clock source chosen by the FSCM is determined by the IRCF<2:0> bits of the OSCCON register. This allows the internal oscillator to be configured before a failure occurs.

3.8.3 FAIL-SAFE CONDITION CLEARING

The Fail-Safe condition is cleared after a Reset, executing a SLEEP instruction or toggling the SCS bit of the OSCCON register. When the SCS bit is toggled, the OST is restarted. While the OST is running, the device continues to operate from the INTOSC selected in OSCCON. When the OST times out, the Fail-Safe condition is cleared and the device will be operating from the external clock source. The Fail-Safe condition must be cleared before the OSFIF flag can be cleared.

3.8.4 RESET OR WAKE-UP FROM SLEEP

The FSCM is designed to detect an oscillator failure after the Oscillator Start-up Timer (OST) has expired. The OST is used after waking up from Sleep and after any type of Reset. The OST is not used with the EC or RC Clock modes so that the FSCM will be active as soon as the Reset or wake-up has completed. When the FSCM is enabled, the Two-Speed Start-up is also enabled. Therefore, the device will always be executing code while the OST is operating.

Note: Due to the wide range of oscillator start-up times, the Fail-Safe circuit is not active during oscillator start-up (i.e., after exiting Reset or Sleep). After an appropriate amount of time, the user should check the OSTS bit of the OSCCON register to verify the oscillator start-up and that the system clock switchover has successfully completed.
FIGURE 3-9: FSCM TIMING DIAGRAM

Note: The system clock is normally at a much higher frequency than the sample clock. The relative frequencies in this example have been chosen for clarity.

TABLE 3-2: SUMMARY OF REGISTERS ASSOCIATED WITH CLOCK SOURCES

<table>
<thead>
<tr>
<th>Name</th>
<th>Bit 7</th>
<th>Bit 6</th>
<th>Bit 5</th>
<th>Bit 4</th>
<th>Bit 3</th>
<th>Bit 2</th>
<th>Bit 1</th>
<th>Bit 0</th>
<th>Value on: POR, BOR</th>
<th>Value on all other Resets(1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONFIG</td>
<td>CPD</td>
<td>CP</td>
<td>MCLRE</td>
<td>PWRTE</td>
<td>WDTE</td>
<td>FOSC2</td>
<td>FOSC1</td>
<td>FOSC0</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>OSCCON</td>
<td>—</td>
<td>IRCF2</td>
<td>IRCF1</td>
<td>IRCF0</td>
<td>OSTS</td>
<td>HTS</td>
<td>LTS</td>
<td>SCS</td>
<td>-110 x000</td>
<td>-110 x000</td>
</tr>
<tr>
<td>OSCTUNE</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>TUN4</td>
<td>TUN3</td>
<td>TUN2</td>
<td>TUN1</td>
<td>TUN0</td>
<td>---0 0000</td>
<td>---u uuuu</td>
</tr>
<tr>
<td>PIE1</td>
<td>EEIE</td>
<td>ADIE</td>
<td>CCP1IE</td>
<td>C2IE</td>
<td>C1IE</td>
<td>OSFIE</td>
<td>TMR2IE</td>
<td>TMR1IE</td>
<td>0000 0000</td>
<td>0000 0000</td>
</tr>
<tr>
<td>PIR1</td>
<td>EEIF</td>
<td>ADIF</td>
<td>CCP1IF</td>
<td>C2IF</td>
<td>C1IF</td>
<td>OSFIF</td>
<td>TMR2IF</td>
<td>TMR1IF</td>
<td>0000 0000</td>
<td>0000 0000</td>
</tr>
</tbody>
</table>

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

Note 1: Other (non Power-up) Resets include MCLR Reset and Watchdog Timer Reset during normal operation.

Note 2: See Configuration Word register (Register 12-1) for operation of all register bits.
4.0  I/O PORTS

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.

4.1  PORTA and the TRISA Registers

PORTA is a 6-bit wide, bidirectional port. The corresponding data direction register is TRISA (Register 4-2). Setting a TRISA bit (= 1) will make the corresponding PORTA pin an input (i.e., disable the output driver). Clearing a TRISA bit (= 0) will make the corresponding PORTA pin an output (i.e., enables output driver and puts 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 4-1 shows how to initialize PORTA.

Reading the PORTA register (Register 4-1) 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 MCLRE = 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 and CMCON0 registers must be initialized to configure an analog channel as a digital input. Pins configured as analog inputs will read ‘0’.

EXAMPLE 4-1:  INITIALIZING PORTA

Note: The ANSEL and CMCON0 registers must be initialized to configure an analog channel as a digital input. Pins configured as analog inputs will read ‘0’.

```assembly
BCF STATUS,RP0 ;Bank 0
CLRF PORTA ;Init PORTA
MOVLW 07h ;Set RA<2:0> to
MOVWF CMCON0 ;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
```

REGISTER 4-1:  PORTA: PORTA REGISTER

<table>
<thead>
<tr>
<th>U-0</th>
<th>U-0</th>
<th>R/W-x</th>
<th>R/W-0</th>
<th>R-x</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>bit 7</th>
<th>bit 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>—</td>
<td>—</td>
<td>RA5</td>
<td>RA4</td>
<td>RA3</td>
<td>RA2</td>
<td>RA1</td>
<td>RA0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

bit 7-6  Unimplemented: Read as ‘0’
bit 5-0  RA<5:0>: PORTA I/O Pin bit

1 = PORTA pin is > VIH
0 = PORTA pin is < VIH

REGISTER 4-2:  TRISA: PORTA TRI-STATE REGISTER

<table>
<thead>
<tr>
<th>U-0</th>
<th>U-0</th>
<th>R/W-1</th>
<th>R/W-1</th>
<th>R-1</th>
<th>R/W-1</th>
<th>R/W-1</th>
<th>R/W-1</th>
<th>bit 7</th>
<th>bit 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>—</td>
<td>—</td>
<td>TRISA5</td>
<td>TRISA4</td>
<td>TRISA3</td>
<td>TRISA2</td>
<td>TRISA1</td>
<td>TRISA0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

bit 7-6  Unimplemented: Read as ‘0’
bit 5-0  TRISA<5:0>: PORTA Tri-State Control bit

1 = PORTA pin configured as an input (tri-stated)
0 = PORTA pin configured as an output

Note 1: TRISA<3> always reads ‘1’.
2: TRISA<5:4> always reads ‘1’ in XT, HS and LP OSC modes.
4.2 Additional Pin Functions

Every PORTA pin on the PIC16F684 has an interrupt-on-change option and a weak pull-up option. RA0 has an Ultra Low-Power Wake-up option. The next three sections describe these functions.

4.2.1 ANSEL REGISTER

The ANSEL register is used to configure the Input mode of an I/O pin to analog. Setting the appropriate ANSEL bit high will cause all digital reads on the pin to be read as ‘0’ and allow analog functions on the pin to operate correctly.

The state of the ANSEL bits has no affect on digital output functions. A pin with TRIS clear and ANSEL set will still operate as a digital output, but the Input mode will be analog. This can cause unexpected behavior when executing read-modify-write instructions on the affected port.

4.2.2 WEAK PULL-UPS

Each of the PORTA pins, except RA3, has an individually configurable internal weak pull-up. Control bits WPUAx enable or disable each pull-up. Refer to Register 4-4. 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 RAPU bit of the OPTION register). A weak pull-up is automatically enabled for RA3 when configured as MCLR and disabled when RA3 is an I/O. There is no software control of the MCLR pull-up.

4.2.3 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 4-5. 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 (Register 2-3).

This interrupt can wake the device from Sleep. The user, in the Interrupt Service Routine, clears the interrupt by:

a) Any read or write of PORTA. This will end the mismatch condition, then,

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. The latch holding the last read value is not affected by a MCLR nor Brown-out Reset. After these resets, the RAIF flag will continue to be set if a mismatch is present.

Note: If a change on the I/O pin should occur when any PORTA operation is being executed, then the RAIF interrupt flag may not get set.

REGISTER 4-3: ANSEL: ANALOG SELECT REGISTER

<table>
<thead>
<tr>
<th>R/W-1</th>
<th>R/W-1</th>
<th>R/W-1</th>
<th>R/W-1</th>
<th>R/W-1</th>
<th>R/W-1</th>
<th>R/W-1</th>
<th>R/W-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANS7</td>
<td>ANS6</td>
<td>ANS5</td>
<td>ANS4</td>
<td>ANS3</td>
<td>ANS2</td>
<td>ANS1</td>
<td>ANS0</td>
</tr>
</tbody>
</table>

bit 7-0

Legend:
R = Readable bit  
W = Writable bit  
U = Unimplemented bit, read as ‘0’  
‘1’ = Bit is set  
‘0’ = Bit is cleared  
x = Bit is unknown

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.
REGISTER 4-4:  WPUA: WEAK PULL-UP PORTA REGISTER

<table>
<thead>
<tr>
<th>bit 7</th>
<th>bit 6</th>
<th>bit 5-4</th>
<th>bit 3</th>
<th>bit 2-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>—</td>
<td>—</td>
<td>WPUA5</td>
<td>—</td>
<td>WPUA4</td>
</tr>
<tr>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>WPUA2</td>
</tr>
<tr>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>WPUA1</td>
</tr>
<tr>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>WPUA0</td>
</tr>
</tbody>
</table>

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

- **bit 7-6**: Unimplemented: Read as ‘0’
- **bit 5-4**: WPUA<5:4>: Weak Pull-up Control bits
  - 1 = Pull-up enabled
  - 0 = Pull-up disabled
- **bit 3**: Unimplemented: Read as ‘0’
- **bit 2-0**: WPUA<2:0>: Weak Pull-up Control bits
  - 1 = Pull-up enabled
  - 0 = Pull-up disabled

**Note 1:** Global RAPU must be enabled for individual pull-ups to be enabled.

**Note 2:** The weak pull-up device is automatically disabled if the pin is in Output mode (TRISA = 0).

**Note 3:** The RA3 pull-up is enabled when configured as MCLR and disabled as an I/O in the Configuration Word.

**Note 4:** WPUA<5:4> always reads ‘1’ in XT, HS and LP OSC modes.

REGISTER 4-5:  IOCA: INTERRUPT-ON-CHANGE PORTA REGISTER

<table>
<thead>
<tr>
<th>bit 7</th>
<th>bit 6</th>
<th>bit 5-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>—</td>
<td>—</td>
<td>IOCA5</td>
</tr>
<tr>
<td>—</td>
<td>—</td>
<td>IOCA4</td>
</tr>
<tr>
<td>—</td>
<td>—</td>
<td>IOCA3</td>
</tr>
<tr>
<td>—</td>
<td>—</td>
<td>IOCA2</td>
</tr>
<tr>
<td>—</td>
<td>—</td>
<td>IOCA1</td>
</tr>
<tr>
<td>—</td>
<td>—</td>
<td>IOCA0</td>
</tr>
</tbody>
</table>

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

- **bit 7-6**: Unimplemented: Read as ‘0’
- **bit 5-0**: IOCA<5:0>: Interrupt-on-change PORTA Control bit
  - 1 = Interrupt-on-change enabled
  - 0 = Interrupt-on-change disabled

**Note 1:** Global Interrupt Enable (GIE) must be enabled for individual interrupts to be recognized.

**Note 2:** IOCA<5:4> always reads ‘1’ in XT, HS and LP OSC modes.
4.2.4 ULTRA LOW-POWER WAKE-UP

The Ultra Low-Power Wake-Up (ULPWU) on RA0 allows a slow falling voltage to generate an interrupt-on-change on RA0 without excess current consumption. The mode is selected by setting the ULPWUE bit of the PCON register. This enables a small current sink which can be used to discharge a capacitor on RA0.

To use this feature, the RA0 pin is first configured to output '1' to charge the capacitor. Then interrupt-on-change for RA0 is enabled, and RA0 is configured as an input. The ULPWUE bit is set to begin the discharge and a SLEEP instruction is performed. When the voltage on RA0 drops below VIL, the device will wake-up and execute the next instruction. If the GIE bit of the INTCON register is set, the device will call the interrupt service routine (0004h). See Section 4.2.3 “Interrupt-on-Change” and Section 12.4.3 “PORTA Interrupt-on-Change” for more information.

This feature provides a low-power technique for periodically waking up the device from Sleep. The time-out is dependent on the discharge time of the RC circuit on RA0. See Example 4-2 for initializing the Ultra Low-Power Wake-Up module.

The series resistor provides overcurrent protection for the RA0 pin and can allow for software calibration of the time-out (see Figure 4-1). A timer can be used to measure the charge time and discharge time of the capacitor. The charge time can then be adjusted to provide the desired interrupt delay. This technique will compensate for the affects of temperature, voltage and component accuracy. The Ultra Low-Power Wake-Up peripheral can also be configured as a simple Programmable Low Voltage Detect or temperature sensor.

Note: For more information, refer to AN879, “Using the Microchip Ultra Low-Power Wake-Up Module” Application Note (DS00879).

EXAMPLE 4-2: ULTRA LOW-POWER WAKE-UP INITIALIZATION

```
BANKSEL CMCON0 ;
MOVLW H'7' ;Turn off
MOVWF CMCON0 ;comparators
BANKSEL ANSEL ;
BCF ANSEL,0 ;RA0 to digital I/O
BCF TRISA,0 ;Output high to
BANKSEL PORTA ;
BSF PORTA,0 ;charge capacitor
CALL CapDelay ;
BANKSEL PCON ;
BSF PCON,ULPWUE ;Enable ULP Wake-up
BSF IOCA,0 ;Select RA0 IOC
BSF TRISA,0 ;RA0 to input
MOVLW B'10001000' ;Enable interrupt
MOVWF INTCON ; and clear flag
SLEEP ;Wait for IOC
```
4.2.5 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 ADC, refer to the appropriate section in this data sheet.

4.2.5.1 RA0/AN0/C1IN+/ICSPDAT/ULPWU

Figure 4-1 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 ADC
- an analog non-inverting input to the comparator
- In-Circuit Serial Programming data
- an analog input for the Ultra Low-Power Wake-Up

FIGURE 4-1: BLOCK DIAGRAM OF RA0

Note 1: Comparator mode and ANSEL determines Analog Input mode.

4.2.5.2 RA1/AN1/C1IN-/VREF/ICSPCLK

Figure 4-2 shows the diagram for this pin. The RA1 pin is configurable to function as one of the following:
- a general purpose I/O
- an analog input for the ADC
- an analog inverting input to the comparator
- a voltage reference input for the ADC
- In-Circuit Serial Programming clock
FIGURE 4-2: BLOCK DIAGRAM OF RA1

4.2.5.3 RA2/AN2/T0CKI/INT/C1OUT

Figure 4-3 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 ADC
- the clock input for TMR0
- an external edge triggered interrupt
- a digital output from Comparator 1

Note 1: Comparator mode and ANSEL determines Analog Input mode.

FIGURE 4-3: BLOCK DIAGRAM OF RA2

Note 1: Analog Input mode is generated by ANSEL.
4.2.5.4 RA3/MCLR/VPP

Figure 4-4 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 with weak pull-up

**FIGURE 4-4: BLOCK DIAGRAM OF RA3**

4.2.5.5 RA4/AN3/T1G/OSC2/CLKOUT

Figure 4-5 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 ADC
- a Timer1 gate (count enable)
- a crystal/resonator connection
- a clock output

**FIGURE 4-5: BLOCK DIAGRAM OF RA4**

---

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

2: With CLKOUT option.

3: Analog Input mode comes from ANSEL.
4.2.5.6 RA5/T1CKI/OSC1/CLKIN

Figure 4-6 shows the diagram for this pin. The RA5 pin is configurable to function as one of the following:

- a general purpose I/O
- a Timer1 clock input
- a crystal/resonator connection
- a clock input

FIGURE 4-6: BLOCK DIAGRAM OF RA5

Note 1: Timer1 LP Oscillator enabled.
### TABLE 4-1: SUMMARY OF REGISTERS ASSOCIATED WITH PORTA

<table>
<thead>
<tr>
<th>Name</th>
<th>Bit 7</th>
<th>Bit 6</th>
<th>Bit 5</th>
<th>Bit 4</th>
<th>Bit 3</th>
<th>Bit 2</th>
<th>Bit 1</th>
<th>Bit 0</th>
<th>Value on: POR, BOR</th>
<th>Value on all other Resets</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANSEL</td>
<td>ANS7</td>
<td>ANS6</td>
<td>ANS5</td>
<td>ANS4</td>
<td>ANS3</td>
<td>ANS2</td>
<td>ANS1</td>
<td>ANS0</td>
<td>1111 1111</td>
<td>1111 1111</td>
</tr>
<tr>
<td>CMCON0</td>
<td>C2OUT</td>
<td>C1OUT</td>
<td>C2INV</td>
<td>C1INV</td>
<td>CIS</td>
<td>CM2</td>
<td>CM1</td>
<td>CM0</td>
<td>0000 0000</td>
<td>0000 0000</td>
</tr>
<tr>
<td>PCON</td>
<td></td>
<td></td>
<td>ULPWJE</td>
<td>SB0REN</td>
<td></td>
<td></td>
<td>POR</td>
<td>BOR</td>
<td>--01 --qq</td>
<td>--0u --uu</td>
</tr>
<tr>
<td>INTCON</td>
<td>GIE</td>
<td>PEIE</td>
<td>T0IE</td>
<td>INTE</td>
<td>RAIE</td>
<td>T0IF</td>
<td>INTF</td>
<td>RAIF</td>
<td>0000 0000</td>
<td>0000 0000</td>
</tr>
<tr>
<td>IOCA</td>
<td></td>
<td></td>
<td>IOCA5</td>
<td>IOCA4</td>
<td>IOCA3</td>
<td>IOCA2</td>
<td>IOCA1</td>
<td>IOCA0</td>
<td>--00 0000</td>
<td>--00 0000</td>
</tr>
<tr>
<td>OPTION_REG</td>
<td>RAPU</td>
<td>INTEDG</td>
<td>T0CS</td>
<td>T0SE</td>
<td>PSA</td>
<td>PS2</td>
<td>PS1</td>
<td>PS0</td>
<td>1111 1111</td>
<td>1111 1111</td>
</tr>
<tr>
<td>PORTA</td>
<td></td>
<td></td>
<td>RA5</td>
<td>RA4</td>
<td>RA3</td>
<td>RA2</td>
<td>RA1</td>
<td>RA0</td>
<td>--x0 x000</td>
<td>--uu uu00</td>
</tr>
<tr>
<td>TRISA</td>
<td></td>
<td></td>
<td>TRISA5</td>
<td>TRISA4</td>
<td>TRISA3</td>
<td>TRISA2</td>
<td>TRISA1</td>
<td>TRISA0</td>
<td>--11 1111</td>
<td>--11 1111</td>
</tr>
<tr>
<td>WPUA</td>
<td></td>
<td></td>
<td>WPUA5</td>
<td>WPUA4</td>
<td></td>
<td></td>
<td>WPUA2</td>
<td>WPUA1</td>
<td>--11 -111</td>
<td>--11 -111</td>
</tr>
</tbody>
</table>

**Legend:**  
- x = unknown  
- u = unchanged  
- – = unimplemented locations read as ‘0’. Shaded cells are not used by PORTA.
4.3 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 (ADC) or Comparator. For specific information about individual functions such as the Enhanced CCP or the ADC, refer to the appropriate section in this data sheet.

Note: The ANSEL and CMCON0 registers must be initialized to configure an analog channel as a digital input. Pins configured as analog inputs will read ‘0’.

EXAMPLE 4-3: INITIALIZING PORTC

```assembly
BANKSEL PORTC ; Init PORTC
CLRF PORTC
MOVLW 07h ; Set RC<4,1:0> to digital I/O
MOVWF CMCON0
BANKSEL ANSEL ; digital I/O
CLRF ANSEL
MOVLW 0Ch ; Set RC<3:2> as inputs
MOVWF TRISC
BANKSEL PORTC ; and set RC<5:4,1:0> as outputs
BCF STATUS,RP0 ; Bank 0
```

REGISTER 4-6: PORTC: PORTC REGISTER

<table>
<thead>
<tr>
<th>U-0</th>
<th>U-0</th>
<th>R/W-x</th>
<th>R/W-x</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>—</td>
<td>—</td>
<td>RC5</td>
<td>RC4</td>
<td>RC3</td>
<td>RC2</td>
<td>RC1</td>
<td>RC0</td>
</tr>
</tbody>
</table>

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

bit 7-6 Unimplemented: Read as ‘0’
bit 5-0 RC<5:0>: PORTC I/O Pin bit
1 = PORTC pin is > VIH
0 = PORTC pin is < VIH

REGISTER 4-7: TRISC: PORTC TRI-STATE REGISTER

<table>
<thead>
<tr>
<th>U-0</th>
<th>U-0</th>
<th>R/W-1</th>
<th>R/W-1</th>
<th>R/W-1</th>
<th>R/W-1</th>
<th>R/W-1</th>
<th>R/W-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>—</td>
<td>—</td>
<td>TRISC5</td>
<td>TRISC4</td>
<td>TRISC3</td>
<td>TRISC2</td>
<td>TRISC1</td>
<td>TRISC0</td>
</tr>
</tbody>
</table>

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

bit 7-6 Unimplemented: Read as ‘0’
bit 5-0 TRISC<5:0>: PORTC Tri-State Control bit
1 = PORTC pin configured as an input (tri-stated)
0 = PORTC pin configured as an output
4.3.1 RC0/AN4/C2IN+
The RC0 is configurable to function as one of the following:
• a general purpose I/O
• an analog input for the ADC
• an analog non-inverting input to the comparator

4.3.2 RC1/AN5/C2IN-
The RC1 is configurable to function as one of the following:
• a general purpose I/O
• an analog input for the ADC
• an analog inverting input to the comparator

FIGURE 4-7: BLOCK DIAGRAM OF RC0 AND RC1

4.3.3 RC2/AN6/P1D
The RC2 is configurable to function as one of the following:
• a general purpose I/O
• an analog input for the ADC
• a digital output from the Enhanced CCP

4.3.4 RC3/AN7/P1C
The RC3 is configurable to function as one of the following:
• a general purpose I/O
• an analog input for the ADC
• a digital output from the Enhanced CCP

FIGURE 4-8: BLOCK DIAGRAM OF RC2 AND RC3

Note 1: Analog Input mode comes from ANSEL.

Note 1: Analog Input mode comes from ANSEL.
4.3.5 RC4/C2OUT/P1B
The RC4 is configurable to function as one of the following:
- a general purpose I/O
- a digital output from the comparator
- a digital output from the Enhanced CCP

**Note:** Enabling both C2OUT and P1B will cause a conflict on RC4 and create unpredictable results. Therefore, if C2OUT is enabled, the ECCP cannot be used in Half-Bridge or Full-Bridge mode and vice-versa.

**FIGURE 4-9: BLOCK DIAGRAM OF RC4**

![Block Diagram of RC4]

**TABLE 4-2: SUMMARY OF REGISTERS ASSOCIATED WITH PORTC**

<table>
<thead>
<tr>
<th>Name</th>
<th>Bit 7</th>
<th>Bit 6</th>
<th>Bit 5</th>
<th>Bit 4</th>
<th>Bit 3</th>
<th>Bit 2</th>
<th>Bit 1</th>
<th>Bit 0</th>
<th>Value on: POR, BOR</th>
<th>Value on all other Resets</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANSEL</td>
<td>ANS7</td>
<td>ANS6</td>
<td>ANS5</td>
<td>ANS4</td>
<td>ANS3</td>
<td>ANS2</td>
<td>ANS1</td>
<td>ANS0</td>
<td>1111 1111</td>
<td>1111 1111</td>
</tr>
<tr>
<td>CMCON0</td>
<td>C2OUT</td>
<td>C1OUT</td>
<td>C2INV</td>
<td>C1INV</td>
<td>CIS</td>
<td>CM2</td>
<td>CM1</td>
<td>CM0</td>
<td>0000 0000</td>
<td>0000 0000</td>
</tr>
<tr>
<td>PORTC</td>
<td>—</td>
<td>—</td>
<td>RC5</td>
<td>RC4</td>
<td>RC3</td>
<td>RC2</td>
<td>RC1</td>
<td>RC0</td>
<td>--xx 0000</td>
<td>--xx uu00</td>
</tr>
<tr>
<td>TRISC</td>
<td>—</td>
<td>—</td>
<td>TRISC5</td>
<td>TRISC4</td>
<td>TRISC3</td>
<td>TRISC2</td>
<td>TRISC1</td>
<td>TRISC0</td>
<td>--11 1111</td>
<td>--11 1111</td>
</tr>
</tbody>
</table>

**Legend:**
- x = unknown, u = unchanged, - = unimplemented locations read as ‘0’. Shaded cells are not used by PORTC.
5.0  TIMER0 MODULE

The Timer0 module is an 8-bit timer/counter with the following features:

- 8-bit timer/counter register (TMR0)
- 8-bit prescaler (shared with Watchdog Timer)
- Programmable internal or external clock source
- Programmable external clock edge selection
- Interrupt on overflow

Figure 5-1 is a block diagram of the Timer0 module.

5.1  Timer0 Operation

When used as a timer, the Timer0 module can be used as either an 8-bit timer or an 8-bit counter.

5.1.1 8-BIT TIMER MODE

When used as a timer, the Timer0 module will increment every instruction cycle (without prescaler). Timer mode is selected by clearing the T0CS bit of the OPTION register to ‘0’.

When TMR0 is written, the increment is inhibited for two instruction cycles immediately following the write.

Note: The value written to the TMR0 register can be adjusted, in order to account for the two instruction cycle delay when TMR0 is written.

5.1.2 8-BIT COUNTER MODE

When used as a counter, the Timer0 module will increment on every rising or falling edge of the T0CKI pin. The incrementing edge is determined by the T0SE bit of the OPTION register. Counter mode is selected by setting the T0CS bit of the OPTION register to ‘1’.

FIGURE 5-1:  BLOCK DIAGRAM OF THE TIMER0/WDT PRESCALER

Note 1: T0SE, T0CS, PSA, PS<2:0> are bits in the OPTION register.
Note 2: SWDTEN and WDTPS<3:0> are bits in the WDTCON register.
Note 3: WDTE bit is in the Configuration Word register.
5.1.3 SOFTWARE PROGRAMMABLE PRESCALER

A single software programmable prescaler is available for use with either Timer0 or the Watchdog Timer (WDT), but not both simultaneously. The prescaler assignment is controlled by the PSA bit of the OPTION register. To assign the prescaler to Timer0, the PSA bit must be cleared to a '0'.

There are 8 prescaler options for the Timer0 module ranging from 1:2 to 1:256. The prescale values are selectable via the PS<2:0> bits of the OPTION register. In order to have a 1:1 prescaler value for the Timer0 module, the prescaler must be assigned to the WDT module.

The prescaler is not readable or writable. When assigned to the Timer0 module, all instructions writing to the TMR0 register will clear the prescaler.

When the prescaler is assigned to WDT, a `CLRWDT` instruction will clear the prescaler along with the WDT.

5.1.3.1 Switching Prescaler Between Timer0 and WDT Modules

As a result of having the prescaler assigned to either Timer0 or the WDT, it is possible to generate an unintended device Reset when switching prescaler values. When changing the prescaler assignment from Timer0 to the WDT module, the instruction sequence shown in Example 5-1 must be executed.

**EXAMPLE 5-1: CHANGING PRESCALER (TIMER0 → WDT)**

```assembly
BANKSEL TMR0 ;
CLRWDT ;Clear WDT
CLRF TMR0 ;Clear TMR0 and
; prescaler
BANKSEL OPTION_REG ;
BSF OPTION_REG,PSA ;Select WDT
CLRWDT ;

MOVLW b'11110000' ;Mask prescaler
ANDWF OPTION_REG,W ;bits
IORLW b'00000101' ;Set WDT prescaler
MOVWF OPTION_REG ; to 1:16
```

When changing the prescaler assignment from the WDT to the Timer0 module, the following instruction sequence must be executed (see Example 5-2).

**EXAMPLE 5-2: CHANGING PRESCALER (WDT → TIMER0)**

```assembly
CLRWDT ;Clear WDT and
; prescaler
BANKSEL OPTION_REG ;
MOVLW b'11110000' ;Mask TMR0 select and
ANDWF OPTION_REG,W ; prescaler bits
IORLW b'00000111' ;Set prescale to 1:16
MOVWF OPTION_REG ;
```

5.1.4 TIMER0 INTERRUPT

Timer0 will generate an interrupt when the TMR0 register overflows from FFh to 00h. The T0IF interrupt flag bit of the INTCON register is set every time the TMR0 register overflows, regardless of whether or not the Timer0 interrupt is enabled. The T0IF bit must be cleared in software. The Timer0 interrupt enable is the T0IE bit of the INTCON register.

**Note:** The Timer0 interrupt cannot wake the processor from Sleep since the timer is frozen during Sleep.

5.1.5 USING TIMER0 WITH AN EXTERNAL CLOCK

When Timer0 is in Counter mode, the synchronization of the T0CKI input and the Timer0 register is accomplished by sampling the prescaler output on the Q2 and Q4 cycles of the internal phase clocks. Therefore, the high and low periods of the external clock source must meet the timing requirements as shown in Section 15.0 “Electrical Specifications”.
### REGISTER 5-1: OPTION_REG: OPTION REGISTER

<table>
<thead>
<tr>
<th>R/W-1</th>
<th>R/W-1</th>
<th>R/W-1</th>
<th>R/W-1</th>
<th>R/W-1</th>
<th>R/W-1</th>
<th>R/W-1</th>
<th>R/W-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAPU</td>
<td>INTEDG</td>
<td>T0CS</td>
<td>T0SE</td>
<td>PSA</td>
<td>PS2</td>
<td>PS1</td>
<td>PS0</td>
</tr>
</tbody>
</table>

bit 7  
**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 INT pin  
0 = Interrupt on falling edge of INT pin

bit 5  
**T0CS**: TMR0 Clock Source Select bit  
1 = Transition on T0CKI pin  
0 = Internal instruction cycle clock (FOSC/4)

bit 4  
**T0SE**: TMR0 Source Edge Select bit  
1 = Increment on high-to-low transition on T0CKI pin  
0 = Increment on low-to-high transition on 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  
**PS<2:0>**: Prescaler Rate Select bits

<table>
<thead>
<tr>
<th>BIT VALUE</th>
<th>TMR0 RATE</th>
<th>WDT RATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>1 : 2</td>
<td>1 : 1</td>
</tr>
<tr>
<td>001</td>
<td>1 : 4</td>
<td>1 : 2</td>
</tr>
<tr>
<td>010</td>
<td>1 : 8</td>
<td>1 : 4</td>
</tr>
<tr>
<td>011</td>
<td>1 : 16</td>
<td>1 : 8</td>
</tr>
<tr>
<td>100</td>
<td>1 : 32</td>
<td>1 : 16</td>
</tr>
<tr>
<td>101</td>
<td>1 : 64</td>
<td>1 : 32</td>
</tr>
<tr>
<td>110</td>
<td>1 : 128</td>
<td>1 : 64</td>
</tr>
<tr>
<td>111</td>
<td>1 : 256</td>
<td>1 : 128</td>
</tr>
</tbody>
</table>

**Note 1:** A dedicated 16-bit WDT postscaler is available. See Section 12.6 “Watchdog Timer (WDT)” for more information.

### TABLE 5-1: SUMMARY OF REGISTERS ASSOCIATED WITH TIMERO

<table>
<thead>
<tr>
<th>Name</th>
<th>Bit 7</th>
<th>Bit 6</th>
<th>Bit 5</th>
<th>Bit 4</th>
<th>Bit 3</th>
<th>Bit 2</th>
<th>Bit 1</th>
<th>Bit 0</th>
<th>Value on: POR, BOR</th>
<th>Value on all other Resets</th>
</tr>
</thead>
<tbody>
<tr>
<td>TMR0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>xxxxx</td>
<td>uuuuuuuuuuuuuuuuuuuuuu</td>
<td></td>
</tr>
<tr>
<td>INTCON</td>
<td>GIE</td>
<td>PEIE</td>
<td>TOIE</td>
<td>INTE</td>
<td>RAIE</td>
<td>T0IF</td>
<td>INTF</td>
<td>RAIF</td>
<td>0000 0000</td>
<td>0000 0000</td>
</tr>
<tr>
<td>OPTION_REG</td>
<td>RAPU</td>
<td>INTEDG</td>
<td>T0CS</td>
<td>T0SE</td>
<td>PSA</td>
<td>PS2</td>
<td>PS1</td>
<td>PS0</td>
<td>1111 1111</td>
<td>1111 1111</td>
</tr>
<tr>
<td>TRISA</td>
<td>—</td>
<td>—</td>
<td>TRISA5</td>
<td>TRISA4</td>
<td>TRISA3</td>
<td>TRISA2</td>
<td>TRISA1</td>
<td>TRISA0</td>
<td>--11 1111</td>
<td>--11 1111</td>
</tr>
</tbody>
</table>

**Legend:**  
- Unimplemented locations, read as '0', u = unchanged, x = unknown. Shaded cells are not used by the Timer0 module.
6.0  TIMER1 MODULE WITH GATE CONTROL

The Timer1 module is a 16-bit timer/counter with the following features:
- 16-bit timer/counter register pair (TMR1H:TMR1L)
- Programmable internal or external clock source
- 3-bit prescaler
- Optional LP oscillator
- Synchronous or asynchronous operation
- Timer1 gate (count enable) via comparator or T1G pin
- Interrupt on overflow
- Wake-up on overflow (external clock, Asynchronous mode only)
- Time base for the Capture/Compare function
- Special Event Trigger (with ECCP)
- Comparator output synchronization to Timer1 clock

Figure 6-1 is a block diagram of the Timer1 module.

6.1  Timer1 Operation

The Timer1 module is a 16-bit incrementing counter which is accessed through the TMR1H:TMR1L register pair. Writes to TMR1H or TMR1L directly update the counter.

When used with an internal clock source, the module is a timer. When used with an external clock source, the module can be used as either a timer or counter.

6.2  Clock Source Selection

The TMR1CS bit of the T1CON register is used to select the clock source. When TMR1CS = 0, the clock source is Fosc/4. When TMR1CS = 1, the clock source is supplied externally.

<table>
<thead>
<tr>
<th>Clock Source</th>
<th>TMR1CS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fosc/4</td>
<td>0</td>
</tr>
<tr>
<td>T1CKI pin</td>
<td>1</td>
</tr>
</tbody>
</table>

Note 1: Timer1 register increments on rising edge.
Note 2: Synchronize does not operate while in Sleep.
6.2.1 INTERNAL CLOCK SOURCE
When the internal clock source is selected, the TMR1H:TMR1L register pair will increment on multiples of Fosc as determined by the Timer1 prescaler.

6.2.2 EXTERNAL CLOCK SOURCE
When the external clock source is selected, the Timer1 module may work as a timer or a counter.

When counting, 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.

If an external clock oscillator is needed (and the microcontroller is using the INTOSC without 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 after any one or more of the following conditions:
- Timer1 enabled after POR reset
- Write to TMR1H or TMR1L
- Timer1 is disabled
- Timer1 is disabled (TMR1ON 0) when T1CKI is high then Timer1 is enabled (TMR1ON=1) when T1CKI is low.

See Figure 6-2

6.3 Timer1 Prescaler
Timer1 has four prescaler options allowing 1, 2, 4 or 8 divisions of the clock input. The T1CKPS bits of the T1CON register 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.

6.4 Timer1 Oscillator
A low-power 32.768 kHz crystal oscillator is built-in between pins OSC1 (input) and OSC2 (amplifier output). The oscillator is enabled by setting the T1OSCEN control bit of the T1CON register. The oscillator will continue to run during Sleep.

The Timer1 oscillator is shared with the system LP oscillator. Thus, Timer1 can use this mode only when the primary system clock is derived from the internal oscillator or when the oscillator is in the LP mode. 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 bits read as '0' and TRISA5 and TRISA4 bits read as '1'.

6.5 Timer1 Operation in Asynchronous Counter Mode
If control bit T1SYNC of the T1CON register is set, the external clock source is not synchronized. The timer increments asynchronously to the internal phase clocks. If external clock source is selected then 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 (see Section 6.5.1 “Reading and Writing Timer1 in Asynchronous Counter Mode”).

Note: When switching from synchronous to asynchronous operation, it is possible to skip an increment. When switching from asynchronous to synchronous operation, it is possible to produce an additional increment.

6.5.1 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 or carry out of TMR1L to TMR1H 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 TMR1H:TMR1L register pair.
6.6 Timer1 Gate

Timer1 gate source is software configurable to be the T1G pin or the output of Comparator 2. This allows the device to directly time external events using T1G or analog events using Comparator 2. See the CMCON1 register (Register 8-2) for selecting the Timer1 gate source. This feature can simplify the software for a Delta-Sigma A/D converter and many other applications. For more information on Delta-Sigma A/D converters, see the Microchip web site (www.microchip.com).

Timer1 gate can be inverted using the T1GINV bit of the T1CON register, whether it originates from the T1G pin or Comparator 2 output. This configures Timer1 to measure either the active-high or active-low time between events.

6.7 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 of the PIR1 register is set. To enable the interrupt on rollover, you must set these bits:

- TMR1ON bit or the T1CON register
- TMR1IE bit of the PIE1 register
- PEIE bit of the INTCON register
- GIE bit of the INTCON register

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

6.8 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 set up the timer to wake the device:

- TMR1ON bit of the T1CON register must be set
- TMR1IE bit of the PIE1 register must be set
- PEIE bit of the INTCON register must be set
- T1SYNC bit of the T1CON register
- TMR1CS bit of the T1CON register
- T1OSCEN bit of the T1CON register (can be set)

The device will wake-up on an overflow and execute the next instruction. If the GIE bit of the INTCON register is set, the device will call the Interrupt Service Routine (0004h).

6.9 ECCP Capture/Compare Time Base

The ECCP module uses the TMR1H:TMR1L register pair as the time base when operating in Capture or Compare mode.

In Capture mode, the value in the TMR1H:TMR1L register pair is copied into the CCPR1H:CCPR1L register pair on a configured event.

In Compare mode, an event is triggered when the value CCPR1H:CCPR1L register pair matches the value in the TMR1H:TMR1L register pair. This event can be a Special Event Trigger.

For more information, see Section 11.0 “Enhanced Capture/Compare/PWM (With Auto-Shutdown and Dead Band) Module”.

6.10 ECCP Special Event Trigger

When the ECCP is configured to trigger a special event, the trigger will clear the TMR1H:TMR1L register pair. This special event does not cause a Timer1 interrupt. The ECCP module may still be configured to generate a ECCP interrupt.

In this mode of operation, the CCPR1H:CCPR1L register pair effectively becomes the period register for Timer1.

Timer1 should be synchronized to the Fosc to utilize the Special Event Trigger. Asynchronous operation of Timer1 can cause a Special Event Trigger to be missed.

In the event that a write to TMR1H or TMR1L coincides with a Special Event Trigger from the ECCP, the write will take precedence.

For more information, see Section 11.2.4 “Special Event Trigger”.

6.11 Comparator Synchronization

The same clock used to increment Timer1 can also be used to synchronize the comparator output. This feature is enabled in the Comparator module.

When using the comparator for Timer1 gate, the comparator output should be synchronized to Timer1. This ensures Timer1 does not miss an increment if the comparator changes.

For more information, see Section 8.9 “Synchronizing Comparator C2 Output to Timer1”.

Note:

- TMR1GE bit of the T1CON register must be set to use either T1G or C2OUT as the Timer1 gate source. See the CMCON1 register (Register 8-2) for more information on selecting the Timer1 gate source.
- The TMR1H:TMR1L register pair and the TMR1IF bit should be cleared before enabling interrupts.
FIGURE 6-2: TIMER1 INCREMENTING EDGE

Note 1: Arrows indicate counter increments.
Note 2: In Counter mode, a falling edge must be registered by the counter prior to the first incrementing rising edge of the clock.

6.12 Timer1 Control Register

The Timer1 Control register (T1CON), shown in Register 6-1, is used to control Timer1 and select the various features of the Timer1 module.

REGISTER 6-1: T1CON: TIMER 1 CONTROL REGISTER

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Value at POR</th>
<th>Setting by software</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>T1GINV: Timer1 Gate Invert bit(1)</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1 = Timer1 gate is active high (Timer1 counts when gate is high)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 = Timer1 gate is active low (Timer1 counts when gate is low)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>TMR1GE: Timer1 Gate Enable bit(2)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>If TMR1ON = 0:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>This bit is ignored</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>If TMR1ON = 1:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 = Timer1 is on if Timer1 gate is not active</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 = Timer1 is on</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5-4</td>
<td>T1CKPS&lt;1:0&gt;: Timer1 Input Clock Prescale Select bits</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>11 = 1:8 Prescale Value</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>10 = 1:4 Prescale Value</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>01 = 1:2 Prescale Value</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>00 = 1:1 Prescale Value</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>T1OSCEN: LP Oscillator Enable Control bit</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>If INTOSC without CLKOUT oscillator is active:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 = LP oscillator is enabled for Timer1 clock</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 = LP oscillator is off</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Else:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>This bit is ignored</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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 6-1: SUMMARY OF REGISTERS ASSOCIATED WITH TIMER1

<table>
<thead>
<tr>
<th>Name</th>
<th>Bit 7</th>
<th>Bit 6</th>
<th>Bit 5</th>
<th>Bit 4</th>
<th>Bit 3</th>
<th>Bit 2</th>
<th>Bit 1</th>
<th>Bit 0</th>
<th>Value on: POR, BOR</th>
<th>Value on all other Resets</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMCON1</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—— —10</td>
</tr>
<tr>
<td>INTCON</td>
<td>GIE</td>
<td>PEIE</td>
<td>T0IE</td>
<td>INTE</td>
<td>RAIE</td>
<td>T0IF</td>
<td>INTF</td>
<td>RAIF</td>
<td>0000 0000</td>
<td>0000 0000</td>
</tr>
<tr>
<td>PIE1</td>
<td>EEIE</td>
<td>ADIE</td>
<td>CCP1IE</td>
<td>C2IE</td>
<td>C1IE</td>
<td>OSEIE</td>
<td>TMRIE2</td>
<td>TMRIE1</td>
<td>0000 0000</td>
<td>0000 0000</td>
</tr>
<tr>
<td>PIR1</td>
<td>EEIF</td>
<td>ADIF</td>
<td>CCP1IF</td>
<td>C2IF</td>
<td>C1IF</td>
<td>OSEIF</td>
<td>TMRIE2</td>
<td>TMRIE1</td>
<td>0000 0000</td>
<td>0000 0000</td>
</tr>
<tr>
<td>TMR1H</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>xxxxxx</td>
<td>xxxxxxxxxx</td>
</tr>
<tr>
<td>TMR1L</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>xxxxxx</td>
<td>xxxxxxxxxx</td>
</tr>
<tr>
<td>T1CON</td>
<td>—</td>
<td>—</td>
<td>T1GINV</td>
<td>TMR1GE</td>
<td>T1CKPS1</td>
<td>T1CKPS0</td>
<td>T1OSCEN</td>
<td>TSYNC</td>
<td>0000 0000</td>
<td>0000 0000</td>
</tr>
</tbody>
</table>

Legend:  
\( x \) = unknown, \( u \) = unchanged, \( — \) = unimplemented, read as ‘0’. Shaded cells are not used by the Timer1 module.
7.0 TIMER2 MODULE

The Timer2 module is an eight-bit timer with the following features:

- 8-bit timer register (TMR2)
- 8-bit period register (PR2)
- Interrupt on TMR2 match with PR2
- Software programmable prescaler (1:1, 1:4, 1:16)
- Software programmable postscaler (1:1 to 1:16)

See Figure 7-1 for a block diagram of Timer2.

7.1 Timer2 Operation

The clock input to the Timer2 module is the system instruction clock (FOSC/4). The clock is fed into the Timer2 prescaler, which has prescale options of 1:1, 1:4 or 1:16. The output of the prescaler is then used to increment the TMR2 register.

The values of TMR2 and PR2 are constantly compared to determine when they match. TMR2 will increment from 00h until it matches the value in PR2. When a match occurs, two things happen:

- TMR2 is reset to 00h on the next increment cycle.
- The Timer2 postscaler is incremented

The match output of the Timer2/PR2 comparator is fed into the Timer2 postscaler. The postscaler has postscale options of 1:1 to 1:16 inclusive. The output of the Timer2 postscaler is used to set the TMR2IF interrupt flag bit in the PIR1 register.

The TMR2 and PR2 registers are both fully readable and writable. On any Reset, the TMR2 register is set to 00h and the PR2 register is set to FFh.

Timer2 is turned on by setting the TMR2ON bit in the T2CON register to a '1'. Timer2 is turned off by clearing the TMR2ON bit to a '0'.

The Timer2 prescaler is controlled by the T2CKPS bits in the T2CON register. The Timer2 postscaler is controlled by the TOUTPS bits in the T2CON register. The prescaler and postscaler counters are cleared when:

- A write to TMR2 occurs.
- A write to T2CON occurs.
- Any device Reset occurs (Power-on Reset, MCLR Reset, Watchdog Timer Reset, or Brown-out Reset).

**Note:** TMR2 is not cleared when T2CON is written.

---

**FIGURE 7-1: TIMER2 BLOCK DIAGRAM**

![Timer2 Block Diagram](image-url)
### REGISTER 7-1: T2CON: TIMER 2 CONTROL REGISTER

<table>
<thead>
<tr>
<th>Bit 7</th>
<th>Bit 6-3</th>
<th>Value on POR, BOR</th>
<th>Value on all other Resets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unimplemented</td>
<td>TOUTPS&lt;3:0&gt;: Timer2 Output Postscaler Select bits</td>
<td>0000 = 1:1 Postscaler</td>
<td>0000 0000 0000 0000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0001 = 1:2 Postscaler</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0010 = 1:3 Postscaler</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0011 = 1:4 Postscaler</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0100 = 1:5 Postscaler</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0101 = 1:6 Postscaler</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0110 = 1:7 Postscaler</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0111 = 1:8 Postscaler</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1000 = 1:9 Postscaler</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1001 = 1:10 Postscaler</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1010 = 1:11 Postscaler</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1011 = 1:12 Postscaler</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1100 = 1:13 Postscaler</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1101 = 1:14 Postscaler</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1110 = 1:15 Postscaler</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1111 = 1:16 Postscaler</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bit 2</th>
<th>Value on POR, BOR</th>
<th>Value on all other Resets</th>
</tr>
</thead>
<tbody>
<tr>
<td>TMR2ON: Timer2 On bit</td>
<td>1 = Timer2 is on</td>
<td>0000 0000 0000 0000</td>
</tr>
<tr>
<td></td>
<td>0 = Timer2 is off</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bit 1-0</th>
<th>Value on POR, BOR</th>
<th>Value on all other Resets</th>
</tr>
</thead>
<tbody>
<tr>
<td>T2CKPS&lt;1:0&gt;: Timer2 Clock Prescale Select bits</td>
<td>00 = Prescaler is 1</td>
<td>0000 0000 0000 0000</td>
</tr>
<tr>
<td></td>
<td>01 = Prescaler is 4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1x = Prescaler is 16</td>
<td></td>
</tr>
</tbody>
</table>

### TABLE 7-1: SUMMARY OF ASSOCIATED TIMER2 REGISTERS

<table>
<thead>
<tr>
<th>Name</th>
<th>Bit 7</th>
<th>Bit 6</th>
<th>Bit 5</th>
<th>Bit 4</th>
<th>Bit 3</th>
<th>Bit 2</th>
<th>Bit 1</th>
<th>Bit 0</th>
<th>Value on POR, BOR</th>
<th>Value on all other Resets</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTCON</td>
<td>GIE</td>
<td>PEIE</td>
<td>TOIE</td>
<td>INTE</td>
<td>RAIE</td>
<td>T0IF</td>
<td>INTF</td>
<td>RAIF</td>
<td>0000 0000 0000 0000</td>
<td></td>
</tr>
<tr>
<td>PIE1</td>
<td>EEIE</td>
<td>ADIE</td>
<td>CCP1IE</td>
<td>C2IE</td>
<td>C1IE</td>
<td>OSFIE</td>
<td>TMR2IE</td>
<td>TMR1IE</td>
<td>0000 0000 0000 0000</td>
<td></td>
</tr>
<tr>
<td>PIR1</td>
<td>EEIF</td>
<td>ADIF</td>
<td>CCP1IF</td>
<td>C2IF</td>
<td>C1IF</td>
<td>OSFIF</td>
<td>TMR2IF</td>
<td>TMR1IF</td>
<td>0000 0000 0000 0000</td>
<td></td>
</tr>
<tr>
<td>PR2</td>
<td>Timer2 Module Period Register</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1111 1111 1111 1111</td>
<td></td>
</tr>
<tr>
<td>TMR2</td>
<td>Holding Register for the 8-bit TMR2 Register</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0000 0000 0000 0000</td>
<td></td>
</tr>
<tr>
<td>T2CON</td>
<td>TOUTPS3</td>
<td>TOUTPS2</td>
<td>TOUTPS1</td>
<td>TOUTPS0</td>
<td>TMR2ON</td>
<td>T2CKPS1</td>
<td>T2CKPS0</td>
<td></td>
<td>-000 0000 -000 0000</td>
<td></td>
</tr>
</tbody>
</table>

**Legend:**
- x = unknown, u = unchanged, — = unimplemented read as ‘0’. Shaded cells are not used for Timer2 module.
8.0 COMPARATOR MODULE

Comparators are used to interface analog circuits to a digital circuit by comparing two analog voltages and providing a digital indication of their relative magnitudes. The comparators are very useful mixed signal building blocks because they provide analog functionality independent of the device program execution. The analog Comparator module includes the following features:

- Dual comparators
- Multiple comparator configurations
- Comparator outputs are available internally/externally
- Programmable output polarity
- Interrupt-on-change
- Wake-up from Sleep
- Timer1 gate (count enable)
- Output synchronization to Timer1 clock input
- Programmable voltage reference

**Note:** Only Comparator C2 can be linked to Timer1.

8.1 Comparator Overview

A comparator is shown in Figure 8-1 along with the relationship between the analog input levels and the digital output. When the analog voltage at VIN+ is less than the analog voltage at VIN-, the output of the comparator is a digital low level. When the analog voltage at VIN+ is greater than the analog voltage at VIN-, the output of the comparator is a digital high level.

**FIGURE 8-1: SINGLE COMPARATOR**

This device contains two comparators as shown in Figure 8-2 and Figure 8-3. The comparators are not independently configurable.
FIGURE 8-2: COMPARATOR C1 OUTPUT BLOCK DIAGRAM

Note 1: Q1 and Q3 are phases of the four-phase system clock (Fosc).
2: Q1 is held high during Sleep mode.

FIGURE 8-3: COMPARATOR C2 OUTPUT BLOCK DIAGRAM

Note 1: Comparator output is latched on falling edge of Timer1 clock source.
2: Q1 and Q3 are phases of the four-phase system clock (Fosc).
3: Q1 is held high during Sleep mode.
8.1.1 ANALOG INPUT CONNECTION CONSIDERATIONS

A simplified circuit for an analog input is shown in Figure 8-4. Since the analog input pins share their connection with a digital input, they have reverse biased ESD protection 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. Also, any external component connected to an analog input pin, such as a capacitor or a Zener diode, should have very little leakage current to minimize inaccuracies introduced.

Note 1: When reading a PORT register, all pins configured as analog inputs will read as a ‘0’. Pins configured as digital inputs will convert as an analog input, according to the input specification.

2: Analog levels on any pin defined as a digital input, may cause the input buffer to consume more current than is specified.

FIGURE 8-4: ANALOG INPUT MODEL

Legend:
- CPIN = Input Capacitance
- ILEAKAGE = Leakage Current at the pin due to various junctions
- RIC = Interconnect Resistance
- RS = Source Impedance
- VA = Analog Voltage
- VT = Threshold Voltage
8.2 Comparator Configuration

There are eight modes of operation for the comparator. The CM<2:0> bits of the CMCON0 register are used to select these modes as shown in Figure 8-5. I/O lines change as a function of the mode and are designated as follows:

- Analog function (A): digital input buffer is disabled
- Digital function (D): comparator digital output, overrides port function
- Normal port function (I/O): independent of comparator

The port pins denoted as “A” will read as a ‘0’ regardless of the state of the I/O pin or the I/O control TRIS bit. Pins used as analog inputs should also have the corresponding TRIS bit set to ’1’ to disable the digital output driver. Pins denoted as “D” should have the corresponding TRIS bit set to ‘0’ to enable the digital output driver.

**Note:** Comparator interrupts should be disabled during a Comparator mode change to prevent unintended interrupts.

---

### FIGURE 8-5: COMPARATOR I/O OPERATING MODES

<table>
<thead>
<tr>
<th>Mode Description</th>
<th>CM&lt;2:0&gt; Setting</th>
<th>Diagram</th>
</tr>
</thead>
<tbody>
<tr>
<td>Comparators Reset (POR Default Value)</td>
<td>000</td>
<td><img src="image" alt="Comparator Reset Diagram" /></td>
</tr>
<tr>
<td>C1IN- A, C1IN+ A, C2IN- A, C2IN+ A, CM&lt;2:0&gt; = 000</td>
<td></td>
<td><img src="image" alt="Comparator Diagram" /></td>
</tr>
<tr>
<td>Two Independent Comparators</td>
<td>100</td>
<td><img src="image" alt="Two Independent Comparators Diagram" /></td>
</tr>
<tr>
<td>C1IN- A, C1IN+ A, C2IN- A, C2IN+ A, CM&lt;2:0&gt; = 100</td>
<td></td>
<td><img src="image" alt="Comparator Diagram" /></td>
</tr>
<tr>
<td>Three Inputs Multiplexed to Two Comparators</td>
<td>001</td>
<td><img src="image" alt="Three Inputs Multiplexed to Two Comparators Diagram" /></td>
</tr>
<tr>
<td>C1IN- A, C1IN+ A, C2IN- A, C2IN+ A, CM&lt;2:0&gt; = 001</td>
<td></td>
<td><img src="image" alt="Comparator Diagram" /></td>
</tr>
<tr>
<td>Four Inputs Multiplexed to Two Comparators</td>
<td>010</td>
<td><img src="image" alt="Four Inputs Multiplexed to Two Comparators Diagram" /></td>
</tr>
<tr>
<td>C1IN- A, C1IN+ A, C2IN- A, C2IN+ A, CM&lt;2:0&gt; = 010</td>
<td></td>
<td><img src="image" alt="Comparator Diagram" /></td>
</tr>
<tr>
<td>Two Common Reference Comparators with Outputs</td>
<td>110</td>
<td><img src="image" alt="Two Common Reference Comparators with Outputs Diagram" /></td>
</tr>
<tr>
<td>C1IN- A, C1IN+ A, C2IN- A, C2IN+ A, CM&lt;2:0&gt; = 110</td>
<td></td>
<td><img src="image" alt="Comparator Diagram" /></td>
</tr>
<tr>
<td>Comparators Off ( Lowest Power)</td>
<td>111</td>
<td><img src="image" alt="Comparators Off Diagram" /></td>
</tr>
<tr>
<td>C1IN- A, C1IN+ A, C2IN- A, C2IN+ A, CM&lt;2:0&gt; = 111</td>
<td></td>
<td><img src="image" alt="Comparator Diagram" /></td>
</tr>
</tbody>
</table>

**Legend:**
- A = Analog Input, ports always reads ‘0’
- I/O = Normal port I/O
- CIS = Comparator Input Switch (CMCON0<3>)
- D = Comparator Digital Output

**Note 1:** Reads as ‘0’, unless CxINV = 1.
8.3 Comparator Control

The CMCON0 register (Register 8-1) provides access to the following comparator features:

- Mode selection
- Output state
- Output polarity
- Input switch

8.3.1 COMPARATOR OUTPUT STATE

Each comparator state can always be read internally via the associated CxOUT bit of the CMCON0 register. The comparator outputs are directed to the CxOUT pins when CM<2:0> = 110. When this mode is selected, the TRIS bits for the associated CxOUT pins must be cleared to enable the output drivers.

8.3.2 COMPARATOR OUTPUT POLARITY

Inverting the output of a comparator is functionally equivalent to swapping the comparator inputs. The polarity of a comparator output can be inverted by setting the CxINV bits of the CMCON0 register. Clearing CxINV results in a non-inverted output. A complete table showing the output state versus input conditions and the polarity bit is shown in Table 8-1.

<table>
<thead>
<tr>
<th>Input Conditions</th>
<th>CxINV</th>
<th>CxOUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIN- &gt; VIN+</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>VIN- &lt; VIN+</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>VIN- &gt; VIN+</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>VIN- &lt; VIN+</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Note: CxOUT refers to both the register bit and output pin.

8.3.3 COMPARATOR INPUT SWITCH

The inverting input of the comparators may be switched between two analog pins in the following modes:

- CM<2:0> = 001 (Comparator C1 only)
- CM<2:0> = 010 (Comparators C1 and C2)

In the above modes, both pins remain in Analog mode regardless of which pin is selected as the input. The C1S bit of the CMCON0 register controls the comparator input switch.

8.4 Comparator Response Time

The comparator output is indeterminate for a period of time after the change of an input source or the selection of a new reference voltage. This period is referred to as the response time. The response time of the comparator differs from the settling time of the voltage reference. Therefore, both of these times must be considered when determining the total response time to a comparator input change. See Comparator and Voltage Reference specifications of Section 15.0 “Electrical Specifications” for more details.

8.5 Comparator Interrupt Operation

The comparator interrupt flag is set whenever there is a change in the output value of the comparator. Changes are recognized by means of a mismatch circuit which consists of two latches and an exclusive-or gate (see Figure 8-2 and Figure 8-3). One latch is updated with the comparator output level when the CMCON0 register is read. This latch retains the value until the next read of the CMCON0 register or the occurrence of a reset. The other latch of the mismatch circuit is updated on every Q1 system clock. A mismatch condition will occur when a comparator output change is clocked through the second latch on the Q1 clock cycle. The mismatch condition will persist, holding the CxIF bit of the PIR1 register true, until either the CMCON0 register is read or the comparator output returns to the previous state.

Software will need to maintain information about the status of the comparator output to determine the actual change that has occurred.

The CxIF bit of the PIR1 register 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 CxIE bit of the PIE1 register and the PEIE and GIE bits of the INTCON register must all be set to enable comparator interrupts. If any of these bits are cleared, the interrupt is not enabled, although the CxIF bit of the PIR1 register 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 CMCON0. This will end the mismatch condition.

b) Clear the CxIF interrupt flag.
A persistent mismatch condition will preclude clearing the CxIF interrupt flag. Reading CMCON0 will end the mismatch condition and allow the CxIF bit to be cleared.

**FIGURE 8-6:** COMPARATOR INTERRUPT TIMING W/O CMCON0 READ

**FIGURE 8-7:** COMPARATOR INTERRUPT TIMING WITH CMCON0 READ

8.6 Operation During Sleep

The comparator, if enabled before entering Sleep mode, remains active during Sleep. The additional current consumed by the comparator is shown separately in Section 15.0 “Electrical Specifications”. If the comparator is not used to wake the device, power consumption can be minimized while in Sleep mode by turning off the comparator. The comparator is turned off by selecting mode CM<2:0> = 000 or CM<2:0> = 111 of the CMCON0 register.

A change to the comparator output can wake-up the device from Sleep. To enable the comparator to wake the device from Sleep, the CxIE bit of the PIE1 register and the PEIE bit of the INTCON register must be set. The instruction following the Sleep instruction always executes following a wake from Sleep. If the GIE bit of the INTCON register is also set, the device will then execute the interrupt service routine.

8.7 Effects of a Reset

A device Reset forces the CMCON0 and CMCON1 registers to their Reset states. This forces the Comparator module to be in the Comparator Reset mode (CM<2:0> = 000). Thus, all comparator inputs are analog inputs with the comparator disabled to consume the smallest current possible.

**Note 1:** If a change in the CM1CON0 register (CxOUT) should occur when a read operation is being executed (start of the Q2 cycle), then the CxIF Interrupt Flag bit of the PIR1 register may not get set.

**2:** When either comparator is first enabled, bias circuitry in the Comparator module may cause an invalid output from the comparator until the bias circuitry is stable. Allow about 1 μs for bias settling then clear the mismatch condition and interrupt flags before enabling comparator interrupts.
REGISTER 8-1: CMCON0: COMPARATOR CONFIGURATION REGISTER

<table>
<thead>
<tr>
<th>R-0</th>
<th>R-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>bit 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>C2OUT</td>
<td>C1OUT</td>
<td>C2INV</td>
<td>C1INV</td>
<td>CIS</td>
<td>CM2</td>
<td>CM1</td>
<td>CM0</td>
<td></td>
</tr>
</tbody>
</table>

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

bit 7  C2OUT: Comparator 2 Output bit
When C2INV = 0:
1 = C2 VIN+ > C2 VIN-
0 = C2 VIN+ < C2 VIN-
When C2INV = 1:
1 = C2 VIN+ < C2 VIN-
0 = C2 VIN+ > C2 VIN-

bit 6  C1OUT: Comparator 1 Output bit
When C1INV = 0:
1 = C1 VIN+ > C1 VIN-
0 = C1 VIN+ < C1 VIN-
When C1INV = 1:
1 = C1 VIN+ < C1 VIN-
0 = C1 VIN+ > C1 VIN-

bit 5  C2INV: Comparator 2 Output Inversion bit
1 = C2 output inverted
0 = C2 output not inverted

bit 4  C1INV: Comparator 1 Output Inversion bit
1 = C1 Output inverted
0 = C1 Output not inverted

bit 3  CIS: Comparator Input Switch bit
When CM<2:0> = 010:
1 = C1IN+ connects to C1 VIN-
C2IN+ connects to C2 VIN-
0 = C1IN- connects to C1 VIN-
C2IN- connects to C2 VIN-
When CM<2:0> = 001:
1 = C1IN+ connects to C1 VIN-
0 = C1IN- connects to C1 VIN-

bit 2-0  CM<2:0>: Comparator Mode bits (See Figure 8-5)
000 = Comparators off. CxIN pins are configured as analog
001 = Three inputs multiplexed to two comparators
010 = Four inputs multiplexed to two comparators
011 = Two common reference comparators
100 = Two independent comparators
101 = One independent comparator
110 = Two common reference comparators with outputs
111 = Comparators off. CxIN pins are configured as digital I/O
8.8 Comparator C2 Gating Timer1

This feature can be used to time the duration or interval of analog events. Clearing the T1GSS bit of the CMCON1 register will enable Timer1 to increment based on the output of Comparator C2. This requires that Timer1 is on and gating is enabled. See Section 6.0 “Timer1 Module with Gate Control” for details.

It is recommended to synchronize Comparator C2 with Timer1 by setting the C2SYNC bit when the comparator is used as the Timer1 gate source. This ensures Timer1 does not miss an increment if the comparator changes during an increment.

8.9 Synchronizing Comparator C2 Output to Timer1

The output of Comparator C2 can be synchronized with Timer1 by setting the C2SYNC bit of the CMCON1 register. When enabled, the comparator output is latched on the falling edge of the Timer1 clock source. If a prescaler is used with Timer1, the comparator output is latched after the prescaling function. To prevent a race condition, the comparator output is latched on the falling edge of the Timer1 clock source and Timer1 increments on the rising edge of its clock source. Reference the comparator block diagrams (Figure 8-2 and Figure 8-3) and the Timer1 Block Diagram (Figure 6-1) for more information.

REGISTER 8-2: CMCON1: COMPARATOR CONFIGURATION REGISTER

<table>
<thead>
<tr>
<th></th>
<th>U-0</th>
<th>U-0</th>
<th>U-0</th>
<th>U-0</th>
<th>U-0</th>
<th>U-0</th>
<th>R/W-1</th>
<th>R/W-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>bit 7</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>T1GSS</td>
<td>—</td>
</tr>
<tr>
<td>bit 0</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>C2SYNC</td>
<td>—</td>
</tr>
</tbody>
</table>

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

bit 7-2 Unimplemented: Read as ‘0’

bit 1 T1GSS: Timer1 Gate Source Select bit(1)

1 = Timer1 gate source is T1G pin (pin should be configured as digital input)
0 = Timer1 gate source is Comparator C2 output

bit 0 C2SYNC: Comparator C2 Output Synchronization bit(2)

1 = Output is synchronized with falling edge of Timer1 clock
0 = Output is asynchronous

Note 1: Refer to Section 6.6 “Timer1 Gate”.
2: Refer to Figure 8-3.
8.10 Comparator Voltage Reference

The comparator voltage reference module provides an internally generated voltage reference for the comparators. The following features are available:

- Independent from Comparator operation
- Two 16-level voltage ranges
- Output clamped to VSS
- Ratiometric with VDD

The VRCON register (Register 8) controls the voltage reference module shown in Figure 8-8.

8.10.1 INDEPENDENT OPERATION

The comparator voltage reference is independent of the comparator configuration. Setting the VREN bit of the VRCON register will enable the voltage reference.

8.10.2 OUTPUT VOLTAGE SELECTION

The CVREF voltage reference has 2 ranges with 16 voltage levels in each range. Range selection is controlled by the VRR bit of the VRCON register. The 16 levels are set with the VR<3:0> bits of the VRCON register.

The CVREF output voltage is determined by the following equations:

\[
EQUATION 8-1: \quad CVREF OUTPUT VOLTAGE
\]

\[
\begin{align*}
VRR = 1 \text{ (low range):} & \quad CVREF = (VR<3:0>/24) \times VDD \\
VRR = 0 \text{ (high range):} & \quad CVREF = (VDD/4) + (VR<3:0>/32) \times VDD
\end{align*}
\]

The full range of VSS to VDD cannot be realized due to the construction of the module. See Figure 8-8.

8.10.3 OUTPUT CLAMPED TO VSS

The CVREF output voltage can be set to Vss with no power consumption by configuring VRCON as follows:

- VREN = 0
- VRR = 1
- VR<3:0> = 0000

This allows the comparator to detect a zero-crossing while not consuming additional CVREF module current.

8.10.4 OUTPUT RATIOMETRIC TO VDD

The comparator voltage reference is VDD derived and therefore, the CVREF output changes with fluctuations in VDD. The tested absolute accuracy of the Comparator Voltage Reference can be found in Section 15.0 "Electrical Specifications".

VRCON: VOLTAGE REFERENCE CONTROL REGISTER

<table>
<thead>
<tr>
<th>R/W-0</th>
<th>U-0</th>
<th>R/W-0</th>
<th>U-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>VREN</td>
<td>—</td>
<td>VRR</td>
<td>—</td>
<td>VR3</td>
<td>VR2</td>
<td>VR1</td>
<td>VR0</td>
</tr>
</tbody>
</table>

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

- **bit 7 VREN:** CVREF Enable bit
  - 1 = CVREF circuit powered on
  - 0 = CVREF circuit powered down, no IDD drain and CVREF = VSS.

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

- **bit 5 VRR:** CVREF Range Selection bit
  - 1 = Low range
  - 0 = High range

- **bit 4 Unimplemented:** Read as ‘0’

- **bit 3-0 VR<3:0>:** CVREF Value Selection bits (0 ≤ VR<3:0> ≤ 15)
  - When VRR = 1: CVREF = (VR<3:0>/24) \times VDD
  - When VRR = 0: CVREF = VDD/4 + (VR<3:0>/32) \times VDD

© 2006 Microchip Technology Inc. DS41202D-page 63
TABLE 8-2: SUMMARY OF REGISTERS ASSOCIATED WITH THE COMPARATOR AND VOLTAGE REFERENCE MODULES

<table>
<thead>
<tr>
<th>Name</th>
<th>Bit 7</th>
<th>Bit 6</th>
<th>Bit 5</th>
<th>Bit 4</th>
<th>Bit 3</th>
<th>Bit 2</th>
<th>Bit 1</th>
<th>Bit 0</th>
<th>Value on: POR, BOR</th>
<th>Value on all other Resets</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANSEL</td>
<td>ANS7</td>
<td>ANS6</td>
<td>ANS5</td>
<td>ANS4</td>
<td>ANS3</td>
<td>ANS2</td>
<td>ANS1</td>
<td>ANS0</td>
<td>1111 1111 1111 1111</td>
<td>0000 0000 0000 0000</td>
</tr>
<tr>
<td>CMCON0</td>
<td>C2OUT</td>
<td>C1OUT</td>
<td>C2INV</td>
<td>C1INV</td>
<td>CIS</td>
<td>CM2</td>
<td>CM1</td>
<td>CM0</td>
<td>0000 0000 0000 0000</td>
<td>0000 0000 0000 0000</td>
</tr>
<tr>
<td>CMCON1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>T1GSS -10 -10 -10 -10</td>
<td></td>
</tr>
<tr>
<td>INTCON</td>
<td>GIE</td>
<td>PEIE</td>
<td>T0IE</td>
<td>INTE</td>
<td>RAIE</td>
<td>T0IF</td>
<td>INTF</td>
<td>RAIF</td>
<td>0000 0000 0000 0000</td>
<td></td>
</tr>
<tr>
<td>PIE1</td>
<td>EEIE</td>
<td>ADIE</td>
<td>CCP1IE</td>
<td>C2IE</td>
<td>C1IE</td>
<td>OSFIE</td>
<td>TMR2IE</td>
<td>TMR1IE</td>
<td>0000 0000 0000 0000</td>
<td></td>
</tr>
<tr>
<td>PIR1</td>
<td>EEIF</td>
<td>ADIF</td>
<td>CCP1IF</td>
<td>C2IF</td>
<td>C1IF</td>
<td>OSFIF</td>
<td>TMR2IF</td>
<td>TMR1IF</td>
<td>0000 0000 0000 0000</td>
<td></td>
</tr>
<tr>
<td>PORTA</td>
<td></td>
<td></td>
<td>RA5</td>
<td>RA4</td>
<td>RA3</td>
<td>RA2</td>
<td>RA1</td>
<td>RA0</td>
<td>--x0 x000 x000 x000</td>
<td></td>
</tr>
<tr>
<td>PORTC</td>
<td></td>
<td></td>
<td>RC5</td>
<td>RC4</td>
<td>RC3</td>
<td>RC2</td>
<td>RC1</td>
<td>RC0</td>
<td>--xx 0000 0000 0000</td>
<td></td>
</tr>
<tr>
<td>TRISA</td>
<td></td>
<td></td>
<td>TRISA5</td>
<td>TRISA4</td>
<td>TRISA3</td>
<td>TRISA2</td>
<td>TRISA1</td>
<td>TRISA0</td>
<td>--11 1111 --11 1111</td>
<td></td>
</tr>
<tr>
<td>TRISC</td>
<td></td>
<td></td>
<td>TRISC5</td>
<td>TRISC4</td>
<td>TRISC3</td>
<td>TRISC2</td>
<td>TRISC1</td>
<td>TRISC0</td>
<td>--11 1111 --11 1111</td>
<td></td>
</tr>
<tr>
<td>VRCON</td>
<td>VREN</td>
<td></td>
<td>VRR</td>
<td></td>
<td>VR3</td>
<td>VR2</td>
<td>VR1</td>
<td>VR0</td>
<td>0-0- 0000 0-0- 0000</td>
<td></td>
</tr>
</tbody>
</table>

Legend:  
- x = unknown, u = unchanged, - = unimplemented, read as ’0’. Shaded cells are not used for comparator.

Note 1: Care should be taken to ensure VREF remains within the comparator common mode input range. See Section 15.0 “Electrical Specifications” for more detail.
9.0 ANALOG-TO-DIGITAL CONVERTER (ADC) MODULE

The Analog-to-Digital Converter (ADC) allows conversion of an analog input signal to a 10-bit binary representation of that signal. This device uses analog inputs, which are multiplexed into a single sample and hold circuit. The output of the sample and hold is connected to the input of the converter. The converter generates a 10-bit binary result via successive approximation and stores the conversion result into the ADC result registers (ADRESL and ADRESH).

The ADC voltage reference is software selectable to either VDD or a voltage applied to the external reference pins.

The ADC can generate an interrupt upon completion of a conversion. This interrupt can be used to wake-up the device from Sleep.

Figure 9-1 shows the block diagram of the ADC.

FIGURE 9-1: ADC BLOCK DIAGRAM
9.1 ADC Configuration

When configuring and using the ADC the following functions must be considered:
- Port configuration
- Channel selection
- ADC voltage reference selection
- ADC conversion clock source
- Interrupt control
- Results formatting

9.1.1 PORT CONFIGURATION

The ADC can be used to convert both analog and digital signals. When converting analog signals, the I/O pin should be configured for analog by setting the associated TRIS and ANSEL bits. See the corresponding port section for more information.

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

9.1.2 CHANNEL SELECTION

The CHS bits of the ADCON0 register determine which channel is connected to the sample and hold circuit. When changing channels, a delay is required before starting the next conversion. Refer to Section 9.2 “ADC Operation” for more information.

9.1.3 ADC VOLTAGE REFERENCE

The VCFG bit of the ADCON0 register provides control of the positive voltage reference. The positive voltage reference can be either VDD or an external voltage source. The negative voltage reference is always connected to the ground reference.

9.1.4 CONVERSION CLOCK

The source of the conversion clock is software selectable via the ADCS bits of the ADCON1 register. There are seven possible clock options:
- FOSC/2
- FOSC/4
- FOSC/8
- FOSC/16
- FOSC/32
- FOSC/64
- FRC (dedicated internal oscillator)

The time to complete one bit conversion is defined as TAD. One full 10-bit conversion requires 11 TAD periods as shown in Figure 9-3.

For correct conversion, the appropriate TAD specification must be met. See A/D conversion requirements in Section 15.0 “Electrical Specifications” for more information. Table 9-1 gives examples of appropriate ADC clock selections.

Note: Unless using the FRC, any changes in the system clock frequency will change the ADC clock frequency, which may adversely affect the ADC result.

### TABLE 9-1: ADC CLOCK PERIOD (TAD) VS. DEVICE OPERATING FREQUENCIES (VDD ≥ 3.0V)

<table>
<thead>
<tr>
<th>ADC Clock Source</th>
<th>ADCS&lt;2:0&gt;</th>
<th>20 MHz</th>
<th>8 MHz</th>
<th>4 MHz</th>
<th>1 MHz</th>
</tr>
</thead>
<tbody>
<tr>
<td>FOSC/2</td>
<td>000</td>
<td>100 ns</td>
<td>250 ns</td>
<td>500 ns</td>
<td>2.0 μs</td>
</tr>
<tr>
<td>FOSC/4</td>
<td>100</td>
<td>200 ns</td>
<td>500 ns</td>
<td>1.0 μs</td>
<td>4.0 μs</td>
</tr>
<tr>
<td>FOSC/8</td>
<td>001</td>
<td>400 ns</td>
<td>1.0 μs</td>
<td>2.0 μs</td>
<td>8.0 μs</td>
</tr>
<tr>
<td>FOSC/16</td>
<td>011</td>
<td>800 ns</td>
<td>2.0 μs</td>
<td>4.0 μs</td>
<td>16.0 μs</td>
</tr>
<tr>
<td>FOSC/32</td>
<td>010</td>
<td>1.6 μs</td>
<td>4.0 μs</td>
<td>8.0 μs</td>
<td>32.0 μs</td>
</tr>
<tr>
<td>FOSC/64</td>
<td>110</td>
<td>3.2 μs</td>
<td>8.0 μs</td>
<td>16.0 μs</td>
<td>64.0 μs</td>
</tr>
<tr>
<td>FRC</td>
<td>x11</td>
<td>2-6 μs</td>
<td>2-6 μs</td>
<td>2-6 μs</td>
<td>2-6 μs</td>
</tr>
</tbody>
</table>

Legend: Shaded cells are outside of recommended range.

Note 1: The FRC 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 FRC clock source is only recommended if the conversion will be performed during Sleep.
9.1.5 INTERRUPTS
The ADC module allows for the ability to generate an interrupt upon completion of an analog-to-digital conversion. The ADC interrupt flag is the ADIF bit in the PIR1 register. The ADC interrupt enable is the ADIE bit in the PIE1 register. The ADIF bit must be cleared in software.

This interrupt can be generated while the device is operating or while in Sleep. If the device is in Sleep, the interrupt will wake-up the device. Upon waking from Sleep, the next instruction following the SLEEP instruction is always executed. If the user is attempting to wake-up from Sleep and resume in-line code execution, the global interrupt must be disabled. If the global interrupt is enabled, execution will switch to the interrupt service routine.

Please see Section 9.1.5 “Interrupts” for more information.

9.1.6 RESULT FORMATTING
The 10-bit A/D conversion result can be supplied in two formats, left justified or right justified. The ADFM bit of the ADCON0 register controls the output format.

Figure 9-4 shows the two output formats.
9.2 ADC Operation

9.2.1 STARTING A CONVERSION
To enable the ADC module, the ADON bit of the ADCON0 register must be set to a ‘1’. Setting the GO/DONE bit of the ADCON0 register to a ‘1’ will start the analog-to-digital conversion.

Note: The GO/DONE bit should not be set in the same instruction that turns on the ADC. Refer to Section 9.2.6 “A/D Conversion Procedure”.

9.2.2 COMPLETION OF A CONVERSION
When the conversion is complete, the ADC module will:
- Clear the GO/DONE bit
- Set the ADIF flag bit
- Update the ADRESH:ADRESL registers with new conversion result

9.2.3 TERMINATING A CONVERSION
If a conversion must be terminated before completion, the GO/DONE bit can be cleared in software. The ADRESH:ADRESL registers will not be updated with the partially complete analog-to-digital conversion sample. Instead, the ADRESH:ADRESL register pair will retain the value of the previous conversion. Additionally, a 2 TAD delay is required before another acquisition can be initiated. Following this delay, an input acquisition is automatically started on the selected channel.

Note: A device Reset forces all registers to their Reset state. Thus, the ADC module is turned off and any pending conversion is terminated.

9.2.4 ADC OPERATION DURING SLEEP
The ADC module can operate during Sleep. This requires the ADC clock source to be set to the FRC option. When the FRC clock source is selected, the ADC waits one additional instruction before starting the conversion. This allows the SLEEP instruction to be executed, which can reduce system noise during the conversion. If the ADC interrupt is enabled, the device will wake-up from Sleep when the conversion completes. If the ADC interrupt is disabled, the ADC module is turned off after the conversion completes, although the ADON bit remains set.

When the ADC clock source is something other than FRC, a SLEEP instruction causes the present conversion to be aborted and the ADC module is turned off, although the ADON bit remains set.

9.2.5 SPECIAL EVENT TRIGGER
The ECCP Special Event Trigger allows periodic ADC measurements without software intervention. When this trigger occurs, the GO/DONE bit is set by hardware and the Timer1 counter resets to zero.

Using the Special Event Trigger does not ensure proper ADC timing. It is the user’s responsibility to ensure that the ADC timing requirements are met.

See Section 11.0 “Enhanced Capture/Compare/PWM (With Auto-Shutdown and Dead Band) Module” for more information.

9.2.6 A/D CONVERSION PROCEDURE
This is an example procedure for using the ADC to perform an Analog-to-Digital conversion:

1. Configure Port:
   - Disable pin output driver (See TRIS register)
   - Configure pin as analog
2. Configure the ADC module:
   - Select ADC conversion clock
   - Configure voltage reference
   - Select ADC input channel
   - Select result format
   - Turn on ADC module
3. Configure ADC interrupt (optional):
   - Clear ADC interrupt flag
   - Enable ADC interrupt
   - Enable peripheral interrupt
   - Enable global interrupt(1)
4. Wait the required acquisition time(2).
5. Start conversion by setting the GO/DONE bit.
6. Wait for ADC conversion to complete by one of the following:
   - Polling the GO/DONE bit
   - Waiting for the ADC interrupt (interrupts enabled)
7. Read ADC Result
8. Clear the ADC interrupt flag (required if interrupt is enabled).

Note 1: The global interrupt can be disabled if the user is attempting to wake-up from Sleep and resume in-line code execution.
2: See Section 9.3 “A/D Acquisition Requirements”. 
EXAMPLE 9-1: A/D CONVERSION

;This code block configures the ADC
;for polling, Vdd reference, Frc clock
;and AN0 input.
;
;Conversion start & polling for completion
;are included.
;
BANKSEL ADCON1 ;
MOVLR B’01110000’ ;ADC Frc clock
MOVWF ADCON1 ;
BANKSEL TRISA ;
BSF TRISA,0 ;Set RA0 to input
BANKSEL ANSEL ;
BSF ANSEL,0 ;Set RA0 to analog
BANKSEL ADCON0 ;
MOVLR B’10000001’ ;Right justify,
MOVWF ADCON0 ; Vdd Vref, AN0, On
CALL SampleTime ;Acquisition delay
BSF ADCON0,GO ;Start conversion
BTFSC ADCON0,GO ;Is conversion done?
GOTO §-1 ;No, test again
BANKSEL ADRESH ;
MOVWF ADRESH,W ;Read upper 2 bits
MOVWF RESULTHI ;store in GPR space
BANKSEL ADRESL ;
MOVWF ADRESL,W ;Read lower 8 bits
MOVWF RESULTLO ;Store in GPR space
9.2.7 ADC REGISTER DEFINITIONS

The following registers are used to control the operation of the ADC.

REGISTER 9-1: ADCON0: A/D CONTROL REGISTER 0

<table>
<thead>
<tr>
<th>R/W-0</th>
<th>R/W-0</th>
<th>U-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADFM</td>
<td>VCFG</td>
<td>CHS2</td>
<td>CHS1</td>
<td>CHS0</td>
<td>GO/DONE</td>
<td>ADON</td>
<td></td>
</tr>
</tbody>
</table>

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>
<thead>
<tr>
<th>bit 7</th>
<th>ADFM: A/D Conversion Result Format Select bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Right justified</td>
</tr>
<tr>
<td>0</td>
<td>Left justified</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>bit 6</th>
<th>VCFG: Voltage Reference bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>VREF pin</td>
</tr>
<tr>
<td>0</td>
<td>VDD</td>
</tr>
</tbody>
</table>

| bit 5 | Unimplemented: Read as ‘0’ |

<table>
<thead>
<tr>
<th>bit 4-2</th>
<th>CHS&lt;2:0&gt;: Analog Channel Select bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>AN0</td>
</tr>
<tr>
<td>001</td>
<td>AN1</td>
</tr>
<tr>
<td>010</td>
<td>AN2</td>
</tr>
<tr>
<td>011</td>
<td>AN3</td>
</tr>
<tr>
<td>100</td>
<td>AN4</td>
</tr>
<tr>
<td>101</td>
<td>AN5</td>
</tr>
<tr>
<td>110</td>
<td>AN6</td>
</tr>
<tr>
<td>111</td>
<td>AN7</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>bit 1</th>
<th>GO/DONE: A/D Conversion Status bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>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.</td>
</tr>
<tr>
<td>0</td>
<td>A/D conversion completed/not in progress</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>bit 0</th>
<th>ADON: ADC Enable bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ADC is enabled</td>
</tr>
<tr>
<td>0</td>
<td>ADC is disabled and consumes no operating current</td>
</tr>
</tbody>
</table>

REGISTER 9-2: ADCON1: A/D CONTROL REGISTER 1

<table>
<thead>
<tr>
<th>U-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>U-0</th>
<th>U-0</th>
<th>U-0</th>
<th>U-0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ADCS2</td>
<td>ADCS1</td>
<td>ADCS0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

| bit 7 | Unimplemented: Read as ‘0’ |

<table>
<thead>
<tr>
<th>bit 6-4</th>
<th>ADCS&lt;2:0&gt;: A/D Conversion Clock Select bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>Fosc/2</td>
</tr>
<tr>
<td>001</td>
<td>Fosc/8</td>
</tr>
<tr>
<td>010</td>
<td>Fosc/32</td>
</tr>
<tr>
<td>x11</td>
<td>Frc (clock derived from a dedicated internal oscillator = 500 kHz max)</td>
</tr>
<tr>
<td>100</td>
<td>Fosc/4</td>
</tr>
<tr>
<td>101</td>
<td>Fosc/16</td>
</tr>
<tr>
<td>110</td>
<td>Fosc/64</td>
</tr>
</tbody>
</table>

| bit 3-0 | Unimplemented: Read as ‘0’ |
**REGISTER 9-3: ADRESH: ADC RESULT REGISTER HIGH (ADRESH) ADFM = 0**

<table>
<thead>
<tr>
<th>R/W-x</th>
<th>R/W-x</th>
<th>R/W-x</th>
<th>R/W-x</th>
<th>R/W-x</th>
<th>R/W-x</th>
<th>R/W-x</th>
<th>R/W-x</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADRES9</td>
<td>ADRES8</td>
<td>ADRES7</td>
<td>ADRES6</td>
<td>ADRES5</td>
<td>ADRES4</td>
<td>ADRES3</td>
<td>ADRES2</td>
</tr>
</tbody>
</table>

bit 7

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

bit 7-0

ADRES<9:2>: ADC Result Register bits
Upper 8 bits of 10-bit conversion result

**REGISTER 9-4: ADRESL: ADC RESULT REGISTER LOW (ADRESL) ADFM = 0**

<table>
<thead>
<tr>
<th>R/W-x</th>
<th>R/W-x</th>
<th>R/W-x</th>
<th>R/W-x</th>
<th>R/W-x</th>
<th>R/W-x</th>
<th>R/W-x</th>
<th>R/W-x</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADRES1</td>
<td>ADRES0</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

bit 7

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

bit 7-6

ADRES<1:0>: ADC Result Register bits
Lower 2 bits of 10-bit conversion result

bit 5-0

Reserved: Do not use.

**REGISTER 9-5: ADRESH: ADC RESULT REGISTER HIGH (ADRESH) ADFM = 1**

<table>
<thead>
<tr>
<th>R/W-x</th>
<th>R/W-x</th>
<th>R/W-x</th>
<th>R/W-x</th>
<th>R/W-x</th>
<th>R/W-x</th>
<th>R/W-x</th>
<th>R/W-x</th>
</tr>
</thead>
<tbody>
<tr>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>ADRES9</td>
</tr>
<tr>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>ADRES8</td>
</tr>
</tbody>
</table>

bit 7

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

bit 7-2

Reserved: Do not use.

bit 1-0

ADRES<9:8>: ADC Result Register bits
Upper 2 bits of 10-bit conversion result

**REGISTER 9-6: ADRESL: ADC RESULT REGISTER LOW (ADRESL) ADFM = 1**

<table>
<thead>
<tr>
<th>R/W-x</th>
<th>R/W-x</th>
<th>R/W-x</th>
<th>R/W-x</th>
<th>R/W-x</th>
<th>R/W-x</th>
<th>R/W-x</th>
<th>R/W-x</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADRES7</td>
<td>ADRES6</td>
<td>ADRES5</td>
<td>ADRES4</td>
<td>ADRES3</td>
<td>ADRES2</td>
<td>ADRES1</td>
<td>ADRES0</td>
</tr>
</tbody>
</table>

bit 7

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

bit 7-0

ADRES<7:0>: ADC Result Register bits
Lower 8 bits of 10-bit conversion result
9.3 A/D Acquisition Requirements

For the ADC 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 9-4. 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 9-4. The maximum recommended impedance for analog sources is 10 kΩ. As the source impedance is decreased, the acquisition time may be decreased. After the analog input channel is selected (or changed), an A/D acquisition must be done before the conversion can be started. To calculate the minimum acquisition time, Equation 9-1 may be used. This equation assumes that 1/2 LSb error is used (1024 steps for the ADC). The 1/2 LSb error is the maximum error allowed for the ADC to meet its specified resolution.

**EQUATION 9-1: ACQUISITION TIME EXAMPLE**

Assumptions: Temperature = 50°C and external impedance of 10kΩ 5.0V VDD

\[ T_{ACQ} = \text{Amplifier Settling Time} + \text{Hold Capacitor Charging Time} + \text{Temperature Coefficient} \]

\[ = T_{AMP} + T_{C} + T_{COFF} \]

\[ = 5\mu s + T_{C} + [(\text{Temperature} - 25^\circ\text{C})(0.05\mu\text{s/}^\circ\text{C})] \]

The value for TC can be approximated with the following equations:

\[ V_{APPLIED}\left(1 - \frac{1}{2047}\right) = V_{CHOLD} \quad ;[1] \text{VCHOLD charged to within 1/2 lsb} \]

\[ V_{APPLIED}\left(1 - e^{-\frac{T_{C}}{RC}}\right) = V_{CHOLD} \quad ;[2] \text{VCHOLD charge response to VAPPLIED} \]

\[ V_{APPLIED}\left(1 - e^{-\frac{T_{C}}{RC}}\right) = V_{APPLIED}\left(1 - \frac{1}{2047}\right) \quad ; \text{combining [1] and [2]} \]

Solving for TC:

\[ T_{C} = -CHOLD(R_{IC} + RSS + RS) \ln(1/2047) \]

\[ = -10\mu F(1k\Omega + 7k\Omega + 10k\Omega) \ln(0.0004885) \]

\[ = 1.37\mu s \]

Therefore:

\[ T_{ACQ} = 5\mu s + 1.37\mu s + [(50^\circ\text{C} - 25^\circ\text{C})(0.05\mu\text{s/}^\circ\text{C})] \]

\[ = 7.67\mu s \]

**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 9-4: 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

FIGURE 9-5: ADC TRANSFER FUNCTION
### TABLE 9-2: SUMMARY OF ASSOCIATED ADC REGISTERS

<table>
<thead>
<tr>
<th>Name</th>
<th>Bit 7</th>
<th>Bit 6</th>
<th>Bit 5</th>
<th>Bit 4</th>
<th>Bit 3</th>
<th>Bit 2</th>
<th>Bit 1</th>
<th>Bit 0</th>
<th>Value on: POR, BOR</th>
<th>Value on all other Resets</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADCON0</td>
<td>ADFM</td>
<td>VCFG</td>
<td>—</td>
<td>CHS2</td>
<td>CHS1</td>
<td>CHS0</td>
<td>GO/DONE</td>
<td>ADON</td>
<td>0000 0000</td>
<td>0000 0000</td>
</tr>
<tr>
<td>ADCON1</td>
<td>—</td>
<td>ADCS2</td>
<td>ADCS1</td>
<td>ADCS0</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>-000 ----</td>
<td>-000 ----</td>
</tr>
<tr>
<td>ANSEL</td>
<td>ANS7</td>
<td>ANS6</td>
<td>ANS5</td>
<td>ANS4</td>
<td>ANS3</td>
<td>ANS2</td>
<td>ANS1</td>
<td>ANS0</td>
<td>1111 1111</td>
<td>1111 1111</td>
</tr>
<tr>
<td>ADRESH</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>ADRESL</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>INTCON</td>
<td>GIE</td>
<td>PEIE</td>
<td>T0IE</td>
<td>INTE</td>
<td>RAIE</td>
<td>T0IF</td>
<td>INTF</td>
<td>RAIF</td>
<td>0000 0000</td>
<td>0000 0000</td>
</tr>
<tr>
<td>PIE1</td>
<td>EEIE</td>
<td>ADIE</td>
<td>CCPIE</td>
<td>C2IE</td>
<td>C1IE</td>
<td>OSFIE</td>
<td>TMR2IE</td>
<td>TMR1IE</td>
<td>0000 0000</td>
<td>0000 0000</td>
</tr>
<tr>
<td>PIR1</td>
<td>EEIF</td>
<td>ADIF</td>
<td>CCPIF</td>
<td>C2IF</td>
<td>C1IF</td>
<td>OSFIF</td>
<td>TMR2IF</td>
<td>TMR1IF</td>
<td>0000 0000</td>
<td>0000 0000</td>
</tr>
<tr>
<td>PORTA</td>
<td>—</td>
<td>—</td>
<td>RA5</td>
<td>RA4</td>
<td>RA3</td>
<td>RA2</td>
<td>RA1</td>
<td>RA0</td>
<td>--x0 x000</td>
<td>--uu uuuu</td>
</tr>
<tr>
<td>PORTC</td>
<td>—</td>
<td>—</td>
<td>RC5</td>
<td>RC4</td>
<td>RC3</td>
<td>RC2</td>
<td>RC1</td>
<td>RC0</td>
<td>--xx 0000</td>
<td>--uu uuuu</td>
</tr>
<tr>
<td>TRISA</td>
<td>—</td>
<td>—</td>
<td>TRISA5</td>
<td>TRISA4</td>
<td>TRISA3</td>
<td>TRISA2</td>
<td>TRISA1</td>
<td>TRISA0</td>
<td>--11 1111</td>
<td>--11 1111</td>
</tr>
<tr>
<td>TRISC</td>
<td>—</td>
<td>—</td>
<td>TRISC5</td>
<td>TRISC4</td>
<td>TRISC3</td>
<td>TRISC2</td>
<td>TRISC1</td>
<td>TRISC0</td>
<td>--11 1111</td>
<td>--11 1111</td>
</tr>
</tbody>
</table>

**Legend:**  
- x = unknown,  
- u = unchanged,  
- — = unimplemented read as ‘0’.  
Shaded cells are not used for ADC module.
10.0 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)
- EEDAT
- EEADR

EEDAT holds the 8-bit data for read/write, and EEADR holds the address of the EEPROM location being accessed. PIC16F684 has 256 bytes of data EEPROM with an address range from 0h to FFh.

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 in Section 15.0 “Electrical 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 the data EEPROM data and will read zeroes.

REGISTER 10-1: EEDAT: EEPROM DATA REGISTER

<table>
<thead>
<tr>
<th>R/W</th>
<th>R/W</th>
<th>R/W</th>
<th>R/W</th>
<th>R/W</th>
<th>R/W</th>
<th>R/W</th>
<th>R/W</th>
</tr>
</thead>
<tbody>
<tr>
<td>EEDAT7</td>
<td>EEDAT6</td>
<td>EEDAT5</td>
<td>EEDAT4</td>
<td>EEDAT3</td>
<td>EEDAT2</td>
<td>EEDAT1</td>
<td>EEDAT0</td>
</tr>
</tbody>
</table>

bit 7-0  

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

bit 7-0  

EEDATn: Byte Value to Write To or Read From Data EEPROM bits

REGISTER 10-2: EEADR: EEPROM ADDRESS REGISTER

<table>
<thead>
<tr>
<th>R/W</th>
<th>R/W</th>
<th>R/W</th>
<th>R/W</th>
<th>R/W</th>
<th>R/W</th>
<th>R/W</th>
<th>R/W</th>
</tr>
</thead>
<tbody>
<tr>
<td>EEADR7</td>
<td>EEADR6</td>
<td>EEADR5</td>
<td>EEADR4</td>
<td>EEADR3</td>
<td>EEADR2</td>
<td>EEADR1</td>
<td>EEADR0</td>
</tr>
</tbody>
</table>

bit 7-0  

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

bit 7-0  

EEADR: Specifies One of 256 Locations for EEPROM Read/Write Operation bits
10.1 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 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 EEDAT and EEADR registers will need to be re-initialized.

Interrupt flag, EEIF bit of the PIR1 register, is set when 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 10-3: EECON1: EEPROM CONTROL REGISTER

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th>R/W-x</th>
<th>R/W-0</th>
<th>R/S-0</th>
<th>R/S-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>U-0</td>
<td>U-0</td>
<td>U-0</td>
<td>U-0</td>
<td>WRERR</td>
<td>WREN</td>
<td>WR</td>
<td>RD</td>
</tr>
</tbody>
</table>

bit 7-4  Unimplemented: Read as '0'
bit 3    WRERR: EEPROM Error Flag bit
          1 = A write operation is prematurely terminated (any MCLR Reset, any WDT Reset during normal operation or BOR Reset)
          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

Note: The EECON1, EEDAT and EEADR registers should not be modified during a data EEPROM write (WR bit = 1).
10.2 Reading the EEPROM Data Memory

To read a data memory location, the user must write the address to the EEAADR register and then set control bit RD of the EECON1 register, as shown in Example 10-1. The data is available, at the very next cycle, in the EEDAT register. Therefore, it can be read in the next instruction. EEDAT holds this value until another read, or until it is written to by the user (during a write operation).

**EXAMPLE 10-1: DATA EEPROM READ**

<table>
<thead>
<tr>
<th>BANKSEL</th>
<th>EEADR</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOV L W</td>
<td>CONFIG_ADDR</td>
</tr>
<tr>
<td>MOV W F</td>
<td>EEAADR ;Address to read</td>
</tr>
<tr>
<td>BSF</td>
<td>EECON1,RD ;BE Read</td>
</tr>
<tr>
<td>MOV F V</td>
<td>EEDAT,W ;Move data to W</td>
</tr>
</tbody>
</table>

10.3 Writing to the EEPROM Data Memory

To write an EEPROM data location, the user must first write the address to the EEAADR register and the data to the EEDAT register. Then the user must follow a specific sequence to initiate the write for each byte, as shown in Example 10-2.

**EXAMPLE 10-2: DATA EEPROM WRITE**

<table>
<thead>
<tr>
<th>BANKSEL</th>
<th>EECON1</th>
</tr>
</thead>
<tbody>
<tr>
<td>BSF</td>
<td>EECON1,WREN ;Enable write</td>
</tr>
<tr>
<td>BCF</td>
<td>INTCON,GIE ;Disable INTs</td>
</tr>
<tr>
<td>BTFSC</td>
<td>INTCON,GIE ;See AN576</td>
</tr>
<tr>
<td>GOTO</td>
<td>$-2</td>
</tr>
<tr>
<td>MOVL W</td>
<td>55h ;Unlock write</td>
</tr>
<tr>
<td>MOV W F</td>
<td>EECON2</td>
</tr>
<tr>
<td>MOV W L</td>
<td>AAh</td>
</tr>
<tr>
<td>MOV W F</td>
<td>EECON2</td>
</tr>
<tr>
<td>BSF</td>
<td>EECON1,WR ;Start the write</td>
</tr>
<tr>
<td>BSF INT</td>
<td>INTCON,GIE ;Enable INTs</td>
</tr>
</tbody>
</table>

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 of the PIR1 register must be cleared by software.

10.4 Write Verify

Depending on the application, good programming practice may dictate that the value written to the data EEPROM should be verified (see Example 10-3) to the desired value to be written.

**EXAMPLE 10-3: WRITE VERIFY**

<table>
<thead>
<tr>
<th>BANKSEL EEDAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOV F EEDAT,W ;EEDAT not changed from previous write</td>
</tr>
<tr>
<td>BSF EECON1,RD ;YES, Read the value written</td>
</tr>
<tr>
<td>XOR W F EEDAT,W</td>
</tr>
<tr>
<td>BTF S S STATUS,Z ;Is data the same</td>
</tr>
<tr>
<td>GOTO WRITE_ERR ;No, handle error</td>
</tr>
</tbody>
</table>

10.4.1 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). When variables in one section change frequently, while variables in another section do not change, it is possible to exceed the total number of write cycles to the EEPROM (specification D124) without exceeding the total number of write cycles to a single byte (specifications D120 and D120A). If this is the case, then a refresh of the array must be performed. For this reason, variables that change infrequently (such as constants, IDs, calibration, etc.) should be stored in Flash program memory.
10.5 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 (64 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

10.6 Data EEPROM Operation During Code-Protect

Data memory can be code-protected by programming the CPD bit in the Configuration Word register (Register 12-1) 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 NOPs) to reach an added routine, programmed in unused program memory, which outputs the contents of data memory. Programming unused locations in program memory to ‘0’ will also help prevent data memory code protection from becoming breached.

### TABLE 10-1: SUMMARY OF ASSOCIATED DATA EEPROM REGISTERS

<table>
<thead>
<tr>
<th>Name</th>
<th>Bit 7</th>
<th>Bit 6</th>
<th>Bit 5</th>
<th>Bit 4</th>
<th>Bit 3</th>
<th>Bit 2</th>
<th>Bit 1</th>
<th>Bit 0</th>
<th>Value on: POR, BOR</th>
<th>Value on all other Resets</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTCON</td>
<td>GIE</td>
<td>PEIE</td>
<td>T0IE</td>
<td>INTE</td>
<td>RAIE</td>
<td>T0IF</td>
<td>INTF</td>
<td>RAIF</td>
<td>0000 0000</td>
<td>0000 0000</td>
</tr>
<tr>
<td>PIR1</td>
<td>EEIF</td>
<td>ADIF</td>
<td>CCP1IF</td>
<td>C2IF</td>
<td>C1IF</td>
<td>OSFIF</td>
<td>TMR2IF</td>
<td>TMR1IF</td>
<td>0000 0000</td>
<td>0000 0000</td>
</tr>
<tr>
<td>PIE1</td>
<td>EEIE</td>
<td>ADIE</td>
<td>CCP1IE</td>
<td>C2IE</td>
<td>C1IE</td>
<td>OSFIE</td>
<td>TMR2IE</td>
<td>TMR1IE</td>
<td>0000 0000</td>
<td>0000 0000</td>
</tr>
<tr>
<td>EEDAT</td>
<td>EEDAT7</td>
<td>EEDAT6</td>
<td>EEDAT5</td>
<td>EEDAT4</td>
<td>EEDAT3</td>
<td>EEDAT2</td>
<td>EEDAT1</td>
<td>EEDAT0</td>
<td>0000 0000</td>
<td>0000 0000</td>
</tr>
<tr>
<td>EEADR</td>
<td>EEADR7</td>
<td>EEADR6</td>
<td>EEADR5</td>
<td>EEADR4</td>
<td>EEADR3</td>
<td>EEADR2</td>
<td>EEADR1</td>
<td>EEADR0</td>
<td>0000 0000</td>
<td>0000 0000</td>
</tr>
<tr>
<td>EECON1</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>WRERR</td>
<td>WREN</td>
<td>WR</td>
<td>—----- x000</td>
<td>—----- q000</td>
</tr>
<tr>
<td>EECON2(f)</td>
<td>EEPROM Control Register 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**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.
11.0 ENHANCED CAPTURE/COMPARE/PWM (WITH AUTO-SHUTDOWN AND DEAD BAND) MODULE

The Enhanced Capture/Compare/PWM module is a peripheral which allows the user to time and control different events. In Capture mode, the peripheral allows the timing of the duration of an event. The Compare mode allows the user to trigger an external event when a predetermined amount of time has expired. The PWM mode can generate a Pulse-Width Modulated signal of varying frequency and duty cycle.

Table 11-1 shows the timer resources required by the ECCP module.

<table>
<thead>
<tr>
<th>ECCP Mode</th>
<th>Timer Resource</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capture</td>
<td>Timer1</td>
</tr>
<tr>
<td>Compare</td>
<td>Timer1</td>
</tr>
<tr>
<td>PWM</td>
<td>Timer2</td>
</tr>
</tbody>
</table>

REGISTER 11-1: CCP1CON: ENHANCED CCP1 CONTROL REGISTER

<table>
<thead>
<tr>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1M1</td>
<td>P1M0</td>
<td>DC1B1</td>
<td>DC1B0</td>
<td>CCP1M3</td>
<td>CCP1M2</td>
<td>CCP1M1</td>
<td>CCP1M0</td>
</tr>
</tbody>
</table>

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

bit 7-6  P1M<1:0>: PWM Output Configuration bits
If CCP1M<3:2> = 00, 01, 10:
xx = P1A assigned as Capture/Compare input; P1B, P1C, P1D assigned as port pins
If CCP1M<3:2> = 11:
00 = Single output; P1A modulated; P1B, P1C, P1D assigned as port pins
01 = Full-bridge output forward; P1D modulated; P1A active; P1B, P1C inactive
10 = Half-bridge output; P1A, P1B modulated with dead-band control; P1C, P1D assigned as port pins
11 = Full-bridge output reverse; P1B modulated; P1C active; P1A, P1D inactive

bit 5-4  DC1B<1:0>: PWM Duty Cycle Least Significant bits
Capture mode:
Unused.
Compare mode:
Unused.
PWM mode:
These bits are the two LSbs of the PWM duty cycle. The eight MSbs are found in CCPR1L.

bit 3-0  CCP1M<3:0>: ECCP Mode Select bits
0000 = Capture/Compare/PWM off (resets ECCP module)
0001 = Unused (reserved)
0010 = Compare mode, toggle output on match (CCP1IF bit is set)
0011 = Unused (reserved)
0100 = Capture mode, every falling edge
0101 = Capture mode, every rising edge
0110 = Capture mode, every 4th rising edge
0111 = Capture mode, every 16th rising edge
1000 = Compare mode, set output on match (CCP1IF bit is set)
1001 = Compare mode, clear output on match (CCP1IF bit is set)
1010 = Compare mode, generate software interrupt on match (CCP1IF bit is set, CCP1 pin is unaffected)
1011 = Compare mode, trigger special event (CCP1IF bit is set; CCP1 resets TMR1 or TMR2, and starts an A/D conversion, if the ADC module is enabled)
1100 = PWM mode; P1A, P1C active-high; P1B, P1D active-high
1101 = PWM mode; P1A, P1C active-high; P1B, P1D active-low
1110 = PWM mode; P1A, P1C active-low; P1B, P1D active-high
1111 = PWM mode; P1A, P1C active-low; P1B, P1D active-low
11.1 Capture Mode

In Capture mode, CCPR1H:CCPR1L captures the 16-bit value of the TMR1 register when an event occurs on pin CCP1. An event is defined as one of the following and is configured by the CCP1M<3:0> bits of the CCP1CON register:

- Every falling edge
- Every rising edge
- Every 4th rising edge
- Every 16th rising edge

When a capture is made, the Interrupt Request Flag bit CCP1IF of the PIR1 register is set. The interrupt flag must be cleared in software. If another capture occurs before the value in the CCPR1H, CCPR1L register pair is read, the old captured value is overwritten by the new captured value (see Figure 11-1).

11.1.1 CCP1 PIN CONFIGURATION

In Capture mode, the CCP1 pin should be configured as an input by setting the associated TRIS control bit.

---

**FIGURE 11-1: CAPTURE MODE OPERATION BLOCK DIAGRAM**

- Prescaler: 1, 4, 16
- Edge Detect: and
- Set Flag bit CCP1IF (PIR1 register)
- Capture Enable
- CCP1CON<3:0>
- System Clock (Fosc)
- CCPR1H CCPR1L
- CCP1 pin

---

11.1.2 TIMER1 MODE SELECTION

Timer1 must be running in Timer mode or Synchronized Counter mode for the CCP module to use the capture feature. In Asynchronous Counter mode, the capture operation may not work.

11.1.3 SOFTWARE INTERRUPT

When the Capture mode is changed, a false capture interrupt may be generated. The user should keep the CCP1IE interrupt enable bit of the PIE1 register clear to avoid false interrupts. Additionally, the user should clear the CCP1IF interrupt flag bit of the PIR1 register following any change in operating mode.

11.1.4 CCP PRESCALER

There are four prescaler settings specified by the CCP1M<3:0> bits of the CCP1CON register. Whenever the CCP module is turned off, or the CCP module is not in Capture mode, the prescaler counter is cleared. Any Reset will clear the prescaler counter.

Switching from one capture prescaler to another does not clear the prescaler and may generate a false interrupt. To avoid this unexpected operation, turn the module off by clearing the CCP1CON register before changing the prescaler (see Example 11-1).

---

**EXAMPLE 11-1: CHANGING BETWEEN CAPTURE PRESCALERS**

```assembly
BANKSEL CCP1CON ;Set Bank bits to point ;to CCP1CON
CLR CCP1CON ;Turn CCP module off
MOVLW NEW_CAPT_PS ;Load the W reg with ;the new prescaler
;move value and CCP ON
MOVWF CCP1CON ;Load CCP1CON with this ;value
```
11.2 Compare Mode

In Compare mode, the 16-bit CCPR1 register value is constantly compared against the TMR1 register pair value. When a match occurs, the CCP module may:

- Toggle the CCP1 output
- Set the CCP1 output
- Clear the CCP1 output
- Generate a Special Event Trigger
- Generate a Software Interrupt

The action on the pin is based on the value of the CCP1M<3:0> control bits of the CCP1CON register.

All Compare modes can generate an interrupt.

**FIGURE 11-2: COMPARE MODE OPERATION BLOCK DIAGRAM**

![Diagram showing CCP1 pin configuration and operation blocks]

11.2.1 CCP1 PIN CONFIGURATION

The user must configure the CCP1 pin as an output by clearing the associated TRIS bit.

**Note:** Clearing the CCP1CON register will force the CCP1 compare output latch to the default low level. This is not the port I/O data latch.

11.2.2 TIMER1 MODE SELECTION

In Compare mode, Timer1 must be running in either Timer mode or Synchronized Counter mode. The compare operation may not work in Asynchronous Counter mode.

11.2.3 SOFTWARE INTERRUPT MODE

When Generate Software Interrupt mode is chosen (CCP1M<3:0> = 1010), the CCP module does not assert control of the CCP1 pin (see the CCP1CON register).

11.2.4 SPECIAL EVENT TRIGGER

When Special Event Trigger mode is chosen (CCP1M<3:0> = 1011), the CCP module does the following:

- Resets Timer1
- Starts an ADC conversion if ADC is enabled

The CCP module does not assert control of the CCP1 pin in this mode (see the CCP1CON register).

The Special Event Trigger output of the CCP occurs immediately upon a match between the TMR1H, TMR1L register pair and the CCPR1H, CCPR1L register pair. The TMR1H, TMR1L register pair is not reset until the next rising edge of the Timer1 clock. This allows the CCPR1H, CCPR1L register pair to effectively provide a 16-bit programmable period register for Timer1.

**Note:** The Special Event Trigger from the CCP module does not set interrupt flag bit TMR1IF of the PIR1 register.

2: Removing the match condition by changing the contents of the CCPR1H and CCPR1L register pair, between the clock edge that generates the Special Event Trigger and the clock edge that generates the Timer1 Reset, will preclude the Reset from occurring.
11.3 PWM Mode

The PWM mode generates a Pulse-Width Modulated signal on the CCP1 pin. The duty cycle, period and resolution are determined by the following registers:

- PR2
- T2CON
- CCPR1L
- CCP1CON

In Pulse-Width Modulation (PWM) mode, the CCP module produces up to a 10-bit resolution PWM output on the CCP1 pin. Since the CCP1 pin is multiplexed with the PORT data latch, the TRIS for that pin must be cleared to enable the CCP1 pin output driver.

**Note:** Clearing the CCP1CON register will relinquish CCP1 control of the CCP1 pin.

Figure 11-3 shows a simplified block diagram of PWM operation.

Figure 11-4 shows a typical waveform of the PWM signal.

For a step by step procedure on how to set up the CCP module for PWM operation, see Section 11.3.7 “Setup for PWM Operation”.

**FIGURE 11-3: SIMPLIFIED PWM BLOCK DIAGRAM**

The PWM output (Figure 11-4) has a time base (period) and a time that the output stays high (duty cycle).

**FIGURE 11-4: CCP PWM OUTPUT**

---

1. The 8-bit timer TMR2 register is concatenated with the 2-bit internal system clock (Fosc), or 2 bits of the prescaler, to create the 10-bit time base.
2. In PWM mode, CCPR1H is a read-only register.
11.3.1 PWM PERIOD

The PWM period is specified by the PR2 register of Timer2. The PWM period can be calculated using the formula of Equation 11-1.

**EQUATION 11-1: PWM PERIOD**

\[
\text{PWM Period} = \frac{(PR2 + 1)}{4} \cdot \text{Tosc} \cdot \frac{1}{(\text{TMR2 Prescale Value})}
\]

When TMR2 is equal to PR2, the following three events occur on the next increment cycle:

- TMR2 is cleared
- The CCP1 pin is set. (Exception: If the PWM duty cycle = 0%, the pin will not be set.)
- The PWM duty cycle is latched from CCPR1L into CCP1H.

**Note:** The Timer2 postscaler (see Section 7.1 “Timer2 Operation”) is not used in the determination of the PWM frequency.

11.3.2 PWM DUTY CYCLE

The PWM duty cycle is specified by writing a 10-bit value to multiple registers: CCPR1L register and CCP1<1:0> bits of the CCP1CON register. The CCPR1L contains the eight MSbs and the CCP1<1:0> bits of the CCP1CON register contain the two LSbs. CCPR1L and CCP1<1:0> bits of the CCP1CON register can be written to at any time. The duty cycle value is not latched into CCPR1H until after the period completes (i.e., a match between PR2 and TMR2 registers occurs). While using the PWM, the CCPR1H register is read-only.

Equation 11-2 is used to calculate the PWM pulse width.

Equation 11-3 is used to calculate the PWM duty cycle ratio.

**EQUATION 11-2: PULSE WIDTH**

\[
Pulse \text{ Width} = (\text{CCPR1L}:\text{CCP1CON}<5:4>) \cdot \frac{\text{Tosc}}{(\text{TMR2 Prescale Value})}
\]

**EQUATION 11-3: DUTY CYCLE RATIO**

\[
\text{Duty Cycle Ratio} = \frac{(\text{CCPR1L}:\text{CCP1CON}<5:4>)}{4(PR2 + 1)}
\]

The CCPR1H register and a 2-bit internal latch are used to double buffer the PWM duty cycle. This double buffering is essential for glitchless PWM operation.

The 8-bit timer TMR2 register is concatenated with either the 2-bit internal system clock (Fosc), or 2 bits of the prescaler, to create the 10-bit time base. The system clock is used if the Timer2 prescaler is set to 1:1.

When the 10-bit time base matches the CCPR1H and 2-bit latch, then the CCP1 pin is cleared (see Figure 11-3).

11.3.3 PWM RESOLUTION

The resolution determines the number of available duty cycles for a given period. For example, a 10-bit resolution will result in 1024 discrete duty cycles, whereas an 8-bit resolution will result in 256 discrete duty cycles.

The maximum PWM resolution is 10 bits when PR2 is 255. The resolution is a function of the PR2 register value as shown by Equation 11-4.

**EQUATION 11-4: PWM RESOLUTION**

\[
\text{Resolution} = \log_{2}\left(\frac{4(PR2 + 1)}{\log(2)}\right) \text{ bits}
\]

**Note:** If the pulse width value is greater than the period the assigned PWM pin(s) will remain unchanged.

### TABLE 11-2: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS (Fosc = 20 MHz)

<table>
<thead>
<tr>
<th>PWM Frequency</th>
<th>1.22 kHz</th>
<th>4.88 kHz</th>
<th>19.53 kHz</th>
<th>78.12 kHz</th>
<th>156.3 kHz</th>
<th>208.3 kHz</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timer Prescale (1, 4, 16)</td>
<td>16</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>PR2 Value</td>
<td>0xFF</td>
<td>0xFF</td>
<td>0xFF</td>
<td>0x3F</td>
<td>0x1F</td>
<td>0x17</td>
</tr>
<tr>
<td>Maximum Resolution (bits)</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>8</td>
<td>7</td>
<td>6.6</td>
</tr>
</tbody>
</table>

### TABLE 11-3: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS (Fosc = 8 MHz)

<table>
<thead>
<tr>
<th>PWM Frequency</th>
<th>1.22 kHz</th>
<th>4.90 kHz</th>
<th>19.61 kHz</th>
<th>76.92 kHz</th>
<th>153.85 kHz</th>
<th>200.0 kHz</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timer Prescale (1, 4, 16)</td>
<td>16</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>PR2 Value</td>
<td>0x65</td>
<td>0x65</td>
<td>0x65</td>
<td>0x19</td>
<td>0x0C</td>
<td>0x09</td>
</tr>
<tr>
<td>Maximum Resolution (bits)</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>6</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>
11.3.4 OPERATION IN SLEEP MODE
In Sleep mode, the TMR2 register will not increment and the state of the module will not change. If the CCP1 pin is driving a value, it will continue to drive that value. When the device wakes up, TMR2 will continue from its previous state.

11.3.5 CHANGES IN SYSTEM CLOCK FREQUENCY
The PWM frequency is derived from the system clock frequency. Any changes in the system clock frequency will result in changes to the PWM frequency. See Section 3.0 “Oscillator Module (With Fail-Safe Clock Monitor)” for additional details.

11.3.6 EFFECTS OF RESET
Any Reset will force all ports to Input mode and the CCP registers to their Reset states.

11.3.7 SETUP FOR PWM OPERATION
The following steps should be taken when configuring the CCP module for PWM operation:
1. Disable the PWM pin (CCP1) output driver by setting the associated TRIS bit.
2. Set the PWM period by loading the PR2 register.
3. Configure the CCP module for the PWM mode by loading the CCP1CON register with the appropriate values.
4. Set the PWM duty cycle by loading the CCPR1L register and CCP1 bits of the CCP1CON register.
5. Configure and start Timer2:
   • Clear the TMR2IF interrupt flag bit of the PIR1 register.
   • Set the Timer2 prescale value by loading the T2CKPS bits of the T2CON register.
   • Enable Timer2 by setting the TMR2ON bit of the T2CON register.
6. Enable PWM output after a new PWM cycle has started:
   • Wait until Timer2 overflows (TMR2IF bit of the PIR1 register is set).
   • Enable the CCP1 pin output driver by clearing the associated TRIS bit.
11.4 PWM (Enhanced Mode)

The Enhanced PWM Mode can generate a PWM signal on up to four different output pins with up to 10-bits of resolution. It can do this through four different PWM output modes:

- Single PWM
- Half-Bridge PWM
- Full-Bridge PWM, Forward mode
- Full-Bridge PWM, Reverse mode

To select an Enhanced PWM mode, the P1M bits of the CCP1CON register must be set appropriately.

The PWM outputs are multiplexed with I/O pins and are designated P1A, P1B, P1C and P1D. The polarity of the PWM pins is configurable and is selected by setting the CCP1M bits in the CCP1CON register appropriately.

Table 11-4 shows the pin assignments for each Enhanced PWM mode.

Figure 11-5 shows an example of a simplified block diagram of the Enhanced PWM module.

**TABLE 11-4: EXAMPLE PIN ASSIGNMENTS FOR VARIOUS PWM ENHANCED MODES**

<table>
<thead>
<tr>
<th>ECCP Mode</th>
<th>P1M</th>
<th>CCP1/P1A</th>
<th>P1B</th>
<th>P1C</th>
<th>P1D</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single</td>
<td>00</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Half-Bridge</td>
<td>10</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Full-Bridge, Forward</td>
<td>01</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Full-Bridge, Reverse</td>
<td>11</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Note 1: The 8-bit timer TMR2 register is concatenated with the 2-bit internal Q clock, or 2 bits of the prescaler to create the 10-bit time base.

Note 2: Clearing the CCP1CON register will relinquish ECCP control of all PWM output pins.

Note 3: Any pin not used by an Enhanced PWM mode is available for alternate pin functions.
FIGURE 11-6: EXAMPLE PWM (ENHANCED MODE) OUTPUT RELATIONSHIPS (ACTIVE-HIGH STATE)

<table>
<thead>
<tr>
<th>P1M&lt;1:0&gt;</th>
<th>Signal</th>
<th>00 (Single Output)</th>
<th>10 (Half-Bridge)</th>
<th>01 (Full-Bridge, Forward)</th>
<th>11 (Full-Bridge, Reverse)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PR2+1</td>
<td>P1A Modulated</td>
<td>P1B Modulated</td>
<td>P1A Active</td>
<td>P1B Modulated</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 (Pulse Width)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Period</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Delay(I)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Delay(I)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Relationships:
- Period = 4 * Tosc * (PR2 + 1) * (TMR2 Prescale Value)
- Pulse Width = Tosc * (CCPR1L<7:0>:CCP1CON<5:4>) * (TMR2 Prescale Value)
- Delay = 4 * Tosc * (PWM1CON<6:0>)

Note 1: Dead-band delay is programmed using the PWM1CON register (Section 11.4.6 “Programmable Dead-Band Delay mode”).
FIGURE 11-7: EXAMPLE ENHANCED PWM OUTPUT RELATIONSHIPS (ACTIVE-LOW STATE)

Relationships:
- Period = $4 \times \text{Tosc} \times (\text{PR2} + 1) \times (\text{TMR2 Prescale Value})$
- Pulse Width = $\text{Tosc} \times (\text{CCPR1L<7:0>:CCP1CON<5:4>}) \times (\text{TMR2 Prescale Value})$
- Delay = $4 \times \text{Tosc} \times (\text{PWM1CON<6:0>})$

Note 1: Dead-band delay is programmed using the PWM1CON register (Section 11.4.6 “Programmable Dead-Band Delay mode”).
11.4.1 HALF-BRIDGE MODE

In Half-Bridge mode, two pins are used as outputs to drive push-pull loads. The PWM output signal is output on the CCP1/P1A pin, while the complementary PWM output signal is output on the P1B pin (see Figure 11-16). This mode can be used for half-bridge applications, as shown in Figure 11-17, or for full-bridge applications, where four power switches are being modulated with two PWM signals.

In Half-Bridge mode, the programmable dead-band delay can be used to prevent shoot-through current in half-bridge power devices. The value of the PDC<6:0> bits of the PWM1CON register sets the number of instruction cycles before the output is driven active. If the value is greater than the duty cycle, the corresponding output remains inactive during the entire cycle. See Section 11.4.6 “Programmable Dead-Band Delay mode” for more details of the dead-band delay operations.

Since the P1A and P1B outputs are multiplexed with the PORT data latches, the associated TRIS bits must be cleared to configure P1A and P1B as outputs.

FIGURE 11-8: EXAMPLE OF HALF-BRIDGE PWM OUTPUT

FIGURE 11-9: EXAMPLE OF HALF-BRIDGE APPLICATIONS

<table>
<thead>
<tr>
<th>Period</th>
<th>Pulse Width</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1A(1)</td>
<td>td(1)</td>
</tr>
<tr>
<td>P1B(2)</td>
<td>td(2)</td>
</tr>
</tbody>
</table>

Note 1: At this time, the TMR2 register is equal to the PR2 register.
Note 2: Output signals are shown as active-high.
11.4.2 FULL-BRIDGE MODE

In Full-Bridge mode, all four pins are used as outputs. An example of full-bridge application is shown in Figure 11-10.

In the Forward mode, pin CCP1/P1A is driven to its active state, pin P1D is modulated, while P1B and P1C will be driven to their inactive state as shown in Figure 11-11.

In the Reverse mode, P1C is driven to its active state, pin P1B is modulated, while P1A and P1D will be driven to their inactive state as shown Figure 11-11.

P1A, P1B, P1C and P1D outputs are multiplexed with the PORT data latches. The associated TRIS bits must be cleared to configure the P1A, P1B, P1C and P1D pins as outputs.

FIGURE 11-10: EXAMPLE OF FULL-BRIDGE APPLICATION
FIGURE 11-11: EXAMPLE OF FULL-BRIDGE PWM OUTPUT

<table>
<thead>
<tr>
<th>Forward Mode</th>
<th>Reverse Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1A(2)</td>
<td>P1A(2)</td>
</tr>
<tr>
<td>P1B(2)</td>
<td>P1B(2)</td>
</tr>
<tr>
<td>P1C(2)</td>
<td>P1C(2)</td>
</tr>
<tr>
<td>P1D(2)</td>
<td>P1D(2)</td>
</tr>
</tbody>
</table>

(1)

Note 1: At this time, the TMR2 register is equal to the PR2 register.
2: Output signal is shown as active-high.
11.4.2.1 Direction Change in Full-Bridge Mode

In the Full-Bridge mode, the P1M1 bit in the CCP1CON register allows users to control the forward/reverse direction. When the application firmware changes this direction control bit, the module will change to the new direction on the next PWM cycle.

A direction change is initiated in software by changing the P1M1 bit of the CCP1CON register. The following sequence occurs four Timer2 cycles prior to the end of the current PWM period:

- The modulated outputs (P1B and P1D) are placed in their inactive state.
- The associated unmodulated outputs (P1A and P1C) are switched to drive in the opposite direction.
- PWM modulation resumes at the beginning of the next period.

See Figure 11-12 for an illustration of this sequence.

The Full-Bridge mode does not provide dead-band delay. As one output is modulated at a time, dead-band delay is generally not required. There is a situation where dead-band delay is required. This situation occurs when both of the following conditions are true:

1. The direction of the PWM output changes when the duty cycle of the output is at or near 100%.
2. The turn off time of the power switch, including the power device and driver circuit, is greater than the turn on time.

Figure 11-13 shows an example of the PWM direction changing from forward to reverse, at a near 100% duty cycle. In this example, at time t1, the output P1A and P1D become inactive, while output P1C becomes active. Since the turn off time of the power devices is longer than the turn on time, a shoot-through current will flow through power devices QC and QD (see Figure 11-10) for the duration of ‘t’. The same phenomenon will occur to power devices QA and QB for PWM direction change from reverse to forward.

If changing PWM direction at high duty cycle is required for an application, two possible solutions for eliminating the shoot-through current are:

1. Reduce PWM duty cycle for one PWM period before changing directions.
2. Use switch drivers that can drive the switches off faster than they can drive them on.

Other options to prevent shoot-through current may exist.

FIGURE 11-12: EXAMPLE OF PWM DIRECTION CHANGE

<table>
<thead>
<tr>
<th>Signal</th>
<th>Period(1)</th>
<th>Period</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1A (Active-High)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>P1B (Active-High)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>P1C (Active-High)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>P1D (Active-High)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Pulse Width

Note 1: The direction bit P1M1 of the CCP1CON register is written any time during the PWM cycle.

2: When changing directions, the P1A and P1C signals switch before the end of the current PWM cycle. The modulated P1B and P1D signals are inactive at this time. The length of this time is four Timer2 counts.
11.4.3 START-UP CONSIDERATIONS
When any PWM mode is used, the application hardware must use the proper external pull-up and/or pull-down resistors on the PWM output pins.

Note: When the microcontroller is released from Reset, all of the I/O pins are in the high-impedance state. The external circuits must keep the power switch devices in the OFF state until the microcontroller drives the I/O pins with the proper signal levels or activates the PWM output(s).

The CCP1M<1:0> bits of the CCP1CON register allow the user to choose whether the PWM output signals are active-high or active-low for each pair of PWM output pins (P1A/P1C and P1B/P1D). The PWM output polarities must be selected before the PWM pin output drivers are enabled. Changing the polarity configuration while the PWM pin output drivers are enabled is not recommended since it may result in damage to the application circuits.

The P1A, P1B, P1C and P1D output latches may not be in the proper states when the PWM module is initialized. Enabling the PWM pin output drivers at the same time as the Enhanced PWM modes may cause damage to the application circuit. The Enhanced PWM modes must be enabled in the proper Output mode and complete a full PWM cycle before enabling the PWM pin output drivers. The completion of a full PWM cycle is indicated by the TMR2IF bit of the PIR1 register being set as the second PWM period begins.
11.4.4 ENHANCED PWM AUTO-SHUTDOWN MODE

The PWM mode supports an Auto-Shutdown mode that will disable the PWM outputs when an external shutdown event occurs. Auto-Shutdown mode places the PWM output pins into a predetermined state. This mode is used to help prevent the PWM from damaging the application.

The auto-shutdown sources are selected using the ECCPASx bits of the ECCPAS register. A shutdown event may be generated by:
- A logic ‘0’ on the INT pin
- Comparator 1
- Comparator 2
- Setting the ECCPASE bit in firmware

A shutdown condition is indicated by the ECCPASE (Auto-Shutdown Event Status) bit of the ECCPAS register. If the bit is a ‘0’, the PWM pins are operating normally. If the bit is a ‘1’, the PWM outputs are in the shutdown state.

When a shutdown event occurs, two things happen:
- The ECCPASE bit is set to ‘1’. The ECCPASE will remain set until cleared in firmware or an auto-restart occurs (see Section 11.4.5 “Auto-Restart Mode”).
- The enabled PWM pins are asynchronously placed in their shutdown states. The PWM output pins are grouped into pairs [P1A/P1C] and [P1B/P1D]. The state of each pin pair is determined by the PSSAC and PSSBD bits of the ECCPAS register. Each pin pair may be placed into one of three states:
  - Drive logic ‘1’
  - Drive logic ‘0’
  - Tri-state (high-impedance)

REGISTER 11-2: ECCPAS: ENHANCED CAPTURE/COMPARE/PWM AUTO-SHUTDOWN CONTROL REGISTER

<table>
<thead>
<tr>
<th>U-0</th>
<th>U-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECCPASE</td>
<td>ECCPAS2</td>
<td>ECCPAS1</td>
<td>ECCPAS0</td>
<td>PSSAC1</td>
<td>PSSAC0</td>
<td>PSSBD1</td>
<td>PSSBD0</td>
</tr>
<tr>
<td>bit 7</td>
<td>bit 6-4</td>
<td>bit 5-4</td>
<td>bit 3-2</td>
<td>bit 1-0</td>
<td>bit 0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

bit 7  **ECCPASE**: ECCP Auto-Shutdown Event Status bit
- 1 = A shutdown event has occurred; ECCP outputs are in shutdown state
- 0 = ECCP outputs are operating

bit 6-4 **ECCPAS<2:0>**: ECCP Auto-shutdown Source Select bits
- 000 = Auto-Shutdown is disabled
- 001 = Comparator 1 output change
- 010 = Comparator 2 output change
- 011 = Either Comparator 1 or 2 change
- 100 = VIL on INT pin
- 101 = VIL on INT pin or Comparator 1 change
- 110 = VIL on INT pin or Comparator 2 change
- 111 = VIL on INT pin or Comparator 1 or 2 change

bit 3-2 **PSSACn**: Pins P1A and P1C Shutdown State Control bits
- 00 = Drive pins P1A and P1C to ‘0’
- 01 = Drive pins P1A and P1C to ‘1’
- 1x = Pins P1A and P1C tri-state

bit 1-0 **PSSBDn**: Pins P1B and P1D Shutdown State Control bits
- 00 = Drive pins P1B and P1D to ‘0’
- 01 = Drive pins P1B and P1D to ‘1’
- 1x = Pins P1B and P1D tri-state
11.4.5 AUTO-RESTART MODE

The Enhanced PWM can be configured to automatically restart the PWM signal once the auto-shutdown condition has been removed. Auto-restart is enabled by setting the PRSEN bit in the PWM1CON register.

If auto-restart is enabled, the ECCPASE bit will remain set as long as the auto-shutdown condition is active. When the auto-shutdown condition is removed, the ECCPASE bit will be cleared via hardware and normal operation will resume.

### FIGURE 11-15: PWM AUTO-SHUTDOWN WITH AUTO-RESTART ENABLED (PRSEN = 1)
11.4.6 PROGRAMMABLE DEAD-BAND DELAY MODE

In half-bridge applications where all power switches are modulated at the PWM frequency, the power switches normally require more time to turn off than to turn on. If both the upper and lower power switches are switched at the same time (one turned on, and the other turned off), both switches may be on for a short period of time until one switch completely turns off. During this brief interval, a very high current (shoot-through current) will flow through both power switches, shorting the bridge supply. To avoid this potentially destructive shoot-through current from flowing during switching, turning on either of the power switches is normally delayed to allow the other switch to completely turn off.

In Half-Bridge mode, a digitally programmable dead-band delay is available to avoid shoot-through current from destroying the bridge power switches. The delay occurs at the signal transition from the non-active state to the active state. See Section FIGURE 11-17: “Example of Half-Bridge Applications” for illustration. The lower seven bits of the associated PWM1ICON register (Register 11-3) sets the delay period in terms of microcontroller instruction cycles (TCY or 4 Tosc).

FIGURE 11-17: EXAMPLE OF HALF-BRIDGE APPLICATIONS

Note 1: At this time, the TMR2 register is equal to the PR2 register.
2: Output signals are shown as active-high.
## REGISTER 11-3: PWM1CON: ENHANCED PWM CONTROL REGISTER

<table>
<thead>
<tr>
<th>Bit</th>
<th>PRSEN</th>
<th>PDC6</th>
<th>PDC5</th>
<th>PDC4</th>
<th>PDC3</th>
<th>PDC2</th>
<th>PDC1</th>
<th>PDC0</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
</tr>
<tr>
<td>0</td>
<td>bit 7</td>
<td>bit 6-0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 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

#### bit 7: PRSEN
- **PWM Restart Enable bit**
  - **1** = Upon auto-shutdown, the ECCPASE bit clears automatically once the shutdown event goes away; the PWM restarts automatically
  - **0** = Upon auto-shutdown, ECCPASE must be cleared in software to restart the PWM

#### bit 6-0: PDC<6:0>:
- **PWM Delay Count bits**
  - **PDCn** = Number of Fosc/4 (4 * Tosc) cycles between the scheduled time when a PWM signal should transition active and the actual time it transitions active

### Note 1:
- Bit resets to ‘0’ with Two-Speed Start-up and LP, XT or HS selected as the Oscillator mode or Fail-Safe mode is enabled.

## TABLE 11-5: SUMMARY OF REGISTERS ASSOCIATED WITH CAPTURE, COMPARE AND PWM

<table>
<thead>
<tr>
<th>Name</th>
<th>Bit 7</th>
<th>Bit 6</th>
<th>Bit 5</th>
<th>Bit 4</th>
<th>Bit 3</th>
<th>Bit 2</th>
<th>Bit 1</th>
<th>Bit 0</th>
<th>Value on: POR, BOR</th>
<th>Value on all other Resets</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCPR1L</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>xxxxx xxxxx</td>
<td>uuuu uuuu</td>
</tr>
<tr>
<td>CCPR1H</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>xxxxx xxxxx</td>
<td>uuuu uuuu</td>
</tr>
<tr>
<td>CCP1CON</td>
<td>P1M1</td>
<td>P1M0</td>
<td>DC1B1</td>
<td>DC1B0</td>
<td>CCP1M3</td>
<td>CCP1M2</td>
<td>CCP1M1</td>
<td>CCP1M0</td>
<td>0000 0000</td>
<td>0000 0000</td>
</tr>
<tr>
<td>CMCON0</td>
<td>C2OUT</td>
<td>C1OUT</td>
<td>C2INV</td>
<td>C1INV</td>
<td>CIS</td>
<td>CM2</td>
<td>CM1</td>
<td>CM0</td>
<td>0000 0000</td>
<td>0000 0000</td>
</tr>
<tr>
<td>CMCON1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0000 0000</td>
<td>0000 0000</td>
</tr>
<tr>
<td>ECCPAS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ECCPAS</td>
<td>ECCPAS</td>
<td>ECCPAS</td>
<td>ECCPAS</td>
<td>0000 0000</td>
<td>0000 0000</td>
</tr>
<tr>
<td>INTCON</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>GIE</td>
<td>PEIE</td>
<td>T0IE</td>
<td>RAIE</td>
<td>0000 0000</td>
<td>0000 0000</td>
</tr>
<tr>
<td>PIE1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>EEIE</td>
<td>ADIE</td>
<td>CCP1IE</td>
<td>C2IE</td>
<td>0000 0000</td>
<td>0000 0000</td>
</tr>
<tr>
<td>PIR1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>EEIF</td>
<td>ADIF</td>
<td>CCP1IF</td>
<td>C2IF</td>
<td>0000 0000</td>
<td>0000 0000</td>
</tr>
<tr>
<td>PR2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1111 1111</td>
<td>1111 1111</td>
</tr>
<tr>
<td>PWM1CON</td>
<td>PRSEN</td>
<td>PDC6</td>
<td>PDC5</td>
<td>PDC4</td>
<td>PDC3</td>
<td>PDC2</td>
<td>PDC1</td>
<td>PDC0</td>
<td>0000 0000</td>
<td>0000 0000</td>
</tr>
<tr>
<td>T1CON</td>
<td>T1GINV</td>
<td>TMR1GE</td>
<td>T1CKPS1</td>
<td>T1CKPS0</td>
<td>T1OSCEN</td>
<td>T1SYNC</td>
<td>TMR1CS</td>
<td>TMR1ON</td>
<td>0000 0000</td>
<td>0000 0000</td>
</tr>
<tr>
<td>T2CON</td>
<td>TOUTPS3</td>
<td>TOUTPS2</td>
<td>TOUTPS1</td>
<td>TOUTPS0</td>
<td>TMR2ON</td>
<td>T2CKPS1</td>
<td>T2CKPS0</td>
<td>-000 0000</td>
<td>-000 0000</td>
<td>-000 0000</td>
</tr>
<tr>
<td>TMR1L</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>xxxxx xxxxx</td>
<td>uuuu uuuu</td>
</tr>
<tr>
<td>TMR1H</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>xxxxx xxxxx</td>
<td>uuuu uuuu</td>
</tr>
<tr>
<td>TMR2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0000 0000</td>
<td>0000 0000</td>
</tr>
</tbody>
</table>

### Legend:
- **- = Unimplemented locations, read as ‘0’**, **u = unchanged**, **x = unknown**
- Shaded cells are not used by the Capture, Compare and PWM.

---

DS41202D-page 96 © 2006 Microchip Technology Inc.
12.0 SPECIAL FEATURES OF THE CPU

The PIC16F684 has a host of features intended to maximize system reliability, minimize cost through elimination of external components, provide power saving features and offer code protection.

These features are:
- Reset
  - Power-on Reset (POR)
  - Power-up Timer (PWRT)
  - Oscillator Start-up Timer (OST)
  - Brown-out Reset (BOR)
- Interrupts
- Watchdog Timer (WDT)
- Oscillator selection
- Sleep
- Code protection
- ID Locations
- In-Circuit Serial Programming

The PIC16F684 has 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 64 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 use the Power-up Timer to provide at least a 64 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 12-1).

12.1 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 12-1. 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 “PIC12F6XX/16F6XX Memory Programming Specification” (DS41204) for more information.
REGISTER 12-1: CONFIG: CONFIGURATION WORD REGISTER

| bit 15-12 | Unimplemented: Read as ‘1’ |
| bit 11    | FCMEN: Fail-Safe Clock Monitor Enabled bit |
|          | 1 = Fail-Safe Clock Monitor is enabled |
|          | 0 = Fail-Safe Clock Monitor is disabled |
| bit 10   | IESO: Internal External Switchover bit |
|          | 1 = Internal External Switchover mode is enabled |
|          | 0 = Internal External Switchover mode is disabled |
| bit 9-8  | BOREN<1:0>: Brown-out Reset Selection bits<sup>(1)</sup> |
|          | 11 = BOR enabled |
|          | 10 = BOR enabled during operation and disabled in Sleep |
|          | 01 = BOR controlled by SBOREN bit of the PCON register |
|          | 00 = BOR disabled |
| bit 7    | CPD: Data Code Protection bit<sup>(2)</sup> |
|          | 1 = Data memory code protection is disabled |
|          | 0 = Data memory code protection is enabled |
| bit 6    | CP: Code Protection bit<sup>(3)</sup> |
|          | 1 = Program memory code protection is disabled |
|          | 0 = Program memory code protection is enabled |
| bit 5    | MCLRE: RA3/MCLR pin function select bit<sup>(4)</sup> |
|          | 1 = RA3/MCLR pin function is MCLR |
|          | 0 = RA3/MCLR pin function is digital input, MCLR internally tied to VDD |
| bit 4    | 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 and can be enabled by SWDTEN bit of the WDTCN register |
| bit 2-0  | FOSC<2:0>: Oscillator Selection bits |
|          | 111 = RC oscillator: CLKOUT function on RA4/OSC2/CLKOUT pin, RC on RA5/OSC1/CLKIN |
|          | 110 = RCIO 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 = INTOSCIO 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: Enabling Brown-out Reset does not automatically enable Power-up Timer.
2: The entire data EEPROM will be erased when the code protection is turned off.
3: The entire program memory will be erased when the code protection is turned off.
4: When MCLR is asserted in INTOSC or RC mode, the internal clock oscillator is disabled.
12.2 Calibration Bits

Brown-out Reset (BOR), Power-on Reset (POR) and 8 MHz internal oscillator (HFINTOSC) are factory calibrated. These calibration values are stored in fuses located in the Calibration Word (2009h). The Calibration Word is not erased when using the specified bulk erase sequence in the “PIC12F6XX/16F6XX Memory Programming Specification” (DS41244) and thus, does not require reprogramming.

12.3 Reset

The PIC16F684 differentiates between various kinds of Reset:

a) Power-on Reset (POR)
b) WDT Reset during normal operation
c) WDT Reset during Sleep
d) MCLR Reset during normal operation
e) MCLR Reset during Sleep
f) Brown-out Reset (BOR)

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
- MCLR Reset
- MCLR Reset during Sleep
- WDT Reset
- Brown-out Reset (BOR)

WDT wake-up does not cause register resets in the same manner as a WDT Reset since wake-up is viewed as the resumption of normal operation. TO and PD bits are set or cleared differently in different Reset situations, as indicated in Table 12-2. Software can use these bits to determine the nature of the Reset. See Table 12-4 for a full description of Reset states of all registers.

A simplified block diagram of the On-Chip Reset Circuit is shown in Figure 12-1.

FIGURE 12-1: SIMPLIFIED BLOCK DIAGRAM OF ON-CHIP RESET CIRCUIT

Note 1: Refer to the Configuration Word register (Register 12-1).
12.3.1 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 connect 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 15.0 “Electrical Specifications” for details. If the BOR is enabled, the maximum rise time specification does not apply. The BOR circuitry will keep the device in Reset until VDD reaches VBOR (see Section 12.3.4 “Brown-Out Reset (BOR)”).

**Note:** The POR circuit does not produce an internal Reset when VDD declines. To re-enable the POR, VDD must reach Vss for a minimum of 100 μs.

When the device starts normal operation (exits the Reset condition), device operating parameters (i.e., voltage, frequency, temperature, etc.) must be met to ensure proper 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” (DS00607).

12.3.2 MCLR
PIC16F684 has 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.

Voltages applied to the MCLR 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 12-2, is suggested.

An internal MCLR option is enabled by clearing the MCLRE bit in the Configuration Word register. When MCLRE = 0, the Reset signal to the chip is generated internally. When the MCLRE = 1, the RA3/MCLR pin becomes an external Reset input. In this mode, the RA3/MCLR pin has a weak pull-up to VDD.

12.3.3 POWER-UP TIMER (PWRT)
The Power-up Timer provides a fixed 64 ms (nominal) time-out on power-up only, from POR or Brown-out Reset. The Power-up Timer operates from the 31 kHz LFINTOSC oscillator. For more information, see Section 3.5 “Internal Clock Modes”. 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, PWRT, can disable (if set) or enable (if cleared or programmed) the Power-up Timer. The Power-up Timer should be enabled when Brown-out Reset is enabled, although it is not required.

The Power-up Timer delay will varies from chip-to-chip due to:
- VDD variation
- Temperature variation
- Process variation

See DC parameters for details (Section 15.0 “Electrical Specifications”).

**Note:** Voltage spikes below Vss at the 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 MCLR pin, rather than pulling this pin directly to Vss.
12.3.4 BROWN-OUT RESET (BOR)

The BOREN0 and BOREN1 bits in the Configuration Word register select one of four BOR modes. Two modes have been added to allow software or hardware control of the BOR enable. When BOREN<1:0> = 01, the SBOREN bit of the PCON register enables/disables the BOR, allowing it to be controlled in software. By selecting BOREN<1:0> = 10, the BOR is automatically disabled in Sleep to conserve power and enabled on wake-up. In this mode, the SBOREN bit is disabled. See Register 12-1 for the Configuration Word definition.

A brown-out occurs when VDD falls below VBOR for greater than parameter TBOR (see Section 15.0 “Electrical Specifications”). The brown-out condition will reset the device. This will occur regardless of VDD slew rate. A Brown-out Reset may not occur if VDD falls below VBOR for less than parameter TBOR.

On any Reset (Power-on, Brown-out Reset, Watchdog Timer, etc.), the chip will remain in Reset until VDD rises above VBOR (see Figure 12-3). If enabled, the Power-up Timer will be invoked by the Reset and keep the chip in Reset an additional 64 ms.

If VDD drops below VBOR while the Power-up Timer is running, the chip will go back into a Brown-out Reset and the Power-up Timer will be re-initialized. Once VDD rises above VBOR, the Power-up Timer will execute a 64 ms Reset.

12.3.5 BOR CALIBRATION

The PIC16F684 stores the BOR calibration values in fuses located in the Calibration Word register (2008h). The Calibration Word register is not erased when using the specified bulk erase sequence in the “PIC12F6XX/16F6XX Memory Programming Specification” (DS41204) and thus, does not require reprogramming.

**Note:** Address 2008h 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 “PIC12F6XX/16F6XX Memory Programming Specification” (DS41204) for more information.

**Note:** The Power-up Timer is enabled by the PWRTE bit in the Configuration Word register.

---

**FIGURE 12-3: BROWN-OUT SITUATIONS**

![Brown-out Situations Diagram](image)

**Note 1:** 64 ms delay only if PWRTE bit is programmed to ‘0’.
12.3.6 TIME-OUT SEQUENCE
On power-up, the time-out sequence is as follows:
• PWRT time-out is invoked after POR has expired.
• OST is activated after the PWRT time-out has expired.

The total time-out will vary based on oscillator configuration and PWRT bit status. For example, in EC mode with PWRT bit erased (PWRT disabled), there will be no time-out at all. Figure 12-4, Figure 12-5 and Figure 12-6 depict time-out sequences. The device can execute code from the INTOSC while OST is active by enabling Two-Speed Start-up or Fail-Safe Monitor (see Section 3.7.2 “Two-speed Start-up Sequence” and Section 3.8 “Fail-Safe Clock Monitor”).

Since the time-outs occur from the POR pulse, if MCLR is kept low long enough, the time-outs will expire. Then, bringing MCLR high will begin execution immediately (see Figure 12-5). This is useful for testing purposes or to synchronize more than one PIC16F684 device operating in parallel.

Table 12-5 shows the Reset conditions for some special registers, while Table 12-4 shows the Reset conditions for all the registers.

TABLE 12-1: TIME-OUT IN VARIOUS SITUATIONS

<table>
<thead>
<tr>
<th>Oscillator Configuration</th>
<th>Power-up</th>
<th>Brown-out Reset</th>
<th>Wake-up from Sleep</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PWRT = 0</td>
<td>PWRT = 1</td>
<td>PWRT = 0</td>
</tr>
<tr>
<td>XT, HS, LP</td>
<td>TPWRT + 1024 • TOSC</td>
<td>1024 • TOSC</td>
<td>TPWRT + 1024 • TOSC</td>
</tr>
<tr>
<td>RC, EC, INTOSC</td>
<td>TPWRT</td>
<td>—</td>
<td>TPWRT</td>
</tr>
</tbody>
</table>

TABLE 12-2: STATUS/PCON BITS AND THEIR SIGNIFICANCE

<table>
<thead>
<tr>
<th>POR</th>
<th>BOR</th>
<th>TO</th>
<th>PD</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>x</td>
<td>1</td>
<td>1</td>
<td>Power-on Reset</td>
</tr>
<tr>
<td>u</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>Brown-out Reset</td>
</tr>
<tr>
<td>u</td>
<td>u</td>
<td>0</td>
<td>u</td>
<td>WDT Reset</td>
</tr>
<tr>
<td>u</td>
<td>u</td>
<td>0</td>
<td>0</td>
<td>WDT Wake-up</td>
</tr>
<tr>
<td>u</td>
<td>u</td>
<td>u</td>
<td>u</td>
<td>MCLR Reset during normal operation</td>
</tr>
<tr>
<td>u</td>
<td>u</td>
<td>1</td>
<td>0</td>
<td>MCLR Reset during Sleep</td>
</tr>
</tbody>
</table>

Legend:  u = unchanged,  x = unknown

TABLE 12-3: SUMMARY OF REGISTERS ASSOCIATED WITH BROWN-OUT RESET

<table>
<thead>
<tr>
<th>Name</th>
<th>Bit 7</th>
<th>Bit 6</th>
<th>Bit 5</th>
<th>Bit 4</th>
<th>Bit 3</th>
<th>Bit 2</th>
<th>Bit 1</th>
<th>Bit 0</th>
<th>Value on: POR, BOR</th>
<th>Value on all other Resets(1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCON</td>
<td>—</td>
<td>—</td>
<td>ULPWUE</td>
<td>SBOREN</td>
<td>—</td>
<td>—</td>
<td>POR</td>
<td>BOR</td>
<td>-01</td>
<td>-00u, -0u</td>
</tr>
<tr>
<td>STATUS</td>
<td>IRP</td>
<td>RP1</td>
<td>RP0</td>
<td>TO</td>
<td>PD</td>
<td>Z</td>
<td>DC</td>
<td>C</td>
<td>0001 xxx</td>
<td>000q quuu</td>
</tr>
</tbody>
</table>

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

Shaded cells are not used by BOR.

Note 1: Other (non Power-up) Resets include MCLR Reset and Watchdog Timer Reset during normal operation.
FIGURE 12-4: TIME-OUT SEQUENCE ON POWER-UP (DELAYED MCLR): CASE 1

FIGURE 12-5: TIME-OUT SEQUENCE ON POWER-UP (DELAYED MCLR): CASE 2

FIGURE 12-6: TIME-OUT SEQUENCE ON POWER-UP (MCLR WITH VDD)
### TABLE 12-4: INITIALIZATION CONDITION FOR REGISTERS

<table>
<thead>
<tr>
<th>Register</th>
<th>Address</th>
<th>Power-on Reset</th>
<th>MCLR Reset</th>
<th>WDT Reset</th>
<th>Brown-out Reset $^{(1)}$</th>
<th>Wake-up from Sleep through Interrupt</th>
<th>Wake-up from Sleep through WDT Time-out</th>
</tr>
</thead>
<tbody>
<tr>
<td>W</td>
<td>—</td>
<td>xxxxxxxxxx</td>
<td>uuuuu</td>
<td>uuuuu</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>INDF</td>
<td>00h/80h</td>
<td>xxxxxxxx</td>
<td>xxxxxxxxx</td>
<td>uuuuu</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TMR0</td>
<td>01h</td>
<td>xxxxxxxx</td>
<td>uuuuu</td>
<td>uuuuu</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PCL</td>
<td>02h/82h</td>
<td>0000 0000</td>
<td>0000 0000</td>
<td>uuuuu</td>
<td>PC + 1 $^{(3)}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STATUS</td>
<td>03h/83h</td>
<td>0001 1xxxxx</td>
<td>000q quuu</td>
<td>uuuuu</td>
<td>quuuq quuu $^{(4)}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FSR</td>
<td>04h/84h</td>
<td>xxxxxxxx</td>
<td>uuuuu</td>
<td>uuuuu</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PORTA $^{(6)}$</td>
<td>05h</td>
<td>--x0 x000</td>
<td>--u0 u000</td>
<td>--uu</td>
<td>uuuu</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PORTC $^{(6)}$</td>
<td>07h</td>
<td>--xx 0000</td>
<td>--uu 0000</td>
<td>--uu</td>
<td>uuuu</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PCLATH</td>
<td>0Ah/8Ah</td>
<td>0000 0000</td>
<td>0000 0000</td>
<td>uuuuu</td>
<td>uuuu</td>
<td></td>
<td></td>
</tr>
<tr>
<td>INTCON</td>
<td>0Bh</td>
<td>0000 0000</td>
<td>uuuuu</td>
<td>uuuuu</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PIR1</td>
<td>0Ch</td>
<td>0000 0000</td>
<td>0000 0000</td>
<td>uuuuuu</td>
<td>uuuuuu</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TMR1L</td>
<td>0Dh</td>
<td>xxxxxxxx</td>
<td>uuuuu</td>
<td>uuuuu</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TMR1H</td>
<td>0Fh</td>
<td>xxxxxxxx</td>
<td>uuuuu</td>
<td>uuuuu</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T1CON</td>
<td>10h</td>
<td>0000 0000</td>
<td>uuuuu</td>
<td>-uuu</td>
<td>uuuu</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TMR2</td>
<td>11h</td>
<td>0000 0000</td>
<td>0000 0000</td>
<td>uuuuuu</td>
<td>uuuuuu</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T2CON</td>
<td>12h</td>
<td>-000 0000</td>
<td>-000 0000</td>
<td>-uuu</td>
<td>uuuu</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CCP1L</td>
<td>13h</td>
<td>xxxxxxxx</td>
<td>uuuuu</td>
<td>uuuuuu</td>
<td>uuuuuu</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CCP1H</td>
<td>14h</td>
<td>xxxxxxxx</td>
<td>uuuuu</td>
<td>uuuuuu</td>
<td>uuuuuu</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CCP1CON</td>
<td>15h</td>
<td>0000 0000</td>
<td>0000 0000</td>
<td>uuuuuu</td>
<td>uuuuuu</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PWM1CON</td>
<td>16h</td>
<td>0000 0000</td>
<td>0000 0000</td>
<td>uuuuuu</td>
<td>uuuuuu</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ECCPAS</td>
<td>17h</td>
<td>0000 0000</td>
<td>0000 0000</td>
<td>uuuuuu</td>
<td>uuuuuu</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WDTCON</td>
<td>18h</td>
<td>--0 1000</td>
<td>--0 1000</td>
<td>-uuu</td>
<td>uuuu</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CMCON0</td>
<td>19h</td>
<td>0000 0000</td>
<td>0000 0000</td>
<td>uuuuuu</td>
<td>uuuuuu</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CMCON1</td>
<td>1Ah</td>
<td>---- --10</td>
<td>----- --10</td>
<td>----- --uu</td>
<td>uuuuuu</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADRESH</td>
<td>1Eh</td>
<td>xxxxxxxx</td>
<td>uuuuu</td>
<td>uuuuuu</td>
<td>uuuuuu</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADCON0</td>
<td>1Fh</td>
<td>00-0 0000</td>
<td>000-0 0000</td>
<td>uu-u</td>
<td>uu-u</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OPTION_REG</td>
<td>81h</td>
<td>1111 1111</td>
<td>1111 1111</td>
<td>uuuuuu</td>
<td>uuuuuu</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TRISA</td>
<td>85h</td>
<td>--11 1111</td>
<td>--11 1111</td>
<td>-uuu</td>
<td>uuuu</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TRISC</td>
<td>87h</td>
<td>--11 1111</td>
<td>--11 1111</td>
<td>-uuu</td>
<td>uuuu</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PIE1</td>
<td>8Ch</td>
<td>0000 0000</td>
<td>0000 0000</td>
<td>uuuuuu</td>
<td>uuuuuu</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PCON</td>
<td>8Ch</td>
<td>--01 --0x</td>
<td>--0u --uq</td>
<td>uuuuuu</td>
<td>uuuuuu</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OSCCON</td>
<td>8Fh</td>
<td>-110 x000</td>
<td>-110 q000</td>
<td>uuuuuu</td>
<td>uuuuuu</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**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.

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

**Note 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).

**Note 4:** See Table 12-5 for Reset value for specific condition.

**Note 5:** If Reset was due to brown-out, then bit = 0. All other Resets will cause bit = u.

**Note 6:** Port pins with analog functions controlled by the ANSEL register will read ‘0’ immediately after a Reset even though the data latches are either undefined (POR) or unchanged (other Resets).
### TABLE 12-4: INITIALIZATION CONDITION FOR REGISTERS (CONTINUED)

<table>
<thead>
<tr>
<th>Register</th>
<th>Address</th>
<th>Power-on Reset</th>
<th>MCLR Reset</th>
<th>WDT Reset (Continued)</th>
<th>Brown-out Reset(1)</th>
<th>Wake-up from Sleep through Interrupt Wake-up from Sleep through WDT Time-out (Continued)</th>
</tr>
</thead>
<tbody>
<tr>
<td>OSCTUNE</td>
<td>90h</td>
<td>--0 0000</td>
<td>--u uuuu</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ANSEL</td>
<td>91h</td>
<td>1111 1111</td>
<td>1111 1111</td>
<td></td>
<td>uuuu</td>
<td></td>
</tr>
<tr>
<td>PR2</td>
<td>92h</td>
<td>1111 1111</td>
<td>1111 1111</td>
<td></td>
<td>1111 1111</td>
<td></td>
</tr>
<tr>
<td>WPUA</td>
<td>95h</td>
<td>--11 -111</td>
<td>--00 0000</td>
<td></td>
<td>--uu uuuu</td>
<td></td>
</tr>
<tr>
<td>IOCA</td>
<td>96h</td>
<td>--00 0000</td>
<td>--00 0000</td>
<td></td>
<td>--uu uuuu</td>
<td></td>
</tr>
<tr>
<td>VRCON</td>
<td>99h</td>
<td>0-0- 0000</td>
<td>0-0- 0000</td>
<td></td>
<td>u-u- uuuu</td>
<td></td>
</tr>
<tr>
<td>EEDAT</td>
<td>9Ah</td>
<td>0000 0000</td>
<td>0000 0000</td>
<td></td>
<td>uuuu</td>
<td></td>
</tr>
<tr>
<td>EEADR</td>
<td>9Bh</td>
<td>0000 0000</td>
<td>0000 0000</td>
<td></td>
<td>uuuu</td>
<td></td>
</tr>
<tr>
<td>EECON1</td>
<td>9Ch</td>
<td>---- x000</td>
<td>---- q000</td>
<td></td>
<td>---- uuuu</td>
<td></td>
</tr>
<tr>
<td>EECON2</td>
<td>9Dh</td>
<td>---- -----</td>
<td>---- -----</td>
<td></td>
<td>---- ----</td>
<td></td>
</tr>
<tr>
<td>ADRESL</td>
<td>9 Eh</td>
<td>xxxx xxxx</td>
<td>uuuu uuuu</td>
<td></td>
<td>uuuu</td>
<td></td>
</tr>
<tr>
<td>ADCON1</td>
<td>9Fh</td>
<td>-000 ----</td>
<td>-000 ----</td>
<td></td>
<td>-uuu ----</td>
<td></td>
</tr>
</tbody>
</table>

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.
- One or more bits in INTCON and/or PIR1 will be affected (to cause wake-up).
- When the wake-up is due to an interrupt and the GIE bit is set, the PC is loaded with the interrupt vector (0004h).
- See Table 12-5 for Reset value for specific condition.
- If Reset was due to brown-out, then bit 0 = 0. All other Resets will cause bit 0 = u.
- Port pins with analog functions controlled by the ANSEL register will read ‘0’ immediately after a Reset even though the data latches are either undefined (POR) or unchanged (other Resets).

### TABLE 12-5: INITIALIZATION CONDITION FOR SPECIAL REGISTERS

<table>
<thead>
<tr>
<th>Condition</th>
<th>Program Counter</th>
<th>Status Register</th>
<th>PCON Register</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power-on Reset</td>
<td>000h</td>
<td>0001 lxxx</td>
<td>--01 --0x</td>
</tr>
<tr>
<td>MCLR Reset during normal operation</td>
<td>000h</td>
<td>000u uuuu</td>
<td>--0u --uu</td>
</tr>
<tr>
<td>MCLR Reset during Sleep</td>
<td>000h</td>
<td>001 0uuu</td>
<td>--0u --uu</td>
</tr>
<tr>
<td>WDT Reset</td>
<td>000h</td>
<td>0000 uuuu</td>
<td>--0u --uu</td>
</tr>
<tr>
<td>WDT Wake-up</td>
<td>PC + 1</td>
<td>uuu0 0uuu</td>
<td>--uu --uu</td>
</tr>
<tr>
<td>Brown-out Reset</td>
<td>000h</td>
<td>001 1uuu</td>
<td>--01 --u0</td>
</tr>
<tr>
<td>Interrupt Wake-up from Sleep</td>
<td>PC + 1(1)</td>
<td>uuua 0uuu</td>
<td>--uu --uu</td>
</tr>
</tbody>
</table>

Legend:  
- \( u \) = unchanged,  
- \( x \) = unknown,  
- \( - \) = unimplemented bit, reads as ‘0’.

Note 1:  
- When the wake-up is due to an interrupt and Global Interrupt Enable bit, GIE, is set, the PC is loaded with the interrupt vector (0004h) after execution of PC + 1.
12.4 Interrupts

The PIC16F684 has 11 sources of interrupt:

- External Interrupt RA2/INT
- Timer0 Overflow Interrupt
- PORTA Change Interrupts
- 2 Comparator Interrupts
- A/D Interrupt
- Timer1 Overflow Interrupt
- Timer2 Match Interrupt
- EEPROM Data Write Interrupt
- Fail-Safe Clock Monitor Interrupt
- Enhanced CCP Interrupt

The Interrupt Control register (INTCON) and Peripheral Interrupt Request Register 1 (PIR1) record individual interrupt requests in flag bits. The INTCON register also has individual and global interrupt enable bits.

The Global Interrupt Enable bit, GIE of the INTCON register, enables (if set) all unmasked interrupts, or disables (if cleared) all interrupts. Individual interrupts can be disabled through their corresponding enable bits in the INTCON register and PIE1 register. GIE is cleared on Reset.

When an interrupt is serviced, the following actions occur automatically:

- The GIE is cleared to disable any further interrupt.
- The return address is pushed onto the stack.
- The PC is loaded with 0004h.
- The Return from Interrupt instruction, RETFIE, exits the 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
- Timer0 Overflow Interrupt

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

The following interrupt flags are contained in the PIR1 register:

- EEPROM Data Write Interrupt
- A/D Interrupt
- 2 Comparator Interrupts
- Timer1 Overflow Interrupt
- Timer2 Match Interrupt
- Fail-Safe Clock Monitor Interrupt
- Enhanced CCP Interrupt

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 12-8). 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.

For additional information on Timer1, Timer2, comparators, A/D, data EEPROM or Enhanced CCP modules, refer to the respective peripheral section.

12.4.1 RA2/INT INTERRUPT

The external interrupt on the RA2/INT pin is edge-triggered; either on the rising edge if the INTEDG bit of the OPTION register is set, or the falling edge, if the INTEDG bit is clear. When a valid edge appears on the RA2/INT pin, the INTF bit of the INTCON register is set. This interrupt can be disabled by clearing the INTE control bit of the INTCON register. The INTE bit must be cleared by 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. See Section 12.7 “Power-Down Mode (Sleep)” for details on Sleep and Figure 12-10 for timing of wake-up from Sleep through RA2/INT interrupt.

Note: The ANSEL and CMCON0 registers must be initialized to configure an analog channel as a digital input. Pins configured as analog inputs will read ‘0’ and cannot generate an interrupt.
12.4.2 TIMER0 INTERRUPT
An overflow (FFh → 00h) in the TMR0 register will set the T0IF bit of the INTCON register. The interrupt can be enabled/disabled by setting/clearing T0IE bit of the INTCON register. See Section 5.0 “Timer0 Module” for operation of the Timer0 module.

12.4.3 PORTA INTERRUPT-ON-CHANGE
An input change on PORTA sets the RAIF bit of the INTCON register. The interrupt can be enabled/disabled by setting/clearing the RAIE bit of the INTCON register. Plus, individual pins can be configured through the IOCA register.

Note: If a change on the I/O pin should occur when any PORTA operation is being executed, then the RAIF interrupt flag may not get set.

FIGURE 12-7: INTERRUPT LOGIC

Note 1: Some peripherals depend upon the system clock for operation. Since the system clock is suspended during Sleep, only those peripherals which do not depend upon the system clock will wake the part from Sleep. See Section 12.7.1 “Wake-up from Sleep”.
FIGURE 12-8: INT PIN INTERRUPT TIMING

TABLE 12-6: SUMMARY OF REGISTERS ASSOCIATED WITH INTERRUPTS
12.5 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 and STATUS registers). This must be implemented in software.

Temporary holding registers W_TEMP and STATUS_TEMP should be placed in the last 16 bytes of GPR (see Figure 2-2). These 16 locations are common to all banks and do not require banking. This makes context save and restore operations simpler.

The code shown in Example 12-1 can be used to:

- Store the W register
- Store the STATUS register
- Execute the ISR code
- Restore the Status (and Bank Select Bit register)
- Restore the W register

Note: The PIC16F684 does not require saving the PCLATH. However, if computed GOTOs are used in both the ISR and the main code, the PCLATH must be saved and restored in the ISR.

**EXAMPLE 12-1: SAVING STATUS AND W REGISTERS IN RAM**

<table>
<thead>
<tr>
<th>MOVWF</th>
<th>W_TEMP</th>
<th>;Copy W to TEMP register</th>
</tr>
</thead>
<tbody>
<tr>
<td>SWAPF</td>
<td>STATUS,W</td>
<td>;Swap status to be saved into W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>;Swaps are used because they do not affect the status bits</td>
</tr>
<tr>
<td>MOVWF</td>
<td>STATUS_TEMP</td>
<td>;Save status to bank zero STATUS_TEMP register</td>
</tr>
<tr>
<td>:</td>
<td></td>
<td>;Insert user code here</td>
</tr>
<tr>
<td>:</td>
<td>(ISR)</td>
<td>;Insert user code here</td>
</tr>
<tr>
<td>:</td>
<td></td>
<td>;Sets bank to original state</td>
</tr>
<tr>
<td>SWAPF</td>
<td>STATUS_TEMP,W</td>
<td>;Swap STATUS_TEMP register into W</td>
</tr>
<tr>
<td>MOVWF</td>
<td>STATUS</td>
<td>;Move W into STATUS register</td>
</tr>
<tr>
<td>SWAPF</td>
<td>W_TEMP,F</td>
<td>;Swap W_TEMP</td>
</tr>
<tr>
<td>SWAPF</td>
<td>W_TEMP,W</td>
<td>;Swap W_TEMP into W</td>
</tr>
</tbody>
</table>
12.6 Watchdog Timer (WDT)

The WDT has the following features:

- Operates from the LFINTOSC (31 kHz)
- Contains a 16-bit prescaler
- Shares an 8-bit prescaler with Timer0
- Time-out period is from 1 ms to 268 seconds
- Configuration bit and software controlled

WDT is cleared under certain conditions described in Table 12-7.

12.6.1 WDT OSCILLATOR

The WDT derives its time base from the 31 kHz LFINTOSC. The LTS bit of the OSCCON register does not reflect that the LFINTOSC is enabled.

The value of WDTCON is ‘---0 1000’ on all Resets. This gives a nominal time base of 17 ms.

**Note:** When the Oscillator Start-up Timer (OST) is invoked, the WDT is held in Reset, because the WDT Ripple Counter is used by the OST to perform the oscillator delay count. When the OST count has expired, the WDT will begin counting (if enabled).

12.6.2 WDT CONTROL

The WDTE bit is located in the Configuration Word register. When set, the WDT runs continuously.

When the WDTE bit in the Configuration Word register is set, the SWDTEN bit of the WDTCON register has no effect. If WDTE is clear, then the SWDTEN bit can be used to enable and disable the WDT. Setting the bit will enable it and clearing the bit will disable it.

The PSA and PS<2:0> bits of the OPTION register have the same function as in previous versions of the PIC16F684 Family of microcontrollers. See Section 5.0 "Timer0 Module" for more information.

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

**TABLE 12-7: WDT STATUS**

<table>
<thead>
<tr>
<th>Conditions</th>
<th>WDT</th>
</tr>
</thead>
<tbody>
<tr>
<td>WDTE = 0</td>
<td></td>
</tr>
<tr>
<td>CLRWDT Command</td>
<td>Cleared</td>
</tr>
<tr>
<td>Oscillator Fail Detected</td>
<td></td>
</tr>
<tr>
<td>Exit Sleep + System Clock = T1OSC, EXTRC, INTRC, EXTCLK</td>
<td></td>
</tr>
<tr>
<td>Exit Sleep + System Clock = XT, HS, LP</td>
<td>Cleared until the end of OST</td>
</tr>
</tbody>
</table>
### REGISTER 12-2: WDTCON: WATCHDOG TIMER CONTROL REGISTER

<table>
<thead>
<tr>
<th>Bit 7-5</th>
<th>Unimplemented: Read as ‘0’</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bit 4-1</td>
<td>WDTPS&lt;3:0&gt;: Watchdog Timer Period Select bits</td>
</tr>
<tr>
<td>Bit 0</td>
<td>SWDTEN: Software Enable or Disable the Watchdog Timer&lt;sup&gt;(1)&lt;/sup&gt;</td>
</tr>
</tbody>
</table>

#### Bit 7-5
- Unimplemented: Read as ‘0’

#### Bit 4-1
- **WDTPS<3:0>:** Watchdog Timer Period Select bits
  - Bit Value = Prescale Rate
  - 0000 = 1:32
  - 0001 = 1:64
  - 0010 = 1:128
  - 0011 = 1:256
  - 0100 = 1:512 (Reset value)
  - 0101 = 1:1024
  - 0110 = 1:2048
  - 0111 = 1:4096
  - 1000 = 1:8192
  - 1001 = 1:16384
  - 1010 = 1:32768
  - 1011 = 1:65536
  - 1100 = Reserved
  - 1101 = Reserved
  - 1110 = Reserved
  - 1111 = Reserved

#### Bit 0
- **SWDTEN:** Software Enable or Disable the Watchdog Timer<sup>(1)</sup>
  - 1 = WDT is turned on
  - 0 = WDT is turned off (Reset value)

<sup>Note 1:</sup> If WDTE Configuration bit = 1, then WDT is always enabled, irrespective of this control bit. If WDTE Configuration bit = 0, then it is possible to turn WDT on/off with this control bit.

### TABLE 12-8: SUMMARY OF REGISTERS ASSOCIATED WITH WATCHDOG TIMER

<table>
<thead>
<tr>
<th>Name</th>
<th>Bit 7</th>
<th>Bit 6</th>
<th>Bit 5</th>
<th>Bit 4</th>
<th>Bit 3</th>
<th>Bit 2</th>
<th>Bit 1</th>
<th>Bit 0</th>
<th>Value on: POR, BOR</th>
<th>Value on all other Resets</th>
</tr>
</thead>
<tbody>
<tr>
<td>WDTCON</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>WTPS3</td>
<td>WTPS2</td>
<td>WTPS1</td>
<td>WTPS0</td>
<td>SWDTEN</td>
<td>--- 0 1000</td>
</tr>
<tr>
<td>OPTION_REG</td>
<td>RAPU</td>
<td>INTEDG</td>
<td>T0CS</td>
<td>T0SE</td>
<td>PSA</td>
<td>PS2</td>
<td>PS1</td>
<td>PS0</td>
<td></td>
<td>1111 1111</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1111 1111</td>
</tr>
<tr>
<td>CONFIG</td>
<td>CPD</td>
<td>CP</td>
<td>MCLRE</td>
<td>PWLRT</td>
<td>WDE</td>
<td>FOSC2</td>
<td>FOSC1</td>
<td>FOSC0</td>
<td></td>
<td>—</td>
</tr>
</tbody>
</table>

**Legend:**
- Shaded cells are not used by the Watchdog Timer.
- See Register 12-1 for operation of all Configuration Word register bits.
12.7 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.

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

12.7.1 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 a 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.

The following peripheral interrupts can wake the device from Sleep:
1. Timer1 interrupt. Timer1 must be operating as an asynchronous counter.
2. ECCP Capture mode interrupt.
3. A/D conversion (when A/D clock source is FRC).
4. EEPROM write operation completion.
5. Comparator output changes state.
6. Interrupt-on-change.
7. External Interrupt from INT pin.

Other peripherals cannot generate interrupts since during Sleep, no on-chip clocks are present.

When the SLEEP instruction is being executed, the next instruction (PC + 1) is prefetched. For the device to wake-up through an interrupt event, the corresponding interrupt enable bit must be set (enabled). Wake-up occurs 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 a NOP after the SLEEP instruction.

Note: If the global interrupts are disabled (GIE is cleared) and 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 WDT is cleared when the device wakes up from Sleep, regardless of the source of wake-up.

12.7.2 WAKE-UP USING INTERRUPTS

When global interrupts are disabled (GIE cleared) and any interrupt source has both its interrupt enable bit and interrupt flag bit set, one of the following will occur:
• If the interrupt occurs before the execution of a SLEEP instruction, the SLEEP instruction will complete as a NOP. Therefore, the WDT and WDT prescaler and postscaler (if enabled) will not be cleared, the TO bit will not be set and the PD bit will not be cleared.
• If the interrupt occurs during or after the execution of a SLEEP instruction, the device will immediately wake-up from Sleep. The SLEEP instruction is executed. Therefore, the WDT and WDT prescaler and postscaler (if enabled) will be cleared, the TO bit will be set and the PD bit will be cleared.

Even if the flag bits were checked before executing a SLEEP instruction, it may be possible for flag bits to become set before the SLEEP instruction completes. To determine whether a SLEEP instruction executed, test the PD bit. If the PD bit is set, the SLEEP instruction was executed as a NOP.

To ensure that the WDT is cleared, a CLRWDT instruction should be executed before a SLEEP instruction. See Figure 12-10 for more details.
### 12.8 Code Protection

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

#### Note:
- The entire data EEPROM and Flash program memory will be erased when the code protection is turned off. See the “PIC12F6XX/16F6XX Memory Programming Specification” (DS41204) for more information.

### 12.9 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 mode. Only the Least Significant 7 bits of the ID locations are used.

#### Note:
- XT, HS or LP Oscillator mode assumed.
- 1: \( T_{OST} = 1024 \ T_{OSC} \) (drawing not to scale). This delay does not apply to EC, INTOSC and RC Oscillator modes.
- 2: \( GIE = '1' \) assumed. In this case after wake-up, the processor jumps to 0004h. If \( GIE = '0' \), execution will continue in-line.
- 3: CLKOUT is not available in XT, HS, LP or EC Oscillator modes, but shown here for timing reference.
12.10 In-Circuit Serial Programming

The PIC16F684 microcontrollers can be serially programmed while in the end application circuit. This is simply done with five connections for:

- clock
- data
- 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 MCLR (VPP) pin from VIL to VIHH. See the “PIC12F6XX/16F6XX Memory Programming Specification” (DS41204) for more information. RA0 becomes the programming data and RA1 becomes the programming clock. Both RA0 and RA1 are Schmitt Trigger inputs in Program/Verify mode.

A typical In-Circuit Serial Programming connection is shown in Figure 12-11.

FIGURE 12-11: TYPICAL IN-CIRCUIT SERIAL PROGRAMMING CONNECTION

12.11 In-Circuit Debugger

Since in-circuit debugging requires access to three pins, MPLAB® ICD 2 development with a 14-pin device is not practical. A special 20-pin PIC16F684 ICD device is used with MPLAB ICD 2 to provide separate clock, data and MCLR pins and frees all normally available pins to the user.

A special debugging adapter allows the ICD device to be used in place of a PIC16F684 device. The debugging adapter is the only source of the ICD device.

When the ICD pin on the PIC16F684 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 12-9 shows which features are consumed by the background debugger.

TABLE 12-9: DEBUGGER RESOURCES

<table>
<thead>
<tr>
<th>Resource</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>I/O pins</td>
<td>ICDCLK, ICDDATA</td>
</tr>
<tr>
<td>Stack</td>
<td>1 level</td>
</tr>
<tr>
<td>Program Memory</td>
<td>Address 0h must be NOP 700h-7FFh</td>
</tr>
</tbody>
</table>

For more information, see “MPLAB® ICD 2 In-Circuit Debugger User’s Guide” (DS51331), available on Microchip’s web site (www.microchip.com).

FIGURE 12-12: 20-PIN ICD PINOUT
13.0 INSTRUCTION SET SUMMARY

The PIC16F684 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 13-1, while the various opcode fields are summarized in Table 13-1.

Table 13-2 lists the instructions recognized by the MPASM™ assembler.

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 nominal 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.

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

### 13.1 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 consequence of clearing the condition that set the RAIF flag.

---

**TABLE 13-1: OPCODE FIELD DESCRIPTIONS**

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

**FIGURE 13-1: GENERAL FORMAT FOR INSTRUCTIONS**

- **Byte-oriented file register operations**
  - OPCODE: 13 8 7 6 0
  - d = 0 for destination W
  - d = 1 for destination f
  - f = 7-bit file register address

- **Bit-oriented file register operations**
  - OPCODE: 13 10 9 7 6 0
  - b = 3-bit bit address
  - f = 7-bit file register address

- **Literal and control operations**
  - **General**
    - OPCODE: 13 8 7 0
    - k = 8-bit immediate value
  - **CALL and GOTO instructions only**
    - OPCODE: 13 11 10 0
    - k = 11-bit immediate value
<table>
<thead>
<tr>
<th>Mnemonic, Operands</th>
<th>Description</th>
<th>Cycles</th>
<th>14-Bit Opcode</th>
<th>Status Affected</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>MSb</td>
<td>LSb</td>
<td></td>
</tr>
<tr>
<td><strong>BYTE-ORIENTED FILE REGISTER OPERATIONS</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADDWF f, d</td>
<td>Add W and f</td>
<td>1</td>
<td>00 0111 dfff ffff</td>
<td>C, DC, Z</td>
<td>1, 2</td>
</tr>
<tr>
<td>ANDWF f, d</td>
<td>AND W with f</td>
<td>1</td>
<td>00 0111 dfff ffff</td>
<td>Z</td>
<td>1, 2</td>
</tr>
<tr>
<td>CLRF f</td>
<td>Clear f</td>
<td>1</td>
<td>00 0001 1fff ffff</td>
<td>Z</td>
<td>2</td>
</tr>
<tr>
<td>CLRW – Clear W</td>
<td></td>
<td>1</td>
<td>00 0001 0xxx xxxx</td>
<td>Z</td>
<td>1</td>
</tr>
<tr>
<td>COMF f, d</td>
<td>Complement f</td>
<td>1</td>
<td>00 1001 dfff ffff</td>
<td>Z</td>
<td>1, 2</td>
</tr>
<tr>
<td>DECf f, d</td>
<td>Decrement f</td>
<td>1</td>
<td>00 0011 dfff ffff</td>
<td>Z</td>
<td>1, 2</td>
</tr>
<tr>
<td>DECSFZ f, d</td>
<td>Decrement f, Skip if 0</td>
<td>1(2)</td>
<td>00 1011 dfff ffff</td>
<td>Z, 1, 2</td>
<td></td>
</tr>
<tr>
<td>INCf f, d</td>
<td>Increment f</td>
<td>1</td>
<td>00 1010 dfff ffff</td>
<td>Z</td>
<td>1, 2</td>
</tr>
<tr>
<td>INCSFZ f, d</td>
<td>Increment f, Skip if 0</td>
<td>1(2)</td>
<td>00 1111 dfff ffff</td>
<td>Z, 1, 2</td>
<td></td>
</tr>
<tr>
<td>IORWF f, d</td>
<td>Inclusive OR W with f</td>
<td>1</td>
<td>00 1010 dfff ffff</td>
<td>Z</td>
<td>1, 2</td>
</tr>
<tr>
<td>MOVF f, d</td>
<td>Move f</td>
<td>1</td>
<td>00 1000 dfff ffff</td>
<td>Z</td>
<td>1, 2</td>
</tr>
<tr>
<td>MOVWF f</td>
<td>Move W to f</td>
<td>1</td>
<td>00 1000 lfff ffff</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NOP – No Operation</td>
<td></td>
<td>1</td>
<td>00 0000 0xx0 0000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RLF f, d</td>
<td>Rotate Left f through Carry</td>
<td>1</td>
<td>00 1101 dfff ffff</td>
<td>C</td>
<td>1, 2</td>
</tr>
<tr>
<td>RRF f, d</td>
<td>Rotate Right f through Carry</td>
<td>1</td>
<td>00 1100 dfff ffff</td>
<td>C</td>
<td>1, 2</td>
</tr>
<tr>
<td>SUBWF f, d</td>
<td>Subtract W from f</td>
<td>1</td>
<td>00 0110 dfff ffff</td>
<td>C, DC, Z</td>
<td>1, 2</td>
</tr>
<tr>
<td>SWAPF f, d</td>
<td>Swap nibbles in f</td>
<td>1</td>
<td>00 1110 dfff ffff</td>
<td>C</td>
<td>1, 2</td>
</tr>
<tr>
<td>XORWF f, d</td>
<td>Exclusive OR W with f</td>
<td>1</td>
<td>00 0110 dfff ffff</td>
<td>Z</td>
<td>1, 2</td>
</tr>
<tr>
<td><strong>BIT-ORIENTED FILE REGISTER OPERATIONS</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BCF f, b</td>
<td>Bit Clear f</td>
<td>1</td>
<td>01 00bb bfff ffff</td>
<td>1, 2</td>
<td></td>
</tr>
<tr>
<td>BSF f, b</td>
<td>Bit Set f</td>
<td>1</td>
<td>01 10bb bfff ffff</td>
<td>1, 2</td>
<td></td>
</tr>
<tr>
<td>BTFSC f, b</td>
<td>Bit Test f, Skip if Clear</td>
<td>1(2)</td>
<td>01 10bb bfff ffff</td>
<td>1, 2</td>
<td></td>
</tr>
<tr>
<td>BTBFS f, b</td>
<td>Bit Test f, Skip if Set</td>
<td>1(2)</td>
<td>01 10bb bfff ffff</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td><strong>LITERAL AND CONTROL OPERATIONS</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADDLW k</td>
<td>Add literal and W</td>
<td>1</td>
<td>11 111x kkkk kkkk kkkk</td>
<td>C, DC, Z</td>
<td></td>
</tr>
<tr>
<td>ANDLW k</td>
<td>AND literal with W</td>
<td>1</td>
<td>11 1001 kkkk kkkk kkkk</td>
<td>Z</td>
<td></td>
</tr>
<tr>
<td>CALL k</td>
<td>Call Subroutine</td>
<td>2</td>
<td>10 0kkk kkkk kkkk kkkk</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLRWDT – Clear Watchdog Timer</td>
<td>1</td>
<td>00 0000 0110 0100</td>
<td>TO, PD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GOTO k</td>
<td>Go to address</td>
<td>2</td>
<td>10 1kkk kkkk kkkk kkkk</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IORLW k</td>
<td>Inclusive OR literal with W</td>
<td>1</td>
<td>11 1000 kkkk kkkk kkkk</td>
<td>Z</td>
<td></td>
</tr>
<tr>
<td>MOVLW k</td>
<td>Move literal to W</td>
<td>1</td>
<td>11 0xxx kkkk kkkk kkkk</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RETFIE – Return from interrupt</td>
<td>2</td>
<td>00 0000 0000 1001</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RETLW k</td>
<td>Return with literal in W</td>
<td>2</td>
<td>11 01xx kkkk kkkk kkkk</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RETURN – Return from Subroutine</td>
<td>2</td>
<td>00 0000 0000 1000</td>
<td>TO, PD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SLEEP – Go into Standby mode</td>
<td>1</td>
<td>00 0000 0110 0011</td>
<td>TO, PD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SUBLW k</td>
<td>Subtract W from literal</td>
<td>1</td>
<td>11 110x kkkk kkkk kkkk</td>
<td>C, DC, Z</td>
<td></td>
</tr>
<tr>
<td>XORLW k</td>
<td>Exclusive OR literal with W</td>
<td>1</td>
<td>11 1010 kkkk kkkk kkkk</td>
<td>Z</td>
<td></td>
</tr>
</tbody>
</table>

Note 1: When an I/O register is modified as a function of itself (e.g., `MOVF GPIO, 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 the Program Counter (PC) is modified, or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a NOP.
### 13.2 Instruction Descriptions

#### ADDLW Add literal and W
- **Syntax:** `[ label ] ADDLW k
- **Operands:** $0 \leq k \leq 255$
- **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:** $0 \leq f \leq 127$
  $d \in [0,1]$
- **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:** $0 \leq k \leq 255$
- **Operation:** $(W) \cdot \text{AND.} (k) \rightarrow (W)$
- **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:** $0 \leq f \leq 127$
  $d \in [0,1]$
- **Operation:** $(W) \cdot \text{AND.} (f) \rightarrow$ (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:** $0 \leq f \leq 127$
  $0 \leq b \leq 7$
- **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:** $0 \leq f \leq 127$
  $0 \leq b \leq 7$
- **Operation:** $1 \rightarrow (f<b>)$
- **Status Affected:** None
- **Description:** Bit ‘b’ in register ‘f’ is set.

#### BTFSC Bit Test, Skip if Clear
- **Syntax:** `[ label ] BTFSC f,b
- **Operands:** $0 \leq f \leq 127$
  $0 \leq b \leq 7$
- **Operation:** skip if $(f<b>) = 0$
- **Status Affected:** None
- **Description:** If bit ‘b’ in register ‘f’ is ‘1’, the next instruction is executed. 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.
### BTFSS
**Bit Test f, Skip if Set**

**Syntax:**

\[ \text{[ label]} \ BTFSS \ f,b \]

**Operands:**

\[ 0 \leq f \leq 127 \]
\[ 0 \leq b < 7 \]

**Operation:**

skip if \((f<b) = 1\)

**Status Affected:** None

**Description:**
If bit ‘b’ in register ‘f’ is ‘0’, the next instruction is executed. If bit ‘b’ is ‘1’, then the next instruction is discarded and a **NOP** is executed instead, making this a 2-cycle instruction.

### CALL
**Call Subroutine**

**Syntax:**

\[ \text{[ label]} \ CALL \ k \]

**Operands:**

\[ 0 \leq k \leq 2047 \]

**Operation:**

\( (\text{PC}) + 1 \rightarrow \text{TOS}, \)
\( k \rightarrow \text{PC}<10:0>, \)
\( (\text{PCLATH}<4:3>) \rightarrow \text{PC}<12:11> \)

**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:**

\[ \text{[ label]} \ CLRF \ f \]

**Operands:**

\[ 0 \leq f \leq 127 \]

**Operation:**

\( 00h \rightarrow (f) \)
\( 1 \rightarrow \text{Z} \)

**Status Affected:** Z

**Description:**
The contents of register ‘f’ are cleared and the Z bit is set.

### CLRW
**Clear W**

**Syntax:**

\[ \text{[ label]} \ CLRW \]

**Operands:** None

**Operation:**

\( 00h \rightarrow (W) \)
\( 1 \rightarrow \text{Z} \)

**Status Affected:** Z

**Description:**
W register is cleared. Zero bit (Z) is set.

### CLRWDT
**Clear Watchdog Timer**

**Syntax:**

\[ \text{[ label]} \ CLRWDT \]

**Operands:** None

**Operation:**

\( 00h \rightarrow \text{WDT} \)
\( 0 \rightarrow \text{WDT prescaler} \)
\( 1 \rightarrow \text{T0} \)
\( 1 \rightarrow \text{PD} \)

**Status Affected:** \( \text{T0, PD} \)

**Description:**
**CLRWDT** instruction resets the Watchdog Timer. It also resets the prescaler of the WDT. Status bits T0 and PD are set.

### COMF
**Complement f**

**Syntax:**

\[ \text{[ label]} \ COMF \ f,d \]

**Operands:**

\[ 0 \leq f \leq 127 \]
\[ d \in [0,1] \]

**Operation:**

\( (f) \rightarrow (\text{destination}) \)

**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:**

\[ \text{[ label]} \ DECF \ f,d \]

**Operands:**

\[ 0 \leq f \leq 127 \]
\[ d \in [0,1] \]

**Operation:**

\( (f) - 1 \rightarrow (\text{destination}) \)

**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
d \in \{0,1\}
Operation: (f) - 1 \rightarrow (destination);
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’. 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.

INCFSZ
Increment f, Skip if 0

Syntax: \[ label \] INCFSZ f,d
Operands: 0 \leq f \leq 127
d \in \{0,1\}
Operation: (f) + 1 \rightarrow (destination),
skip if result = 0
Status Affected: None
Description: The contents of register ‘f’ are incremented. If ‘d’ is ‘0’, the result is placed back in register ‘f’. If ‘d’ is ‘1’, the result is placed back in register ‘f’. 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.

GOTO
Unconditional Branch

Syntax: \[ label \] GOTO k
Operands: 0 \leq k \leq 2047
Operation: k \rightarrow PC<10:0>
PCLATH<4:3> \rightarrow 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.

IORLW
Inclusive OR literal with W

Syntax: \[ label \] IORLW k
Operands: 0 \leq k \leq 255
Operation: (W) .OR. k \rightarrow (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.

INCF
Increment f

Syntax: \[ label \] INCF f,d
Operands: 0 \leq f \leq 127
d \in \{0,1\}
Operation: (f) + 1 \rightarrow (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’.

IORWF
Inclusive OR W with f

Syntax: \[ label \] IORWF f,d
Operands: 0 \leq f \leq 127
d \in \{0,1\}
Operation: (W) .OR. (f) \rightarrow (destination)
Status Affected: Z
Description: Inclusive OR the W register with register ‘f’. If ‘d’ is ‘0’, the result is placed back in register ‘f’. If ‘d’ is ‘1’, the result is placed back in register ‘f’.
### MOVF  Move f

**Syntax:** \[ \text{label} \] MOVF f,d

**Operands:** \( 0 \leq f \leq 127 \)  
\( d \in [0,1] \)

**Operation:** \((f) \rightarrow (\text{dest})\)

**Status Affected:** Z

**Description:** The contents of register f is moved to a destination dependent 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.

**Words:** 1  
**Cycles:** 1

**Example:**

```
MOVF FSR, 0

After Instruction  
W = value in FSR register  
Z = 1
```

### MOVLW Move literal to W

**Syntax:** \[ \text{label} \] MOVLW k

**Operands:** \( 0 \leq k \leq 255 \)

**Operation:** \( k \rightarrow (W) \)

**Status Affected:** None

**Description:** The eight-bit literal 'k' is loaded into W register. The “don’t cares” will assemble as ‘0’s.

**Words:** 1  
**Cycles:** 1

**Example:**

```
MOVLW 0x5A

After Instruction  
W = 0x5A
```

### MOVWF Move W to f

**Syntax:** \[ \text{label} \] MOVWF f

**Operands:** \( 0 \leq f \leq 127 \)

**Operation:** \((W) \rightarrow (f)\)

**Status Affected:** None

**Description:** Move data from W register to register 'f'.

**Words:** 1  
**Cycles:** 1

**Example:**

```
MOVWF OPTION

Before Instruction  
OPTION = 0xFF  
W = 0x4F

After Instruction  
OPTION = 0x4F  
W = 0x4F
```

### NOP No Operation

**Syntax:** \[ \text{label} \] NOP

**Operands:** None

**Operation:** No operation

**Status Affected:** None

**Description:** No operation.

**Words:** 1  
**Cycles:** 1

**Example:**

```
NOP
```
### RETFIE Return from Interrupt

**Syntax:**

```
[label]    RETFIE
```

**Operands:** None

**Operation:**

- TOS → PC,
- 1 → GIE

**Status Affected:** None

**Description:** Return from Interrupt. Stack is POPed and Top-of-Stack (TOS) is loaded in the PC. Interrupts are enabled by setting Global Interrupt Enable bit, GIE (INTCON<7>). This is a two-cycle instruction.

**Words:** 1

**Cycles:** 2

**Example:**

After Interrupt

| PC = TOS |
| GIE = 1 |

### RETLW Return with literal in W

**Syntax:**

```
[label]    RETLW   k
```

**Operands:**

- 0 ≤ k ≤ 255

**Operation:**

- k → (W);
- 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.

**Words:** 1

**Cycles:** 2

**Example:**

```
TABLE
CALL TABLE;W contains table
;offset value
          ;W now has table value
          
ADDWF PC ;W = offset
RETLW k1 ;Begin table
RETLW k2 ;
          
RETLW kn ; End of table
```

**Before Instruction**

W = 0x07

**After Instruction**

W = value of k8

### 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.
**RLF**

**Rotate Left f through Carry**

**Syntax:**

\[
\quad \text{[ } \text{label} \text{ ] RLF \ f,d}
\]

**Operands:**

\[
0 \leq f \leq 127 \\
\quad d \in [0,1]
\]

**Operation:**

See description below

**Status Affected:**

\( \text{C} \)

**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’.

**Words:**

1

**Cycles:**

1

**Example:**

\[
\quad \text{RLF REG1,0}
\]

**Before Instruction**

\[
\begin{align*}
\text{REG1} &= 1110 \ 0110 \\
\text{C} &= 0
\end{align*}
\]

**After Instruction**

\[
\begin{align*}
\text{REG1} &= 1110 \ 0110 \\
\text{W} &= 1100 \ 1100 \\
\text{C} &= 1
\end{align*}
\]

---

**RRF**

**Rotate Right f through Carry**

**Syntax:**

\[
\quad \text{[ } \text{label} \text{ ] RRF \ f,d}
\]

**Operands:**

\[
0 \leq f \leq 127 \\
\quad d \in [0,1]
\]

**Operation:**

See description below

**Status Affected:**

\( \text{C} \)

**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 stored back in register ‘f’.

---

**SLEEP**

**Enter Sleep mode**

**Syntax:**

\[
\quad \text{[ } \text{label} \text{ ] SLEEP}
\]

**Operands:**

None

**Operation:**

\[
00h \rightarrow \text{WDT}, \\
0 \rightarrow \text{WDT prescaler}, \\
1 \rightarrow \text{TO}, \\
0 \rightarrow \text{PD}
\]

**Status Affected:**

\( \text{TO}, \text{PD} \)

**Description:**

The power-down Status bit, PD is cleared. Time-out Status bit, TO is set. Watchdog Timer and its prescaler are cleared.

The processor is put into Sleep mode with the oscillator stopped.

---

**SUBLW**

**Subtract W from literal**

**Syntax:**

\[
\quad \text{[ } \text{label} \text{ ] SUBLW \ k}
\]

**Operands:**

\[
0 \leq k \leq 255
\]

**Operation:**

\[
k - (W) \rightarrow (W)
\]

**Status Affected:**

\( \text{C, DC, Z} \)

**Description:**

The W register is subtracted (2’s complement method) from the eight-bit literal ‘k’. The result is placed in the W register.

\[
\begin{array}{c|c}
\text{C} = 0 & W > k \\
\text{C} = 1 & W \leq k \\
\text{DC} = 0 & W<3:0> > k<3:0> \\
\text{DC} = 1 & W<3:0> \leq k<3:0>
\end{array}
\]
### SUBWF
**Subtract W from f**

**Syntax:**

```plaintext
[ label ] SUBWF f,d
```

**Operands:**

- `0 ≤ f ≤ 127`
- `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'.

<table>
<thead>
<tr>
<th>C = 0</th>
<th>W &gt; f</th>
</tr>
</thead>
<tbody>
<tr>
<td>C = 1</td>
<td>W ≤ f</td>
</tr>
<tr>
<td>DC = 0</td>
<td>W&lt;3:0&gt; &gt; f&lt;3:0&gt;</td>
</tr>
<tr>
<td>DC = 1</td>
<td>W&lt;3:0&gt; ≤ f&lt;3:0&gt;</td>
</tr>
</tbody>
</table>

### SWAPF
**Swap Nibbles in f**

**Syntax:**

```plaintext
[ label ] SWAPF f,d
```

**Operands:**

- `0 ≤ f ≤ 127`
- `d ∈ [0,1]`

**Operation:**

- `(f<3:0>) → (destination<7:4>)`
- `(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:**

```plaintext
[ label ] XORLW k
```

**Operands:**

- `0 ≤ k ≤ 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:**

```plaintext
[ label ] XORWF f,d
```

**Operands:**

- `0 ≤ f ≤ 127`
- `d ∈ [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'.
14.0 DEVELOPMENT SUPPORT

The PICmicro® microcontrollers are supported with a full range of hardware and software development tools:

- Integrated Development Environment
  - MPLAB® IDE Software
- Assemblers/Compilers/Linkers
  - MPASM™ Assembler
  - MPLAB C18 and MPLAB C30 C Compilers
  - MPLINK™ Object Linker/
    MPLIB™ Object Librarian
  - MPLAB ASM30 Assembler/Linker/Library
- Simulators
  - MPLAB SIM Software Simulator
- Emulators
  - MPLAB ICE 2000 In-Circuit Emulator
  - MPLAB ICE 4000 In-Circuit Emulator
- In-Circuit Debugger
  - MPLAB ICD 2
- Device Programmers
  - PICSTART® Plus Development Programmer
  - MPLAB PM3 Device Programmer
  - PICkit™ 2 Development Programmer
- Low-Cost Demonstration and Development
  Boards and Evaluation Kits

14.1 MPLAB Integrated Development Environment Software

The MPLAB IDE software brings an ease of software development previously unseen in the 8/16-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)
  - 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
- Visual device initializer for easy register initialization
- 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 HI-TECH Software C Compilers and IAR C Compilers

The MPLAB IDE allows you to:

- Edit your source files (either assembly or C)
- One touch assemble (or compile) and download to PICmicro MCU emulator and simulator tools (automatically updates all project information)
- Debug using:
  - Source files (assembly or C)
  - Mixed assembly and C
  - 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.
14.2 MPASM Assembler

The MPASM Assembler is a full-featured, universal macro assembler for all PICmicro 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

14.3 MPLAB C18 and MPLAB C30 C Compilers

The MPLAB C18 and MPLAB C30 Code Development Systems are complete ANSI C compilers for Microchip's PIC18 family of microcontrollers and the dsPIC30F, dsPIC33 and PIC24 family of digital signal controllers. These compilers provide powerful integration capabilities, superior code optimization and ease of use not found with other compilers.

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

14.4 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

14.5 MPLAB ASM30 Assembler, Linker and Librarian

MPLAB ASM30 Assembler produces relocatable machine code from symbolic assembly language for dsPIC30F devices. MPLAB C30 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 dsPIC30F instruction set
- Support for fixed-point and floating-point data
- Command line interface
- Rich directive set
- Flexible macro language
- MPLAB IDE compatibility

14.6 MPLAB SIM Software Simulator

The MPLAB SIM Software Simulator allows code development in a PC-hosted environment by simulating the PICmicro 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 C18 and MPLAB C30 C Compilers, and the MPASM and MPLAB ASM30 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.
14.7 MPLAB ICE 2000
High-Performance
In-Circuit Emulator

The MPLAB ICE 2000 In-Circuit Emulator is intended to provide the product development engineer with a complete microcontroller design tool set for PICmicro microcontrollers. Software control of the MPLAB ICE 2000 In-Circuit Emulator is advanced by the MPLAB Integrated Development Environment, which allows editing, building, downloading and source debugging from a single environment.

The MPLAB ICE 2000 is a full-featured emulator system with enhanced trace, trigger and data monitoring features. Interchangeable processor modules allow the system to be easily reconfigured for emulation of different processors. The architecture of the MPLAB ICE 2000 In-Circuit Emulator allows expansion to support new PICmicro microcontrollers.

The MPLAB ICE 2000 In-Circuit Emulator system has been designed as a real-time emulation system with advanced features that are typically found on more expensive development tools. The PC platform and Microsoft® Windows® 32-bit operating system were chosen to best make these features available in a simple, unified application.

14.8 MPLAB ICE 4000
High-Performance
In-Circuit Emulator

The MPLAB ICE 4000 In-Circuit Emulator is intended to provide the product development engineer with a complete microcontroller design tool set for high-end PICmicro MCUs and dsPIC DSCs. Software control of the MPLAB ICE 4000 In-Circuit Emulator is provided by the MPLAB Integrated Development Environment, which allows editing, building, downloading and source debugging from a single environment.

The MPLAB ICE 4000 is a premium emulator system, providing the features of MPLAB ICE 2000, but with increased emulation memory and high-speed performance for dsPIC30F and PIC18XXXX devices. Its advanced emulator features include complex triggering and timing, and up to 2 Mb of emulation memory.

The MPLAB ICE 4000 In-Circuit Emulator system has been designed as a real-time emulation system with advanced features that are typically found on more expensive development tools. The PC platform and Microsoft Windows 32-bit operating system were chosen to best make these features available in a simple, unified application.

14.9 MPLAB ICD 2 In-Circuit Debugger

Microchip’s In-Circuit Debugger, MPLAB ICD 2, is a powerful, low-cost, run-time development tool, connecting to the host PC via an RS-232 or high-speed USB interface. This tool is based on the Flash PICmicro MCUs and can be used to develop for these and other PICmicro MCUs and dsPIC DSCs. The MPLAB ICD 2 utilizes the in-circuit debugging capability built into the Flash devices. This feature, along with Microchip’s In-Circuit Serial Programming™ (ICSP™) protocol, offers cost-effective, in-circuit Flash debugging from the graphical user interface of the MPLAB Integrated Development Environment. This enables a designer to develop and debug source code by setting breakpoints, single stepping and watching variables, and CPU status and peripheral registers. Running at full speed enables testing hardware and applications in real time. MPLAB ICD 2 also serves as a development programmer for selected PICmicro devices.

14.10 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 PICmicro 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 SD/MMC card for file storage and secure data applications.
14.11 PICSTART Plus Development Programmer

The PICSTART Plus Development Programmer is an easy-to-use, low-cost, prototype programmer. It connects to the PC via a COM (RS-232) port. MPLAB Integrated Development Environment software makes using the programmer simple and efficient. The PICSTART Plus Development Programmer supports most PICmicro devices in DIP packages up to 40 pins. Larger pin count devices, such as the PIC16C92X and PIC17C76X, may be supported with an adapter socket. The PICSTART Plus Development Programmer is CE compliant.

14.12 PICkit 2 Development Programmer

The PICkit™ 2 Development Programmer is a low-cost programmer with an easy-to-use interface for programming many of Microchip's baseline, mid-range and PIC18F families of Flash memory microcontrollers. The PICkit 2 Starter Kit includes a prototyping development board, twelve sequential lessons, software and HI-TECH's PICC Lite C compiler, and is designed to help get up to speed quickly using PIC® microcontrollers. The kit provides everything needed to program, evaluate and develop applications using Microchip's powerful, mid-range Flash memory family of microcontrollers.

14.13 Demonstration, Development and Evaluation Boards

A wide variety of demonstration, development and evaluation boards for various PICmicro 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.

Check the Microchip web page (www.microchip.com) and the latest "Product Selector Guide" (DS00148) for the complete list of demonstration, development and evaluation kits.
15.0 ELECTRICAL SPECIFICATIONS

Absolute Maximum Ratings(†)

Ambient temperature under bias ........................................................................................................................................... -40° to +125°C
Storage temperature ................................................................................................................................................................. -65°C to +150°C
Voltage on VDD with respect to Vss ........................................................................................................................................  -0.3V to +6.5V
Voltage on MCLR with respect to Vss ...................................................................................................................................... -0.3V to +13.5V
Voltage on all other pins with respect to VSS ....................................................................................................................... -0.3V to (VDD + 0.3V)
Total power dissipation(†) ............................................................................................................................................................ 800 mW
Maximum current out of VSS pin ................................................................................................................................................... 95 mA
Maximum current into VDD pin .................................................................................................................................................... 95 mA
Input clamp current, IIK (VI < 0 or VI > VDD) ......................................................................................................................... ± 20 mA
Output clamp current, IOK (Vo < 0 or Vo >VDD) ....................................................................................................................... ± 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) .................................................................................................... 90 mA
Maximum current sourced PORTA and PORTC (combined) ..................................................................................................... 90 mA

Note 1: Power dissipation is calculated as follows: PDIS = VDD x (IDD – ∑IOH) + ∑((VDD – VOH) x IOH) + ∑(VIOL x IOL).

† 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 above maximum rating conditions for extended periods may affect device reliability.
FIGURE 15-1: PIC16F684 VOLTAGE-FREQUENCY GRAPH, 
-40°C ≤ TA ≤ +125°C

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

FIGURE 15-2: HFINTOSC FREQUENCY ACCURACY OVER DEVICE VDD AND TEMPERATURE
15.1 DC Characteristics: PIC16F684-I (Industrial)
PIC16F684-E (Extended)

**DC CHARACTERISTICS**

<table>
<thead>
<tr>
<th>Param No.</th>
<th>Sym</th>
<th>Characteristic</th>
<th>Min</th>
<th>Typ†</th>
<th>Max</th>
<th>Unit</th>
<th>Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>D001</td>
<td>VDD</td>
<td>Supply Voltage</td>
<td>2.0</td>
<td>—</td>
<td>5.5</td>
<td>V</td>
<td>Fosc ≤ 8 MHz: HFINTOSC, EC</td>
</tr>
<tr>
<td>D001C</td>
<td></td>
<td></td>
<td>2.0</td>
<td>—</td>
<td>5.5</td>
<td>V</td>
<td>Fosc ≤ 4 MHz</td>
</tr>
<tr>
<td>D001D</td>
<td></td>
<td></td>
<td>3.0</td>
<td>—</td>
<td>5.5</td>
<td>V</td>
<td>Fosc ≤ 10 MHz</td>
</tr>
<tr>
<td>D001</td>
<td></td>
<td></td>
<td>4.5</td>
<td>—</td>
<td>5.5</td>
<td>V</td>
<td>Fosc ≤ 20 MHz</td>
</tr>
<tr>
<td>D002*</td>
<td>VDR</td>
<td>RAM Data Retention Voltage(1)</td>
<td>1.5</td>
<td>—</td>
<td>—</td>
<td>V</td>
<td>Device in Sleep mode</td>
</tr>
<tr>
<td>D003</td>
<td>VPOR</td>
<td>Vdd Start Voltage to ensure internal Power-on Reset signal</td>
<td>—</td>
<td>Vss</td>
<td>—</td>
<td>V</td>
<td>See Section 12.3.1 “Power-On Reset (POR)” for details.</td>
</tr>
<tr>
<td>D004*</td>
<td>SVDD</td>
<td>Vdd Rise Rate to ensure internal Power-on Reset signal</td>
<td>0.05</td>
<td>—</td>
<td>—</td>
<td>V/m</td>
<td>See Section 12.3.1 “Power-On Reset (POR)” for details.</td>
</tr>
</tbody>
</table>

* 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.
### DC Characteristics: PIC16F684-I (Industrial)  
**PIC16F684-E (Extended)**

#### DC CHARACTERISTICS

**Standard Operating Conditions (unless otherwise stated)**  
Operating temperature:  
-40°C ≤ TA ≤ +85°C for industrial  
-40°C ≤ TA ≤ +125°C for extended

<table>
<thead>
<tr>
<th>Param No.</th>
<th>Device Characteristics</th>
<th>Min</th>
<th>Typ†</th>
<th>Max</th>
<th>Units</th>
<th>Conditions</th>
<th>VDD</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>D010</td>
<td>Supply Current (IDD)(1, 2)</td>
<td>—</td>
<td>11</td>
<td>16</td>
<td>μA</td>
<td>2.0 FOSC = 32 kHz</td>
<td>LP Oscillator mode</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td></td>
<td>—</td>
<td>18</td>
<td>28</td>
<td>μA</td>
<td>3.0</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>—</td>
<td>35</td>
<td>54</td>
<td>μA</td>
<td>5.0</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td>D011*</td>
<td></td>
<td>—</td>
<td>140</td>
<td>240</td>
<td>μA</td>
<td>2.0 FOSC = 1 MHz</td>
<td>XT Oscillator mode</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td></td>
<td>—</td>
<td>220</td>
<td>380</td>
<td>μA</td>
<td>3.0</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>—</td>
<td>380</td>
<td>550</td>
<td>μA</td>
<td>5.0</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td>D012</td>
<td></td>
<td>—</td>
<td>260</td>
<td>360</td>
<td>μA</td>
<td>2.0 FOSC = 4 MHz</td>
<td>XT Oscillator mode</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td></td>
<td>—</td>
<td>420</td>
<td>650</td>
<td>μA</td>
<td>3.0</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>—</td>
<td>0.8</td>
<td>1.1</td>
<td>mA</td>
<td>5.0</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td>D013*</td>
<td></td>
<td>—</td>
<td>130</td>
<td>220</td>
<td>μA</td>
<td>2.0 FOSC = 1 MHz</td>
<td>EC Oscillator mode</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td></td>
<td>—</td>
<td>215</td>
<td>360</td>
<td>μA</td>
<td>3.0</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>—</td>
<td>360</td>
<td>520</td>
<td>μA</td>
<td>5.0</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td>D014</td>
<td></td>
<td>—</td>
<td>220</td>
<td>340</td>
<td>μA</td>
<td>2.0 FOSC = 4 MHz</td>
<td>EC Oscillator mode</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td></td>
<td>—</td>
<td>375</td>
<td>550</td>
<td>μA</td>
<td>3.0</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>—</td>
<td>0.65</td>
<td>1.0</td>
<td>mA</td>
<td>5.0</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td>D015</td>
<td></td>
<td>—</td>
<td>8</td>
<td>20</td>
<td>μA</td>
<td>2.0 FOSC = 31 kHz</td>
<td>LFINTOSC mode</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td></td>
<td>—</td>
<td>16</td>
<td>40</td>
<td>μA</td>
<td>3.0</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>—</td>
<td>31</td>
<td>65</td>
<td>μA</td>
<td>5.0</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td>D016*</td>
<td></td>
<td>—</td>
<td>340</td>
<td>450</td>
<td>μA</td>
<td>2.0 FOSC = 4 MHz</td>
<td>HFINTOSC mode</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td></td>
<td>—</td>
<td>500</td>
<td>700</td>
<td>μA</td>
<td>3.0</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>—</td>
<td>31</td>
<td>65</td>
<td>μA</td>
<td>5.0</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td>D017</td>
<td></td>
<td>—</td>
<td>410</td>
<td>650</td>
<td>μA</td>
<td>2.0 FOSC = 8 MHz</td>
<td>HFINTOSC mode</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td></td>
<td>—</td>
<td>700</td>
<td>950</td>
<td>μA</td>
<td>3.0</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>—</td>
<td>1.30</td>
<td>1.65</td>
<td>mA</td>
<td>5.0</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td>D018</td>
<td></td>
<td>—</td>
<td>230</td>
<td>400</td>
<td>μA</td>
<td>2.0 FOSC = 4 MHz</td>
<td>EXTRC mode(3)</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td></td>
<td>—</td>
<td>400</td>
<td>680</td>
<td>μA</td>
<td>3.0</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>—</td>
<td>0.63</td>
<td>1.1</td>
<td>mA</td>
<td>5.0</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td>D019</td>
<td></td>
<td>—</td>
<td>2.6</td>
<td>3.25</td>
<td>mA</td>
<td>4.5 FOSC = 20 MHz</td>
<td>HS Oscillator mode</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td></td>
<td>—</td>
<td>2.8</td>
<td>3.35</td>
<td>mA</td>
<td>5.0</td>
<td>—</td>
<td></td>
</tr>
</tbody>
</table>

* 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:** 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; MCLR = VDD; WDT disabled.  
**Note 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.  
**Note 3:** For RC oscillator configurations, current through REXT is not included. The current through the resistor can be extended by the formula IR = VDD/2REXT (mA) with REXT in kΩ.
15.3 DC Characteristics: PIC16F684-I (Industrial)

<table>
<thead>
<tr>
<th>Param No.</th>
<th>Device Characteristics</th>
<th>Min</th>
<th>Typ†</th>
<th>Max</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>D020</td>
<td>Power-down Base Current(IPD)(2)</td>
<td>—</td>
<td>0.05</td>
<td>1.2</td>
<td>μA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>—</td>
<td>0.15</td>
<td>1.5</td>
<td>μA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>—</td>
<td>0.35</td>
<td>1.8</td>
<td>μA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>—</td>
<td>150</td>
<td>500</td>
<td>nA</td>
</tr>
<tr>
<td>D021</td>
<td></td>
<td>—</td>
<td>1.0</td>
<td>2.2</td>
<td>μA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>—</td>
<td>2.0</td>
<td>4.0</td>
<td>μA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>—</td>
<td>3.0</td>
<td>7.0</td>
<td>μA</td>
</tr>
<tr>
<td>D022</td>
<td></td>
<td>—</td>
<td>42</td>
<td>60</td>
<td>μA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>—</td>
<td>85</td>
<td>122</td>
<td>μA</td>
</tr>
<tr>
<td>D023</td>
<td></td>
<td>—</td>
<td>32</td>
<td>45</td>
<td>μA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>—</td>
<td>60</td>
<td>78</td>
<td>μA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>—</td>
<td>120</td>
<td>160</td>
<td>μA</td>
</tr>
<tr>
<td>D024</td>
<td></td>
<td>—</td>
<td>30</td>
<td>36</td>
<td>μA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>—</td>
<td>45</td>
<td>55</td>
<td>μA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>—</td>
<td>75</td>
<td>95</td>
<td>μA</td>
</tr>
<tr>
<td>D025*</td>
<td></td>
<td>—</td>
<td>39</td>
<td>47</td>
<td>μA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>—</td>
<td>59</td>
<td>72</td>
<td>μA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>—</td>
<td>98</td>
<td>124</td>
<td>μA</td>
</tr>
<tr>
<td>D026</td>
<td></td>
<td>—</td>
<td>4.5</td>
<td>7.0</td>
<td>μA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>—</td>
<td>5.0</td>
<td>8.0</td>
<td>μA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>—</td>
<td>6.0</td>
<td>12</td>
<td>μA</td>
</tr>
<tr>
<td>D027</td>
<td></td>
<td>—</td>
<td>0.30</td>
<td>1.6</td>
<td>μA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>—</td>
<td>0.36</td>
<td>1.9</td>
<td>μA</td>
</tr>
</tbody>
</table>

* 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:** 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.

**Note 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.
## 15.4 DC Characteristics: PIC16F684-E (Extended)

### DC CHARACTERISTICS

<table>
<thead>
<tr>
<th>Param No.</th>
<th>Device Characteristics</th>
<th>Min</th>
<th>Typ†</th>
<th>Max</th>
<th>Units</th>
<th>Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>D020E</td>
<td>Power-down Base Current (IPD)(^{(2)})</td>
<td>—</td>
<td>0.05</td>
<td>9</td>
<td>μA</td>
<td>2.0 WDT, BOR, Comparators, VREF and T1OSC disabled</td>
</tr>
<tr>
<td></td>
<td></td>
<td>—</td>
<td>0.15</td>
<td>11</td>
<td>μA</td>
<td>3.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>—</td>
<td>0.35</td>
<td>15</td>
<td>μA</td>
<td>5.0</td>
</tr>
<tr>
<td>D021E</td>
<td></td>
<td>—</td>
<td>1</td>
<td>17.5</td>
<td>μA</td>
<td>2.0 WDT Current(^{(1)})</td>
</tr>
<tr>
<td></td>
<td></td>
<td>—</td>
<td>2</td>
<td>19</td>
<td>μA</td>
<td>3.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>—</td>
<td>3</td>
<td>22</td>
<td>μA</td>
<td>5.0</td>
</tr>
<tr>
<td>D022E</td>
<td></td>
<td>—</td>
<td>42</td>
<td>65</td>
<td>μA</td>
<td>3.0 BOR Current(^{(1)})</td>
</tr>
<tr>
<td></td>
<td></td>
<td>—</td>
<td>85</td>
<td>127</td>
<td>μA</td>
<td>5.0</td>
</tr>
<tr>
<td>D023E</td>
<td></td>
<td>—</td>
<td>32</td>
<td>45</td>
<td>μA</td>
<td>2.0 Comparator Current(^{(1)}), both comparators enabled</td>
</tr>
<tr>
<td></td>
<td></td>
<td>—</td>
<td>60</td>
<td>78</td>
<td>μA</td>
<td>3.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>—</td>
<td>120</td>
<td>160</td>
<td>μA</td>
<td>5.0</td>
</tr>
<tr>
<td>D024E</td>
<td></td>
<td>—</td>
<td>30</td>
<td>70</td>
<td>μA</td>
<td>2.0 CVREF Current(^{(1)}) (high range)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>—</td>
<td>45</td>
<td>90</td>
<td>μA</td>
<td>3.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>—</td>
<td>75</td>
<td>120</td>
<td>μA</td>
<td>5.0</td>
</tr>
<tr>
<td>D025E*</td>
<td></td>
<td>—</td>
<td>39</td>
<td>91</td>
<td>μA</td>
<td>2.0 CVREF Current(^{(1)}) (low range)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>—</td>
<td>59</td>
<td>117</td>
<td>μA</td>
<td>3.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>—</td>
<td>98</td>
<td>156</td>
<td>μA</td>
<td>5.0</td>
</tr>
<tr>
<td>D026E</td>
<td></td>
<td>—</td>
<td>4.5</td>
<td>25</td>
<td>μA</td>
<td>2.0 T1OSC Current(^{(1)}), 32.768 kHz</td>
</tr>
<tr>
<td></td>
<td></td>
<td>—</td>
<td>5</td>
<td>30</td>
<td>μA</td>
<td>3.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>—</td>
<td>6</td>
<td>40</td>
<td>μA</td>
<td>5.0</td>
</tr>
<tr>
<td>D027E</td>
<td></td>
<td>—</td>
<td>0.30</td>
<td>12</td>
<td>μA</td>
<td>3.0 A/D Current(^{(1)}), no conversion in progress</td>
</tr>
<tr>
<td></td>
<td></td>
<td>—</td>
<td>0.36</td>
<td>16</td>
<td>μA</td>
<td>5.0</td>
</tr>
</tbody>
</table>

* 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:** 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.
## 15.5 DC Characteristics: PIC16F684-I (Industrial)
PIC16F684-E (Extended)

### DC CHARACTERISTICS

<table>
<thead>
<tr>
<th>Param No.</th>
<th>Sym</th>
<th>Characteristic</th>
<th>Min</th>
<th>Typ†</th>
<th>Max</th>
<th>Units</th>
<th>Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>VIL</strong></td>
<td></td>
<td><strong>Input Low Voltage</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D030</td>
<td></td>
<td>I/O Port:</td>
<td>Vss</td>
<td>—</td>
<td>0.8</td>
<td>V</td>
<td>4.5V ≤ VDD ≤ 5.5V</td>
</tr>
<tr>
<td>D030A</td>
<td></td>
<td>with TTL buffer</td>
<td>Vss</td>
<td>—</td>
<td>0.15</td>
<td>VDD</td>
<td>2.0V ≤ VDD ≤ 4.5V</td>
</tr>
<tr>
<td>D031</td>
<td></td>
<td>with Schmitt Trigger buffer</td>
<td>Vss</td>
<td>—</td>
<td>0.2</td>
<td>VDD</td>
<td>2.0V ≤ VDD ≤ 5.5V</td>
</tr>
<tr>
<td>D032</td>
<td></td>
<td>MCLR, OSC1 (RC mode)(t)</td>
<td>Vss</td>
<td>—</td>
<td>0.2</td>
<td>VDD</td>
<td>V</td>
</tr>
<tr>
<td>D033</td>
<td></td>
<td>OSC1 (XT and LP modes)</td>
<td>Vss</td>
<td>—</td>
<td>0.3</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td>D033A</td>
<td></td>
<td>OSC1 (HS mode)</td>
<td>Vss</td>
<td>—</td>
<td>0.3</td>
<td>VDD</td>
<td>V</td>
</tr>
<tr>
<td><strong>VIH</strong></td>
<td></td>
<td><strong>Input High Voltage</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D040</td>
<td></td>
<td>I/O ports:</td>
<td>2.0</td>
<td>—</td>
<td>VDD</td>
<td>V</td>
<td>4.5V ≤ VDD ≤ 5.5V</td>
</tr>
<tr>
<td>D040A</td>
<td></td>
<td>with TTL buffer</td>
<td>0.25</td>
<td>VDD +</td>
<td>0.8</td>
<td>—</td>
<td>VDD</td>
</tr>
<tr>
<td>D041</td>
<td></td>
<td>with Schmitt Trigger buffer</td>
<td>0.8</td>
<td>VDD</td>
<td>—</td>
<td>VDD</td>
<td>2.0V ≤ VDD ≤ 5.5V</td>
</tr>
<tr>
<td>D042</td>
<td></td>
<td>MCLR</td>
<td>0.8</td>
<td>VDD</td>
<td>—</td>
<td>VDD</td>
<td>V</td>
</tr>
<tr>
<td>D043</td>
<td></td>
<td>OSC1 (XT and LP modes)</td>
<td>1.6</td>
<td>—</td>
<td>—</td>
<td>VDD</td>
<td>V</td>
</tr>
<tr>
<td>D043A</td>
<td></td>
<td>OSC1 (HS mode)</td>
<td>0.7</td>
<td>VDD</td>
<td>—</td>
<td>VDD</td>
<td>V</td>
</tr>
<tr>
<td>D043B</td>
<td></td>
<td>OSC1 (RC mode)</td>
<td>0.9</td>
<td>VDD</td>
<td>—</td>
<td>VDD</td>
<td>(Note 1)</td>
</tr>
<tr>
<td><strong>IIL</strong></td>
<td></td>
<td><strong>Input Leakage Current</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D060</td>
<td></td>
<td>I/O ports:</td>
<td>—</td>
<td>± 0.1</td>
<td>1</td>
<td>μA</td>
<td>VSS ≤ VPIN ≤ VDD, Pin at high-impedance</td>
</tr>
<tr>
<td>D061</td>
<td></td>
<td>MCLR(3)</td>
<td>—</td>
<td>± 0.1</td>
<td>5</td>
<td>μA</td>
<td>VSS ≤ VPIN ≤ VDD</td>
</tr>
<tr>
<td>D063</td>
<td></td>
<td>OSC1</td>
<td>—</td>
<td>± 0.1</td>
<td>5</td>
<td>μA</td>
<td>VSS ≤ VPIN ≤ VDD, XT, HS and LP oscillator configuration</td>
</tr>
<tr>
<td>D070*</td>
<td></td>
<td>IPUR PORTA Weak Pull-up Current</td>
<td>50</td>
<td>250</td>
<td>400</td>
<td>μA</td>
<td>VDD = 5.0V, VPIN = VSS</td>
</tr>
<tr>
<td>D080</td>
<td></td>
<td>VOL Output Low Voltage(6)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>IOL = 8.5 mA, VDD = 4.5V (Ind.)</td>
</tr>
<tr>
<td>D090</td>
<td></td>
<td>VOH Output High Voltage(6)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>IOL = -3.0 mA, VDD = 4.5V (Ind.)</td>
</tr>
</tbody>
</table>

* 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.

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

**3:** The leakage current on the 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.

**4:** See Section 10.4.1 “Using the Data EEPROM” for additional information.

**5:** Including OSC2 in CLKOUT mode.
### 15.5 DC Characteristics: PIC16F684-I (Industrial) PIC16F684-E (Extended) (Continued)

<table>
<thead>
<tr>
<th>Param No.</th>
<th>Sym</th>
<th>Characteristic</th>
<th>Min</th>
<th>Typ†</th>
<th>Max</th>
<th>Units</th>
<th>Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>D100</td>
<td>IULP</td>
<td>Ultra Low-Power Wake-Up Current</td>
<td>—</td>
<td>200</td>
<td>—</td>
<td>nA</td>
<td>See Application Note AN879, “Using the Microchip Ultra Low-Power Wake-up Module” (DS00879)</td>
</tr>
<tr>
<td>D101*</td>
<td>COSC2</td>
<td>Capacitive Loading Specs on Output Pins</td>
<td>OSC2 pin</td>
<td>—</td>
<td>15</td>
<td>pF</td>
<td>In XT, HS and LP modes when external clock is used to drive OSC1</td>
</tr>
<tr>
<td>D101A*</td>
<td>CIO</td>
<td>All I/O pins</td>
<td>—</td>
<td>50</td>
<td>—</td>
<td>pF</td>
<td></td>
</tr>
<tr>
<td>D120</td>
<td>Ed</td>
<td>Byte Endurance</td>
<td>100K</td>
<td>1M</td>
<td>—</td>
<td>E/W</td>
<td>-40°C ≤ TA ≤ +85°C</td>
</tr>
<tr>
<td>D120A</td>
<td>Ed</td>
<td>Byte Endurance</td>
<td>10K</td>
<td>100K</td>
<td>—</td>
<td>E/W</td>
<td>+85°C ≤ TA ≤ +125°C</td>
</tr>
<tr>
<td>D121</td>
<td>VDRW</td>
<td>VDD for Read/Write</td>
<td>VMIN</td>
<td>—</td>
<td>5.5</td>
<td>V</td>
<td>Using EECON1 to read/write VMIN = Minimum operating voltage</td>
</tr>
<tr>
<td>D122</td>
<td>TDEW</td>
<td>Erase/Write Cycle Time</td>
<td>—</td>
<td>5</td>
<td>6</td>
<td>ms</td>
<td></td>
</tr>
<tr>
<td>D123</td>
<td>TRET</td>
<td>Characteristic Retention</td>
<td>40</td>
<td>—</td>
<td>—</td>
<td>Year</td>
<td>Provided no other specifications are violated</td>
</tr>
<tr>
<td>D124</td>
<td>TREF</td>
<td>Number of Total Erase/Write Cycles before Refresh(4)</td>
<td>1M</td>
<td>10M</td>
<td>—</td>
<td>E/W</td>
<td>-40°C ≤ TA ≤ +85°C</td>
</tr>
<tr>
<td>D130</td>
<td>EP</td>
<td>Cell Endurance</td>
<td>10K</td>
<td>100K</td>
<td>—</td>
<td>E/W</td>
<td>-40°C ≤ TA ≤ +85°C</td>
</tr>
<tr>
<td>D130A</td>
<td>Ed</td>
<td>Cell Endurance</td>
<td>1K</td>
<td>10K</td>
<td>—</td>
<td>E/W</td>
<td>+85°C ≤ TA ≤ +125°C</td>
</tr>
<tr>
<td>D131</td>
<td>VPR</td>
<td>VDD for Read</td>
<td>VMIN</td>
<td>—</td>
<td>5.5</td>
<td>V</td>
<td>VMIN = Minimum operating voltage</td>
</tr>
<tr>
<td>D132</td>
<td>VPEW</td>
<td>VDD for Erase/Write</td>
<td>4.5</td>
<td>—</td>
<td>5.5</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td>D133</td>
<td>TPEW</td>
<td>Erase/Write cycle time</td>
<td>—</td>
<td>2</td>
<td>2.5</td>
<td>ms</td>
<td></td>
</tr>
<tr>
<td>D134</td>
<td>TRET</td>
<td>Characteristic Retention</td>
<td>40</td>
<td>—</td>
<td>—</td>
<td>Year</td>
<td>Provided no other specifications are violated</td>
</tr>
</tbody>
</table>

* 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: Negative current is defined as current sourced by the pin.
3: The leakage current on the 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.
4: See Section 10.4.1 “Using the Data EEPROM” for additional information.
5: Including OSC2 in CLKOUT mode.
# 15.6 Thermal Considerations

## Standard Operating Conditions (unless otherwise stated)

**Operating temperature**  
-40°C ≤ TA ≤ +125°C

<table>
<thead>
<tr>
<th>Param No.</th>
<th>Sym</th>
<th>Characteristic</th>
<th>Typ</th>
<th>Units</th>
<th>Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>TH01</td>
<td>θJA</td>
<td>Thermal Resistance Junction to Ambient</td>
<td>69.8</td>
<td>C/W</td>
<td>14-pin PDIP package</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>85.0</td>
<td>C/W</td>
<td>14-pin SOIC package</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>100.4</td>
<td>C/W</td>
<td>14-pin TSSOP package</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>46.3</td>
<td>C/W</td>
<td>16-pin QFN 4x0.9mm package</td>
</tr>
<tr>
<td>TH02</td>
<td>θJC</td>
<td>Thermal Resistance Junction to Case</td>
<td>32.5</td>
<td>C/W</td>
<td>14-pin PDIP package</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>31.0</td>
<td>C/W</td>
<td>14-pin SOIC package</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>31.7</td>
<td>C/W</td>
<td>14-pin TSSOP package</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2.6</td>
<td>C/W</td>
<td>16-pin QFN 4x0.9mm package</td>
</tr>
<tr>
<td>TH03</td>
<td>TJ</td>
<td>Junction Temperature</td>
<td>150</td>
<td>C</td>
<td>For derated power calculations</td>
</tr>
<tr>
<td>TH04</td>
<td>PD</td>
<td>Power Dissipation</td>
<td>—</td>
<td>W</td>
<td>PD = PINTERNAL + P/I/O</td>
</tr>
<tr>
<td>TH05</td>
<td>PINTERNAL</td>
<td>Internal Power Dissipation</td>
<td>—</td>
<td>W</td>
<td>PINTERNAL = IDD x VDD</td>
</tr>
<tr>
<td>(NOTE 1)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TH06</td>
<td>P/I/O</td>
<td>I/O Power Dissipation</td>
<td>—</td>
<td>W</td>
<td>P/I/O = Σ (IOL * VOL) + Σ (IOH * (VDD - VOH))</td>
</tr>
<tr>
<td>TH07</td>
<td>PDER</td>
<td>Derated Power</td>
<td>—</td>
<td>W</td>
<td>PDER = (TJ - TA)/θJA</td>
</tr>
<tr>
<td>(NOTE 2, 3)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note 1:** IDD is current to run the chip alone without driving any load on the output pins.  
**Note 2:** TA = Ambient Temperature.  
**Note 3:** Maximum allowable power dissipation is the lower value of either the absolute maximum total power dissipation or derated power (PDER).
15.7 Timing Parameter Symbology

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

1. TppS2ppS
2. TppS

<table>
<thead>
<tr>
<th>T</th>
<th>Frequency</th>
<th>T</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Lowercase letters (pp) and their meanings:

<table>
<thead>
<tr>
<th>pp</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>cc</td>
<td>CCP1</td>
</tr>
<tr>
<td>ck</td>
<td>CLKOUT</td>
</tr>
<tr>
<td>cs</td>
<td>CS</td>
</tr>
<tr>
<td>di</td>
<td>SDI</td>
</tr>
<tr>
<td>do</td>
<td>SDO</td>
</tr>
<tr>
<td>dt</td>
<td>Data in</td>
</tr>
<tr>
<td>io</td>
<td>I/O PORT</td>
</tr>
<tr>
<td>mc</td>
<td>MCLR</td>
</tr>
<tr>
<td>rp</td>
<td>RD</td>
</tr>
<tr>
<td>rw</td>
<td>RD or WR</td>
</tr>
<tr>
<td>sc</td>
<td>SCK</td>
</tr>
<tr>
<td>ss</td>
<td>SS</td>
</tr>
<tr>
<td>t0</td>
<td>T0CKI</td>
</tr>
<tr>
<td>t1</td>
<td>T1CKI</td>
</tr>
<tr>
<td>wr</td>
<td>WR</td>
</tr>
</tbody>
</table>

Uppercase letters and their meanings:

<table>
<thead>
<tr>
<th>S</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>Fall</td>
</tr>
<tr>
<td>H</td>
<td>High</td>
</tr>
<tr>
<td>I</td>
<td>Invalid (High-impedance)</td>
</tr>
<tr>
<td>L</td>
<td>Low</td>
</tr>
<tr>
<td>P</td>
<td>Period</td>
</tr>
<tr>
<td>R</td>
<td>Rise</td>
</tr>
<tr>
<td>V</td>
<td>Valid</td>
</tr>
<tr>
<td>Z</td>
<td>High-impedance</td>
</tr>
</tbody>
</table>

FIGURE 15-3: LOAD CONDITIONS

Legend: CL = 50 pF for all pins
        15 pF for OSC2 output
15.8 AC Characteristics: PIC16F684 (Industrial, Extended)

**TABLE 15-1: CLOCK OSCILLATOR TIMING REQUIREMENTS**

<table>
<thead>
<tr>
<th>Param No.</th>
<th>Sym</th>
<th>Characteristic</th>
<th>Min</th>
<th>Typ†</th>
<th>Max</th>
<th>Units</th>
<th>Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>OS01</td>
<td>Fosc</td>
<td>External CLKin Frequency(1)</td>
<td>DC</td>
<td>—</td>
<td>37</td>
<td>kHz</td>
<td>LP Oscillator mode</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DC</td>
<td>—</td>
<td>4</td>
<td>MHz</td>
<td>XT Oscillator mode</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DC</td>
<td>—</td>
<td>20</td>
<td>MHz</td>
<td>HS Oscillator mode</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DC</td>
<td>—</td>
<td>20</td>
<td>MHz</td>
<td>EC Oscillator mode</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Oscillator Frequency(1)</td>
<td>—</td>
<td>32.768</td>
<td>—</td>
<td>kHz</td>
<td>LP Oscillator mode</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.1</td>
<td>—</td>
<td>4</td>
<td>MHz</td>
<td>XT Oscillator mode</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>—</td>
<td>20</td>
<td>MHz</td>
<td>HS Oscillator mode</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DC</td>
<td>—</td>
<td>4</td>
<td>MHz</td>
<td>RC Oscillator mode</td>
</tr>
<tr>
<td>OS02</td>
<td>Tosc</td>
<td>External CLKin Period(1)</td>
<td>27</td>
<td>—</td>
<td>•</td>
<td>μs</td>
<td>LP Oscillator mode</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>250</td>
<td>—</td>
<td>•</td>
<td>ns</td>
<td>XT Oscillator mode</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>50</td>
<td>—</td>
<td>•</td>
<td>ns</td>
<td>HS Oscillator mode</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>50</td>
<td>—</td>
<td>•</td>
<td>ns</td>
<td>EC Oscillator mode</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Oscillator Period(1)</td>
<td>—</td>
<td>30.5</td>
<td>—</td>
<td>μs</td>
<td>LP Oscillator mode</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>250</td>
<td>—</td>
<td>10,000</td>
<td>ns</td>
<td>XT Oscillator mode</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>50</td>
<td>—</td>
<td>1,000</td>
<td>ns</td>
<td>HS Oscillator mode</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>250</td>
<td>—</td>
<td>—</td>
<td>ns</td>
<td>RC Oscillator mode</td>
</tr>
<tr>
<td>OS03</td>
<td>Tcy</td>
<td>Instruction Cycle Time(1)</td>
<td>200</td>
<td>Tcy</td>
<td>DC</td>
<td>ns</td>
<td>Tcy = 4/Fosc</td>
</tr>
<tr>
<td>OS04*</td>
<td>TosH</td>
<td>External CLKin High, External CLKin Low</td>
<td>2</td>
<td>—</td>
<td>—</td>
<td>μs</td>
<td>LP oscillator</td>
</tr>
<tr>
<td></td>
<td>TosL</td>
<td></td>
<td>100</td>
<td>—</td>
<td>—</td>
<td>ns</td>
<td>XT oscillator</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>20</td>
<td>—</td>
<td>—</td>
<td>ns</td>
<td>HS oscillator</td>
</tr>
<tr>
<td>OS05*</td>
<td>TosR</td>
<td>External CLKin Rise, External CLKin Fall</td>
<td>0</td>
<td>—</td>
<td>•</td>
<td>ns</td>
<td>LP oscillator</td>
</tr>
<tr>
<td></td>
<td>TosF</td>
<td></td>
<td>0</td>
<td>—</td>
<td>•</td>
<td>ns</td>
<td>XT oscillator</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td>—</td>
<td>•</td>
<td>ns</td>
<td>HS oscillator</td>
</tr>
</tbody>
</table>

* 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 15-2: OSCILLATOR PARAMETERS

<table>
<thead>
<tr>
<th>Param No.</th>
<th>Sym</th>
<th>Characteristic</th>
<th>Freq Tolerance</th>
<th>Min</th>
<th>Typ†</th>
<th>Max</th>
<th>Units</th>
<th>Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>OS06</td>
<td>TWARM</td>
<td>Internal Oscillator Switch</td>
<td>—</td>
<td>—</td>
<td>2</td>
<td>Tosc</td>
<td>—</td>
<td>Slowest clock</td>
</tr>
<tr>
<td>OS07</td>
<td>Tsc</td>
<td>Fail-Safe Sample Clock Period(1)</td>
<td>—</td>
<td>—</td>
<td>21</td>
<td>ms</td>
<td>—</td>
<td>LFINTOSC/64</td>
</tr>
<tr>
<td>OS08</td>
<td>HFosc</td>
<td>Internal Calibrated HFINTOSC Frequency(2)</td>
<td>±1%</td>
<td>7.92</td>
<td>8.0</td>
<td>8.08</td>
<td>MHz</td>
<td>VDD = 3.5V, 25°C</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>±2%</td>
<td>7.84</td>
<td>8.0</td>
<td>8.16</td>
<td>MHz</td>
<td>2.5V ≤ VDD ≤ 5.5V, 0°C ≤ TA ≤ +85°C</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>±5%</td>
<td>7.60</td>
<td>8.0</td>
<td>8.40</td>
<td>MHz</td>
<td>2.0V ≤ VDD ≤ 5.5V, -40°C ≤ TA ≤ +85°C (Ind.), -40°C ≤ TA ≤ +125°C (Ext.)</td>
</tr>
<tr>
<td>OS09*</td>
<td>LFosc</td>
<td>Internal Uncalibrated LFINTOSC Frequency</td>
<td>—</td>
<td>15</td>
<td>31</td>
<td>45</td>
<td>kHz</td>
<td>—</td>
</tr>
<tr>
<td>OS10*</td>
<td>Tiosc ST</td>
<td>HFINTOSC Oscillator Wake-up from Sleep Start-up Time</td>
<td>—</td>
<td>5.5</td>
<td>12</td>
<td>24</td>
<td>μs</td>
<td>VDD = 2.0V, -40°C to +85°C</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>—</td>
<td>3.5</td>
<td>7</td>
<td>14</td>
<td>μs</td>
<td>VDD = 3.0V, -40°C to +85°C</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>—</td>
<td>3</td>
<td>6</td>
<td>11</td>
<td>μs</td>
<td>VDD = 5.0V, -40°C to +85°C</td>
</tr>
</tbody>
</table>

* 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:** 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 the OSC1 pin. When an external clock input is used, the ‘max’ cycle time limit is ‘DC’ (no clock) for all devices.

**Note 2:** To ensure these oscillator frequency tolerances, VDD and VSS must be capacitively decoupled as close to the device as possible. 0.1 μF and 0.01 μF values in parallel are recommended.

**Note 3:** By design.
FIGURE 15-5: CLKOUT AND I/O TIMING

TABLE 15-3: CLKOUT AND I/O TIMING PARAMETERS

<table>
<thead>
<tr>
<th>Param No.</th>
<th>Sym</th>
<th>Characteristic</th>
<th>Min</th>
<th>Typ†</th>
<th>Max</th>
<th>Units</th>
<th>Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>OS11</td>
<td>ToSH2cXL</td>
<td>Fosc↑ to CLKOUT↓ (1)</td>
<td>—</td>
<td>—</td>
<td>70</td>
<td>ns</td>
<td>VDD = 5.0V</td>
</tr>
<tr>
<td>OS12</td>
<td>ToSH2cH</td>
<td>Fosc↑ to CLKOUT↑ (1)</td>
<td>—</td>
<td>—</td>
<td>72</td>
<td>ns</td>
<td>VDD = 5.0V</td>
</tr>
<tr>
<td>OS13</td>
<td>TcxL2oV</td>
<td>CLKOUT↓ to Port out valid (1)</td>
<td>—</td>
<td>—</td>
<td>20</td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td>OS14</td>
<td>TioV2cKH</td>
<td>Port input valid before CLKOUT↑ (1) TOSC + 200 ns</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td>OS15</td>
<td>ToSH2oV</td>
<td>Fosc↑ (Q1 cycle) to Port out valid</td>
<td>—</td>
<td>50</td>
<td>70*</td>
<td>ns</td>
<td>VDD = 5.0V</td>
</tr>
<tr>
<td>OS16</td>
<td>ToSH2oI</td>
<td>Fosc↑ (Q2 cycle) to Port input invalid</td>
<td>50</td>
<td>—</td>
<td>—</td>
<td>ns</td>
<td>VDD = 5.0V</td>
</tr>
<tr>
<td>OS17</td>
<td>TioV2osH</td>
<td>Port input valid to Fosc↑ (Q2 cycle) (I/O in hold time)</td>
<td>20</td>
<td>—</td>
<td>—</td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td>OS18</td>
<td>TioR</td>
<td>Port output rise time (2)</td>
<td>—</td>
<td>15</td>
<td>72</td>
<td>ns</td>
<td>VDD = 2.0V</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>40</td>
<td>32</td>
<td></td>
<td>VDD = 5.0V</td>
</tr>
<tr>
<td>OS19</td>
<td>TioF</td>
<td>Port output fall time (2)</td>
<td>—</td>
<td>28</td>
<td>55</td>
<td>ns</td>
<td>VDD = 2.0V</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>15</td>
<td>30</td>
<td></td>
<td>VDD = 5.0V</td>
</tr>
<tr>
<td>OS20*</td>
<td>Tinp</td>
<td>INT pin input high or low time</td>
<td>25</td>
<td>—</td>
<td>—</td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td>OS21*</td>
<td>Trap</td>
<td>PORTA interrupt-on-change new input level time</td>
<td>Tcy</td>
<td>—</td>
<td>—</td>
<td>ns</td>
<td></td>
</tr>
</tbody>
</table>

* 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 4 x Tosc.
2: Includes OSC2 in CLKOUT mode.
FIGURE 15-6:  
RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING

![Diagram of reset, watchdog timer, oscillator start-up timer, and power-up timer timing]

**Note 1:** Asserted low.

FIGURE 15-7:  
BROWN-OUT RESET TIMING AND CHARACTERISTICS

![Diagram of brown-out reset timing]

* 64 ms delay only if PWRT bit in the Configuration Word register is programmed to ‘0’.
**TABLE 15-4: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER, POWER-UP TIMER AND BROWN-OUT RESET PARAMETERS**

<table>
<thead>
<tr>
<th>Param No.</th>
<th>Sym</th>
<th>Characteristic</th>
<th>Min</th>
<th>Typ†</th>
<th>Max</th>
<th>Units</th>
<th>Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>Tmcl</td>
<td>MCLR Pulse Width (low)</td>
<td>2</td>
<td>—</td>
<td>—</td>
<td>μs</td>
<td>VDD = 5V, -40°C to +85°C</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5</td>
<td>—</td>
<td>—</td>
<td>μs</td>
<td>VDD = 5V</td>
</tr>
<tr>
<td>31</td>
<td>Twdt</td>
<td>Watchdog Timer Time-out Period (No Prescaler)</td>
<td>10</td>
<td>16</td>
<td>29</td>
<td>ms</td>
<td>VDD = 5V, -40°C to +85°C</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>10</td>
<td>16</td>
<td>31</td>
<td>ms</td>
<td>VDD = 5V</td>
</tr>
<tr>
<td>32</td>
<td>Tost</td>
<td>Oscillation Start-up Timer Period(1, 2)</td>
<td>—</td>
<td>1024</td>
<td>—</td>
<td>Tosc</td>
<td>(NOTE 3)</td>
</tr>
<tr>
<td>33*</td>
<td>Tpwrt</td>
<td>Power-up Timer Period</td>
<td>40</td>
<td>65</td>
<td>140</td>
<td>ms</td>
<td></td>
</tr>
<tr>
<td>34*</td>
<td>Tioz</td>
<td>I/O High-impedance from MCLR Low or Watchdog Timer Reset</td>
<td>—</td>
<td>—</td>
<td>2.0</td>
<td>μs</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>Vbor</td>
<td>Brown-out Reset Voltage</td>
<td>2.0</td>
<td>—</td>
<td>2.2</td>
<td>V</td>
<td>(NOTE 4)</td>
</tr>
<tr>
<td>36*</td>
<td>Vhyst</td>
<td>Brown-out Reset Hysteresis</td>
<td>—</td>
<td>50</td>
<td>—</td>
<td>mV</td>
<td></td>
</tr>
<tr>
<td>37*</td>
<td>Tbor</td>
<td>Brown-out Reset Minimum Detection Period</td>
<td>100</td>
<td>—</td>
<td>—</td>
<td>μs</td>
<td>VDD ≤ VBOR</td>
</tr>
</tbody>
</table>

* 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 input applied to the OSC1 pin. When an external clock input is used, the ‘max’ cycle time limit is ‘DC’ (no clock) for all devices.

2: By design.
3: Period of the slower clock.
4: To ensure these voltage tolerances, VDD and VSS must be capacitively decoupled as close to the device as possible. 0.1 μF and 0.01 μF values in parallel are recommended.
FIGURE 15-8: TIMER0 AND TIMER1 EXTERNAL CLOCK TIMINGS

TABLE 15-5: TIMER0 AND TIMER1 EXTERNAL CLOCK REQUIREMENTS

<table>
<thead>
<tr>
<th>Param No.</th>
<th>Sym</th>
<th>Characteristic</th>
<th>Min</th>
<th>Typ†</th>
<th>Max</th>
<th>Units</th>
<th>Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>40</td>
<td>T0H</td>
<td>T0CKI High Pulse Width</td>
<td>No Prescaler</td>
<td>0.5 TCY + 20</td>
<td>—</td>
<td>—</td>
<td>ns</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>With Prescaler</td>
<td>10</td>
<td>—</td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>T0L</td>
<td>T0CKI Low Pulse Width</td>
<td>No Prescaler</td>
<td>0.5 TCY + 20</td>
<td>—</td>
<td>—</td>
<td>ns</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>With Prescaler</td>
<td>10</td>
<td>—</td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>T0P</td>
<td>T0CKI Period</td>
<td>Greater of:</td>
<td>20 or TCY + 40</td>
<td>—</td>
<td>—</td>
<td>ns</td>
</tr>
<tr>
<td>45</td>
<td>T1H</td>
<td>T1CKI High Time</td>
<td>Synchronous, No Prescaler</td>
<td>0.5 TCY + 20</td>
<td>—</td>
<td>—</td>
<td>ns</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Synchronous, with Prescaler</td>
<td>15</td>
<td>—</td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asynchronous</td>
<td>30</td>
<td>—</td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td>46</td>
<td>T1L</td>
<td>T1CKI Low Time</td>
<td>Synchronous, No Prescaler</td>
<td>0.5 TCY + 20</td>
<td>—</td>
<td>—</td>
<td>ns</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Synchronous, with Prescaler</td>
<td>15</td>
<td>—</td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asynchronous</td>
<td>30</td>
<td>—</td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td>47</td>
<td>T1P</td>
<td>T1CKI Input Period</td>
<td>Synchronous</td>
<td>Greater of:</td>
<td>30 or TCY + 40</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asynchronous</td>
<td>60</td>
<td>—</td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td>48</td>
<td>T1</td>
<td>Timer1 Oscillator Input Frequency Range</td>
<td>—</td>
<td>32.768</td>
<td>—</td>
<td>kHz</td>
<td>(oscillator enabled by setting bit T1OSCEN)</td>
</tr>
<tr>
<td>49</td>
<td>TCKE</td>
<td>Delay from External Clock Edge to Timer Increment</td>
<td>2 Tosc</td>
<td>—</td>
<td>7 Tosc</td>
<td>—</td>
<td>Timers in Sync mode</td>
</tr>
</tbody>
</table>
FIGURE 15-9: CAPTURE/COMPARE/PWM TIMINGS (ECCP)

Note: Refer to Figure 15-3 for load conditions.

TABLE 15-6: CAPTURE/COMPARE/PWM REQUIREMENTS (ECCP)

Standard Operating Conditions (unless otherwise stated)
Operating Temperature -40°C ≤ TA ≤ +125°C

<table>
<thead>
<tr>
<th>Param No.</th>
<th>Sym</th>
<th>Characteristic</th>
<th>Min</th>
<th>Typ†</th>
<th>Max</th>
<th>Units</th>
<th>Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>CC01*</td>
<td>TccL</td>
<td>CCP1 Input Low Time</td>
<td>No Prescaler</td>
<td>0.5Tcy + 20</td>
<td>—</td>
<td>—</td>
<td>ns</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>With Prescaler</td>
<td>20</td>
<td>—</td>
<td>—</td>
<td>ns</td>
</tr>
<tr>
<td>CC02*</td>
<td>TccH</td>
<td>CCP1 Input High Time</td>
<td>No Prescaler</td>
<td>0.5Tcy + 20</td>
<td>—</td>
<td>—</td>
<td>ns</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>With Prescaler</td>
<td>20</td>
<td>—</td>
<td>—</td>
<td>ns</td>
</tr>
<tr>
<td>CC03*</td>
<td>TccP</td>
<td>CCP1 Input Period</td>
<td></td>
<td>3Tcy + 40/N</td>
<td>—</td>
<td>—</td>
<td>ns</td>
</tr>
</tbody>
</table>

* 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 15-7: COMPARATOR SPECIFICATIONS

<table>
<thead>
<tr>
<th>Param No.</th>
<th>Sym</th>
<th>Characteristics</th>
<th>Min</th>
<th>Typ†</th>
<th>Max</th>
<th>Units</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>CM01</td>
<td>VOS</td>
<td>Input Offset Voltage</td>
<td>—</td>
<td>± 5.0</td>
<td>± 10</td>
<td>mV</td>
<td>(VDD - 1.5)/2</td>
</tr>
<tr>
<td>CM02</td>
<td>VCM</td>
<td>Input Common Mode Voltage</td>
<td>0</td>
<td>—</td>
<td>VDD - 1.5</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td>CM03*</td>
<td>CMRR</td>
<td>Common Mode Rejection Ratio</td>
<td>+55</td>
<td>—</td>
<td>—</td>
<td>dB</td>
<td></td>
</tr>
<tr>
<td>CM04*</td>
<td>TRT</td>
<td>Response Time</td>
<td>Falling</td>
<td>150</td>
<td>600</td>
<td>ns</td>
<td>(NOTE 1)</td>
</tr>
<tr>
<td>CM05*</td>
<td>TmC2COV</td>
<td>Comparator Mode Change to Output Valid</td>
<td>—</td>
<td>—</td>
<td>10</td>
<td>μs</td>
<td></td>
</tr>
</tbody>
</table>

* 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:** Response time is measured with one comparator input at (VDD - 1.5)/2 - 100 mV to (VDD - 1.5)/2 + 20 mV.

### TABLE 15-8: COMPARATOR VOLTAGE REFERENCE (CVREF) SPECIFICATIONS

<table>
<thead>
<tr>
<th>Param No.</th>
<th>Sym</th>
<th>Characteristics</th>
<th>Min</th>
<th>Typ†</th>
<th>Max</th>
<th>Units</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>CV01*</td>
<td>CLSB</td>
<td>Step Size[^2]</td>
<td>—</td>
<td>VDD/24</td>
<td>—</td>
<td>V</td>
<td>Low Range (VRR = 1)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>VDD/32</td>
<td>—</td>
<td>V</td>
<td>High Range (VRR = 0)</td>
</tr>
<tr>
<td>CV02*</td>
<td>CACC</td>
<td>Absolute Accuracy</td>
<td>—</td>
<td>—</td>
<td>± 1/2</td>
<td>LSb</td>
<td>Low Range (VRR = 1)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>—</td>
<td>± 1/2</td>
<td>LSb</td>
<td>High Range (VRR = 0)</td>
</tr>
<tr>
<td>CV03*</td>
<td>CR</td>
<td>Unit Resistor Value (R)</td>
<td>—</td>
<td>2k</td>
<td>—</td>
<td>Ω</td>
<td></td>
</tr>
</tbody>
</table>

* 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:** Settling time measured while VRR = 1 and VR<3:0> transitions from ‘0000’ to ‘1111’.

2: See Section 8.10 “Comparator Voltage Reference” for more information.
**TABLE 15-9: PIC16F684 A/D CONVERTER (ADC) CHARACTERISTICS**

Standard Operating Conditions (unless otherwise stated)
Operating temperature: 
-40°C ≤ TA ≤ +125°C

<table>
<thead>
<tr>
<th>Param No.</th>
<th>Sym</th>
<th>Characteristic</th>
<th>Min</th>
<th>Typ†</th>
<th>Max</th>
<th>Units</th>
<th>Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>AD01</td>
<td>NR</td>
<td>Resolution</td>
<td>—</td>
<td>—</td>
<td>10 bits bit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AD02</td>
<td>EIL</td>
<td>Integral Error</td>
<td>—</td>
<td>—</td>
<td>±1 LSb</td>
<td>VREF = 5.12V</td>
<td>No missing codes to 10 bits</td>
</tr>
<tr>
<td>AD03</td>
<td>EDL</td>
<td>Differential Error</td>
<td>—</td>
<td>—</td>
<td>±1 LSb</td>
<td>VREF = 5.12V</td>
<td></td>
</tr>
<tr>
<td>AD04</td>
<td>EOFF</td>
<td>Offset Error</td>
<td>—</td>
<td>—</td>
<td>±1 LSb</td>
<td>VREF = 5.12V</td>
<td></td>
</tr>
<tr>
<td>AD07</td>
<td>EGN</td>
<td>Gain Error</td>
<td>—</td>
<td>—</td>
<td>±1 LSb</td>
<td>VREF = 5.12V</td>
<td></td>
</tr>
<tr>
<td>AD06</td>
<td>VREF</td>
<td>Reference Voltage&lt;sup&gt;(3)&lt;/sup&gt;</td>
<td>2.2</td>
<td>—</td>
<td>—</td>
<td>VDD</td>
<td>Absolute minimum to ensure 1 LSb accuracy</td>
</tr>
<tr>
<td>AD06A</td>
<td></td>
<td></td>
<td>2.7</td>
<td>—</td>
<td>—</td>
<td>VDD</td>
<td></td>
</tr>
<tr>
<td>AD07</td>
<td>VAIN</td>
<td>Full-Scale Range</td>
<td>VSS</td>
<td>—</td>
<td>VREF</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td>AD08</td>
<td>ZAIN</td>
<td>Recommended Impedance of Analog Voltage Source</td>
<td>—</td>
<td>—</td>
<td>10 kΩ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AD09&lt;sup&gt;*&lt;/sup&gt;</td>
<td>IREF</td>
<td>VREF Input Current&lt;sup&gt;(3)&lt;/sup&gt;</td>
<td>10</td>
<td>—</td>
<td>1000</td>
<td>μA</td>
<td>During VAIN acquisition. Based on differential of VHOLD to VAIN.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>—</td>
<td>—</td>
<td>50</td>
<td>μA</td>
<td>During A/D conversion cycle.</td>
</tr>
</tbody>
</table>

* 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. Total Absolute Error includes integral, differential, offset and gain errors.
2. The A/D conversion result never decreases with an increase in the input voltage and has no missing codes.
3. ADC VREF is from external VREF or VDD pin, whichever is selected as reference input.
4. When ADC is off, it will not consume any current other than leakage current. The power-down current specification includes any such leakage from the ADC module.
# TABLE 15-10: PIC16F684 A/D CONVERSION REQUIREMENTS

<table>
<thead>
<tr>
<th>Param No.</th>
<th>Sym</th>
<th>Characteristic</th>
<th>Min</th>
<th>Typ†</th>
<th>Max</th>
<th>Units</th>
<th>Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>AD130*</td>
<td>TAD</td>
<td>A/D Clock Period</td>
<td>1.6</td>
<td>—</td>
<td>9.0</td>
<td>μs</td>
<td>TOSC-based, VREF ≥ 3.0V</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A/D Internal RC Oscillator Period</td>
<td>3.0</td>
<td>6.0</td>
<td>9.0</td>
<td>μs</td>
<td>TOSC-based, VREF full range</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1.6</td>
<td>4.0</td>
<td>6.0</td>
<td>μs</td>
<td>ADCS&lt;1:0&gt; = 11 (ADRC mode)</td>
</tr>
<tr>
<td>AD131</td>
<td>TCVN</td>
<td>Conversion Time (not including Acquisition Time)</td>
<td>—</td>
<td>11</td>
<td>—</td>
<td>TAD</td>
<td>Set GO/DONE bit to new data in A/D Result register</td>
</tr>
<tr>
<td>AD132*</td>
<td>TACQ</td>
<td>Acquisition Time</td>
<td>—</td>
<td>11.5</td>
<td>—</td>
<td>μs</td>
<td></td>
</tr>
<tr>
<td>AD133*</td>
<td>TAMP</td>
<td>Amplifier Settling Time</td>
<td>—</td>
<td>—</td>
<td>5</td>
<td>μs</td>
<td></td>
</tr>
<tr>
<td>AD134</td>
<td>TGQ</td>
<td>Q4 to A/D Clock Start</td>
<td>—</td>
<td>TOSC/2</td>
<td>—</td>
<td>—</td>
<td>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.</td>
</tr>
</tbody>
</table>

* 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:** ADRESH and ADRESL registers may be read on the following TCY cycle.

**Note 2:** See Section 9.3 “A/D Acquisition Requirements” for minimum conditions.
FIGURE 15-10: PIC16F684 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.

FIGURE 15-11: PIC16F684 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.
16.0 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.

**Note:** The graphs and tables provided following this note are a statistical summary based on a limited number of samples and are provided for informational purposes only. The performance characteristics listed herein are not tested or guaranteed. In some graphs or tables, the data presented may be outside the specified operating range (e.g., outside specified power supply range) and therefore, outside the warranted range.

“Typical” represents the mean of the distribution at 25°C. “Maximum” or “minimum” represents (mean + 3σ) or (mean - 3σ) respectively, where σ is a standard deviation, over each temperature range.

**FIGURE 16-1:** TYPICAL IDD VS. FOSC OVER VDD (EC MODE)

![Typical: Statistical Mean @25°C
Maximum: Mean (Worst Case Temp) + 3σ
(-40°C to 125°C)](image)
FIGURE 16-2: MAXIMUM IDD vs. FOSC OVER VDD (EC MODE)

Typical: Statistical Mean @25°C
Maximum: Mean (Worst Case Temp) + 3σ
(-40°C to 125°C)

FIGURE 16-3: TYPICAL IDD vs. FOSC OVER VDD (HS MODE)

Typical: Statistical Mean @25°C
Maximum: Mean (Worst Case Temp) + 3σ
(-40°C to 125°C)
FIGURE 16-4: MAXIMUM IDD vs. FOSC OVER VDD (HS MODE)

Typical: Statistical Mean @25°C
Maximum: Mean (Worst Case Temp) + 3σ
(-40°C to 125°C)

FIGURE 16-5: TYPICAL IDD vs. VDD OVER FOSC (XT MODE)

Typical: Statistical Mean @25°C
Maximum: Mean (Worst Case Temp) + 3σ
(-40°C to 125°C)
FIGURE 16-6: MAXIMUM IDD vs. VDD OVER FOSC (XT MODE)

- Typical: Statistical Mean @25°C
- Maximum: Mean (Worst Case Temp) + 3σ
  (-40°C to 125°C)

FIGURE 16-7: TYPICAL IDD vs. VDD OVER FOSC (EXTRC MODE)

- Typical: Statistical Mean @25°C
- Maximum: Mean (Worst Case Temp) + 3σ
  (-40°C to 125°C)
FIGURE 16-8: MAXIMUM IDD vs. VDD OVER FOSC (EXTRC MODE)

Typical: Statistical Mean @25°C
Maximum: Mean (Worst Case Temp) + 3σ
(-40°C to +125°C)

FIGURE 16-9: IDD vs. VDD OVER FOSC (LFINTOSC MODE, 31 kHz)

Typical: Statistical Mean @25°C
Maximum: Mean (Worst Case Temp) + 3σ
(-40°C to 125°C)
FIGURE 16-10: IDD vs. VDD OVER FOSC (LP MODE)

Typical: Statistical Mean @25°C
Maximum: Mean (Worst Case Temp) + 3σ
(-40°C to 125°C)

32 kHz Maximum
32 kHz Typical

VDD (V) vs. IDD (μA)

VDD (V) vs. IDD (μA)

FIGURE 16-11: TYPICAL IDD vs. FOSC OVER VDD (HFINTOSC MODE)

Typical: Statistical Mean @25°C
Maximum: Mean (Worst Case Temp) + 3σ
(-40°C to 125°C)

FOSC vs. IDD (μA)

FOSC vs. IDD (μA)
FIGURE 16-12: MAXIMUM IDD vs. FOSC OVER VDD (HFINTOSC MODE)

FIGURE 16-13: TYPICAL IPD vs. VDD (SLEEP MODE, ALL PERIPHERALS DISABLED)
FIGURE 16-14: MAXIMUM IPD vs. VDD (SLEEP MODE, ALL PERIPHERALS DISABLED)

FIGURE 16-15: COMPARATOR IPD vs. VDD (BOTH COMPARATORS ENABLED)
FIGURE 16-16: BOR IPD vs. VDD OVER TEMPERATURE

Typical: Statistical Mean @25°C
Maximum: Mean (Worst Case Temp) + 3σ
(-40°C to 125°C)

FIGURE 16-17: TYPICAL WDT IPD vs. VDD OVER TEMPERATURE

Typical: Statistical Mean @25°C
Maximum: Mean (Worst Case Temp) + 3σ
(-40°C to 125°C)
FIGURE 16-18: MAXIMUM WDT IPD vs. VDD OVER TEMPERATURE

<table>
<thead>
<tr>
<th>VDD (V)</th>
<th>Max. 125°C</th>
<th>Max. 85°C</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>2.5</td>
<td>5.0</td>
<td>5.0</td>
</tr>
<tr>
<td>3.0</td>
<td>10.0</td>
<td>10.0</td>
</tr>
<tr>
<td>3.5</td>
<td>15.0</td>
<td>15.0</td>
</tr>
<tr>
<td>4.0</td>
<td>20.0</td>
<td>20.0</td>
</tr>
<tr>
<td>4.5</td>
<td>25.0</td>
<td>25.0</td>
</tr>
</tbody>
</table>

Max. (125°C): Typical: Statistical Mean @25°C
Maximum: Mean (Worst Case Temp) + 3σ (-40°C to 125°C)

FIGURE 16-19: WDT PERIOD vs. VDD OVER TEMPERATURE

<table>
<thead>
<tr>
<th>VDD (V)</th>
<th>Minimum</th>
<th>Typical</th>
<th>Max. (125°C)</th>
<th>Max. (85°C)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.0</td>
<td>10</td>
<td>12</td>
<td>14</td>
<td>16</td>
</tr>
<tr>
<td>2.5</td>
<td>18</td>
<td>20</td>
<td>22</td>
<td>24</td>
</tr>
<tr>
<td>3.0</td>
<td>26</td>
<td>28</td>
<td>30</td>
<td>32</td>
</tr>
</tbody>
</table>

Typical: Statistical Mean @25°C
Maximum: Mean (Worst Case Temp) + 3σ (-40°C to 125°C)
FIGURE 16-20: WDT PERIOD vs. TEMPERATURE OVER VDD (5.0V)

-40°C  25°C  85°C  125°C

Vdd = 5V

10  20  30

Time (ms)

- Maximum
- Typical
- Minimum

Temperature (°C)

Typical: Statistical Mean @25°C
Maximum: Mean (Worst Case Temp) + 3σ
(-40°C to 125°C)

FIGURE 16-21: CVREF IPD vs. VDD OVER TEMPERATURE (HIGH RANGE)

0  20  40  60  80  100  120  140

Typical: Statistical Mean @25°C
Maximum: Mean (Worst Case Temp) + 3σ
(-40°C to 125°C)

0  2.0  2.5  3.0  3.5  4.0  4.5  5.0  5.5

Vdd (V)

Max. 125°C
Max. 85°C
Typical
FIGURE 16-22: CVREF IPD vs. VDD OVER TEMPERATURE (LOW RANGE)

Typical: Statistical Mean @25°C
Maximum: Mean (Worst Case Temp) + 3σ
(-40°C to 125°C)

Typical
Max. 125°C
Max. 85°C

Vdd (V)

0
20
40
60
80
100
120
140
160
180

IPD (μA)

2.0 2.5 3.0 3.5 4.0 4.5 5.0 5.5

Vdd (V)

2.0 2.5 3.0 3.5 4.0 4.5 5.0 5.5

FIGURE 16-23: VOL vs. IOL OVER TEMPERATURE (VDD = 3.0V)

Typical: Statistical Mean @25°C
Maximum: Mean (Worst Case Temp) + 3σ
(-40°C to 125°C)

Typical 25°C
Max. 125°C
Max. 85°C
Min. -40°C

VOL (V)

0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8

IOL (mA)

5.0 5.5 6.0 6.5 7.0 7.5 8.0 8.5 9.0 9.5 10.0

VOL (V)

0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8

IOL (mA)

5.0 5.5 6.0 6.5 7.0 7.5 8.0 8.5 9.0 9.5 10.0

Typical: Statistical Mean @25°C
Maximum: Mean (Worst Case Temp) + 3σ
(-40°C to 125°C)
FIGURE 16-24: $V_{OL}$ vs. $I_{OL}$ OVER TEMPERATURE ($V_{DD} = 5.0V$)

Typical: Statistical Mean @25°C
Maximum: Mean (Worst Case Temp) + 3σ
(-40°C to 125°C)

FIGURE 16-25: $V_{OH}$ vs. $I_{OH}$ OVER TEMPERATURE ($V_{DD} = 3.0V$)

Typical: Statistical Mean @25°C
Maximum: Mean (Worst Case Temp) + 3σ
(-40°C to 125°C)
FIGURE 16-26:  \( V_{OH} \) vs. \( I_{OH} \) OVER TEMPERATURE (\( V_{DD} = 5.0V \))

![Graph showing \( V_{OH} vs. I_{OH} \) over temperature.]

Typical: Statistical Mean @25°C
Maximum: Mean (Worst Case Temp) + 3\( \sigma \)
(-40°C to 125°C)

FIGURE 16-27: TTL INPUT THRESHOLD \( V_{IN} \) vs. \( V_{DD} \) OVER TEMPERATURE

![Graph showing \( V_{IN} vs. V_{DD} \) over temperature.]

Typical: Statistical Mean @25°C
Maximum: Mean (Worst Case Temp) + 3\( \sigma \)
(-40°C to 125°C)
FIGURE 16-28: SCHMITT TRIGGER INPUT THRESHOLD $V_{IN}$ vs. $V_{DD}$ OVER TEMPERATURE

- $V_{IH}$ Max. 125°C
- $V_{IH}$ Min. -40°C
- $V_{IL}$ Min. 125°C
- $V_{IL}$ Max. -40°C

Typical: Statistical Mean @25°C
Maximum: Mean (Worst Case Temp) + 3σ
(-40°C to 125°C)

FIGURE 16-29: $T1_{OSC}$ $I_{PD}$ vs. $V_{DD}$ OVER TEMPERATURE (32 kHz)

- Maximum: Mean + 3σ (-40°C to 125°C)
- Typical: Statistical Mean @25°C
- Maximum: Mean (Worst Case Temp) + 3σ
(-40°C to 125°C)
FIGURE 16-30: COMPARATOR RESPONSE TIME (RISING EDGE)

Note:
- \( V_{CM} = \frac{V_{DD} - 1.5V}{2} \)
- \( V^+ \) input = \( V_{CM} \)
- \( V^- \) input = Transition from \( V_{CM} + 100\text{mV} \) to \( V_{CM} - 20\text{mV} \)

FIGURE 16-31: COMPARATOR RESPONSE TIME (FALLING EDGE)

Note:
- \( V_{CM} = \frac{V_{DD} - 1.5V}{2} \)
- \( V^+ \) input = \( V_{CM} \)
- \( V^- \) input = Transition from \( V_{CM} - 100\text{mV} \) to \( V_{CM} + 20\text{mV} \)
FIGURE 16-32: LFINTOSC FREQUENCY vs. VDD OVER TEMPERATURE (31 kHz)

![Graph showing LFINTOSC frequency vs. VDD over temperature.]

Typical: Statistical Mean @25°C
Maximum: Mean (Worst Case Temp) + 3σ
(-40°C to 125°C)

FIGURE 16-33: ADC CLOCK PERIOD vs. VDD OVER TEMPERATURE

![Graph showing ADC clock period vs. VDD over temperature.]

Typical: Statistical Mean @25°C
Maximum: Mean (Worst Case Temp) + 3σ
(-40°C to 125°C)
FIGURE 16-34: TYPICAL HFINTOSC START-UP TIMES vs. VDD OVER TEMPERATURE

FIGURE 16-35: MAXIMUM HFINTOSC START-UP TIMES vs. VDD OVER TEMPERATURE

Typical: Statistical Mean @25°C
Maximum: Mean (Worst Case Temp) + 3σ
(-40°C to 125°C)
FIGURE 16-36: MINIMUM HFINTOSC START-UP TIMES vs. VDD OVER TEMPERATURE

Typical: Statistical Mean @25°C
Maximum: Mean (Worst Case Temp) + 3σ
(-40°C to 125°C)

FIGURE 16-37: TYPICAL HFINTOSC FREQUENCY CHANGE vs. VDD (25°C)
FIGURE 16-38: TYPICAL HFINTOSC FREQUENCY CHANGE OVER DEVICE VDD (85°C)

FIGURE 16-39: TYPICAL HFINTOSC FREQUENCY CHANGE vs. VDD (125°C)
FIGURE 16-40: TYPICAL HFINTOSC FREQUENCY CHANGE vs. VDD (-40°C)
17.0 PACKAGING INFORMATION

17.1 Package Marking Information

<table>
<thead>
<tr>
<th>Package Type</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>14-Lead PDIP</td>
<td><img src="pic16f684-1/p.png" alt="Example" /></td>
</tr>
<tr>
<td>14-Lead SOIC (.150&quot;)</td>
<td><img src="pic16f684-e.png" alt="Example" /></td>
</tr>
<tr>
<td>14-Lead TSSOP</td>
<td><img src="pic16f684-tssop.png" alt="Example" /></td>
</tr>
<tr>
<td>16-Lead QFN</td>
<td><img src="pic16f684-qfn.png" alt="Example" /></td>
</tr>
</tbody>
</table>

**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.

* Standard PICmicro device marking consists of Microchip part number, year code, week code, and traceability code. For PICmicro device marking beyond this, certain price adders apply. Please check with your Microchip Sales Office. For QTP devices, any special marking adders are included in QTP price.
17.2 Package Details

The following sections give the technical details of the packages.

14-Lead Plastic Dual In-line (P) – 300 mil (PDIP)

<table>
<thead>
<tr>
<th>Units</th>
<th>INCHES*</th>
<th>MILLIMETERS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Pins</td>
<td>n</td>
<td>14</td>
</tr>
<tr>
<td>Pitch</td>
<td>p</td>
<td>.100</td>
</tr>
<tr>
<td>Top to Seating Plane A</td>
<td>.140</td>
<td>3.56</td>
</tr>
<tr>
<td>Molded Package Thickness A2</td>
<td>.115</td>
<td>2.92</td>
</tr>
<tr>
<td>Base to Seating Plane A1</td>
<td>.015</td>
<td>0.38</td>
</tr>
<tr>
<td>Shoulder to Shoulder Width</td>
<td>.300</td>
<td>7.62</td>
</tr>
<tr>
<td>Molded Package Width E1</td>
<td>.240</td>
<td>6.10</td>
</tr>
<tr>
<td>Overall Length D</td>
<td>.740</td>
<td>18.80</td>
</tr>
<tr>
<td>Tip to Seating Plane L</td>
<td>.125</td>
<td>3.18</td>
</tr>
<tr>
<td>Lead Thickness c</td>
<td>.008</td>
<td>0.20</td>
</tr>
<tr>
<td>Upper Lead Width B1</td>
<td>.045</td>
<td>1.14</td>
</tr>
<tr>
<td>Lower Lead Width B</td>
<td>.014</td>
<td>0.36</td>
</tr>
<tr>
<td>Overall Row Spacing eB</td>
<td>.310</td>
<td>7.87</td>
</tr>
<tr>
<td>Mold Draft Angle Top α</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>Mold Draft Angle Bottom β</td>
<td>5</td>
<td>10</td>
</tr>
</tbody>
</table>

* Controlling Parameter
§ Significant Characteristic

Notes:
Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010” (0.254mm) per side.
JEDEC Equivalent: MS-001
Drawing No. C04-005
14-Lead Plastic Small Outline (SL) – Narrow, 150 mil (SOIC)

### Dimension Limits

<table>
<thead>
<tr>
<th>Units</th>
<th>INCHES*</th>
<th>MILLIMETERS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MIN</td>
<td>NOM</td>
</tr>
<tr>
<td>Number of Pins</td>
<td>n</td>
<td>14</td>
</tr>
<tr>
<td>Pitch</td>
<td>P</td>
<td>.050</td>
</tr>
<tr>
<td>Overall Height</td>
<td>A</td>
<td>.053 .061</td>
</tr>
<tr>
<td>Molded Package Thickness</td>
<td>A2</td>
<td>.052 .056</td>
</tr>
<tr>
<td>Standoff</td>
<td>§ A1</td>
<td>.004 .007</td>
</tr>
<tr>
<td>Overall Width</td>
<td>E</td>
<td>.228 .236</td>
</tr>
<tr>
<td>Molded Package Width</td>
<td>E1</td>
<td>.150 .154</td>
</tr>
<tr>
<td>Overall Length</td>
<td>D</td>
<td>.337 .342</td>
</tr>
<tr>
<td>Chamfer Distance</td>
<td>h</td>
<td>.010 .015</td>
</tr>
<tr>
<td>Foot Length</td>
<td>L</td>
<td>.016 .033</td>
</tr>
<tr>
<td>Foot Angle</td>
<td>e</td>
<td>0 4 8</td>
</tr>
<tr>
<td>Lead Thickness</td>
<td>c</td>
<td>.008 .009</td>
</tr>
<tr>
<td>Lead Width</td>
<td>B</td>
<td>.014 .017</td>
</tr>
<tr>
<td>Mold Draft Angle Top</td>
<td>α</td>
<td>0 12 15</td>
</tr>
<tr>
<td>Mold Draft Angle Bottom</td>
<td>β</td>
<td>0 12 15</td>
</tr>
</tbody>
</table>

*Controlling Parameter

§ Significant Characteristic

**Notes:**
- Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010” (0.254mm) per side.
- JEDEC Equivalent: MS-012
- Drawing No. C04-065
## PIC16F684

14-Lead Plastic Thin Shrink Small Outline (ST) – 4.4 mm (TSSOP)

<table>
<thead>
<tr>
<th>Units</th>
<th>INCHES</th>
<th>MILLIMETERS*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dimension Limits</td>
<td>MIN</td>
<td>NOM</td>
</tr>
<tr>
<td>Number of Pins</td>
<td>$n$</td>
<td>14</td>
</tr>
<tr>
<td>Pitch</td>
<td>$P$</td>
<td>0.026 BSC</td>
</tr>
<tr>
<td>Overall Height</td>
<td>$A$</td>
<td>0.039</td>
</tr>
<tr>
<td>Molded Package Thickness</td>
<td>$A2$</td>
<td>0.033</td>
</tr>
<tr>
<td>Standoff</td>
<td>$A1$</td>
<td>0.002</td>
</tr>
<tr>
<td>Overall Width</td>
<td>$E$</td>
<td>0.246</td>
</tr>
<tr>
<td>Molded Package Width</td>
<td>$E1$</td>
<td>0.169</td>
</tr>
<tr>
<td>Molded Package Length</td>
<td>$D$</td>
<td>0.193</td>
</tr>
<tr>
<td>Foot Length</td>
<td>$L$</td>
<td>0.020</td>
</tr>
<tr>
<td>Foot Angle</td>
<td>$\theta$</td>
<td>0°</td>
</tr>
<tr>
<td>Lead Thickness</td>
<td>$c$</td>
<td>0.004</td>
</tr>
<tr>
<td>Lead Width</td>
<td>$B$</td>
<td>0.007</td>
</tr>
<tr>
<td>Mold Draft Angle Top</td>
<td>$\alpha$</td>
<td>12° REF</td>
</tr>
<tr>
<td>Mold Draft Angle Bottom</td>
<td>$\beta$</td>
<td>12° REF</td>
</tr>
</tbody>
</table>

### Notes:

Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.005" (0.127mm) per side.

*Controlling Parameter

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

See ASME Y14.5M

REF: Reference Dimension, usually without tolerance, for information purposes only.

See ASME Y14.5M

JEDEC Equivalent: MO-153 AB-1

Drawing No. C04-087

Revised: 08-17-05
16-Lead Plastic Quad Flat No Lead Package (ML) – 4x4x0.9 mm Body (QFN) Saw Singulated

<table>
<thead>
<tr>
<th>Units</th>
<th>INCHES</th>
<th>MILLIMETERS*</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MIN</td>
<td>NOM</td>
</tr>
<tr>
<td>Number of Pins</td>
<td>n</td>
<td>16</td>
</tr>
<tr>
<td>Pitch</td>
<td>e</td>
<td>0.026 BSC</td>
</tr>
<tr>
<td>Overall Height</td>
<td>A</td>
<td>0.031</td>
</tr>
<tr>
<td>Standoff</td>
<td>A1</td>
<td>0.000</td>
</tr>
<tr>
<td>Contact Thickness</td>
<td>A3</td>
<td>0.008 REF</td>
</tr>
<tr>
<td>Overall Width</td>
<td>E</td>
<td>0.152</td>
</tr>
<tr>
<td>Exposed Pad Width</td>
<td>E2</td>
<td>0.090**</td>
</tr>
<tr>
<td>Overall Length</td>
<td>D</td>
<td>0.152</td>
</tr>
<tr>
<td>Exposed Pad Length</td>
<td>D2</td>
<td>0.090</td>
</tr>
<tr>
<td>Contact Width</td>
<td>b</td>
<td>0.010</td>
</tr>
<tr>
<td>Contact Length</td>
<td>L</td>
<td>0.012</td>
</tr>
<tr>
<td>Contact-to-Exposed Pad</td>
<td>K</td>
<td>0.008</td>
</tr>
</tbody>
</table>

* Controlling Parameter
** Outside JEDEC Specification
§ Significant Characteristic

Notes:
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Exposed pad varies according to die attach paddle size.
3. BSC: Basic Dimension. Theoretically exact value shown without tolerances.
4. See ASME Y14.5M
5. REF: Reference Dimension, usually without tolerance, for information purposes only.
6. See ASME Y14.5M
7. JEDEC equivalent: MO-220 VGGC-3
8. Drawing No. C04-127

Revised 09-13-05
APPENDIX A: DATA SHEET

REVISION HISTORY

Revision A
This is a new data sheet.

Revision B
Rewrites of the Oscillator and Special Features of the CPU Sections. General corrections to Figures and formatting.

Revision C
Added Characterization Data, Updated Electrical Specifications. Incorporated Golden Chapter Sections for the following:
- Section 3.0 “Oscillator Module (With Fail-Safe Clock Monitor)”
- Section 5.0 “Timer0 Module”
- Section 6.0 “Timer1 Module with Gate Control”
- Section 7.0 “Timer2 Module”
- Section 8.0 “Comparator Module”
- Section 9.0 “Analog-to-Digital Converter (ADC) Module”
- Section 11.0 “Enhanced Capture/Compare/PWM (With Auto-Shutdown and Dead Band) Module”

APPENDIX B: MIGRATING FROM OTHER PICmicro® DEVICES

This discusses some of the issues in migrating from other PICmicro devices to the PIC16F6XX Family of devices.

B.1 PIC16F676 to PIC16F684

TABLE B-1: FEATURE COMPARISON

<table>
<thead>
<tr>
<th>Feature</th>
<th>PIC16F676</th>
<th>PIC16F684</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max Operating Speed</td>
<td>20 MHz</td>
<td>20 MHz</td>
</tr>
<tr>
<td>Max Program Memory (Words)</td>
<td>1024</td>
<td>2048</td>
</tr>
<tr>
<td>SRAM (bytes)</td>
<td>64</td>
<td>128</td>
</tr>
<tr>
<td>A/D Resolution</td>
<td>10-bit</td>
<td>10-bit</td>
</tr>
<tr>
<td>Data EEPROM (Bytes)</td>
<td>128</td>
<td>256</td>
</tr>
<tr>
<td>Timers (8/16-bit)</td>
<td>1/1</td>
<td>2/1</td>
</tr>
<tr>
<td>Oscillator Modes</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>Brown-out Reset</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Internal Pull-ups</td>
<td>RA0/1/2/4/5</td>
<td>RA0/1/2/4/5, MCLR</td>
</tr>
<tr>
<td>Interrupt-on-change</td>
<td>RA0/1/2/3/4/5</td>
<td>RA0/1/2/3/4/5</td>
</tr>
<tr>
<td>Comparator</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>ECCP</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>Ultra Low-Power Wake-Up</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>Extended WDT</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>Software Control Option of WDT/BOR</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>INTOSC Frequencies</td>
<td>4 MHz</td>
<td>32 kHz-8 MHz</td>
</tr>
<tr>
<td>Clock Switching</td>
<td>N</td>
<td>Y</td>
</tr>
</tbody>
</table>

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.
TRISA Register ......................................................... 31
TRISC Register ......................................................... 40
Two-Speed Clock Start-up Mode ......................... 27

U
Ultra Low-Power Wake-Up ...................................... 6, 32, 34

V
Voltage Reference. See Comparator Voltage Reference
(CVREF)
Voltage References
Associated Registers ............................................. 65
VREF. SEE ADC Reference Voltage

W
Wake-up Using Interrupts .......................................... 114
Watchdog Timer (WDT) .............................................. 112
Associated Registers .............................................. 113
Clock Source ......................................................... 112
Modes ..................................................................... 112
Period ..................................................................... 112
Specifications ...................................................... 145
WDTCON Register ................................................... 113
WPUA Register ......................................................... 33
WWW Address ......................................................... 189
WWW, On-Line Support ........................................... 4
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
RE: Reader Response

From: Name ____________________________
Company ____________________________
Address ____________________________
City / State / ZIP / Country ____________
Telephone: (______) _________ - _________  FAX: (______) _________ - _________

Application (optional):

Would you like a reply? _____ Y _____ N

Device: PIC16F684  Literature Number: DS41202D

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.

<table>
<thead>
<tr>
<th>PART NO.</th>
<th>X</th>
<th>XX</th>
<th>XXX</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device:</td>
<td>PIC16F684, PIC16F684T(1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>VDD range 2.0V to 5.5V</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Temperature Range:</td>
<td>I = -40°C to +85°C (Industrial)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>E = -40°C to +125°C (Extended)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Package:</td>
<td>ML = Quad Flat No Leads (QFN)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>P = Plastic DIP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SL = 14-lead Small Outline (150 mil)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ST = Thin Shrink Small Outline (4.4 mm)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pattern:</td>
<td>QTP, SQTPSM or ROM Code; Special Requirements (blank otherwise)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Examples:

a) PIC16F684-E/P 301 = Extended Temp., PDIP package, 20 MHz, QTP pattern #301
b) PIC16F684-I/SO = Industrial Temp., SOIC package, 20 MHz

Note 1: T = in tape and reel TSSOP, SOIC and QFN packages only.