PIC16F716
Data Sheet

8-bit Flash-based Microcontroller
with A/D Converter and
Enhanced Capture/Compare/PWM
Note the following details of the code protection feature on Microchip devices:

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

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

Information contained in this publication regarding device applications and the like is intended through suggestion only and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. No representation or warranty is given and no liability is assumed by Microchip Technology Incorporated with respect to the accuracy or use of such information, or infringement of patents or other intellectual property rights arising from such use or otherwise. Use of Microchip's products as critical components in life support systems is not authorized except with express written approval by Microchip. No licenses are conveyed, implicitly or otherwise, under any intellectual property rights.

© 2003, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

Printed on recycled paper.

Microchip received QS-9000 quality system certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona in July 1999 and Mountain View, California in March 2002. The Company's quality system processes and procedures are QS-9000 compliant for its PICmicro® 8-bit MCUs, dsPIC® code hopping devices, Serial EEPROMs, microperipherals, non-volatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001 certified.
**Microcontroller Core Features:**
- High-performance RISC CPU
- Only 35 single-word instructions to learn
  - All single-cycle instructions except for program branches which are two-cycle
- Operating speed: DC - 20 MHz clock input
  DC - 200 ns instruction cycle
- Interrupt capability
  (up to 7 internal/external interrupt sources)
- 8-level deep hardware stack
- Direct, Indirect and Relative Addressing modes

**Special Microcontroller Features**
- Power-on Reset (POR)
- Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Dual level Brown-out Reset circuitry
  - 2.5 V BOR (Typical)
  - 4.0 V BOR (Typical)
- Programmable code protection
- Power saving Sleep mode
- Selectable oscillator options
- Fully static design
- In-Circuit Serial Programming™ (ICSP™)

**CMOS Technology**
- Wide operating voltage range:
  - Industrial: 2.0V to 5.5V
  - Extended: 3.0V to 5.5V
- High Sink/Source Current 25/25 mA
- Wide temperature range:
  - Industrial: -40°C to 85°C
  - Extended: -40°C to 125°C

**Low-Power Features:**
- Standby Current:
  - 100 nA @ 2.0V, typical
- Operating Current:
  - 14 µA @ 32 kHz, 2.0V, typical
  - 120 µA @ 1 MHz, 2.0V, typical
- Watchdog Timer Circuit:
  - 1 µA @ 2.0V, typical
- Timer1 Oscillator Current:
  - 3.0 µA @ 32 kHz, 2.0V, typical

**Peripheral Features:**
- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler can be incremented during Sleep via external crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period register, prescaler and postscaler
- Enhanced Capture, Compare, PWM module:
  - Capture is 16-bit, max. resolution is 12.5 ns
  - Compare is 16-bit, max. resolution is 200 ns
  - PWM maximum resolution is 10-bit
- Enhanced PWM:
  - Single, Half-Bridge and Full-Bridge modes
  - Digitally programmable dead-band delay
  - Auto-shutdown/restart
- 8-bit multi-channel Analog-to-Digital converter
- 13 I/O pins with individual direction control
- Programmable weak pull-ups onPORTB

<table>
<thead>
<tr>
<th>Device</th>
<th>Memory</th>
<th>I/O</th>
<th>8-bit A/D (ch)</th>
<th>Timers 8/16</th>
<th>PWM (outputs)</th>
<th>VDD Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIC16F716</td>
<td>2048 x 14</td>
<td>128 x 8</td>
<td>13</td>
<td>4</td>
<td>2/1</td>
<td>1/2/4</td>
</tr>
</tbody>
</table>

© 2003 Microchip Technology Inc. Preliminary DS41206A-page 1
Pin Diagrams

18-pin PDIP, SOIC

20-pin SSOP
## Table of Contents

1.0 Device Overview .............................................................................................................. 5  
2.0 Memory Organization ........................................................................................................ 7  
3.0 I/O Ports .......................................................................................................................... 19  
4.0 Timer0 Module ................................................................................................................. 27  
5.0 Timer1 Module ................................................................................................................. 29  
6.0 Timer2 Module ................................................................................................................. 31  
7.0 Enhanced Capture/Compare/PWM (ECCP) Module .................................................... 33  
8.0 Analog-to-Digital Converter (A/D) Module ................................................................. 49  
9.0 Special Features of the CPU ............................................................................................ 55  
10.0 Instruction Set Summary ............................................................................................... 71  
11.0 Development Support ................................................................................................... 85  
12.0 Electrical Characteristics ............................................................................................. 91  
13.0 DC and AC Characteristics Graphs and Tables ........................................................ 107  
14.0 Packaging Information ............................................................................................... 109  
Appendix A: Revision History ............................................................................................. 113  
Appendix B: Conversion Considerations ........................................................................... 113  
Appendix C: Migration from Base-line to MID-RANGE Devices ...................................... 114  
On-Line Support .................................................................................................................. 115  
Systems Information and Upgrade Hot Line ..................................................................... 115  
Reader Response ................................................................................................................ 116  
Index .................................................................................................................................. 117  
Product Identification System ............................................................................................. 123

---

### 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@mail.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)
- The Microchip Corporate Literature Center; U.S. FAX: (480) 792-7277

When contacting a sales office or the literature center, 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/cn to receive the most current information on all of our products.
1.0 DEVICE OVERVIEW

This document contains device specific information for the PIC16F716. Additional information may be found in the PICmicro® Mid-Range Reference Manual, (DS33023), which may be obtained from your local Microchip Sales Representative or downloaded from the Microchip web site (www.microchip.com). The Reference Manual should be considered a complementary document to this data sheet, and is highly recommended reading for a better understanding of the device architecture and operation of the peripheral modules.

Figure 1-1 is the block diagram for the PIC16F716 device. The pinouts are listed in Table 1-1.

FIGURE 1-1: PIC16F716 BLOCK DIAGRAM

Note 1: Higher order bits are from the Status register.
### TABLE 1-1: PIC16F716 PINOUT DESCRIPTION

<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>MCLR/VPP</td>
<td>MCLR</td>
<td>ST</td>
<td>—</td>
<td>Master clear (Reset) input. This pin is an active low Reset to the device.</td>
</tr>
<tr>
<td>VPP</td>
<td>P</td>
<td>—</td>
<td>—</td>
<td>Programming voltage input</td>
</tr>
<tr>
<td>OSC1/CLKIN</td>
<td>OSC1</td>
<td>XTAL</td>
<td>—</td>
<td>Oscillator crystal input</td>
</tr>
<tr>
<td>CLKIN</td>
<td>CMOS</td>
<td>—</td>
<td>—</td>
<td>External clock source input</td>
</tr>
<tr>
<td>OSC2/CLKOUT</td>
<td>OSC2</td>
<td>XTAL</td>
<td>—</td>
<td>Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode.</td>
</tr>
<tr>
<td></td>
<td>CLKOUT</td>
<td>—</td>
<td>CMOS</td>
<td>In RC mode, OSC2 pin outputs CLKOUT which has 1/4 the frequency of OSC1, and denotes the instruction cycle rate.</td>
</tr>
<tr>
<td>RA0/AN0</td>
<td>RA0</td>
<td>TTL</td>
<td>CMOS</td>
<td>Bidirectional I/O</td>
</tr>
<tr>
<td></td>
<td>AN0</td>
<td>—</td>
<td>—</td>
<td>Analog Channel 0 input</td>
</tr>
<tr>
<td>RA1/AN1</td>
<td>RA1</td>
<td>TTL</td>
<td>CMOS</td>
<td>Bidirectional I/O</td>
</tr>
<tr>
<td></td>
<td>AN1</td>
<td>—</td>
<td>—</td>
<td>Analog Channel 1 input</td>
</tr>
<tr>
<td>RA2/AN2</td>
<td>RA2</td>
<td>TTL</td>
<td>CMOS</td>
<td>Bidirectional I/O</td>
</tr>
<tr>
<td></td>
<td>AN2</td>
<td>—</td>
<td>—</td>
<td>Analog Channel 2 input</td>
</tr>
<tr>
<td>RA3/AN3/VREF</td>
<td>RA3</td>
<td>TTL</td>
<td>CMOS</td>
<td>Bidirectional I/O</td>
</tr>
<tr>
<td></td>
<td>AN3</td>
<td>—</td>
<td>—</td>
<td>Analog Channel 3 input</td>
</tr>
<tr>
<td></td>
<td>VREF</td>
<td>—</td>
<td>—</td>
<td>A/D reference voltage input</td>
</tr>
<tr>
<td>RA4/T0CKI</td>
<td>RA4</td>
<td>ST</td>
<td>OD</td>
<td>Bidirectional I/O, Open drain when configured as output.</td>
</tr>
<tr>
<td></td>
<td>T0CKI</td>
<td>ST</td>
<td>—</td>
<td>Timer0 external clock input</td>
</tr>
<tr>
<td>RB0/INT/ECCPAS2</td>
<td>RB0</td>
<td>TTL</td>
<td>CMOS</td>
<td>Bidirectional I/O, Programmable weak pull-up.</td>
</tr>
<tr>
<td></td>
<td>INT</td>
<td>ST</td>
<td>—</td>
<td>External Interrupt</td>
</tr>
<tr>
<td></td>
<td>ECCPAS2</td>
<td>ST</td>
<td>—</td>
<td>ECCP Auto-Shutdown pin</td>
</tr>
<tr>
<td>RB1/T1OSO/T1CKI</td>
<td>RB1</td>
<td>TTL</td>
<td>CMOS</td>
<td>Bidirectional I/O, Programmable weak pull-up.</td>
</tr>
<tr>
<td></td>
<td>T1OSO</td>
<td>—</td>
<td>XTAL</td>
<td>Timer1 oscillator output. Connects to crystal in Oscillator mode.</td>
</tr>
<tr>
<td></td>
<td>T1CKI</td>
<td>ST</td>
<td>—</td>
<td>Timer1 external clock input</td>
</tr>
<tr>
<td>RB2/T1OSI</td>
<td>RB2</td>
<td>TTL</td>
<td>CMOS</td>
<td>Bidirectional I/O, Programmable weak pull-up.</td>
</tr>
<tr>
<td></td>
<td>T1OSI</td>
<td>XTAL</td>
<td>—</td>
<td>Timer1 oscillator input. Connects to crystal in Oscillator mode.</td>
</tr>
<tr>
<td>RB3/CCP1/P1A</td>
<td>RB3</td>
<td>TTL</td>
<td>CMOS</td>
<td>Bidirectional I/O, Programmable weak pull-up.</td>
</tr>
<tr>
<td></td>
<td>CCP1</td>
<td>ST</td>
<td>CMOS</td>
<td>Capture1 input, Compare1 output, PWM1 output.</td>
</tr>
<tr>
<td></td>
<td>P1A</td>
<td>—</td>
<td>CMOS</td>
<td>PWM P1A output</td>
</tr>
<tr>
<td>RB4/ECCPAS0</td>
<td>RB4</td>
<td>TTL</td>
<td>CMOS</td>
<td>Bidirectional I/O, Programmable weak pull-up, Interrupt-on-change.</td>
</tr>
<tr>
<td></td>
<td>ECCPAS0</td>
<td>ST</td>
<td>—</td>
<td>ECCP Auto-Shutdown pin</td>
</tr>
<tr>
<td>RB5/P1B</td>
<td>RB5</td>
<td>TTL</td>
<td>CMOS</td>
<td>Bidirectional I/O, Programmable weak pull-up, Interrupt-on-change.</td>
</tr>
<tr>
<td></td>
<td>P1B</td>
<td>—</td>
<td>CMOS</td>
<td>PWM P1B output</td>
</tr>
<tr>
<td>RB6/P1C</td>
<td>RB6</td>
<td>TTL</td>
<td>CMOS</td>
<td>Bidirectional I/O, Programmable weak pull-up, Interrupt-on-change.</td>
</tr>
<tr>
<td></td>
<td>P1C</td>
<td>—</td>
<td>CMOS</td>
<td>PWM P1C output</td>
</tr>
<tr>
<td>RB7/P1D</td>
<td>RB7</td>
<td>TTL</td>
<td>CMOS</td>
<td>Bidirectional I/O, Programmable weak pull-up, Interrupt-on-change.</td>
</tr>
<tr>
<td></td>
<td>P1D</td>
<td>—</td>
<td>CMOS</td>
<td>PWM P1D output</td>
</tr>
<tr>
<td>Vss</td>
<td>Vss</td>
<td>P</td>
<td>—</td>
<td>Ground reference for logic and I/O pins.</td>
</tr>
<tr>
<td>Vdd</td>
<td>Vdd</td>
<td>P</td>
<td>—</td>
<td>Positive supply for logic and I/O pins.</td>
</tr>
</tbody>
</table>

**Legend:**
- I = Input
- AN = Analog input or output
- OD = Open drain
- O = Output
- TTL = TTL compatible input
- ST = Schmitt Trigger input with CMOS levels
- P = Power
- XTAL = Crystal
- CMOS = CMOS compatible input or output
2.0 MEMORY ORGANIZATION

There are two memory blocks in the PIC16F716 PICmicro® microcontroller device. Each block (program memory and data memory) has its own bus so that concurrent access can occur.

Additional information on device memory may be found in the PICmicro® Mid-Range Reference Manual, (DS33023).

2.1 Program Memory Organization

The PIC16F716 has a 13-bit program counter capable of addressing an 8K x 14 program memory space. The PIC16F716 has 2K x 14 words of program memory. Accessing a location above the physically implemented address will cause a wrap-around.

The Reset vector is at 0000h and the interrupt vector is at 0004h.

2.2 Data Memory Organization

The data memory is partitioned into multiple banks which contain the General Purpose Registers (GPR) and the Special Function Registers (SFR). Bits RP1 and RP0 of the Status register are the bank select bits.

<table>
<thead>
<tr>
<th>RP1:RP0(1) (status&lt;6:5&gt;)</th>
<th>Bank</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0</td>
</tr>
<tr>
<td>01</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>2(2)</td>
</tr>
<tr>
<td>11</td>
<td>3(2)</td>
</tr>
</tbody>
</table>

Note 1: Maintain Status bit 6 clear to ensure upward compatibility with future products.

2: Not implemented

Each bank extends up to 7Fh (128 bytes). The lower locations of each bank are reserved for the Special Function Registers. Above the Special Function Registers are General Purpose Registers, implemented as static RAM. All implemented banks contain Special Function Registers. The upper 16 bytes of GPR space and some “high use” Special Function Registers in Bank 0 are mirrored in Bank 1 for code reduction and quicker access.
### GENERAL PURPOSE REGISTER FILE

The register file can be accessed either directly or indirectly through the File Select Register FSR (Section 2.5 "Indirect Addressing, INDF and FSR Registers").

![REGISTER FILE MAP](image)

<table>
<thead>
<tr>
<th>File Address</th>
<th>File Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>00h</td>
<td>INDF(1)</td>
</tr>
<tr>
<td>01h</td>
<td>TMR0 OPTION_REG</td>
</tr>
<tr>
<td>02h</td>
<td>PCL</td>
</tr>
<tr>
<td>03h</td>
<td>STATUS</td>
</tr>
<tr>
<td>04h</td>
<td>FSR</td>
</tr>
<tr>
<td>05h</td>
<td>PORTA TRISA</td>
</tr>
<tr>
<td>06h</td>
<td>PORTB TRISB</td>
</tr>
<tr>
<td>07h</td>
<td></td>
</tr>
<tr>
<td>08h</td>
<td></td>
</tr>
<tr>
<td>09h</td>
<td></td>
</tr>
<tr>
<td>0Ah</td>
<td>PCLATH</td>
</tr>
<tr>
<td>0Bh</td>
<td>INTCON</td>
</tr>
<tr>
<td>0Ch</td>
<td>PIR1 PIE1</td>
</tr>
<tr>
<td>0Dh</td>
<td></td>
</tr>
<tr>
<td>0Eh</td>
<td>TMR1L PCON</td>
</tr>
<tr>
<td>0Fh</td>
<td>TMR1H</td>
</tr>
<tr>
<td>10h</td>
<td>T1CON</td>
</tr>
<tr>
<td>11h</td>
<td>TMR2</td>
</tr>
<tr>
<td>12h</td>
<td>T2CON PR2</td>
</tr>
<tr>
<td>13h</td>
<td></td>
</tr>
<tr>
<td>14h</td>
<td></td>
</tr>
<tr>
<td>15h</td>
<td>CCPR1L</td>
</tr>
<tr>
<td>16h</td>
<td>CCPR1H</td>
</tr>
<tr>
<td>17h</td>
<td>CCP1CON</td>
</tr>
<tr>
<td>18h</td>
<td>PWM1CON</td>
</tr>
<tr>
<td>19h</td>
<td>ECCPAS</td>
</tr>
<tr>
<td>1Ah</td>
<td></td>
</tr>
<tr>
<td>1Bh</td>
<td></td>
</tr>
<tr>
<td>1Ch</td>
<td></td>
</tr>
<tr>
<td>1Dh</td>
<td></td>
</tr>
<tr>
<td>1Eh</td>
<td>ADRES</td>
</tr>
<tr>
<td>1Fh</td>
<td>ADCON0</td>
</tr>
<tr>
<td>20h</td>
<td>General Purpose Registers</td>
</tr>
<tr>
<td></td>
<td>General Purpose Registers 32 Bytes</td>
</tr>
<tr>
<td>6Fh</td>
<td></td>
</tr>
<tr>
<td>70h</td>
<td>16 Bytes Accesses</td>
</tr>
<tr>
<td>77h</td>
<td></td>
</tr>
</tbody>
</table>

Note 1: Not a physical register.
2.2.2 SPECIAL FUNCTION REGISTERS

The Special Function Registers are registers used by the CPU and peripheral modules for controlling the desired operation of the device. These registers are implemented as static RAM. A list of these registers is given in Table 2-1.

The Special Function Registers can be classified into two sets; core (CPU) and peripheral. Those registers associated with the core functions are described in detail in this section. Those related to the operation of the peripheral features are described in detail in that peripheral feature section.

**TABLE 2-1: SPECIAL FUNCTION REGISTER SUMMARY BANK 0**

<table>
<thead>
<tr>
<th>Address</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(1)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>01h</td>
<td>TMR0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>02h</td>
<td>PCL(1)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>03h</td>
<td>STATUS(1)</td>
<td>IRP(4)</td>
<td>RP1(4)</td>
<td>RP0</td>
<td>TQ</td>
<td>PD</td>
<td>Z</td>
<td>DC</td>
<td>C</td>
<td></td>
<td></td>
</tr>
<tr>
<td>04h</td>
<td>FSR(1)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>05h</td>
<td>PORTA (6, 8)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>PORTA Data Latch when written: PORTA pins when read</td>
<td></td>
</tr>
<tr>
<td>06h</td>
<td>PORTB (6, 8)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>PORTB Data Latch when written: PORTB pins when read</td>
<td></td>
</tr>
<tr>
<td>07h-09h</td>
<td></td>
<td></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(1, 2)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Write Buffer for the upper 5 bits of the Program Counter</td>
<td></td>
</tr>
<tr>
<td>0Bh</td>
<td>INTCON(1)</td>
<td>GIE</td>
<td>PEIE</td>
<td>TOIE</td>
<td>INTE</td>
<td>RBIE</td>
<td>T0IF</td>
<td>INTF</td>
<td>RBIF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0Ch</td>
<td>PIR1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CCP1IF</td>
<td>TMR2IF</td>
</tr>
<tr>
<td>0Dh</td>
<td></td>
<td></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></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0Fh</td>
<td>TMR1H</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10h</td>
<td>T1CON</td>
<td></td>
<td></td>
<td>T1CKPS1</td>
<td>T1CKPS0</td>
<td>T1OSCEN</td>
<td>T1SYNC</td>
<td>TMR1CS</td>
<td>TMR1ON</td>
<td></td>
<td></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>Timer2 module’s register</td>
<td></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></td>
<td></td>
</tr>
<tr>
<td>13h-14h</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15h</td>
<td>CCP1CON(1)</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></td>
<td></td>
</tr>
<tr>
<td>16h</td>
<td>CCP1RH(1)</td>
<td>Capture/Compare/PWM Register 1 (LSB)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17h</td>
<td>CCP1ICON</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></td>
<td></td>
</tr>
<tr>
<td>18h</td>
<td>PWM1ICON</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></td>
<td></td>
</tr>
<tr>
<td>19h</td>
<td>ECCPAS(8)</td>
<td>ECCPASE</td>
<td>ECCPAS2</td>
<td>ECCPAS0</td>
<td>PSSAC1</td>
<td>PSSAC0</td>
<td>PSSBD1</td>
<td>PSSBD0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1Ah-1Dh</td>
<td></td>
<td></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>ADRES</td>
<td>A/D Result Register</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1Fh</td>
<td>ADCCON0</td>
<td>ADS1</td>
<td>ADCS0</td>
<td>CHS2</td>
<td>CHS1</td>
<td>CHS0</td>
<td>GO/DONE</td>
<td></td>
<td>ADON</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

**Note:**
1: These registers can be addressed from either bank.
2: The upper byte of the program counter is not directly accessible. PCLATH is a holding register for PC<12:8> whose contents are transferred to the upper byte of the program counter.
3: Other (non Power-up) Resets include: external Reset through MCLR and the Watchdog Timer Reset.
4: The IRP and RP1 bits are reserved. Always maintain these bits clear.
5: On any device Reset, these pins are configured as inputs.
6: This is the value that will be in the port output latch.
7: Reserved bits, do not use.
8: ECCPAS1 bit is not used on PIC16F716.
### TABLE 2-2: SPECIAL FUNCTION REGISTER SUMMARY BANK 1

<table>
<thead>
<tr>
<th>Address</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:</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>80h</td>
<td>INDF</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Addressing this location uses contents of FSR to address data memory (not a physical register)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0000 0000</td>
<td>18</td>
</tr>
<tr>
<td>81h</td>
<td>OPTION_REG</td>
<td>RBPU</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>12</td>
</tr>
<tr>
<td>82h</td>
<td>PCL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1111 1111</td>
<td>17</td>
</tr>
<tr>
<td>83h</td>
<td>STATUS</td>
<td>IRP(4)</td>
<td>RP1(4)</td>
<td>RP0</td>
<td>T0</td>
<td>PD</td>
<td>Z</td>
<td>DC</td>
<td>C</td>
<td>0001 1xxx</td>
<td>11</td>
</tr>
<tr>
<td>84h</td>
<td>FSR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>18</td>
</tr>
<tr>
<td>85h</td>
<td>TRISA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>86h</td>
<td>TRISB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>87h-89h</td>
<td>Unimplemented</td>
<td></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></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8Bh</td>
<td>INTCON</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8Ch</td>
<td>PIE1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8Dh</td>
<td>Unimplemented</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8Eh</td>
<td>PCON</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8Fh-91h</td>
<td>Unimplemented</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>92h</td>
<td>PR2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>93h-99h</td>
<td>Unimplemented</td>
<td></td>
<td></td>
<td></td>
<td></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></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Legend:**
- x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, read as '0', Shaded locations are unimplemented, read as '0'.

**Note:**
1: These registers can be addressed from either bank.
2: The upper byte of the program counter is not directly accessible. PCLATH is a holding register for PC<12:8> whose contents are transferred to the upper byte of the program counter.
3: Other (non Power-up) Resets include: external Reset through MCLR and the Watchdog Timer Reset.
4: The IRP and RP1 bits are reserved. Always maintain these bits clear.
5: On any device Reset, these pins are configured as inputs.
6: This is the value that will be in the port output latch.
7: Reserved bits, do not use.

**Notes:**
- Note 1: These registers can be addressed from either bank.
- Note 2: The upper byte of the program counter is not directly accessible. PCLATH is a holding register for PC<12:8> whose contents are transferred to the upper byte of the program counter.
- Note 3: Other (non Power-up) Resets include: external Reset through MCLR and the Watchdog Timer Reset.
- Note 4: The IRP and RP1 bits are reserved. Always maintain these bits clear.
- Note 5: On any device Reset, these pins are configured as inputs.
- Note 6: This is the value that will be in the port output latch.
- Note 7: Reserved bits, do not use.
2.2.2.1 Status Register

The Status register, shown in Register 2-1, contains the arithmetic status of the ALU, the Reset status and the bank select bits for data memory.

The Status register can be the destination for any instruction, as with 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 uulu (where u = unchanged).

It is recommended, therefore, that only BCF, BSF, SWAPF and MOVWF instructions are used to alter the Status register because these instructions do not affect the Z, C or DC bits from the Status register. For other instructions, not affecting any Status bits, see the “Instruction Set Summary.”

Note 1: The PIC16F716 does not use bits IRP and RP1 (STATUS<7:6>). Maintain these bits clear to ensure upward compatibility with future products.

2: The C and DC bits operate as a borrow and digit borrow bit, respectively, in subtraction. See the SUBLW and SUBWF instructions for examples.

REGISTER 2-1: STATUS REGISTER (ADDRESS: 03h, 83h)

<table>
<thead>
<tr>
<th>R/W-0</th>
<th>R/W-0</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(1)</td>
<td>RP1(1)</td>
<td>RP0</td>
<td>TO</td>
<td>PD</td>
<td>Z</td>
<td>DC</td>
<td>C</td>
</tr>
</tbody>
</table>

bit 7    IRP: Register Bank Select bit (used for indirect addressing)(1)
1 = Bank 2, 3 (100h – 1FFh)
0 = Bank 0, 1 (00h – FFh)

bit 6-5 RP1(1):RP0: Register Bank Select bits (used for direct addressing)
01 = Bank 1 (80h – FFh)
00 = Bank 0 (00h – 7Fh)
Each bank is 128 bytes

bit 4 TO: Time-out bit
1 = After power-up, CLRWDT instruction or SLEEP instruction
0 = A WDT time-out occurred

bit 3 PD: Power-down bit
1 = After power-up or by the CLRWDT instruction
0 = By execution of the SLEEP instruction

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

bit 1 DC: Digit carry/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 (ADDWF, ADDLW, SUBLW, SUBWF instructions)(2)
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: Reserved, maintain clear

2: 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 or low order bit of the source register.

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

© 2003 Microchip Technology Inc.
### OPTION_REG Register

The OPTION_REG register is a readable and writable register, which contains various control bits to configure the TMR0 prescaler/WDT postscaler (single assignable register known also as the prescaler), the External INT Interrupt, TMR0 and the weak pull-ups on PORTB.

**REGISTER 2-2: OPTION_REG REGISTER (ADDRESS: 81h)**

<table>
<thead>
<tr>
<th>Bit 7</th>
<th>Bit 6</th>
<th>Bit 5</th>
<th>Bit 4</th>
<th>Bit 3</th>
<th>Bit 2-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>RBPU</td>
<td>INTEDG</td>
<td>T0CS</td>
<td>T0SE</td>
<td>PSA</td>
<td>PS2:PS0</td>
</tr>
</tbody>
</table>

**Bit Values:**

- **RBPU** (PORTB Weak Pull-up Enable bit)
  - 1 = PORTB weak pull-ups are disabled
  - 0 = PORTB weak pull-ups are determined by alternate function or TRISBn bit value

- **INTEDG** (Interrupt Edge Select bit)
  - 1 = Interrupt on rising edge of RB0/INT pin
  - 0 = Interrupt on falling edge of RB0/INT pin

- **T0CS** (TMR0 Clock Source Select bit)
  - 1 = Transition on RA4/T0CKI pin
  - 0 = Internal instruction cycle clock (CLKOUT)

- **T0SE** (TMR0 Source Edge Select bit)
  - 1 = Increment on high-to-low transition on RA4/T0CKI pin
  - 0 = Increment on low-to-high transition on RA4/T0CKI pin

- **PSA** (Prescaler Assignment bit)
  - 1 = Prescaler is assigned to the WDT
  - 0 = Prescaler is assigned to the Timer0 module

- **PS2:PS0** (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>

**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

---

**Note:** To achieve a 1:1 prescaler assignment for the TMR0 register, assign the prescaler to the Watchdog Timer.
### 2.2.2.3 INTCON Register

The INTCON Register is a readable and writable register which contains various enable and flag bits for the TMR0 register overflow, RB Port change and external RB0/INT pin interrupts.

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

#### REGISTER 2-3: INTCON REGISTER (ADDRESS: 0Bh, 8Bh)

<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-x</th>
</tr>
</thead>
<tbody>
<tr>
<td>GIE</td>
<td>PEIE</td>
<td>T0IE</td>
<td>INTE</td>
<td>RBIE</td>
<td>T0IF</td>
<td>INTF</td>
<td>RBIF</td>
</tr>
</tbody>
</table>

- **bit 7**
  - **GIE**: Global Interrupt Enable bit
    - 1 = Enables all un-masked interrupts
    - 0 = Disables all interrupts

- **bit 6**
  - **PEIE**: Peripheral Interrupt Enable bit
    - 1 = Enables all un-masked peripheral interrupts
    - 0 = Disables all peripheral interrupts

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

- **bit 4**
  - **INTE**: RB0/INT External Interrupt Enable bit
    - 1 = Enables the RB0/INT external interrupt
    - 0 = Disables the RB0/INT external interrupt

- **bit 3**
  - **RBIE**: RB Port Change Interrupt Enable bit
    - 1 = Enables the RB port change interrupt
    - 0 = Disables the RB port change interrupt

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

- **bit 1**
  - **INTF**: RB0/INT External Interrupt Flag bit
    - 1 = The RB0/INT external interrupt occurred (must be cleared in software)
    - 0 = The RB0/INT external interrupt did not occur

- **bit 0**
  - **RBIF**: RB Port Change Interrupt Flag bit
    - 1 = At least one of the RB7:RB4 pins changed state (must be cleared in software)
    - 0 = None of the RB7:RB4 pins have changed state

**Legend:**
- R = Readable bit
- W = Writable bit
- U = Unimplemented bit, read as ‘0’
- -n = Value at POR
- ‘1’ = Bit is set
- ‘0’ = Bit is cleared
- x = Bit is unknown
2.2.2.4 PIE1 Register

This register contains the individual enable bits for the peripheral interrupts.

**REGISTER 2-4: PIE1 REGISTER (ADDRESS: 8Ch)**

<table>
<thead>
<tr>
<th>bit 7</th>
<th>bit 6</th>
<th>bit 5-3</th>
<th>bit 2</th>
<th>bit 1</th>
<th>bit 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>U-0</td>
<td>R/W-0</td>
<td>U-0</td>
<td>U-0</td>
<td>U-0</td>
<td>R/W-0</td>
</tr>
<tr>
<td>—</td>
<td>ADIE</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>CCP1IE</td>
</tr>
<tr>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>TMR2IE</td>
</tr>
<tr>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>TMR1IE</td>
</tr>
<tr>
<td>U-0</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>R/W-0</td>
</tr>
<tr>
<td>U-0</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>R/W-0</td>
</tr>
<tr>
<td>U-0</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>R/W-0</td>
</tr>
</tbody>
</table>

- **bit 7** Unimplemented: Read as '0'
- **bit 6** ADIE: A/D Converter Interrupt Enable bit
  - 1 = Enables the A/D interrupt
  - 0 = Disables the A/D interrupt
- **bit 5-3** Unimplemented: Read as '0'
- **bit 2** CCP1IE: CCP1 Interrupt Enable bit
  - 1 = Enables the CCP1 interrupt
  - 0 = Disables the CCP1 interrupt
- **bit 1** TMR2IE: TMR2 to PR2 Match Interrupt Enable bit
  - 1 = Enables the TMR2 to PR2 match interrupt
  - 0 = Disables the TMR2 to PR2 match interrupt
- **bit 0** TMR1IE: TMR1 Overflow Interrupt Enable bit
  - 1 = Enables the TMR1 overflow interrupt
  - 0 = Disables the TMR1 overflow interrupt

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

**Note:** Bit PEIE (INTCON<6>) must be set to enable any peripheral interrupt.
2.2.2.5 PIR1 Register

This register contains the individual flag bits for the peripheral interrupts.

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

**REGISTER 2-5: PIR1 REGISTER (ADDRESS: 0Ch)**

<table>
<thead>
<tr>
<th></th>
<th>U-0</th>
<th>R/W-0</th>
<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>
</tr>
</thead>
<tbody>
<tr>
<td>—</td>
<td>—</td>
<td>ADIF</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>CCP1IF</td>
<td>TMR2IF</td>
<td>TMR1IF</td>
</tr>
</tbody>
</table>

bit 7

Unimplemented: Read as ‘0’

bit 6

ADIF: A/D Converter Interrupt Flag bit

1 = An A/D conversion completed (must be cleared in software)
0 = The A/D conversion is not complete

bit 5-3

Unimplemented: Read as ‘0’

bit 2

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 1

TMR2IF: TMR2 to PR2 Match Interrupt Flag bit

1 = TMR2 to PR2 match occurred (must be cleared in software)
0 = No TMR2 to PR2 match occurred

bit 0

TMR1IF: TMR1 Overflow Interrupt Flag bit

1 = TMR1 register overflowed (must be cleared in software)
0 = TMR1 register did not overflow

**Legend:**

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

The Power Control (PCON) register contains a flag bit to allow differentiation between a Power-on Reset (POR) to an external MCLR Reset or WDT Reset. These devices contain an additional bit to differentiate a Brown-out Reset condition from a Power-on Reset condition.

### REGISTER 2-6: PCON REGISTER (ADDRESS: 8Eh)

<table>
<thead>
<tr>
<th>Bit 7-2</th>
<th>Description</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unimplemented: Read as ‘0’</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bit 1</th>
<th>Description</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>POR: Power-on Reset Status bit</td>
<td>1 = No Power-on Reset occurred</td>
<td>0 = A Power-on Reset occurred (must be set in software after a Power-on Reset occurs)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bit 0</th>
<th>Description</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>BOR: Brown-out Reset Status bit</td>
<td>1 = No Brown-out Reset occurred</td>
<td>0 = A Brown-out Reset occurred (must be set in software after a Brown-out Reset occurs)</td>
</tr>
</tbody>
</table>

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

**Note:** If the BOREN configuration bit is set, BOR is ‘1’ on Power-on Reset and reset to ‘0’ when a Brown-out condition occurs. BOR must then be set by the user and checked on subsequent resets to see if it is clear, indicating that another Brown-out has occurred.

If the BOREN configuration bit is clear, BOR is unknown on Power-on Reset.
2.3 PCL and PCLATH

The Program Counter (PC) specifies the address of the instruction to fetch for execution. The PC is 13 bits wide. The low byte is called the PCL register. This register is readable and writable. The high byte is called the PCH register. This register contains the PC<12:8> bits and is not directly readable or writable. All updates to the PCH register go through the PCLATH register.

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 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. When the lower 8 bits are then 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.

Care should be exercised when modifying the PCL register to jump into a look-up table or program branch table (computed GOTO). With PCLATH set to the table start address, if the table 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 address.

2.3.2 PROGRAM MEMORY PAGING

The CALL and GOTO instructions provide 11 bits of address to allow branching within any 2K program memory page. When doing a CALL or GOTO instruction, the upper bit of the address is provided by PCLATH<3>. When doing a CALL or GOTO instruction, the user must ensure that the page select bit is programmed so that the desired program memory page is addressed. If a RETURN from a CALL instruction (or interrupt) is executed, the entire 13-bit PC is pushed onto the stack. Therefore, manipulation of the PCLATH<3> bit is not required for the RETURN instructions (which POPs the address from the stack).

2.4 Stack

The stack allows a combination of up to 8 program calls and interrupts to occur. The stack contains the return address from this branch in program execution.

Mid-range devices have an 8-level deep x 13-bit wide hardware stack. The stack space is not part of either program or data space, and the stack pointer is not readable or writable. The PC is PUSHed onto the stack when a CALL instruction is executed or an interrupt causes a branch. The stack is POPed in the event of a RETURN, RETLW or a RETFIE instruction execution. PCLATH is not modified when the stack is PUSHed or POPed.

After the stack has been PUSHed 8 times, the ninth push overwrites the value that was stored from the first push. The tenth push overwrites the second push (and so on).
2.5 Indirect Addressing, INDF and FSR Registers

The INDF register is not a physical register. Addressing INDF actually addresses the register whose address is contained in the FSR register (FSR is a pointer). This is indirect addressing.

**EXAMPLE 2-1: INDIRECT ADDRESSING**

- Register file 05 contains the value 10h
- Register file 06 contains the value 0Ah
- Load the value 05 into the FSR register
- A read of the INDF register will return the value of 10h
- Increment the value of the FSR register by one (FSR = 06)
- A read of the INDR register now will return the value of 0Ah.

Reading INDF itself indirectly (FSR = 0) will produce 00h. Writing to the INDF register indirectly results in a no-operation (although Status bits may be affected).

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

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

<table>
<thead>
<tr>
<th>Direct Addressing</th>
<th>Indirect Addressing</th>
</tr>
</thead>
<tbody>
<tr>
<td>RP1:RP0</td>
<td>IRP</td>
</tr>
<tr>
<td>[2]</td>
<td>[2]</td>
</tr>
<tr>
<td>bank select</td>
<td>bank select</td>
</tr>
<tr>
<td>location select</td>
<td>location select</td>
</tr>
<tr>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>from opcode</td>
<td>FSR register</td>
</tr>
<tr>
<td>00h</td>
<td>0</td>
</tr>
<tr>
<td>01h</td>
<td>1</td>
</tr>
<tr>
<td>10h</td>
<td>11</td>
</tr>
<tr>
<td>180h</td>
<td>(3)</td>
</tr>
<tr>
<td>1FFh</td>
<td>(3)</td>
</tr>
<tr>
<td>17Fh</td>
<td>FFh</td>
</tr>
<tr>
<td>7Fh</td>
<td>00h</td>
</tr>
<tr>
<td>Bank 0</td>
<td>Bank 1</td>
</tr>
<tr>
<td>Bank 2</td>
<td>Bank 3</td>
</tr>
</tbody>
</table>

**Note**
1: For register file map detail see Figure 2-2.
2: Maintain clear for upward compatibility with future products.
3: Not implemented.

An effective 9-bit address is obtained by concatenating the 8-bit FSR register and the IRP bit (STATUS<7>), as shown in Figure 2-4. However, IRP is not used in the PIC16F716.
3.0 I/O PORTS

Some pins for these I/O ports are multiplexed with an alternate function for the peripheral features on the device. In general, when a peripheral is enabled, that pin may not be used as a general purpose I/O pin.

Additional information on I/O ports may be found in the PICmicro® Mid-Range Reference Manual, (DS33023).

3.1 PORTA and the TRISA Register

PORTA is a 5-bit wide bidirectional port. The corresponding data direction register is TRISA. Setting a TRISA bit (= 1) will make the corresponding PORTA pin an input (i.e., put the corresponding output driver in a High-impedance mode). Clearing a TRISA bit (= 0) will make the corresponding PORTA pin an output (i.e., put the contents of the output latch on the selected pin).

Reading the PORTA register reads the status of the pins, whereas writing to it will write to the port latch. All write operations are read-modify-write operations. Therefore, a write to a port implies that the port pins are read, the value is modified and then written to the port data latch.

Pin RA4 is multiplexed with the Timer0 module clock input to become the RA4/T0CKI pin. The RA4/T0CKI pin is a Schmitt Trigger input and an open drain output. All other RA port pins have TTL input levels and full CMOS output drivers.

PORTA pins, RA3:0, are multiplexed with analog inputs and analog VREF input. The operation of each pin is selected by clearing/setting the control bits in the ADCON1 register (A/D Control Register 1).

Note: On a Power-on Reset, these pins are configured as analog inputs and read as ‘0’.

The TRISA register controls the direction of the RA 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.

Note: Setting RA3:0 to output while in Analog mode will force pins to output contents of data latch.

EXAMPLE 3-1: INITIALIZING PORTA

```
BCF STATUS, RP0 ; Initialize PORTA
CLRF PORTA ; clearing output
BSF STATUS, RP0 ; Select Bank 1
MOVLW 0xEF ; Value used to
MOVWF TRISA ; initialize data
MOVWF TRISA ; direction
// Set RA<3:0> as inputs
// RA<4> as outputs
BCF STATUS, RP0 ; Return to Bank 0
```

FIGURE 3-1: BLOCK DIAGRAM OF RA3:RA0
FIGURE 3-2: BLOCK DIAGRAM OF RA4/T0CKI PIN

TABLE 3-1: PORTA FUNCTIONS

<table>
<thead>
<tr>
<th>Name</th>
<th>Bit#</th>
<th>Buffer</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>RA0/AN0</td>
<td>bit 0</td>
<td>TTL</td>
<td>Input/output or analog input</td>
</tr>
<tr>
<td>RA1/AN1</td>
<td>bit 1</td>
<td>TTL</td>
<td>Input/output or analog input</td>
</tr>
<tr>
<td>RA2/AN2</td>
<td>bit 2</td>
<td>TTL</td>
<td>Input/output or analog input</td>
</tr>
<tr>
<td>RA3/AN3/VREF</td>
<td>bit 3</td>
<td>TTL</td>
<td>Input/output or analog input or VREF</td>
</tr>
<tr>
<td>RA4/T0CKI</td>
<td>bit 4</td>
<td>ST</td>
<td>Input/output or external clock input for Timer0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Output is open drain type</td>
</tr>
</tbody>
</table>

Legend: TTL = TTL input, ST = Schmitt Trigger input

TABLE 3-2: SUMMARY OF REGISTERS ASSOCIATED WITH PORTA

<table>
<thead>
<tr>
<th>Address</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>Value on all other Resets</th>
</tr>
</thead>
<tbody>
<tr>
<td>05h</td>
<td>PORTA</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>RA4</td>
<td>RA3</td>
<td>RA2</td>
<td>RA1</td>
<td>RA0</td>
<td>--xx 0000</td>
<td>--uu uuuu</td>
</tr>
<tr>
<td>85h</td>
<td>TRISA</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>PORTA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>--11 1111</td>
<td>--11 1111</td>
</tr>
<tr>
<td>9Fh</td>
<td>ADCON1</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>PCFG2</td>
<td>PCFG1</td>
<td>PCFG0</td>
<td></td>
<td></td>
<td>----- 0000</td>
<td>----- -000</td>
</tr>
</tbody>
</table>

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

Note 1: Reserved bits, do not use.
3.2 PORTB and the TRISB Register

PORTB is an 8-bit wide bidirectional port. The corresponding data direction register is TRISB. Setting a TRISB bit (= 1) will make the corresponding PORTB pin an input (i.e., put the corresponding output driver in a High-impedance mode). Clearing a TRISB bit (= 0) will make the corresponding PORTB pin an output (i.e., put the contents of the output latch on the selected pin).

EXAMPLE 3-2: INITIALIZING PORTB

```
BCF STATUS, RP0        ; select Bank 0
CLRF PORTB            ; clearing output
                     ; data latches
BSF STATUS, RP0       ; Select Bank 1
MOVLW 0xCF           ; Value used to
                     ; initialize data
                     ; direction
MOVWF TRISB           ; Set RB<3:0> as inputs
                     ; RB<5:4> as outputs
                     ; RB<7:6> as inputs
```

Each of the PORTB pins has a weak internal pull-up. A single control bit can turn on all the pull-ups. This is performed by clearing bit RBPU (OPTION_REG<7>). The 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.

FIGURE 3-3: BLOCK DIAGRAM OF RB0/INT/ECCPAS2 PIN

PORTB pins RB7:RB0 are multiplexed with several peripheral functions (Table 3-3).

When enabling peripheral functions, care should be taken in defining TRIS bits for each PORTB pin. Some peripherals override the TRIS bit to make a pin an input, while other peripherals override the TRIS bit to make a pin an output. Since the TRIS bit override is in effect while the peripheral is enabled, read-modify-write instructions (such as BSF, BCF, XORWF) with TRISB as the destination should be avoided. The user should refer to the corresponding peripheral section for the correct TRIS bit settings.

Four of PORTB’s pins, RB7:RB4, have an interrupt-on-change feature. Only pins configured as inputs can cause this interrupt to occur (i.e., any RB7:RB4 pin configured as an output is excluded from the interrupt-on-change comparison). The input pins, RB7:RB4, are compared with the old value latched on the last read of PORTB. The “mismatch” outputs of RB7:RB4 are OR’ed together to generate the RB Port Change Interrupt with flag bit RBIF (INTCON<0>).

This interrupt can wake the device from Sleep. The user, in the interrupt service routine, can clear the interrupt in the following manner:

1. Perform a read of PORTB to end the mismatch condition.
2. Clear flag bit RBIF.

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

The interrupt-on-change feature is recommended for wake-up on key depression operation and operations where PORTB is only used for the interrupt-on-change feature. Polling of PORTB is not recommended while using the interrupt-on-change feature.

Note 1: To enable weak pull-ups, set the appropriate TRIS bit(s) and clear the RBPU bit (OPTION_REG<7>).
FIGURE 3-4: BLOCK DIAGRAM OF RB1/T1OSO/T1CKI PIN

Note 1: To enable weak pull-ups, set the appropriate TRIS bit(s) and clear the RBPU bit (OPTION_REG<7>).

FIGURE 3-5: BLOCK DIAGRAM OF RB2/T1OSI PIN

Note 1: To enable weak pull-ups, set the appropriate TRIS bit(s) and clear the RBPU bit (OPTION_REG<7>).
FIGURE 3-6: BLOCK DIAGRAM OF RB3/CCP1/P1A PIN

Note 1: To enable weak pull-ups, set the appropriate TRIS bit(s) and clear the RBPU bit (OPTION_REG<7>).

FIGURE 3-7: BLOCK DIAGRAM OF RB4/ECCPAS0 PIN

Note 1: To enable weak pull-ups, set the appropriate TRIS bit(s) and clear the RBPU bit (OPTION_REG<7>).
FIGURE 3-8: BLOCK DIAGRAM OF RB5/P1B PIN

FIGURE 3-9: BLOCK DIAGRAM OF RB6/P1C PIN

Note 1: To enable weak pull-ups, set the appropriate TRIS bit(s) and clear the RBPU bit (OPTION_REG<7>).

Note 1: To enable weak pull-ups, set the appropriate TRIS bit(s) and clear the RBPU bit (OPTION_REG<7>).
TABLE 3-3: PORTB FUNCTIONS

<table>
<thead>
<tr>
<th>Name</th>
<th>Bit#</th>
<th>Buffer</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>RB0/INT/ ECCPAS2</td>
<td>bit 0</td>
<td>TTL/ST(1)</td>
<td>Input/output pin or external interrupt input. Internal software programmable weak pull-up. ECCP auto-shutdown input.</td>
</tr>
<tr>
<td>RB1/T1OS0/ T1CKI</td>
<td>bit 1</td>
<td>TTL/ST(1)</td>
<td>Input/output pin or Timer1 oscillator output, or Timer1 clock input. Internal software programmable weak pull-up. See Section 5.0 “Timer1 Module” for detailed operation.</td>
</tr>
<tr>
<td>RB2/T1OSI</td>
<td>bit 2</td>
<td>TTL/XTAL</td>
<td>Input/output pin or Timer1 oscillator input. Internal software programmable weak pull-up. See Section 5.0 “Timer1 Module” for detailed operation.</td>
</tr>
<tr>
<td>RB3/CCP1/ P1A</td>
<td>bit 3</td>
<td>TTL/ST(1)</td>
<td>Input/output pin or Capture1 input, or Compare1 output, or PWM A output. Internal software programmable weak pull-up. See CCP1 section for detailed operation.</td>
</tr>
<tr>
<td>RB4/ ECCPAS0</td>
<td>bit 4</td>
<td>TTL</td>
<td>Input/output pin (with interrupt-on-change). Internal software programmable weak pull-up. ECCP auto-shutdown input.</td>
</tr>
<tr>
<td>RB5/P1B</td>
<td>bit 5</td>
<td>TTL</td>
<td>Input/output pin (with interrupt-on-change). Internal software programmable weak pull-up. PWM B output.</td>
</tr>
<tr>
<td>RB6/P1C</td>
<td>bit 6</td>
<td>TTL/ST(2)</td>
<td>Input/output pin (with interrupt-on-change). Internal software programmable weak pull-up. PWM C output. Serial programming clock.</td>
</tr>
<tr>
<td>RB7/P1D</td>
<td>bit 7</td>
<td>TTL/ST(2)</td>
<td>Input/output pin (with interrupt-on-change). Internal software programmable weak pull-up. PWM D output. Serial programming data.</td>
</tr>
</tbody>
</table>

Legend: TTL = TTL input, ST = Schmitt Trigger input, XTAL = Crystal Oscillator input

Note 1: This buffer is a Schmitt Trigger input when configured as the external interrupt or peripheral input.

Note 2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.
### TABLE 3-4: SUMMARY OF REGISTERS ASSOCIATED WITH PORTB

<table>
<thead>
<tr>
<th>Address</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>Value on all other Resets</th>
</tr>
</thead>
<tbody>
<tr>
<td>06h</td>
<td>PORTB</td>
<td>RB7</td>
<td>RB6</td>
<td>RB5</td>
<td>RB4</td>
<td>RB3</td>
<td>RB2</td>
<td>RB1</td>
<td>RB0</td>
<td>xxxx xxxx</td>
<td>bbuu uuoo</td>
</tr>
<tr>
<td>86h</td>
<td>TRISB</td>
<td>PORTB Data Direction Register</td>
<td>1111 1111</td>
<td>1111 1111</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>81h</td>
<td>OPTION_REG</td>
<td>RBPU</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>
</tbody>
</table>

**Legend:**
- \( \times \) = unknown,
- \( \_ \) = unchanged.

Shaded cells are not used by PORTB.
4.0 TIMER0 MODULE

The Timer0 module timer/counter has the following features:
- 8-bit timer/counter
- Readable and writable
- Internal or external clock select
- Edge select for external clock
- 8-bit software programmable prescaler
- Interrupt on overflow from FFh to 00h

Figure 4-1 is a simplified block diagram of the Timer0 module.

Additional information on timer modules is available in the PICmicro® Mid-Range Reference Manual, (DS33023).

4.1 Timer0 Operation

Timer0 can operate as a timer or as a counter.

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

Counter mode is selected by setting bit T0CS (OPTION_REG<5>). In Counter mode, Timer0 will increment on every rising or falling edge of pin RA4/T0CKI. The incrementing edge is determined by the Timer0 Source Edge Select bit T0SE (OPTION_REG<4>). Clearing bit T0SE selects the rising edge. Restrictions on the external clock input are discussed below.

When an external clock input is used for Timer0, it must meet certain requirements. The requirements ensure the external clock can be synchronized with the internal phase clock (Tosc). Also, there is a delay in the actual incrementing of Timer0 after synchronization.

Additional information on external clock requirements is available in the PICmicro® Mid-Range Reference Manual, (DS33023).

4.2 Prescaler

An 8-bit counter is available as a prescaler for the Timer0 module or as a postscaler for the Watchdog Timer, respectively (Figure 4-2). For simplicity, this counter is being referred to as “prescaler” throughout this data sheet.

Note: There is only one prescaler available, which is mutually exclusively shared between the Timer0 module and Watchdog Timer. Thus, a prescaler assignment for the Timer0 module means that there is no prescaler for the Watchdog Timer and vice-versa.

The prescaler is not readable or writable.

The PSA and PS2:PS0 bits (OPTION_REG<3:0>) determine the prescaler assignment and prescaler ratio.

Clearing bit PSA will assign the prescaler to the Timer0 module. When the prescaler is assigned to the Timer0 module, prescale values of 1:2, 1:4, ..., 1:256 are selectable.

Setting bit PSA will assign the prescaler to the Watchdog Timer (WDT). When the prescaler is assigned to the WDT, prescale values of 1:1, 1:2, ..., 1:128 are selectable.

When assigned to the Timer0 module, all instructions writing to the TMR0 register (e.g. CLRF 1, MOVWF 1, BSF 1,x...etc.) will clear the prescaler. When assigned to WDT, a CLRWDT instruction will clear the prescaler along with the WDT.

Note: Writing to TMR0 when the prescaler is assigned to Timer0 will clear the prescaler count, but will not change the prescaler assignment.

Note: To achieve a 1:1 prescaler assignment for the TMR0 register, assign the prescaler to the Watchdog Timer.
4.2.1 SWITCHING PRESCALER ASSIGNMENT

The prescaler assignment is fully under software control (i.e., it can be changed “on-the-fly” during program execution).

**Note:** To avoid an unintended device Reset, a specific instruction sequence (shown in the PICmicro® Mid-Range Reference Manual, DS33023) must be executed when changing the prescaler assignment from Timer0 to the WDT. This sequence must be followed even if the WDT is disabled.

4.3 Timer0 Interrupt

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

**FIGURE 4-2: BLOCK DIAGRAM OF THE TIMER0/WDT PRESCALER**

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

<table>
<thead>
<tr>
<th>Address</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>Value on all other Resets</th>
</tr>
</thead>
<tbody>
<tr>
<td>01h</td>
<td>TMR0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>xxxx xxxx</td>
<td>uuuu uuuu</td>
</tr>
<tr>
<td>0Bh,8Bh</td>
<td>INTCON</td>
<td>GIE</td>
<td>PEIE</td>
<td>T0IE</td>
<td>INTE</td>
<td>RBIE</td>
<td>T0IF</td>
<td>INTF</td>
<td>RBIF</td>
<td>0000 000x</td>
<td>0000 0000</td>
</tr>
<tr>
<td>81h</td>
<td>OPTION_REG</td>
<td>RBPU</td>
<td>INTEG</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>85h</td>
<td>TRISA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>PSA</td>
<td>PS2</td>
<td>PS1</td>
<td>PS0</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 Timer0.

**Note 1:** Reserved bits, do not use.
5.0 TIMER1 MODULE

The Timer1 module timer/counter has the following features:
- 16-bit timer/counter (Two 8-bit registers; TMR1H and TMR1L)
- Readable and writable (Both registers)
- Internal or external clock select
- Interrupt on overflow from FFFFh to 0000h
- Reset from ECCP module trigger

Timer1 has a control register, shown in Register 5-1. Timer1 can be enabled/disabled by setting/clearing control bit TMR1ON (T1CON<0>).

Figure 5-1 is a simplified block diagram of the Timer1 module.

Additional information on timer modules is available in the PICmicro® Mid-Range Reference Manual, (DS33023).

5.1 Timer1 Operation

Timer1 can operate in one of these modes:
- As a timer
- As a synchronous counter
- As an asynchronous counter

The operating mode is determined by the clock select bit, TMR1CS (T1CON<1>).

In Timer mode, Timer1 increments every instruction cycle. In Counter mode, it increments on every rising edge of the external clock input.

When the Timer1 oscillator is enabled (T1OSCEN is set), the RB2/T1OSI and RB1/T1OSO/T1CKI pins become inputs. That is, the TRISB<2:1> value is ignored.

Timer1 also has an internal “Reset input”. This Reset can be generated by the ECCP module (Section 7.0 “Enhanced Capture/Compare/PWM (ECCP) Module”).

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

<table>
<thead>
<tr>
<th>Bit 7-6</th>
<th>Bit 5-4</th>
<th>Bit 3</th>
<th>Bit 2</th>
<th>Bit 1</th>
<th>Bit 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>—</td>
<td>—</td>
<td>T1CKPS1</td>
<td>T1CKPS0</td>
<td>T1OSCEN</td>
<td>T1SYNC</td>
</tr>
<tr>
<td>—</td>
<td>—</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
</tr>
</tbody>
</table>

Unimplemented: Read as ‘0’

T1CKPS1: T1CKPS0: Timer1 Input Clock Prescale Select bits
11 = 1:8 Prescale value
10 = 1:4 Prescale value
01 = 1:2 Prescale value
00 = 1:1 Prescale value

T1OSCEN: Timer1 Oscillator Enable Control bit
1 = Oscillator is enabled
0 = Oscillator is shut off

T1SYNC: Timer1 External Clock Input Synchronization Control bit
TMR1CS = 1
1 = Do not synchronize external clock input
0 = Synchronize external clock input

TMR1CS = 0
This bit is ignored. Timer1 uses the internal clock when TMR1CS = 0.

TMR1CS: Timer1 Clock Source Select bit
1 = External clock from pin RB1/T1OSO/T1CKI (on the rising edge after the first falling edge)
0 = Internal clock (Fosc/4)

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

Note 1: The oscillator inverter and feedback resistor are turned off to eliminate power drain.

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

© 2003 Microchip Technology Inc.
5.2 Timer1 Oscillator

A crystal oscillator circuit is built in between pins T1OSI (input) and T1OSO (amplifier output). It is enabled by setting control bit T1OSCEN (T1CON<3>). The oscillator is a low-power oscillator designed to operate with a 32.768 kHz tuning fork crystal. It will continue to run during Sleep.

The user must provide a software time delay to ensure proper oscillator start-up.

Note 1: When the T1OSCEN bit is cleared, the inverter and feedback resistor are turned off. This eliminates power drain.

5.3 Timer1 Interrupt

The TMR1 Register pair (TMR1H:TMR1L) increments from 0000h to FFFFh and rolls over to 0000h. The TMR1 interrupt, if enabled, is generated on overflow which is latched in interrupt flag bit TMR1IF (PIR1<0>). This interrupt can be enabled/disabled by setting/clearing TMR1 interrupt enable bit TMR1IE (PIE1<0>).

5.4 Resetting Timer1 using an ECCP Trigger Output

If the ECCP module is configured in Compare mode to generate a “special event trigger” (CCP1M3:CCP1M0 = 1011), this signal will reset Timer1 and start an A/D conversion (if the A/D module is enabled).

Note: The special event triggers from the ECCP module will not set interrupt flag bit TMR1IF (PIR1<0>).

Timer1 must be configured for either Timer or Synchronized Counter mode to take advantage of this feature. If Timer1 is running in Asynchronous Counter mode, this Reset operation may not work.

In the event that a write to Timer1 coincides with a special event trigger from the ECCP, the write will take precedence.

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

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

<table>
<thead>
<tr>
<th>Address</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>Value on all other Resets</th>
</tr>
</thead>
<tbody>
<tr>
<td>0Bh,8Bh</td>
<td>INTCON</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0000 000x</td>
<td>0000 000u</td>
</tr>
<tr>
<td>0Ch</td>
<td>PIR1</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—0— —000</td>
<td>—0— —000</td>
</tr>
<tr>
<td>8Ch</td>
<td>PIE1</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—0— —000</td>
<td>—0— —000</td>
</tr>
<tr>
<td>0Eh</td>
<td>TMR1L</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>xxxx xxxx</td>
<td>xxxxxxxxxx</td>
</tr>
<tr>
<td>0Fh</td>
<td>TMR1H</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>xxxxxxx</td>
<td>xxxxxxxxxx</td>
</tr>
<tr>
<td>10h</td>
<td>T1CON</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—00 0000</td>
<td>—00 0000</td>
</tr>
</tbody>
</table>

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

The Timer2 module timer has the following features:
  • 8-bit timer (TMR2 register)
  • 8-bit period register (PR2)
  • Readable and writable (both registers)
  • Software programmable prescaler (1:1, 1:4, 1:16)
  • Software programmable postscaler (1:1 to 1:16)
  • Interrupt on TMR2 match of PR2

Timer2 has a control register, shown in Register 6-1. Timer2 can be shut off by clearing control bit TMR2ON (T2CON<2>) to minimize power consumption.

Figure 6-1 is a simplified block diagram of the Timer2 module.

Additional information on timer modules is available in the PICmicro® Mid-Range Reference Manual, (DS33023).

REGISTER 6-1: T2CON: TIMER2 CONTROL REGISTER (ADDRESS 12h)

<table>
<thead>
<tr>
<th>bit 7</th>
<th>bit 6-3</th>
<th>bit 2</th>
<th>bit 1-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>—</td>
<td>TOUTPS3</td>
<td>TOUTPS2</td>
<td>TOUTPS1</td>
</tr>
<tr>
<td></td>
<td>TOUTPS0</td>
<td>TMR2ON</td>
<td>T2CKPS1</td>
</tr>
<tr>
<td></td>
<td>T2CKPS0</td>
<td></td>
<td></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

- 0000 = 1:1 Postscale
- 0001 = 1:2 Postscale
- 0010 = 1:3 Postscale
- 0011 = 1:4 Postscale
- 0100 = 1:5 Postscale
- 0101 = 1:6 Postscale
- 0110 = 1:7 Postscale
- 0111 = 1:8 Postscale
- 1000 = 1:9 Postscale
- 1001 = 1:10 Postscale
- 1010 = 1:11 Postscale
- 1011 = 1:12 Postscale
- 1100 = 1:13 Postscale
- 1101 = 1:14 Postscale
- 1110 = 1:15 Postscale
- 1111 = 1:16 Postscale

- 1 = Timer2 is on
- 0 = Timer2 is off

- 00 = Prescaler is 1
- 01 = Prescaler is 4
- 1x = Prescaler is 16
6.1 Timer2 Operation

Timer2 can be used as the PWM time base for PWM mode of the ECCP module.

The TMR2 register is readable and writable, and is cleared on any device Reset.

The input clock (FOSC/4) has a prescale option of 1:1, 1:4 or 1:16, selected by control bits T2CKPS1:T2CKPS0 (T2CON<1:0>).

The match output of TMR2 goes through a 4-bit postscaler (which gives a 1:1 to 1:16 scaling inclusive) to generate a TMR2 interrupt (latched in flag bit TMR2IF, (PIR1<1>).

The prescaler and postscaler counters are cleared when any of the following occurs:

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

TMR2 is not cleared when T2CON is written.

6.2 Timer2 Interrupt

The Timer2 module has an 8-bit Period register (PR2). Timer2 increments from 00h until it matches PR2 and then resets to 00h on the next increment cycle. PR2 is a readable and writable register. The PR2 register is initialized to FFh upon Reset.

### FIGURE 6-1: TIMER2 BLOCK DIAGRAM

![Timer2 Block Diagram](image)

### TABLE 6-1: REGISTERS ASSOCIATED WITH TIMER2 AS A TIMER/COUNTER

<table>
<thead>
<tr>
<th>Address</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>Value on all other Resets</th>
</tr>
</thead>
<tbody>
<tr>
<td>0Bh</td>
<td>GIE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0000 000x</td>
<td>0000 0000</td>
</tr>
<tr>
<td>0Ch</td>
<td>GIE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0000 000x</td>
<td>0000 0000</td>
</tr>
<tr>
<td>08h</td>
<td>T0IE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>-00-000</td>
<td>-00-000</td>
</tr>
<tr>
<td>09h</td>
<td>CCP1IF</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>-00-000</td>
<td>-00-000</td>
</tr>
<tr>
<td>0Ah</td>
<td>TMR1IF</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>-00-000</td>
<td>-00-000</td>
</tr>
<tr>
<td>0Bh</td>
<td>RBIF</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0000 000u</td>
<td>0000 0000</td>
</tr>
<tr>
<td>0Ch</td>
<td>TMR1IE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>-00-000</td>
<td>-00-000</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 000x</td>
<td>0000 0000</td>
</tr>
<tr>
<td>12h</td>
<td>TMR2ON</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>9Ch</td>
<td>T2CON</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0000 000x</td>
<td>0000 0000</td>
</tr>
<tr>
<td>92h</td>
<td>PR2</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:**

- `x` = unknown
- `u` = unchanged
- `~` = unimplemented read as ‘0’

Shaded cells are not used by the Timer2 module.
7.0 ENHANCED CAPTURE/COMPARE/PWM (ECCP) MODULE

The ECCP (Enhanced Capture/Compare/PWM) module contains a 16-bit register, which can operate as:

- 16-bit Capture register
- 16-bit Compare register
- PWM Master/Slave Duty Cycle register

Table 7-1 shows the timer resources of the ECCP module modes.

Capture/Compare/PWM Register 1 (CCPR1) is comprised of two 8-bit registers: CCPR1L (low byte) and CCPR1H (high byte).

REGISTER 7-1: CCP1CON REGISTER (ADDRESS: 17h)

<table>
<thead>
<tr>
<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>
</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>

- **bit 7-6**: P1M1: P1M0: PWM Output Configuration bits
  - CCP1M<3:2> = 00, 01, 10
  - 00 = Single output, P1A modulated. P1B, P1C, P1D assigned as Port pins.
  - 01 = Quad output forward. P1D modulated, P1A active. P1B and P1C inactive.
  - 10 = Quad output reverse. P1B modulated, P1C active. P1A and P1D inactive.

- **bit 5-4**: DC1B1:DC1B0: PWM 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**: CCP1M3:CCP1M0: ECCP Mode Select bits
  - 0000 = Compare/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 CCP1 output on match (CCP1IF bit is set)
  - 1001 = Compare mode, clear CCP1 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, TMR1 is reset, and an A/D conversion is started if the A/D module is enabled. CCP1 pin is unaffected).
  - 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.

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

The CCP1CON register controls ECCP operation. All the CCP1CON bits are readable and writable.

Additional information on the ECCP module is available in the PICmicro® Mid-Range Reference Manual (DS33023).

<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>

Table 7-1: ECCP Mode - Timer Resource
7.1 Capture Mode

In Capture mode, CCPR1H:CCPR1L captures the 16-bit value of the TMR1 register when an event occurs on pin RB3/CCP1/P1A. An event is defined as:

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

An event is selected by control bits CCP1M3:CCP1M0 (CCP1CON<3:0>). When a capture is made, the interrupt request flag bit CCP1IF (PIR1<2>) is set. It must be cleared in software. If another capture occurs before the value in register CCPR1 is read, the old captured value will be lost.

Note: Always reset the ECCP module (CCP1M3:CCP1M0 = '0000') between changing from one capture mode to another. This is necessary to reset the internal capture counter.

7.1.1 CCP1 PIN CONFIGURATION

In Capture mode, the RB3/CCP1/P1A pin should be configured as an input by setting the TRISB<3> bit.

Note: If the RB3/CCP1/P1A is configured as an output, a write to PORTB can cause a capture condition.

7.1.2 TIMER1 MODE SELECTION

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

7.1.3 SOFTWARE INTERRUPT

When the Capture mode is changed, a false capture interrupt may be generated. The user should keep bit CCP1IE (PIE1<2>) clear to avoid false interrupts and should clear the flag bit CCP1IF following any such change in operating mode.

7.1.4 ECCP PRESCALER

There are four prescaler settings, specified by bits CCP1M3:CCP1M0. Whenever the ECCP module is turned off, or the ECCP module is not in Capture mode, the prescaler counter is cleared. This means that any Reset will clear the prescaler counter.

Switching from one capture prescaler to another may generate an interrupt. Also, the prescaler counter will not be cleared, therefore, the first capture may be from a non-zero prescaler. Example 7-1 shows the recommended method for switching between capture prescalers. This example also clears the prescaler counter and will not generate the "false" interrupt.

**EXAMPLE 7-1: CHANGING BETWEEN CAPTURE PRESCALERS**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLRF CCP1CON</td>
<td>Turn ECCP module off</td>
</tr>
<tr>
<td>MOVWF CCP1CON</td>
<td>Load CCP1CON with this value</td>
</tr>
<tr>
<td>MOVLW NEW_CAPT_PS MOVWF CCP1CON</td>
<td>Load the W reg with the new prescaler mode value and ECCP ON</td>
</tr>
</tbody>
</table>

7.2 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 RB3/CCP1/P1A pin is either:

- Driven High
- Driven Low
- Toggle output (high-to-low or low-to-high)
- Remains Unchanged

The action on the pin is based on the value of control bits CCP1M3:CCP1M0 (CCP1CON<3:0>). At the same time, interrupt flag bit CCP1IF is set.

Changing the ECCP mode to clear output on match (CCP1M<3:0> = 1000) presets the CCP1 output latch to the logic 1 level. Changing the ECCP mode to set output on match (CCP1M<3:0> = 1001) presets the CCP1 output latch to the logic 0 level.
7.2.1 CCP1 PIN CONFIGURATION

The user must configure the RB3/CCP1/P1A pin as the CCP1 output by clearing the TRISB<3> bit.

**Note:** Clearing the CCP1CON register will force the RB3/CCP1/P1A compare output latch to the default low level. This is not the PORTB I/O data latch.

7.2.2 TIMER1 MODE SELECTION

Timer1 must be running in Timer mode or Synchronized Counter mode if the ECCP module is using the compare feature. In Asynchronous Counter mode, the compare operation may not work.

7.2.3 SOFTWARE INTERRUPT MODE

When Generate Software Interrupt mode is chosen, the CCP1 pin is not affected. Only a CCP interrupt is generated (if enabled).

7.2.4 SPECIAL EVENT TRIGGER

In this mode, an internal hardware trigger is generated which may be used to initiate an action.

The special event trigger output of the ECCP resets the TMR1 register pair. This allows the CCP1 register to effectively be a 16-bit programmable period register for Timer1.

The special event trigger output of the ECCP also starts an A/D conversion (if the A/D module is enabled).

### FIGURE 7-2: COMPARE MODE OPERATION BLOCK DIAGRAM

![Compare Mode Operation Block Diagram](image)

**Note:** The special event trigger from the ECCP module will not set interrupt flag bit TMR1IF (PIR1<0>).

### TABLE 7-2: REGISTERS ASSOCIATED WITH CAPTURE, COMPARE, AND TIMER1

<table>
<thead>
<tr>
<th>Address</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>Value on all other Resets</th>
</tr>
</thead>
<tbody>
<tr>
<td>0Bh,8Bh</td>
<td>iNTCON</td>
<td>GIE</td>
<td>PEIE</td>
<td>TOIE</td>
<td>INTE</td>
<td>RBIE</td>
<td>TOIF</td>
<td>INTF</td>
<td>RBIF</td>
<td>0000 00xx</td>
<td>0000 000u</td>
</tr>
<tr>
<td>0Ch</td>
<td>PIR1</td>
<td>—</td>
<td>ADIF</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>CCP1IF</td>
<td>TMR2IF</td>
<td>—0-- -000</td>
<td>—0-- -000</td>
</tr>
<tr>
<td>0Eh</td>
<td>TMR1L</td>
<td>Holding register for the Least Significant Byte of the 16-bit TMR1 register</td>
<td>xxxx xxxx</td>
<td>uuuu uuuu</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>uuuu uuuu</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10h</td>
<td>T1CON</td>
<td>—</td>
<td>—</td>
<td>T1CKPS1</td>
<td>T1CKPS0</td>
<td>T1OSCEN</td>
<td>T1SYNC</td>
<td>TMR1CS</td>
<td>TMR1ON</td>
<td>--00 0000</td>
<td>--uu uuuu</td>
</tr>
<tr>
<td>15h</td>
<td>CCP1RL</td>
<td>Capture/Compare/PWM register1 (LSB)</td>
<td>xxxx xxxx</td>
<td>uuuu uuuu</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16h</td>
<td>CCP1RH</td>
<td>Capture/Compare/PWM register1 (MSB)</td>
<td>xxxx xxxx</td>
<td>uuuu uuuu</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17h</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>0000 0000</td>
</tr>
<tr>
<td>86h</td>
<td>TRISB</td>
<td>PORTB Data direction register</td>
<td>1111 1111</td>
<td>1111 1111</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8Ch</td>
<td>PIE1</td>
<td>—</td>
<td>ADIE</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>CCP1IE</td>
<td>TMR2IE</td>
<td>—0-- -000</td>
<td>—0-- -000</td>
</tr>
</tbody>
</table>

**Legend:** x = unknown, u = unchanged, – = unimplemented read as ‘0’. Shaded cells are not used by Capture and Timer1.
7.3 PWM Mode

In Pulse Width Modulation (PWM) mode, the CCP1 pin produces up to a 10-bit resolution PWM output. Since the CCP1 pin is multiplexed with the PORTB data latch, the TRISB<3> bit must be cleared to make the CCP1 pin an output.

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

Figure 7-3 shows a simplified block diagram of the CCP module in PWM mode.

For a step-by-step procedure on how to setup the ECCP module for PWM operation, see Section 7.3.3 “Set-Up for PWM Operation”.

A PWM output (Figure 7-4) has a time base (period) and a time that the output stays high (duty cycle). The frequency of the PWM is the inverse of the period (1/period).

**FIGURE 7-4: PWM OUTPUT**

![PWM Diagram](image)

**FIGURE 7-3: SIMPLIFIED PWM BLOCK DIAGRAM**

**7.3.1 PWM PERIOD**

The PWM period is specified by writing to the PR2 register. The PWM period can be calculated using the following formula:

**EQUATION 7-1:**

\[
\text{PWM Period} = \frac{1}{(\text{PR2} + 1) \times 4 \times \text{Tosc} \times (\text{TMR2 prescale value})}
\]

PWM frequency is defined as 1/[PWM period].

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 PWM duty cycle = 0%, the CCP1 pin will not be set)
- The PWM duty cycle is latched from CCPR1L into CCPR1H

Note: The Timer2 postscaler (see Section 6.0 “Timer2 Module”) is not used in the determination of the PWM frequency. The postscaler could be used to have a servo update rate at a different frequency than the PWM output.
7.3.2 PWM DUTY CYCLE

The PWM duty cycle is specified by writing to the CCPR1L register and to the CCP1CON<5:4> bits. Up to 10-bit resolution is available. The CCPR1L contains the eight MSbs and the CCP1CON<5:4> contains the two LSbs. This 10-bit value is represented by CCPR1L:CCP1CON<5:4>. The following equation is used to calculate the PWM duty cycle in time:

\[
\text{EQUATION 7-2:} \quad \text{PWM Duty Cycle} = (\text{CCPR1L}:\text{CCP1CON}<5:4> \cdot \frac{T_{\text{OSC}}}{\text{TMR2 prescale value}})
\]

CCPR1L and CCP1CON<5:4> can be written to at any time, but the duty cycle value is not latched into CCPR1H until a match between PR2 and TMR2 occurs (i.e., the period is complete). In PWM mode, CCPR1H is a read-only register.

The CCP1H 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. When the CCP1H and 2-bit latch match TMR2 concatenated with an internal 2-bit Q clock or 2 bits of the TMR2 prescaler, the CCP1 pin is cleared.

Maximum PWM resolution (bits) for a given PWM frequency is given by the following equation:

\[
\text{EQUATION 7-3:} \quad \text{Max resolution} = \log \left( \frac{\text{F}_{\text{OSC}}}{\text{F}_{\text{PWM}}} \right) \text{ bits}
\]

Note: If the PWM duty cycle value is longer than the PWM period the CCP1 pin will not be cleared.

For an example PWM period and duty cycle calculation, see the PICmicro® Mid-Range Reference Manual, (DS33023).

7.3.3 SET-UP FOR PWM OPERATION

The following steps should be taken when configuring the ECCP module for PWM operation:

1. Set the PWM period by writing to the PR2 register.
2. Set the PWM duty cycle by writing to the CCPR1L register and CCP1CON<5:4> bits.
3. Make the CCP1 pin an output by clearing the TRISB<3> bit.
4. Set the TMR2 prescale value and enable Timer2 by writing to T2CON.
5. Configure the CCP1 module for PWM operation.

### TABLE 7-3: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS AT 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 Prescaler (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 7-4: REGISTERS ASSOCIATED WITH PWM AND TIMER2

<table>
<thead>
<tr>
<th>Address</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>Value on all other Resets</th>
</tr>
</thead>
<tbody>
<tr>
<td>0Bh,8Bh</td>
<td>INTCON</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0000 000x 0000 000u</td>
<td></td>
</tr>
<tr>
<td>0Ch</td>
<td>PIR1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0000 000x 0000 000u</td>
<td></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 0000 0000</td>
<td></td>
</tr>
<tr>
<td>12h</td>
<td>T2CON</td>
<td></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>15h</td>
<td>CCP1L</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>xxxx xxxx xxxx xxxx</td>
<td></td>
</tr>
<tr>
<td>16h</td>
<td>CCP1H</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>xxxx xxxx xxxx xxxx</td>
<td></td>
</tr>
<tr>
<td>17h</td>
<td>CCP1CON</td>
<td></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>86h</td>
<td>TRISB</td>
<td></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>8Ch</td>
<td>PIE1</td>
<td></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>92h</td>
<td>PR2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1111 1111 1111 1111</td>
<td></td>
</tr>
</tbody>
</table>

Legend: \( x = \text{unknown}, \ u = \text{unchanged}, \ - = \text{unimplemented read as '0'} \). Shaded cells are not used by PWM and Timer2.
7.4 ENHANCED PWM MODE

The Enhanced PWM mode provides additional PWM output options for a broader range of control applications. The module is an upwardly compatible version of the standard CCP module and offers up to four outputs, designated P1A through P1D. Users are also able to select the polarity of the signal (either active-high or active-low). The module's Output mode and polarity are configured by setting the P1M1:P1M0 and CCP1M3:CCP1M0 bits of the CCP1CON register (CCP1CON<7:6> and CCP1CON<3:0>, respectively).

Figure 7-5 shows a simplified block diagram of PWM operation. All control registers are double buffered and are loaded at the beginning of a new PWM cycle (the period boundary when Timer2 resets) in order to prevent glitches on any of the outputs. The exception is the PWM Dead-band Delay, which is loaded at either the duty cycle boundary or the boundary period (whichever comes first). Because of the buffering, the module waits until the assigned timer resets, instead of starting immediately. This means that enhanced PWM waveforms do not exactly match the standard PWM waveforms, but are instead offset by one full instruction cycle (4 Tosc).

As before, the user must manually configure the appropriate TRISB bits for output.

7.4.1 PWM OUTPUT CONFIGURATIONS

The P1M1:P1M0 bits in the CCP1CON register allows one of four configurations:
- Single Output
- Half-Bridge Output
- Full-Bridge Output Forward mode
- Full-Bridge Output Reverse mode

The Single Output mode is the Standard PWM mode discussed in Section 7.3 “PWM Mode”. The Half-Bridge and Full-Bridge Output modes are covered in detail in the sections that follow.

The general relationship of the outputs in all configurations is summarized in Figure 7-6.

---

FIGURE 7-5: SIMPLIFIED BLOCK DIAGRAM OF THE ENHANCED PWM MODULE

---

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.
FIGURE 7-6: PWM OUTPUT RELATIONSHIPS (P1A, P1B, P1C, P1D ACTIVE-HIGH STATE)

<table>
<thead>
<tr>
<th>CCP1CON &lt;7:6&gt;</th>
<th>SIGNAL</th>
<th>0</th>
<th>Duty Cycle</th>
<th>Period</th>
</tr>
</thead>
<tbody>
<tr>
<td>00 (Single Output)</td>
<td>P1A Modulated</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>P1A Modulated</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10 (Half-Bridge)</td>
<td>P1B Modulated</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>P1A Active</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>P1B Inactive</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>P1C Inactive</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>P1D Modulated</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>01 (Full-Bridge, Forward)</td>
<td>P1B Inactive</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>P1C Inactive</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>P1D Modulated</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>P1A Inactive</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11 (Full-Bridge, Reverse)</td>
<td>P1B Modulated</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>P1C Active</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>P1D Inactive</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Relationships:
- Period = 4 * Tosc * (PR2 + 1) * (TMR2 prescale value)
- Duty Cycle = 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 7.4.4 “Programmable Dead-Band Delay”).

FIGURE 7-7: PWM OUTPUT RELATIONSHIPS (P1A, P1B, P1C, P1D ACTIVE-LOW STATE)

<table>
<thead>
<tr>
<th>CCP1CON &lt;7:6&gt;</th>
<th>SIGNAL</th>
<th>0</th>
<th>Duty Cycle</th>
<th>Period</th>
</tr>
</thead>
<tbody>
<tr>
<td>00 (Single Output)</td>
<td>P1A Modulated</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>P1A Modulated</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10 (Half-Bridge)</td>
<td>P1B Modulated</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>P1A Active</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>P1B Inactive</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>P1C Inactive</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>P1D Modulated</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>01 (Full-Bridge, Forward)</td>
<td>P1B Inactive</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>P1C Inactive</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>P1D Modulated</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>P1A Inactive</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11 (Full-Bridge, Reverse)</td>
<td>P1B Modulated</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>P1C Active</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>P1D Inactive</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Relationships:
- Period = 4 * Tosc * (PR2 + 1) * (TMR2 prescale value)
- Duty Cycle = 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 7.4.4 “Programmable Dead-Band Delay”).
FIGURE 7-8: PWM OUTPUT RELATIONSHIPS (P1A, P1C ACTIVE-HIGH. P1B, P1D ACTIVE-LOW)

<table>
<thead>
<tr>
<th>CCP1CON &lt;7:6&gt;</th>
<th>SIGNAL</th>
<th>0</th>
<th>Period</th>
<th>PR2+1</th>
</tr>
</thead>
<tbody>
<tr>
<td>00 (Single Output)</td>
<td>P1A Modulated</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>P1B Modulated</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10 (Half-Bridge)</td>
<td>P1A Modulated</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>P1B Modulated</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>01 (Full-Bridge, Forward)</td>
<td>P1A Active</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>P1B Inactive</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>P1C Inactive</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>P1D Modulated</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11 (Full-Bridge, Reverse)</td>
<td>P1A Inactive</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>P1B Modulated</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>P1C Active</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>P1D Inactive</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
FIGURE 7-9: PWM OUTPUT RELATIONSHIPS (P1A, P1C ACTIVE-LOW. P1B, P1D ACTIVE-HIGH)

<table>
<thead>
<tr>
<th>CCP1CON &lt;7:6&gt;</th>
<th>SIGNAL</th>
<th>PR2+1</th>
</tr>
</thead>
<tbody>
<tr>
<td>00 (Single Output)</td>
<td>P1A Modulated</td>
<td></td>
</tr>
<tr>
<td>10 (Half-Bridge)</td>
<td>P1B Modulated</td>
<td></td>
</tr>
<tr>
<td>01 (Full-Bridge, Forward)</td>
<td>P1A Active</td>
<td></td>
</tr>
<tr>
<td></td>
<td>P1B Inactive</td>
<td></td>
</tr>
<tr>
<td></td>
<td>P1C Inactive</td>
<td></td>
</tr>
<tr>
<td></td>
<td>P1D Modulated</td>
<td></td>
</tr>
<tr>
<td>11 (Full-Bridge, Reverse)</td>
<td>P1A Inactive</td>
<td></td>
</tr>
<tr>
<td></td>
<td>P1B Modulated</td>
<td></td>
</tr>
<tr>
<td></td>
<td>P1C Active</td>
<td></td>
</tr>
<tr>
<td></td>
<td>P1D Inactive</td>
<td></td>
</tr>
</tbody>
</table>

Relationships:
- Period = 4 * Tosc * (PR2 + 1) * (TMR2 prescale value)
- Duty Cycle = 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 7.4.4 “Programmable Dead-Band Delay”).
7.4.2 HALF-BRIDGE MODE

In the Half-Bridge Output mode, two pins are used as outputs to drive push-pull loads. The PWM output signal is output on the RB3/CCP1/P1A pin, while the complementary PWM output signal is output on the RB5/P1B pin (Figure 7-12). This mode can be used for half-bridge applications, as shown in Figure 7-11 or for full-bridge applications, where four power switches are being modulated with two PWM signals.

In Half-Bridge Output mode, the programmable dead-band delay can be used to prevent shoot-through current in half-bridge power devices. The value of PWM1CON bits PDC6:PDC0 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 7.4.4 “Programmable Dead-Band Delay” for more details of the dead-band delay operations.

Since the P1A and P1B outputs are multiplexed with the PORTB<3> and PORTB<5> data latches, the TRISB<3> and TRISB<5> bits must be cleared to configure P1A and P1B as outputs.

**FIGURE 7-10:** HALF-BRIDGE PWM OUTPUT

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

**FIGURE 7-11:** EXAMPLES OF HALF-BRIDGE OUTPUT MODE APPLICATIONS

- **Standard Half-Bridge Circuit (“Push-Pull”)**

  - PIC16F716
  - P1A
  - P1B
  - FET Driver
  - Load
  - V+
  - V-

- **Half-Bridge Output Driving a Full-Bridge Circuit**

  - PIC16F716
  - P1A
  - P1B
  - FET Driver
  - Load
  - FET Driver
  - V+
  - V-
7.4.3 FULL-BRIDGE MODE

In Full-Bridge Output mode, four pins are used as outputs; however, only two outputs are active at a time. In the Forward mode, pin RB3/CCP1/P1A is continuously active, and pin RB7/P1D is modulated. In the Reverse mode, RB6/P1C pin is continuously active, and RB5/P1B pin is modulated. These are illustrated in Figure 7-6 through Figure 7-9.

FIGURE 7-12: EXAMPLE OF FULL-BRIDGE APPLICATION

7.4.3.1 Direction Change in Full-Bridge Mode

In the Full-Bridge Output mode, the P1M1 bit in the CCP1CON register allows the user to control the Forward/Reverse direction. When the application firmware changes this direction control bit, the module will assume the new direction on the next PWM cycle.

Just before the end of the current PWM period, the modulated outputs (P1B and P1D) are placed in their inactive state, while the unmodulated outputs (P1A and P1C) are switched to drive in the opposite direction. This occurs in a time interval of \(4 \times T_{OSC} \times (\text{Timer2 Prescale value})\) before the next PWM period begins. The Timer2 prescaler will be either 1, 4 or 16, depending on the value of the T2CKPSx bits (T2CON<1:0>). During the interval from the switch of the unmodulated outputs to the beginning of the next period, the modulated outputs (P1B and P1D) remain inactive. This relationship is shown in Figure 7-13.

Note: In the Full-Bridge Output mode, the ECCP module does not provide any dead-band delay. In general, since only one output is modulated at all times, dead-band delay is not required. However, there is a situation where a dead-band delay might be 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.

P1A, P1B, P1C and P1D outputs are multiplexed with the PORTB<3> and PORTB<5:7> data latches. The TRISB<3> and TRISB<5:7> bits must be cleared to make the P1A, P1B, P1C, and P1D pins output.
Figure 7-14 shows an example where the PWM direction changes from forward to reverse, at a near 100% duty cycle. At time t1, the output P1A and P1D become inactive, while output P1C becomes active. In this example, since the turn-off time of the power devices is longer than the turn-on time, a shoot-through current may flow through power devices QC and QD (see Figure 7-12) 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, one of the following requirements must be met:

1. Reduce PWM for a 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 7-13: PWM DIRECTION CHANGE**

<table>
<thead>
<tr>
<th>SIGNAL</th>
<th>Forward Period</th>
<th>t1</th>
<th>Reverse Period</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1A (Active-High)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P1B (Active-High)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P1C (Active-High)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P1D (Active-High)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note 1:** The direction bit in the CCP1 Control Register (CCP1CON<7>) is written any time during the PWM cycle.

**Note 2:** When changing directions, the P1A and P1C signals switch before the end of the current PWM cycle at intervals of 4 Tosc, 16 Tosc or 64 Tosc, depending on the Timer2 prescaler value. The modulated P1B and P1D signals are inactive at this time.

**FIGURE 7-14: PWM DIRECTION CHANGE AT NEAR 100% DUTY CYCLE**

<table>
<thead>
<tr>
<th>SIGNAL</th>
<th>Forward Period</th>
<th>t1</th>
<th>Reverse Period</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1A</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P1B</td>
<td></td>
<td></td>
<td>DC</td>
</tr>
<tr>
<td>P1C</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P1D</td>
<td>DC</td>
<td></td>
<td>t_on</td>
</tr>
<tr>
<td>External Switch C</td>
<td></td>
<td></td>
<td>t_off</td>
</tr>
<tr>
<td>External Switch D</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Potential Shoot-Through Current</td>
<td></td>
<td></td>
<td>t = t_off - t_on</td>
</tr>
</tbody>
</table>

**Note 1:** All signals are shown as active-high.

**Note 2:** t_on is the turn-on delay of power switch QC and its driver.

**Note 3:** t_off is the turn-off delay of power switch QD and its driver.
7.4.4 PROGRAMMABLE DEAD-BAND DELAY

In half-bridge applications where all power switches are modulated at the PWM frequency at all times, 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) may 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 the Half-Bridge Output 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 Figure 7-10 for illustration. The lower seven bits of the PWM1CON register (Register 7-2) sets the delay period in terms of microcontroller instruction cycles (TCY or 4 Tosc).

7.4.5 ENHANCED PWM AUTO-SHUTDOWN

When the ECCP is programmed for any of the enhanced PWM modes, the active output pins may be configured for auto-shutdown. Auto-shutdown immediately places the enhanced PWM output pins into a defined shutdown state when a shutdown event occurs.

A shutdown event can be caused by a logic low level on either or both of the RB0/INT/ECCPAS2 or RB4/ECCPAS0 pins. The auto-shutdown feature can be disabled by not selecting any auto-shutdown sources. The auto-shutdown sources to be used are selected using the ECCPAS2 and ECCPAS0 bits (ECCPAS<6> and ECCPAS<4>);

When a shutdown occurs, the output pins are asynchronously placed in their shutdown states, specified by the PSSAC1:PSSAC0 and PSSBD1:PSSBD0 bits (ECCPAS<3:0>). Each pin pair (P1A/P1C and P1B/P1D) may be set to drive high, drive low, or be tri-stated (not driving). The ECCPASE bit (ECCPAS<7>) is also set to hold the enhanced PWM outputs in their shutdown states.

The ECCPASE bit is set by hardware when a shutdown event occurs. If automatic restarts are not enabled, the ECCPASE bit must be cleared by firmware when the cause of the shutdown clears. If automatic restarts are enabled, the ECCPASE bit is automatically cleared when the cause of the auto-shutdown has cleared.

If the ECCPASE bit is set when a PWM period begins, the PWM outputs remain in their shutdown state for that entire PWM period. When the ECCPASE bit is cleared, the PWM outputs will return to normal operation at the beginning of the next PWM period.

Note: Writing to the ECCPASE bit is disabled while a shutdown condition is active.
### REGISTER 7-2: PWM1CON: PWM CONFIGURATION REGISTER (ADDRESS: 18h)

<table>
<thead>
<tr>
<th>bit 7</th>
<th>PRSEN: PWM Restart Enable bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Upon auto-shutdown, the ECCPASE bit clears automatically once the shutdown event goes away; the PWM restarts automatically.</td>
</tr>
<tr>
<td>0</td>
<td>Upon auto-shutdown, ECCPASE must be cleared in firmware to restart the PWM.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>bit 6-0</th>
<th>PDC&lt;6:0&gt;: PWM Delay Count bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of $\frac{FOSC}{4 (4\times TOsc)}$ cycles between the scheduled time when a PWM signal should transition active, and the actual time it transitions active.</td>
<td></td>
</tr>
</tbody>
</table>

### REGISTER 7-3: ECCPAS – ENHANCED CCP AUTO SHUT DOWN REGISTER (ADDRESS: 19h)

<table>
<thead>
<tr>
<th>bit 7</th>
<th>ECCPASE: ECCP Auto-Shutdown Event Status bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A shutdown event has occurred. Must be reset in firmware to re-enable ECCP if PRSEN = 0</td>
</tr>
<tr>
<td>0</td>
<td>ECCP outputs enabled, no shutdown event</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>bit 6</th>
<th>ECCPAS2: ECCP Auto-Shutdown bit 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RB0 (INT) pin low level ('0') causes shutdown</td>
</tr>
<tr>
<td>0</td>
<td>RB0 (INT) pin has no effect on ECCP</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>bit 5</th>
<th>Unimplemented: Read as ‘0’</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>bit 4</th>
<th>ECCPAS0: ECCP Auto-Shutdown bit ‘0’</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RB4 pin low level ('0') causes shutdown</td>
</tr>
<tr>
<td>0</td>
<td>RB4 pin has no effect on ECCP</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>bit 3-2</th>
<th>PSSAC&lt;1:0&gt;: Pin P1A and P1C Shutdown State Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>Drive Pins P1A and P1C to ‘0’</td>
</tr>
<tr>
<td>01</td>
<td>Drive Pins P1A and P1C to ‘1’</td>
</tr>
<tr>
<td>1x</td>
<td>Pins P1A and P1C tri-state</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>bit 1-0</th>
<th>PSSBD&lt;1:0&gt;: Pin P1B and P1D Shutdown State Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>Drive Pins P1B and P1D to ‘0’</td>
</tr>
<tr>
<td>01</td>
<td>Drive Pins P1B and P1D to ‘1’</td>
</tr>
<tr>
<td>1x</td>
<td>Pins P1B and P1D tri-state</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
7.4.5.1 Auto-Shutdown and Automatic Restart

The auto-shutdown feature can be configured to allow automatic restarts of the module following a shutdown event. This is enabled by setting the PRSEN bit of the PWM1CON register (PWM1CON<7>).

In Shutdown mode with PRSEN = 1 (PWM1CON<7>) (Figure 7-15), the ECCPASE bit will remain set for as long as the cause of the shutdown continues. When the shutdown condition clears, the ECCPASE bit is cleared. If PRSEN = 0 (Figure 7-16), once a shutdown condition occurs, the ECCPASE bit will remain set until it is cleared by firmware. Once ECCPASE is cleared, the enhanced PWM will resume at the beginning of the next PWM period.

**Note:** Writing to the ECCPASE bit is disabled while a shutdown condition is active.

The ECCPASE bit cannot be cleared as long as the cause of the shutdown persists.

The Auto-shutdown mode can be forced by writing a ‘1’ to the ECCPASE bit.

7.4.6 START-UP CONSIDERATIONS

When the ECCP module is used in the PWM mode, the application hardware must use the proper external pull-up and/or pull-down resistors on the PWM output pins. 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 CCP1M1:CCP1M0 bits (CCP1CON<1:0>) 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 pins are configured as outputs. Changing the polarity configuration while the PWM pins are configured as outputs 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 pins for output at the same time as the ECCP module may cause damage to the application circuit. The ECCP module must be enabled in the proper Output mode and complete a full PWM cycle before configuring the PWM pins as outputs. The completion of a full PWM cycle is indicated by the TMR2IF bit being set as the second PWM period begins.
7.4.7 SETUP FOR PWM OPERATION
The following steps should be taken when configuring the ECCP module for PWM operation:

1. Configure the PWM pins P1A and P1B (and P1C and P1D, if used) as inputs by setting the corresponding TRISB bits.
2. Set the PWM period by loading the PR2 register.
3. Configure the ECCP module for the desired PWM mode and configuration by loading the CCP1CON register with the appropriate values:
   - Select one of the available output configurations and direction with the P1M1:P1M0 bits.
   - Select the polarities of the PWM output signals with the CCP1M3:CCP1M0 bits.
4. Set the PWM duty cycle by loading the CCPR1L register and CCP1CON<5:4> bits.
5. For Half-Bridge Output mode, set the dead-band delay by loading PWM1CON<6:0> with the appropriate value.
6. If auto-shutdown operation is required, load the ECCPAS register.
   - Select the auto-shutdown sources using the ECCPAS<2> AND ECCPAS<0> bits.
   - Select the shutdown states of the PWM output pins using PSSAC1:PSSAC0 and PSSBD1:PSSBD0 bits.
   - Set the ECCPASE bit (ECCPAS<7>).
7. If auto-restart operation is required, set the PRSEN bit (PWM1CON<7>).

8. Configure and start TMR2:
   - Clear the TMR2 interrupt flag bit by clearing the TMR2IF bit (PIR1<1>).
   - Set the TMR2 prescale value by loading the T2CKPSx bits (T2CON<1:0>).
   - Enable Timer2 by setting the TMR2ON bit (T2CON<2>).
9. Enable PWM outputs after a new PWM cycle has started:
   - Wait until TMR2 overflows (TMR2IF bit is set).
   - Enable the CCP1/P1A, P1B, P1C and/or P1D pin outputs by clearing the respective TRISB bits.
   - Clear the ECCPASE bit (ECCPAS<7>).

See the previous section for additional details.

7.4.8 EFFECTS OF A RESET
Both Power-on and subsequent Resets will force all ports to Input mode and the ECCP registers to their Reset states.
This forces the Enhanced CCP module to reset to a state compatible with the standard ECCP module.

### TABLE 7-5: REGISTERS ASSOCIATED WITH ENHANCED PWM AND TIMER2

<table>
<thead>
<tr>
<th>Address</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>Value on all other Resets</th>
</tr>
</thead>
<tbody>
<tr>
<td>0Bh</td>
<td>INTCON</td>
<td>GIE</td>
<td>PEIE</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>0000 000x</td>
<td>0000 000u</td>
</tr>
<tr>
<td>0Ch</td>
<td>PIR1</td>
<td>—</td>
<td>ADIF</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>8Ch</td>
<td>PIE1</td>
<td>—</td>
<td>ADIE</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</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>0000 0000</td>
</tr>
<tr>
<td>92h</td>
<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>12h</td>
<td>T2CON</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>16h</td>
<td>CCR1H</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>15h</td>
<td>CCR1L</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>17h</td>
<td>CCP1CON</td>
<td>P1M1</td>
<td>P1M0</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>19h</td>
<td>ECCPAS</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>18h</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>—</td>
<td>—</td>
</tr>
</tbody>
</table>

Legend:  
- \( x \) = unknown, \( u \) = unchanged, \( - \) = unimplemented, read as \( '0' \). Shaded cells are not used by the ECCP module in enhanced PWM mode.
8.0 ANALOG-TO-DIGITAL CONVERTER (A/D) MODULE

The Analog-to-Digital (A/D) Converter module has four inputs.

The A/D allows conversion of an analog input signal to a corresponding 8-bit digital number (refer to Application Note AN546 for use of A/D Converter). The output of the sample and hold is the input into the converter, which generates the result via successive approximation. The analog reference voltage is software selectable to either the device's positive supply voltage (VDD) or the voltage level on the RA3/AN3/VREF pin.

The A/D converter has a unique feature of being able to operate while the device is in Sleep mode. To operate in Sleep, the A/D conversion clock must be derived from the A/D’s internal RC oscillator.

Additional information on the A/D module is available in the PICmicro® Mid-Range Reference Manual, (DS33023).

The A/D module has three registers. These registers are:

- A/D Result Register (ADRES)
- A/D Control Register 0 (ADCON0)
- A/D Control Register 1 (ADCON1)

A device Reset forces all registers to their Reset state. This forces the A/D module to be turned off and any conversion is aborted.

The ADCON0 register, shown in Register 8-1, controls the operation of the A/D module. The ADCON1 register, shown in Register 8-2, configures the functions of the port pins. The port pins can be configured as analog inputs (RA3 can also be a voltage reference) or as digital I/O.

REGISTER 8-1: ADCON0 REGISTER (ADDRESS: 1Fh)

<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>ADSC1</td>
<td>ADCS0</td>
<td>CHS2</td>
<td>CHS1</td>
<td>CHS0</td>
<td>GO/DONE</td>
<td>—</td>
<td>ADON</td>
</tr>
</tbody>
</table>

bit 7-6  
ADCS1:ADCS0: A/D Conversion Clock Select bits
00 = Fosc/2
01 = Fosc/8
10 = Fosc/32
11 = FRC (Clock derived from the internal ADC RC oscillator)

bit 5-3  
CHS2:CHS0: Analog Channel Select bits
000 = channel 0, (RA0/AN0)
001 = channel 1, (RA1/AN1)
010 = channel 2, (RA2/AN2)
011 = channel 3, (RA3/AN3)
1xx = reserved, do not use

bit 2  
GO/DONE: A/D Conversion Status bit
If ADON = 1
1 = A/D conversion in progress (Setting this bit starts the A/D conversion)
0 = A/D conversion not in progress (This bit is automatically cleared by hardware when the A/D conversion is complete)

bit 1  
Reserved: Maintain this bit as ‘0’

bit 0  
ADON: A/D On bit
1 = A/D converter module is operating
0 = A/D converter module is shutoff and consumes no operating current

Legend:
R = Readable bit
W = Writable bit
U = Unimplemented bit, read as ‘0’
-n = Value at POR
‘1’ = Bit is set
‘0’ = Bit is cleared
x = Bit is unknown
REGISTER 8-2: ADCON1 REGISTER (ADDRESS: 9Fh)

<table>
<thead>
<tr>
<th>U-0</th>
<th>U-0</th>
<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>
</tr>
</thead>
<tbody>
<tr>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>PCFG2</td>
<td>PCFG1</td>
<td>PCFG0</td>
</tr>
</tbody>
</table>

bit 7

bit 7-3 Unimplemented: Read as ‘0’

bit 2-0 PCFG2:PCFG0: A/D Port Configuration Control bits

<table>
<thead>
<tr>
<th>PCFG2:PCFG0</th>
<th>AN3 RA3</th>
<th>AN2 RA2</th>
<th>AN2 RA1</th>
<th>AN0 RA0</th>
<th>VREF</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x0</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>VDD</td>
</tr>
<tr>
<td>0x1</td>
<td>VREF</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>RA3</td>
</tr>
<tr>
<td>100</td>
<td>A</td>
<td>D</td>
<td>A</td>
<td>A</td>
<td>VDD</td>
</tr>
<tr>
<td>101</td>
<td>VREF</td>
<td>D</td>
<td>A</td>
<td>A</td>
<td>RA3</td>
</tr>
<tr>
<td>11x</td>
<td>D</td>
<td>D</td>
<td>D</td>
<td>D</td>
<td>VDD</td>
</tr>
</tbody>
</table>

Legend: A = Analog input, D = Digital I/O

The ADRES register contains the result of the A/D conversion. When the A/D conversion is complete, the result is loaded into the ADRES register, the GO/DONE bit (ADCON0<2>) is cleared and the A/D interrupt flag bit ADIF is set. The block diagram of the A/D module is shown in Figure 8-1.

The value that is in the ADRES register is not modified for any Reset. The ADRES register will contain unknown data after a Power-on Reset.

After the A/D module has been configured as desired, the selected channel must be acquired before the conversion is started. The analog input channels must have their corresponding TRIS bits selected as an input. To determine acquisition time, see Section 8.1 “A/D Acquisition Requirements”. After this acquisition time has elapsed, the A/D conversion can be started. The following steps should be followed for doing an A/D conversion:

1. Configure the A/D module:
   - Configure analog pins/voltage reference/and digital I/O (ADCON1)
   - Select A/D input channel (ADCON0)
   - Select A/D conversion clock (ADCON0)
   - Turn on A/D module (ADCON0)

2. Configure A/D interrupt (if desired):
   - Clear ADIF bit
   - Set ADIE bit
   - Set GIE bit

3. Wait the required acquisition time.

4. Start conversion:
   - Set GO/DONE bit (ADCON0)

5. Wait for A/D conversion to complete, by either:
   - Polling for the GO/DONE bit to be cleared OR
   - Waiting for the A/D interrupt

6. Read A/D Result register (ADRES), clear bit ADIF if required.

7. For the next conversion, go to step 1 or step 2 as required. The A/D conversion time per bit is defined as TA. A minimum wait of 2TA is required before next acquisition starts.
8.1 A/D Acquisition Requirements

For the A/D converter to meet its specified accuracy, the charge holding capacitor (CHOLD) must be allowed to fully charge to the input channel voltage level. The analog input model is shown in Figure 8-2. 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). The source impedance affects the offset voltage at the analog input (due to pin leakage current). The maximum recommended impedance for analog sources is 10 kΩ. After the analog input channel is selected (changed) this acquisition must be done before the conversion can be started.

To calculate the minimum acquisition time, TACQ, see the PICmicro® Mid-Range Reference Manual, (DS33023). This equation calculates the acquisition time to within 1/2 LSb error. The 1/2 LSb error is the maximum error allowed for the A/D to meet its specified accuracy.

Note: When the conversion is started, the holding capacitor is disconnected from the input pin.

Legend:
- C_PIN = input capacitance
- V_T = threshold voltage
- I Leakage = leakage current at the pin due to various junctions
- R_IC = interconnect resistance
- SS = sampling switch
- CHOLD = sample/hold capacitance (from DAC)
8.2 Selecting the A/D Conversion Clock

The A/D conversion time per bit is defined as \( T_{AD} \). The A/D conversion requires 9.5 \( T_{AD} \) per 8-bit conversion. The source of the A/D conversion clock is software selectable. The four possible options for \( T_{AD} \) are:

- 2 Tosc
- 8 Tosc
- 32 Tosc
- Internal RC oscillator

For correct A/D conversions, the A/D conversion clock (\( T_{AD} \)) must be selected to ensure a minimum \( T_{AD} \) time of 1.6 \( \mu s \).

Table 8-1 shows the resultant \( T_{AD} \) times derived from the device operating frequencies and the A/D clock source selected.

8.3 Configuring Analog Port Pins

The ADCON1 and TRISA registers control the operation of the A/D port pins. The port pins that are desired as analog inputs must have their corresponding TRIS bits set (input). If the TRIS bit is cleared (output), the digital output level (\( V_{OH} \) or \( V_{OL} \)) will be converted.

The A/D operation is independent of the state of the CHS2:CHS0 bits and the TRIS bits.

### TABLE 8-1: \( T_{AD} \) vs. DEVICE OPERATING FREQUENCIES

<table>
<thead>
<tr>
<th>AD Clock Source (TAD)</th>
<th>Device Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operation</td>
<td>20 MHz</td>
</tr>
<tr>
<td>2 Tosc</td>
<td>100 ns(2)</td>
</tr>
<tr>
<td>8 Tosc</td>
<td>400 ns(2)</td>
</tr>
<tr>
<td>32 Tosc</td>
<td>1.6 ( \mu s )</td>
</tr>
<tr>
<td>RC</td>
<td>2-6 ( \mu s )(1),(4)</td>
</tr>
</tbody>
</table>

Legend: Shaded cells are outside of recommended range.

**Note 1:** The RC source has a typical \( T_{AD} \) time of 4 \( \mu s \).

**2:** These values violate the minimum required \( T_{AD} \) time.

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

**4:** When device frequency is greater than 1 MHz, the RC A/D conversion clock source is recommended for Sleep operation only.
8.4 A/D Conversions

**Note:** The GO/DONE bit should **NOT** be set in the same instruction that turns on the A/D.

8.5 Use of the ECCP Trigger

An A/D conversion can be started by the “special event trigger” of the ECCP module. This requires that the CCP1M3:CCP1M0 bits (CCP1CON<3:0>) be programmed as ‘1011’ and that the A/D module is enabled (ADON bit is set). When the trigger occurs, the GO/DONE bit will be set, starting the A/D conversion, and the Timer1 counter will be reset to zero. Timer1 is reset to automatically repeat the A/D acquisition period with minimal software overhead (moving the ADRES to the desired location). The appropriate analog input channel must be selected and the minimum acquisition done before the “special event trigger” sets the GO/DONE bit (starts a conversion).

If the A/D module is not enabled (ADON is cleared), then the “special event trigger” will be ignored by the A/D module, but will still reset the Timer1 counter.

### TABLE 8-2: SUMMARY OF A/D REGISTERS

<table>
<thead>
<tr>
<th>Address</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>Value on all other Resets</th>
</tr>
</thead>
<tbody>
<tr>
<td>05h</td>
<td>PORTA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>RA4</td>
<td>RA3</td>
<td>RA2</td>
<td>RA1</td>
<td>RA0</td>
<td>--xx 0000</td>
</tr>
<tr>
<td>05h,08h</td>
<td>INTCON</td>
<td>GIE</td>
<td>PEIE</td>
<td>T0IE</td>
<td>INTE</td>
<td>RBIE</td>
<td>T0IF</td>
<td>INTF</td>
<td>RBIF</td>
<td>0000 000x</td>
<td>0000 000u</td>
</tr>
<tr>
<td>0Ch</td>
<td>PIR1</td>
<td>ADIF</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>1Eh</td>
<td>ADRES</td>
<td>A/D Result Register</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>xxxxx</td>
<td>xxxxx</td>
</tr>
<tr>
<td>1Fh</td>
<td>ADCON0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>GO/DONE(1)</td>
<td>0000 0000</td>
<td>0000 0000</td>
</tr>
<tr>
<td>85h</td>
<td>TRISA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1111</td>
<td>1111</td>
</tr>
<tr>
<td>8Ch</td>
<td>PIE1</td>
<td>ADIE</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>9Fh</td>
<td>ADCON1</td>
<td></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’.

**Note 1:** Reserved bit, do not use.
9.0 SPECIAL FEATURES OF THE CPU

The PIC16F716 device has a host of features intended to maximize system reliability, minimize cost through elimination of external components, provide power saving operating modes and offer code protection. These are:

- OSC Selection
- Reset
  - Power-on Reset (POR)
  - Power-up Timer (PWRT)
  - Oscillator Start-up Timer (OST)
  - Brown-out Reset (BOR)
- Interrupts
- Watchdog Timer (WDT)
- Sleep
- Code protection
- ID locations
- In-Circuit Serial Programming™ (ICSP™)

The PIC16F716 device has a Watchdog Timer, which can be shut off only through configuration bits. It runs off its own RC oscillator for added reliability. There are two timers that offer necessary delays on power-up. One is the Oscillator Start-up Timer (OST), intended to keep the chip in Reset until the crystal oscillator is stable. The other is the Power-up Timer (PWRT), which provides a fixed delay on power-up only and is designed to keep the part in Reset while the power supply stabilizes. With these two timers on-chip, most applications need no external Reset circuitry.

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, or through an interrupt. Several oscillator options are also made available to allow the part to fit the application. The RC oscillator option saves system cost, while the LP crystal option saves power. A set of configuration bits are used to select various options.

Additional information on special features is available in the PICmicro® Mid-Range Reference Manual, (DS33023).

9.1 Configuration Bits

The configuration bits can be programmed (read as ‘0’ or left unprogrammed (read as ‘1’) to select various device configurations. These bits are mapped in program memory location 2007h.

The user will note that address 2007h is beyond the user program memory space. In fact, it belongs to the special configuration memory space (2000h – 3FFFh), which can be accessed only during programming.
REGISTER 9-1: CONFIGURATION WORD

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Value 1</th>
<th>Value 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>CP: Flash Program Memory Code Protection bit</td>
<td>1 = Code protection off</td>
<td>0 = All program memory code protected</td>
</tr>
<tr>
<td>12-8</td>
<td>Unimplemented: Read as ‘1’</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>BORV: Brown-out Reset Voltage bit</td>
<td>1 = VBOR set to 4.0V</td>
<td>0 = VBOR set to 2.5V</td>
</tr>
<tr>
<td>6</td>
<td>BOREN: Brown-out Reset Enable bit (1)</td>
<td>1 = BOR enabled</td>
<td>0 = BOR disabled</td>
</tr>
<tr>
<td>5-4</td>
<td>Unimplemented: Read as ‘1’</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>PWRTE: Power-up Timer Enable bit (1)</td>
<td>1 = PWRT disabled</td>
<td>0 = PWRT enabled</td>
</tr>
<tr>
<td>2</td>
<td>WDTE: Watchdog Timer Enable bit</td>
<td>1 = WDT enabled</td>
<td>0 = WDT disabled</td>
</tr>
<tr>
<td>1-0</td>
<td>FOSC1:FOSC0: Oscillator Selection bits</td>
<td>11 = RC oscillator</td>
<td>10 = HS oscillator</td>
</tr>
</tbody>
</table>

Note 1: Enabling Brown-out Reset does not automatically enable Power-up Timer (PWRTE).

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
9.2 Oscillator Configurations

9.2.1 OSCILLATOR TYPES

The PIC16F716 can be operated in four different oscillator modes. The user can program two configuration bits (FOSC1 and FOSC0) to select one of these four modes:

- LP - Low-power Crystal
- XT - Crystal/Resonator
- HS - High-speed Crystal/Resonator
- RC - Resistor/Capacitor

9.2.2 CRYSTAL OSCILLATOR/CERAMIC RESONATORS

In XT, LP or HS modes, a crystal or ceramic resonator is connected to the OSC1/CLKIN and OSC2/CLKOUT pins to establish oscillation (Figure 9-1). The PIC16F716 oscillator design requires the use of a parallel cut crystal. Use of a series cut crystal may give a frequency out of the crystal manufacturers specifications. When in XT, LP or HS modes, the device can have an external clock source to drive the OSC1/CLKIN pin (Figure 9-2).

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

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

---

### TABLE 9-1: CERAMIC RESONATORS

<table>
<thead>
<tr>
<th>Mode</th>
<th>Freq</th>
<th>OSC1 (C1)</th>
<th>OSC2 (C2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>XT</td>
<td>455 kHz</td>
<td>68-100 pF</td>
<td>68-100 pF</td>
</tr>
<tr>
<td></td>
<td>2.0 MHz</td>
<td>15-68 pF</td>
<td>15-68 pF</td>
</tr>
<tr>
<td>HS</td>
<td>4.0 MHz</td>
<td>10-68 pF</td>
<td>10-68 pF</td>
</tr>
<tr>
<td></td>
<td>8.0 MHz</td>
<td>15-68 pF</td>
<td>15-68 pF</td>
</tr>
<tr>
<td></td>
<td>16.0 MHz</td>
<td>10-22 pF</td>
<td>10-22 pF</td>
</tr>
</tbody>
</table>

**Note 1:** These values are for design guidance only. See notes at bottom of page.

---

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

<table>
<thead>
<tr>
<th>Osc Type</th>
<th>Crystal Freq</th>
<th>Cap. Range C1</th>
<th>Cap. Range C2</th>
</tr>
</thead>
<tbody>
<tr>
<td>LP</td>
<td>32 kHz</td>
<td>15-33 pF</td>
<td>15-33 pF</td>
</tr>
<tr>
<td></td>
<td>200 kHz</td>
<td>5-10 pF</td>
<td>5-10 pF</td>
</tr>
<tr>
<td>XT</td>
<td>200 kHz</td>
<td>47-68 pF</td>
<td>47-68 pF</td>
</tr>
<tr>
<td></td>
<td>1 MHz</td>
<td>15-33 pF</td>
<td>15-33 pF</td>
</tr>
<tr>
<td></td>
<td>4 MHz</td>
<td>15-33 pF</td>
<td>15-33 pF</td>
</tr>
<tr>
<td>HS</td>
<td>4 MHz</td>
<td>15-33 pF</td>
<td>15-33 pF</td>
</tr>
<tr>
<td></td>
<td>8 MHz</td>
<td>15-33 pF</td>
<td>15-33 pF</td>
</tr>
<tr>
<td></td>
<td>20 MHz</td>
<td>15-33 pF</td>
<td>15-33 pF</td>
</tr>
</tbody>
</table>

**Note 1:** These values are for design guidance only. See notes at bottom of page.

---

**Note 1:** Higher capacitance increases the stability of the oscillator, but also increases the start-up time.

**2:** Since each resonator/crystal has its own characteristics, the user should consult the resonator/crystal manufacturer for appropriate values of external components.

**3:** RS may be required to avoid overdriving crystals with low drive level specification.

**4:** When using an external clock for the OSC1 input, loading of the OSC2 pin must be kept to a minimum by leaving the OSC2 pin unconnected.

---

**Note 1:** See Table 9-1 and Table 9-2 for recommended values of C1 and C2.

- **Note 2:** A series resistor (RS) may be required.
- **Note 3:** RF varies with the crystal chosen.
9.2.3 RC OSCILLATOR

For timing insensitive applications, the "RC" device option offers additional cost savings. The RC oscillator frequency is a function of the supply voltage, the resistor (R<sub>EXT</sub>) and capacitor (C<sub>EXT</sub>) values and the operating temperature. In addition to this, the oscillator frequency will vary from unit to unit due to normal process parameter variation. Furthermore, the difference in lead frame capacitance between package types will also affect the oscillation frequency, especially for low C<sub>EXT</sub> values. The user also needs to take into account variation due to tolerance of external R and C components used. Figure 9-3 shows how the R/C combination is connected to the PIC16F716.

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

A simplified block diagram of the On-chip Reset circuit is shown in Figure 9-5. The PICmicro® microcontrollers have an MCLR 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 the MCLR pin low.

9.4 Power-On Reset (POR)

A Power-on Reset pulse is generated on-chip when V<sub>DD</sub> rise is detected. To take advantage of the POR, just tie the MCLR pin directly (or through a resistor) to V<sub>DD</sub>. This will eliminate external RC components usually needed to create a Power-on Reset. A maximum rise time for V<sub>DD</sub> is specified (parameter D004). For a slow rise time, see Figure 9-4.

When the device starts normal operation (exits the Reset condition), device operating parameters (voltage, frequency, temperature,...) must be met to ensure operation. If these conditions are not met, the device must be held in Reset until the operating conditions are met. Brown-out Reset may be used to meet the start-up conditions.

**FIGURE 9-4:** EXTERNAL POWER-ON RESET CIRCUIT (FOR SLOW V<sub>DD</sub> POWER-UP)
9.5 Power-up Timer (PWRT)

The Power-up Timer provides a fixed nominal time-out, on power-up only, from the POR. The Power-up Timer operates on an internal RC oscillator. The chip is kept in Reset as long as the PWRT is active. The PWRT's time delay allows VDD to rise to an acceptable level. The power-up timer enable configuration bit, PWRTE, is provided to enable/disable the PWRT.

The power-up time delay will vary from chip-to-chip due to VDD, temperature and process variation. See AC parameters for details.

9.6 Oscillator Start-up Timer (OST)

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

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

9.7 Programmable Brown-Out Reset (PBOR)

The PIC16F716 has on-chip Brown-out Reset circuitry. A configuration bit, BOREN, can disable (if clear/programmed) or enable (if set) the Brown-out Reset circuitry.

The BORV configuration bit selects the programmable Brown-out Reset threshold voltage (V<BOR>). When BORV is 1, VBOR is 4.0V. When BORV is 0, VBOR is 2.5V.

A Brown-out Reset occurs when VDD falls below VBOR for a time greater than parameter T<sub>BOR</sub> (see Table 12-4). A Brown-out Reset is not guaranteed to occur if VDD falls below VBOR for less than parameter T<sub>BOR</sub>.

On any Reset (Power-on, Brown-out, Watchdog, etc.) the chip will remain in Reset until VDD rises above VBOR. The Power-up Timer will be invoked and will keep the chip in Reset an additional 72 ms only if the Power-up Timer enable bit in the configuration register is set to 0 (PWRTE = 0).

If the Power-up Timer is enabled and 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 72 ms Reset. See Figure 9-6.

For operations where the desired brown-out voltage is other than 4.0V or 2.5V, an external brown-out circuit must be used. Figure 9-8, Figure 9-9 and Figure 9-10 show examples of external Brown-out Protection circuits.
FIGURE 9-5: SIMPLIFIED BLOCK DIAGRAM OF ON-CHIP RESET CIRCUIT

Note 1: This is a separate oscillator from the RC oscillator of the CLkin pin.

FIGURE 9-6: BROWN-OUT SITUATIONS (PWRT ≤ 0)
**FIGURE 9-7: EXTERNAL BROWN-OUT PROTECTION CIRCUIT 1**

Note 1: This circuit will activate Reset when VDD goes below (Vz + 0.7V) where Vz = Zener voltage.

Note 2: Internal Brown-out Reset circuitry should be disabled when using this circuit.

**FIGURE 9-8: EXTERNAL BROWN-OUT PROTECTION CIRCUIT 2**

Note 1: This brown-out circuit is less expensive, albeit less accurate. Transistor Q1 turns off when VDD is below a certain level such that:

\[
V_{DD} \times \frac{R1}{R1 + R2} = 0.7 \text{ V}
\]

Note 2: Internal Brown-out Reset should be disabled when using this circuit.

Note 3: Resistors should be adjusted for the characteristics of the transistor.

**FIGURE 9-9: EXTERNAL BROWN-OUT PROTECTION CIRCUIT 3**

Note 1: This brown-out protection circuit employs Microchip Technology’s MCP809 microcontroller supervisor. The MCP8XX and MCP1XX families of supervisors provide push-pull and open collector outputs with both high and low active Reset pins. There are 7 different trip point selections to accommodate 5V and 3V systems.
9.8 Time-out Sequence

On power-up, the time-out sequence is as follows: First PWRT time-out is invoked after the POR time delay has expired. Then OST is activated. The total time-out will vary based on oscillator configuration and the status of the PWRT. For example, in RC mode with the PWRT disabled, there will be no time-out at all. Figure 9-10, Figure 9-11, and Figure 9-12 depict time-out sequences on power-up.

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 (Figure 9-12). This is useful for testing purposes or to synchronize more than one PIC16F716 device operating in parallel.

Table 9-5 shows the Reset conditions for some special function registers, while Table 9-6 shows the Reset conditions for all the registers.

TABLE 9-3: TIME-OUT IN VARIOUS SITUATIONS

<table>
<thead>
<tr>
<th>Oscillator Configuration</th>
<th>Power-up or Brown-out</th>
<th>Wake-up from Sleep</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PWRT = 0</td>
<td>PWRT = 1</td>
</tr>
<tr>
<td>XT, HS, LP</td>
<td>72 ms + 1024 Tosc</td>
<td>1024 Tosc</td>
</tr>
<tr>
<td>RC</td>
<td>72 ms</td>
<td>—</td>
</tr>
</tbody>
</table>

TABLE 9-4: STATUS BITS AND THEIR SIGNIFICANCE

<table>
<thead>
<tr>
<th>POR</th>
<th>BOR</th>
<th>TO</th>
<th>PD</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>x</td>
<td>1</td>
<td>1</td>
<td>Power-on Reset (BOREN = 0)</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>Power-on Reset (BOREN = 1)</td>
</tr>
<tr>
<td>0</td>
<td>x</td>
<td>0</td>
<td>x</td>
<td>Illegal, TO is set on POR</td>
</tr>
<tr>
<td>0</td>
<td>x</td>
<td>x</td>
<td>0</td>
<td>Illegal, PD is set on POR</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>Brown-out Reset</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>WDT Reset</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>WDT Wake-up</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>u</td>
<td>u</td>
<td>MCLR Reset during normal operation</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>MCLR Reset during Sleep or interrupt wake-up from Sleep</td>
</tr>
</tbody>
</table>
### TABLE 9-5:  RESET 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 (BOREN = 0)</td>
<td>000h</td>
<td>0001 1xxx</td>
<td>---- --0x</td>
</tr>
<tr>
<td>Power-on Reset (BOREN = 1)</td>
<td>000h</td>
<td>0001 1xxx</td>
<td>---- --01</td>
</tr>
<tr>
<td>MCLR Reset during normal operation</td>
<td>000h</td>
<td>000u uuuu</td>
<td>---- --uu</td>
</tr>
<tr>
<td>MCLR Reset during Sleep</td>
<td>000h</td>
<td>0001 0uuu</td>
<td>---- --uu</td>
</tr>
<tr>
<td>WDT Reset</td>
<td>000h</td>
<td>0000 1uuu</td>
<td>---- --uu</td>
</tr>
<tr>
<td>WDT Wake-up</td>
<td>PC + 1</td>
<td>uuu0 0uuu</td>
<td>---- --uu</td>
</tr>
<tr>
<td>Brown-out Reset</td>
<td>000h</td>
<td>0001 1uuu</td>
<td>---- --u0</td>
</tr>
<tr>
<td>Interrupt wake-up from Sleep</td>
<td>PC + 1(1)</td>
<td>uuu1 0uuu</td>
<td>---- --uu</td>
</tr>
</tbody>
</table>

**Legend:**  
- u = unchanged, x = unknown, - = unimplemented bit read as ‘0’.

**Note 1:** When the wake-up is due to an interrupt and the GIE bit is set, the PC is loaded with the interrupt vector (0004h).
TABLE 9-6: INITIALIZATION CONDITIONS FOR ALL REGISTERS OF THE PIC16F716

<table>
<thead>
<tr>
<th>Register</th>
<th>Power-on Reset, Brown-out Reset</th>
<th>MCLR Resets WDT Reset</th>
<th>Wake-up via WDT or Interrupt</th>
</tr>
</thead>
<tbody>
<tr>
<td>W</td>
<td>xxxxxxxx</td>
<td>uuuuuuu</td>
<td>uuuuuuu</td>
</tr>
<tr>
<td>INDF</td>
<td>N/A</td>
<td>uuuuuuu</td>
<td>N/A</td>
</tr>
<tr>
<td>TMR0</td>
<td>xxxxxxxx</td>
<td>uuuuuuu</td>
<td>uuuuuuu</td>
</tr>
<tr>
<td>PCL</td>
<td>0000h</td>
<td>0000h</td>
<td>PC + 1&lt;sup&gt;(2)&lt;/sup&gt;</td>
</tr>
<tr>
<td>STATUS</td>
<td>0001 1xxxx</td>
<td>000q quuu&lt;sup&gt;(3)&lt;/sup&gt;</td>
<td>uuqq quuuu&lt;sup&gt;(3)&lt;/sup&gt;</td>
</tr>
<tr>
<td>FSR</td>
<td>xxxxxxxx</td>
<td>uuuuuuu</td>
<td>uuuuuuu</td>
</tr>
<tr>
<td>PORTA&lt;sup&gt;(4), (5), (6)&lt;/sup&gt;</td>
<td>--xx 0000</td>
<td>--xx 0000</td>
<td>--uu uuuuu</td>
</tr>
<tr>
<td>PORTB&lt;sup&gt;(4), (5)&lt;/sup&gt;</td>
<td>xxxxxxxx</td>
<td>uuuuuuu</td>
<td>uuuuuuu</td>
</tr>
<tr>
<td>PCLATH</td>
<td>---0 0000</td>
<td>---0 0000</td>
<td>---uu uuuuu</td>
</tr>
<tr>
<td>INTCON</td>
<td>0000 --00x</td>
<td>0000 --00u</td>
<td>uuuu --uuu&lt;sup&gt;(1)&lt;/sup&gt;</td>
</tr>
<tr>
<td>PIR1</td>
<td>--0-- --000</td>
<td>--0-- --000</td>
<td>--0-- --uuu&lt;sup&gt;(1)&lt;/sup&gt;</td>
</tr>
<tr>
<td>TMR1L</td>
<td>xxxxxxxx</td>
<td>uuuuuuu</td>
<td>uuuuuuu</td>
</tr>
<tr>
<td>TMR1H</td>
<td>xxxxxxxx</td>
<td>uuuuuuu</td>
<td>uuuuuuu</td>
</tr>
<tr>
<td>T1CON</td>
<td>--00 0000</td>
<td>--uu uuuuu</td>
<td>--uu uuuuu</td>
</tr>
<tr>
<td>TMR2</td>
<td>0000 0000</td>
<td>0000 0000</td>
<td>uuuuuuu</td>
</tr>
<tr>
<td>T2CON</td>
<td>--00 0000</td>
<td>--00 0000</td>
<td>--00 uuuuu</td>
</tr>
<tr>
<td>CCP1L</td>
<td>xxxxxxxx</td>
<td>uuuuuuu</td>
<td>uuuuuuu</td>
</tr>
<tr>
<td>CCP1H</td>
<td>xxxxxxxx</td>
<td>uuuuuuu</td>
<td>uuuuuuu</td>
</tr>
<tr>
<td>CCP1CON</td>
<td>0000 0000</td>
<td>0000 0000</td>
<td>uuuuuuu</td>
</tr>
<tr>
<td>PWM1CON</td>
<td>0000 0000</td>
<td>0000 0000</td>
<td>uuuuuuu</td>
</tr>
<tr>
<td>ECCPAS</td>
<td>00-0 0000</td>
<td>00-0 0000</td>
<td>uu-uu uuuuu</td>
</tr>
<tr>
<td>ADRES</td>
<td>xxxxxxxx</td>
<td>uuuuuuu</td>
<td>uuuuuuu</td>
</tr>
<tr>
<td>ADCON0</td>
<td>0000 0000</td>
<td>0000 0000</td>
<td>uuuuuuu</td>
</tr>
<tr>
<td>OPTION_REG</td>
<td>11111 1111</td>
<td>11111 1111</td>
<td>uuuuuuu</td>
</tr>
<tr>
<td>TRISA</td>
<td>--11 1111</td>
<td>--11 1111</td>
<td>--uu uuuuu</td>
</tr>
<tr>
<td>TRIS8</td>
<td>11111 1111</td>
<td>11111 1111</td>
<td>uuuuuuu</td>
</tr>
<tr>
<td>PIE1</td>
<td>--0-- --000</td>
<td>--0-- --000</td>
<td>--0-- --uuu</td>
</tr>
<tr>
<td>PCON</td>
<td>----- --qq</td>
<td>----- --uu</td>
<td>----- --uu</td>
</tr>
<tr>
<td>PR2</td>
<td>11111 1111</td>
<td>11111 1111</td>
<td>uuuuuuu</td>
</tr>
<tr>
<td>ADCON1</td>
<td>----- --000</td>
<td>----- --000</td>
<td>----- --uuu</td>
</tr>
</tbody>
</table>

Legend:  
- = unchanged,  x = unknown,  - = unimplemented bit, read as ’0’,  q = value depends on condition

Note 1: One or more bits in INTCON and/or PIR1 will be affected (to cause wake-up).
2: When the wake-up is due to an interrupt and the GIE bit is set, the PC is loaded with the interrupt vector (0004h).
3: See Table 9-5 for Reset value for specific condition.
4: On any device Reset, these pins are configured as inputs.
5: This is the value that will be in the port output latch.
6: Output latches are unknown or unchanged. Analog inputs default to analog and read ‘0’.
FIGURE 9-10: TIME-OUT SEQUENCE ON POWER-UP (MCLR TIED TO V_{DD})

V_{DD}  
MCLR  
INTERNAL POR  

PWRT TIME-OUT  

OST TIME-OUT  

INTERNAL RESET

FIGURE 9-11: TIME-OUT SEQUENCE ON POWER-UP (MCLR NOT TIED TO V_{DD}): CASE 1

V_{DD}  
MCLR  
INTERNAL POR  

PWRT TIME-OUT  

OST TIME-OUT  

INTERNAL RESET

FIGURE 9-12: TIME-OUT SEQUENCE ON POWER-UP (MCLR NOT TIED TO V_{DD}): CASE 2

V_{DD}  
MCLR  
INTERNAL POR  

PWRT TIME-OUT  

OST TIME-OUT  

INTERNAL RESET
9.10 Interrupts

The PIC16F716 devices have up to 7 sources of interrupt. The Interrupt Control Register (INTCON) records individual interrupt requests in flag bits. It also has individual and global interrupt enable bits.

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

A Global Interrupt Enable bit, GIE (INTCON<7>) enables all un-masked interrupts when set, or disables all interrupts when cleared. When bit GIE is enabled, and an interrupt's flag bit and mask bit are set, the interrupt will vector immediately. Individual interrupts can be disabled through their corresponding enable bits in various registers. Individual interrupt bits are set, regardless of the status of the GIE bit. The GIE bit is cleared on Reset and when an interrupt vector occurs.

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

The RB0/INT pin interrupt, the RB port change interrupt and the TMR0 overflow interrupt flags are contained in the INTCON register.

The peripheral interrupt flags are contained in the special function registers, PIR1 and PIR2. The corresponding interrupt enable bits are contained in special function registers, PIE1 and PIE2, and the peripheral interrupt enable bit is contained in special function register, INTCON.

When an interrupt is responded to, the GIE bit is cleared to disable any further interrupt, the return address is pushed onto the stack and the PC is loaded with 0004h. Once in the interrupt service routine, the source(s) of the interrupt can be determined by polling the interrupt flag bits. The interrupt flag bit(s) must be cleared in software before re-enabling interrupts to avoid recursive interrupts.

For external interrupt events, such as the INT pin or PORTB change interrupt, the interrupt latency will be three or four instruction cycles. The exact latency depends when the interrupt event occurs. The latency is the same for one or two-cycle instructions. Individual interrupt flag bits are set, regardless of the status of their corresponding mask bit or the GIE bit.
9.10.1 INT INTERRUPT

External interrupt on RB0/INT pin is edge triggered, either rising if bit INTEDG (OPTION_REG<6>) is set, or falling if the INTEGD bit is clear. When a valid edge appears on the RB0/INT pin, flag bit INTF (INTCON<1>) is set. This interrupt can be disabled by clearing enable bit INTE (INTCON<4>). Flag bit INTF must be cleared in software in the interrupt service routine before re-enabling this interrupt. The INT interrupt can wake-up the processor from Sleep, if bit INTE was set prior to going into Sleep. The status of global interrupt enable bit GIE decides whether or not the processor branches to the interrupt vector following wake-up. See Section 9.13 “Power-down Mode (Sleep)” for details on Sleep mode.

9.10.2 TMR0 INTERRUPT

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

9.10.3 PORTB INTCON CHANGE

An input change on PORTB<7:4> sets flag bit RBIF (INTCON<0>). The interrupt can be enabled/disabled by setting/clearing enable bit RBIE (INTCON<4>). (Section 3.2 “PORTB and the TRISB Register”).

9.11 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, (i.e., W register and Status register). This will have to be implemented in firmware.

Example 9-1 stores and restores the W, Status, PCLATH and FSR registers. Context storage registers, W_TEMP, STATUS_TEMP, PCLATH_TEMP and FSR_TEMP, must be defined in Common RAM which are those addresses between 70h-7Fh in Bank 0 and between F0h-FFh in Bank 1.

The example:

a) Stores the W register.
b) Stores the Status register in Bank 0.
c) Stores the PCLATH register
d) Stores the FSR register.
e) Executes the interrupt service routine code (User-generated).
f) Restores all saved registers in reverse order from which they were stored

EXAMPLE 9-1: SAVING STATUS, W, AND PCLATH REGISTERS IN RAM

```
MOVF W_TEMP , W ;Copy W to TEMP register, could be bank one or zero
SWAPF STATUS , W ;Swap status to be saved into W
MOVWF STATUS_TEMP ;Save status to bank zero STATUS_TEMP register
MOVF PCLATH , W ;Only required if using pages 1, 2 and/or 3
MOVWF PCLATH_TEMP ;Save PCLATH into W
CLRF PCLATH ;Page zero, regardless of current page
BCF STATUS , IRP ;Return to Bank 0
MOVF FSR , W ;Copy FSR to W
MOVWF FSR_TEMP ;Copy FSR from W to FSR_TEMP
: ;(ISR)
: 
MOVF FSR_TEMP , W ;Restore FSR
MOVWF FSR ;Move W into FSR
MOVF PCLATH_TEMP , W ;Restore PCLATH
MOVWF PCLATH ;Move W into PCLATH
SWAPF STATUS_TEMP , W ;Swap STATUS_TEMP register into W
MOVF STATUS , W ;Move W into STATUS register
SWAPF W_TEMP , F ;Swap W_TEMP
SWAPF W_TEMP , W ;Swap W_TEMP into W
RETFIE ;Return from interrupt and enable GIE
```
9.12 Watchdog Timer (WDT)

The Watchdog Timer is a free running, on-chip, RC oscillator which does not require any external components. This RC oscillator is separate from the RC oscillator of the OSC1/CLKIN pin. That means that the WDT will run, even if the clock on the OSC1/CLKIN and OSC2/CLKOUT pins of the device have been stopped, for example, by execution of a SLEEP instruction.

During normal operation, a WDT time-out generates a device Reset (Watchdog Timer Reset). If the device is in Sleep mode, a WDT time-out causes the device to wake-up and continue with normal operation (Watchdog Timer Wake-up). The TO bit in the Status register will be cleared upon a Watchdog Timer time-out.

The WDT can be permanently disabled by clearing configuration bit WDTE (Section 9.1 “Configuration Bits”).

WDT time-out period values may be found in the Electrical Specifications section under TWDT (parameter #31). Values for the WDT prescaler (actually a postscaler, but shared with the Timer0 prescaler) may be assigned using the OPTION_REG register.

Note: The CLRWDT and SLEEP instructions clear the WDT and the postscaler, if assigned to the WDT, and prevent it from timing out and generating a device Reset condition.

Note: When a CLRWDT instruction is executed and the prescaler is assigned to the WDT, the prescaler count will be cleared, but the prescaler assignment is not changed.

FIGURE 9-14: WATCHDOG TIMER BLOCK DIAGRAM

From TMR0 Clock Source (Figure 4-2)

WDT Timer

WDT Enable Bit

MUX

PSA

8

PS2:PS0

8 - to - 1 MUX

To TMR0 (Figure 4-2)

WDT Time-out

Note: PSA and PS2:PS0 are bits in the OPTION_REG register.

FIGURE 9-15: SUMMARY OF WATCHDOG TIMER REGISTERS

<table>
<thead>
<tr>
<th>Address</th>
<th>Name</th>
<th>Bits 13:8</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>
</tr>
</thead>
<tbody>
<tr>
<td>2007h</td>
<td>Config. bits</td>
<td>(1)</td>
<td>BORV(1)</td>
<td>BOREN(1)</td>
<td>—</td>
<td>—</td>
<td>PWRT(1)</td>
<td>WDTE</td>
<td>FOSC1</td>
<td>FOSC0</td>
</tr>
<tr>
<td>81h</td>
<td>OPTION_REG</td>
<td>N/A</td>
<td>RBPU</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>

Legend: Shaded cells are not used by the Watchdog Timer.

Note 1: See Register 9-1 for operation of these bits.
9.13 Power-down Mode (Sleep)

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

If enabled, the Watchdog Timer will be cleared but keeps running, the PD bit (STATUS<3>) is cleared, the TO (STATUS<4>) bit is set, and the oscillator driver is turned off. The I/O ports maintain the status they had, before the **SLEEP** instruction was executed (driving high, low or high-impedance).

For lowest current consumption in this mode, place all I/O pins at either VDD or VSS, ensure no external circuitry is drawing current from the I/O pin, power-down the A/D and the disable external clocks. Pull all I/O pins that are hi-impedance inputs, 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 PORTB should be considered.

The MCLR pin must be at a logic high level (parameter D042).

9.13.1 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 INT pin, RB port change or some peripheral interrupts.

External MCLR Reset will cause a device Reset. All other events are considered a continuation of program execution and cause a "wake-up". The TO and PD bits in the Status register can be used to determine the cause of device Reset. The PD bit, which is set on power-up, is cleared when Sleep is invoked. The TO bit is cleared if a WDT time-out occurred (and caused wake-up).

The following peripheral interrupts can wake the device from Sleep:

1. TMR1 interrupt. Timer1 must be operating as an asynchronous counter.
2. ECCP capture mode interrupt.
3. ADC running in ADRC mode.

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 pre-fetched. For the device to wake-up through an interrupt event, the corresponding interrupt enable bit must be set (enabled). Wake-up is regardless of the state of the GIE bit. If the GIE bit is clear (disabled), the device continues execution at the instruction after the **SLEEP** instruction. If the GIE bit is set (enabled), the device executes the instruction after the **SLEEP** instruction and 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.

9.13.2 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 postscaler will not be cleared, the TO bit will not be set and PD bits 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 will be completely executed before the wake-up. Therefore, the WDT and WDT postscaler 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.
9.14 Program Verification/Code Protection

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

9.15 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. It is recommended that only the 4 Least Significant bits of the ID location are used.

9.16 In-Circuit Serial Programming™

PIC16F716 microcontrollers can be serially programmed while in the end application circuit. This is simply done with two lines for clock and data, and three other lines for power, ground and the 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.

For complete details on serial programming, please refer to the In-Circuit Serial Programming™ (ICSP™) Specification, (DS40245).
10.0 INSTRUCTION SET SUMMARY

Each PIC16F716 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 PIC16F716 instruction set summary in Table 10-2 lists byte-oriented, bit-oriented, and literal and control operations. Table 10-1 shows the opcode field descriptions. 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 number of the bit affected by the operation, while ‘f’ represents the number of the file in which the bit is located.

For literal and control operations, ‘k’ represents an eight or eleven bit constant or literal value.

**TABLE 10-1: 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)</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>label</td>
<td>Label name</td>
</tr>
<tr>
<td>TOS</td>
<td>Top of Stack</td>
</tr>
<tr>
<td>PC</td>
<td>Program Counter</td>
</tr>
<tr>
<td>PCLATH</td>
<td>Program Counter High Latch</td>
</tr>
<tr>
<td>GIE</td>
<td>Global Interrupt Enable bit</td>
</tr>
<tr>
<td>WDT</td>
<td>Watchdog Timer/Counter</td>
</tr>
<tr>
<td>TO</td>
<td>Time-out bit</td>
</tr>
<tr>
<td>PD</td>
<td>Power-down bit</td>
</tr>
<tr>
<td>dest</td>
<td>Destination either the W register or the specified register file location</td>
</tr>
<tr>
<td>( )</td>
<td>Options</td>
</tr>
<tr>
<td>( )</td>
<td>Contents</td>
</tr>
<tr>
<td>→</td>
<td>Assigned to</td>
</tr>
<tr>
<td>&lt; &gt;</td>
<td>Register bit field</td>
</tr>
<tr>
<td>∈</td>
<td>In the set of</td>
</tr>
<tr>
<td>italics</td>
<td>User defined term (font is courier)</td>
</tr>
</tbody>
</table>

The instruction set is highly orthogonal and is grouped into three basic categories:

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

All instructions are executed within one single instruction cycle, unless a conditional test is true or the program counter is changed as a result of an instruction. In this case, the execution takes two instruction cycles with the second cycle executed as a NOP. One instruction cycle consists of four oscillator periods. Thus, for an oscillator frequency of 4 MHz, the normal instruction execution time is 1 µs. If a conditional test is true or the program counter is changed as a result of an instruction, the instruction execution time is 2 µs.

Table 10-2 lists the instructions recognized by the MPASM™ assembler.

Figure 10-1 shows the three general formats that the instructions can have.

**Note 1:** Any unused opcode is reserved. Use of any reserved opcode may cause unexpected operation.

**Note 2:** To maintain upward compatibility with future PICmicro products, do not use the OPTION and TRIS instructions.

All examples use the following format to represent a hexadecimal number:

0xhh

where h signifies a hexadecimal digit.

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

<table>
<thead>
<tr>
<th>Byte-oriented file register operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>13 8 7 6 0</td>
</tr>
<tr>
<td>d = 0 for destination W</td>
</tr>
<tr>
<td>d = 1 for destination f</td>
</tr>
<tr>
<td>f = 7-bit file register address</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bit-oriented file register operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>13 10 9 7 6 0</td>
</tr>
<tr>
<td>b = 3-bit address</td>
</tr>
<tr>
<td>f = 7-bit file register address</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Literal and control operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>General</td>
</tr>
<tr>
<td>13 8 7 0</td>
</tr>
<tr>
<td>k = 8-bit immediate value</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CALL and GOTO instructions only</th>
</tr>
</thead>
<tbody>
<tr>
<td>13 11 10 0</td>
</tr>
<tr>
<td>k = 11-bit immediate value</td>
</tr>
<tr>
<td>Mnemonic, Operands</td>
</tr>
<tr>
<td>-------------------</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>BYTE-ORIENTED FILE REGISTER OPERATIONS</strong></td>
</tr>
<tr>
<td>ADDWF f, d</td>
</tr>
<tr>
<td>ANDWF f, d</td>
</tr>
<tr>
<td>CLRWF —</td>
</tr>
<tr>
<td>COMWF f, d</td>
</tr>
<tr>
<td>DECF f, d</td>
</tr>
<tr>
<td>DECFSZ f, d</td>
</tr>
<tr>
<td>INCF f, d</td>
</tr>
<tr>
<td>INCFSZ f, d</td>
</tr>
<tr>
<td>IORWF f, d</td>
</tr>
<tr>
<td>MOVF f, d</td>
</tr>
<tr>
<td>MOVWF f</td>
</tr>
<tr>
<td>NOP —</td>
</tr>
<tr>
<td>RLF f, d</td>
</tr>
<tr>
<td>RRF f, d</td>
</tr>
<tr>
<td>SUBWF f, d</td>
</tr>
<tr>
<td>SWAPF f, d</td>
</tr>
<tr>
<td>XORWF f, d</td>
</tr>
<tr>
<td><strong>BIT-ORIENTED FILE REGISTER OPERATIONS</strong></td>
</tr>
<tr>
<td>BCF f, b</td>
</tr>
<tr>
<td>BSF f, b</td>
</tr>
<tr>
<td>BTFSC f, b</td>
</tr>
<tr>
<td>BITFSS f, b</td>
</tr>
<tr>
<td><strong>LITERAL AND CONTROL OPERATIONS</strong></td>
</tr>
<tr>
<td>ADDLW k</td>
</tr>
<tr>
<td>ANDLW k</td>
</tr>
<tr>
<td>CALL k</td>
</tr>
<tr>
<td>CLRWDT —</td>
</tr>
<tr>
<td>GOTO k</td>
</tr>
<tr>
<td>IORLW k</td>
</tr>
<tr>
<td>MOVLRW k</td>
</tr>
<tr>
<td>RETFIE —</td>
</tr>
<tr>
<td>RETLW k</td>
</tr>
<tr>
<td>RETURN —</td>
</tr>
<tr>
<td>SLEEP —</td>
</tr>
<tr>
<td>SUBBLW k</td>
</tr>
<tr>
<td>XORLW k</td>
</tr>
</tbody>
</table>

**Note 1:** When an I/O register is modified as a function of itself (e.g., MOVF PORTB, 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'.

**Note 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.

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

### ADDLW

**Add Literal and W**

<table>
<thead>
<tr>
<th>Syntax:</th>
<th>[label] ADDLW k</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operands:</td>
<td>0 ≤ k ≤ 255</td>
</tr>
<tr>
<td>Operation:</td>
<td>(W) + k → (W)</td>
</tr>
<tr>
<td>Status Affected:</td>
<td>C, DC, Z</td>
</tr>
<tr>
<td>Encoding:</td>
<td>11 11xx kkkk kkkk</td>
</tr>
<tr>
<td>Description:</td>
<td>The contents of the W register are added to the eight bit literal 'k' and the result is placed in the W register.</td>
</tr>
<tr>
<td>Words:</td>
<td>1</td>
</tr>
<tr>
<td>Cycles:</td>
<td>1</td>
</tr>
<tr>
<td>Example:</td>
<td>ADDLW 0x15</td>
</tr>
</tbody>
</table>

**Example**

Before Instruction
W = 0x10
After Instruction
W = 0x25

### ANDLW

**AND Literal with W**

<table>
<thead>
<tr>
<th>Syntax:</th>
<th>[label] ANDLW k</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operands:</td>
<td>0 ≤ k ≤ 255</td>
</tr>
<tr>
<td>Operation:</td>
<td>(W) .AND. (k) → (W)</td>
</tr>
<tr>
<td>Status Affected:</td>
<td>Z</td>
</tr>
<tr>
<td>Encoding:</td>
<td>11 1001 kkkk kkkk</td>
</tr>
<tr>
<td>Description:</td>
<td>The contents of W register are AND'ed with the eight bit literal 'k'. The result is placed in the W register.</td>
</tr>
<tr>
<td>Words:</td>
<td>1</td>
</tr>
<tr>
<td>Cycles:</td>
<td>1</td>
</tr>
<tr>
<td>Example:</td>
<td>ANDLW 0x5F</td>
</tr>
</tbody>
</table>

**Example**

Before Instruction
W = 0xA3
After Instruction
W = 0x03

### ADDWF

**Add W and f**

<table>
<thead>
<tr>
<th>Syntax:</th>
<th>[label] ADDWF f,d</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operands:</td>
<td>0 ≤ f ≤ 127</td>
</tr>
<tr>
<td>d ∈ [0,1]</td>
<td></td>
</tr>
<tr>
<td>Operation:</td>
<td>(W) + (f) → (dest)</td>
</tr>
<tr>
<td>Status Affected:</td>
<td>C, DC, Z</td>
</tr>
<tr>
<td>Encoding:</td>
<td>00 0111 dfff ffff</td>
</tr>
<tr>
<td>Description:</td>
<td>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'.</td>
</tr>
<tr>
<td>Words:</td>
<td>1</td>
</tr>
<tr>
<td>Cycles:</td>
<td>1</td>
</tr>
<tr>
<td>Example:</td>
<td>ADDWF REG1, 0</td>
</tr>
</tbody>
</table>

**Example**

Before Instruction
W = 0x17
REG1 = 0xC2
After Instruction
W = 0xD9
REG1 = 0xC2
Z = 0
C = 0
DC = 0

### ANDWF

**AND W with f**

<table>
<thead>
<tr>
<th>Syntax:</th>
<th>[label] ANDWF f,d</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operands:</td>
<td>0 ≤ f ≤ 127</td>
</tr>
<tr>
<td>d ∈ [0,1]</td>
<td></td>
</tr>
<tr>
<td>Operation:</td>
<td>(W) .AND. (f) → (dest)</td>
</tr>
<tr>
<td>Status Affected:</td>
<td>Z</td>
</tr>
<tr>
<td>Encoding:</td>
<td>00 0101 dfff ffff</td>
</tr>
<tr>
<td>Description:</td>
<td>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'.</td>
</tr>
<tr>
<td>Words:</td>
<td>1</td>
</tr>
<tr>
<td>Cycles:</td>
<td>1</td>
</tr>
<tr>
<td>Example:</td>
<td>ANDWF REG1, 1</td>
</tr>
</tbody>
</table>

**Example**

Before Instruction
W = 0x17
REG1 = 0xC2
After Instruction
W = 0x17
REG1 = 0x02
### BCF  Bit Clear f

**Syntax:**

```
[ label ] BCF f, b
```

**Operands:**

\[ 0 \leq f \leq 127 \]
\[ 0 \leq b \leq 7 \]

**Operation:**

\[ 0 \rightarrow (f \text{<} b) \]

**Status Affected:** None

**Encoding:**

```
01 00bb bfff ffff
```

**Description:** Bit ‘b’ in register ‘f’ is cleared.

**Words:** 1

**Cycles:** 1

**Example**

```c
BCF REG1, 7
```

**Before Instruction**

REG1 = 0xC7

**After Instruction**

REG1 = 0x47

### BSF  Bit Set f

**Syntax:**

```
[ label ] BSF f, b
```

**Operands:**

\[ 0 \leq f \leq 127 \]
\[ 0 \leq b \leq 7 \]

**Operation:**

\[ 1 \rightarrow (f \text{<} b) \]

**Status Affected:** None

**Encoding:**

```
01 01bb bfff ffff
```

**Description:** Bit ‘b’ in register ‘f’ is set.

**Words:** 1

**Cycles:** 1

**Example**

```c
BSF REG1, 7
```

**Before Instruction**

REG1 = 0xA

**After Instruction**

REG1 = 0x8A

### BTFSC  Bit Test f, Skip if Clear

**Syntax:**

```
[ label ] BTFSC f, b
```

**Operands:**

\[ 0 \leq f \leq 127 \]
\[ 0 \leq b \leq 7 \]

**Operation:**

\[ \text{skip if } (f \text{<} b) = 0 \]

**Status Affected:** None

**Encoding:**

```
01 10bb bfff ffff
```

**Description:** If bit ‘b’ in register ‘f’ is ‘0’ then the next instruction is skipped.

**Words:** 1

**Cycles:** 1(2)

**Example**

```c
HERE BTFSC REG1, 7
FALSE GOTO PROCESS_CODE
TRUE
```

**Before Instruction**

PC = address HERE

**After Instruction**

if REG<1> = 0,
PC = address TRUE
if REG<1>=1,
PC = address FALSE
BTFSS  Bit Test f, Skip if Set

Syntax:  \[ label \] BTFSS f, b
Operands:  0 ≤ f ≤ 127
          0 ≤ b < 7
Operation:  skip if (f < b) = 1
Status Affected:  None
Encoding:  \begin{array}{c} \text{01} \\ \text{11bb} \\ \text{bfff} \\ \text{ffft} \end{array}
Description:  If bit 'b' in register 'f' is '1' then the next instruction is skipped.
If bit 'b' is '1', then the next instruction fetched during the current instruction execution, is discarded and a NOP is executed instead, making this a two-cycle instruction.

CALL  Call Subroutine

Syntax:  \[ label \] CALL k
Operands:  0 ≤ k ≤ 2047
Operation:  \begin{align*}
& (\text{PC}+1) \rightarrow \text{TOS}, \\
& k \rightarrow \text{PC}<10:0>, \\
& (\text{PCLATH}<4:3>) \rightarrow \text{PC}<12:11>
\end{align*}
Status Affected:  None
Encoding:  \begin{array}{c} \text{10} \\ \text{0kkk} \\ \text{kkkk} \\ \text{kkkk} \end{array}
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.

Words:  1
Cycles:  2

Example
\begin{verbatim}
HERE CALL THERE
\end{verbatim}
Before Instruction
PC = Address HERE
After Instruction
PC = Address THERE
TOS = Address HERE+1

CLRF  Clear f

Syntax:  \[ label \] CLRF f
Operands:  0 ≤ f ≤ 127
Operation:  \begin{align*}
& 00\text{h} \rightarrow (f), \\
& 1 \rightarrow Z
\end{align*}
Status Affected:  Z
Encoding:  \begin{array}{c} \text{00} \\ \text{0001} \\ \text{ffff} \\ \text{ffft} \end{array}
Description:  The contents of register 'f' are cleared and the Z bit is set.

Words:  1
Cycles:  1

Example
\begin{verbatim}
CLRF REG1
\end{verbatim}
Before Instruction
REG1 = 0x5A
After Instruction
REG1 = 0x00
Z = 1
### CLRW
**Clear W**

*Syntax:*  
`[ label ] CLRW`

*Operands:* None

*Operation:*  
`00h → (W)`
`1 → Z`

*Status Affected:* Z

*Encoding:*  
```
00 0001 0000 0011
```

*Description:* W register is cleared. Zero bit (Z) is set.

*Words:* 1  

*Cycles:* 1

*Example*

Before Instruction  
W = 0x5A

After Instruction  
W = 0x00
Z = 1

### CLRWDT
**Clear Watchdog Timer**

*Syntax:*  
`[ label ] CLRWDT`

*Operands:* None

*Operation:*  
`00h → WDT`
`0 → WDT prescaler,`
`1 → TO`
`1 → PD`

*Status Affected:* TO, PD

*Encoding:*  
```
00 0000 0110 0100
```

*Description:* CLRWDT instruction resets the Watchdog Timer. It also resets the prescaler of the WDT. Status bits TO and PD are set.

*Words:* 1  

*Cycles:* 1

*Example*

Before Instruction  
WDT counter = ?

After Instruction  
WDT counter = 0x00
WDT prescaler = 0
TO = 1
PD = 1

### COMF
**Complement f**

*Syntax:*  
`[ label ] COMF f,d`

*Operands:*  
`0 ≤ f ≤ 127`
`d ∈ [0,1]`

*Operation:*  
`(f) → (dest)`

*Status Affected:* Z

*Encoding:*  
```
00 1001 dfff ffff
```

*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'.

*Words:* 1

*Cycles:* 1

*Example*

Before Instruction  
REG1 = 0x13

After Instruction  
REG1 = 0x13
W = 0xEC

### DECF
**Decrement f**

*Syntax:*  
`[ label ] DECF f,d`

*Operands:*  
`0 ≤ f ≤ 127`
`d ∈ [0,1]`

*Operation:*  
`(f) - 1 → (dest)`

*Status Affected:* Z

*Encoding:*  
```
00 0011 dfff ffff
```

*Description:* Decrement register 'f'. If 'd' is 0 the result is stored in W. If 'd' is 1 the result is stored back in register 'f'.

*Words:* 1

*Cycles:* 1

*Example*

Before Instruction  
CNT = 0x01
Z = 0

After Instruction  
CNT = 0x00
Z = 1
DECFSZ  Decrement f, Skip if 0

Syntax:       [ label ] DECFSZ f,d
Operands:     0 ≤ f ≤ 127
              d ∈ [0,1]
Operation:    (f) - 1 → (dest);   skip if result = 0
Status Affected: None
Encoding:     00 1011 dfff ffff
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 0, the next
instruction, which is already
fetched, is discarded. A NOP is
executed instead making it a
two-cycle instruction.

Words:        1
Cycles:       1(2)
Example

HERE     DECFSZ   REG1, 1
CONTINUE ·
·

GOTO      Unconditional Branch

Syntax:       [ label ] GOTO k
Operands:     0 ≤ k ≤ 2047
Operation:    k → PC<10:0>
              PCLATH<4:3> → PC<12:11>
Status Affected: None
Encoding:     10 1kkk kkkk kkkk
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.

Words:        1
Cycles:       2
Example

GOTO THERE

After Instruction
  PC = Address THERE
### INCF  Increment f

**Syntax:** \[ label \] INCF f,d
**Operands:** \(0 \leq f \leq 127\)
\(d \in \{0,1\}\)
**Operation:** \((f) + 1 \rightarrow (dest)\)
**Status Affected:** Z
**Encoding:**
\[
00\ 1010\ dfff\ ffff
\]
**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’.
**Words:** 1
**Cycles:** 1

**Example**

<table>
<thead>
<tr>
<th>Before Instruction</th>
<th>After Instruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>REG1 = 0xFF</td>
<td>REG1 = 0x00</td>
</tr>
<tr>
<td>Z = 0</td>
<td>Z = 1</td>
</tr>
</tbody>
</table>

### INCFSZ  Increment f, Skip if 0

**Syntax:** \[ label \] INCFSZ f,d
**Operands:** \(0 \leq f \leq 127\)
\(d \in \{0,1\}\)
**Operation:** \((f) + 1 \rightarrow (dest)\), skip if result = 0
**Status Affected:** None
**Encoding:**
\[
00\ 1111\ dfff\ ffff
\]
**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’.
If the result is 0, the next instruction, which is already fetched, is discarded. A NOP is executed instead making it a two-cycle instruction.
**Words:** 1
**Cycles:** 1(2)

**Example**

<table>
<thead>
<tr>
<th>Before Instruction</th>
<th>After Instruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>PC = address HERE</td>
<td>PC = address HERE</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CONTINUE</th>
<th>LOOP</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Before Instruction</th>
<th>After Instruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>REG1 = REG1 + 1</td>
<td>REG1 = REG1 + 1</td>
</tr>
<tr>
<td>if CNT = 0,</td>
<td></td>
</tr>
<tr>
<td>PC = address CONTINUE</td>
<td></td>
</tr>
<tr>
<td>if REG1 ≠ 0,</td>
<td></td>
</tr>
<tr>
<td>PC = address HERE +1</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>HERE</th>
<th>INCFSZ</th>
<th>REG1, 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>GOTO</td>
<td>LOOP</td>
<td></td>
</tr>
<tr>
<td>CONTINUE</td>
<td>•</td>
<td></td>
</tr>
<tr>
<td>•</td>
<td>•</td>
<td></td>
</tr>
</tbody>
</table>
**IORLW**  
Inclusive OR Literal with W

| Syntax: | [label] IORLW k |
| Operands: | 0 ≤ k ≤ 255 |
| Operation: | (W) .OR. k → (W) |
| Status Affected: | Z |
| Encoding: | 11 1000 kkkk kkkk |
| Description: | The contents of the W register is OR'ed with the eight bit literal 'k'. The result is placed in the W register. |
| Words: | 1 |
| Cycles: | 1 |
| Example | IORLW 0x35 |

**Example**

**Before Instruction**

W = 0x9A

**After Instruction**

W = 0xBF

Z = 0

---

**IORWF**  
Inclusive OR W with f

| Syntax: | [label] IORWF f,d |
| Operands: | 0 ≤ f ≤ 127 |
| d ∈ [0,1] |
| Operation: | (W) .OR. (f) → (dest) |
| Status Affected: | Z |
| Encoding: | 00 0100 dfff ffff |
| Description: | Inclusive OR the W register with register 'f'. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is placed back in register 'f'. |
| Words: | 1 |
| Cycles: | 1 |
| Example | IORWF REG1, 0 |

**Example**

**Before Instruction**

REG1 = 0x13
W = 0x91

**After Instruction**

REG1 = 0x13
W = 0x93
Z = 1

---

**MOVLW**  
Move Literal to W

| Syntax: | [label] MOVLW k |
| Operands: | 0 ≤ k ≤ 255 |
| Operation: | k → (W) |
| Status Affected: | None |
| Encoding: | 11 00xx kkkk kkkk |
| 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 |

**Example**

After Instruction

W = 0x5A

---

**MOVF**  
Move f

| Syntax: | [label] MOVF f,d |
| Operands: | 0 ≤ f ≤ 127 |
| d ∈ [0,1] |
| Operation: | (f) → (dest) |
| Status Affected: | Z |
| Encoding: | 00 1000 dfff ffff |
| 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 REG1, 0 |

**Example**

After Instruction

W = value in REG1 register
Z = 1
MOVWF Move W to f

Syntax: [label] MOVWF f
Operands: 0 ≤ f ≤ 127
Operation: (W) → (f)
Status Affected: None
Encoding: 00 0000 1fff ffff
Description: Move data from W register to register ‘f’.
Words: 1
Cycles: 1
Example

<table>
<thead>
<tr>
<th>Before Instruction</th>
<th>After Instruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>REG1 = 0xFF</td>
<td>REG1 = 0x4F</td>
</tr>
<tr>
<td>W = 0x4F</td>
<td>W = 0x4F</td>
</tr>
</tbody>
</table>

NOP No Operation

Syntax: [label] NOP
Operands: None
Operation: No operation
Status Affected: None
Encoding: 00 0000 0xx0 0000
Description: No operation.
Words: 1
Cycles: 1
Example

OPTION Load Option Register

Syntax: [label] OPTION
Operands: None
Operation: (W) → OPTION
Status Affected: None
Encoding: 00 0000 0110 0010
Description: The contents of the W register are loaded in the OPTION register. This instruction is supported for code compatibility with PIC16C5X products. Since OPTION is a readable/writable register, the user can directly address it. Using only register instruction such as MOVWF.
Words: 1
Cycles: 1
Example

To maintain upward compatibility with future PICmicro® products, do not use this instruction.

RETFIE Return from Interrupt

Syntax: [label] RETFIE
Operands: None
Operation: TOS → PC,
GIE = 1
Status Affected: None
Encoding: 00 0000 0000 1001
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**

<table>
<thead>
<tr>
<th>Syntax:</th>
<th>[label] RETLW k</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operands:</td>
<td>0 ≤ k ≤ 255</td>
</tr>
<tr>
<td>Operation:</td>
<td>k → (W); TOS → PC</td>
</tr>
<tr>
<td>Status Affected:</td>
<td>None</td>
</tr>
<tr>
<td>Encoding:</td>
<td>11 01xx kkkk kkkk</td>
</tr>
<tr>
<td>Description:</td>
<td>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.</td>
</tr>
</tbody>
</table>

**Example**

```assembly
CALL TABLE; W contains table offset value
  ; W now has table value
  ...

TABLE
  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

<table>
<thead>
<tr>
<th>Words:</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cycles:</td>
<td>2</td>
</tr>
</tbody>
</table>

### RLF

**Rotate Left f through Carry**

<table>
<thead>
<tr>
<th>Syntax:</th>
<th>[label] RLF f,d</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operands:</td>
<td>0 ≤ f ≤ 127</td>
</tr>
<tr>
<td>d ∈ [0,1]</td>
<td></td>
</tr>
<tr>
<td>Operation:</td>
<td>See description below</td>
</tr>
<tr>
<td>Status Affected:</td>
<td>C</td>
</tr>
<tr>
<td>Encoding:</td>
<td>00 1101 dfff ffff</td>
</tr>
<tr>
<td>Description:</td>
<td>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'.</td>
</tr>
</tbody>
</table>

**Example**

```assembly
RLF REG1, 0
```

**Before Instruction**
- REG1 = 1110 0110
- C = 0

**After Instruction**
- REG1 = 1110 0110
- W = 1100 1100
- C = 1

### RETURN

**Return from Subroutine**

<table>
<thead>
<tr>
<th>Syntax:</th>
<th>[label] RETURN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operands:</td>
<td>None</td>
</tr>
<tr>
<td>Operation:</td>
<td>TOS → PC</td>
</tr>
<tr>
<td>Status Affected:</td>
<td>None</td>
</tr>
<tr>
<td>Encoding:</td>
<td>00 0000 0000 1000</td>
</tr>
<tr>
<td>Description:</td>
<td>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.</td>
</tr>
</tbody>
</table>

**Example**

```assembly
RETURN
```

**After Interrupt**
- PC = TOS
## RRF
**Rotate Right f through Carry**

<table>
<thead>
<tr>
<th>Syntax:</th>
<th>[ label ] RRF f,d</th>
</tr>
</thead>
</table>
| Operands: | \(0 \leq f \leq 127\)  
| | \(d \in [0,1]\) |
| Operation: | See description below |
| Status Affected: | C |
| Encoding: | 00 1100 dfff ffff |
| Description: | The contents of register \(f\) are rotated one bit to the right through the Carry Flag. If \(d\) is 0 the result is placed in the W register. If \(d\) is 1 the result is placed back in register \(f\). |
| Words: | 1 |
| Cycles: | 1 |
| Example: | RRF REG1, 0 |

**Example**

**Before Instruction**

| REG1 = 1110 0110 |
| C = 0 |

**After Instruction**

| REG1 = 1110 0110 |
| W = 0111 0011 |
| C = 0 |

## SLEEP
**Subtract W from Literal**

<table>
<thead>
<tr>
<th>Syntax:</th>
<th>[ label ] SUBLW k</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operands:</td>
<td>(0 \leq k \leq 255)</td>
</tr>
<tr>
<td>Operation:</td>
<td>(k - (W) \rightarrow (W))</td>
</tr>
<tr>
<td>Status Affected:</td>
<td>C, DC, Z</td>
</tr>
<tr>
<td>Encoding:</td>
<td>11110x kkkk kkkk</td>
</tr>
<tr>
<td>Description:</td>
<td>The W register is subtracted (2's complement method) from the eight bit literal (k). The result is placed in the W register.</td>
</tr>
<tr>
<td>Words:</td>
<td>1</td>
</tr>
<tr>
<td>Cycles:</td>
<td>1</td>
</tr>
<tr>
<td>Example 1:</td>
<td>SUBLW 0x02</td>
</tr>
</tbody>
</table>

**Before Instruction**

| W = 1 |
| C = ? |

**After Instruction**

| W = 1 |
| C = 1; result is positive |

**Example 2: Before Instruction**

| W = 2 |
| C = ? |

**After Instruction**

| W = 0 |
| C = 1; result is zero |

**Example 3: Before Instruction**

| W = 3 |
| C = ? |

**After Instruction**

| W = 0xFF |
| C = 0; result is negative |
### SUBWF
**Subtract W from f**

- **Syntax:** \([\text{label}]\) SUBWF \(f,d\)
- **Operands:** \(0 \leq f \leq 127\)
- **d \in [0,1]\)
- **Operation:** \((f) - (W) \rightarrow (\text{dest})\)
- **Status Affected:** C, DC, Z

<table>
<thead>
<tr>
<th>Encoding</th>
<th>d</th>
<th>ffff</th>
<th>dfff</th>
</tr>
</thead>
<tbody>
<tr>
<td>00 0010</td>
<td>d</td>
<td>ffff</td>
<td>dfff</td>
</tr>
</tbody>
</table>

**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'.

**Words:** 1
**Cycles:** 1

**Example 1:** SUBWF REG1, 1

**Before Instruction**
- REG1 = 3
- W = 2
- C = ?

**After Instruction**
- REG1 = 1
- W = 2
- C = 1; result is positive
- Z = 0
- DC = 1

**Example 2:** Before Instruction
- REG1 = 2
- W = 2
- C = ?

**After Instruction**
- REG1 = 0
- W = 2
- C = 1; result is zero
- Z = DC = 1

**Example 3:** Before Instruction
- REG1 = 1
- W = 2
- C = ?

**After Instruction**
- REG1 = 0xFF
- W = 2
- C = 0; result is negative
- Z = DC = 0

---

### SWAPF
**Swap Nibbles in f**

- **Syntax:** \([\text{label}]\) SWAPF \(f,d\)
- **Operands:** \(0 \leq f \leq 127\)
- **d \in [0,1]\)
- **Operation:** \((f<3:0>) \rightarrow (\text{dest}<7:4>), (f<7:4>) \rightarrow (\text{dest}<3:0>)\)

**Status Affected:** None

<table>
<thead>
<tr>
<th>Encoding</th>
<th>d</th>
<th>ffff</th>
<th>dfff</th>
</tr>
</thead>
<tbody>
<tr>
<td>00 1110</td>
<td>d</td>
<td>ffff</td>
<td>dfff</td>
</tr>
</tbody>
</table>

**Description:** The upper and lower nibbles of register 'f' are exchanged. If 'd' is 0 the result is placed in W register. If 'd' is 1 the result is placed in register 'f'.

**Words:** 1
**Cycles:** 1

**Example** SWAPF REG1, 0

**Before Instruction**
- REG1 = 0xA5

**After Instruction**
- REG1 = 0xA5
- W = 0x5A

---

### TRIS
**Load TRIS Register**

- **Syntax:** \([\text{label}]\) TRIS \(f\)
- **Operands:** \(5 \leq f \leq 6\)
- **Operation:** \((W) \rightarrow \text{TRIS register } f;\)

**Status Affected:** None

<table>
<thead>
<tr>
<th>Encoding</th>
<th>0fff</th>
<th>0110</th>
<th>0000</th>
<th>0000</th>
</tr>
</thead>
</table>

**Description:** The instruction is supported for code compatibility with the PIC16C5X products. Since TRIS registers are readable and writable, the user can directly address them.

**Words:** 1
**Cycles:** 1

**Example**

To maintain upward compatibility with future PICmicro® products, do not use this instruction.
### XORLW

**Exclusive OR Literal with W**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>[label] XORLW k</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operands</td>
<td>0 ≤ k ≤ 255</td>
</tr>
<tr>
<td>Operation</td>
<td>(W) .XOR. k → (W)</td>
</tr>
<tr>
<td>Status Affected</td>
<td>Z</td>
</tr>
<tr>
<td>Encoding</td>
<td>11 1010 kkkk kkkk</td>
</tr>
<tr>
<td>Description</td>
<td>The contents of the W register are XOR'ed with the eight bit literal 'k'. The result is placed in the W register.</td>
</tr>
<tr>
<td>Words</td>
<td>1</td>
</tr>
<tr>
<td>Cycles</td>
<td>1</td>
</tr>
<tr>
<td>Example</td>
<td>XORLW 0xAF</td>
</tr>
</tbody>
</table>

**Example**

*Before Instruction*

\[
W = 0xB5
\]

*After Instruction*

\[
W = 0x1A
\]

### XORWF

**Exclusive OR W with f**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>[label] XORWF f,d</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operands</td>
<td>0 ≤ f ≤ 127</td>
</tr>
<tr>
<td>d ∈ [0,1]</td>
<td></td>
</tr>
<tr>
<td>Operation</td>
<td>(W) .XOR. (f) → (dest)</td>
</tr>
<tr>
<td>Status Affected</td>
<td>Z</td>
</tr>
<tr>
<td>Encoding</td>
<td>00 0110 dfff ffff</td>
</tr>
<tr>
<td>Description</td>
<td>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'.</td>
</tr>
<tr>
<td>Words</td>
<td>1</td>
</tr>
<tr>
<td>Cycles</td>
<td>1</td>
</tr>
<tr>
<td>Example</td>
<td>XORWF REG1, 1</td>
</tr>
</tbody>
</table>

**Example**

*Before Instruction*

\[
REG1 = 0xAF \\
W = 0xB5
\]

*After Instruction*

\[
REG1 = 0x1A \\
W = 0xB5
\]
11.0 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 C17 and MPLAB C18 C Compilers
  - MPLINK™ Object Linker/
    MPLIB™ Object Librarian
  - MPLAB C30 C Compiler
  - MPLAB ASM30 Assembler/Linker/Library
- Simulators
  - MPLAB SIM Software Simulator
  - MPLAB dsPIC30 Software Simulator
- Emulators
  - MPLAB ICE 2000 In-Circuit Emulator
  - MPLAB ICE 4000 In-Circuit Emulator
- In-Circuit Debugger
  - MPLAB ICD 2
- Device Programmers
  - PRO MATE® II Universal Device Programmer
  - PICSTART® Plus Development Programmer
- Low Cost Demonstration Boards
  - PICDEM™ 1 Demonstration Board
  - PICDEM.net™ Demonstration Board
  - PICDEM 2 Plus Demonstration Board
  - PICDEM 3 Demonstration Board
  - PICDEM 4 Demonstration Board
  - PICDEM 17 Demonstration Board
  - PICDEM 18R Demonstration Board
  - PICDEM LIN Demonstration Board
  - PICDEM USB Demonstration Board
- Evaluation Kits
  - KEELoo®
  - PICDEM MSC
  - microID®
  - CAN
  - PowerSmart®
  - Analog

11.1 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® based application that contains:

- An interface to 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
- Mouse over variable inspection
- Extensive on-line help

The MPLAB IDE allows you to:

- Edit your source files (either assembly or C)
- One touch assemble (or compile) and download to PICmicro emulator and simulator tools (automatically updates all project information)
- Debug using:
  - source files (assembly or C)
  - absolute listing file (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 increasing flexibility and power.

11.2 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
11.3 MPLAB C17 and MPLAB C18 C Compilers

The MPLAB C17 and MPLAB C18 Code Development Systems are complete ANSI C compilers for Microchip's PIC17CXXX and PIC18CXXX family of microcontrollers. 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.

11.4 MPLINK Object Linker/
MPLIB Object Librarian

The MPLINK object linker combines relocatable objects created by the MPASM assembler and the MPLAB C17 and MPLAB C18 C compilers. It can link relocatable objects from pre-compiled libraries, using directives from a linker script.

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

The object linker/library features include:

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

11.5 MPLAB C30 C Compiler

The MPLAB C30 C compiler is a full-featured, ANSI compliant, optimizing compiler that translates standard ANSI C programs into dsPIC30F assembly language source. The compiler also supports many command-line options and language extensions to take full advantage of the dsPIC30F device hardware capabilities, and afford fine control of the compiler code generator.

MPLAB C30 is distributed with a complete ANSI C standard library. All library functions have been validated and conform to the ANSI C library standard. The library includes functions for string manipulation, dynamic memory allocation, data conversion, time-keeping, and math functions (trigonometric, exponential and hyperbolic). The compiler provides symbolic information for high level source debugging with the MPLAB IDE.

11.6 MPLAB ASM30 Assembler, Linker, and Librarian

MPLAB ASM30 assembler produces relocatable machine code from symbolic assembly language for dsPIC30F devices. MPLAB C30 compiler uses the assembler to produce it's 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

11.7 MPLAB SIM Software Simulator

The MPLAB SIM software simulator allows code development in a PC hosted environment by simulating the PICmicro series microcontrollers on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a file, or user defined key press, to any pin. The execution can be performed in Single-Step, Execute Until Break, or Trace mode.

The MPLAB SIM simulator fully supports symbolic debugging using the MPLAB C17 and MPLAB C18 C Compilers, as well as the MPASM assembler. The software simulator offers the flexibility to develop and debug code outside of the laboratory environment, making it an excellent, economical software development tool.

11.8 MPLAB SIM30 Software Simulator

The MPLAB SIM30 software simulator allows code development in a PC hosted environment by simulating the dsPIC30F series microcontrollers on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a file, or user defined key press, to any of the pins.

The MPLAB SIM30 simulator fully supports symbolic debugging using the MPLAB C30 C Compiler and MPLAB ASM30 assembler. The simulator runs in either a Command Line mode for automated tasks, or from MPLAB IDE. This high speed simulator is designed to debug, analyze and optimize time intensive DSP routines.
11.9 MPLAB ICE 2000
High Performance Universal In-Circuit Emulator

The MPLAB ICE 2000 universal 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 universal architecture of the MPLAB ICE 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.

11.10 MPLAB ICE 4000
High Performance Universal In-Circuit Emulator

The MPLAB ICE 4000 universal in-circuit emulator is intended to provide the product development engineer with a complete microcontroller design tool set for high-end PICmicro microcontrollers. Software control of the MPLAB ICE 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, up to 2 Mb of emulation memory, and the ability to view variables in real-time.

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.

11.11 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 MCU and can be used to develop these and other PICmicro microcontrollers. 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, 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.

11.12 PRO MATE II Universal Device Programmer

The PRO MATE II is a universal, CE compliant device programmer with programmable voltage verification at VDDMIN and VDDMAX for maximum reliability. It features an LCD display for instructions and error messages and a modular detachable socket assembly to support various package types. In Stand-alone mode, the PRO MATE II device programmer can read, verify, and program PICmicro devices without a PC connection. It can also set code protection in this mode.

11.13 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 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.
11.14 PICDEM 1 PICmicro Demonstration Board

The PICDEM 1 demonstration board demonstrates the capabilities of the PIC16C5X (PIC16C54 to PIC16C58A), PIC16C61, PIC16C62X, PIC16C71, PIC16C8X, PIC17C42, PIC17C43 and PIC17C44. All necessary hardware and software is included to run basic demo programs. The sample microcontrollers provided with the PICDEM 1 demonstration board can be programmed with a PRO MATE II device programmer, or a PICSTART Plus development programmer. The PICDEM 1 demonstration board can be connected to the MPLAB ICE in-circuit emulator for testing. A prototype area extends the circuitry for additional application components. Features include an RS-232 interface, a potentiometer for simulated analog input, push button switches and eight LEDs.

11.15 PICDEM.net Internet/Ethernet Demonstration Board

The PICDEM.net demonstration board is an Internet/Ethernet demonstration board using the PIC18F452 microcontroller and TCP/IP firmware. The board supports any 40-pin DIP device that conforms to the standard pinout used by the PIC16F877 or PIC18C452. This kit features a user friendly TCP/IP stack, a 24L256 Serial EEPROM for Xmodem download to web pages into Serial EEPROM, ICSP/MPLAB ICD 2 interface connector, an Ethernet interface, RS-232 interface, and a 16 x 2 LCD display. Also included is the book and CD-ROM “TCP/IP Lean, Web Servers for Embedded Systems,” by Jeremy Bentham.

11.16 PICDEM 2 Plus Demonstration Board

The PICDEM 2 Plus demonstration board supports many 18-, 28-, and 40-pin microcontrollers, including PIC16F87X and PIC16FXX2 devices. All the necessary hardware and software is included to run the demonstration programs. The sample microcontrollers provided with the PICDEM 2 demonstration board can be programmed with a PRO MATE II device programmer, PICSTART Plus development programmer, or MPLAB ICD 2 with a Universal Programmer Adapter. The MPLAB ICD 2 and MPLAB ICE in-circuit emulators may also be used with the PICDEM 2 demonstration board to test firmware. A prototype area extends the circuitry for additional application components. Some of the features include an RS-232 interface, a 2 x 16 LCD display, a piezo speaker, an on-board temperature sensor, four LEDs, and sample PIC18F452 and PIC16F877 Flash microcontrollers.

11.17 PICDEM 3 PIC16C92X Demonstration Board

The PICDEM 3 demonstration board supports the PIC16C923 and PIC16C924 in the PLCC package. All the necessary hardware and software is included to run the demonstration programs.

11.18 PICDEM 4 8/14/18-Pin Demonstration Board

The PICDEM 4 can be used to demonstrate the capabilities of the 8-, 14-, and 18-pin PIC16XXXX and PIC18XXXX MCUs, including the PIC16F818/819, PIC16F87/88, PIC16F62XA and the PIC18F1320 family of microcontrollers. PICDEM 4 is intended to showcase the many features of these low pin count parts, including LIN and Motor Control using ECCP. Special provisions are made for low power operation with the super capacitor circuit, and jumpers allow on-board hardware to be disabled to eliminate current draw in this mode. Included on the demo board are provisions for Crystal, RC or Canned Oscillator modes, a five volt regulator for use with a nine volt wall adapter or battery, DB-9 RS-232 interface, ICD connector for programming via ICSP and development with MPLAB ICD 2, 2x16 liquid crystal display, PCB footprints for H-Bridge motor driver, LIN transceiver and EEPROM. Also included are: header for expansion, eight LEDs, four potentiometers, three push buttons and a prototyping area. Included with the kit is a PIC16F627A and a PIC18F1320. Tutorial firmware is included along with the User’s Guide.

11.19 PICDEM 17 Demonstration Board

The PICDEM 17 demonstration board is an evaluation board that demonstrates the capabilities of several Microchip microcontrollers, including PIC17C752, PIC17C756A, PIC17C762 and PIC17C766. A programmed sample is included. The PRO MATE II device programmer, or the PICSTART Plus development programmer, can be used to reprogram the device for user tailored application development. The PICDEM 17 demonstration board supports program download and execution from external on-board Flash memory. A generous prototype area is available for user hardware expansion.
11.20 PICDEM 18R PIC18C601/801 Demonstration Board

The PICDEM 18R demonstration board serves to assist development of the PIC18C601/801 family of Microchip microcontrollers. It provides hardware implementation of both 8-bit Multiplexed/De-multiplexed and 16-bit memory modes. The board includes 2 Mb external Flash memory and 128 Kb SRAM memory, as well as serial EEPROM, allowing access to the wide range of memory types supported by the PIC18C601/801.

11.21 PICDEM LIN PIC16C43X Demonstration Board

The powerful LIN hardware and software kit includes a series of boards and three PICmicro microcontrollers. The small footprint PIC16C432 and PIC16C433 are used as slaves in the LIN communication and feature on-board LIN transceivers. A PIC16F874 Flash microcontroller serves as the master. All three microcontrollers are programmed with firmware to provide LIN bus communication.

11.22 PICkit™ 1 Flash Starter Kit

A complete "development system in a box", the PICkit Flash Starter Kit includes a convenient multi-section board for programming, evaluation, and development of 8/14-pin Flash PIC® microcontrollers. Powered via USB, the board operates under a simple Windows GUI. The PICkit 1 Starter Kit includes the user's guide (on CD ROM), PICkit 1 tutorial software and code for various applications. Also included are MPLAB® IDE (Integrated Development Environment) software, software and hardware “Tips ‘n Tricks for 8-pin Flash PIC® Microcontrollers” Handbook and a USB Interface Cable. Supports all current 8/14-pin Flash PIC microcontrollers, as well as many future planned devices.

11.23 PICDEM USB PIC16C7X5 Demonstration Board

The PICDEM USB Demonstration Board shows off the capabilities of the PIC16C745 and PIC16C765 USB microcontrollers. This board provides the basis for future USB products.

11.24 Evaluation and Programming Tools

In addition to the PICDEM series of circuits, Microchip has a line of evaluation kits and demonstration software for these products.

- KEELoQ evaluation and programming tools for Microchip’s HCS Secure Data Products
- CAN developers kit for automotive network applications
- Analog design boards and filter design software
- PowerSmart battery charging evaluation/calibration kits
- IrDA® development kit
- microID development and rfLab™ development software
- SEEVAL® designer kit for memory evaluation and endurance calculations
- PICDEM MSC demo boards for Switching mode power supply, high power IR driver, delta sigma ADC, and flow rate sensor

Check the Microchip web page and the latest Product Line Card for the complete list of demonstration and evaluation kits.
12.0 ELECTRICAL CHARACTERISTICS

Absolute Maximum Ratings(†)

Ambient temperature under bias ........................................................................................................... -55°C to +125°C
Storage temperature .................................................................................................................................. -65°C to +150°C
Voltage on any pin with respect to VSS (except VDD, MCLR, and RA4) ........................................... -0.3V to (VDD +0.3V)
Voltage on VDD with respect to VSS ...................................................................................................... -0.3V to +7.5V
Voltage on MCLR with respect to VSS (Note 2)................................................................................... 0V to +13.25V
Voltage on RA4 with respect to Vss ........................................................................................................ 0V to +8.5V
Total power dissipation (Note 1) (PDIP and SOIC)............................................................................. 1.0W
Total power dissipation (Note 1) (SSOP) ............................................................................................. 0.65W
Maximum current out of VSS pin .............................................................................................................. 300 mA
Maximum current into VDD pin ............................................................................................................... 250 mA
Input clamp current, IIC (V<sub>I</sub> < 0 or V<sub>I</sub> > VDD)........................................................................... ±20 mA
Output clamp current, IOC (V<sub>O</sub> < 0 or V<sub>O</sub> > 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 PORTB (combined)................................................................. 200 mA
Maximum current sourced by PORTA and PORTB (combined)........................................................... 200 mA

Note 1: Power dissipation is calculated as follows: Pdis = VDD x (IDD - ∑ IOH) + ∑ (VDD-VOH) x IOH + ∑ (VOL x IOL)

2: Voltage spikes below VSS at the MCLR/VPP pin, inducing currents greater than 80 mA, may cause latch-up. Thus, a series resistor of 50-100Ω should be used when applying a “low” level to the MCLR/VPP pin rather than pulling this pin directly to VSS.

† NOTICE: Stresses above those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.
FIGURE 12-1: PIC16F716 VOLTAGE-FREQUENCY GRAPH, -40°C ≤ TA ≤ +85°C(1)

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

FIGURE 12-2: PIC16F716 VOLTAGE-FREQUENCY GRAPH, 85°C ≤ TA ≤ +125°C(1)

Note 1: The shaded region indicates the permissible combinations of voltage and frequency.
### 12.1 DC Characteristics: PIC16F716 (Industrial, 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>D001</td>
<td>VDD</td>
<td>Supply Voltage</td>
<td>2.0</td>
<td>—</td>
<td>5.5</td>
<td>V</td>
<td>Industrial</td>
</tr>
<tr>
<td>D001A</td>
<td></td>
<td></td>
<td>3.0</td>
<td>—</td>
<td>5.5</td>
<td>V</td>
<td>Extended</td>
</tr>
<tr>
<td>D002*</td>
<td>VDR</td>
<td>RAM Data Retention Voltage(1)</td>
<td>—</td>
<td>1.5*</td>
<td>—</td>
<td>V</td>
<td></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 on Power-on Reset 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>TBD</td>
<td>—</td>
<td>V/ms</td>
<td>PWRT enabled (PWRT bit clear)</td>
</tr>
<tr>
<td>D004A*</td>
<td></td>
<td></td>
<td></td>
<td>TBD</td>
<td>—</td>
<td>—</td>
<td>PWRT disabled (PWRT bit set) See section on Power-on Reset for details</td>
</tr>
<tr>
<td>D005</td>
<td>VBOR</td>
<td>Brown-out Reset voltage trip point</td>
<td>3.65</td>
<td>4.0</td>
<td>4.35</td>
<td>V</td>
<td>BOREN bit set, BOR bit = ‘1’</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TBD</td>
<td>2.5</td>
<td>TBD</td>
<td>V</td>
<td>BOREN bit set, BOR bit = ‘0’</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:** This is the limit to which VDD can be lowered without losing RAM data.
### 12.2 DC Characteristics: PIC16F716 (Industrial)

#### 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>VDD</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>—</td>
<td>—</td>
</tr>
<tr>
<td>D020</td>
<td>IPD</td>
<td>Power-down Base Current</td>
<td></td>
<td>—</td>
<td>0.1</td>
<td>μA</td>
<td>2.0</td>
<td>WDT, BOR and T1OSC: disabled</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>—</td>
<td>0.8</td>
<td>μA</td>
<td>—</td>
<td>WDT, BOR and T1OSC: disabled</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>—</td>
<td>0.85</td>
<td>μA</td>
<td>3.0</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>—</td>
<td>2.7</td>
<td>μA</td>
<td>5.0</td>
<td>—</td>
</tr>
<tr>
<td>D021</td>
<td>△IMOD</td>
<td>Peripheral Module Current(1)</td>
<td></td>
<td>—</td>
<td>1</td>
<td>2.0</td>
<td>μA</td>
<td>2.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>—</td>
<td>3.5</td>
<td>μA</td>
<td>3.0</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>—</td>
<td>13.5</td>
<td>μA</td>
<td>5.0</td>
<td>—</td>
</tr>
<tr>
<td>D022</td>
<td></td>
<td>TBD TBD μA</td>
<td>3.0</td>
<td>BOR Current</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>—</td>
<td>40</td>
<td>μA</td>
<td>4.5</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>—</td>
<td>45</td>
<td>μA</td>
<td>5.0</td>
<td>—</td>
</tr>
<tr>
<td>D025</td>
<td></td>
<td>1.8 TBD μA</td>
<td>2.0</td>
<td>T1osc Current</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>—</td>
<td>2.6</td>
<td>μA</td>
<td>3.0</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>—</td>
<td>3.0</td>
<td>μA</td>
<td>5.0</td>
<td>—</td>
</tr>
<tr>
<td>D010</td>
<td>IDD</td>
<td>Supply Current</td>
<td></td>
<td>—</td>
<td>14</td>
<td>μA</td>
<td>2.0</td>
<td>Fosc = 32 kHz</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>—</td>
<td>23</td>
<td>μA</td>
<td>3.0</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>—</td>
<td>45</td>
<td>μA</td>
<td>5.0</td>
<td>—</td>
</tr>
<tr>
<td>D011</td>
<td></td>
<td>120 160 μA</td>
<td>2.0</td>
<td>Fosc = 1 MHz</td>
<td>XT Oscillator mode</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>—</td>
<td>180</td>
<td>μA</td>
<td>3.0</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>—</td>
<td>290</td>
<td>μA</td>
<td>5.0</td>
<td>—</td>
</tr>
<tr>
<td>D012</td>
<td></td>
<td>220 300 μA</td>
<td>2.0</td>
<td>Fosc = 4 MHz</td>
<td>XT Oscillator mode</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>—</td>
<td>350</td>
<td>μA</td>
<td>3.0</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>—</td>
<td>600</td>
<td>μA</td>
<td>5.0</td>
<td>—</td>
</tr>
<tr>
<td>D013</td>
<td></td>
<td>2.1 2.9 mA</td>
<td>4.5</td>
<td>Fosc = 20 MHz</td>
<td>HS Oscillator mode</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>—</td>
<td>2.5</td>
<td>mA</td>
<td>5.0</td>
<td>—</td>
</tr>
</tbody>
</table>

† 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:** The “△” current is the additional current consumed when this peripheral is enabled. This current should be added to the base IDD or IPD measurement. Max values should be used when calculating total current consumption.

**Note 2:** ADC on, not converting.
12.3  DC Characteristics: PIC16F716 (Extended)

<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>V DD</th>
<th>Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>D001</td>
<td>V DD</td>
<td>Supply Voltage</td>
<td>3.0</td>
<td>—</td>
<td>5.5</td>
<td>V</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>D020E</td>
<td>IPD</td>
<td>Power-down Base Current</td>
<td>—</td>
<td>0.1 TBD</td>
<td>µA</td>
<td>3.0</td>
<td>WDT, BOR and T1OSC: disabled</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.2 TBD</td>
<td>µA</td>
<td>5.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D021E</td>
<td>ΔIMOD</td>
<td>Peripheral Module Current(1)</td>
<td>—</td>
<td>2 TBD</td>
<td>µA</td>
<td>3.0</td>
<td>WDT Current</td>
<td></td>
</tr>
<tr>
<td>D022E</td>
<td></td>
<td></td>
<td></td>
<td>9 TBD</td>
<td>µA</td>
<td>5.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D025E</td>
<td></td>
<td></td>
<td></td>
<td>40 TBD</td>
<td>µA</td>
<td>4.5</td>
<td>BOR Current</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>45 TBD</td>
<td>µA</td>
<td>5.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D010E</td>
<td>IDD</td>
<td>Supply Current</td>
<td>—</td>
<td>21 TBD</td>
<td>µA</td>
<td>3.0</td>
<td>FOSC = 32 kHz LP Oscillator mode</td>
<td></td>
</tr>
<tr>
<td>D011E</td>
<td></td>
<td></td>
<td></td>
<td>38 TBD</td>
<td>µA</td>
<td>5.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D012E</td>
<td></td>
<td></td>
<td></td>
<td>182 TBD</td>
<td>µA</td>
<td>3.0</td>
<td>FOSC = 1 MHz XT Oscillator mode</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>293 TBD</td>
<td>µA</td>
<td>5.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D013E</td>
<td></td>
<td></td>
<td></td>
<td>371 TBD</td>
<td>µA</td>
<td>3.0</td>
<td>FOSC = 4 MHz XT Oscillator mode</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>668 TBD</td>
<td>µA</td>
<td>5.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2.6 TBD</td>
<td>mA</td>
<td>4.5</td>
<td>FOSC = 20 MHz HS Oscillator mode</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3 TBD</td>
<td>mA</td>
<td>5.0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

† 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:** The “Δ” current is the additional current consumed when this peripheral is enabled. This current should be added to the base IDD or IPD measurement. Max values should be used when calculating total current consumption.

2: ADC on, not converting.
12.4 DC Characteristics: PIC16F716 (Industrial, Extended)

<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>D030</td>
<td>VIL</td>
<td>Input Low Voltage</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>0.15VDD</td>
<td>V</td>
<td>otherwise</td>
</tr>
<tr>
<td>D031</td>
<td></td>
<td>with Schmitt Trigger buffer</td>
<td>Vss</td>
<td>—</td>
<td>0.2VDD</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td>D032</td>
<td>MCLR, OSC1 (in RC mode)</td>
<td>Vss</td>
<td>—</td>
<td>0.2VDD</td>
<td>V</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D033</td>
<td>OSC1 (in HS mode)</td>
<td>Vss</td>
<td>—</td>
<td>0.3VDD</td>
<td>V</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>OSC1 (in XT and LP modes)</td>
<td>Vss</td>
<td>—</td>
<td>0.6V</td>
<td>V</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D040</td>
<td>VIH</td>
<td>Input High Voltage</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>V</td>
<td>4.5V ≤ VDD ≤ 5.5V</td>
</tr>
<tr>
<td>D040A</td>
<td></td>
<td>with TTL buffer</td>
<td>2.0</td>
<td>—</td>
<td>VDD</td>
<td>V</td>
<td>otherwise</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.25VDD + 0.8V</td>
<td>—</td>
<td>VDD</td>
<td>—</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td>D041</td>
<td></td>
<td>with Schmitt Trigger buffer</td>
<td>0.8VDD</td>
<td>—</td>
<td>VDD</td>
<td>V</td>
<td>For entire VDD range</td>
</tr>
<tr>
<td>D042</td>
<td>MCLR</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td>D042A</td>
<td>OSC1 (XT, HS and LP modes)</td>
<td>0.7VDD</td>
<td>—</td>
<td>VDD</td>
<td>—</td>
<td>V</td>
<td>(Note1)</td>
</tr>
<tr>
<td>D043</td>
<td>OSC1 (in RC mode)</td>
<td>0.9VDD</td>
<td>—</td>
<td>VDD</td>
<td>—</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td>D060</td>
<td>IL</td>
<td>Input Leakage Current(2), (3)</td>
<td>—</td>
<td>—</td>
<td>±1</td>
<td>µA</td>
<td>Vss ≤ VPIN ≤ VDD, Pin at high-impedance</td>
</tr>
<tr>
<td></td>
<td></td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>±500</td>
<td>nA</td>
<td>Vss ≤ VPIN ≤ VDD, Pin configured as analog input</td>
</tr>
<tr>
<td>D061</td>
<td></td>
<td>MCLR, RA4/T0CKI</td>
<td>—</td>
<td>—</td>
<td>±5</td>
<td>µA</td>
<td>Vss ≤ VPIN ≤ VDD</td>
</tr>
<tr>
<td>D063</td>
<td></td>
<td>OSC1/CLKIN</td>
<td>—</td>
<td>—</td>
<td>±5</td>
<td>µA</td>
<td>Vss ≤ VPIN ≤ VDD, XT, HS and LP osc modes</td>
</tr>
<tr>
<td>D070</td>
<td>IPURB</td>
<td>PORTB weak pull-up current</td>
<td>50</td>
<td>250</td>
<td>400</td>
<td>µA</td>
<td>VDD = 5V, VPIN = VSS</td>
</tr>
<tr>
<td>D080</td>
<td>VOL</td>
<td>Output Low Voltage</td>
<td>—</td>
<td>—</td>
<td>0.6</td>
<td>V</td>
<td>IOL = 8.5 mA, VDD = 4.5V, -40°C to +85°C</td>
</tr>
<tr>
<td></td>
<td></td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>0.6</td>
<td>V</td>
<td>IOL = 7.0 mA, VDD = 4.5V, -40°C to +125°C</td>
</tr>
<tr>
<td>D083</td>
<td></td>
<td>OSC2/CLKOUT (RC Osc mode)</td>
<td>—</td>
<td>—</td>
<td>0.6</td>
<td>V</td>
<td>IOL = 1.6 mA, VDD = 4.5V, -40°C to +85°C</td>
</tr>
<tr>
<td></td>
<td></td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>0.6</td>
<td>V</td>
<td>IOL = 1.2 mA, VDD = 4.5V, -40°C to +125°C</td>
</tr>
<tr>
<td>D090</td>
<td>VOH</td>
<td>Output High Voltage</td>
<td>VDD-0.7</td>
<td>—</td>
<td>—</td>
<td>V</td>
<td>IOH = -3.0 mA, VDD = 4.5V, -40°C to +85°C</td>
</tr>
<tr>
<td></td>
<td></td>
<td>VDD-0.7</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>V</td>
<td>IOH = -2.5 mA, VDD = 4.5V, -40°C to +125°C</td>
</tr>
<tr>
<td>D092</td>
<td></td>
<td>OSC2/CLKOUT (RC Osc mode)</td>
<td>VDD-0.7</td>
<td>—</td>
<td>—</td>
<td>V</td>
<td>IOH = -1.3 mA, VDD = 4.5V, -40°C to +85°C</td>
</tr>
<tr>
<td></td>
<td></td>
<td>VDD-0.7</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>V</td>
<td>IOH = -1.0 mA, VDD = 4.5V, -40°C to +125°C</td>
</tr>
<tr>
<td>D150</td>
<td>VOD</td>
<td>Open-Drain High Voltage</td>
<td>—</td>
<td>—</td>
<td>8.5</td>
<td>V</td>
<td>VDD = 5V, RA4 pin</td>
</tr>
</tbody>
</table>

* These parameters are characterized but not tested.
† Data in “Type” column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: In RC Oscillator mode, the OSC1/CLKIN pin is a Schmitt Trigger input. It is not recommended that the PICmicro be driven with external clock in RC mode.

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

(Note1)
12.5 AC (Timing) Characteristics

12.5.1 TIMING PARAMETER SYMBOLOGY

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

1. TppS2ppS
2. TppS

\[
\begin{array}{|c|c|c|c|c|c|}
\hline
\text{Sym} & \text{Characteristic} & \text{Min} & \text{Typ} & \text{Max} & \text{Units} \\
\hline
D100 & COSC2 & OSC2/CLKOUT pin & - & - & 15 & pF \\
D101 & CIo & All I/O pins and OSC2 (in RC mode) & - & - & 50 & pF \\
\hline
\end{array}
\]

- These parameters are characterized but not tested.
- † Data in “Type” column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note:
1. In RC Oscillator mode, the OSC1/CLKIN pin is a Schmitt Trigger input. It is not recommended that the PICmicro be driven with external clock in RC mode.
2. The leakage current on the MCLR/VPP pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.
3. Negative current is defined as current sourced by the pin.

Lowercase letters (pp) and their meanings:

\[
\begin{array}{|c|}
\hline
\text{pp} & \text{pp} \\
\hline
cc & CCP1 \\
ck & CLKOUT \\
cs & CS \\
di & SDI \\
do & SDO \\
dt & Data in \\
io & I/O port \\
mc & MCLR \\
\hline
\end{array}
\]

Uppercase letters and their meanings:

\[
\begin{array}{|c|}
\hline
\text{S} & \text{S} \\
\hline
F & \text{Fall} \\
H & \text{High} \\
I & \text{Invalid (Hi-impedance)} \\
L & \text{Low} \\
\hline
\end{array}
\]

\[
\begin{array}{|c|}
\hline
\text{S} & \text{S} \\
\hline
P & \text{Period} \\
R & \text{Rise} \\
V & \text{Valid} \\
Z & \text{Hi-impedance} \\
\hline
\end{array}
\]
12.5.2 TIMING CONDITIONS

The temperature and voltages specified in Table 12-1 apply to all timing specifications, unless otherwise noted. Figure 12-3 specifies the load conditions for the timing specifications.

TABLE 12-1: TEMPERATURE AND VOLTAGE SPECIFICATIONS - AC

<table>
<thead>
<tr>
<th>AC CHARACTERISTICS</th>
<th>Standard Operating Conditions (unless otherwise stated)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating temperature</td>
<td>-40°C ≤ TA ≤ +85°C for industrial</td>
</tr>
<tr>
<td>Operating voltage VDD range as described in DC spec Section 12.1 “DC Characteristics: PIC16F716 (Industrial, Extended)” and Section 12.4 “DC Characteristics: PIC16F716 (Industrial, Extended)”. LC parts operate for commercial/industrial temp's only.</td>
<td></td>
</tr>
</tbody>
</table>

FIGURE 12-3: LOAD CONDITIONS FOR DEVICE TIMING SPECIFICATIONS

Load condition 1

Load condition 2

RL = 464Ω

CL = 50 pF for all pins except OSC2/CLKOUT

15 pF for OSC2 output

12.5.3 TIMING DIAGRAMS AND SPECIFICATIONS

FIGURE 12-4: EXTERNAL CLOCK TIMING

Q1 Q2 Q3 Q4

OSC1

CLKOUT
TABLE 12-2: EXTERNAL CLOCK 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>1A</td>
<td>Fosc</td>
<td>Ext. Clock Input Frequency(1)</td>
<td>DC</td>
<td>—</td>
<td>4</td>
<td>MHz</td>
<td>RC and XT Osc modes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DC</td>
<td>—</td>
<td>20</td>
<td>MHz</td>
<td>HS Osc mode</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DC</td>
<td>—</td>
<td>200</td>
<td>kHz</td>
<td>LP Osc mode</td>
</tr>
<tr>
<td></td>
<td>Oscillator Frequency(1)</td>
<td>DC</td>
<td>—</td>
<td>4</td>
<td>MHz</td>
<td>RC Osc mode</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.1</td>
<td>—</td>
<td>4</td>
<td>MHz</td>
<td>XT Osc mode</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4</td>
<td>—</td>
<td>20</td>
<td>MHz</td>
<td>HS Osc mode</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5</td>
<td>—</td>
<td>200</td>
<td>kHz</td>
<td>LP Osc mode</td>
</tr>
<tr>
<td>1</td>
<td>Tosc</td>
<td>External CLKIN Period(1)</td>
<td>250</td>
<td>—</td>
<td>—</td>
<td>ns</td>
<td>RC and XT Osc modes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>50</td>
<td>—</td>
<td>—</td>
<td>ns</td>
<td>HS Osc mode</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5</td>
<td>—</td>
<td>—</td>
<td>μs</td>
<td>LP Osc mode</td>
</tr>
<tr>
<td></td>
<td>Oscillator Period(1)</td>
<td>250</td>
<td>—</td>
<td>—</td>
<td>ns</td>
<td>RC Osc mode</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>250</td>
<td>—</td>
<td>10,000</td>
<td>ns</td>
<td>XT Osc mode</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>50</td>
<td>—</td>
<td>250</td>
<td>ns</td>
<td>HS Osc mode</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5</td>
<td>—</td>
<td>—</td>
<td>μs</td>
<td>LP Osc mode</td>
</tr>
<tr>
<td>2</td>
<td>Tcy</td>
<td>Instruction Cycle Time(1)</td>
<td>200</td>
<td>—</td>
<td>DC</td>
<td>ns</td>
<td>Tcy = 4/Fosc</td>
</tr>
<tr>
<td>3*</td>
<td>TosL, TosH</td>
<td>External Clock in (OSC1) High or Low Time</td>
<td>100</td>
<td>—</td>
<td>—</td>
<td>ns</td>
<td>XT oscillator</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2.5</td>
<td>—</td>
<td>—</td>
<td>μs</td>
<td>LP oscillator</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>15</td>
<td>—</td>
<td>—</td>
<td>ns</td>
<td>HS oscillator</td>
</tr>
<tr>
<td>4*</td>
<td>TosR, TosF</td>
<td>External Clock in (OSC1) Rise or Fall Time</td>
<td>—</td>
<td>—</td>
<td>25</td>
<td>ns</td>
<td>XT oscillator</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>—</td>
<td>—</td>
<td>50</td>
<td>ns</td>
<td>LP oscillator</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>—</td>
<td>—</td>
<td>15</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 the OSC1/CLKIN pin. When an external clock input is used, the “Max.” cycle time limit is "DC" (no clock) for all devices.
## FIGURE 12-5: CLKOUT AND I/O TIMING

![CLKOUT and I/O Timing Diagram]

### TABLE 12-3: CLKOUT AND I/O 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>10†</td>
<td>TOSH2ckL</td>
<td>OSC1↑ to CLKOUT↓</td>
<td>—</td>
<td>75</td>
<td>200</td>
<td>ns</td>
<td>(Note 1)</td>
</tr>
<tr>
<td>11†</td>
<td>TOSH2ckH</td>
<td>OSC1↑ to CLKOUT↑</td>
<td>—</td>
<td>75</td>
<td>200</td>
<td>ns</td>
<td>(Note 1)</td>
</tr>
<tr>
<td>12</td>
<td>TckR</td>
<td>CLKOUT rise time</td>
<td>—</td>
<td>35</td>
<td>100</td>
<td>ns</td>
<td>(Note 1)</td>
</tr>
<tr>
<td>13</td>
<td>TckF</td>
<td>CLKOUT fall time</td>
<td>—</td>
<td>35</td>
<td>100</td>
<td>ns</td>
<td>(Note 1)</td>
</tr>
<tr>
<td>14</td>
<td>TckL2ioV</td>
<td>CLKOUT ↓ to Port out valid</td>
<td>—</td>
<td>—</td>
<td>20</td>
<td>ns</td>
<td>(Note 1)</td>
</tr>
<tr>
<td>15</td>
<td>TioV2ckH</td>
<td>Port input valid before CLKOUT↑</td>
<td>Tosc + 200</td>
<td>—</td>
<td>—</td>
<td>ns</td>
<td>(Note 1)</td>
</tr>
<tr>
<td>16</td>
<td>TckH2ioL</td>
<td>Port input hold after CLKOUT↑</td>
<td>0</td>
<td>—</td>
<td>—</td>
<td>ns</td>
<td>(Note 1)</td>
</tr>
<tr>
<td>17</td>
<td>TosH2ioV</td>
<td>OSC1↑ (Q1 cycle) to Port out valid</td>
<td>—</td>
<td>50</td>
<td>150</td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td>18†</td>
<td>TosH2ioL</td>
<td>OSC1↑ (Q2 cycle) to Port output invalid</td>
<td>Standard</td>
<td>100</td>
<td>—</td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td>18A†</td>
<td>TosH2ioL</td>
<td>Port input invalid (I/O in hold time)</td>
<td>Extended (LC)</td>
<td>200</td>
<td>—</td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>TioV2osH</td>
<td>Port input valid to OSC1↑ (I/O in setup time)</td>
<td>0</td>
<td>—</td>
<td>—</td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td>20†</td>
<td>TioR</td>
<td>Port output rise time</td>
<td>Standard</td>
<td>—</td>
<td>10</td>
<td>40</td>
<td>ns</td>
</tr>
<tr>
<td>20A†</td>
<td>TioR</td>
<td>Port output rise time</td>
<td>Extended (LC)</td>
<td>—</td>
<td>80</td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td>21†</td>
<td>TioF</td>
<td>Port output fall time</td>
<td>Standard</td>
<td>—</td>
<td>10</td>
<td>40</td>
<td>ns</td>
</tr>
<tr>
<td>21A†</td>
<td>TioF</td>
<td>Port output fall time</td>
<td>Extended (LC)</td>
<td>—</td>
<td>80</td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td>22††</td>
<td>TINP</td>
<td>INT pin high or low time</td>
<td>Tcy</td>
<td>—</td>
<td>—</td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td>23††</td>
<td>TRBP</td>
<td>RB7:RB4 change INT high or low 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 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.
†† These parameters are asynchronous events not related to any internal clock edge.

**Note 1:** Measurements are taken in RC mode where CLKOUT output is 4 x Tosc.
FIGURE 12-6: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING(1)

VDD
MCLR
Internal POR
PWRT
Time-out
OSC
Time-out
Internal
Reset
Watchdog
Timer
Reset
I/O Pins

Note 1: Refer to Figure 12-3 for load conditions.

FIGURE 12-7: BROWN-OUT RESET TIMING

VDD
BVDD

35

TABLE 12-4: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER, POWER-UP TIMER, AND BROWN-OUT RESET 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>30</td>
<td>TMC</td>
<td>MCLR Pulse Width (low)</td>
<td>2</td>
<td>—</td>
<td>—</td>
<td>μs</td>
<td>VDD = 5V, -40°C to +125°C</td>
</tr>
<tr>
<td>31*</td>
<td>TWDT</td>
<td>Watchdog Timer Time-out Period (No Prescaler)</td>
<td>7</td>
<td>18</td>
<td>33</td>
<td>ms</td>
<td>VDD = 5V, -40°C to +85°C</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TDB</td>
<td>TBD</td>
<td>TBD</td>
<td>ms</td>
<td>VDD = 5V, +85°C to +125°C</td>
</tr>
<tr>
<td>32</td>
<td>TOST</td>
<td>Oscillation Start-up Timer Period</td>
<td>—</td>
<td>1024</td>
<td>Tosc</td>
<td>ms</td>
<td>Tosc = OSC1 period</td>
</tr>
<tr>
<td>33*</td>
<td>TPWRT</td>
<td>Power-up Timer Period</td>
<td>28</td>
<td>72</td>
<td>132</td>
<td>ms</td>
<td>VDD = 5V, -40°C to +85°C</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TDB</td>
<td>TBD</td>
<td>TBD</td>
<td>ms</td>
<td>VDD = 5V, +85°C to +125°C</td>
</tr>
<tr>
<td>34</td>
<td>TI0Z</td>
<td>I/O high-impedance from MCLR Low or WDT Reset</td>
<td>—</td>
<td>—</td>
<td>2.1</td>
<td>μs</td>
<td>VDD ≤ BVDD (D005)</td>
</tr>
<tr>
<td>35</td>
<td>TBOR</td>
<td>Brown-out Reset Pulse Width</td>
<td>100</td>
<td>—</td>
<td>—</td>
<td>μs</td>
<td>VDD ≤ BVDD (D005)</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.
### FIGURE 12-8: TIMER0 AND TIMER1 EXTERNAL CLOCK TIMINGS\(^{(1)}\)

![Diagram of Timer0 and Timer1 external clock timings]

### TABLE 12-5: 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 0.5(T_{CY}) + 20 ns</td>
<td>—</td>
<td>—</td>
<td>ns</td>
<td>Must also meet parameter 42</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>With Prescaler 10 ns</td>
<td>—</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 0.5(T_{CY}) + 20 ns</td>
<td>—</td>
<td>—</td>
<td>ns</td>
<td>Must also meet parameter 42</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>With Prescaler 10 ns</td>
<td>—</td>
<td>—</td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>T0P</td>
<td>T0CKI Period</td>
<td>No Prescaler (T_{CY} + 40) ns</td>
<td>—</td>
<td>—</td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>With Prescaler Greater of: (20) or (T_{CY} + 40) ns (N) = prescale value (2, 4, ..., 256)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>45</td>
<td>T1H</td>
<td>T1CKI High Time</td>
<td>Synchronous, Prescaler = 1 (0.5T_{CY} + 20) ns</td>
<td>—</td>
<td>—</td>
<td>ns</td>
<td>Must also meet parameter 47</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Standard 15 ns</td>
<td>—</td>
<td>—</td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asynchronous 30 ns</td>
<td>—</td>
<td>—</td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td>46</td>
<td>T1L</td>
<td>T1CKI Low Time</td>
<td>Synchronous, Prescaler = 1 (0.5T_{CY} + 20) ns</td>
<td>—</td>
<td>—</td>
<td>ns</td>
<td>Must also meet parameter 47</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Standard 15 ns</td>
<td>—</td>
<td>—</td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asynchronous 30 ns</td>
<td>—</td>
<td>—</td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td>47</td>
<td>T1P</td>
<td>T1CKI input period</td>
<td>Synchronous Standard</td>
<td>—</td>
<td>—</td>
<td>ns</td>
<td>(N = \text{prescale value (1, 2, 4, 8)})</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asynchronous 60 ns</td>
<td>—</td>
<td>—</td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td>F1</td>
<td></td>
<td>Timer1 oscillator input frequency range (oscillator enabled by setting bit T1OSCEN)</td>
<td>32.768 kHz</td>
<td>—</td>
<td>—</td>
<td>kHz</td>
<td></td>
</tr>
<tr>
<td>48</td>
<td>TCCKE</td>
<td>Delay from external clock edge to timer increment</td>
<td>2(T_{osc}) ns</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 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: Refer to Figure 12-3 for load conditions.
FIGURE 12-9: CAPTURE/COMPARE/PWM TIMINGS(1)

TABLE 12-6: CAPTURE/COMPARE/PWM 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>50*</td>
<td>TccL</td>
<td>CCP1 input low time</td>
<td>0.5Tcy + 20</td>
<td>—</td>
<td>—</td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>No Prescaler</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>With Prescaler</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>51*</td>
<td>TccH</td>
<td>CCP1 input high time</td>
<td>0.5Tcy + 20</td>
<td>—</td>
<td>—</td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>No Prescaler</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>With Prescaler</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>52*</td>
<td>TccP</td>
<td>CCP1 input period</td>
<td>3Tcy + 40/N</td>
<td>—</td>
<td>—</td>
<td>ns</td>
<td>N = prescale value (1, 4, or 16)</td>
</tr>
<tr>
<td>53*</td>
<td>TccR</td>
<td>CCP1 output rise time</td>
<td>—</td>
<td>10</td>
<td>40</td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td>53A*</td>
<td></td>
<td>Extended</td>
<td>—</td>
<td>—</td>
<td>80</td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td>54*</td>
<td>TccF</td>
<td>CCP1 output fall time</td>
<td>—</td>
<td>10</td>
<td>40</td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td>54A*</td>
<td></td>
<td>Extended</td>
<td>—</td>
<td>—</td>
<td>80</td>
<td>ns</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: Refer to Figure 12-3 for load conditions.
### Table 12-7: A/D Converter Characteristics: PIC16F716 (Industrial, Extended)

<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>A00</td>
<td>VDD</td>
<td>VDD Operation</td>
<td>2.5</td>
<td>—</td>
<td>5.5</td>
<td>V</td>
<td>VDD — 5.5 V</td>
</tr>
<tr>
<td>A01</td>
<td>NR</td>
<td>Resolution</td>
<td>—</td>
<td>—</td>
<td>8-bits</td>
<td>bit</td>
<td>VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF</td>
</tr>
<tr>
<td>A02</td>
<td>EABS</td>
<td>Total Absolute error</td>
<td>—</td>
<td>—</td>
<td>&lt; ± 1</td>
<td>LSb</td>
<td>VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF</td>
</tr>
<tr>
<td>A03</td>
<td>EIL</td>
<td>Integral linearity error</td>
<td>—</td>
<td>—</td>
<td>&lt; ± 1</td>
<td>LSb</td>
<td>VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF</td>
</tr>
<tr>
<td>A04</td>
<td>EDL</td>
<td>Differential linearity error</td>
<td>—</td>
<td>—</td>
<td>&lt; ± 1</td>
<td>LSb</td>
<td>VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF</td>
</tr>
<tr>
<td>A05</td>
<td>EFS</td>
<td>Full scale error</td>
<td>—</td>
<td>—</td>
<td>&lt; ± 1</td>
<td>LSb</td>
<td>VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF</td>
</tr>
<tr>
<td>A06</td>
<td>EOFF</td>
<td>Offset error</td>
<td>—</td>
<td>—</td>
<td>&lt; ± 1</td>
<td>LSb</td>
<td>VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF</td>
</tr>
<tr>
<td>A10</td>
<td>—</td>
<td>Monotonicity</td>
<td>—</td>
<td>guaranteed(3)</td>
<td>—</td>
<td>—</td>
<td>VSS ≤ VAIN ≤ VREF</td>
</tr>
<tr>
<td>A20</td>
<td>VREF</td>
<td>Reference voltage</td>
<td>2.5V</td>
<td>—</td>
<td>VDD + 0.3</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td>A25</td>
<td>VAIN</td>
<td>Analog input voltage</td>
<td>VSS - 0.3</td>
<td>—</td>
<td>VREF + 0.3</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td>A30</td>
<td>ZAIN</td>
<td>Recommended impedance of analog voltage source</td>
<td>—</td>
<td>—</td>
<td>10.0</td>
<td>kΩ</td>
<td></td>
</tr>
<tr>
<td>A40</td>
<td>IAD</td>
<td>A/D conversion current (VDD)</td>
<td>—</td>
<td>180</td>
<td>—</td>
<td>µA</td>
<td>Average current consumption when A/D is on. (1)</td>
</tr>
<tr>
<td>A50</td>
<td>IREF</td>
<td>VREF input current(2)</td>
<td>10</td>
<td>—</td>
<td>1000</td>
<td>µA</td>
<td>During VAIN acquisition. Based on differential of VHOLD to VAIN to charge CHOLD, see Section 12.1 “DC Characteristics: PIC16F716 (Industrial, Extended)”. During A/D Conversion cycle</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:** When A/D is off, it will not consume any current other than minor leakage current. The power-down current spec includes any such leakage from the A/D module.

**Note 2:** VREF current is from RA3 pin or VDD pin, whichever is selected as reference input.

**Note 3:** The A/D conversion result never decreases with an increase in the input voltage, and has no missing codes.
FIGURE 12-10: A/D CONVERSION TIMING

![A/D Conversion Timing Diagram]

Note 1: If the A/D clock source is selected as RC, a time of Tcy is added before the A/D clock starts. This allows the SLEEP instruction to be executed.

TABLE 12-8: 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>130</td>
<td>TAD</td>
<td>A/D clock period</td>
<td></td>
<td></td>
<td></td>
<td>µs</td>
<td>TOSC based, VREF ≥ 3.0V</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Industrial</td>
<td>1.6</td>
<td>—</td>
<td>—</td>
<td>µs</td>
<td>A/D RC mode</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Extended</td>
<td>1.6</td>
<td>4.0</td>
<td>6.0</td>
<td>µs</td>
<td>A/D RC mode</td>
</tr>
<tr>
<td>131</td>
<td>TCNV</td>
<td>Conversion time (not including S/H time)†</td>
<td>9.5</td>
<td>—</td>
<td>9.5</td>
<td>TAD</td>
<td></td>
</tr>
<tr>
<td>132</td>
<td>TACQ</td>
<td>Acquisition time</td>
<td>(Note 2)</td>
<td>20</td>
<td>—</td>
<td>µs</td>
<td>The minimum time is the amplifier settling time. This may be used if the &quot;new&quot; input voltage has not changed by more than 1 Lsb (i.e., 20.0 mV @ 5.12V) from the last sampled voltage (as stated on CHOLD).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Industrial</td>
<td></td>
<td></td>
<td></td>
<td>µs</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Extended</td>
<td></td>
<td></td>
<td></td>
<td>µs</td>
<td></td>
</tr>
<tr>
<td>134</td>
<td>TGO</td>
<td>Q4 to A/D clock start</td>
<td></td>
<td></td>
<td></td>
<td>µs</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>
<tr>
<td>135</td>
<td>TSWC</td>
<td>Switching from convert → sample time</td>
<td>1.5</td>
<td>**</td>
<td>—</td>
<td>TAD</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: ADRES register may be read on the following Tcy cycle.
2: See Section 12.1 “DC Characteristics: PIC16F716 (Industrial, Extended)” for min. conditions.
13.0 DC AND AC
CHARACTERISTICS GRAPHS
AND TABLES

The graphs and tables provided in this section are for
design guidance and are not tested.

In some graphs or tables, the data presented are
outside specified operating range (i.e., outside
specified VDD range). This is for information only and
devices will operate properly only within the specified
range.

The data presented in this section is a statistical
summary of data collected on units from different lots
over a period of time and matrix samples. 'Typical' repre-
sents the mean of the distribution at 25°C. 'Max' or
'min' represents (mean + 3σ) or (mean - 3σ)
respectively, where σ is standard deviation, over the
whole temperature range.

Graphs and Tables not available at this
time.
14.0 PACKAGING INFORMATION

14.1 Package Marking Information

18-Lead PDIP

- Example
  - PIC16F716-04/P
  - 0023CBA

18-Lead SOIC

- Example
  - PIC16F716
  - -20/SO
  - 0018CDK

20-Lead SSOP

- Example
  - PIC16F716
  - -20I/SS025
  - 0020CBK

Legend:

<table>
<thead>
<tr>
<th>XX...X</th>
<th>Customer specific information*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>Year code (last digit of calendar year)</td>
</tr>
<tr>
<td>YY</td>
<td>Year code (last 2 digits of calendar year)</td>
</tr>
<tr>
<td>WW</td>
<td>Week code (week of January 1 is week '01')</td>
</tr>
<tr>
<td>NNN</td>
<td>Alphanumeric traceability code</td>
</tr>
</tbody>
</table>

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.
PIC16F716

18-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>18</td>
</tr>
<tr>
<td>Pitch</td>
<td>P</td>
<td>.100</td>
</tr>
<tr>
<td>Top to Seating Plane</td>
<td>A</td>
<td>.140</td>
</tr>
<tr>
<td>Molded Package Thickness</td>
<td>A2</td>
<td>.115</td>
</tr>
<tr>
<td>Base to Seating Plane</td>
<td>A1</td>
<td>.015</td>
</tr>
<tr>
<td>Shoulder to Shoulder Width</td>
<td>E</td>
<td>.300</td>
</tr>
<tr>
<td>Molded Package Width</td>
<td>E1</td>
<td>.240</td>
</tr>
<tr>
<td>Overall Length</td>
<td>D</td>
<td>.890</td>
</tr>
<tr>
<td>Tip to Seating Plane</td>
<td>L</td>
<td>.125</td>
</tr>
<tr>
<td>Lead Thickness</td>
<td>c</td>
<td>.008</td>
</tr>
<tr>
<td>Upper Lead Width</td>
<td>B1</td>
<td>.045</td>
</tr>
<tr>
<td>Lower Lead Width</td>
<td>B</td>
<td>.014</td>
</tr>
<tr>
<td>Overall Row Spacing</td>
<td>eB</td>
<td>.310</td>
</tr>
<tr>
<td>Mold Draft Angle Top</td>
<td>α</td>
<td>5</td>
</tr>
<tr>
<td>Mold Draft Angle Bottom</td>
<td>β</td>
<td>5</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. C04-007
### PIC16F716

18-Lead Plastic Small Outline (SO) – Wide, 300 mil (SOIC)

#### 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-013
- Drawing No. C04-051

<table>
<thead>
<tr>
<th>Dimension Limits</th>
<th>MILIMETERS</th>
<th>INCHES*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Pins</td>
<td>n</td>
<td>18</td>
</tr>
<tr>
<td>Pitch</td>
<td>P</td>
<td>.050</td>
</tr>
<tr>
<td>Overall Height</td>
<td>A</td>
<td>.093</td>
</tr>
<tr>
<td>Molded Package Thickness</td>
<td>A2</td>
<td>.088</td>
</tr>
<tr>
<td>Standoff §</td>
<td>A1</td>
<td>.004</td>
</tr>
<tr>
<td>Overall Width</td>
<td>E</td>
<td>.394</td>
</tr>
<tr>
<td>Molded Package Width</td>
<td>E1</td>
<td>.291</td>
</tr>
<tr>
<td>Overall Length</td>
<td>D</td>
<td>.446</td>
</tr>
<tr>
<td>Chamfer Distance</td>
<td>h</td>
<td>.010</td>
</tr>
<tr>
<td>Foot Length</td>
<td>L</td>
<td>.016</td>
</tr>
<tr>
<td>Foot Angle</td>
<td>φ</td>
<td>0</td>
</tr>
<tr>
<td>Lead Thickness</td>
<td>c</td>
<td>.009</td>
</tr>
<tr>
<td>Lead Width</td>
<td>B</td>
<td>.014</td>
</tr>
<tr>
<td>Mold Draft Angle Top</td>
<td>α</td>
<td>0</td>
</tr>
<tr>
<td>Mold Draft Angle Bottom</td>
<td>β</td>
<td>0</td>
</tr>
</tbody>
</table>

*Controlling Parameter
§ Significant Characteristic

---

© 2003 Microchip Technology Inc.

Preliminary
20-Lead Plastic Shrink Small Outline (SS) – 209 mil, 5.30 mm (SSOP)

<table>
<thead>
<tr>
<th>Dimension Limits</th>
<th>INCHES*</th>
<th>MILLIMETERS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Pins</td>
<td>n</td>
<td>20</td>
</tr>
<tr>
<td>Pitch</td>
<td>p</td>
<td>.026</td>
</tr>
<tr>
<td>Overall Height</td>
<td>A</td>
<td>.068 .073 .078</td>
</tr>
<tr>
<td>Molded Package Thickness</td>
<td>A2</td>
<td>.064 .068 .072</td>
</tr>
<tr>
<td>Standoff §</td>
<td>A1</td>
<td>.002 .006 .010</td>
</tr>
<tr>
<td>Overall Width</td>
<td>E</td>
<td>.299 .309 .322</td>
</tr>
<tr>
<td>Molded Package Width</td>
<td>E1</td>
<td>.201 .207 .212</td>
</tr>
<tr>
<td>Overall Length</td>
<td>D</td>
<td>.278 .284 .289</td>
</tr>
<tr>
<td>Foot Length</td>
<td>L</td>
<td>.022 .030 .037</td>
</tr>
<tr>
<td>Lead Thickness</td>
<td>c</td>
<td>.004 .007 .010</td>
</tr>
<tr>
<td>Foot Angle</td>
<td>φ</td>
<td>0 4 8</td>
</tr>
<tr>
<td>Lead Length</td>
<td>B</td>
<td>.010 .013 .015</td>
</tr>
<tr>
<td>Mold Draft Angle Top</td>
<td>α</td>
<td>0 5 10</td>
</tr>
<tr>
<td>Mold Draft Angle Bottom</td>
<td>β</td>
<td>0 5 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: MO-150
Drawing No. C04-072
APPENDIX A: REVISION HISTORY

Revision A (June 2003)

Original data sheet. However, the device described in this data sheet are upgrades to PIC16C716.

APPENDIX B: CONVERSION CONSIDERATIONS

This is a Flash program memory version of the PIC16C716 device. Refer to the migration document, DS40059, for more information about differences between the PIC16F716 and PIC16C716.
APPENDIX C: MIGRATION FROM BASE-LINE TO MID-RANGE DEVICES

This section discusses how to migrate from a baseline device (i.e., PIC16C5X) to a mid-range device (i.e., PIC16F716).

The following are the list of modifications over the PIC16C5X microcontroller family:

1. Instruction word length is increased to 14-bits. This allows larger page sizes both in program memory (2K now as opposed to 512 before) and register file (128 bytes now versus 32 bytes before).
2. A PC high latch register (PCLATH) is added to handle program memory paging. Bits PA2, PA1, PA0 are removed from Status register.
3. Data memory paging is redefined slightly. Status register is modified.
4. Four new instructions have been added: RETURN, RETFIE, ADDLW, and SUBLW. Two instructions TRIS and OPTION are being phased out although they are kept for compatibility with PIC16C5X.
5. OPTION_REG and TRIS registers are made addressable.
6. Interrupt capability is added. Interrupt vector is at 0004h.
7. Stack size is increased to 8 deep.
8. Reset vector is changed to 0000h.
9. Reset of all registers is revisited. Five different Reset (and wake-up) types are recognized. Registers are reset differently.
10. Wake-up from Sleep through interrupt is added.
11. Two separate timers, Oscillator Start-up Timer (OST) and Power-up Timer (PWRT) are included for more reliable power-up. These timers are invoked selectively to avoid unnecessary delays on power-up and wake-up.
12. PORTB has weak pull-ups and interrupt-on-change feature.
13. T0CKI pin is also a port pin (RA4) now.
14. FSR is made a full eight bit register.
15. “In-circuit serial programming” is made possible. The user can program PIC16F716 devices using only five pins: Vcc, Vss, MCLR/VPP, RB6 (clock) and RB7 (data in/out).
16. PCON status register is added with a Power-on Reset Status bit (POR).
17. Brown-out protection circuitry has been added. Controlled by configuration word bits BOREN and BORV. Brown-out Reset ensures the device is placed in a Reset condition if Vcc dips below a fixed setpoint.

To convert code written for PIC16C5X to PIC16F716, the user should take the following steps:

1. Remove any program memory page select operations (PA2, PA1, PA0 bits) for CALL, GOTO.
2. Revisit any computed jump operations (write to PC or add to PC, etc.) to make sure page bits are set properly under the new scheme.
3. Eliminate any data memory page switching. Redefine data variables to reallocate them.
4. Verify all writes to Status, Option, and FSR registers since these have changed.
5. Change Reset vector to 0000h.

Note 1: 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.

2: The user should verify that the device oscillator starts and performs as expected. Adjusting the loading capacitor values and/or the Oscillator mode may be required.
ON-LINE SUPPORT

Microchip provides on-line support on the Microchip World Wide Web site.

The web site is used by Microchip as a means to make files and information easily available to customers. To view the site, the user must have access to the Internet and a web browser, such as Netscape® or Microsoft® Internet Explorer. Files are also available for FTP download from our FTP site.

Connecting to the Microchip Internet Web Site

The Microchip web site is available at the following URL:

www.microchip.com

The file transfer site is available by using an FTP service to connect to:

ftp://ftp.microchip.com

The web site and file transfer site provide a variety of services. Users may download files for the latest Development Tools, Data Sheets, Application Notes, User's Guides, Articles and Sample Programs. A variety of Microchip specific business information is also available, including listings of Microchip sales offices, distributors and factory representatives. Other data available for consideration is:

• Latest Microchip Press Releases
• Technical Support Section with Frequently Asked Questions
• Design Tips
• Device Errata
• Job Postings
• Microchip Consultant Program Member Listing
• Links to other useful web sites related to Microchip Products
• Conferences for products, Development Systems, technical information and more
• Listing of seminars and events

SYSTEMS INFORMATION AND UPGRADE HOT LINE

The Systems Information and Upgrade Line provides system users a listing of the latest versions of all of Microchip’s development systems software products. Plus, this line provides information on how customers can receive the most current upgrade kits. The Hot Line Numbers are:

1-800-755-2345 for U.S. and most of Canada, and
1-480-792-7302 for the rest of the world.

042003
READER RESPONSE

It is our intention to provide you with the best documentation possible to ensure successful use of your Microchip product. If you wish to provide your comments on organization, clarity, subject matter, and ways in which our documentation can better serve you, please FAX your comments to the Technical Publications Manager at (480) 792-4150.

Please list the following information, and use this outline to provide us with your comments about this document.

To: Technical Publications Manager
   Total Pages Sent ________

RE: Reader Response

From: Name ________________________________________________
      Company _______________________________________________
      Address ________________________________________________
      City / State / ZIP / Country _________________________________
      Telephone: (______) _________ - _________  FAX: (______) _________ - _________

Application (optional):

Would you like a reply? Y N

Device: PIC16F716  Literature Number: DS41206A

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?
INDEX

A
A/D ................................................................. 49
A/D Converter Enable (ADIE Bit) .......................... 14
A/D Converter Flag (ADIF Bit) .............................. 15, 50
A/D Converter Interrupt, Configuring .................... 50
ADCON0 Register .............................................. 9, 49
ADCON1 Register .............................................. 10, 50
ADRES Register ................................................ 9
Analog Port Pins, Configuring ............................... 52
Channel Select (CHS2:CHS0 Bits) ......................... 49
Clock Select (ADCS1:ADCS0 Bits) ......................... 49
Configuring the Module ..................................... 50
Conversion Clock (Tad) ...................................... 52
Conversion Status (GO/DONE Bit) ......................... 49, 50
Conversions ..................................................... 53
Converter Characteristics .................................. 104
Module On/Off (ADON Bit) .................................. 49
Port Configuration Control (PCFG2:PCFG0 Bits) ...... 50
Sampling Requirements ...................................... 51
Special Event Trigger (CCP) ............................... 35, 53
Timing Diagram ............................................... 105
Absolute Maximum Ratings ............................... 91
ADCON0 Register .............................................. 9
ADCON1 Register .............................................. 10
ADDLW Instruction ............................................ 73
ADDWF Instruction ............................................ 73
ADRES Register ................................................ 9
Analog Input Model .......................................... 51
Analog-to-Digital Converter. See A/D ....................
andi Instruction ................................................ 77
ASL Instruction .................................................. 77
ASR Instruction .................................................. 77
B
Banking, Data Memory ....................................... 7, 11
BCF Instruction .................................................. 74
Block Diagrams
A/D .................................................................... 51
Capture (CCP Module) ....................................... 34
Compare (CCP Module) ....................................... 35
Interrupt Sources ............................................. 65
On-Chip Reset Circuit ....................................... 60
PIC16F716 ....................................................... 5
PORTA ............................................................ 19, 20
PORTB ............................................................ 21
PWM (CCP Module) .......................................... 36
PWM (Enhanced) .............................................. 38
RB1/T1OSO/T1CIF ............................................ 22
RB2/T1OSI ........................................................ 22
RB3/CCP1/P1A .................................................. 23
RB4 ................................................................. 23
RB5 ................................................................. 24
RB6/P1C .......................................................... 24
RB7/P1D .......................................................... 25
Timer0 ............................................................. 27
Timer0/WDHW Prescaler .................................... 28
Timer1 ............................................................. 30
Timer2 ............................................................. 32
Watchdog Timer (WDT) ..................................... 67
BOR, See Brown-out Reset ................................. 59
Brown-out Detect (BOD) .................................... 59
Brown-out Reset (BOR) .................................... 55, 58, 62, 63
BOR Enable (BODEN Bit) ................................. 56
BOR Status (BOR Bit) ....................................... 16
Timing Diagram ............................................... 101
BSF Instruction .................................................. 74
BTFSC Instruction ............................................. 74
BTFSS Instruction ............................................. 75
C
C Compilers
MPLAB C17 ..................................................... 86
MPLAB C18 ..................................................... 86
MPLAB C30 ..................................................... 86
CALL Instruction .............................................. 75
Capture (CCP Module) ....................................... 34
CCP Pin Configuration ....................................... 34
CCPR1H:CCPR1L Registers ............................... 34
Software Interrupt ............................................ 34
Timer1 Mode Selection ..................................... 34
Capture/Compare/PWM (CCP) ............................ 33
Capture Mode. See Capture
CCP1CON Register .......................................... 9
CCPR1H Register ............................................ 9, 33
CCPR1L Register ............................................ 9, 33
Compare Mode. See Compare
Enable (CCP1IE Bit) ........................................ 14
Flag (CCP1IF Bit) ............................................. 15
Software Interrupt ........................................... 35
PWM Mode. See PWM
Timer Resources .............................................. 33
Timing Diagram ............................................... 103
CLRF Instruction ............................................... 75
CLRW Instruction ............................................. 76
CLRWD instruction .......................................... 76
Code Examples
Capture (CCP Module)
Changing Between Capture Prescalers .................. 34
How to Clear RAM Using Indirect Addressing ........ 18
Initializing PORTA .......................................... 19
Initializing PORTB .......................................... 21
Code Protection ............................................... 55, 69
CP1:CP0 Bits .................................................. 56
COMF Instruction ............................................ 76
Compare (CCP Module) ..................................... 34
CCP Pin Configuration ..................................... 35
Software Interrupt ........................................... 35
Special Event Trigger ....................................... 30, 35, 53
Timer1 Mode Selection ..................................... 35
Configuration Bits ........................................... 55
Conversion Considerations ................................. 113
D
Data Memory ................................................... 7
Bank Select (RP1:RP0 Bits) ................................. 7, 11
General Purpose Registers ............................... 8
Register File Map ............................................. 8
Special Function Registers ............................... 9
DC Characteristics ........................................... 93, 94, 95, 96
DECF Instruction ............................................. 76
DECFZ Instruction ........................................... 77
Demonstration Boards
PICDEM 1 ..................................................... 88
PICDEM 17 .................................................... 88
PICDEM 18R PIC18C601/801 ....................... 89
PICDEM 2 Plus .............................................. 88
PICDEM 3 PIC16C92X ...................................... 88

© 2003 Microchip Technology Inc.
### PIC16F716

#### Development Support
- PICDEM 4 .................................................. 88
- PICDEM LIN PIC16C43X .................................. 89
- PICDEM USB PIC16C75 ................................. 89
- PICDEM.net Internet/Ethernet .......................... 88

#### Direct Addressing
- Reset, Normal Operation ................................. 58, 62, 63
- Enhanced CCP Auto-Shutdown ......................... 45
- Start-up Considerations .................................. 47
- Electrical Characteristics ............................... 91
- Enhanced Capture/Compare/PWM (ECCP) ......... 45
- Enhanced PWM Mode. See PWM (ECCP Module) 45
- Errata ......................................................... 3
- Evaluation and Programming Tools .................. 89
- External Power-on Reset Circuit ...................... 58

#### Instruction Set Summary
- GOTO Instruction ........................................... 77
- MOVF Instruction .......................................... 80
- DELAY Instruction ......................................... 81
- CALL Instruction ........................................... 82
- BCF Instruction ............................................. 83
- BSF Instruction ............................................. 84
- CLRF Instruction ........................................... 85
- CLRWDT Instruction ....................................... 86
- COMF Instruction ........................................... 87
- DECF Instruction ........................................... 88
- DECEFSZ Instruction ...................................... 89
- GOTO Instruction ........................................... 90
- INCFSZ Instruction ......................................... 91
- INCF Instruction ........................................... 92
- INDIRECT Addressing ..................................... 93
- IORLW Instruction ......................................... 94
- IORWF Instruction ......................................... 95
- SUBWF Instruction ......................................... 96
- SWAPF Instruction .......................................... 97
- TRIS Instruction ............................................ 98
- XORLW Instruction ......................................... 99
- XORWF Instruction ......................................... 100

#### Interrupts, Enable Bits
- A/D Conversion Enable (ADIE Bit) .................... 14
- CCP1 Enable (CCP1IE Bit) .............................. 14, 34
- Global Interrupt Enable (GIE Bit) .................... 13, 65
- Interrupt-on-Change (RB7:RB4) Enable (RBIE Bit) 13, 66
- Peripheral Interrupt Enable (PEIE Bit) ............... 13
- RB0/INT Enable (INTE Bit) ............................... 13
- TMR0 Overflow Enable (TOIE Bit) ..................... 13
- TMR1 Overflow Enable (TMRIE Bit) .................... 14
- TMR2 to PR2 Match Enable (TMR2IE Bit) ............. 14

#### Interrupts, Flag Bits
- A/D Conversion Flag (ADIF Bit) ....................... 15, 50
- CCP1 Flag (CCP1IF Bit) .................................. 15, 34
- Interrupt-on-Change (RB7:RB4) Flag (RBIF Bit) .. 13
- RB0/INT Flag (INTF Bit) ................................. 13
- TMR0 Overflow Flag (TOIF Bit) ....................... 13
- TMR1 Overflow Flag (TMRIF Bit) ...................... 15
- TMR2 to PR2 Match Flag (TMR2IF Bit) ............... 15

#### Memory Organization
- Data Memory .............................................. 7
- Program Memory ......................................... 7

#### Migration from Base-Line to Mid-Range Devices
- MPLAB ASM30 Assembler, Linker, Librarian .......... 86
- MPLAB ICE 2 In-Circuit Debugger ...................... 87
- MPLAB ICE 2000 High Performance Universal ......... 87
- MPLAB ICE 4000 High Performance Universal ......... 87
POR Status (POR Bit) ............................................................. 16
Power Control (PCON) Register ........................................ 62
Power-down (PD Bit) ............................................................ 11, 58
Power-on Reset Circuit, External ........................................ 58
Power-up Timer (PWRT) .................................................... 55, 59
PWRT Enable (PWRTIE Bit) .................................................. 56
Time-out (T0 Bit) ................................................................ 11, 58
Time-out Sequence ........................................................... 62
Time-out Sequence on Power-up ...................................... 64
Timing Diagram ................................................................ 101
Prescaler, Capture ............................................................... 34
Prescaler, Timer0 ................................................................. 27
Assignment (PSA Bit) ......................................................... 12, 27
Rate Select (PS2:PS0 Bits) .................................................. 12, 27
Switching Between Timer0 and WDT ................................. 28
Prescaler, Timer1 ................................................................. 30
Select (T1CKPS1:T1CKPS0 Bits) ........................................ 29
Prescaler, Timer2 ................................................................. 37
Select (T2CKPS1:T2CKPS0 Bits) ......................................... 31
PRO MATE II Universal Device Programmer ...................... 87
Program Counter ................................................................. 87
PCL Register ........................................................................ 9, 17
PCLATH Register ............................................................... 9, 17, 66
Reset Conditions ............................................................... 62
Program Memory ................................................................. 7
Interrupt Vector ................................................................... 7
Paging .................................................................................. 7
Program Memory Map ......................................................... 7
Reset Vector ......................................................................... 7
Program Verification ........................................................... 69
PWM (CCP Module) ............................................................ 36
CCPR1H:CCPR1L Registers ............................................... 36
Duty Cycle ......................................................................... 37
Example Frequencies/Resolutions ...................................... 37
Output Diagram ................................................................... 36
Period .................................................................................. 36
Set-Up for PWM Operation ............................................... 37
TMR2 to PR2 Match ........................................................... 31, 36
TMR2 to PR2 Match Enable (TMR2IE Bit) ......................... 14
TMR2 to PR2 Match Flag (TMR2IF Bit) ............................... 15
PWM (ECCP Module) .......................................................... 38
Associated Registers .......................................................... 48
Direction Change in Full-Bridge Output Mode...................... 43
Effects of Reset ................................................................. 48
Full-Bridge Application Example ...................................... 43
Full-Bridge Mode ............................................................... 43
Half-Bridge Mode .............................................................. 42
Half-Bridge Output Mode Applications Example ............... 42
Output Configurations .......................................................... 38
Output Relationships (Active High and Low) ...................... 40
Output Relationships (Active-High and Low) ....................... 39
Output Relationships Diagram ......................................... 39, 41
Programmable Dead-band Delay ...................................... 45
Setup for Operation ............................................................ 48
Shoot-through Current ....................................................... 45
Start-up Considerations ..................................................... 47
Q
Q-Clock ............................................................................. 37, 100
PIC16F716

R
RA3:RA0 ................................................................. 19
RA4/T0CKI Pin ......................................................... 20
RAM. See Data Memory.
RBO Pin .................................................................. 21
Register File ............................................................... 8
Register File Map ....................................................... 8
Registers
A/D ADCON0 ........................................................... 49
A/D ADCON1 .......................................................... 49
A/D ADDR .................................................................. 49, 50
ADCON0 ADCS1:ADCS0 Bits .................................... 49
ADCON0 ADON Bit .................................................. 49
ADCON0 CHS2:CHS0 Bits ........................................ 49
ADCON0 GO/DONE Bit .......................................... 49, 50
ADCON1 PCFG2:PCFG0 Bits .................................. 50
CCP1CON CCP1M3:CCP1M0 Bits ......................... 33
CCP1CON CCP1X:CCP1Y Bits ............................... 33
Compare (CCP Module)
CCPR1H:CCPR1L .................................................. 34
INTCON Register
RBIF ........................................................................ 21
PWM1CON (Enhanced PWM Configuration) ............ 46
T1CON Register
T1CKPS1:T1CKPS0 Bits ........................................ 29
T1OSCEN Bit ........................................................ 29
T1SYNC Bit .......................................................... 29
TM1CS Bit ............................................................ 29
TM1ON Bit .............................................................. 29
T2CON Register T2CKPS1:T2CKPS0 Bits ............... 31
T2CON Register TMR2ON Bit .................................. 31
T2CON Register TOUTPS3:TOUTPS0 Bits ............... 31
Timer2
PR2 ........................................................................ 36
Timer2 PR2 Register .............................................. 31
Timer2 TMR2 Register ........................................... 31
TMR1H Timer1 Register ......................................... 29
TMR1L Timer1 Register .......................................... 29
Reset ..................................................................... 55, 58
Brown-out Reset (BOR). See Brown-out Reset (BOR)
MCLR Reset. See MCLR
Power-on Reset (POR). See Power-on Reset (POR)
Reset Conditions for PCON Register ........................ 62
Reset Conditions for Program Counter ................... 62
Reset Conditions for Status Register ..................... 62
Timing Diagram ....................................................... 101
WDT Reset. See Watchdog Timer (WDT)
RETFIE Instruction .................................................. 80
RETLW Instruction .................................................... 81
RETURN Instruction ................................................ 81
Revision History ...................................................... 113
RLF Instruction ....................................................... 81
RRF instruction ....................................................... 82
S
Shoot-through Current ........................................... 45
Sleep ................................................................... 55, 58, 68
Sleep Instruction ..................................................... 82
Software Simulator (MPLAB SIM) ......................... 86
Software Simulator (MPLAB SIM30) ......................... 86
Special Event Trigger. See Compare
Special Features of the CPU ................................... 55
Special Function Registers .................................... 9
Speed, Operating ..................................................... 1
Stack ................................................................. 17
Status Register ...................................................... 9, 66
C Bit .................................................................. 11
DC Bit .................................................................. 11
IRP Bit .................................................................. 11
PD Bit ................................................................ 11, 58
RP1:RP0 Bits ....................................................... 11
TO Bit ................................................................ 11, 58
Z Bit ................................................................... 11
SUBLW Instruction .................................................. 82
SUBWF Instruction .................................................. 83
SWAPF Instruction .................................................. 83
T
T1CON Register ........................................................ 9
T2CON Register ........................................................ 9
Timer0 .................................................................. 27
Clock Source Edge Select (T0SE Bit) ...................... 12, 27
Clock Source Select (T0CS Bit) .............................. 12, 27
Overflow Enable (T0IE Bit) .................................... 13
Overflow Flag (T0IF Bit) ......................................... 13, 66
Overflow Interrupt .............................................. 28, 66
Prescaler. See Prescaler, Timer0
Timing Diagram .................................................... 102
TMR0 Register ........................................................ 9
Timer1 .................................................................. 29
Clock Source Select (T1M1CS Bit) ......................... 29
External Clock Input Sync (T1SYNC Bit) ................. 29
Module On/Off (TMR1ON Bit) ................................. 29
Oscillator ................................................................ 29, 30
Oscillator Enable (T1OSCEN Bit) ......................... 29
Overflow Enable (TMR1IE Bit) ............................. 14
Overflow Flag (TMR1IF Bit) .................................... 15
Overflow Interrupt .............................................. 29, 30
Prescaler. See Prescaler, Timer1
Special Event Trigger (CCP) .................................. 30, 35
T1CON Register ...................................................... 9
Timing Diagram .................................................... 102
TMR1H Register ..................................................... 9
TMR1L Register ..................................................... 9
Timer2
Postscaler. See Postscaler, Timer2
PR2 Register ........................................................ 10
Prescaler. See Prescaler, Timer2
T2CON Register ...................................................... 9
TMR2 Register ....................................................... 9
TMR2 to PR2 Match Enable (TMR2IE Bit) ............... 14
TMR2 to PR2 Match Flag (TMR2IF Bit) .................. 15
TMR2 to PR2 Match Interrupt ......................... 31, 32, 36
Timing Diagrams .................................................... 42
Half-Bridge PWM Output ........................................ 42
PWN Auto-Shutdown (PRSEN = 0, Auto-Restart Disabled) ......................................................... 47
PWN Auto-Shutdown (PRSEN = 1, Auto-Restart Enabled) ......................................................... 47
PWN Direction Change ........................................ 44
PWN Direction Change at Near 100% Duty Cycle .... 44
Time-out Sequence on Power-up ......................... 64
Wake-up from Sleep via Interrupt ......................... 69
Timing Diagrams and Specifications ..................... 98
A/D Conversion ..................................................... 105
Brown-out Reset (BOR) ......................................... 101
Capture/Compare/PWM (CCP) ......................... 103
CLKOUT and I/O .................................................... 100
External Clock ....................................................... 98
Oscillator Start-up Timer (OST) ......................... 101
Power-up Timer (PWRT) ......................................... 101

© 2003 Microchip Technology Inc.

Preliminary
Reset .................................................................................. 101
Timer0 and Timer1 ............................................................. 102
Watchdog Timer (WDT) .................................................. 101
TRIS Instruction .................................................................. 83

W
W Register .......................................................................... 66
Wake-up from Sleep .......................................................... 55, 68
Interrupts ................................................................. 62, 63
MCLR Reset ................................................................. 63
Timing Diagram .............................................................. 69
WDT Reset ........................................................................ 83
Watchdog Timer (WDT) .................................................. 55, 67
Enable (WDTE Bit) ..................................................... 56, 67
Postscaler. See Postscaler, WDT
Programming Considerations ...................................... 67
RC Oscillator ................................................................. 67
Time-out Period ............................................................. 67
Timing Diagram .............................................................. 101
WDT Reset, Normal Operation .................................... 58, 62, 63
WDT Reset, Sleep .......................................................... 58, 62, 63
WWW, On-Line Support ..................................................... 3

X
XORLW Instruction .......................................................... 84
XORWF Instruction .......................................................... 84
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>PIC16F716, PIC16F716T, 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>SO = SOIC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>P = PDIP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SS = SSOP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pattern</td>
<td>QTP, SQTP, Code or Special Requirements</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(blank otherwise)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Examples:

a) PIC16F716 -I/P 301 = Industrial temp., PDIP package, QTP pattern #301.
b) PIC16F716 - E/SO = Extended temp, SOIC package

Note 1: T = in tape and reel SOIC and SSOP packages only.

Sales and Support

Data Sheets

Products supported by a preliminary Data Sheet may have an errata sheet describing minor operational differences and recommended workarounds. To determine if an errata sheet exists for a particular device, please contact one of the following:

1. Your local Microchip sales office
2. The Microchip Corporate Literature Center U.S. FAX: (480) 792-7277
3. The Microchip Worldwide Site (www.microchip.com)

Please specify which device, revision of silicon and Data Sheet (include Literature #) you are using.

New Customer Notification System

Register on our web site (www.microchip.com/cn) to receive the most current information on our products.