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

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

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

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE**.** Microchip disclaims all liability arising from this information and its use. Use of Microchip'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 Microchip intellectual property rights.

**Trademarks**

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

AmpLab, FilterLab, MXDEV, MXLAB, PICMASTER, SEEVAL, SmartSensor and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, Application Maestro, dsPICDEM, dsPICDEM.net, dsPICworks, ECAN, ECONOMONITOR, FanSense, FlexROM, fuzzyLAB, In-Circuit Serial Programming, ICSP, ICEPIC, Migratable Memory, MPASM, MPLIB, MPLINK, MPSIM, PICkit, PICDEM, PICDEM.net, PICLAB, PICtail, PowerCal, PowerInfo, PowerMate, PowerTool, rfLAB, rfPICDEM, Select Mode, Smart Serial, SmartTel and Total Endurance are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

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

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

© 2004, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

Printed on recycled paper.

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

*Microchip received ISO/TS-16949:2002 quality system certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona and Mountain View, California in October 2003. The Company's quality system processes and procedures are for its PICmicro® 8-bit MCUs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.*


© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

# PIC16F818/819

## 18/20-Pin Enhanced Flash Microcontrollers with nanoWatt Technology

### Low-Power Features:

- Power-Managed modes:
  - Primary Run: XT, RC oscillator, 87 µA, 1 MHz, 2V
  - INTRC: 7 µA, 31.25 kHz, 2V
  - Sleep: 0.2 µA, 2V
- Timer1 oscillator: 1.8 µA, 32 kHz, 2V
- Watchdog Timer: 0.7 µA, 2V
- Wide operating voltage range:
  - Industrial: 2.0V to 5.5V

### Oscillators:

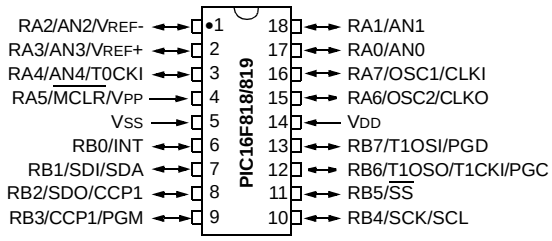
- Three Crystal modes:
  - LP, XT, HS: up to 20 MHz
- Two External RC modes
- One External Clock mode:
  - ECIO: up to 20 MHz
- Internal oscillator block:
  - 8 user selectable frequencies: 31 kHz, 125 kHz, 250 kHz, 500 kHz, 1 MHz, 2 MHz, 4 MHz, 8 MHz

### Peripheral Features:

- 16 I/O pins with individual direction control
- High sink/source current: 25 mA
- 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
- Capture, Compare, PWM (CCP) module:
  - Capture is 16-bit, max. resolution is 12.5 ns
  - Compare is 16-bit, max. resolution is 200 ns
  - PWM max. resolution is 10-bit
- 10-bit, 5-channel Analog-to-Digital converter
- Synchronous Serial Port (SSP) with SPI™ (Master/Slave) and I²C™ (Slave)

### Pin Diagram

**18-Pin PDIP, SOIC**




### Special Microcontroller Features:

- 100,000 erase/write cycles Enhanced Flash program memory typical
- 1,000,000 typical erase/write cycles EEPROM data memory typical
- EEPROM Data Retention: > 40 years
- In-Circuit Serial Programming™ (ICSP™) via two pins
- Processor read/write access to program memory
- Low-Voltage Programming
- In-Circuit Debugging via two pins

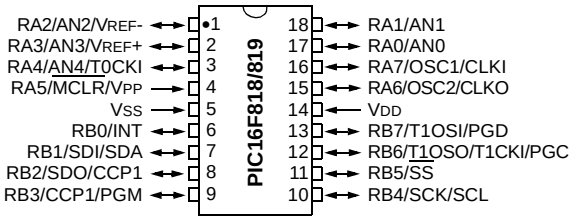
| Device | Program Memory | | Data Memory | | I/O Pins | 10-bit A/D (ch) | CCP (PWM) | SSP | | Timers 8/16-bit |
|---|---|---|---|---|---|---|---|---|---|---|
| | Flash (Bytes) | # Single-Word Instructions | SRAM (Bytes) | EEPROM (Bytes) | | | | SPI™ | Slave I²C™ | |
| PIC16F818 | 1792 | 1024 | 128 | 128 | 16 | 5 | 1 | Y | Y | 2/1 |
| PIC16F819 | 3584 | 2048 | 256 | 256 | 16 | 5 | 1 | Y | Y | 2/1 |

© 2004 Microchip Technology Inc.

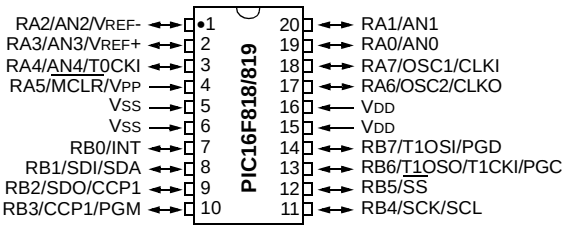

Downloaded from Arrow.com.

## Pin Diagrams

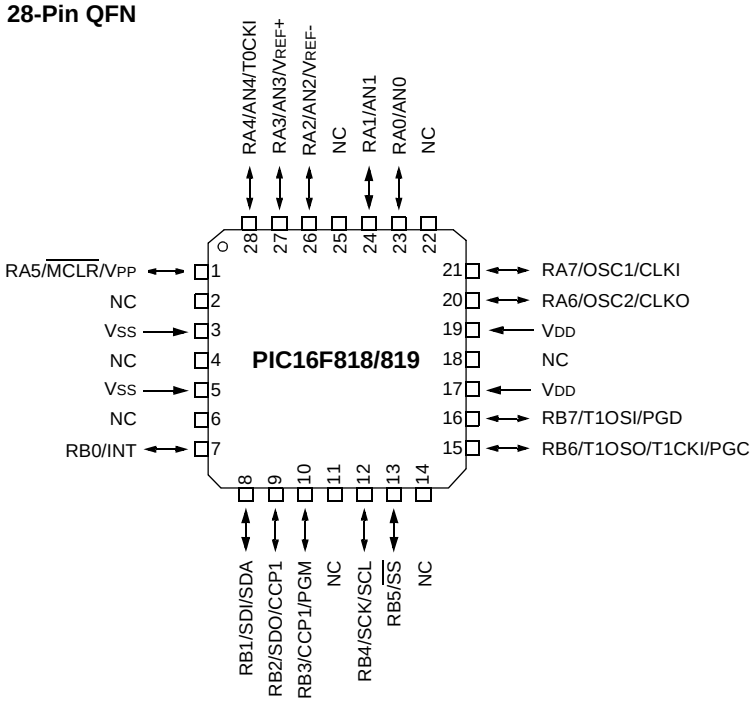
### 18-Pin PDIP, SOIC




### 20-Pin SSOP




### 28-Pin QFN





© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

## Table of Contents

1.0 Device Overview ..... 5
2.0 Memory Organization ..... 9
3.0 Data EEPROM and Flash Program Memory ..... 25
4.0 Oscillator Configurations ..... 33
5.0 I/O Ports ..... 39
6.0 Timer0 Module ..... 53
7.0 Timer1 Module ..... 57
8.0 Timer2 Module ..... 63
9.0 Capture/Compare/PWM (CCP) Module ..... 65
10.0 Synchronous Serial Port (SSP) Module ..... 71
11.0 Analog-to-Digital Converter (A/D) Module ..... 81
12.0 Special Features of the CPU ..... 89
13.0 Instruction Set Summary ..... 103
14.0 Development Support ..... 111
15.0 Electrical Characteristics ..... 117
16.0 DC and AC Characteristics Graphs and Tables ..... 143
17.0 Packaging Information ..... 157
Appendix A: Revision History ..... 163
Appendix B: Device Differences ..... 163
Index ..... 165
On-Line Support ..... 171
Systems Information and Upgrade Hot Line ..... 171
Reader Response ..... 172
PIC16F818/819 Product Identification System ..... 173

### TO OUR VALUED CUSTOMERS

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

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

#### Most Current Data Sheet

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

http://www.microchip.com

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

#### Errata

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

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

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

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

#### Customer Notification System

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



Downloaded from Arrow.com.

**NOTES:**


© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

## 1.0 DEVICE OVERVIEW

This document contains device specific information for the operation of the PIC16F818/819 devices. Additional information may be found in the *"PICmicro® Mid-Range MCU Family Reference Manual"* (DS33023) which may be downloaded from the Microchip web site. The Reference Manual should be considered a complementary document to this data sheet and is highly recommended reading for a better understanding of the device architecture and operation of the peripheral modules.

The PIC16F818/819 belongs to the Mid-Range family of the PICmicro® devices. The devices differ from each other in the amount of Flash program memory, data memory and data EEPROM (see Table 1-1). A block diagram of the devices is shown in Figure 1-1. These devices contain features that are new to the PIC16 product line:

- Internal RC oscillator with eight selectable frequencies, including 31.25 kHz, 125 kHz, 250 kHz, 500 kHz, 1 MHz, 2 MHz, 4 MHz and 8 MHz. The INTRC can be configured as the system clock via the configuration bits. Refer to **Section 4.5 "Internal Oscillator Block"** and **Section 12.1 "Configuration Bits"** for further details.
- The Timer1 module current consumption has been greatly reduced from 20 µA (previous PIC16 devices) to 1.8 µA typical (32 kHz at 2V), which is ideal for real-time clock applications. Refer to **Section 6.0 "Timer0 Module"** for further details.
- The amount of oscillator selections has increased. The RC and INTRC modes can be selected with an I/O pin configured as an I/O or a clock output (FOSC/4). An external clock can be configured with an I/O pin. Refer to **Section 4.0 "Oscillator Configurations"** for further details.

**TABLE 1-1: AVAILABLE MEMORY IN PIC16F818/819 DEVICES**

| Device | Program Flash | Data Memory | Data EEPROM |
|---|---|---|---|
| PIC16F818 | 1K x 14 | 128 x 8 | 128 x 8 |
| PIC16F819 | 2K x14 | 256 x 8 | 256 x 8 |

There are 16 I/O pins that are user configurable on a pin-to-pin basis. Some pins are multiplexed with other device functions. These functions include:

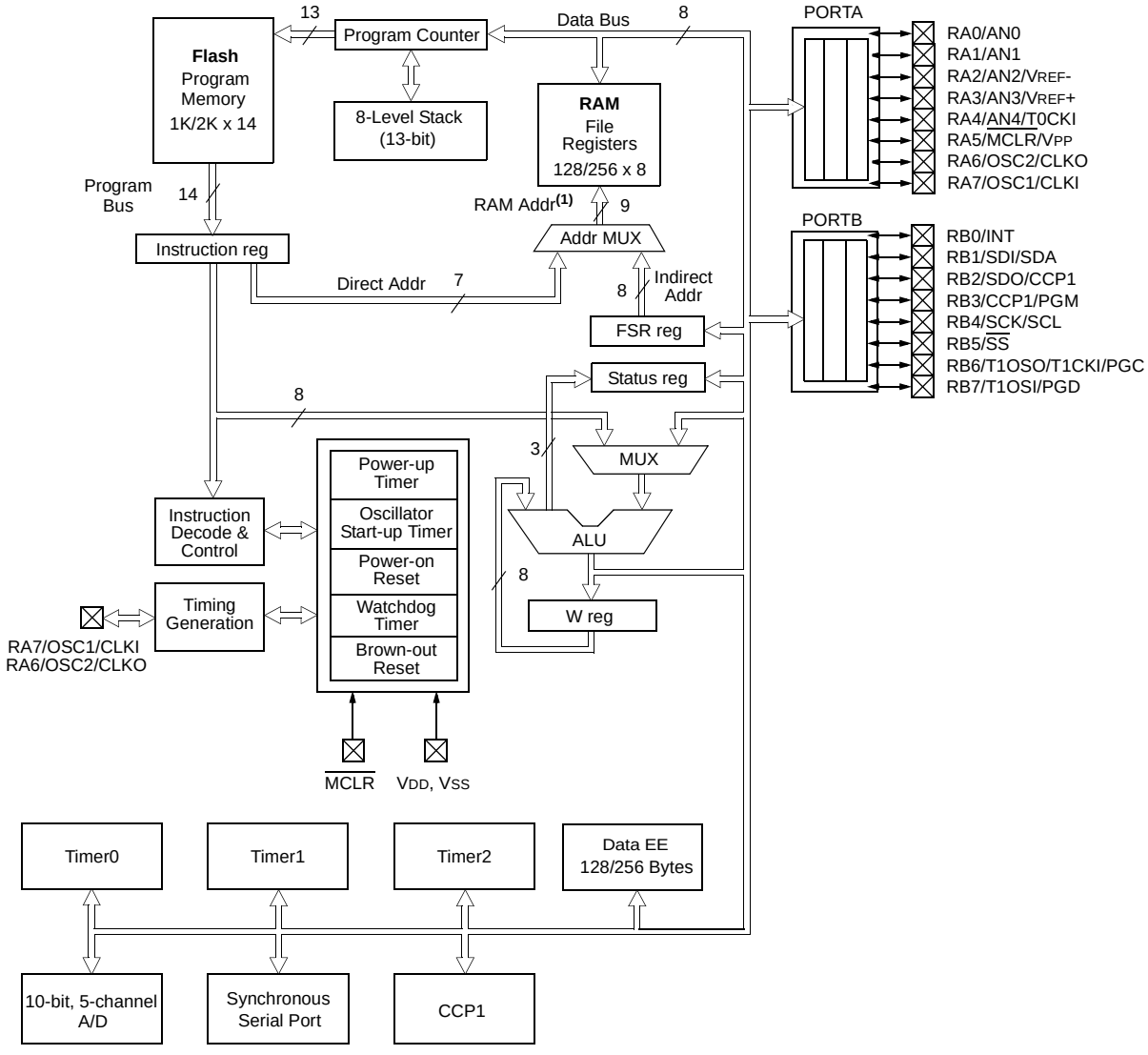
- External Interrupt
- Change on PORTB Interrupt
- Timer0 Clock Input
- Low-Power Timer1 Clock/Oscillator
- Capture/Compare/PWM
- 10-bit, 5-channel Analog-to-Digital Converter
- SPI/$I^2C$
- $\overline{MCLR}$ (RA5) can be configured as an Input

Table 1-2 details the pinout of the devices with descriptions and details for each pin.



Downloaded from Arrow.com.

**FIGURE 1-1: PIC16F818/819 BLOCK DIAGRAM**




**Note 1:** Higher order bits are from the Status register.


© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

**TABLE 1-2: PIC16F818/819 PINOUT DESCRIPTIONS**

| Pin Name | PDIP/ SOIC Pin# | SSOP Pin# | QFN Pin# | I/O/P Type | Buffer Type | Description |
|---|---|---|---|---|---|---|
| | | | | | | PORTA is a bidirectional I/O port. |
| RA0/AN0 | 17 | 19 | 23 | | | |
| RA0 | | | | I/O | TTL | Bidirectional I/O pin. |
| AN0 | | | | I | Analog | Analog input channel 0. |
| RA1/AN1 | 18 | 20 | 24 | | | |
| RA1 | | | | I/O | TTL | Bidirectional I/O pin. |
| AN1 | | | | I | Analog | Analog input channel 1. |
| RA2/AN2/VREF- | 1 | 1 | 26 | | | |
| RA2 | | | | I/O | TTL | Bidirectional I/O pin. |
| AN2 | | | | I | Analog | Analog input channel 2. |
| VREF- | | | | I | Analog | A/D reference voltage (low) input. |
| RA3/AN3/VREF+ | 2 | 2 | 27 | | | |
| RA3 | | | | I/O | TTL | Bidirectional I/O pin. |
| AN3 | | | | I | Analog | Analog input channel 3. |
| VREF+ | | | | I | Analog | A/D reference voltage (high) input. |
| RA4/AN4/T0CKI | 3 | 3 | 28 | | | |
| RA4 | | | | I/O | ST | Bidirectional I/O pin. |
| AN4 | | | | I | Analog | Analog input channel 4. |
| T0CKI | | | | I | ST | Clock input to the TMR0 timer/counter. |
| RA5/$\overline{\text{MCLR}}$/VPP | 4 | 4 | 1 | | | |
| RA5 | | | | I | ST | Input pin. |
| $\overline{\text{MCLR}}$ | | | | I | ST | Master Clear (Reset). Input/programming voltage input. This pin is an active-low Reset to the device. |
| VPP | | | | P | – | Programming threshold voltage. |
| RA6/OSC2/CLKO | 15 | 17 | 20 | | | |
| RA6 | | | | I/O | ST | Bidirectional I/O pin. |
| OSC2 | | | | O | – | Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode. |
| CLKO | | | | O | – | In RC mode, this pin outputs CLKO signal which has 1/4 the frequency of OSC1 and denotes the instruction cycle rate. |
| RA7/OSC1/CLKI | 16 | 18 | 21 | | | |
| RA7 | | | | I/O | ST | Bidirectional I/O pin. |
| OSC1 | | | | I | ST/CMOS(3) | Oscillator crystal input. |
| CLKI | | | | I | – | External clock source input. |

**Legend:** I = Input O = Output I/O = Input/Output P = Power
– = Not used TTL = TTL Input ST = Schmitt Trigger Input

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

**2:** This buffer is a Schmitt Trigger input when used in Serial Programming mode.

**3:** This buffer is a Schmitt Trigger input when configured in RC Oscillator mode and a CMOS input otherwise.



Downloaded from Arrow.com.

**TABLE 1-2: PIC16F818/819 PINOUT DESCRIPTIONS (CONTINUED)**

| Pin Name | PDIP/ SOIC Pin# | SSOP Pin# | QFN Pin# | I/O/P Type | Buffer Type | Description |
|---|---|---|---|---|---|---|
| | | | | | | PORTB is a bidirectional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs. |
| RB0/INT | 6 | 7 | 7 | | | |
| RB0 | | | | I/O | TTL | Bidirectional I/O pin. |
| INT | | | | I | ST(1) | External interrupt pin. |
| RB1/SDI/SDA | 7 | 8 | 8 | | | |
| RB1 | | | | I/O | TTL | Bidirectional I/O pin. |
| SDI | | | | I | ST | SPI™ data in. |
| SDA | | | | I/O | ST | $I^2C$™ data. |
| RB2/SDO/CCP1 | 8 | 9 | 9 | | | |
| RB2 | | | | I/O | TTL | Bidirectional I/O pin. |
| SDO | | | | O | ST | SPI data out. |
| CCP1 | | | | I/O | ST | Capture input, Compare output, PWM output. |
| RB3/CCP1/PGM | 9 | 10 | 10 | | | |
| RB3 | | | | I/O | TTL | Bidirectional I/O pin. |
| CCP1 | | | | I/O | ST | Capture input, Compare output, PWM output. |
| PGM | | | | I | ST | Low-Voltage ICSP™ Programming enable pin. |
| RB4/SCK/SCL | 10 | 11 | 12 | | | |
| RB4 | | | | I/O | TTL | Bidirectional I/O pin. Interrupt-on-change pin. |
| SCK | | | | I/O | ST | Synchronous serial clock input/output for SPI. |
| SCL | | | | I | ST | Synchronous serial clock input for $I^2C$. |
| RB5/$\overline{SS}$ | 11 | 12 | 13 | | | |
| RB5 | | | | I/O | TTL | Bidirectional I/O pin. Interrupt-on-change pin. |
| $\overline{SS}$ | | | | I | TTL | Slave select for SPI in Slave mode. |
| RB6/T1OSO/T1CKI/PGC | 12 | 13 | 15 | | | |
| RB6 | | | | I/O | TTL | Interrupt-on-change pin. |
| T1OSO | | | | O | ST | Timer1 Oscillator output. |
| T1CKI | | | | I | ST | Timer1 clock input. |
| PGC | | | | I | ST(2) | In-circuit debugger and ICSP programming clock pin. |
| RB7/T1OSI/PGD | 13 | 14 | 16 | | | |
| RB7 | | | | I/O | TTL | Interrupt-on-change pin. |
| T1OSI | | | | I | ST | Timer1 oscillator input. |
| PGD | | | | I | ST(2) | In-circuit debugger and ICSP programming data pin. |
| VSS | 5 | 5, 6 | 3, 5 | P | – | Ground reference for logic and I/O pins. |
| VDD | 14 | 15, 16 | 17, 19 | P | – | Positive supply for logic and I/O pins. |

**Legend:** I = Input O = Output I/O = Input/Output P = Power
– = Not used TTL = TTL Input ST = Schmitt Trigger Input

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

**2:** This buffer is a Schmitt Trigger input when used in Serial Programming mode.

**3:** This buffer is a Schmitt Trigger input when configured in RC Oscillator mode and a CMOS input otherwise.


© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

## 2.0 MEMORY ORGANIZATION

There are two memory blocks in the PIC16F818/819. These are the program memory and the data memory. Each block has its own bus, so access to each block can occur during the same oscillator cycle.

The data memory can be further broken down into the general purpose RAM and the Special Function Registers (SFRs). The operation of the SFRs that control the "core" are described here. The SFRs used to control the peripheral modules are described in the section discussing each individual peripheral module.

The data memory area also contains the data EEPROM memory. This memory is not directly mapped into the data memory but is indirectly mapped. That is, an indirect address pointer specifies the address of the data EEPROM memory to read/write. The PIC16F818 device's 128 bytes of data EEPROM memory have the address range of 00h-7Fh and the PIC16F819 device's 256 bytes of data EEPROM memory have the address range of 00h-FFh. More details on the EEPROM memory can be found in **Section 3.0 "Data EEPROM and Flash Program Memory"**.

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

### 2.1 Program Memory Organization

The PIC16F818/819 devices have a 13-bit program counter capable of addressing an 8K x 14 program memory space. For the PIC16F818, the first 1K x 14 (0000h-03FFh) is physically implemented (see Figure 2-1). For the PIC16F819, the first 2K x 14 is located at 0000h-07FFh (see Figure 2-2). Accessing a location above the physically implemented address will cause a wraparound. For example, the same instruction will be accessed at locations 020h, 420h, 820h, C20h, 1020h, 1420h, 1820h and 1C20h.

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

**FIGURE 2-1: PROGRAM MEMORY MAP AND STACK FOR PIC16F818**

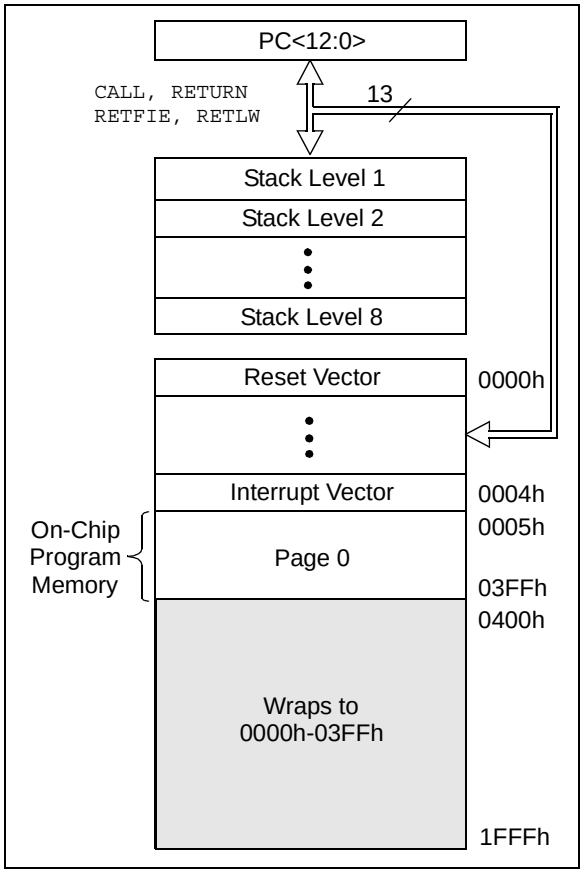
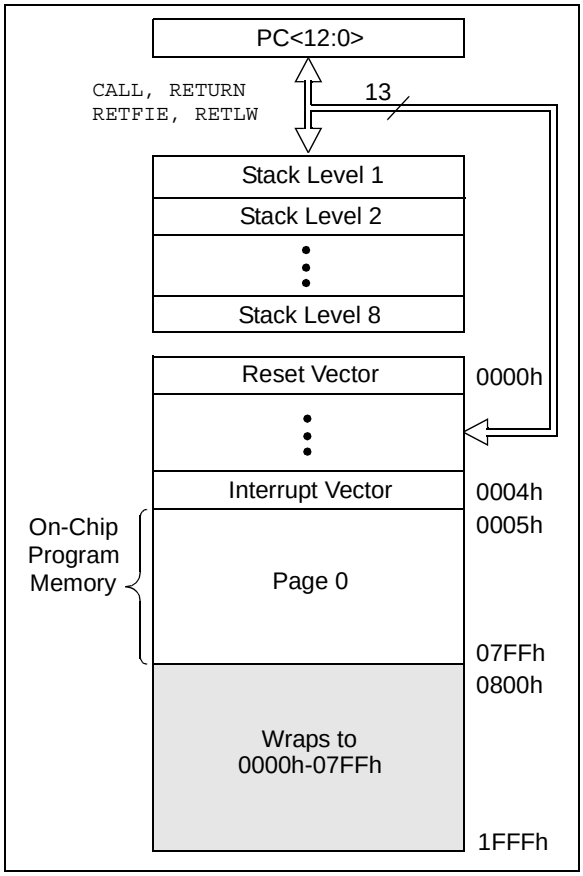



**FIGURE 2-2: PROGRAM MEMORY MAP AND STACK FOR PIC16F819**






Downloaded from Arrow.com.

## 2.2 Data Memory Organization

The data memory is partitioned into multiple banks that contain the General Purpose Registers and the Special Function Registers. Bits RP1 (Status<6>) and RP0 (Status<5>) are the bank select bits.

| RP1:RP0 | Bank |
|---|---|
| 00 | 0 |
| 01 | 1 |
| 10 | 2 |
| 11 | 3 |

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 the General Purpose Registers, implemented as static RAM. All implemented banks contain SFRs. Some “high use” SFRs from one bank may be mirrored in another bank for code reduction and quicker access (e.g., the Status register is in Banks 0-3).

| **Note:** | EEPROM data memory description can be found in **Section 3.0 “Data EEPROM and Flash Program Memory”** of this data sheet. |
|---|---|

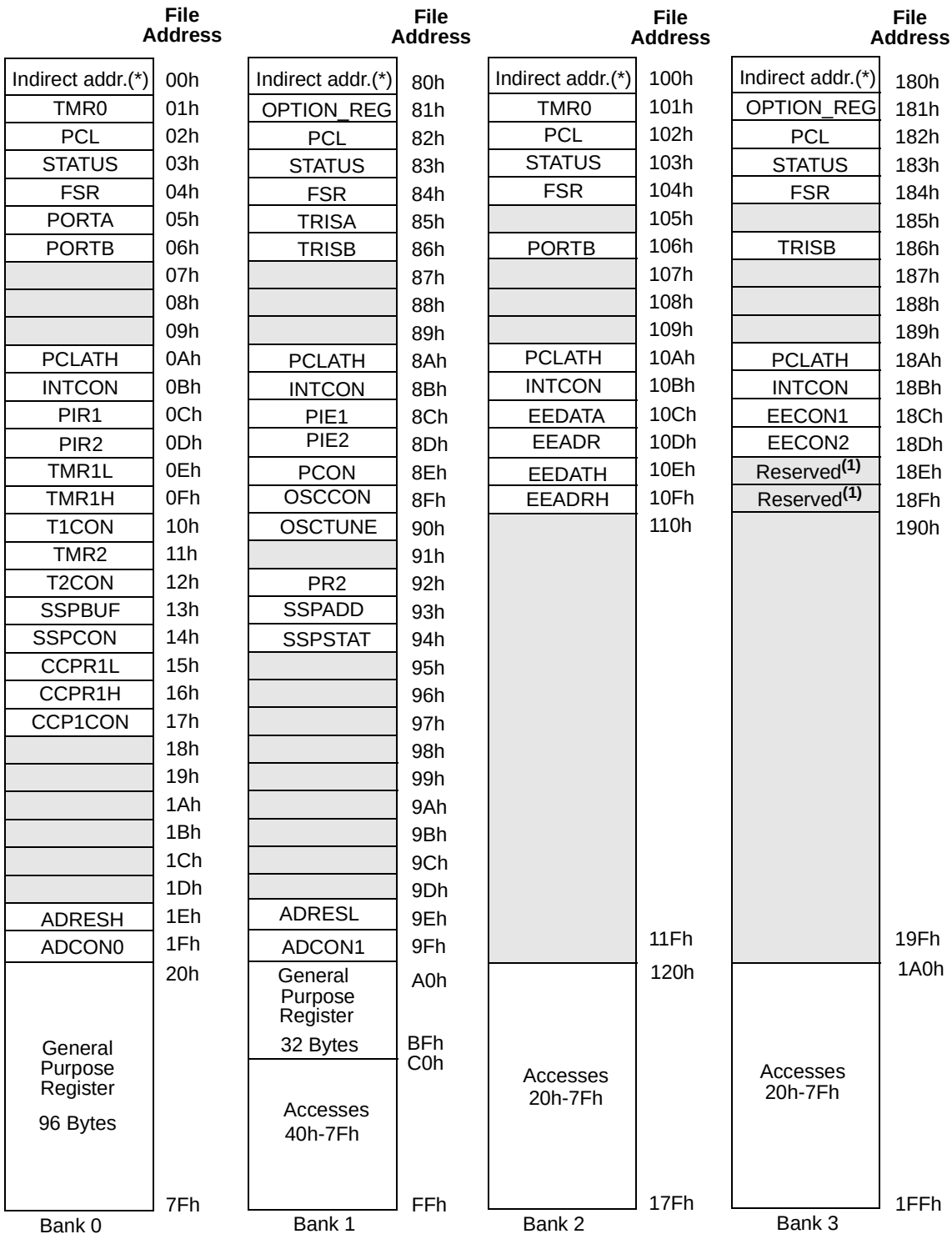
### 2.2.1 GENERAL PURPOSE REGISTER FILE

The register file can be accessed either directly or indirectly through the File Select Register, FSR.


© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

**FIGURE 2-3: PIC16F818 REGISTER FILE MAP**




Unimplemented data memory locations, read as '0'.

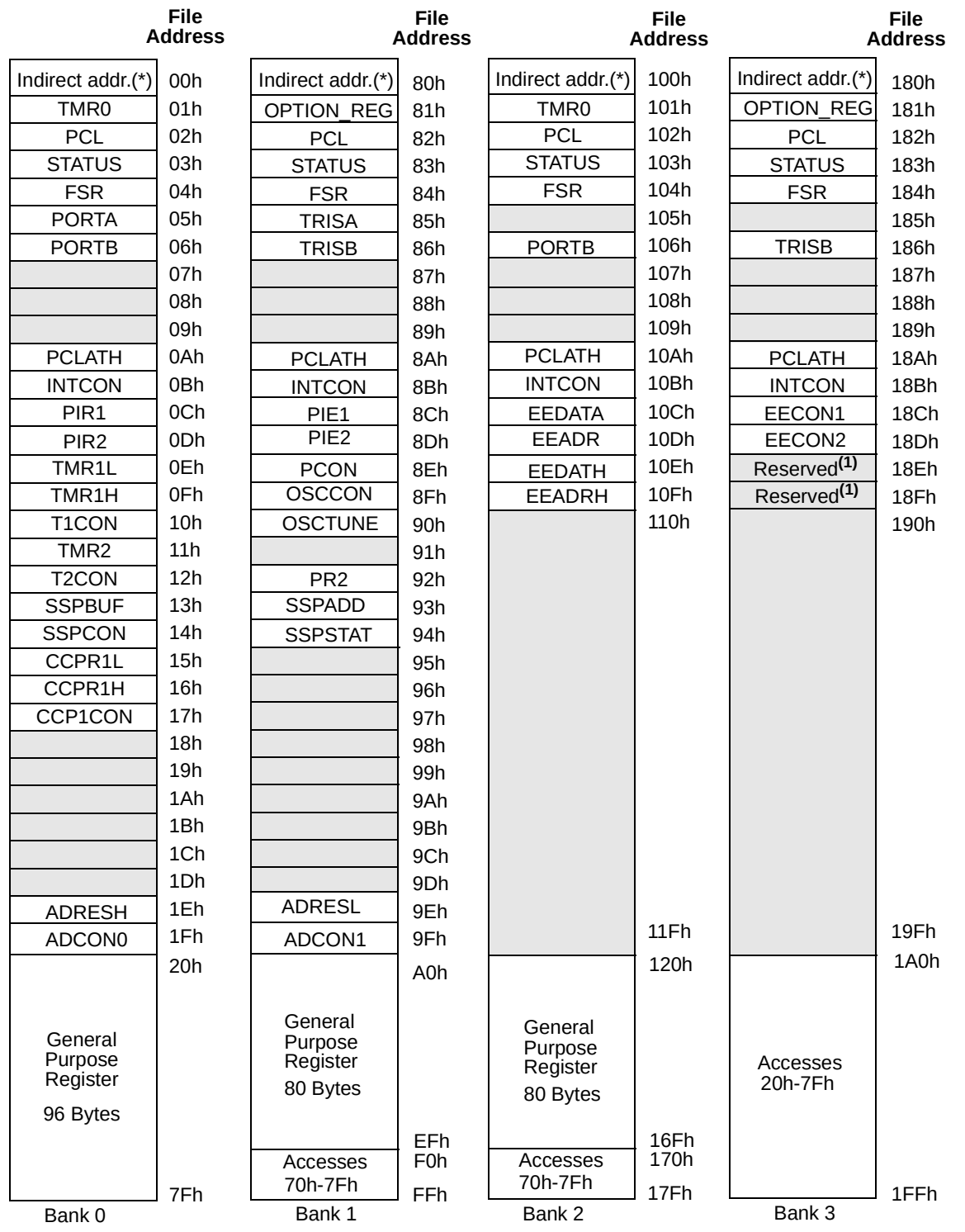
* Not a physical register.

**Note 1:** These registers are reserved; maintain these registers clear.



Downloaded from Arrow.com.

**FIGURE 2-4: PIC16F819 REGISTER FILE MAP**




Unimplemented data memory locations, read as '0'.

* Not a physical register.

**Note 1:** These registers are reserved; maintain these registers clear.



© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

### 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 the peripheral feature section.

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

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR, BOR | Details on page: |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Bank 0** | | | | | | | | | | | |
| 00h(1) | INDF | Addressing this location uses contents of FSR to address data memory (not a physical register) | | | | | | | | 0000 0000 | 23 |
| 01h | TMR0 | Timer0 Module Register | | | | | | | | xxxx xxxx | 53, 17 |
| 02h(1) | PCL | Program Counter's (PC) Least Significant Byte | | | | | | | | 0000 0000 | 23 |
| 03h(1) | STATUS | IRP | RP1 | RP0 | TO | PD | Z | DC | C | 0001 1xxx | 16 |
| 04h(1) | FSR | Indirect Data Memory Address Pointer | | | | | | | | xxxx xxxx | 23 |
| 05h | PORTA | PORTA Data Latch when written; PORTA pins when read | | | | | | | | xxx0 0000 | 39 |
| 06h | PORTB | PORTB Data Latch when written; PORTB pins when read | | | | | | | | xxxx xxxx | 43 |
| 07h | — | Unimplemented | | | | | | | | — | — |
| 08h | — | Unimplemented | | | | | | | | — | — |
| 09h | — | Unimplemented | | | | | | | | — | — |
| 0Ah(1,2) | PCLATH | — | — | — | Write Buffer for the upper 5 bits of the Program Counter | | | | | ---0 0000 | 23 |
| 0Bh(1) | INTCON | GIE | PEIE | TMR0IE | INTE | RBIE | TMR0IF | INTF | RBIF | 0000 000x | 18 |
| 0Ch | PIR1 | — | ADIF | — | — | SSPIF | CCP1IF | TMR2IF | TMR1IF | -0-- 0000 | 20 |
| 0Dh | PIR2 | — | — | — | EEIF | — | — | — | — | ---0 ---- | 21 |
| 0Eh | TMR1L | Holding Register for the Least Significant Byte of the 16-bit TMR1 Register | | | | | | | | xxxx xxxx | 57 |
| 0Fh | TMR1H | Holding Register for the Most Significant Byte of the 16-bit TMR1 Register | | | | | | | | xxxx xxxx | 57 |
| 10h | T1CON | — | — | T1CKPS1 | T1CKPS0 | T1OSCEN | $\overline{\text{T1SYNC}}$ | TMR1CS | TMR1ON | --00 0000 | 57 |
| 11h | TMR2 | Timer2 Module Register | | | | | | | | 0000 0000 | 63 |
| 12h | T2CON | — | TOUTPS3 | TOUTPS2 | TOUTPS1 | TOUTPS0 | TMR2ON | T2CKPS1 | T2CKPS0 | -000 0000 | 64 |
| 13h | SSPBUF | Synchronous Serial Port Receive Buffer/Transmit Register | | | | | | | | xxxx xxxx | 71, 76 |
| 14h | SSPCON | WCOL | SSPOV | SSPEN | CKP | SSPM3 | SSPM2 | SSPM1 | SSPM0 | 0000 0000 | 73 |
| 15h | CCPR1L | Capture/Compare/PWM Register (LSB) | | | | | | | | xxxx xxxx | 66, 67, 68 |
| 16h | CCPR1H | Capture/Compare/PWM Register (MSB) | | | | | | | | xxxx xxxx | 66, 67, 68 |
| 17h | CCP1CON | — | — | CCP1X | CCP1Y | CCP1M3 | CCP1M2 | CCP1M1 | CCP1M0 | --00 0000 | 65 |
| 18h | — | Unimplemented | | | | | | | | — | — |
| 19h | — | Unimplemented | | | | | | | | — | — |
| 1Ah | — | Unimplemented | | | | | | | | — | — |
| 1Bh | — | Unimplemented | | | | | | | | — | — |
| 1Ch | — | Unimplemented | | | | | | | | — | — |
| 1Dh | — | Unimplemented | | | | | | | | — | — |
| 1Eh | ADRESH | A/D Result Register High Byte | | | | | | | | xxxx xxxx | 81 |
| 1Fh | ADCON0 | ADCS1 | ADCS0 | CHS2 | CHS1 | CHS0 | GO/$\overline{\text{DONE}}$ | — | ADON | 0000 00-0 | 81 |

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

**Note 1:** These registers can be addressed from any bank.

**2:** The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8>, whose contents are transferred to the upper byte of the program counter.

**3:** Pin 5 is an input only; the state of the TRISA5 bit has no effect and will always read '1'.



Downloaded from Arrow.com.

**TABLE 2-1: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)**

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR, BOR | Details on page: |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Bank 1** | | | | | | | | | | | |
| 80h(1) | INDF | Addressing this location uses contents of FSR to address data memory (not a physical register) | | | | | | | | 0000 0000 | 23 |
| 81h | OPTION_REG | $\overline{\text{RBPU}}$ | INTEDG | T0CS | T0SE | PSA | PS2 | PS1 | PS0 | 1111 1111 | 17, 54 |
| 82h(1) | PCL | Program Counter's (PC) Least Significant Byte | | | | | | | | 0000 0000 | 23 |
| 83h(1) | STATUS | IRP | RP1 | RP0 | $\overline{\text{TO}}$ | $\overline{\text{PD}}$ | Z | DC | C | 0001 1xxx | 16 |
| 84h(1) | FSR | Indirect Data Memory Address Pointer | | | | | | | | xxxx xxxx | 23 |
| 85h | TRISA | TRISA7 | TRISA6 | TRISA5(3) | PORTA Data Direction Register (TRISA<4:0> | | | | | 1111 1111 | 39 |
| 86h | TRISB | PORTB Data Direction Register | | | | | | | | 1111 1111 | 43 |
| 87h | — | Unimplemented | | | | | | | | — | — |
| 88h | — | Unimplemented | | | | | | | | — | — |
| 89h | — | Unimplemented | | | | | | | | — | — |
| 8Ah(1,2) | PCLATH | — | — | — | Write Buffer for the upper 5 bits of the PC | | | | | ---0 0000 | 23 |
| 8Bh(1) | INTCON | GIE | PEIE | TMR0IE | INTE | RBIE | TMR0IF | INTF | RBIF | 0000 000x | 18 |
| 8Ch | PIE1 | — | ADIE | — | — | SSPIE | CCP1IE | TMR2IE | TMR1IE | -0-- 0000 | 19 |
| 8Dh | PIE2 | — | — | — | EEIE | — | — | — | — | ---0 ---- | 21 |
| 8Eh | PCON | — | — | — | — | — | — | $\overline{\text{POR}}$ | $\overline{\text{BOR}}$ | ---- --qq | 22 |
| 8Fh | OSCCON | — | IRCF2 | IRCF1 | IRCF0 | — | IOFS | — | — | -000 -0-- | 38 |
| 90h(1) | OSCTUNE | — | — | TUN5 | TUN4 | TUN3 | TUN2 | TUN1 | TUN0 | --00 0000 | 36 |
| 91h | — | Unimplemented | | | | | | | | — | — |
| 92h | PR2 | Timer2 Period Register | | | | | | | | 1111 1111 | 68 |
| 93h | SSPADD | Synchronous Serial Port ($I^2C$™ mode) Address Register | | | | | | | | 0000 0000 | 71, 76 |
| 94h | SSPSTAT | SMP | CKE | D/$\overline{\text{A}}$ | P | S | R/$\overline{\text{W}}$ | UA | BF | 0000 0000 | 72 |
| 95h | — | Unimplemented | | | | | | | | — | — |
| 96h | — | Unimplemented | | | | | | | | — | — |
| 97h | — | Unimplemented | | | | | | | | — | — |
| 98h | — | Unimplemented | | | | | | | | — | — |
| 99h | — | Unimplemented | | | | | | | | — | — |
| 9Ah | — | Unimplemented | | | | | | | | — | — |
| 9Bh | — | Unimplemented | | | | | | | | — | — |
| 9Ch | — | Unimplemented | | | | | | | | — | — |
| 9Dh | — | Unimplemented | | | | | | | | — | — |
| 9Eh | ADRESL | A/D Result Register Low Byte | | | | | | | | xxxx xxxx | 81 |
| 9Fh | ADCON1 | ADFM | ADCS2 | — | — | PCFG3 | PCFG2 | PCFG1 | PCFG0 | 00-- 0000 | 82 |

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

**Note 1:** These registers can be addressed from any bank.

**2:** The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8>, whose contents are transferred to the upper byte of the program counter.

**3:** Pin 5 is an input only; the state of the TRISA5 bit has no effect and will always read '1'.


© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

**TABLE 2-1: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)**

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR, BOR | Details on page: |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Bank 2** | | | | | | | | | | | |
| 100h(1) | INDF | Addressing this location uses contents of FSR to address data memory (not a physical register) | | | | | | | | 0000 0000 | 23 |
| 101h | TMR0 | Timer0 Module Register | | | | | | | | xxxx xxxx | 53 |
| 102h(1 | PCL | Program Counter's (PC) Least Significant Byte | | | | | | | | 0000 0000 | 23 |
| 103h(1) | STATUS | IRP | RP1 | RP0 | $\overline{TO}$ | $\overline{PD}$ | Z | DC | C | 0001 1xxx | 16 |
| 104h(1) | FSR | Indirect Data Memory Address Pointer | | | | | | | | xxxx xxxx | 23 |
| 105h | — | Unimplemented | | | | | | | | — | — |
| 106h | PORTB | PORTB Data Latch when written; PORTB pins when read | | | | | | | | xxxx xxxx | 43 |
| 107h | — | Unimplemented | | | | | | | | — | — |
| 108h | — | Unimplemented | | | | | | | | — | — |
| 109h | — | Unimplemented | | | | | | | | — | — |
| 10Ah(1,2) | PCLATH | — | — | — | Write Buffer for the upper 5 bits of the Program Counter | | | | | ---0 0000 | 23 |
| 10Bh(1) | INTCON | GIE | PEIE | TMR0IE | INTE | RBIE | TMR0IF | INTF | RBIF | 0000 000x | 18 |
| 10Ch | EEDATA | EEPROM/Flash Data Register Low Byte | | | | | | | | xxxx xxxx | 25 |
| 10Dh | EEADR | EEPROM/Flash Address Register Low Byte | | | | | | | | xxxx xxxx | 25 |
| 10Eh | EEDATH | — | — | EEPROM/Flash Data Register High Byte | | | | | | --xx xxxx | 25 |
| 10Fh | EEADRH | — | — | — | — | — | EEPROM/Flash Address Register High Byte | | | ---- -xxx | 25 |
| **Bank 3** | | | | | | | | | | | |
| 180h(1) | INDF | Addressing this location uses contents of FSR to address data memory (not a physical register) | | | | | | | | 0000 0000 | 23 |
| 181h | OPTION_REG | $\overline{RBPU}$ | INTEDG | T0CS | T0SE | PSA | PS2 | PS1 | PS0 | 1111 1111 | 17, 54 |
| 182h(1) | PCL | Program Counter's (PC) Least Significant Byte | | | | | | | | 0000 0000 | 23 |
| 183h(1) | STATUS | IRP | RP1 | RP0 | $\overline{TO}$ | $\overline{PD}$ | Z | DC | C | 0001 1xxx | 16 |
| 184h(1) | FSR | Indirect Data Memory Address Pointer | | | | | | | | xxxx xxxx | 23 |
| 185h | — | Unimplemented | | | | | | | | — | — |
| 186h | TRISB | PORTB Data Direction Register | | | | | | | | 1111 1111 | 43 |
| 187h | — | Unimplemented | | | | | | | | — | — |
| 188h | — | Unimplemented | | | | | | | | — | — |
| 189h | — | Unimplemented | | | | | | | | — | — |
| 18Ah(1,2) | PCLATH | — | — | — | Write Buffer for the upper 5 bits of the Program Counter | | | | | ---0 0000 | 23 |
| 18Bh(1) | INTCON | GIE | PEIE | TMR0IE | INTE | RBIE | TMR0IF | INTF | RBIF | 0000 000x | 18 |
| 18Ch | EECON1 | EEPGD | — | — | FREE | WRERR | WREN | WR | RD | x--x x000 | 26 |
| 18Dh | EECON2 | EEPROM Control Register 2 (not a physical register) | | | | | | | | ---- ---- | 25 |
| 18Eh | — | Reserved; maintain clear | | | | | | | | 0000 0000 | — |
| 18Fh | — | Reserved; maintain clear | | | | | | | | 0000 0000 | — |

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

**Note 1:** These registers can be addressed from any bank.

**2:** The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8>, whose contents are transferred to the upper byte of the program counter.

**3:** Pin 5 is an input only; the state of the TRISA5 bit has no effect and will always read '1'.



Downloaded from Arrow.com.

### 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 $\overline{TO}$ and $\overline{PD}$ bits are not writable. Therefore, the result of an instruction with the Status register as destination may be different than intended.

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

It is recommended, therefore, that only `BCF`, `BSF`, `SWAPF` and `MOVWF` instructions are used to alter the Status register because these instructions do not affect the Z, C or DC bits from the Status register. For other instructions not affecting any status bits, see **Section 13.0 "Instruction Set Summary"**.

| **Note:** | The C and DC bits operate as a $\overline{borrow}$ and digit $\overline{borrow}$ bit, respectively, in subtraction. See the `SUBLW` and `SUBWF` instructions for examples. |
|---|---|

**REGISTER 2-1: STATUS: STATUS REGISTER (ADDRESS 03h, 83h, 103h, 183h)**

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

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

bit 6-5 **RP<1:0>:** Register Bank Select bits (used for direct addressing)
`11` = Bank 3 (180h-1FFh)
`10` = Bank 2 (100h-17Fh)
`01` = Bank 1 (80h-FFh)
`00` = Bank 0 (00h-7Fh)
Each bank is 128 bytes.

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

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

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

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

bit 0 **C:** Carry/$\overline{borrow}$ bit (`ADDWF`, `ADDLW`, `SUBLW` and `SUBWF` instructions)**(1,2)**
`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:** For $\overline{borrow}$, the polarity is reversed. A subtraction is executed by adding the two's complement of the second operand.

**2:** 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 |


© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

### 2.2.2.2 OPTION_REG Register

The OPTION_REG register is a readable and writable register that 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.

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

**REGISTER 2-2: OPTION_REG: OPTION REGISTER (ADDRESS 81h, 181h)**

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

bit 7 **$\overline{\text{RBPU}}$**: PORTB Pull-up Enable bit
1 = PORTB pull-ups are disabled
0 = PORTB pull-ups are enabled by individual port latch values

bit 6 **INTEDG:** Interrupt Edge Select bit
1 = Interrupt on rising edge of RB0/INT pin
0 = Interrupt on falling edge of RB0/INT pin

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

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

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

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

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

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



Downloaded from Arrow.com.

### 2.2.2.3 INTCON Register

The INTCON register is a readable and writable register that 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 Interrupt Enable bit, GIE (INTCON<7>). User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt. |
|---|---|

**REGISTER 2-3: INTCON: INTERRUPT CONTROL REGISTER (ADDRESS 0Bh, 8Bh, 10Bh, 18Bh)**

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-x |
|---|---|---|---|---|---|---|---|
| GIE | PEIE | TMR0IE | INTE | RBIE | TMR0IF | INTF | RBIF |
| bit 7 | | | | | | | bit 0 |

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

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

bit 5 **TMR0IE:** 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 **TMR0IF:** 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
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.
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 |


© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

### 2.2.2.4 PIE1 Register

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

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

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

| U-0 | R/W-0 | U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | ADIE | — | — | SSPIE | CCP1IE | TMR2IE | TMR1IE |
| bit 7 | | | | | | | bit 0 |

bit 7 **Unimplemented:** Read as '0'

bit 6 **ADIE:** A/D Converter Interrupt Enable bit
1 = Enables the A/D converter interrupt
0 = Disables the A/D converter interrupt

bit 5-4 **Unimplemented:** Read as '0'

bit 3 **SSPIE:** Synchronous Serial Port Interrupt Enable bit
1 = Enables the SSP interrupt
0 = Disables the SSP interrupt

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 |



Downloaded from Arrow.com.

### 2.2.2.5 PIR1 Register

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

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

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

| U-0 | R/W-0 | U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | ADIF | — | — | SSPIF | CCP1IF | TMR2IF | TMR1IF |
| bit 7 | | | | | | | bit 0 |

bit 7 **Unimplemented:** Read as '0'

bit 6 **ADIF:** A/D Converter Interrupt Flag bit
1 = An A/D conversion completed
0 = The A/D conversion is not complete

bit 5-4 **Unimplemented:** Read as '0'

bit 3 **SSPIF:** Synchronous Serial Port (SSP) Interrupt Flag bit
1 = The SSP interrupt condition has occurred and must be cleared in software before returning from the Interrupt Service Routine. The conditions that will set this bit are a transmission/ reception has taken place.
0 = No SSP interrupt condition has occurred

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 |


© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

### 2.2.2.6 PIE2 Register

The PIE2 register contains the individual enable bit for the EEPROM write operation interrupt.

**REGISTER 2-6: PIE2: PERIPHERAL INTERRUPT ENABLE REGISTER 2 (ADDRESS 8Dh)**

| U-0 | U-0 | U-0 | R/W-0 | U-0 | U-0 | U-0 | U-0 |
|---|---|---|---|---|---|---|---|
| — | — | — | EEIE | — | — | — | — |
| bit 7 | | | | | | | bit 0 |

bit 7-5 **Unimplemented:** Read as '0'

bit 4 **EEIE**: EEPROM Write Operation Interrupt Enable bit
1 = Enable EE write interrupt
0 = Disable EE write interrupt

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

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

### 2.2.2.7 PIR2 Register

The PIR2 register contains the flag bit for the EEPROM write operation interrupt.

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

**REGISTER 2-7: PIR2: PERIPHERAL INTERRUPT REQUEST (FLAG) REGISTER 2 (ADDRESS 0Dh)**

| U-0 | U-0 | U-0 | R/W-0 | U-0 | U-0 | U-0 | U-0 |
|---|---|---|---|---|---|---|---|
| — | — | — | EEIF | — | — | — | — |
| bit 7 | | | | | | | bit 0 |

bit 7-5 **Unimplemented:** Read as '0'

bit 4 **EEIF**: EEPROM Write Operation Interrupt Enable bit
1 = Enable EE write interrupt
0 = Disable EE write interrupt

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

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



Downloaded from Arrow.com.

### 2.2.2.8 PCON Register

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

The Power Control (PCON) register contains a flag bit to allow differentiation between a Power-on Reset (POR), a Brown-out Reset, an external MCLR Reset and WDT Reset.

| | |
|---|---|
| **Note:** | $\overline{BOR}$ is unknown on Power-on Reset. It must then be set by the user and checked on subsequent Resets to see if $\overline{BOR}$ is clear, indicating a brown-out has occurred. The $\overline{BOR}$ status bit is a 'don't care' and is not necessarily predictable if the brown-out circuit is disabled (by clearing the BOREN bit in the Configuration word). |

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

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

bit 7-2 **Unimplemented:** Read as '0'

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

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

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

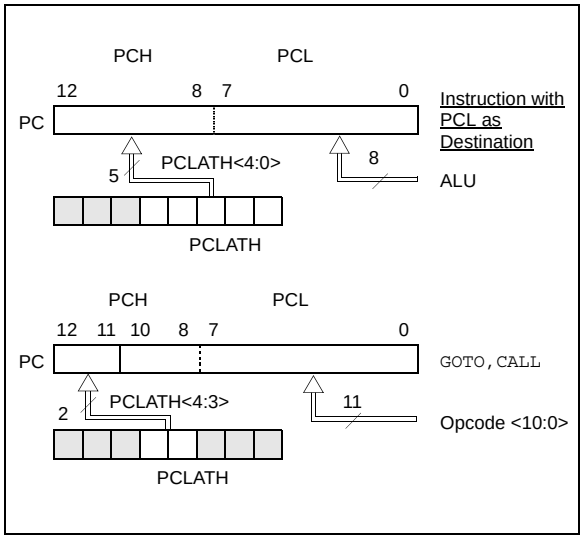

© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

## 2.3 PCL and PCLATH

The Program Counter (PC) is 13 bits wide. The low byte comes from the PCL register, which is a readable and writable register. The upper bits (PC<12:8>) are not readable but are indirectly writable through the PCLATH register. On any Reset, the upper bits of the PC will be cleared. Figure 2-5 shows the two situations for the loading of the PC. The upper example in the figure shows how the PC is loaded on a write to PCL (PCLATH<4:0> → PCH). The lower example in the figure shows how the PC is loaded during a CALL or GOTO instruction (PCLATH<4:3> → PCH).

**FIGURE 2-5: LOADING OF PC IN DIFFERENT SITUATIONS**




### 2.3.1 COMPUTED GOTO

A computed GOTO is accomplished by adding an offset to the program counter (ADDWF PCL). When doing a table read using a computed GOTO method, care should be exercised if the table location crosses a PCL memory boundary (each 256-byte block). Refer to the application note *AN556, "Implementing a Table Read"* (DS00556).

### 2.3.2 STACK

The PIC16F818/819 family has 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 affected by a PUSH or POP operation.

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

| | |
|---|---|
| **Note 1:** | There are no status bits to indicate stack overflow or stack underflow conditions. |
| **2:** | There are no instructions/mnemonics called PUSH or POP. These are actions that occur from the execution of the CALL, RETURN, RETLW and RETFIE instructions or the vectoring to an interrupt address. |

## 2.4 Indirect Addressing: INDF and FSR Registers

The INDF register is not a physical register. Addressing 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 INDF 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.

**EXAMPLE 2-2: HOW TO CLEAR RAM USING INDIRECT ADDRESSING**

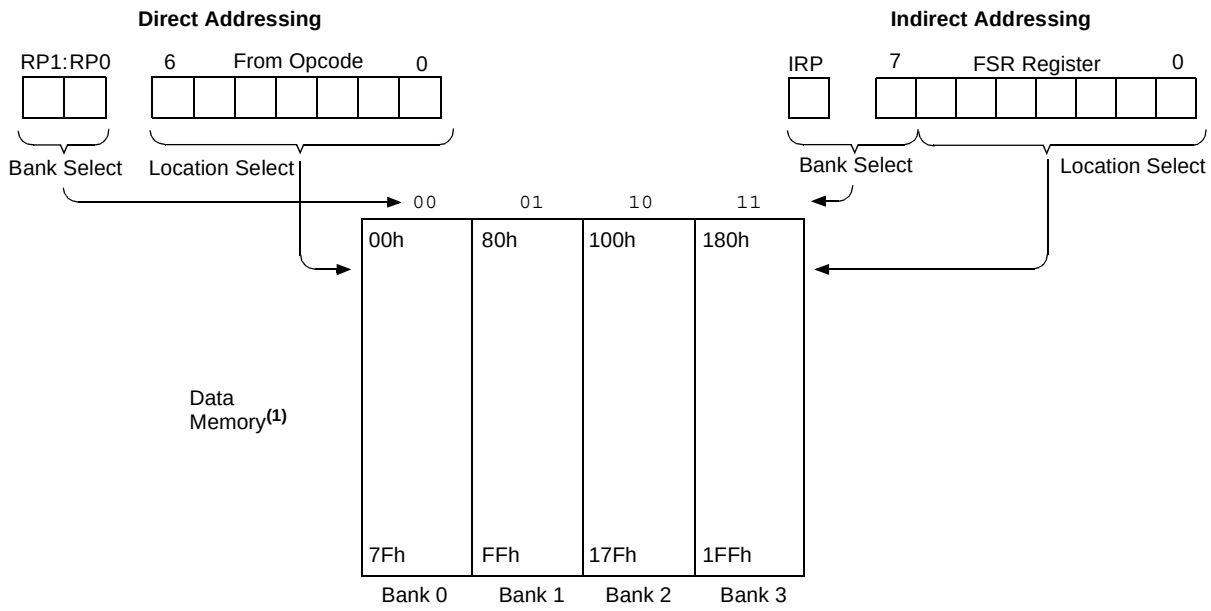
```
        MOVLW  0x20    ;initialize pointer
        MOVWF  FSR     ;to RAM
NEXT    CLRF   INDF    ;clear INDF register
        INCF   FSR     ;inc pointer
        BTFSS  FSR, 4  ;all done?
        GOTO   NEXT    ;NO, clear next
CONTINUE
        :              ;YES, continue
```

An effective 9-bit address is obtained by concatenating the 8-bit FSR register and the IRP bit (Status<7>) as shown in Figure 2-6.



Downloaded from Arrow.com.

**FIGURE 2-6: DIRECT/INDIRECT ADDRESSING**




**Note 1:** For register file map detail, see Figure 2-3 or Figure 2-4.


© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

## 3.0 DATA EEPROM AND FLASH PROGRAM MEMORY

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

- EECON1
- EECON2
- EEDATA
- EEDATH
- EEADR
- EEADRH

This section focuses on reading and writing data EEPROM and Flash program memory during normal operation. Refer to the appropriate device programming specification document for serial programming information.

When interfacing the data memory block, EEDATA holds the 8-bit data for read/write and EEADR holds the address of the EEPROM location being accessed. These devices have 128 or 256 bytes of data EEPROM, with an address range from 00h to 0FFh. Addresses from 80h to FFh are unimplemented on the PIC16F818 device and will read 00h. When writing to unimplemented locations, the charge pump will be turned off.

When interfacing the program memory block, the EEDATA and EEDATH registers form a two-byte word that holds the 14-bit data for read/write and the EEADR and EEADRH registers form a two-byte word that holds the 13-bit address of the EEPROM location being accessed. These devices have 1K or 2K words of program Flash, with an address range from 0000h to 03FFh for the PIC16F818 and 0000h to 07FFh for the PIC16F819. Addresses above the range of the respective device will wraparound to the beginning of program memory.

The EEPROM data memory allows single byte read and write. The Flash program memory allows single-word reads and four-word block writes. Program memory writes must first start with a 32-word block erase, then write in 4-word blocks. A byte write in data EEPROM memory automatically erases the location and writes the new data (erase before write).

The write time is controlled by an on-chip timer. The write/erase voltages are generated by an on-chip charge pump, rated to operate over the voltage range of the device for byte or word operations.

When the device is code-protected, the CPU may continue to read and write the data EEPROM memory. Depending on the settings of the write-protect bits, the device may or may not be able to write certain blocks of the program memory; however, reads of the program memory are allowed. When code-protected, the device programmer can no longer access data or program memory; this does NOT inhibit internal reads or writes.

### 3.1 EEADR and EEADRH

The EEADRH:EEADR register pair can address up to a maximum of 256 bytes of data EEPROM or up to a maximum of 8K words of program EEPROM. When selecting a data address value, only the LSB of the address is written to the EEADR register. When selecting a program address value, the MSB of the address is written to the EEADRH register and the LSB is written to the EEADR register.

If the device contains less memory than the full address reach of the address register pair, the Most Significant bits of the registers are not implemented. For example, if the device has 128 bytes of data EEPROM, the Most Significant bit of EEADR is not implemented on access to data EEPROM.

### 3.2 EECON1 and EECON2 Registers

EECON1 is the control register for memory accesses.

Control bit, EEPGD, determines if the access will be a program or data memory access. When clear, as it is when Reset, any subsequent operations will operate on the data memory. When set, any subsequent operations will operate on the program memory.

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

The WREN bit, when set, will allow a write or erase operation. On power-up, the WREN bit is clear. The WRERR bit is set when a write (or erase) operation is interrupted by a MCLR or a WDT Time-out Reset during normal operation. In these situations, following Reset, the user can check the WRERR bit and rewrite the location. The data and address will be unchanged in the EEDATA and EEADR registers.

Interrupt flag bit, EEIF in the PIR2 register, is set when the write is complete. It must be cleared in software.

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



Downloaded from Arrow.com.

**REGISTER 3-1: EECON1: EEPROM ACCESS CONTROL REGISTER 1 (ADDRESS 18Ch)**

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

bit 7 **EEPGD**: Program/Data EEPROM Select bit
1 = Accesses program memory
0 = Accesses data memory
Reads '0' after a POR; this bit cannot be changed while a write operation is in progress.

bit 6-5 **Unimplemented:** Read as '0'

bit 4 **FREE:** EEPROM Forced Row Erase bit
1 = Erase the program memory row addressed by EEADRH:EEADR on the next WR command
0 = Perform write-only

bit 3 **WRERR:** EEPROM Error Flag bit
1 = A write operation is prematurely terminated (any $\overline{\text{MCLR}}$ or any WDT Reset during normal operation)
0 = The write operation completed

bit 2 **WREN:** EEPROM Write Enable bit
1 = Allows write cycles
0 = Inhibits write to the EEPROM

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

bit 0 **RD:** Read Control bit
1 = Initiates an EEPROM read, RD is cleared in hardware. The RD bit can only be set (not cleared) in software.
0 = Does not initiate an EEPROM read

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


© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

## 3.3 Reading Data EEPROM Memory

To read a data memory location, the user must write the address to the EEADR register, clear the EEPGD control bit (EECON1<7>) and then set control bit, RD (EECON1<0>). The data is available in the very next cycle in the EEDATA register; therefore, it can be read in the next instruction (see Example 3-1). EEDATA will hold this value until another read or until it is written to by the user (during a write operation).

The steps to reading the EEPROM data memory are:

1. Write the address to EEADR. Make sure that the address is not larger than the memory size of the device.
2. Clear the EEPGD bit to point to EEPROM data memory.
3. Set the RD bit to start the read operation.
4. Read the data from the EEDATA register.

**EXAMPLE 3-1: DATA EEPROM READ**

```
BANKSEL EEADR          ; Select Bank of EEADR
MOVF    ADDR, W        ;
MOVWF   EEADR          ; Data Memory Address
                       ; to read
BANKSEL EECON1         ; Select Bank of EECON1
BCF     EECON1, EEPGD  ; Point to Data memory
BSF     EECON1, RD     ; EE Read
BANKSEL EEDATA         ; Select Bank of EEDATA
MOVF    EEDATA, W      ; W = EEDATA
```

## 3.4 Writing to Data EEPROM Memory

To write an EEPROM data location, the user must first write the address to the EEADR register and the data to the EEDATA register. Then, the user must follow a specific write sequence to initiate the write for each byte.

The write will not initiate if the write sequence is not exactly followed (write 55h to EECON2, write AAh to EECON2, then set WR bit) for each byte. We strongly recommend that interrupts be disabled during this code segment (see Example 3-2).

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

After a write sequence has been initiated, clearing the WREN bit will not affect this write cycle. The WR bit will be inhibited from being set unless the WREN bit is set. At the completion of the write cycle, the WR bit is cleared in hardware and the EE Write Complete Interrupt Flag bit (EEIF) is set. The user can either enable this interrupt or poll this bit. EEIF must be cleared by software.

The steps to write to EEPROM data memory are:

1. If step 10 is not implemented, check the WR bit to see if a write is in progress.
2. Write the address to EEADR. Make sure that the address is not larger than the memory size of the device.
3. Write the 8-bit data value to be programmed in the EEDATA register.
4. Clear the EEPGD bit to point to EEPROM data memory.
5. Set the WREN bit to enable program operations.
6. Disable interrupts (if enabled).
7. Execute the special five instruction sequence:
   - Write 55h to EECON2 in two steps (first to W, then to EECON2)
   - Write AAh to EECON2 in two steps (first to W, then to EECON2)
   - Set the WR bit
8. Enable interrupts (if using interrupts).
9. Clear the WREN bit to disable program operations.
10. At the completion of the write cycle, the WR bit is cleared and the EEIF interrupt flag bit is set (EEIF must be cleared by firmware). If step 1 is not implemented, then firmware should check for EEIF to be set, or WR to be clear, to indicate the end of the program cycle.

**EXAMPLE 3-2: DATA EEPROM WRITE**

```
        BANKSEL EECON1         ; Select Bank of
                               ; EECON1
        BTFSC   EECON1, WR     ; Wait for write
        GOTO    $-1            ; to complete
        BANKSEL EEADR          ; Select Bank of
                               ; EEADR
        MOVF    ADDR, W        ;
        MOVWF   EEADR          ; Data Memory
                               ; Address to write
        MOVF    VALUE, W       ;
        MOVWF   EEDATA         ; Data Memory Value
                               ; to write
        BANKSEL EECON1         ; Select Bank of
                               ; EECON1
        BCF     EECON1, EEPGD  ; Point to DATA
                               ; memory
        BSF     EECON1, WREN   ; Enable writes

        BCF     INTCON, GIE    ; Disable INTs.
        MOVLW   55h            ;
Required
Sequence
        MOVWF   EECON2         ; Write 55h
        MOVLW   AAh            ;
        MOVWF   EECON2         ; Write AAh
        BSF     EECON1, WR     ; Set WR bit to
                               ; begin write
        BSF     INTCON, GIE    ; Enable INTs.
        BCF     EECON1, WREN   ; Disable writes
```



Downloaded from Arrow.com.

## 3.5 Reading Flash Program Memory

To read a program memory location, the user must write two bytes of the address to the EEADR and EEADRH registers, set the EEPGD control bit (EECON1<7>) and then set control bit, RD (EECON1<0>). Once the read control bit is set, the program memory Flash controller will use the second instruction cycle to read the data. This causes the second instruction immediately following the "BSF EECON1, RD" instruction to be ignored. The data is available in the very next cycle in the EEDATA and EEDATH registers; therefore, it can be read as two bytes in the following instructions. EEDATA and EEDATH registers will hold this value until another read or until it is written to by the user (during a write operation).

**EXAMPLE 3-3: FLASH PROGRAM READ**

```
BANKSEL EEADRH        ; Select Bank of EEADRH
MOVF    ADDRH, W      ;
MOVWF   EEADRH        ; MS Byte of Program
                      ; Address to read
MOVF    ADDRL, W      ;
MOVWF   EEADR         ; LS Byte of Program
                      ; Address to read
BANKSEL EECON1        ; Select Bank of EECON1
BSF     EECON1, EEPGD ; Point to PROGRAM
                      ; memory
BSF     EECON1, RD    ; EE Read
                      ;
NOP                   ; Any instructions
                      ; here are ignored as
NOP                   ; program memory is
                      ; read in second cycle
                      ; after BSF EECON1,RD
BANKSEL EEDATA        ; Select Bank of EEDATA
MOVF    EEDATA, W     ; DATAL = EEDATA
MOVWF   DATAL         ;
MOVF    EEDATH, W     ; DATAH = EEDATH
MOVWF   DATAH         ;
```

## 3.6 Erasing Flash Program Memory

The minimum erase block is 32 words. Only through the use of an external programmer, or through ICSP control, can larger blocks of program memory be bulk erased. Word erase in the Flash array is not supported.

When initiating an erase sequence from the micro-controller itself, a block of 32 words of program memory is erased. The Most Significant 11 bits of the EEADRH:EEADR point to the block being erased. EEADR< 4:0> are ignored.

The EECON1 register commands the erase operation. The EEPGD bit must be set to point to the Flash program memory. The WREN bit must be set to enable write operations. The FREE bit is set to select an erase operation.

For protection, the write initiate sequence for EECON2 must be used.

After the "BSF EECON1, WR" instruction, the processor requires two cycles to set up the erase operation. The user must place two NOP instructions after the WR bit is set. The processor will halt internal operations for the typical 2 ms, only during the cycle in which the erase takes place. This is not Sleep mode, as the clocks and peripherals will continue to run. After the erase cycle, the processor will resume operation with the third instruction after the EECON1 write instruction.

### 3.6.1 FLASH PROGRAM MEMORY ERASE SEQUENCE

The sequence of events for erasing a block of internal program memory location is:

1. Load EEADRH:EEADR with address of row being erased.
2. Set EEPGD bit to point to program memory; set WREN bit to enable writes and set FREE bit to enable the erase.
3. Disable interrupts.
4. Write 55h to EECON2.
5. Write AAh to EECON2.
6. Set the WR bit. This will begin the row erase cycle.
7. The CPU will stall for duration of the erase.


© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

**EXAMPLE 3-4: ERASING A FLASH PROGRAM MEMORY ROW**

```
          BANKSEL  EEADRH          ; Select Bank of EEADRH
          MOVF     ADDRH, W        ;
          MOVWF    EEADRH          ; MS Byte of Program Address to Erase
          MOVF     ADDRL, W        ;
          MOVWF    EEADR           ; LS Byte of Program Address to Erase
ERASE_ROW
          BANKSEL  EECON1          ; Select Bank of EECON1
          BSF      EECON1, EEPGD   ; Point to PROGRAM memory
          BSF      EECON1, WREN    ; Enable Write to memory
          BSF      EECON1, FREE    ; Enable Row Erase operation
;
          BCF      INTCON, GIE     ; Disable interrupts (if using)
          MOVLW    55h             ;
          MOVWF    EECON2          ; Write 55h
          MOVLW    AAh             ;
          MOVWF    EECON2          ; Write AAh
          BSF      EECON1, WR      ; Start Erase (CPU stall)
          NOP                      ; Any instructions here are ignored as processor
                                   ; halts to begin Erase sequence
          NOP                      ; processor will stop here and wait for Erase complete
                                   ; after Erase processor continues with 3rd instruction
          BCF      EECON1, FREE    ; Disable Row Erase operation
          BCF      EECON1, WREN    ; Disable writes
          BSF      INTCON, GIE     ; Enable interrupts (if using)
```



Downloaded from Arrow.com.

## 3.7 Writing to Flash Program Memory

Flash program memory may only be written to if the destination address is in a segment of memory that is not write-protected, as defined in bits WRT1:WRT0 of the device Configuration Word (Register 12-1). Flash program memory must be written in four-word blocks. A block consists of four words with sequential addresses, with a lower boundary defined by an address, where EEADR<1:0> = 00. At the same time, all block writes to program memory are done as write-only operations. The program memory must first be erased. The write operation is edge-aligned and cannot occur across boundaries.

To write to the program memory, the data must first be loaded into the buffer registers. There are four 14-bit buffer registers and they are addressed by the low 2 bits of EEADR.

The following sequence of events illustrate how to perform a write to program memory:

- Set the EEPGD and WREN bits in the EECON1 register
- Clear the FREE bit in EECON1
- Write address to EEADRH:EEADR
- Write data to EEDATH:EEDATA
- Write 55 to EECON2
- Write AA to EECON2
- Set WR bit in EECON 1

The user must follow the same specific sequence to initiate the write for each word in the program block by writing each program word in sequence (00, 01, 10, 11).

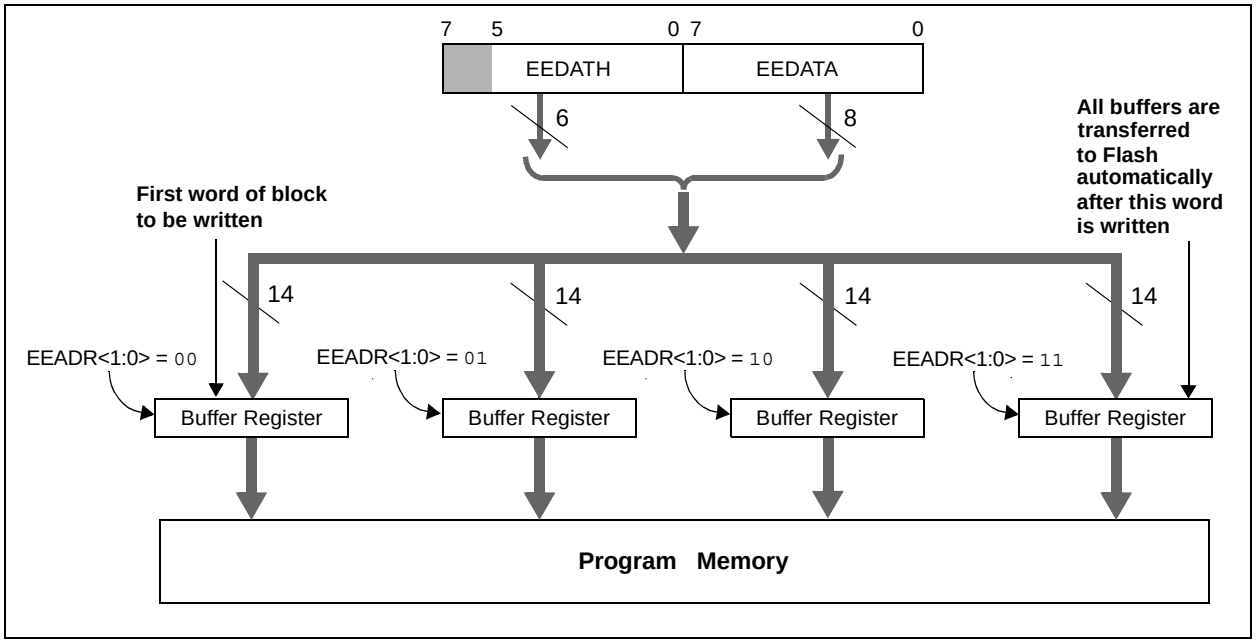
There are 4 buffer register words and all four locations **MUST** be written to with correct data.

After the "BSF EECON1, WR" instruction, if EEADR ≠ xxxxxx11, then a short write will occur. This short write-only transfers the data to the buffer register. The WR bit will be cleared in hardware after one cycle.

After the "BSF EECON1, WR" instruction, if EEADR = xxxxxx11, then a long write will occur. This will simultaneously transfer the data from EEDATH:EEDATA to the buffer registers and begin the write of all four words. The processor will execute the next instruction and then ignore the subsequent instruction. The user should place NOP instructions into the second words. The processor will then halt internal operations for typically 2 msec in which the write takes place. This is not a Sleep mode, as the clocks and peripherals will continue to run. After the write cycle, the processor will resume operation with the 3rd instruction after the EECON1 write instruction.

After each long write, the 4 buffer registers will be reset to 3FFF.

**FIGURE 3-1: BLOCK WRITES TO FLASH PROGRAM MEMORY**





© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

An example of the complete four-word write sequence is shown in Example 3-5. The initial address is loaded into the EEADRH:EEADR register pair; the four words of data are loaded using indirect addressing, assuming that a row erase sequence has already been performed.

**EXAMPLE 3-5: WRITING TO FLASH PROGRAM MEMORY**

```
; This write routine assumes the following:

; 1. The 32 words in the erase block have already been erased.
; 2. A valid starting address (the least significant bits = '00') is loaded into EEADRH:EEADR
; 3. This example is starting at 0x100, this is an application dependent setting.
; 4. The 8 bytes (4 words) of data are loaded, starting at an address in RAM called ARRAY.
; 5. This is an example only, location of data to program is application dependent.
; 6. word_block is located in data memory.

        BANKSEL  EECON1             ;prepare for WRITE procedure
        BSF      EECON1, EEPGD      ;point to program memory
        BSF      EECON1, WREN       ;allow write cycles
        BCF      EECON1, FREE       ;perform write only

        BANKSEL  word_block
        MOVLW    .4
        MOVWF    word_block         ;prepare for 4 words to be written

        BANKSEL  EEADRH             ;Start writing at 0x100
        MOVLW    0x01
        MOVWF    EEADRH             ;load HIGH address
        MOVLW    0x00
        MOVWF    EEADR              ;load LOW address
        BANKSEL  ARRAY
        MOVLW    ARRAY              ;initialize FSR to start of data
        MOVWF    FSR
LOOP
        BANKSEL  EEDATA
        MOVF     INDF, W            ;indirectly load EEDATA
        MOVWF    EEDATA
        INCF     FSR, F             ;increment data pointer
        MOVF     INDF, W            ;indirectly load EEDATH
        MOVWF    EEDATH
        INCF     FSR, F             ;increment data pointer

        BANKSEL  EECON1
        MOVLW    0x55               ;required sequence
        MOVWF    EECON2
        MOVLW    0xAA
        MOVWF    EECON2
        BSF      EECON1, WR         ;set WR bit to begin write
        NOP                         ;instructions here are ignored as processor
        NOP

        BANKSEL  EEADR
        INCF     EEADR, f           ;load next word address
        BANKSEL  word_block
        DECFSZ   word_block, f      ;have 4 words been written?
        GOTO     loop               ;NO, continue with writing

        BANKSEL  EECON1
        BCF      EECON1, WREN       ;YES, 4 words complete, disable writes
        BSF      INTCON, GIE        ;enable interrupts
```

Required Sequence



Downloaded from Arrow.com.

## 3.8 Protection Against Spurious Write

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

The write initiate sequence and the WREN bit together help prevent an accidental write during brown-out, power glitch or software malfunction.

## 3.9 Operation During Code-Protect

When the data EEPROM is code-protected, the microcontroller can read and write to the EEPROM normally. However, all external access to the EEPROM is disabled. External write access to the program memory is also disabled.

When program memory is code-protected, the microcontroller can read and write to program memory normally as well as execute instructions. Writes by the device may be selectively inhibited to regions of the memory depending on the setting of bits, WRT1:WRT0, of the Configuration Word (see **Section 12.1 "Configuration Bits"** for additional information). External access to the memory is also disabled.

**TABLE 3-1: REGISTERS/BITS ASSOCIATED WITH DATA EEPROM AND FLASH PROGRAM MEMORIES**

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on Power-on Reset | Value on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 10Ch | EEDATA | EEPROM/Flash Data Register Low Byte | | | | | | | | xxxx xxxx | uuuu uuuu |
| 10Dh | EEADR | EEPROM/Flash Address Register Low Byte | | | | | | | | xxxx xxxx | uuuu uuuu |
| 10Eh | EEDATH | — | — | EEPROM/Flash Data Register High Byte | | | | | | --xx xxxx | --uu uuuu |
| 10Fh | EEADRH | — | — | — | — | — | EEPROM/Flash Address Register High Byte | | | ---- -xxx | ---- -uuu |
| 18Ch | EECON1 | EEPGD | — | — | FREE | WRERR | WREN | WR | RD | x--x x000 | x--x q000 |
| 18Dh | EECON2 | EEPROM Control Register 2 (not a physical register) | | | | | | | | ---- ---- | ---- ---- |
| 0Dh | PIR2 | — | — | — | EEIF | — | — | — | — | ---0 ---- | ---0 ---- |
| 8Dh | PIE2 | — | — | — | EEIE | — | — | — | — | ---0 ---- | ---0 ---- |

**Legend:** x = unknown, u = unchanged, - = unimplemented, read as '0', q = value depends upon condition.
Shaded cells are not used by data EEPROM or Flash program memory.


© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

## 4.0 OSCILLATOR CONFIGURATIONS

### 4.1 Oscillator Types

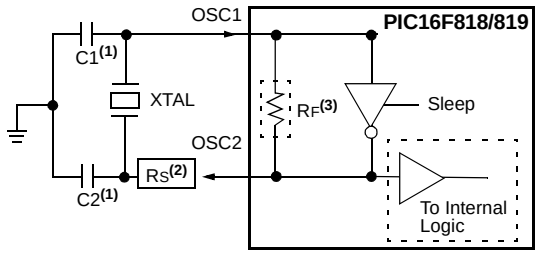
The PIC16F818/819 can be operated in eight different oscillator modes. The user can program three configuration bits (FOSC2:FOSC0) to select one of these eight modes (modes 5-8 are new PIC16 oscillator configurations):

1. LP Low-Power Crystal
2. XT Crystal/Resonator
3. HS High-Speed Crystal/Resonator
4. RC External Resistor/Capacitor with FOSC/4 output on RA6
5. RCIO External Resistor/Capacitor with I/O on RA6
6. INTIO1 Internal Oscillator with FOSC/4 output on RA6 and I/O on RA7
7. INTIO2 Internal Oscillator with I/O on RA6 and RA7
8. ECIO External Clock with I/O on RA6

### 4.2 Crystal Oscillator/Ceramic Resonators

In XT, LP or HS modes, a crystal or ceramic resonator is connected to the OSC1/CLKI and OSC2/CLKO pins to establish oscillation (see Figure 4-1 and Figure 4-2). The PIC16F818/819 oscillator design requires the use of a parallel cut crystal. Use of a series cut crystal may give a frequency out of the crystal manufacturer's specifications.

**FIGURE 4-1: CRYSTAL OPERATION (HS, XT OR LP OSC CONFIGURATION)**




**Note 1:** See Table 4-1 for typical values of C1 and C2.

**2:** A series resistor (RS) may be required for AT strip cut crystals.

**3:** RF varies with the crystal chosen (typically between 2 MΩ to 10 MΩ).

**TABLE 4-1: CAPACITOR SELECTION FOR CRYSTAL OSCILLATOR (FOR DESIGN GUIDANCE ONLY)**

| Osc Type | Crystal Freq | Typical Capacitor Values Tested: | |
|---|---|---|---|
| | | C1 | C2 |
| LP | 32 kHz | 33 pF | 33 pF |
| | 200 kHz | 15 pF | 15 pF |
| XT | 200 kHz | 56 pF | 56 pF |
| | 1 MHz | 15 pF | 15 pF |
| | 4 MHz | 15 pF | 15 pF |
| HS | 4 MHz | 15 pF | 15 pF |
| | 8 MHz | 15 pF | 15 pF |
| | 20 MHz | 15 pF | 15 pF |

**Capacitor values are for design guidance only.**

These capacitors were tested with the crystals listed below for basic start-up and operation. These values were not optimized.

Different capacitor values may be required to produce acceptable oscillator operation. The user should test the performance of the oscillator over the expected VDD and temperature range for the application.

See the notes following this table for additional information.

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

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

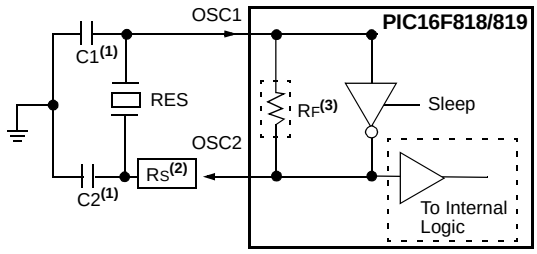
**3:** RS may be required in HS mode, as well as XT mode, to avoid overdriving crystals with low drive level specification.

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



Downloaded from Arrow.com.

**FIGURE 4-2: CERAMIC RESONATOR OPERATION (HS OR XT OSC CONFIGURATION)**




**Note 1:** See Table 4-2 for typical values of C1 and C2.

**2:** A series resistor (RS) may be required.

**3:** RF varies with the resonator chosen (typically between 2 MΩ to 10 MΩ).

**TABLE 4-2: CERAMIC RESONATORS (FOR DESIGN GUIDANCE ONLY)**

| Typical Capacitor Values Used: | | | |
|---|---|---|---|
| **Mode** | **Freq** | **OSC1** | **OSC2** |
| XT | 455 kHz<br>2.0 MHz<br>4.0 MHz | 56 pF<br>47 pF<br>33 pF | 56 pF<br>47 pF<br>33 pF |
| HS | 8.0 MHz<br>16.0 MHz | 27 pF<br>22 pF | 27 pF<br>22 pF |

**Capacitor values are for design guidance only.**

These capacitors were tested with the resonators listed below for basic start-up and operation. These values were not optimized.

Different capacitor values may be required to produce acceptable oscillator operation. The user should test the performance of the oscillator over the expected VDD and temperature range for the application.

See the notes following this table for additional information.

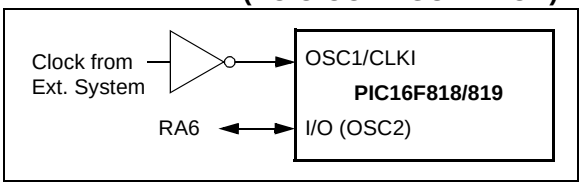
**Note:** When using resonators with frequencies above 3.5 MHz, the use of HS mode rather than XT mode is recommended. HS mode may be used at any VDD for which the controller is rated. If HS is selected, it is possible that the gain of the oscillator will overdrive the resonator. Therefore, a series resistor should be placed between the OSC2 pin and the resonator. As a good starting point, the recommended value of RS is 330Ω.

## 4.3 External Clock Input

The ECIO Oscillator mode requires an external clock source to be connected to the OSC1 pin. There is no oscillator start-up time required after a Power-on Reset or after an exit from Sleep mode.

In the ECIO Oscillator mode, the OSC2 pin becomes an additional general purpose I/O pin. The I/O pin becomes bit 6 of PORTA (RA6). Figure 4-3 shows the pin connections for the ECIO Oscillator mode.

**FIGURE 4-3: EXTERNAL CLOCK INPUT OPERATION (ECIO CONFIGURATION)**





© 2004 Microchip Technology Inc.

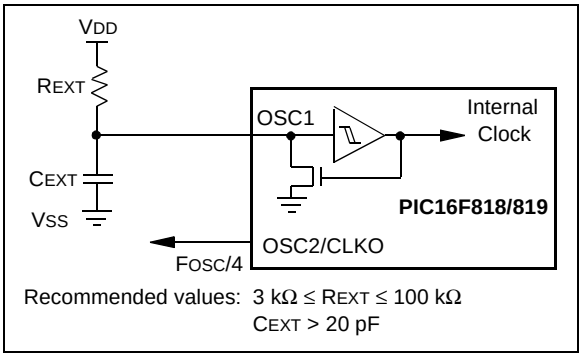
Downloaded from Arrow.com.

## 4.4 RC Oscillator

For timing insensitive applications, the “RC” and “RCIO” device options offer additional cost savings. The RC oscillator frequency is a function of the supply voltage, the resistor (REXT) and capacitor (CEXT) values and the operating temperature. In addition to this, the oscillator frequency will vary from unit to unit due to normal manufacturing variation. Furthermore, the difference in lead frame capacitance between package types will also affect the oscillation frequency, especially for low CEXT values. The user also needs to take into account variation due to tolerance of external R and C components used. Figure 4-4 shows how the R/C combination is connected.

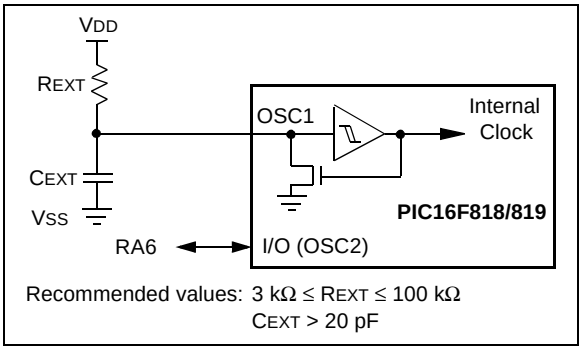
In the RC Oscillator mode, the oscillator frequency divided by 4 is available on the OSC2 pin. This signal may be used for test purposes or to synchronize other logic.

**FIGURE 4-4: RC OSCILLATOR MODE**




The RCIO Oscillator mode (Figure 4-5) functions like the RC mode except that the OSC2 pin becomes an additional general purpose I/O pin. The I/O pin becomes bit 6 of PORTA (RA6).

**FIGURE 4-5: RCIO OSCILLATOR MODE**




## 4.5 Internal Oscillator Block

The PIC16F818/819 devices include an internal oscillator block which generates two different clock signals; either can be used as the system’s clock source. This can eliminate the need for external oscillator circuits on the OSC1 and/or OSC2 pins.

The main output (INTOSC) is an 8 MHz clock source which can be used to directly drive the system clock. It also drives the INTOSC postscaler which can provide a range of clock frequencies from 125 kHz to 4 MHz.

The other clock source is the internal RC oscillator (INTRC) which provides a 31.25 kHz (32 µs nominal period) output. The INTRC oscillator is enabled by selecting the INTRC as the system clock source or when any of the following are enabled:

- Power-up Timer
- Watchdog Timer

These features are discussed in greater detail in **Section 12.0 “Special Features of the CPU”**.

The clock source frequency (INTOSC direct, INTRC direct or INTOSC postscaler) is selected by configuring the IRCF bits of the OSCCON register (Register 4-2).

| **Note:** | Throughout this data sheet, when referring *specifically* to a generic clock source, the term “INTRC” may also be used to refer to the clock modes using the internal oscillator block. This is regardless of whether the actual frequency used is INTOSC (8 MHz), the INTOSC postscaler or INTRC (31.25 kHz). |
|---|---|

### 4.5.1 INTRC MODES

Using the internal oscillator as the clock source can eliminate the need for up to two external oscillator pins, which can then be used for digital I/O. Two distinct configurations are available:

- In INTIO1 mode, the OSC2 pin outputs FOSC/4 while OSC1 functions as RA7 for digital input and output.
- In INTIO2 mode, OSC1 functions as RA7 and OSC2 functions as RA6, both for digital input and output.



Downloaded from Arrow.com.

### 4.5.2 OSCTUNE REGISTER

The internal oscillator's output has been calibrated at the factory but can be adjusted in the application. This is done by writing to the OSCTUNE register (Register 4-1). The tuning sensitivity is constant throughout the tuning range. The OSCTUNE register has a tuning range of ±12.5%.

When the OSCTUNE register is modified, the INTOSC and INTRC frequencies will begin shifting to the new frequency. The INTRC clock will reach the new frequency within 8 clock cycles (approximately 8 * 32 µs = 256 µs); the INTOSC clock will stabilize within 1 ms. Code execution continues during this shift. There is no indication that the shift has occurred. Operation of features that depend on the 31.25 kHz INTRC clock source frequency, such as the WDT, Fail-Safe Clock Monitor and peripherals, will also be affected by the change in frequency.

**REGISTER 4-1: OSCTUNE: OSCILLATOR TUNING REGISTER (ADDRESS 90h)**

| U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | — | TUN5 | TUN4 | TUN3 | TUN2 | TUN1 | TUN0 |
| bit 7 | | | | | | | bit 0 |

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

bit 5-0 **TUN<5:0>:** Frequency Tuning bits

011111 = Maximum frequency
011110 =
•
•
•
000001 =
000000 = Center frequency. Oscillator module is running at the calibrated frequency.
111111 =
•
•
•
100000 = Minimum frequency

| **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 |


© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

### 4.5.3 OSCILLATOR CONTROL REGISTER

The OSCCON register (Register 4-2) controls several aspects of the system clock's operation.

The Internal Oscillator Select bits, IRCF2:IRCF0, select the frequency output of the internal oscillator block that is used to drive the system clock. The choices are the INTRC source (31.25 kHz), the INTOSC source (8 MHz) or one of the six frequencies derived from the INTOSC postscaler (125 kHz to 4 MHz). Changing the configuration of these bits has an immediate change on the multiplexor's frequency output.

### 4.5.4 MODIFYING THE IRCF BITS

The IRCF bits can be modified at any time regardless of which clock source is currently being used as the system clock. The internal oscillator allows users to change the frequency during run time. This is achieved by modifying the IRCF bits in the OSCCON register. The sequence of events that occur after the IRCF bits are modified is dependent upon the initial value of the IRCF bits before they are modified. If the INTRC (31.25 kHz, IRCF<2:0> = 000) is running and the IRCF bits are modified to any other value than '000', a 4 ms (approx.) clock switch delay is turned on. Code execution continues at a higher than expected frequency while the new frequency stabilizes. Time sensitive code should wait for the IOFS bit in the OSCCON register to become set before continuing. This bit can be monitored to ensure that the frequency is stable before using the system clock in time critical applications.

If the IRCF bits are modified while the internal oscillator is running at any other frequency than INTRC (31.25 kHz, IRCF<2:0> ≠ 000), there is no need for a 4 ms (approx.) clock switch delay. The new INTOSC frequency will be stable immediately after the **eight** falling edges. The IOFS bit will remain set after clock switching occurs.

| | |
|---|---|
| **Note:** | Caution must be taken when modifying the IRCF bits using BCF or BSF instructions. It is possible to modify the IRCF bits to a frequency that may be out of the VDD specification range; for example, VDD = 2.0V and IRCF = 111 (8 MHz). |

### 4.5.5 CLOCK TRANSITION SEQUENCE WHEN THE IRCF BITS ARE MODIFIED

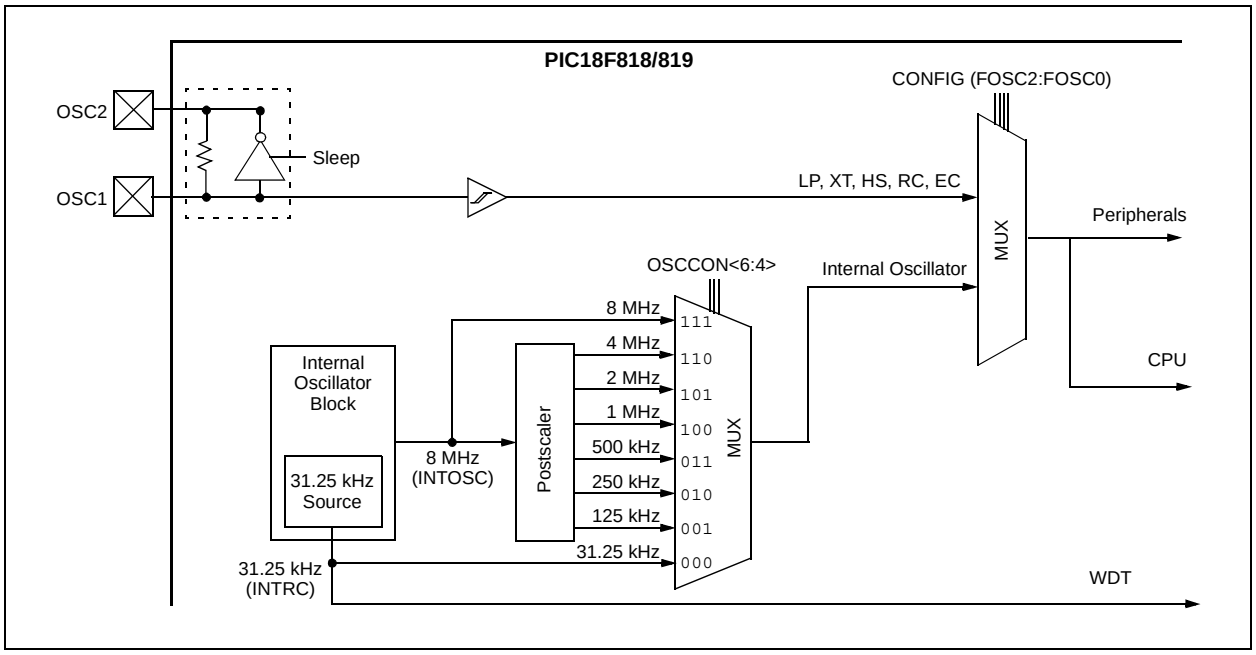
Following are three different sequences for switching the internal RC oscillator frequency.

- Clock before switch: 31.25 kHz (IRCF<2:0> = 000)
  1. IRCF bits are modified to an INTOSC/INTOSC postscaler frequency.
  2. The clock switching circuitry waits for a falling edge of the current clock, at which point CLKO is held low.
  3. The clock switching circuitry then waits for eight falling edges of requested clock, after which it switches CLKO to this new clock source.
  4. The IOFS bit is clear to indicate that the clock is unstable and a 4 ms (approx.) delay is started. Time dependent code should wait for IOFS to become set.
  5. Switchover is complete.
- Clock before switch: One of INTOSC/INTOSC postscaler (IRCF<2:0> ≠ 000)
  1. IRCF bits are modified to INTRC (IRCF<2:0> = 000).
  2. The clock switching circuitry waits for a falling edge of the current clock, at which point CLKO is held low.
  3. The clock switching circuitry then waits for eight falling edges of requested clock, after which it switches CLKO to this new clock source.
  4. Oscillator switchover is complete.
- Clock before switch: One of INTOSC/INTOSC postscaler (IRCF<2:0> ≠ 000)
  1. IRCF bits are modified to a different INTOSC/ INTOSC postscaler frequency.
  2. The clock switching circuitry waits for a falling edge of the current clock, at which point CLKO is held low.
  3. The clock switching circuitry then waits for eight falling edges of requested clock, after which it switches CLKO to this new clock source.
  4. The IOFS bit is set.
  5. Oscillator switchover is complete.



Downloaded from Arrow.com.

**FIGURE 4-6: PIC16F818/819 CLOCK DIAGRAM**




**REGISTER 4-2: OSCCON: OSCILLATOR CONTROL REGISTER (ADDRESS 8Fh)**

| U-0 | R/W-0 | R/W-0 | R/W-0 | U-0 | R-0 | U-0 | U-0 |
|---|---|---|---|---|---|---|---|
| — | IRCF2 | IRCF1 | IRCF0 | — | IOFS | — | — |
| bit 7 | | | | | | | bit 0 |

bit 7 **Unimplemented:** Read as '0'

bit 6-4 **IRCF2:IRCF0:** Internal Oscillator Frequency Select bits
111 = 8 MHz (8 MHz source drives clock directly)
110 = 4 MHz
101 = 2 MHz
100 = 1 MHz
011 = 500 kHz
010 = 250 kHz
001 = 125 kHz
000 = 31.25 kHz (INTRC source drives clock directly)

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

bit 2 **IOFS:** INTOSC Frequency Stable bit
1 = Frequency is stable
0 = Frequency is not stable

bit 1-0 **Unimplemented:** Read as '0'

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


© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

## 5.0 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 MCU Family Reference Manual"* (DS33023).

## 5.1 PORTA and the TRISA Register

PORTA is an 8-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).

| **Note:** | On a Power-on Reset, the pins PORTA<4:0> are configured as analog inputs and read as '0'. |
|---|---|

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

Pin RA4 is multiplexed with the Timer0 module clock input and with an analog input to become the RA4/AN4/T0CKI pin. The RA4/AN4/T0CKI pin is a Schmitt Trigger input and full CMOS output driver.

Pin RA5 is multiplexed with the Master Clear module input. The RA5/MCLR/VPP pin is a Schmitt Trigger input.

Pin RA6 is multiplexed with the oscillator module input and external oscillator output. Pin RA7 is multiplexed with the oscillator module input and external oscillator input. Pin RA6/OSC2/CLKO and pin RA7/OSC1/CLKI are Schmitt Trigger inputs and full CMOS output drivers.

Pins RA<1:0> are multiplexed with analog inputs. Pins RA<3:2> are multiplexed with analog inputs and VREF inputs. Pins RA<3:0> have TTL inputs and full CMOS output drivers.

**EXAMPLE 5-1: INITIALIZING PORTA**

```
BANKSEL  PORTA     ; select bank of PORTA
CLRF     PORTA     ; Initialize PORTA by
                   ; clearing output
                   ; data latches
BANKSEL  ADCON1    ; Select Bank of ADCON1
MOVLW    0x06      ; Configure all pins
MOVWF    ADCON1    ; as digital inputs
MOVLW    0xFF      ; Value used to
                   ; initialize data
                   ; direction
MOVWF    TRISA     ; Set RA<7:0> as inputs
```

**TABLE 5-1: PORTA FUNCTIONS**

| Name | Bit# | Buffer | Function |
|---|---|---|---|
| RA0/AN0 | bit 0 | TTL | Input/output or analog input. |
| RA1/AN1 | bit 1 | TTL | Input/output or analog input. |
| RA2/AN2/VREF- | bit 2 | TTL | Input/output, analog input or VREF-. |
| RA3/AN3/VREF+ | bit 3 | TTL | Input/output, analog input or VREF+. |
| RA4/AN4/T0CKI | bit 4 | ST | Input/output, analog input or external clock input for Timer0. |
| RA5/MCLR/VPP | bit 5 | ST | Input, Master Clear (Reset) or programming voltage input. |
| RA6/OSC2/CLKO | bit 6 | ST | Input/output, connects to crystal or resonator, oscillator output or 1/4 the frequency of OSC1 and denotes the instruction cycle in RC mode. |
| RA7/OSC1/CLKI | bit 7 | ST/CMOS(1) | Input/output, connects to crystal or resonator or oscillator input. |

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

**Note 1:** This buffer is a Schmitt Trigger input when configured in RC Oscillator mode and a CMOS input otherwise.

**TABLE 5-2: SUMMARY OF REGISTERS ASSOCIATED WITH PORTA**

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR, BOR | Value on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 05h | PORTA | RA7 | RA6 | RA5 | RA4 | RA3 | RA2 | RA1 | RA0 | xxx0 0000 | uuu0 0000 |
| 85h | TRISA | TRISA7 | TRISA6 | TRISA5(1) | PORTA Data Direction Register | | | | | 1111 1111 | 1111 1111 |
| 9Fh | ADCON1 | ADFM | ADCS2 | — | — | PCFG3 | PCFG2 | PCFG1 | PCFG0 | 00-- 0000 | 00-- 0000 |

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

**Note 1:** Pin 5 is an input only; the state of the TRISA5 bit has no effect and will always read '1'.



Downloaded from Arrow.com.

**FIGURE 5-1: BLOCK DIAGRAM OF RA0/AN0:RA1/AN1 PINS**

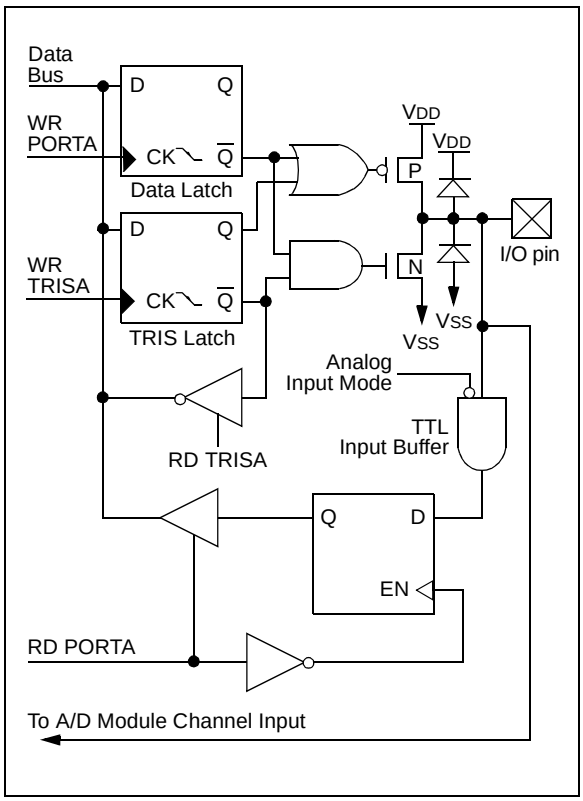



**FIGURE 5-2: BLOCK DIAGRAM OF RA3/AN3/VREF+ PIN**

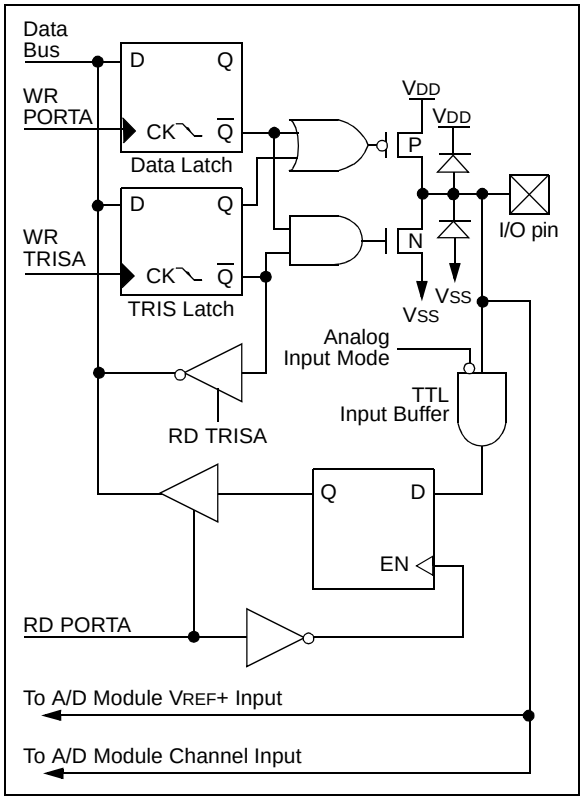



**FIGURE 5-3: BLOCK DIAGRAM OF RA2/AN2/VREF- PIN**

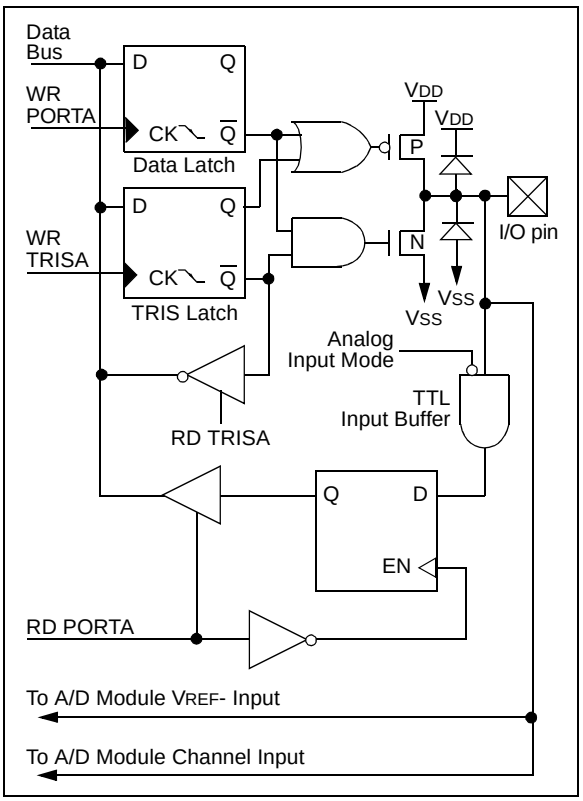
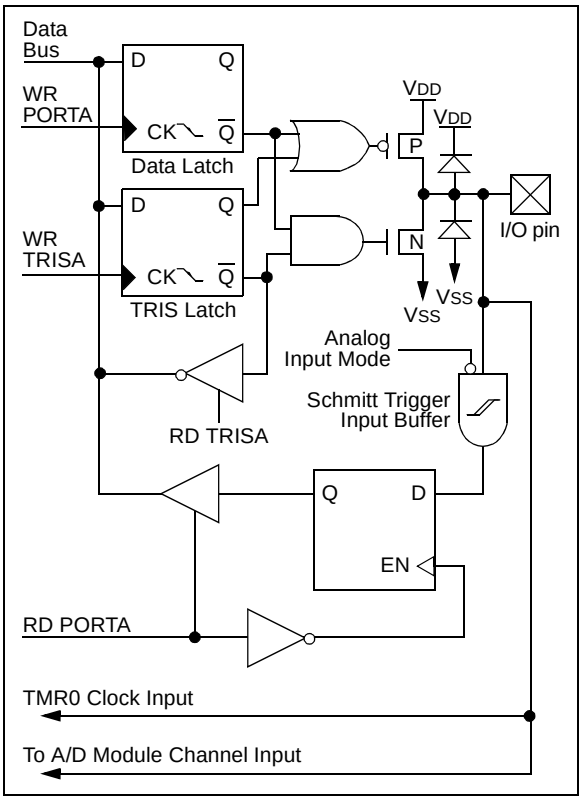



**FIGURE 5-4: BLOCK DIAGRAM OF RA4/AN4/T0CKI PIN**





© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

**FIGURE 5-5: BLOCK DIAGRAM OF RA5/$\overline{\text{MCLR}}$/VPP PIN**

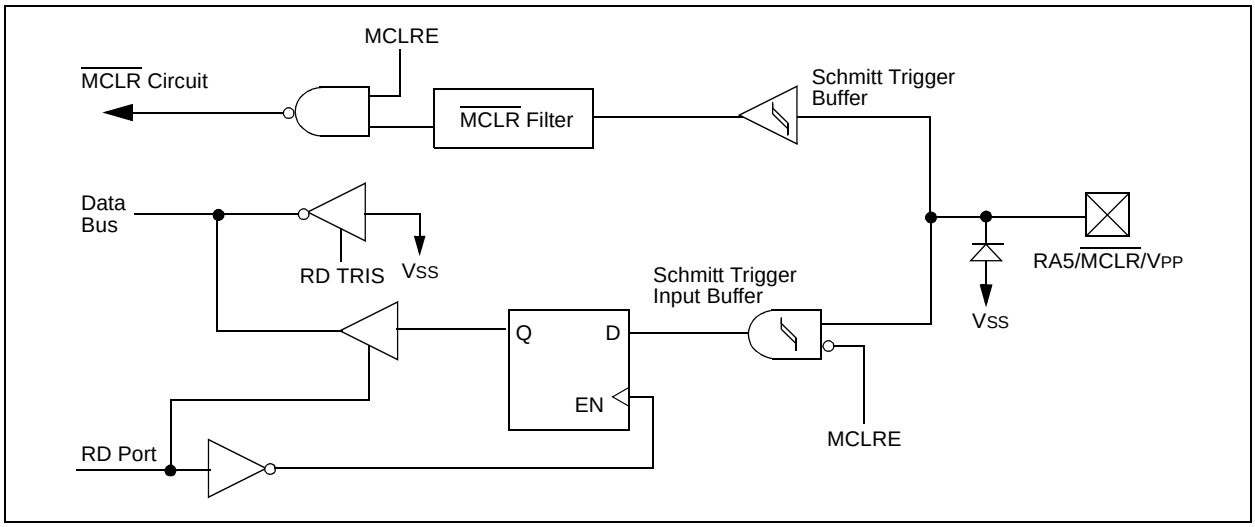
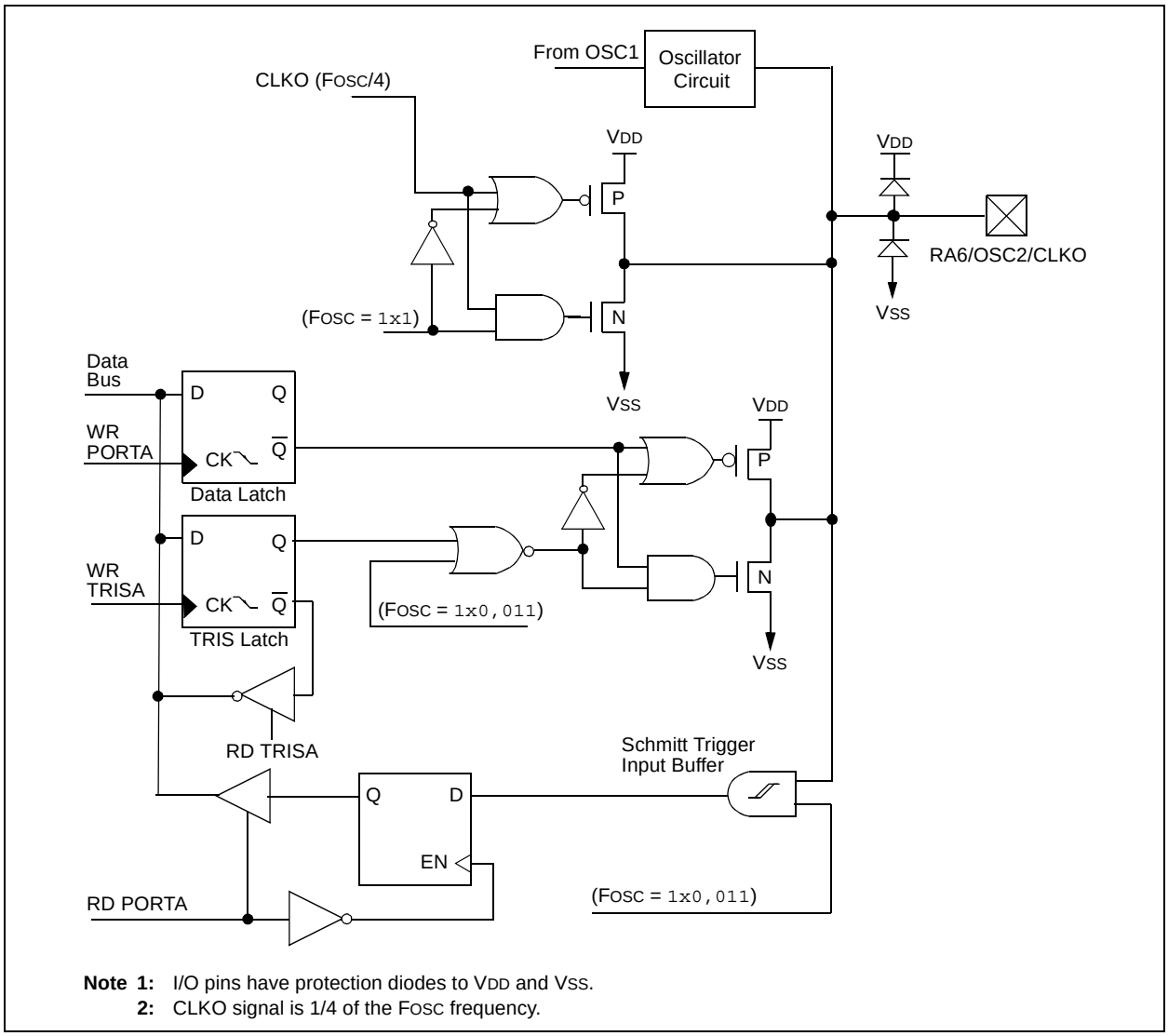



**FIGURE 5-6: BLOCK DIAGRAM OF RA6/OSC2/CLKO PIN**

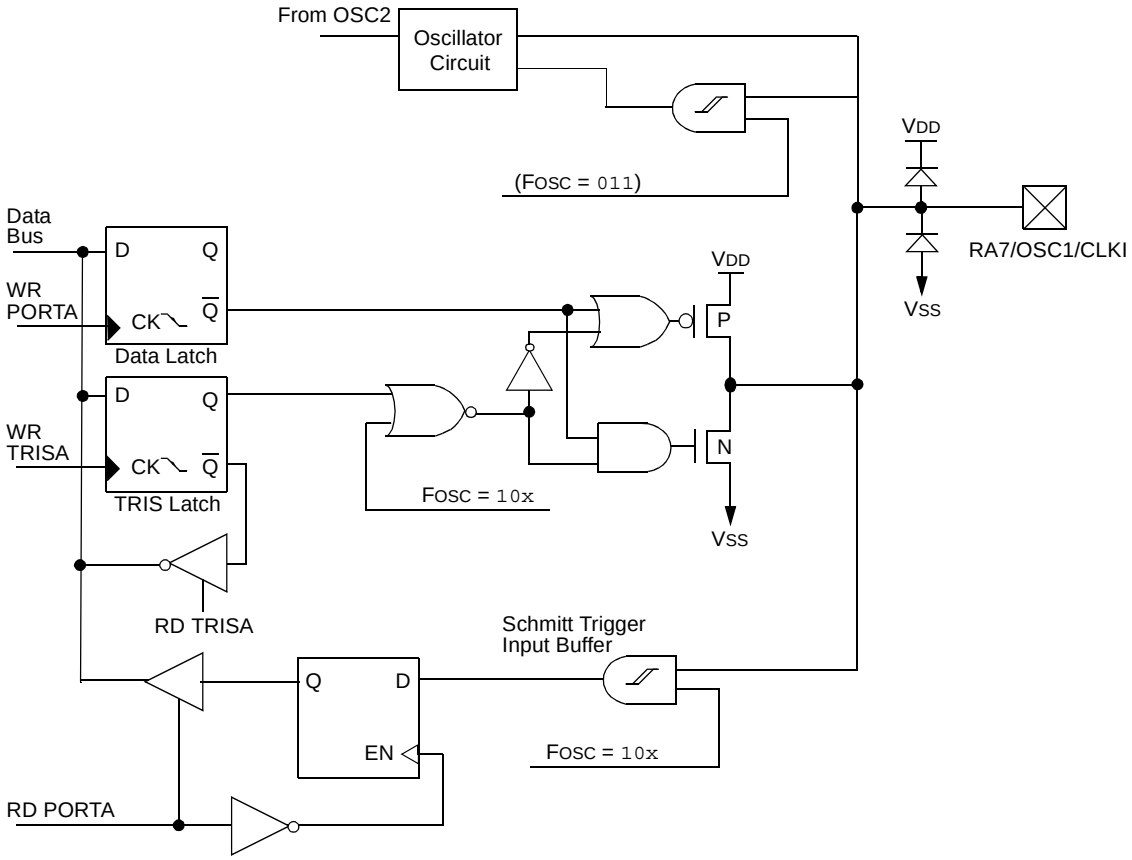



**Note 1:** I/O pins have protection diodes to VDD and VSS.
**2:** CLKO signal is 1/4 of the FOSC frequency.



Downloaded from Arrow.com.

**FIGURE 5-7: BLOCK DIAGRAM OF RA7/OSC1/CLKI PIN**




**Note 1:** I/O pins have protection diodes to VDD and VSS.


© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

## 5.2 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).

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.

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 (of RB7:RB4) are compared with the old value latched on the last read of PORTB. The "mismatch" outputs of RB7:RB4 are ORed 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:

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

RB0/INT is an external interrupt input pin and is configured using the INTEDG bit (OPTION_REG<6>).

PORTB is multiplexed with several peripheral functions (see Table 5-3). PORTB pins have Schmitt Trigger input buffers.

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 output, while other peripherals override the TRIS bit to make a pin an input. Since the TRIS bit override is in effect while the peripheral is enabled, read-modify-write instructions (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.



Downloaded from Arrow.com.

**TABLE 5-3: PORTB FUNCTIONS**

| Name | Bit# | Buffer | Function |
|---|---|---|---|
| RB0/INT | bit 0 | TTL/ST(1) | Input/output pin or external interrupt input. Internal software programmable weak pull-up. |
| RB1/SDI/SDA | bit 1 | TTL/ST(5) | Input/output pin, SPI™ data input pin or $I^2C$™ data I/O pin. Internal software programmable weak pull-up. |
| RB2/SDO/CCP1 | bit 2 | TTL/ST(4) | Input/output pin, SPI data output pin or Capture input/Compare output/PWM output pin. Internal software programmable weak pull-up. |
| RB3/CCP1/PGM(3) | bit 3 | TTL/ST(2) | Input/output pin, Capture input/Compare output/PWM output pin or programming in LVP mode. Internal software programmable weak pull-up. |
| RB4/SCK/SCL | bit 4 | TTL/ST(5) | Input/output pin or SPI and $I^2C$ clock pin (with interrupt-on-change). Internal software programmable weak pull-up. |
| RB5/$\overline{SS}$ | bit 5 | TTL | Input/output pin or SPI slave select pin (with interrupt-on-change). Internal software programmable weak pull-up. |
| RB6/T1OSO/T1CKI/ PGC | bit 6 | TTL/ST(2) | Input/output pin, Timer1 oscillator output pin, Timer1 clock input pin or serial programming clock (with interrupt-on-change). Internal software programmable weak pull-up. |
| RB7/T1OSI/PGD | bit 7 | TTL/ST(2) | Input/output pin, Timer1 oscillator input pin or serial programming data (with interrupt-on-change). Internal software programmable weak pull-up. |

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

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

**2:** This buffer is a Schmitt Trigger input when used in Serial Programming mode.

**3:** Low-Voltage ICSP™ Programming (LVP) is enabled by default which disables the RB3 I/O function. LVP must be disabled to enable RB3 as an I/O pin and allow maximum compatibility to the other 18-pin mid-range devices.

**4:** This buffer is a Schmitt Trigger input when configured for CCP or SSP mode.

**5:** This buffer is a Schmitt Trigger input when configured for SPI or $I^2C$ mode.

**TABLE 5-4: SUMMARY OF REGISTERS ASSOCIATED WITH PORTB**

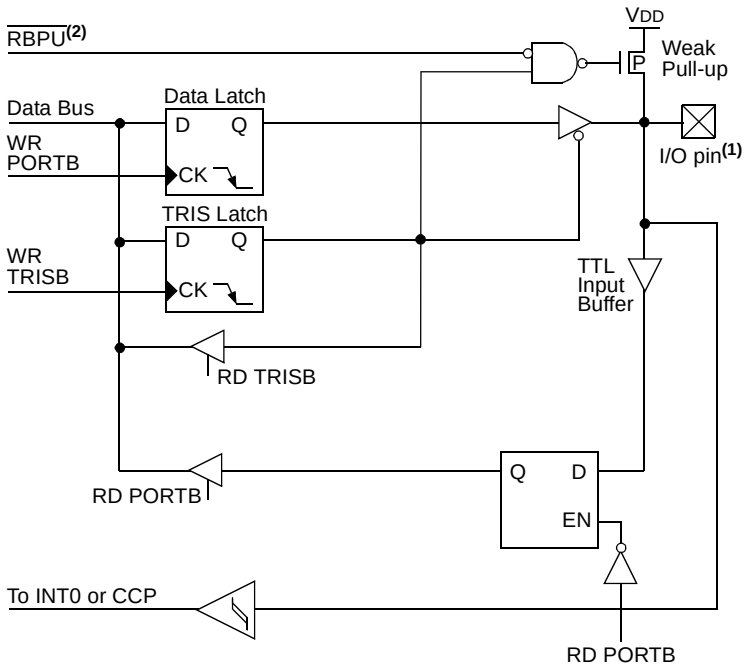
| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR, BOR | Value on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 06h, 106h | PORTB | RB7 | RB6 | RB5 | RB4 | RB3 | RB2 | RB1 | RB0 | xxxx xxxx | uuuu uuuu |
| 86h, 186h | TRISB | PORTB Data Direction Register | | | | | | | | 1111 1111 | 1111 1111 |
| 81h, 181h | OPTION_REG | $\overline{RBPU}$ | INTEDG | T0CS | T0SE | PSA | PS2 | PS1 | PS0 | 1111 1111 | 1111 1111 |

**Legend:** x = unknown, u = unchanged. Shaded cells are not used by PORTB.


© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

**FIGURE 5-8: BLOCK DIAGRAM OF RB0 PIN**

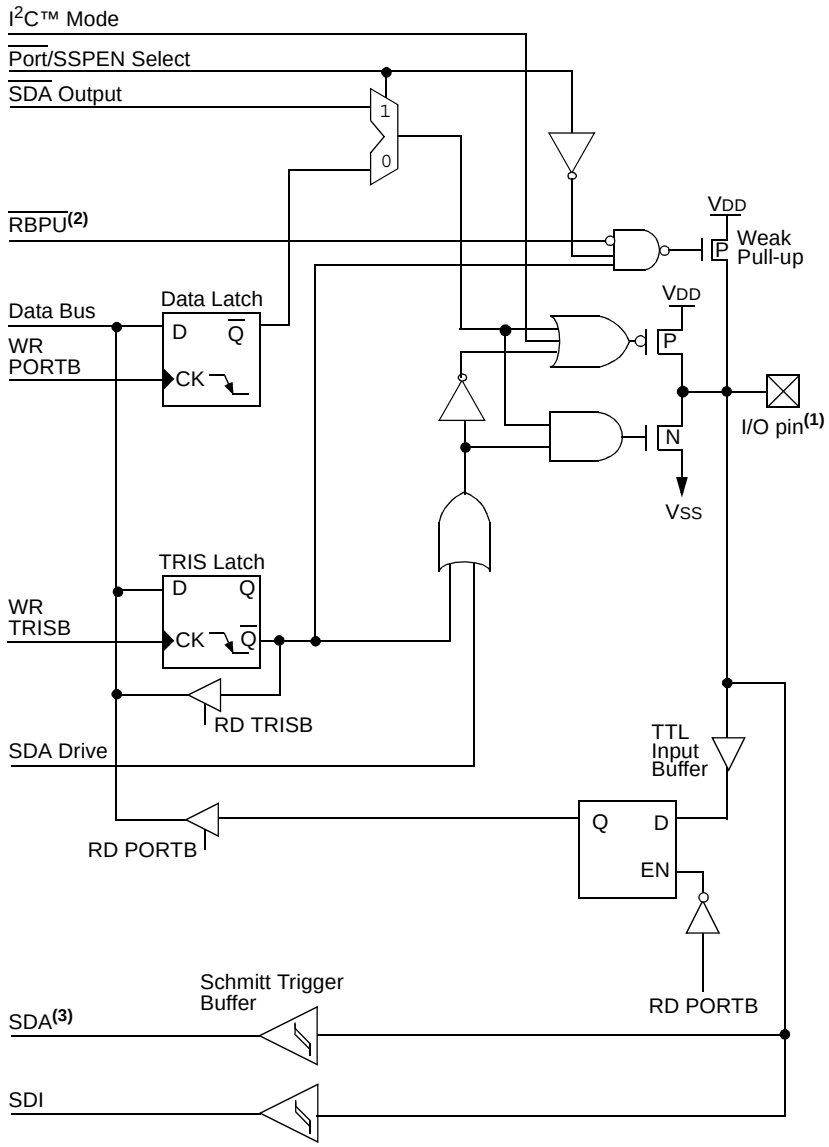



**Note 1:** I/O pins have diode protection to VDD and VSS.
**2:** To enable weak pull-ups, set the appropriate TRIS bit(s) and clear the $\overline{\text{RBPU}}$ bit.



Downloaded from Arrow.com.

**FIGURE 5-9: BLOCK DIAGRAM OF RB1 PIN**

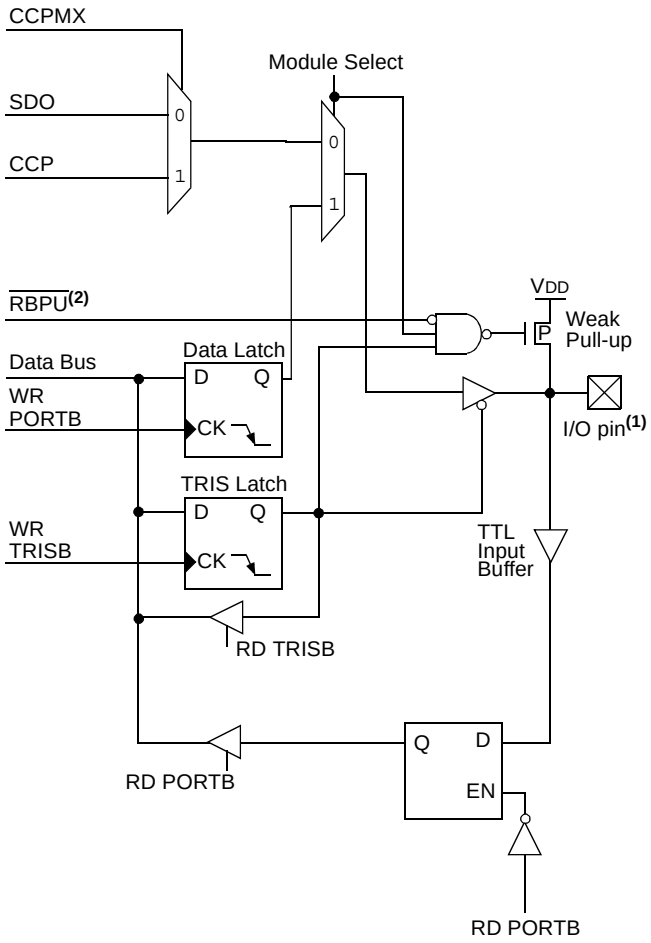



**Note 1:** I/O pins have diode protection to VDD and VSS.
**2:** To enable weak pull-ups, set the appropriate TRIS bit(s) and clear the RBPU bit.
**3:** The SDA Schmitt Trigger conforms to the I2C specification.


© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

**FIGURE 5-10: BLOCK DIAGRAM OF RB2 PIN**

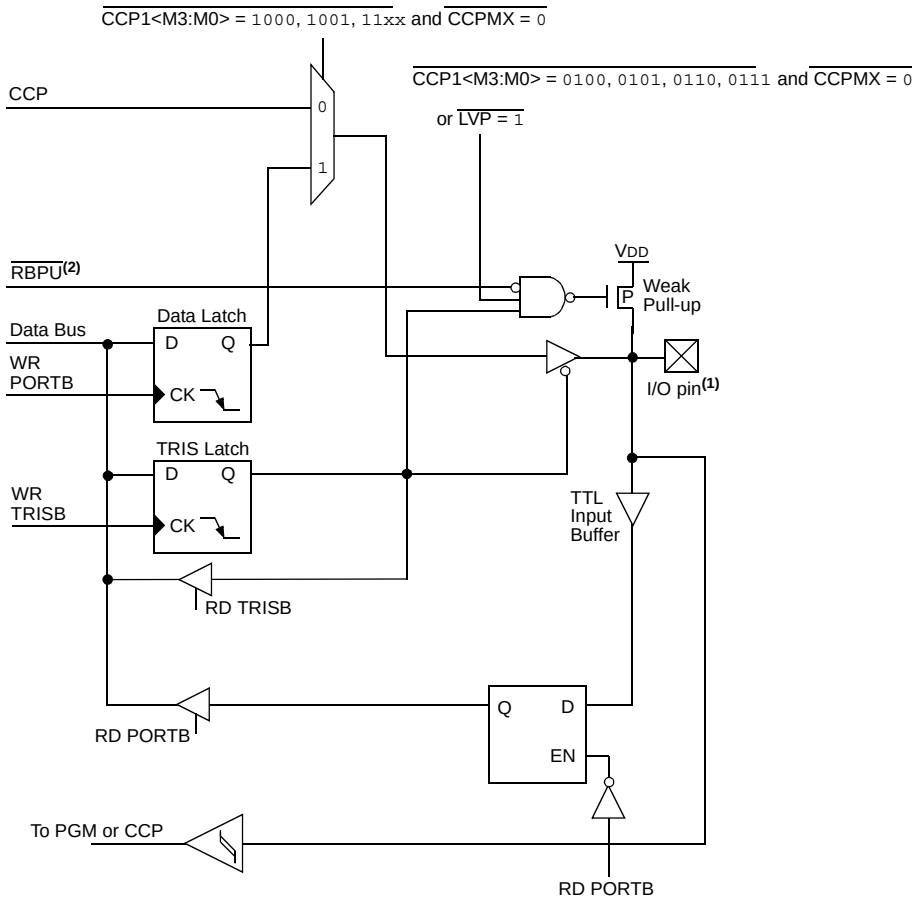



**Note 1:** I/O pins have diode protection to VDD and VSS.
**2:** To enable weak pull-ups, set the appropriate TRIS bit(s) and clear the $\overline{\text{RBPU}}$ bit.



Downloaded from Arrow.com.

**FIGURE 5-11: BLOCK DIAGRAM OF RB3 PIN**

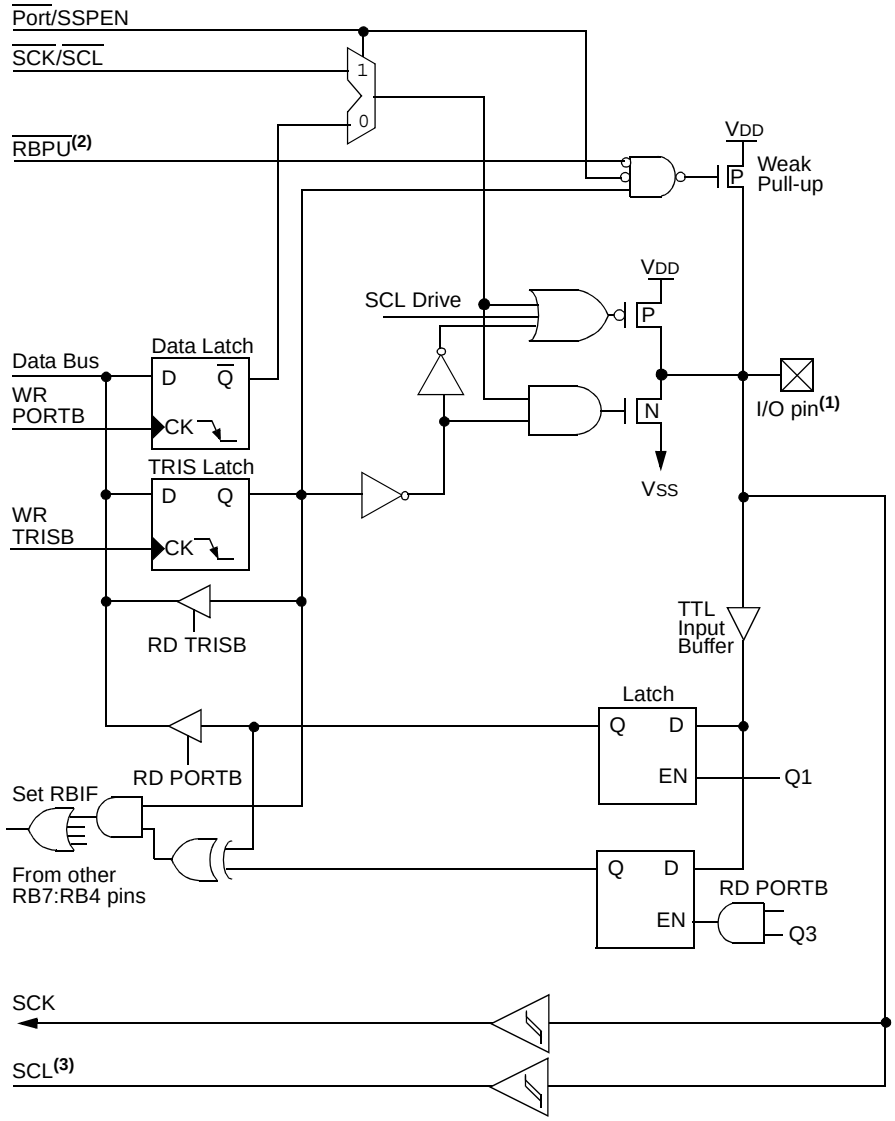



**Note 1:** I/O pins have diode protection to VDD and VSS.
**2:** To enable weak pull-ups, set the appropriate TRIS bit(s) and clear the $\overline{\text{RBPU}}$ bit.


© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

**FIGURE 5-12: BLOCK DIAGRAM OF RB4 PIN**

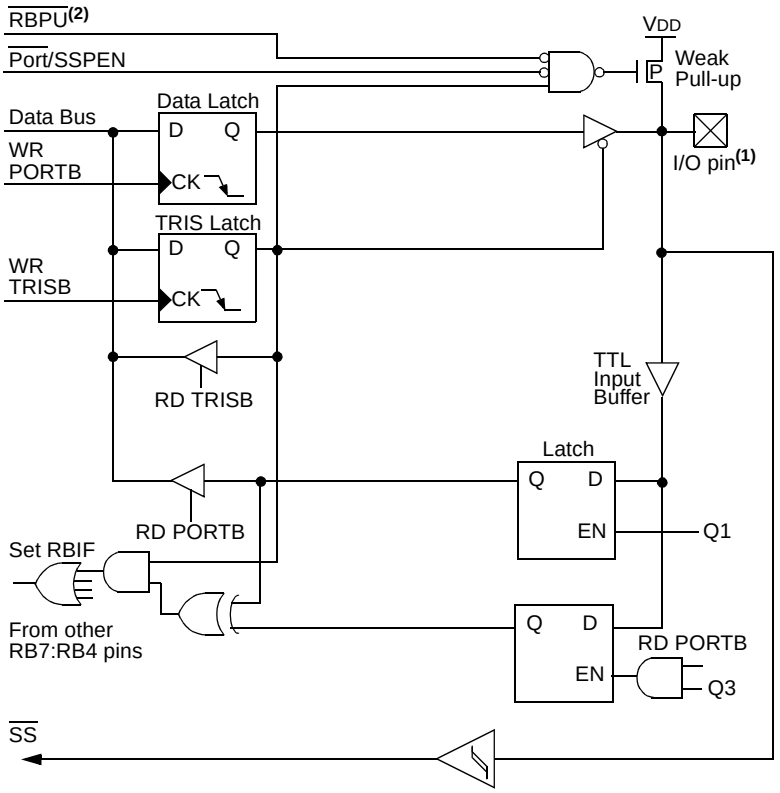



**Note 1:** I/O pins have diode protection to VDD and VSS.
**2:** To enable weak pull-ups, set the appropriate TRIS bit(s) and clear the $\overline{\text{RBPU}}$ bit.
**3:** The SCL Schmitt Trigger conforms to the I²C™ specification.



Downloaded from Arrow.com.

**FIGURE 5-13: BLOCK DIAGRAM OF RB5 PIN**

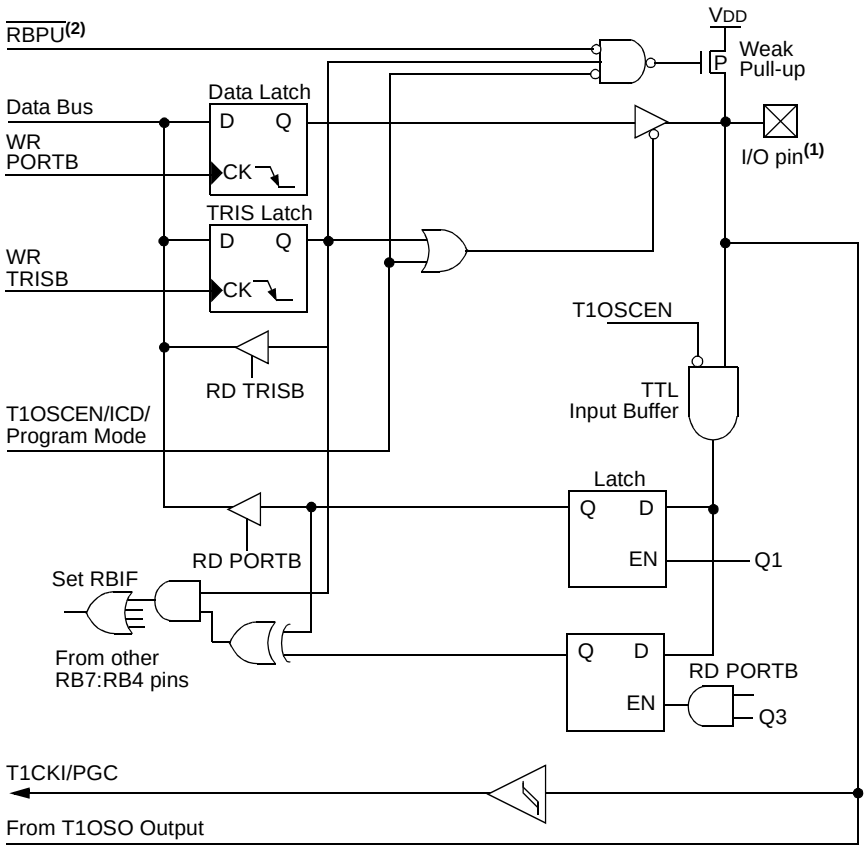



**Note 1:** I/O pins have diode protection to VDD and VSS.
**2:** To enable weak pull-ups, set the appropriate TRIS bit(s) and clear the $\overline{\text{RBPU}}$ bit.


© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

**FIGURE 5-14: BLOCK DIAGRAM OF RB6 PIN**

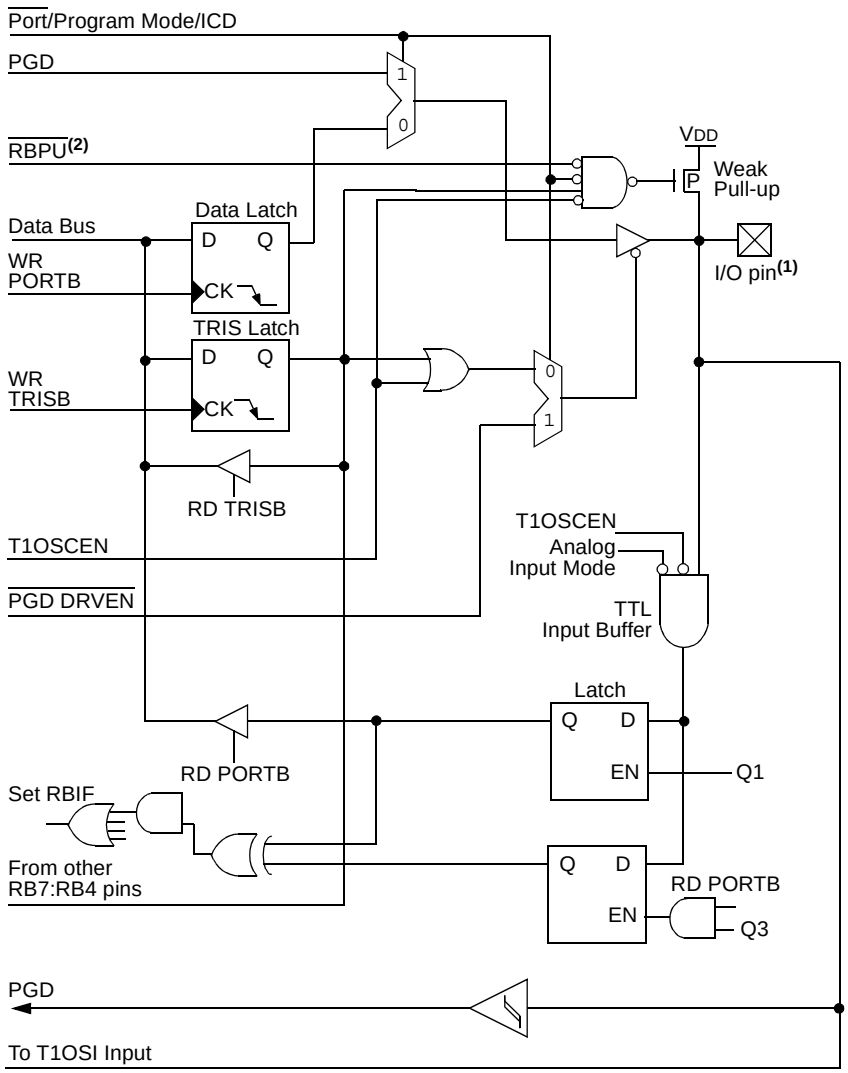



**Note 1:** I/O pins have diode protection to VDD and VSS.
**2:** To enable weak pull-ups, set the appropriate TRIS bit(s) and clear the $\overline{\text{RBPU}}$ bit.



Downloaded from Arrow.com.

**FIGURE 5-15: BLOCK DIAGRAM OF RB7 PIN**




**Note 1:** I/O pins have diode protection to VDD and VSS.
**2:** To enable weak pull-ups, set the appropriate TRIS bit(s) and clear the RBPU bit.


© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

## 6.0 TIMER0 MODULE

The Timer0 module timer/counter has the following features:

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

Additional information on the Timer0 module is available in the *"PICmicro® Mid-Range MCU Family Reference Manual"* (DS33023).

Figure 6-1 is a block diagram of the Timer0 module and the prescaler shared with the WDT.

### 6.1 Timer0 Operation

Timer0 operation is controlled through the OPTION_REG register (see Register 2-2). 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.

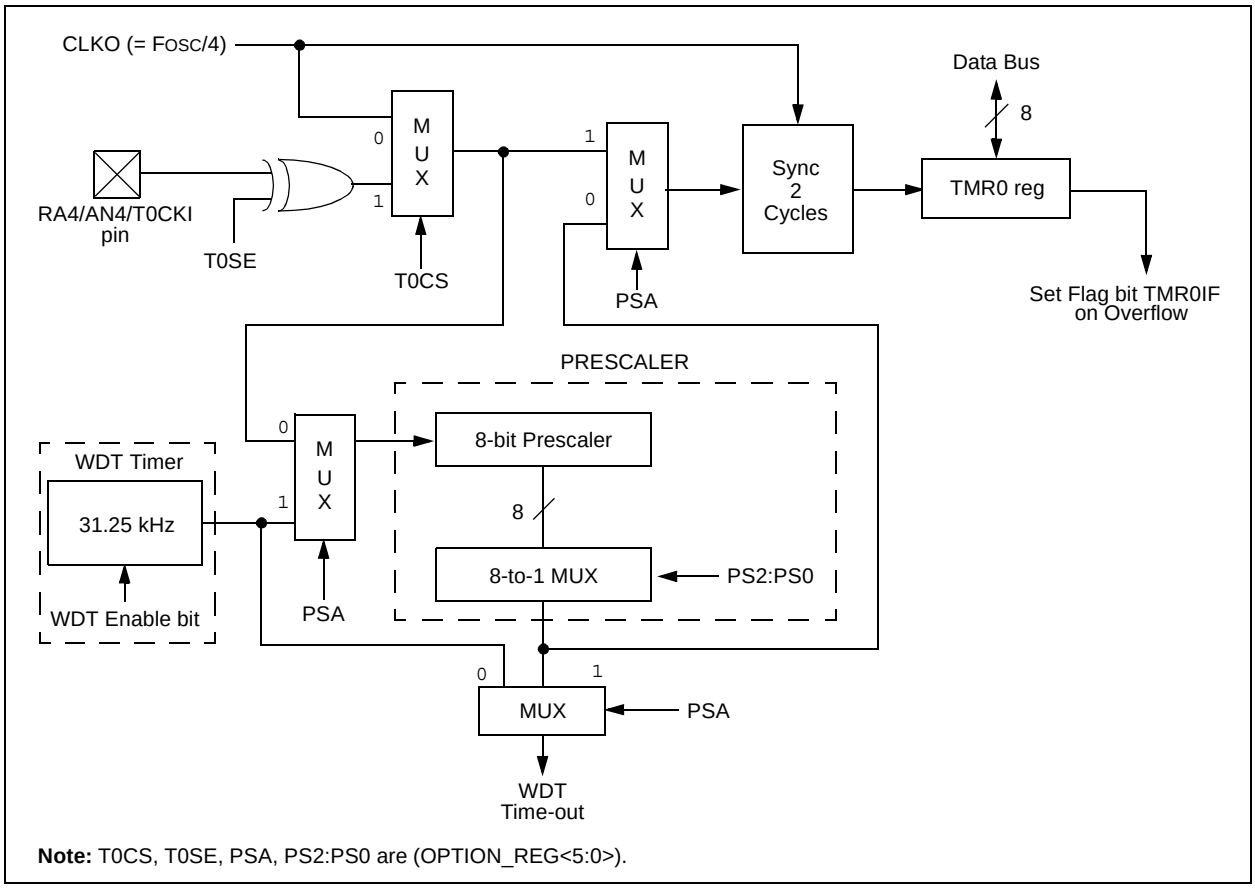
Counter mode is selected by setting bit T0CS (OPTION_REG<5>). In Counter mode, Timer0 will increment either on every rising or falling edge of pin RA4/AN4/T0CKI. The incrementing edge is determined by the Timer0 Source Edge Select bit, T0SE (OPTION_REG<4>). Clearing bit T0SE selects the rising edge. Restrictions on the external clock input are discussed in detail in **Section 6.3 "Using Timer0 with an External Clock"**.

The prescaler is mutually exclusively shared between the Timer0 module and the Watchdog Timer. The prescaler is not readable or writable. **Section 6.4 "Prescaler"** details the operation of the prescaler.

### 6.2 Timer0 Interrupt

The TMR0 interrupt is generated when the TMR0 register overflows from FFh to 00h. This overflow sets bit, TMR0IF (INTCON<2>). The interrupt can be masked by clearing bit, TMR0IE (INTCON<5>). Bit TMR0IF 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 6-1: BLOCK DIAGRAM OF THE TIMER0/WDT PRESCALER**




**Note:** T0CS, T0SE, PSA, PS2:PS0 are (OPTION_REG<5:0>).



Downloaded from Arrow.com.

## 6.3 Using Timer0 with an External Clock

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

## 6.4 Prescaler

There is only one prescaler available which is mutually exclusively shared between the Timer0 module and the Watchdog Timer. A prescaler assignment for the Timer0 module means that there is no prescaler for the Watchdog Timer and vice versa. This prescaler is not readable or writable (see Figure 6-1).

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

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

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

**REGISTER 6-1: OPTION_REG: OPTION REGISTER (ADDRESS 81h, 181h)**

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

bit 7 **$\overline{\text{RBPU}}$**: PORTB Pull-up Enable bit
1 = PORTB pull-ups are disabled
0 = PORTB pull-ups are enabled by individual port latch values

bit 6 **INTEDG:** Interrupt Edge Select bit
1 = Interrupt on rising edge of RB0/INT pin
0 = Interrupt on falling edge of RB0/INT pin

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

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

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

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

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

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

| **Note:** | To avoid an unintended device Reset, the instruction sequence shown in the *"PICmicro® Mid-Range MCU Family 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. |
|---|---|


© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

**EXAMPLE 6-1: CHANGING THE PRESCALER ASSIGNMENT FROM TIMER0 TO WDT**

```
    BANKSEL  OPTION_REG     ; Select Bank of OPTION_REG
    MOVLW    b'xx0x0xxx'    ; Select clock source and prescale value of
    MOVWF    OPTION_REG     ; other than 1:1
    BANKSEL  TMR0           ; Select Bank of TMR0
    CLRF     TMR0           ; Clear TMR0 and prescaler
    BANKSEL  OPTION_REG     ; Select Bank of OPTION_REG
    MOVLW    b'xxxx1xxx'    ; Select WDT, do not change prescale value
    MOVWF    OPTION_REG
    CLRWDT                  ; Clears WDT and prescaler
    MOVLW    b'xxxx1xxx'    ; Select new prescale value and WDT
    MOVWF    OPTION_REG
```

**EXAMPLE 6-2: CHANGING THE PRESCALER ASSIGNMENT FROM WDT TO TIMER0**

```
CLRWDT                  ; Clear WDT and prescaler
BANKSEL  OPTION_REG     ; Select Bank of OPTION_REG
MOVLW    b'xxxx0xxx'    ; Select TMR0, new prescale
MOVWF    OPTION_REG     ; value and clock source
```

**TABLE 6-1: REGISTERS ASSOCIATED WITH TIMER0**

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR, BOR | Value on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 01h,101h | TMR0 | Timer0 Module Register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 0Bh,8Bh, 10Bh,18Bh | INTCON | GIE | PEIE | TMR0IE | INTE | RBIE | TMR0IF | INTF | RBIF | 0000 000x | 0000 000u |
| 81h,181h | OPTION_REG | $\overline{\text{RBPU}}$ | INTEDG | T0CS | T0SE | PSA | PS2 | PS1 | PS0 | 1111 1111 | 1111 1111 |

**Legend:** x = unknown, u = unchanged, - = unimplemented locations read as '0'. Shaded cells are not used by Timer0.



Downloaded from Arrow.com.

**NOTES:**


© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

## 7.0 TIMER1 MODULE

The Timer1 module is a 16-bit timer/counter consisting of two 8-bit registers (TMR1H and TMR1L) which are readable and writable. 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>).

Timer1 can also be used to provide Real-Time Clock (RTC) functionality to applications with only a minimal addition of external components and code overhead.

### 7.1 Timer1 Operation

Timer1 can operate in one of three 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.

Timer1 can be enabled/disabled by setting/clearing control bit, TMR1ON (T1CON<0>).

Timer1 also has an internal “Reset input”. This Reset can be generated by the CCP1 module as the special event trigger (see **Section 9.1 “Capture Mode”**). Register 7-1 shows the Timer1 Control register.

When the Timer1 oscillator is enabled (T1OSCEN is set), the RB6/T1OSO/T1CKI/PGC and RB7/T1OSI/PGD pins become inputs. That is, the TRISB<7:6> value is ignored and these pins read as ‘0’.

Additional information on timer modules is available in the *“PICmicro® Mid-Range MCU Family Reference Manual”* (DS33023).

**REGISTER 7-1: T1CON: TIMER1 CONTROL REGISTER (ADDRESS 10h)**

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

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

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

bit 3 **T1OSCEN:** Timer1 Oscillator Enable Control bit
1 = Oscillator is enabled
0 = Oscillator is shut-off (the oscillator inverter is turned off to eliminate power drain)

bit 2 **$\overline{\text{T1SYNC}}$:** 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.

bit 1 **TMR1CS:** Timer1 Clock Source Select bit
1 = External clock from pin RB6/T1OSO/T1CKI/PGC (on the rising edge)
0 = Internal clock (FOSC/4)

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

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



Downloaded from Arrow.com.

## 7.2 Timer1 Operation in Timer Mode

Timer mode is selected by clearing the TMR1CS (T1CON<1>) bit. In this mode, the input clock to the timer is FOSC/4. The synchronize control bit, $\overline{\text{T1SYNC}}$ (T1CON<2>), has no effect since the internal clock is always in sync.

## 7.3 Timer1 Counter Operation

Timer1 may operate in Asynchronous or Synchronous mode depending on the setting of the TMR1CS bit.

When Timer1 is being incremented via an external source, increments occur on a rising edge. After Timer1 is enabled in Counter mode, the module must first have a falling edge before the counter begins to increment.

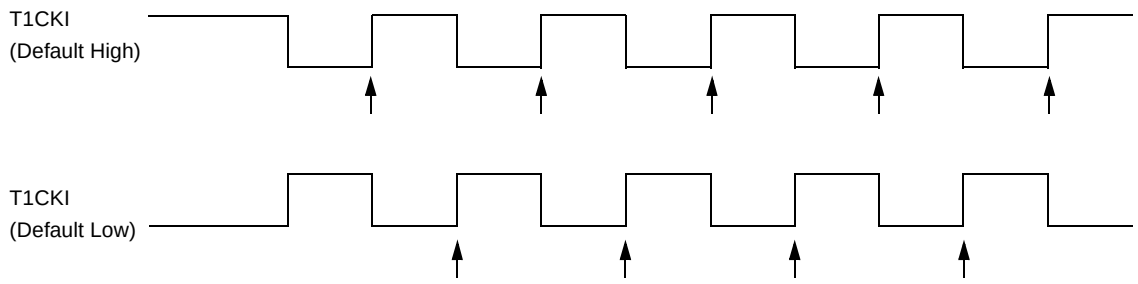
## 7.4 Timer1 Operation in Synchronized Counter Mode

Counter mode is selected by setting bit TMR1CS. In this mode, the timer increments on every rising edge of clock input on pin RB7/T1OSI/PGD when bit T1OSCEN is set, or on pin RB6/T1OSO/T1CKI/PGC when bit T1OSCEN is cleared.

If $\overline{\text{T1SYNC}}$ is cleared, then the external clock input is synchronized with internal phase clocks. The synchronization is done after the prescaler stage. The prescaler stage is an asynchronous ripple counter.

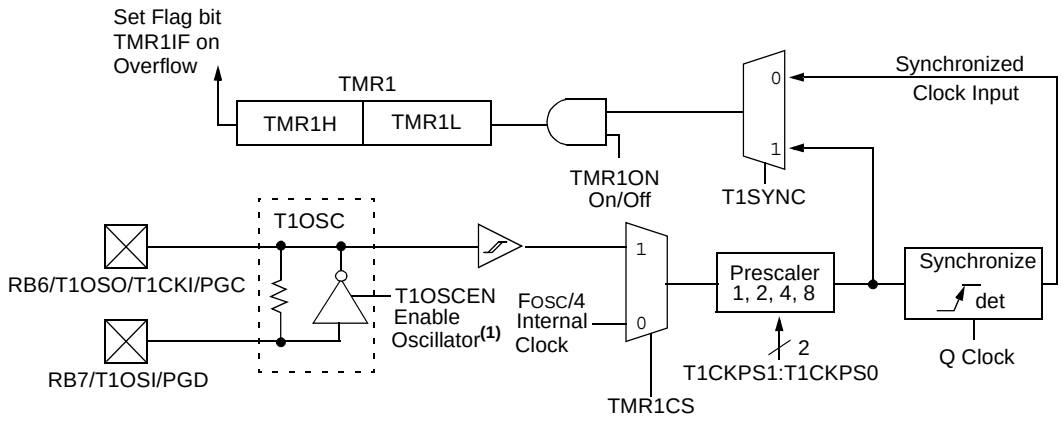
In this configuration, during Sleep mode, Timer1 will not increment even if the external clock is present, since the synchronization circuit is shut-off. The prescaler, however, will continue to increment.

**FIGURE 7-1: TIMER1 INCREMENTING EDGE**




**Note:** Arrows indicate counter increments.

**FIGURE 7-2: TIMER1 BLOCK DIAGRAM**




**Note 1:** When the T1OSCEN bit is cleared, the inverter is turned off. This eliminates power drain.


© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

## 7.5 Timer1 Operation in Asynchronous Counter Mode

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

In Asynchronous Counter mode, Timer1 cannot be used as a time base for capture or compare operations.

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

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

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

Reading the 16-bit value requires some care. The example codes provided in Example 7-1 and Example 7-2 demonstrate how to write to and read Timer1 while it is running in Asynchronous mode.

**EXAMPLE 7-1: WRITING A 16-BIT FREE RUNNING TIMER**

```
; All interrupts are disabled
CLRF     TMR1L        ; Clear Low byte, Ensures no rollover into TMR1H
MOVLW    HI_BYTE      ; Value to load into TMR1H
MOVWF    TMR1H, F     ; Write High byte
MOVLW    LO_BYTE      ; Value to load into TMR1L
MOVWF    TMR1H, F     ; Write Low byte
; Re-enable the Interrupt (if required)
CONTINUE              ; Continue with your code
```

**EXAMPLE 7-2: READING A 16-BIT FREE RUNNING TIMER**

```
; All interrupts are disabled
MOVF     TMR1H, W     ; Read high byte
MOVWF    TMPH
MOVF     TMR1L, W     ; Read low byte
MOVWF    TMPL
MOVF     TMR1H, W     ; Read high byte
SUBWF    TMPH, W      ; Sub 1st read with 2nd read
BTFSC    STATUS, Z    ; Is result = 0
GOTO     CONTINUE     ; Good 16-bit read
; TMR1L may have rolled over between the read of the high and low bytes.
; Reading the high and low bytes now will read a good value.
MOVF     TMR1H, W     ; Read high byte
MOVWF    TMPH
MOVF     TMR1L, W     ; Read low byte
MOVWF    TMPL         ; Re-enable the Interrupt (if required)
CONTINUE              ; Continue with your code
```



Downloaded from Arrow.com.

## 7.6 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, rated up to 32.768 kHz. It will continue to run during Sleep. It is primarily intended for a 32 kHz crystal. The circuit for a typical LP oscillator is shown in Figure 7-3. Table 7-1 shows the capacitor selection for the Timer1 oscillator.

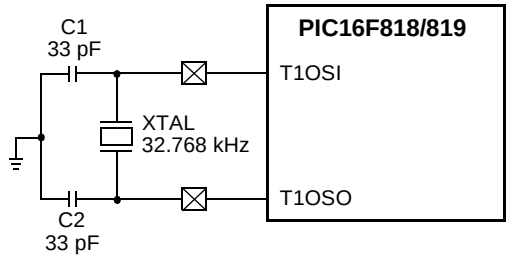
The user must provide a software time delay to ensure proper oscillator start-up.

**Note:** The Timer1 oscillator shares the T1OSI and T1OSO pins with the PGD and PGC pins used for programming and debugging.

When using the Timer1 oscillator, In-Circuit Serial Programming™ (ICSP™) may not function correctly (high-voltage or low-voltage) or the In-Circuit Debugger (ICD) may not communicate with the controller. As a result of using either ICSP or ICD, the Timer1 crystal may be damaged.

If ICSP or ICD operations are required, the crystal should be disconnected from the circuit (disconnect either lead) or installed after programming. The oscillator loading capacitors may remain in-circuit during ICSP or ICD operation.

**FIGURE 7-3: EXTERNAL COMPONENTS FOR THE TIMER1 LP OSCILLATOR**




**Note:** See the Notes with Table 7-1 for additional information about capacitor selection.

**TABLE 7-1: CAPACITOR SELECTION FOR THE TIMER1 OSCILLATOR**

| Osc Type | Freq | C1 | C2 |
|---|---|---|---|
| LP | 32 kHz | 33 pF | 33 pF |

**Note 1:** Microchip suggests this value as a starting point in validating the oscillator circuit.

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

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

**4:** Capacitor values are for design guidance only.

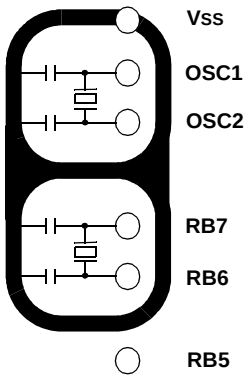
## 7.7 Timer1 Oscillator Layout Considerations

The Timer1 oscillator circuit draws very little power during operation. Due to the low-power nature of the oscillator, it may also be sensitive to rapidly changing signals in close proximity.

The oscillator circuit, shown in Figure 7-3, should be located as close as possible to the microcontroller. There should be no circuits passing within the oscillator circuit boundaries other than VSS or VDD.

If a high-speed circuit must be located near the oscillator, a grounded guard ring around the oscillator circuit, as shown in Figure 7-4, may be helpful when used on a single-sided PCB or in addition to a ground plane.

**FIGURE 7-4: OSCILLATOR CIRCUIT WITH GROUNDED GUARD RING**





© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

## 7.8 Resetting Timer1 Using a CCP Trigger Output

If the CCP1 module is configured in Compare mode to generate a "special event trigger" signal (CCP1M3:CCP1M0 = 1011), the signal will reset Timer1 and start an A/D conversion (if the A/D module is enabled).

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 CCP1, the write will take precedence.

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

## 7.9 Resetting Timer1 Register Pair (TMR1H, TMR1L)

TMR1H and TMR1L registers are not reset to 00h on a POR or any other Reset, except by the CCP1 special event triggers.

T1CON register is reset to 00h on a Power-on Reset or a Brown-out Reset, which shuts off the timer and leaves a 1:1 prescale. In all other Resets, the register is unaffected.

## 7.10 Timer1 Prescaler

The prescaler counter is cleared on writes to the TMR1H or TMR1L registers.

## 7.11 Using Timer1 as a Real-Time Clock

Adding an external LP oscillator to Timer1 (such as the one described in **Section 7.6 "Timer1 Oscillator"**), gives users the option to include RTC functionality in their applications. This is accomplished with an inexpensive watch crystal to provide an accurate time base and several lines of application code to calculate the time. When operating in Sleep mode and using a battery or supercapacitor as a power source, it can completely eliminate the need for a separate RTC device and battery backup.

The application code routine, RTCisr, shown in Example 7-3, demonstrates a simple method to increment a counter at one-second intervals using an Interrupt Service Routine. Incrementing the TMR1 register pair to overflow, triggers the interrupt and calls the routine which increments the seconds counter by one; additional counters for minutes and hours are incremented as the previous counter overflows.

Since the register pair is 16 bits wide, counting up to overflow the register directly from a 32.768 kHz clock would take 2 seconds. To force the overflow at the required one-second intervals, it is necessary to preload it; the simplest method is to set the MSb of TMR1H with a BSF instruction. Note that the TMR1L register is never preloaded or altered; doing so may introduce cumulative error over many cycles.

For this method to be accurate, Timer1 must operate in Asynchronous mode and the Timer1 overflow interrupt must be enabled (PIE1<0> = 1) as shown in the routine, RTCinit. The Timer1 oscillator must also be enabled and running at all times.



Downloaded from Arrow.com.

**EXAMPLE 7-3: IMPLEMENTING A REAL-TIME CLOCK USING A TIMER1 INTERRUPT SERVICE**

```
RTCinit     BANKSEL   TMR1H
            MOVLW     0x80              ; Preload TMR1 register pair
            MOVWF     TMR1H             ; for 1 second overflow
            CLRF      TMR1L
            MOVLW     b'00001111'       ; Configure for external clock,
            MOVWF     T1CON             ; Asynchronous operation, external oscillator
            CLRF      secs              ; Initialize timekeeping registers
            CLRF      mins
            MOVLW     .12
            MOVWF     hours
            BANKSEL   PIE1
            BSF       PIE1, TMR1IE      ; Enable Timer1 interrupt
            RETURN
RTCisr      BANKSEL   TMR1H
            BSF       TMR1H, 7          ; Preload for 1 sec overflow
            BCF       PIR1, TMR1IF      ; Clear interrupt flag
            INCF      secs, F           ; Increment seconds
            MOVF      secs, w
            SUBLW     .60
            BTFSS     STATUS, Z         ; 60 seconds elapsed?
            RETURN                      ; No, done
            CLRF      seconds           ; Clear seconds
            INCF      mins, f           ; Increment minutes
            MOVF      mins, w
            SUBLW     .60
            BTFSS     STATUS, Z         ; 60 seconds elapsed?
            RETURN                      ; No, done
            CLRF      mins              ; Clear minutes
            INCF      hours, f          ; Increment hours
            MOVF      hours, w
            SUBLW     .24
            BTFSS     STATUS, Z         ; 24 hours elapsed?
            RETURN                      ; No, done
            CLRF      hours             ; Clear hours
            RETURN                      ; Done
```

**TABLE 7-2: REGISTERS ASSOCIATED WITH TIMER1 AS A TIMER/COUNTER**

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR, BOR | Value on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0Bh,8Bh, 10Bh,18Bh | INTCON | GIE | PEIE | TMR0IE | INTE | RBIE | TMR0IF | INTF | RBIF | 0000 000x | 0000 000u |
| 0Ch | PIR1 | — | ADIF | — | — | SSPIF | CCP1IF | TMR2IF | TMR1IF | -0-- 0000 | -0-- 0000 |
| 8Ch | PIE1 | — | ADIE | — | — | SSPIE | CCP1IE | TMR2IE | TMR1IE | -0-- 0000 | -0-- 0000 |
| 0Eh | TMR1L | Holding Register for the Least Significant Byte of the 16-bit TMR1 Register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 0Fh | TMR1H | Holding Register for the Most Significant Byte of the 16-bit TMR1 Register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 10h | T1CON | — | — | T1CKPS1 | T1CKPS0 | T1OSCEN | $\overline{\text{T1SYNC}}$ | TMR1CS | TMR1ON | --00 0000 | --uu uuuu |

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


© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

## 8.0 TIMER2 MODULE

Timer2 is an 8-bit timer with a prescaler and a postscaler. It can be used as the PWM time base for the PWM mode of the CCP1 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 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.

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>)).

Timer2 can be shut-off by clearing control bit, TMR2ON (T2CON<2>), to minimize power consumption.

Register 8-1 shows the Timer2 Control register.

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

### 8.1 Timer2 Prescaler and Postscaler

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

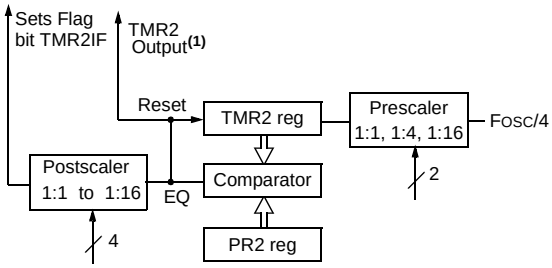
- A write to the TMR2 register
- A write to the T2CON register
- Any device Reset (Power-on Reset, $\overline{\text{MCLR}}$, WDT Reset or Brown-out Reset)

TMR2 is not cleared when T2CON is written.

### 8.2 Output of TMR2

The output of TMR2 (before the postscaler) is fed to the Synchronous Serial Port module which optionally uses it to generate a shift clock.

**FIGURE 8-1: TIMER2 BLOCK DIAGRAM**




**Note 1:** TMR2 register output can be software selected by the SSP module as a baud clock.



Downloaded from Arrow.com.

**REGISTER 8-1: T2CON: TIMER2 CONTROL REGISTER (ADDRESS 12h)**

| U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | TOUTPS3 | TOUTPS2 | TOUTPS1 | TOUTPS0 | TMR2ON | T2CKPS1 | T2CKPS0 |
| bit 7 | | | | | | | bit 0 |

bit 7 **Unimplemented:** Read as '0'

bit 6-3 **TOUTPS3:TOUTPS0:** Timer2 Output Postscale Select bits

0000 = 1:1 Postscale
0001 = 1:2 Postscale
0010 = 1:3 Postscale
•
•
•
1111 = 1:16 Postscale

bit 2 **TMR2ON:** Timer2 On bit

1 = Timer2 is on
0 = Timer2 is off

bit 1-0 **T2CKPS1:T2CKPS0:** Timer2 Clock Prescale Select bits

00 = Prescaler is 1
01 = Prescaler is 4
1x = Prescaler is 16

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

**TABLE 8-1: REGISTERS ASSOCIATED WITH TIMER2 AS A TIMER/COUNTER**

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR, BOR | Value on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0Bh, 8Bh, 10Bh, 18Bh | INTCON | GIE | PEIE | TMR0IE | INTE | RBIE | TMR0IF | INTF | RBIF | 0000 000x | 0000 000u |
| 0Ch | PIR1 | — | ADIF | — | — | SSPIF | CCP1IF | TMR2IF | TMR1IF | -0-- 0000 | -0-- 0000 |
| 8Ch | PIE1 | — | ADIE | — | — | SSPIE | CCP1IE | TMR2IE | TMR1IE | -0-- 0000 | -0-- 0000 |
| 11h | TMR2 | Timer2 Module Register | | | | | | | | 0000 0000 | 0000 0000 |
| 12h | T2CON | — | TOUTPS3 | TOUTPS2 | TOUTPS1 | TOUTPS0 | TMR2ON | T2CKPS1 | T2CKPS0 | -000 0000 | -000 0000 |
| 92h | PR2 | Timer2 Period Register | | | | | | | | 1111 1111 | 1111 1111 |

**Legend:** x = unknown, u = unchanged, - = unimplemented, read as '0'. Shaded cells are not used by the Timer2 module.


© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

## 9.0 CAPTURE/COMPARE/PWM (CCP) MODULE

The Capture/Compare/PWM (CCP) module contains a 16-bit register that can operate as a:

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

Table 9-1 shows the timer resources of the CCP module modes.

Capture/Compare/PWM Register 1 (CCPR1) is comprised of two 8-bit registers: CCPR1L (low byte) and CCPR1H (high byte). The CCP1CON register controls the operation of CCP1. The special event trigger is generated by a compare match which will reset Timer1 and start an A/D conversion (if the A/D module is enabled).

The CCP module's input/output pin (CCP1) can be configured as RB2 or RB3. This selection is set in bit 12 (CCPMX) of the Configuration Word register.

Additional information on the CCP module is available in the *"PICmicro® Mid-Range MCU Family Reference Manual"* (DS33023) and in Application Note *AN594,* "*Using the CCP Module(s)"* (DS00594).

**TABLE 9-1: CCP MODE – TIMER RESOURCE**

| CCP Mode | Timer Resource |
|---|---|
| Capture | Timer1 |
| Compare | Timer1 |
| PWM | Timer2 |

**REGISTER 9-1: CCP1CON: CAPTURE/COMPARE/PWM CONTROL REGISTER 1 (ADDRESS 17h)**

| U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | — | CCP1X | CCP1Y | CCP1M3 | CCP1M2 | CCP1M1 | CCP1M0 |
| bit 7 | | | | | | | bit 0 |

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

bit 5-4 **CCP1X:CCP1Y:** 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 CCPRxL.

bit 3-0 **CCP1M3:CCP1M0:** CCP1 Mode Select bits

0000 = Capture/Compare/PWM disabled (resets CCP1 module)
0100 = Capture mode, every falling edge
0101 = Capture mode, every rising edge
0110 = Capture mode, every 4th rising edge
0111 = Capture mode, every 16th rising edge
1000 = Compare mode, set output on match (CCP1IF bit is set)
1001 = Compare mode, clear output on match (CCP1IF bit is set)
1010 = Compare mode, generate software interrupt on match (CCP1IF bit is set, CCP1 pin is unaffected)
1011 = Compare mode, trigger special event (CCP1IF bit is set, CCP1 pin is unaffected); CCP1 resets TMR1 and starts an A/D conversion (if A/D module is enabled)
11xx = PWM mode

| **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 |



Downloaded from Arrow.com.

## 9.1 Capture Mode

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

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

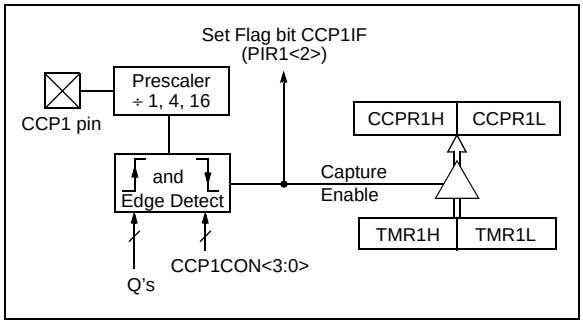
An event is selected by control bits, CCP1M3:CCP1M0 (CCP1CON<3:0>). When a capture is made, the interrupt request flag bit, CCP1IF (PIR1<2>), is set. It must be cleared in software. If another capture occurs before the value in register CCPR1 is read, the old captured value is overwritten by the new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

### 9.1.1 CCP PIN CONFIGURATION

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

| | |
|---|---|
| **Note 1:** | If the CCP1 pin is configured as an output, a write to the port can cause a capture condition. |
| **2:** | The TRISB bit (2 or 3) is dependent upon the setting of configuration bit 12 (CCPMX). |

**FIGURE 9-1: CAPTURE MODE OPERATION BLOCK DIAGRAM**




### 9.1.2 TIMER1 MODE SELECTION

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

### 9.1.3 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.

### 9.1.4 CCP PRESCALER

There are four prescaler settings specified by bits CCP1M3:CCP1M0. Whenever the CCP module is turned off, or the CCP 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 9-1 shows the recommended method for switching between capture prescalers. This example also clears the prescaler counter and will not generate the “false” interrupt.

**EXAMPLE 9-1: CHANGING BETWEEN CAPTURE PRESCALERS**

```
CLRF   CCP1CON     ;Turn CCP module off
MOVLW  NEW_CAPT_PS ;Load the W reg with
                   ;the new prescaler
                   ;move value and CCP ON
MOVWF  CCP1CON     ;Load CCP1CON with this
                   ;value
```


© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

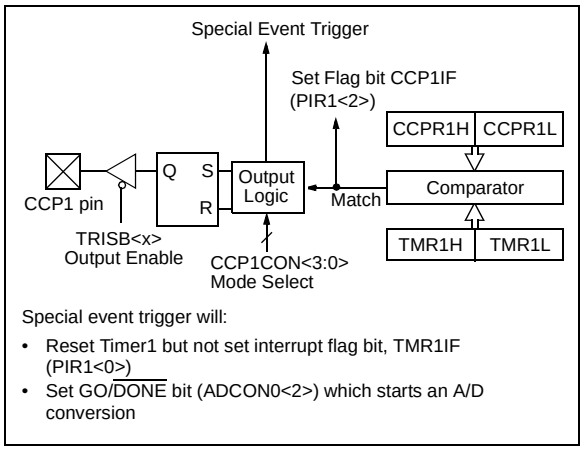
## 9.2 Compare Mode

In Compare mode, the 16-bit CCPR1 register value is constantly compared against the TMR1 register pair value. When a match occurs, the CCP1 pin is:

- Driven high
- Driven low
- 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.

**FIGURE 9-2: COMPARE MODE OPERATION BLOCK DIAGRAM**




### 9.2.1 CCP PIN CONFIGURATION

The user must configure the CCP1 pin as an output by clearing the TRISB<x> bit.

| | |
|---|---|
| **Note 1:** | Clearing the CCP1CON register will force the CCP1 compare output latch to the default low level. This is not the data latch. |
| **2:** | The TRISB bit (2 or 3) is dependent upon the setting of configuration bit 12 (CCPMX). |

### 9.2.2 TIMER1 MODE SELECTION

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

### 9.2.3 SOFTWARE INTERRUPT MODE

When generate software interrupt is chosen, the CCP1 pin is not affected. Only a CCP interrupt is generated (if enabled).

### 9.2.4 SPECIAL EVENT TRIGGER

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

The special event trigger output of CCP1 resets the TMR1 register pair and starts an A/D conversion (if the A/D module is enabled). This allows the CCPR1 register to effectively be a 16-bit programmable period register for Timer1.

| | |
|---|---|
| **Note:** | The special event trigger from the CCP1 module will not set interrupt flag bit, TMR1IF (PIR1<0>). |

**TABLE 9-2: REGISTERS ASSOCIATED WITH CAPTURE, COMPARE AND TIMER1**

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR, BOR | Value on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0Bh,8Bh 10BH,18Bh | INTCON | GIE | PEIE | TMR0IE | INTE | RBIE | TMR0IF | INTF | RBIF | 0000 000x | 0000 000u |
| 0Ch | PIR1 | — | ADIF | — | — | SSPIF | CCP1IF | TMR2IF | TMR1IF | -0-- 0000 | -0-- 0000 |
| 8Ch | PIE1 | — | ADIE | — | — | SSPIE | CCP1IE | TMR2IE | TMR1IE | -0-- 0000 | -0-- 0000 |
| 86h | TRISB | PORTB Data Direction Register | | | | | | | | 1111 1111 | 1111 1111 |
| 0Eh | TMR1L | Holding Register for the Least Significant Byte of the 16-bit TMR1 Register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 0Fh | TMR1H | Holding Register for the Most Significant Byte of the 16-bit TMR1 Register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 10h | T1CON | — | — | T1CKPS1 | T1CKPS0 | T1OSCEN | $\overline{\text{T1SYNC}}$ | TMR1CS | TMR1ON | --00 0000 | --uu uuuu |
| 15h | CCPR1L | Capture/Compare/PWM Register 1 (LSB) | | | | | | | | xxxx xxxx | uuuu uuuu |
| 16h | CCPR1H | Capture/Compare/PWM Register 1 (MSB) | | | | | | | | xxxx xxxx | uuuu uuuu |
| 17h | CCP1CON | — | — | CCP1X | CCP1Y | CCP1M3 | CCP1M2 | CCP1M1 | CCP1M0 | --00 0000 | --00 0000 |

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



Downloaded from Arrow.com.

## 9.3 PWM Mode

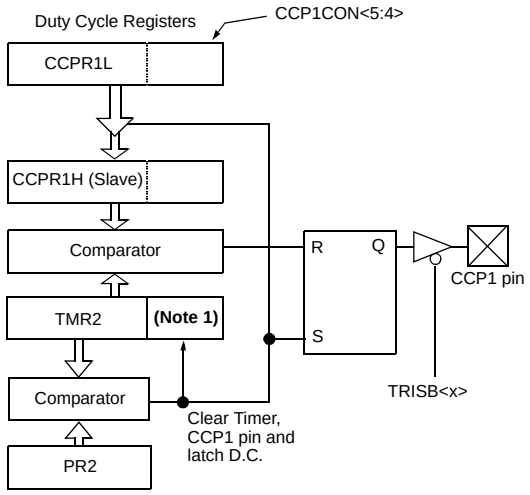
In Pulse-Width Modulation (PWM) mode, the CCP1 pin produces up to a 10-bit resolution PWM output. Since the CCP1 pin is multiplexed with the PORTB data latch, the TRISB<x> bit must be cleared to make the CCP1 pin an output.

| **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 9-3 shows a simplified block diagram of the CCP module in PWM mode.

For a step by step procedure on how to set up the CCP module for PWM operation, see **Section 9.3.3 "Setup for PWM Operation"**.

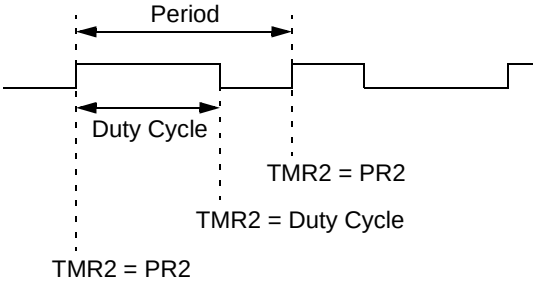
**FIGURE 9-3: SIMPLIFIED PWM BLOCK DIAGRAM**




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

A PWM output (Figure 9-4) 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 9-4: PWM OUTPUT**




### 9.3.1 PWM PERIOD

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

**EQUATION 9-1:**

$$\text{PWM Period} = [(PR2) + 1] \cdot 4 \cdot T_{OSC} \cdot (\text{TMR2 Prescale Value})$$

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 8.0 "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. |
|---|---|

### 9.3.2 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.

**EQUATION 9-2:**

$$\text{PWM Duty Cycle} = (\text{CCPR1L:CCP1CON<5:4>}) \cdot T_{OSC} \cdot (\text{TMR2 Prescale Value})$$

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

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

When the CCPR1H 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.


© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

The maximum PWM resolution (bits) for a given PWM frequency is given by the following formula.

**EQUATION 9-3:**

$$\text{Resolution} = \frac{\log\left(\frac{FOSC}{FPWM}\right)}{\log(2)} \text{ bits}$$

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

### 9.3.3 SETUP FOR PWM OPERATION

The following steps should be taken when configuring the CCP 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<x> bit.
4. Set the TMR2 prescale value and enable Timer2 by writing to T2CON.
5. Configure the CCP1 module for PWM operation.

| **Note:** | The TRISB bit (2 or 3) is dependant upon the setting of configuration bit 12 (CCPMX). |
|---|---|

**TABLE 9-3: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS AT 20 MHz**

| PWM Frequency | 1.22 kHz | 4.88 kHz | 19.53 kHz | 78.12 kHz | 156.3 kHz | 208.3 kHz |
|---|---|---|---|---|---|---|
| Timer Prescaler (1, 4, 16) | 16 | 4 | 1 | 1 | 1 | 1 |
| PR2 Value | 0xFF | 0xFF | 0xFF | 0x3F | 0x1F | 0x17 |
| Maximum Resolution (bits) | 10 | 10 | 10 | 8 | 7 | 5.5 |

**TABLE 9-4: REGISTERS ASSOCIATED WITH PWM AND TIMER2**

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR, BOR | Value on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0Bh,8Bh 10Bh,18Bh | INTCON | GIE | PEIE | TMR0IE | INTE | RBIE | TMR0IF | INTF | RBIF | 0000 000x | 0000 000u |
| 0Ch | PIR1 | — | ADIF | — | — | SSPIF | CCP1IF | TMR2IF | TMR1IF | -0-- 0000 | -0-- 0000 |
| 8Ch | PIE1 | — | ADIE | — | — | SSPIE | CCP1IE | TMR2IE | TMR1IE | -0-- 0000 | -0-- 0000 |
| 86h | TRISB | PORTB Data Direction Register | | | | | | | | 1111 1111 | 1111 1111 |
| 11h | TMR2 | Timer2 Module Register | | | | | | | | 0000 0000 | 0000 0000 |
| 92h | PR2 | Timer2 Module Period Register | | | | | | | | 1111 1111 | 1111 1111 |
| 12h | T2CON | — | TOUTPS3 | TOUTPS2 | TOUTPS1 | TOUTPS0 | TMR2ON | T2CKPS1 | T2CKPS0 | -000 0000 | -000 0000 |
| 15h | CCPR1L | Capture/Compare/PWM Register 1 (LSB) | | | | | | | | xxxx xxxx | uuuu uuuu |
| 16h | CCPR1H | Capture/Compare/PWM Register 1 (MSB) | | | | | | | | xxxx xxxx | uuuu uuuu |
| 17h | CCP1CON | — | — | CCP1X | CCP1Y | CCP1M3 | CCP1M2 | CCP1M1 | CCP1M0 | --00 0000 | --00 0000 |

**Legend:** x = unknown, u = unchanged, - = unimplemented, read as '0'. Shaded cells are not used by PWM and Timer2.



Downloaded from Arrow.com.

**NOTES:**


© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

# 10.0 SYNCHRONOUS SERIAL PORT (SSP) MODULE

## 10.1 SSP Module Overview

The Synchronous Serial Port (SSP) module is a serial interface useful for communicating with other peripheral or microcontroller devices. These peripheral devices may be serial EEPROMs, shift registers, display drivers, A/D converters, etc. The SSP module can operate in one of two modes:

- Serial Peripheral Interface (SPI)
- Inter-Integrated Circuit ($I^2C$)

An overview of $I^2C$ operations and additional information on the SSP module can be found in the *"PICmicro® Mid-Range MCU Family Reference Manual"* (DS33023).

Refer to Application Note *AN578, "Use of the SSP Module in the $I^2C$™ Multi-Master Environment"* (DS00578).

## 10.2 SPI Mode

This section contains register definitions and operational characteristics of the SPI module.

SPI mode allows 8 bits of data to be synchronously transmitted and received simultaneously. To accomplish communication, typically three pins are used:

- Serial Data Out (SDO) RB2/SDO/CCP1
- Serial Data In (SDI) RB1/SDI/SDA
- Serial Clock (SCK) RB4/SCK/SCL

Additionally, a fourth pin may be used when in a Slave mode of operation:

- Slave Select ($\overline{SS}$) RB5/$\overline{SS}$

When initializing the SPI, several options need to be specified. This is done by programming the appropriate control bits in the SSPCON register (SSPCON<5:0>) and the SSPSTAT register (SSPSTAT<7:6>). These control bits allow the following to be specified:

- Master mode (SCK is the clock output)
- Slave mode (SCK is the clock input)
- Clock Polarity (Idle state of SCK)
- Clock Edge (output data on rising/falling edge of SCK)
- Clock Rate (Master mode only)
- Slave Select mode (Slave mode only)

| **Note:** | Before enabling the module in SPI Slave mode, the state of the clock line (SCK) must match the polarity selected for the Idle state. The clock line can be observed by reading the SCK pin. The polarity of the Idle state is determined by the CKP bit (SSPCON<4>). |
|---|---|



Downloaded from Arrow.com.

**REGISTER 10-1: SSPSTAT: SYNCHRONOUS SERIAL PORT STATUS REGISTER (ADDRESS 94h)**

| R/W-0 | R/W-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |
|---|---|---|---|---|---|---|---|
| SMP | CKE | D/$\overline{A}$ | P | S | R/$\overline{W}$ | UA | BF |
| bit 7 | | | | | | | bit 0 |

bit 7 **SMP:** SPI Data Input Sample Phase bit

SPI Master mode:

1 = Input data sampled at end of data output time

0 = Input data sampled at middle of data output time (Microwire)

SPI Slave mode:

This bit must be cleared when SPI is used in Slave mode.

$I^2C$ mode:

This bit must be maintained clear.

bit 6 **CKE:** SPI Clock Edge Select bit

1 = Transmit occurs on transition from active to Idle clock state

0 = Transmit occurs on transition from Idle to active clock state

**Note:** Polarity of clock state is set by the CKP bit (SSPCON<4>).

$I^2C$ mode:

This bit must be maintained clear.

bit 5 **D/$\overline{A}$:** Data/$\overline{\text{Address}}$ bit ($I^2C$ mode only)

In $I^2C$ Slave mode:

1 = Indicates that the last byte received was data

0 = Indicates that the last byte received was address

bit 4 **P:** Stop bit(1) ($I^2C$ mode only)

1 = Indicates that a Stop bit has been detected last

0 = Stop bit was not detected last

bit 3 **S:** Start bit(1) ($I^2C$ mode only)

1 = Indicates that a Start bit has been detected last (this bit is '0' on Reset)

0 = Start bit was not detected last

bit 2 **R/$\overline{W}$:** Read/$\overline{\text{Write}}$ Information bit ($I^2C$ mode only)

Holds the R/$\overline{W}$ bit information following the last address match and is only valid from address match to the next Start bit, Stop bit or $\overline{\text{ACK}}$ bit.

1 = Read

0 = Write

bit 1 **UA:** Update Address bit (10-bit $I^2C$ mode only)

1 = Indicates that the user needs to update the address in the SSPADD register

0 = Address does not need to be updated

bit 0 **BF:** Buffer Full Status bit

Receive (SPI and $I^2C$ modes):

1 = Receive complete, SSPBUF is full

0 = Receive not complete, SSPBUF is empty

Transmit (In $I^2C$ mode only):

1 = Transmit in progress, SSPBUF is full (8 bits)

0 = Transmit complete, SSPBUF is empty

**Note 1:** This bit is cleared when the SSP module is disabled (i.e., the SSPEN bit is cleared).

| **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 |



© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

**REGISTER 10-2: SSPCON: SYNCHRONOUS SERIAL PORT CONTROL REGISTER 1 (ADDRESS 14h)**

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| WCOL | SSPOV | SSPEN | CKP | SSPM3 | SSPM2 | SSPM1 | SSPM0 |
| bit 7 | | | | | | | bit 0 |

bit 7 **WCOL:** Write Collision Detect bit

1 = An attempt to write the SSPBUF register failed because the SSP module is busy (must be cleared in software)

0 = No collision

bit 6 **SSPOV:** Receive Overflow Indicator bit

In SPI mode:

1 = A new byte is received while the SSPBUF register is still holding the previous data. In case of overflow, the data in SSPSR is lost. Overflow can only occur in Slave mode. The user must read the SSPBUF, even if only transmitting data, to avoid setting overflow. In Master mode, the overflow bit is not set since each new reception (and transmission) is initiated by writing to the SSPBUF register.

0 = No overflow

In $I^2C$ mode:

1 = A byte is received while the SSPBUF register is still holding the previous byte. SSPOV is a "don't care" in Transmit mode. SSPOV must be cleared in software in either mode.

0 = No overflow

bit 5 **SSPEN:** Synchronous Serial Port Enable bit**(1)**

In SPI mode:

1 = Enables serial port and configures SCK, SDO and SDI as serial port pins

0 = Disables serial port and configures these pins as I/O port pins

In $I^2C$ mode:

1 = Enables the serial port and configures the SDA and SCL pins as serial port pins

0 = Disables serial port and configures these pins as I/O port pins

**Note 1:** In both modes, when enabled, these pins must be properly configured as input or output.

bit 4 **CKP:** Clock Polarity Select bit

In SPI mode:

1 = Transmit happens on falling edge, receive on rising edge. Idle state for clock is a high level.

0 = Transmit happens on rising edge, receive on falling edge. Idle state for clock is a low level.

In $I^2C$ Slave mode:

SCK release control.

1 = Enable clock

0 = Holds clock low (clock stretch). (Used to ensure data setup time.)

bit 3-0 **SSPM<3:0>:** Synchronous Serial Port Mode Select bits

0000 = SPI Master mode, clock = OSC/4

0001 = SPI Master mode, clock = OSC/16

0010 = SPI Master mode, clock = OSC/64

0011 = SPI Master mode, clock = TMR2 output/2

0100 = SPI Slave mode, clock = SCK pin. $\overline{SS}$ pin control enabled.

0101 = SPI Slave mode, clock = SCK pin. $\overline{SS}$ pin control disabled. $\overline{SS}$ can be used as I/O pin.

0110 = $I^2C$ Slave mode, 7-bit address

0111 = $I^2C$ Slave mode, 10-bit address

1011 = $I^2C$ Firmware Controlled Master mode (Slave Idle)

1110 = $I^2C$ Slave mode, 7-bit address with Start and Stop bit interrupts enabled

1111 = $I^2C$ Slave mode, 10-bit address with Start and Stop bit interrupts enabled

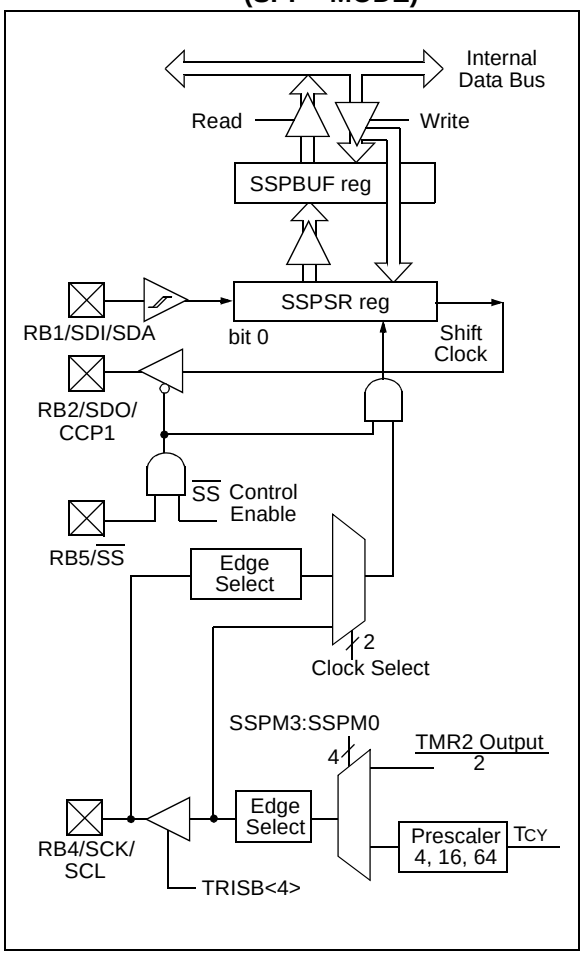
1000, 1001, 1010, 1100, 1101 = **Reserved**

| **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 |



Downloaded from Arrow.com.

**FIGURE 10-1: SSP BLOCK DIAGRAM (SPI™ MODE)**




To enable the serial port, SSP Enable bit, SSPEN (SSPCON<5>), must be set. To reset or reconfigure SPI mode, clear bit SSPEN, reinitialize the SSPCON register and then set bit SSPEN. This configures the SDI, SDO, SCK and $\overline{SS}$ pins as serial port pins. For the pins to behave as the serial port function, they must have their data direction bits (in the TRISB register) appropriately programmed. That is:

- SDI must have TRISB<1> set
- SDO must have TRISB<2> cleared
- SCK (Master mode) must have TRISB<4> cleared
- SCK (Slave mode) must have TRISB<4> set
- $\overline{SS}$ must have TRISB<5> set

**Note 1:** When the SPI is in Slave mode with the $\overline{SS}$ pin control enabled (SSPCON<3:0> = 0100), the SPI module will reset if the $\overline{SS}$ pin is set to VDD.

**2:** If the SPI is used in Slave mode with CKE = 1, then the $\overline{SS}$ pin control must be enabled.

**3:** When the SPI is in Slave mode with the $\overline{SS}$ pin control enabled (SSPCON<3:0> = 0100), the state of the $\overline{SS}$ pin can affect the state read back from the TRISB<2> bit. The peripheral OE signal from the SSP module into PORTB controls the state that is read back from the TRISB<2> bit. If read-modify-write instructions, such as BSF are performed on the TRISB register while the $\overline{SS}$ pin is high, this will cause the TRISB<2> bit to be set, thus disabling the SDO output.

**TABLE 10-1: REGISTERS ASSOCIATED WITH SPI™ OPERATION**

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR, BOR | Value on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0Bh,8Bh 10Bh,18Bh | INTCON | GIE | PEIE | TMR0IE | INTE | RBIE | TMR0IF | INTF | RBIF | 0000 000x | 0000 000u |
| 0Ch | PIR1 | — | ADIF | — | — | SSPIF | CCP1IF | TMR2IF | TMR1IF | -0-- 0000 | -0-- 0000 |
| 8Ch | PIE1 | — | ADIE | — | — | SSPIE | CCP1IE | TMR2IE | TMR1IE | -0-- 0000 | -0-- 0000 |
| 86h | TRISB | PORTB Data Direction Register | | | | | | | | 1111 1111 | 1111 1111 |
| 13h | SSPBUF | Synchronous Serial Port Receive Buffer/Transmit Register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 14h | SSPCON | WCOL | SSPOV | SSPEN | CKP | SSPM3 | SSPM2 | SSPM1 | SSPM0 | 0000 0000 | 0000 0000 |
| 94h | SSPSTAT | SMP | CKE | D/$\overline{A}$ | P | S | R/$\overline{W}$ | UA | BF | 0000 0000 | 0000 0000 |

**Legend:** x = unknown, u = unchanged, - = unimplemented, read as '0'. Shaded cells are not used by the SSP in SPI™ mode.


© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

**FIGURE 10-2: SPI™ MODE TIMING, MASTER MODE**

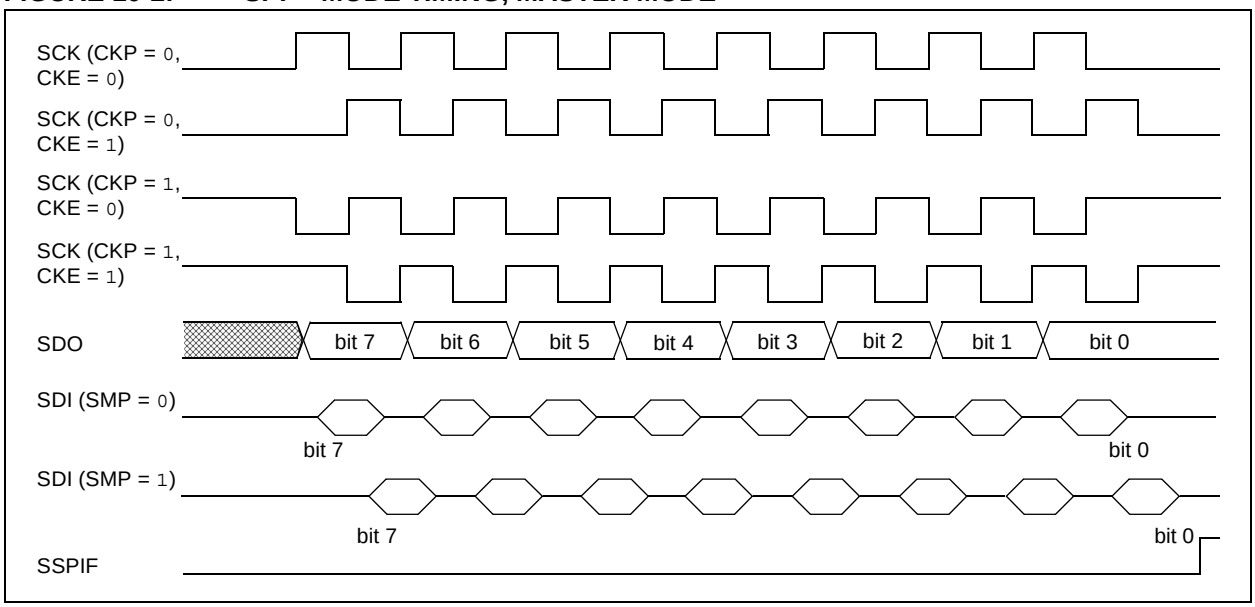



**FIGURE 10-3: SPI™ MODE TIMING (SLAVE MODE WITH CKE = 0)**

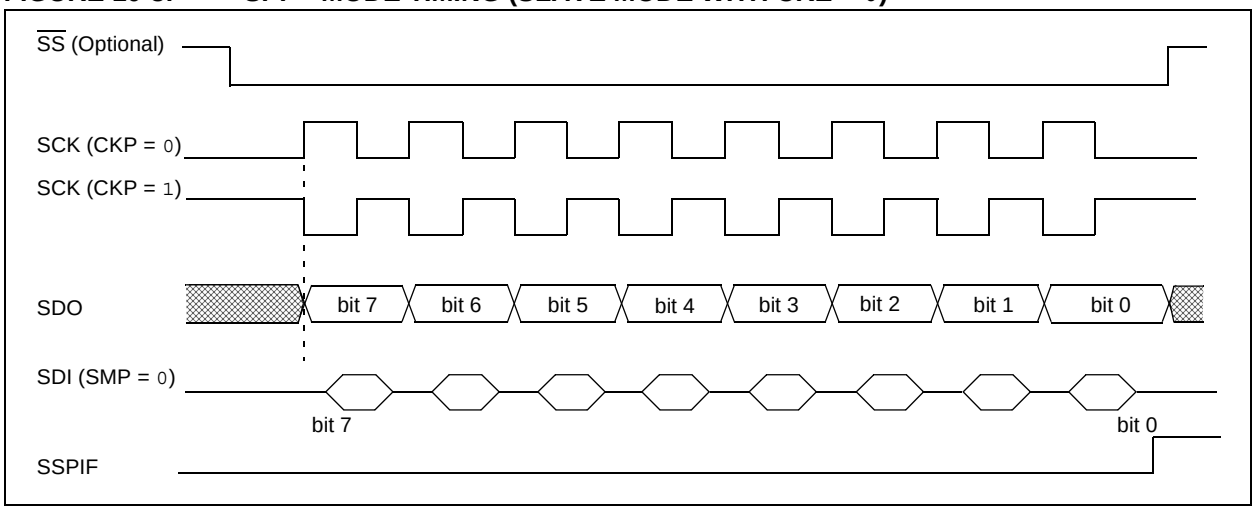
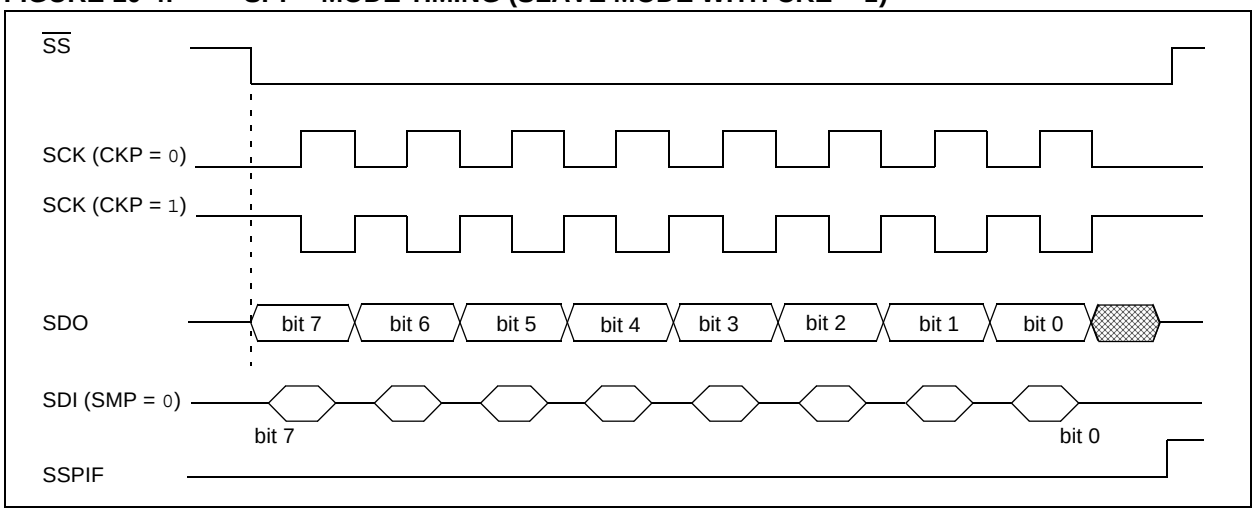



**FIGURE 10-4: SPI™ MODE TIMING (SLAVE MODE WITH CKE = 1)**






Downloaded from Arrow.com.

## 10.3 SSP I²C Mode Operation

The SSP module in I²C mode fully implements all slave functions, except general call support and provides interrupts on Start and Stop bits in hardware to facilitate firmware implementations of the master functions. The SSP module implements the standard mode specifications, as well as 7-bit and 10-bit addressing.

Two pins are used for data transfer. These are the RB4/SCK/SCL pin, which is the clock (SCL) and the RB1/SDI/SDA pin, which is the data (SDA). The user must configure these pins as inputs or outputs through the TRISB<4,1> bits.

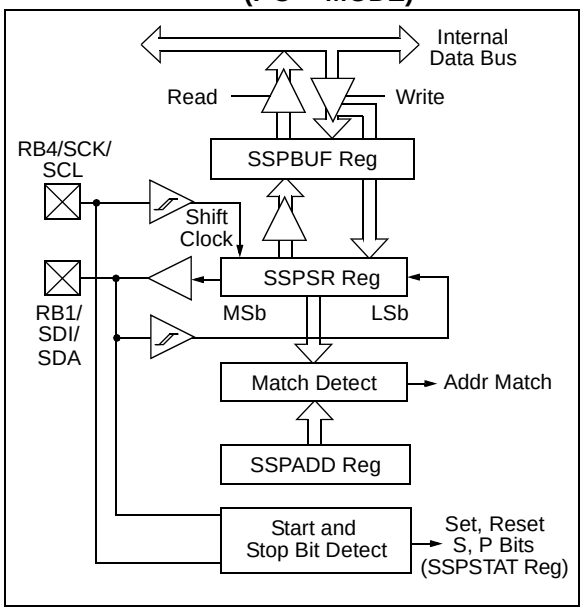
To ensure proper communication of the I²C Slave mode, the TRIS bits (TRISx [SDA, SCL]) corresponding to the I²C pins must be set to '1'. If any TRIS bits (TRISx<7:0>) of the port containing the I²C pins (PORTx [SDA, SCL]) are changed in software during I²C communication using a Read-Modify-Write instruction (BSF, BCF), then the I²C mode may stop functioning properly and I²C communication may suspend. Do not change any of the TRISx bits (TRIS bits of the port containing the I²C pins) using the instruction BSF or BCF during I²C communication. If it is absolutely necessary to change the TRISx bits during communication, the following method can be used:

**EXAMPLE 10-1:**

```
MOVF    TRISC, W        ; Example for an 18-pin part such as the PIC16F818/819
IORLW   0x18            ; Ensures <4:3> bits are '11'
ANDLW   B'11111001'     ; Sets <2:1> as output, but will not alter other bits
                        ; User can use their own logic here, such as IORLW, XORLW and ANDLW
MOVWF   TRISC
```

The SSP module functions are enabled by setting SSP Enable bit, SSPEN (SSPCON<5>).

**FIGURE 10-5: SSP BLOCK DIAGRAM (I²C™ MODE)**




The SSP module has five registers for I²C operation:

- SSP Control Register (SSPCON)
- SSP Status Register (SSPSTAT)
- Serial Receive/Transmit Buffer (SSPBUF)
- SSP Shift Register (SSPSR) – Not directly accessible
- SSP Address Register (SSPADD)

The SSPCON register allows control of the I²C operation. Four mode selection bits (SSPCON<3:0>) allow one of the following I²C modes to be selected:

- I²C Slave mode (7-bit address)
- I²C Slave mode (10-bit address)
- I²C Slave mode (7-bit address) with Start and Stop bit interrupts enabled to support Firmware Master mode
- I²C Slave mode (10-bit address) with Start and Stop bit interrupts enabled to support Firmware Master mode
- I²C Firmware Controlled Master mode with Start and Stop bit interrupts enabled, slave is Idle

Selection of any I²C mode, with the SSPEN bit set, forces the SCL and SDA pins to be open-drain, provided these pins are programmed to inputs by setting the appropriate TRISB bits. Pull-up resistors must be provided externally to the SCL and SDA pins for proper operation of the I²C module.

Additional information on SSP I²C operation may be found in the *"PICmicro® Mid-Range MCU Family Reference Manual"* (DS33023).


© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

### 10.3.1 SLAVE MODE

In Slave mode, the SCL and SDA pins must be configured as inputs (TRISB<4,1> set). The SSP module will override the input state with the output data when required (slave-transmitter).

When an address is matched, or the data transfer after an address match is received, the hardware automatically will generate the Acknowledge (ACK) pulse and then load the SSPBUF register with the received value currently in the SSPSR register.

Either or both of the following conditions will cause the SSP module not to give this ACK pulse:

a) The Buffer Full bit, BF (SSPSTAT<0>), was set before the transfer was received.
b) The overflow bit, SSPOV (SSPCON<6>), was set before the transfer was received.

In this case, the SSPSR register value is not loaded into the SSPBUF but bit, SSPIF (PIR1<3>), is set. Table 10-2 shows what happens when a data transfer byte is received, given the status of bits BF and SSPOV. The shaded cells show the condition where user software did not properly clear the overflow condition. Flag bit BF is cleared by reading the SSPBUF register while bit SSPOV is cleared through software.

The SCL clock input must have a minimum high and low for proper operation. The high and low times of the $I^2C$ specification, as well as the requirement of the SSP module, are shown in timing parameter #100 and parameter #101.

#### 10.3.1.1 Addressing

Once the SSP module has been enabled, it waits for a Start condition to occur. Following the Start condition, the eight bits are shifted into the SSPSR register. All incoming bits are sampled with the rising edge of the clock (SCL) line. The value of register SSPSR<7:1> is compared to the value of the SSPADD register. The address is compared on the falling edge of the eighth clock (SCL) pulse. If the addresses match and the BF and SSPOV bits are clear, the following events occur:

a) The SSPSR register value is loaded into the SSPBUF register.
b) The Buffer Full bit, BF, is set.
c) An ACK pulse is generated.
d) SSP Interrupt Flag bit, SSPIF (PIR1<3>), is set (interrupt is generated if enabled) – on the falling edge of the ninth SCL pulse.

In 10-bit Address mode, two address bytes need to be received by the slave device. The five Most Significant bits (MSbs) of the first address byte specify if this is a 10-bit address. Bit R/W (SSPSTAT<2>) must specify a write so the slave device will receive the second address byte. For a 10-bit address, the first byte would equal '1111 0 A9 A8 0', where A9 and A8 are the two MSbs of the address.

The sequence of events for 10-bit address is as follows, with steps 7-9 for slave-transmitter:

1. Receive first (high) byte of address (bits SSPIF, BF and bit UA (SSPSTAT<1>) are set).
2. Update the SSPADD register with second (low) byte of address (clears bit UA and releases the SCL line).
3. Read the SSPBUF register (clears bit BF) and clear flag bit, SSPIF.
4. Receive second (low) byte of address (bits SSPIF, BF and UA are set).
5. Update the SSPADD register with the first (high) byte of address; if match releases SCL line, this will clear bit UA.
6. Read the SSPBUF register (clears bit BF) and clear flag bit, SSPIF.
7. Receive Repeated Start condition.
8. Receive first (high) byte of address (bits SSPIF and BF are set).
9. Read the SSPBUF register (clears bit BF) and clear flag bit, SSPIF.

#### 10.3.1.2 Reception

When the R/W bit of the address byte is clear and an address match occurs, the R/W bit of the SSPSTAT register is cleared. The received address is loaded into the SSPBUF register.

When the address byte overflow condition exists, then a no Acknowledge (ACK) pulse is given. An overflow condition is indicated if either bit, BF (SSPSTAT<0>), is set or bit, SSPOV (SSPCON<6>), is set.

An SSP interrupt is generated for each data transfer byte. Flag bit, SSPIF (PIR1<3>), must be cleared in software. The SSPSTAT register is used to determine the status of the byte.

#### 10.3.1.3 Transmission

When the R/W bit of the incoming address byte is set and an address match occurs, the R/W bit of the SSPSTAT register is set. The received address is loaded into the SSPBUF register. The ACK pulse will be sent on the ninth bit and pin RB4/SCK/SCL is held low. The transmit data must be loaded into the SSPBUF register which also loads the SSPSR register. Then pin RB4/SCK/SCL should be enabled by setting bit, CKP (SSPCON<4>). The master device must monitor the SCL pin prior to asserting another clock pulse. The slave devices may be holding off the master device by stretching the clock. The eight data bits are shifted out on the falling edge of the SCL input. This ensures that the SDA signal is valid during the SCL high time (Figure 10-7).



Downloaded from Arrow.com.

An SSP interrupt is generated for each data transfer byte. Flag bit SSPIF must be cleared in software and the SSPSTAT register is used to determine the status of the byte. Flag bit SSPIF is set on the falling edge of the ninth clock pulse.

As a slave-transmitter, the $\overline{ACK}$ pulse from the master-receiver is latched on the rising edge of the ninth SCL input pulse. If the SDA line was high (not $\overline{ACK}$), then the data transfer is complete. When the $\overline{ACK}$ is latched by the slave device, the slave logic is reset (resets SSPSTAT register) and the slave device then monitors for another occurrence of the Start bit. If the SDA line was low ($\overline{ACK}$), the transmit data must be loaded into the SSPBUF register which also loads the SSPSR register. Then pin RB4/SCK/SCL should be enabled by setting bit, CKP.

**TABLE 10-2: DATA TRANSFER RECEIVED BYTE ACTIONS**

| Status Bits as Data Transfer is Received | | SSPSR → SSPBUF | Generate $\overline{ACK}$ Pulse | Set bit SSPIF (SSP interrupt occurs if enabled) |
|---|---|---|---|---|
| BF | SSPOV | | | |
| 0 | 0 | Yes | Yes | Yes |
| 1 | 0 | No | No | Yes |
| 1 | 1 | No | No | Yes |
| 0 | 1 | No | No | Yes |

**Note 1:** Shaded cells show the conditions where the user software did not properly clear the overflow condition.

**FIGURE 10-6: I²C™ WAVEFORMS FOR RECEPTION (7-BIT ADDRESS)**

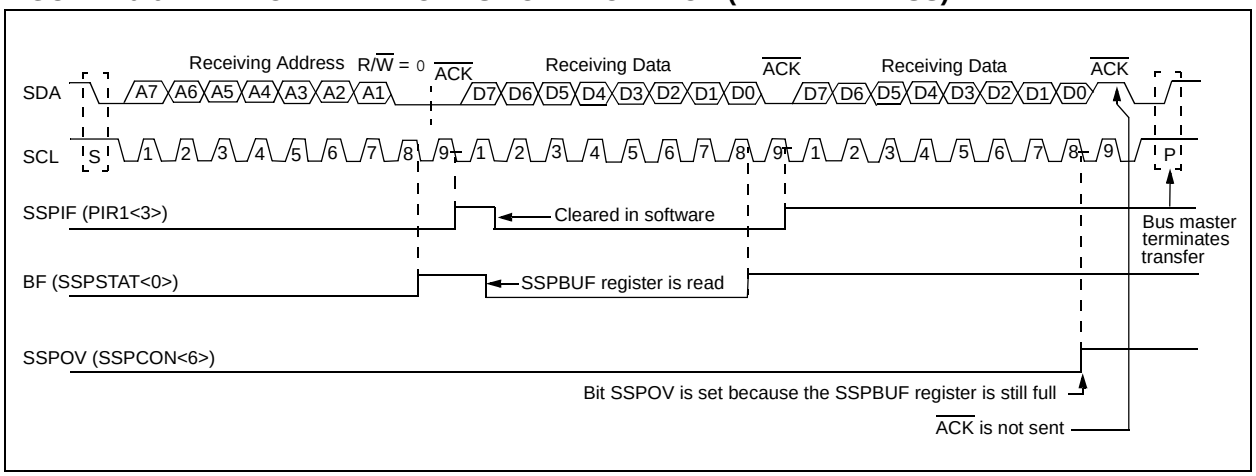
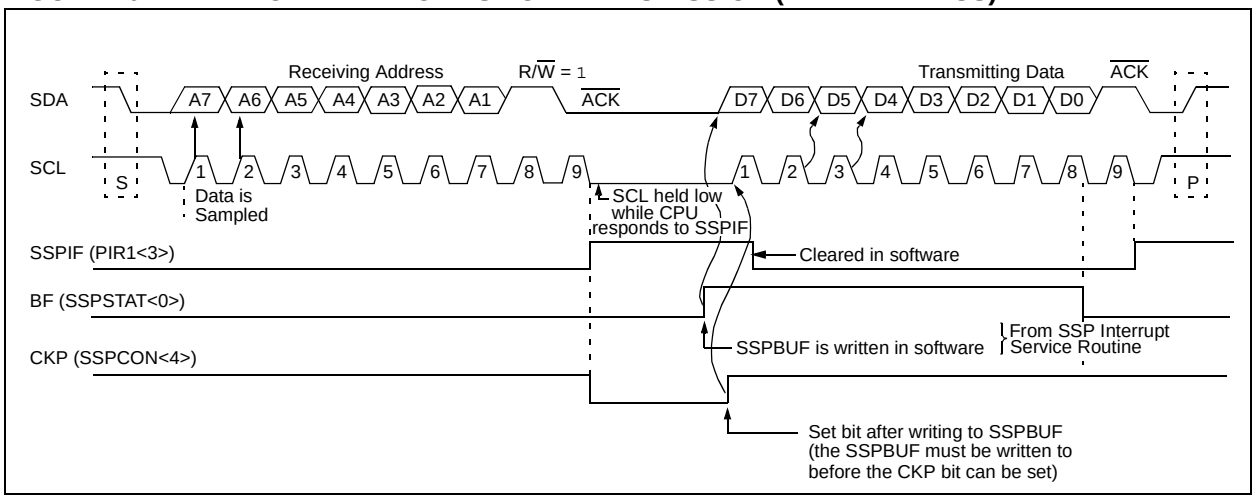



**FIGURE 10-7: I²C™ WAVEFORMS FOR TRANSMISSION (7-BIT ADDRESS)**





© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

### 10.3.2 MASTER MODE OPERATION

Master mode operation is supported in firmware using interrupt generation on the detection of the Start and Stop conditions. The Stop (P) and Start (S) bits are cleared from a Reset or when the SSP module is disabled. The Stop (P) and Start (S) bits will toggle based on the Start and Stop conditions. Control of the $I^2C$ bus may be taken when the P bit is set or the bus is Idle and both the S and P bits are clear.

In Master mode operation, the SCL and SDA lines are manipulated in firmware by clearing the corresponding TRISB<4,1> bit(s). The output level is always low, irrespective of the value(s) in PORTB<4,1>. So when transmitting data, a '1' data bit must have the TRISB<1> bit set (input) and a '0' data bit must have the TRISB<1> bit cleared (output). The same scenario is true for the SCL line with the TRISB<4> bit. Pull-up resistors must be provided externally to the SCL and SDA pins for proper operation of the $I^2C$ module.

The following events will cause the SSP Interrupt Flag bit, SSPIF, to be set (SSP interrupt if enabled):

- Start condition
- Stop condition
- Data transfer byte transmitted/received

Master mode operation can be done with either the Slave mode Idle (SSPM3:SSPM0 = 1011) or with the Slave mode active. When both Master mode operation and Slave modes are used, the software needs to differentiate the source(s) of the interrupt.

For more information on Master mode operation, see *AN554, "Software Implementation of $I^2C$™ Bus Master"* (DS00554).

### 10.3.3 MULTI-MASTER MODE OPERATION

In Multi-Master mode operation, the interrupt generation on the detection of the Start and Stop conditions allows the determination of when the bus is free. The Stop (P) and Start (S) bits are cleared from a Reset or when the SSP module is disabled. The Stop (P) and Start (S) bits will toggle based on the Start and Stop conditions. Control of the $I^2C$ bus may be taken when bit P (SSPSTAT<4>) is set or the bus is Idle and both the S and P bits clear. When the bus is busy, enabling the SSP interrupt will generate the interrupt when the Stop condition occurs.

In Multi-Master mode operation, the SDA line must be monitored to see if the signal level is the expected output level. This check only needs to be done when a high level is output. If a high level is expected and a low level is present, the device needs to release the SDA and SCL lines (set TRISB<4,1>). There are two stages where this arbitration can be lost:

- Address Transfer
- Data Transfer

When the slave logic is enabled, the Slave device continues to receive. If arbitration was lost during the address transfer stage, communication to the device may be in progress. If addressed, an $\overline{ACK}$ pulse will be generated. If arbitration was lost during the data transfer stage, the device will need to retransfer the data at a later time.

For more information on Multi-Master mode operation, see *AN578, "Use of the SSP Module in the $I^2C$™ Multi-Master Environment"* (DS00578).

**TABLE 10-3: REGISTERS ASSOCIATED WITH $I^2C$™ OPERATION**

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR, BOR | Value on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0Bh, 8Bh, 10Bh,18Bh | INTCON | GIE | PEIE | TMR0IE | INTE | RBIE | TMR0IF | INTF | RBIF | 0000 000x | 0000 000u |
| 0Ch | PIR1 | — | ADIF | — | — | SSPIF | CCP1IF | TMR2IF | TMR1IF | -0-- 0000 | -0-- 0000 |
| 8Ch | PIE1 | — | ADIE | — | — | SSPIE | CCP1IE | TMR2IE | TMR1IE | -0-- 0000 | -0-- 0000 |
| 13h | SSPBUF | Synchronous Serial Port Receive Buffer/Transmit Register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 93h | SSPADD | Synchronous Serial Port ($I^2C$™ mode) Address Register | | | | | | | | 0000 0000 | 0000 0000 |
| 14h | SSPCON | WCOL | SSPOV | SSPEN | CKP | SSPM3 | SSPM2 | SSPM1 | SSPM0 | 0000 0000 | 0000 0000 |
| 94h | SSPSTAT | SMP(1) | CKE(1) | D/$\overline{A}$ | P | S | R/$\overline{W}$ | UA | BF | 0000 0000 | 0000 0000 |
| 86h | TRISB | PORTB Data Direction Register | | | | | | | | 1111 1111 | 1111 1111 |

**Legend:** x = unknown, u = unchanged, - = unimplemented locations read as '0'.
Shaded cells are not used by SSP module in SPI™ mode.

**Note 1:** Maintain these bits clear in $I^2C$ mode.



Downloaded from Arrow.com.

**NOTES:**


© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

## 11.0 ANALOG-TO-DIGITAL CONVERTER (A/D) MODULE

The Analog-to-Digital (A/D) converter module has five inputs for 18/20 pin devices.

The conversion of an analog input signal results in a corresponding 10-bit digital number. The A/D module has a high and low-voltage reference input that is software selectable to some combination of VDD, VSS, RA2 or RA3.

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.

The A/D module has four registers:

- A/D Result High Register (ADRESH)
- A/D Result Low Register (ADRESL)
- A/D Control Register 0 (ADCON0)
- A/D Control Register 1 (ADCON1)

The ADCON0 register, shown in Register 11-1, controls the operation of the A/D module. The ADCON1 register, shown in Register 11-2, 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/Os.

Additional information on using the A/D module can be found in the *"PICmicro® Mid-Range MCU Family Reference Manual"* (DS33023).

**REGISTER 11-1: ADCON0: A/D CONTROL REGISTER 0 (ADDRESS 1Fh)**

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | U-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| ADCS1 | ADCS0 | CHS2 | CHS1 | CHS0 | GO/$\overline{\text{DONE}}$ | — | ADON |
| bit 7 | | | | | | | bit 0 |

bit 7-6 **ADCS1:ADCS0:** A/D Conversion Clock Select bits

If ADCS2 = 0:
00 = FOSC/2
01 = FOSC/8
10 = FOSC/32
11 = FRC (clock derived from the internal A/D module RC oscillator)

If ADCS2 = 1:
00 = FOSC/4
01 = FOSC/16
10 = FOSC/64
11 = FRC (clock derived from the internal A/D module 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)
100 = Channel 4 (RA4/AN4)

bit 2 **GO/$\overline{\text{DONE}}$:** 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 **Unimplemented:** Read as '0'

bit 0 **ADON:** A/D On bit

1 = A/D converter module is operating
0 = A/D converter module is shut-off and consumes no operating current

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



Downloaded from Arrow.com.

**REGISTER 11-2: ADCON1: A/D CONTROL REGISTER 1 (ADDRESS 9Fh)**

| R/W-0 | R/W-0 | U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| ADFM | ADCS2 | — | — | PCFG3 | PCFG2 | PCFG1 | PCFG0 |
| bit 7 | | | | | | | bit 0 |

bit 7 **ADFM:** A/D Result Format Select bit
1 = Right justified, 6 Most Significant bits of ADRESH are read as '0'
0 = Left justified, 6 Least Significant bits of ADRESL are read as '0'

bit 6 **ADCS2:** A/D Clock Divide by 2 Select bit
1 = A/D clock source is divided by 2 when system clock is used
0 = Disabled

bit 5-4 **Unimplemented:** Read as '0'

bit 3-0 **PCFG<3:0>:** A/D Port Configuration Control bits

| PCFG | AN4 | AN3 | AN2 | AN1 | AN0 | VREF+ | VREF- | C/R |
|---|---|---|---|---|---|---|---|---|
| 0000 | A | A | A | A | A | AVDD | AVSS | 5/0 |
| 0001 | A | VREF+ | A | A | A | AN3 | AVSS | 4/1 |
| 0010 | A | A | A | A | A | AVDD | AVSS | 5/0 |
| 0011 | A | VREF+ | A | A | A | AN3 | AVSS | 4/1 |
| 0100 | D | A | D | A | A | AVDD | AVSS | 3/0 |
| 0101 | D | VREF+ | D | A | A | AN3 | AVSS | 2/1 |
| 011x | D | D | D | D | D | AVDD | AVSS | 0/0 |
| 1000 | A | VREF+ | VREF- | A | A | AN3 | AN2 | 3/2 |
| 1001 | A | A | A | A | A | AVDD | AVSS | 5/0 |
| 1010 | A | VREF+ | A | A | A | AN3 | AVSS | 4/1 |
| 1011 | A | VREF+ | VREF- | A | A | AN3 | AN2 | 3/2 |
| 1100 | A | VREF+ | VREF- | A | A | AN3 | AN2 | 3/2 |
| 1101 | D | VREF+ | VREF- | A | A | AN3 | AN2 | 2/2 |
| 1110 | D | D | D | D | A | AVDD | AVSS | 1/0 |
| 1111 | D | VREF+ | VREF- | D | A | AN3 | AN2 | 1/2 |

A = Analog input D = Digital I/O
C/R = Number of analog input channels/Number of A/D voltage references

| **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.

Downloaded from Arrow.com.

The ADRESH:ADRESL registers contain the result of the A/D conversion. When the A/D conversion is complete, the result is loaded into the A/D Result register pair, the GO/DONE bit (ADCON0<2>) is cleared and A/D Interrupt Flag bit, ADIF, is set. The block diagram of the A/D module is shown in Figure 11-1.

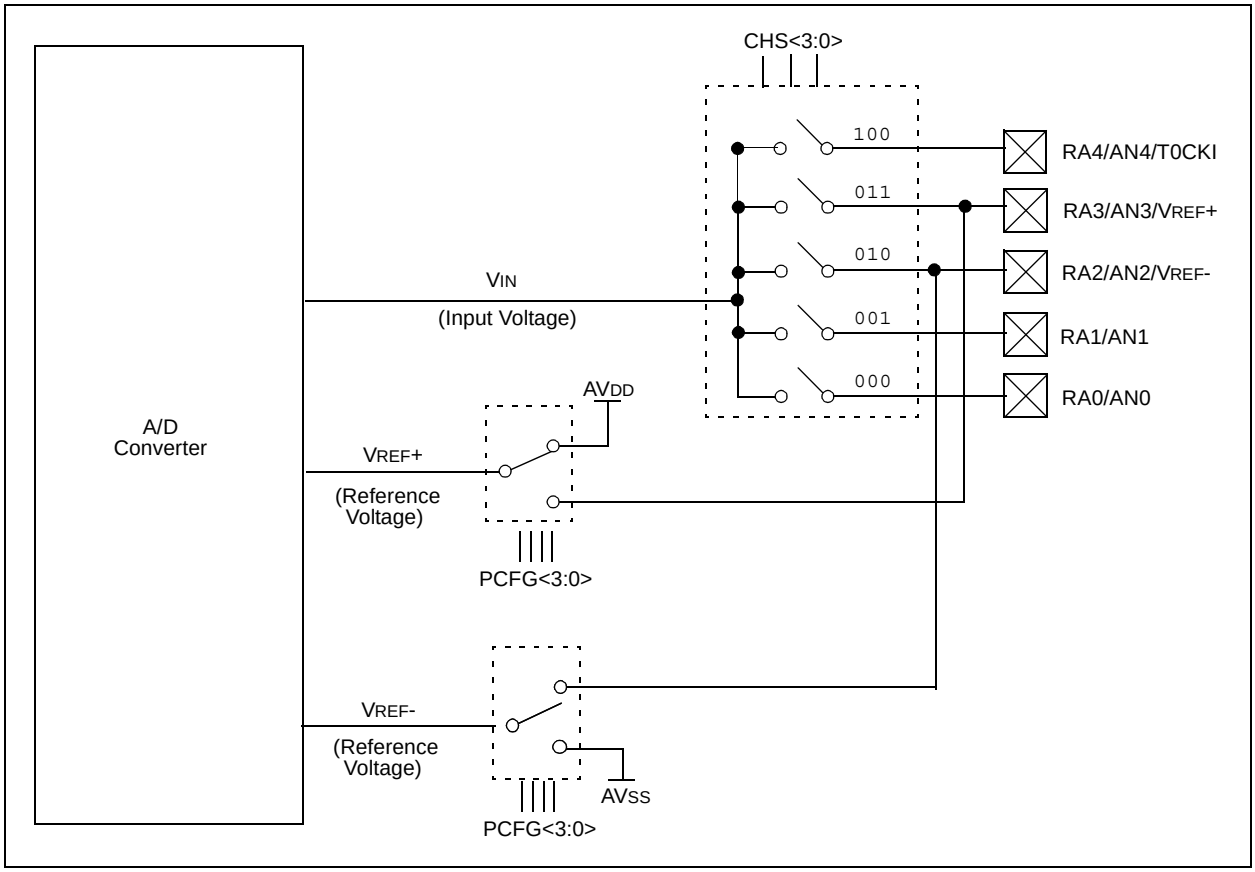
After the A/D module has been configured as desired, the selected channel must be acquired before the conversion is started. The analog input channels must have their corresponding TRIS bits selected as inputs.

To determine sample time, see **Section 11.1 "A/D Acquisition Requirements"**. After this sample time has elapsed, the A/D conversion can be started.

These 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 (with interrupts disabled); OR
   - Waiting for the A/D interrupt
6. Read A/D Result register pair (ADRESH:ADRESL), clear bit ADIF if required.
7. For next conversion, go to step 1 or step 2 as required. The A/D conversion time per bit is defined as TAD. A minimum wait of 2 TAD is required before the next acquisition starts.

**FIGURE 11-1: A/D BLOCK DIAGRAM**






Downloaded from Arrow.com.

## 11.1 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 11-2. 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), see Figure 11-2. **The maximum recommended impedance for analog sources is 2.5 kΩ.** As the impedance is decreased, the acquisition time may be decreased. After the analog input channel is selected (changed), this acquisition must be done before the conversion can be started.

To calculate the minimum acquisition time, Equation 11-1 may be used. This equation assumes that 1/2 LSb error is used (1024 steps for the A/D). The 1/2 LSb error is the maximum error allowed for the A/D to meet its specified resolution.

To calculate the minimum acquisition time, TACQ, see the *"PICmicro® Mid-Range MCU Family Reference Manual"* (DS33023).

**EQUATION 11-1: ACQUISITION TIME**

TACQ = Amplifier Settling Time + Hold Capacitor Charging Time + Temperature Coefficient

= TAMP + TC + TCOFF
= 2 µs + TC + [(Temperature – 25°C)(0.05 µs/°C)]
TC = CHOLD (RIC + RSS + RS) In(1/2047)
= -120 pF (1 kΩ + 7 kΩ + 10 kΩ) In(0.0004885)
= 16.47 µs
TACQ = 2 µs + 16.47 µs + [(50°C – 25°C)(0.05 µs/°C)
= 19.72 µs

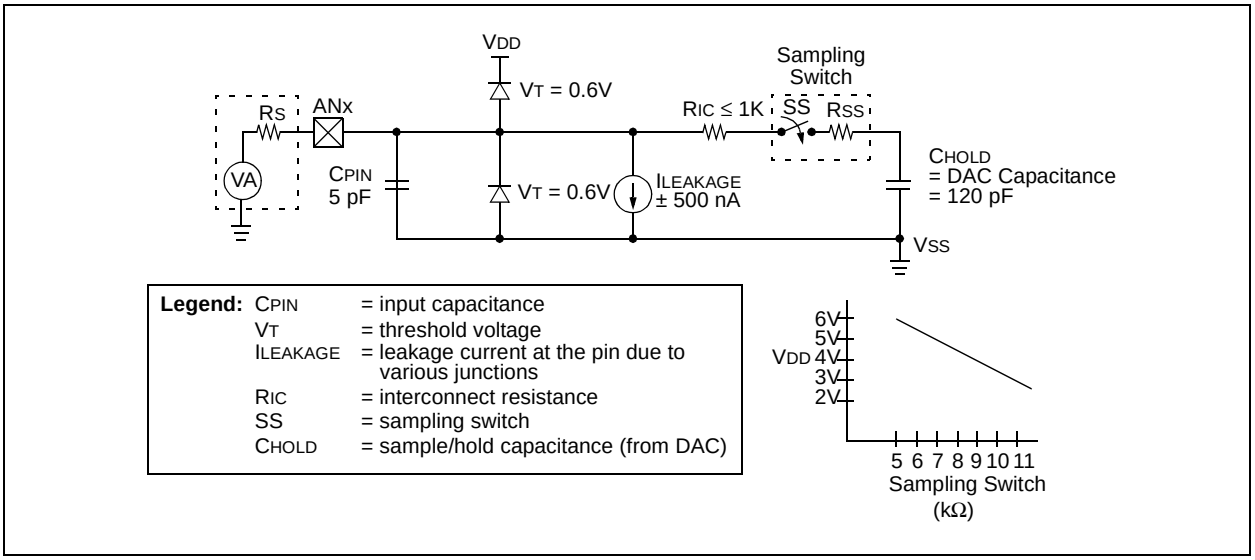
**Note 1:** The reference voltage (VREF) has no effect on the equation since it cancels itself out.

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

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

**4:** After a conversion has completed, a 2.0 TAD delay must complete before acquisition can begin again. During this time, the holding capacitor is not connected to the selected A/D input channel.

**FIGURE 11-2: ANALOG INPUT MODEL**





© 2003 Microchip Technology Inc.

Downloaded from Arrow.com.

## 11.2 Selecting the A/D Conversion Clock

The A/D conversion time per bit is defined as TAD. The A/D conversion requires 9.0 TAD per 10-bit conversion. The source of the A/D conversion clock is software selectable. The seven possible options for TAD are:

- 2 TOSC
- 4 TOSC
- 8 TOSC
- 16 TOSC
- 32 TOSC
- 64 TOSC
- Internal A/D module RC oscillator (2-6 µs)

For correct A/D conversions, the A/D conversion clock (TAD) must be selected to ensure a minimum TAD time as small as possible, but no less than 1.6 µs and not greater than 6.4 µs.

Table 11-1 shows the resultant TAD times derived from the device operating frequencies and the A/D clock source selected.

## 11.3 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 (VOH or VOL) will be converted.

The A/D operation is independent of the state of the CHS<2:0> bits and the TRIS bits.

**Note 1:** When reading the Port register, all pins configured as analog input channels will read as cleared (a low level). Pins configured as digital inputs will convert an analog input. Analog levels on a digitally configured input will not affect the conversion accuracy.

**2:** Analog levels on any pin that is defined as a digital input (including the AN4:AN0 pins) may cause the input buffer to consume current out of the device specification.

**TABLE 11-1: TAD VS. MAXIMUM DEVICE OPERATING FREQUENCIES (STANDARD DEVICES (F))**

| AD Clock Source (TAD) | | | Maximum Device Frequency |
|---|---|---|---|
| Operation | ADCS<2> | ADCS<1:0> | |
| 2 TOSC | 0 | 00 | 1.25 MHz |
| 4 TOSC | 1 | 00 | 2.5 MHz |
| 8 TOSC | 0 | 01 | 5 MHz |
| 16 TOSC | 1 | 01 | 10 MHz |
| 32 TOSC | 0 | 10 | 20 MHz |
| 64 TOSC | 1 | 10 | 20 MHz |
| RC(1,2,3) | X | 11 | **(Note 1)** |

**Note 1:** The RC source has a typical TAD time of 4 µs but can vary between 2-6 µs.

**2:** When the device frequencies are greater than 1 MHz, the RC A/D conversion clock source is only recommended for Sleep operation.

**3:** For extended voltage devices (LF), please refer to **Section 15.0 "Electrical Characteristics"**.



Downloaded from Arrow.com.

## 11.4 A/D Conversions

Clearing the GO/$\overline{\text{DONE}}$ bit during a conversion will abort the current conversion. The A/D Result register pair will NOT be updated with the partially completed A/D conversion sample. That is, the ADRESH:ADRESL registers will continue to contain the value of the last completed conversion (or the last value written to the ADRESH:ADRESL registers). After the A/D conversion is aborted, a 2-TAD wait is required before the next acquisition is started. After this 2-TAD wait, acquisition on the selected channel is automatically started. The GO/$\overline{\text{DONE}}$ bit can then be set to start the conversion.

In Figure 11-3, after the GO bit is set, the first time segment has a minimum of TCY and a maximum of TAD.

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

### 11.4.1 A/D RESULT REGISTERS

The ADRESH:ADRESL register pair is the location where the 10-bit A/D result is loaded at the completion of the A/D conversion. This register pair is 16 bits wide. The A/D module gives the flexibility to left or right justify the 10-bit result in the 16-bit result register. The A/D Format Select bit (ADFM) controls this justification. Figure 11-4 shows the operation of the A/D result justification. The extra bits are loaded with '0's. When an A/D result will not overwrite these locations (A/D disable), these registers may be used as two general purpose 8-bit registers.

**FIGURE 11-3: A/D CONVERSION TAD CYCLES**

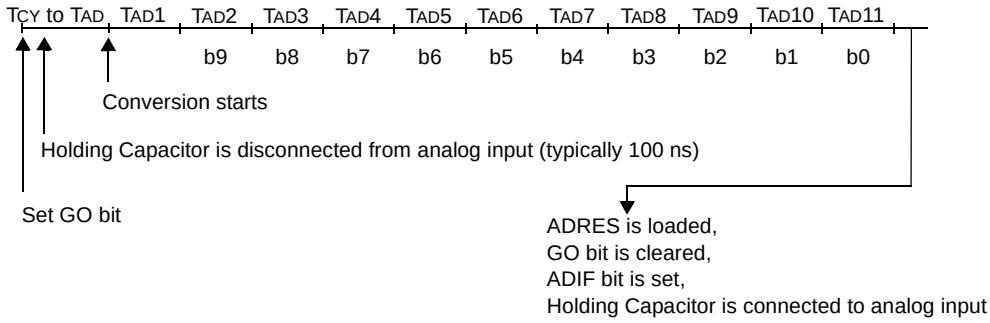
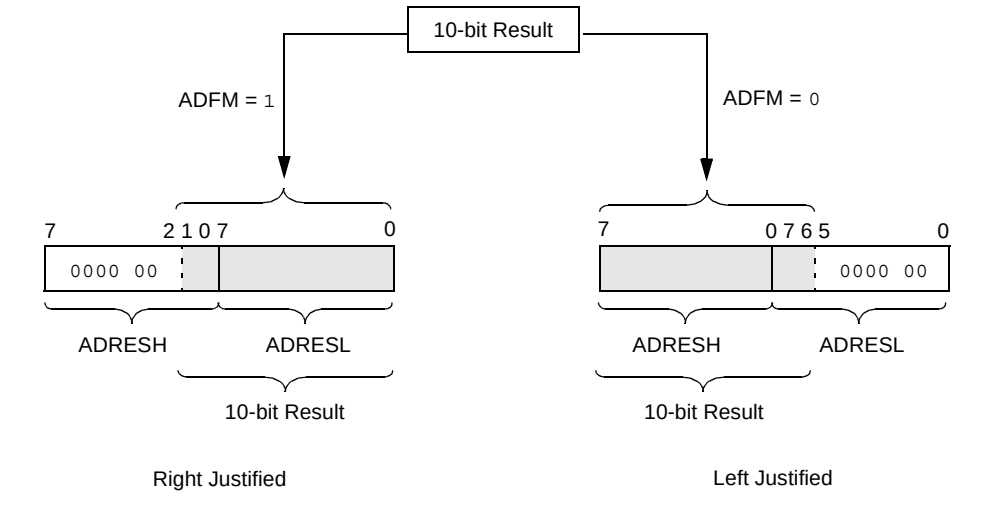



**FIGURE 11-4: A/D RESULT JUSTIFICATION**





© 2003 Microchip Technology Inc.

Downloaded from Arrow.com.

## 11.5 A/D Operation During Sleep

The A/D module can operate during Sleep mode. This requires that the A/D clock source be set to RC (ADCS1:ADCS0 = 11). When the RC clock source is selected, the A/D module waits one instruction cycle before starting the conversion. This allows the SLEEP instruction to be executed which eliminates all digital switching noise from the conversion. When the conversion is completed, the GO/DONE bit will be cleared and the result loaded into the ADRES register. If the A/D interrupt is enabled, the device will wake-up from Sleep. If the A/D interrupt is not enabled, the A/D module will then be turned off, although the ADON bit will remain set.

When the A/D clock source is another clock option (not RC), a SLEEP instruction will cause the present conversion to be aborted and the A/D module to be turned off, though the ADON bit will remain set.

Turning off the A/D places the A/D module in its lowest current consumption state.

| **Note:** | For the A/D module to operate in Sleep, the A/D clock source must be set to RC (ADCS1:ADCS0 = 11). To perform an A/D conversion in Sleep, ensure the SLEEP instruction immediately follows the instruction that sets the GO/DONE bit. |
|---|---|

## 11.6 Effects of a Reset

A device Reset forces all registers to their Reset state. The A/D module is disabled and any conversion in progress is aborted. All A/D input pins are configured as analog inputs.

The value that is in the ADRESH:ADRESL registers is not modified for a Power-on Reset. The ADRESH:ADRESL registers will contain unknown data after a Power-on Reset.

## 11.7 Use of the CCP Trigger

An A/D conversion can be started by the "special event trigger" of the CCP 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 ADRESH:ADRESL 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 11-2: REGISTERS/BITS ASSOCIATED WITH A/D**

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR, BOR | Value on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0Bh,8Bh 10Bh,18Bh | INTCON | GIE | PEIE | TMR0IE | INTE | RBIE | TMR0IF | INTF | RBIF | 0000 000x | 0000 000u |
| 0Ch | PIR1 | — | ADIF | — | — | SSPIF | CCP1IF | TMR2IF | TMR1IF | -0-- 0000 | -0-- 0000 |
| 8Ch | PIE1 | — | ADIE | — | — | SSPIE | CCP1IE | TMR2IE | TMR1IE | -0-- 0000 | -0-- 0000 |
| 1Eh | ADRESH | A/D Result Register High Byte | | | | | | | | xxxx xxxx | uuuu uuuu |
| 9Eh | ADRESL | A/D Result Register Low Byte | | | | | | | | xxxx xxxx | uuuu uuuu |
| 1Fh | ADCON0 | ADCS1 | ADCS0 | CHS2 | CHS1 | CHS0 | GO/DONE | — | ADON | 0000 00-0 | 0000 00-0 |
| 9Fh | ADCON1 | ADFM | ADCS2 | — | — | PCFG3 | PCFG2 | PCFG1 | PCFG0 | 00-- 0000 | 00-- 0000 |
| 05h | PORTA | RA7 | RA6 | RA5 | RA4 | RA3 | RA2 | RA1 | RA0 | xxx0 0000 | uuu0 0000 |
| 85h | TRISA | TRISA7 | TRISA6 | TRISA5 | PORTA Data Direction Register | | | | | 1111 1111 | 1111 1111 |

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



Downloaded from Arrow.com.

**NOTES:**


© 2003 Microchip Technology Inc.

Downloaded from Arrow.com.

## 12.0 SPECIAL FEATURES OF THE CPU

These devices have 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:

- 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

There are two timers that offer necessary delays on power-up. One is the Oscillator Start-up Timer (OST), intended to keep the chip in Reset until the crystal oscillator is stable. The other is the Power-up Timer (PWRT) which provides a fixed delay of 72 ms (nominal) on power-up only. It is designed to keep the part in Reset while the power supply stabilizes and is enabled or disabled using a configuration bit. 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. Configuration bits are used to select the desired oscillator mode.

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

### 12.1 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 which can be accessed only during programming.



Downloaded from Arrow.com.

**REGISTER 12-1: CONFIGURATION WORD (ADDRESS 2007h)(1)**

| R/P-1 | R/P-1 | R/P-1 | R/P-1 | R/P-1 | R/P-1 | R/P-1 | R/P-1 | R/P-1 | R/P-1 | R/P-1 | R/P-1 | R/P-1 | R/P-1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CP | CCPMX | DEBUG | WRT1 | WRT0 | CPD | LVP | BOREN | MCLRE | FOSC2 | $\overline{\text{PWRTEN}}$ | WDTEN | FOSC1 | FOSC0 |
| bit 13 | | | | | | | | | | | | | bit 0 |

bit 13 **CP:** Flash Program Memory Code Protection bit
1 = Code protection off
0 = All memory locations code-protected

bit 12 **CCPMX:** CCP1 Pin Selection bit
1 = CCP1 function on RB2
0 = CCP1 function on RB3

bit 11 **DEBUG:** In-Circuit Debugger Mode bit
1 = In-Circuit Debugger disabled, RB6 and RB7 are general purpose I/O pins
0 = In-Circuit Debugger enabled, RB6 and RB7 are dedicated to the debugger

bit 10-9 **WRT1:WRT0:** Flash Program Memory Write Enable bits
For PIC16F818:
11 = Write protection off
10 = 000h to 01FF write-protected, 0200 to 03FF may be modified by EECON control
01 = 000h to 03FF write-protected
For PIC16F819:
11 = Write protection off
10 = 0000h to 01FFh write-protected, 0200h to 07FFh may be modified by EECON control
01 = 0000h to 03FFh write-protected, 0400h to 07FFh may be modified by EECON control
00 = 0000h to 05FFh write-protected, 0600h to 07FFh may be modified by EECON control

bit 8 **CPD:** Data EE Memory Code Protection bit
1 = Code protection off
0 = Data EE memory locations code-protected

bit 7 **LVP:** Low-Voltage Programming Enable bit
1 = RB3/PGM pin has PGM function, Low-Voltage Programming enabled
0 = RB3/PGM pin has digital I/O function, HV on $\overline{\text{MCLR}}$ must be used for programming

bit 6 **BOREN:** Brown-out Reset Enable bit
1 = BOR enabled
0 = BOR disabled

bit 5 **MCLRE:** RA5/$\overline{\text{MCLR}}$/$V_{PP}$ Pin Function Select bit
1 = RA5/$\overline{\text{MCLR}}$/$V_{PP}$ pin function is $\overline{\text{MCLR}}$
0 = RA5/$\overline{\text{MCLR}}$/$V_{PP}$ pin function is digital I/O, $\overline{\text{MCLR}}$ internally tied to $V_{DD}$

bit 3 **$\overline{\text{PWRTEN}}$:** Power-up Timer Enable bit
1 = PWRT disabled
0 = PWRT enabled

bit 2 **WDTEN:** Watchdog Timer Enable bit
1 = WDT enabled
0 = WDT disabled

bit 4, 1-0 **FOSC2:FOSC0:** Oscillator Selection bits
111 = EXTRC oscillator; CLKO function on RA6/OSC2/CLKO pin
110 = EXTRC oscillator; port I/O function on RA6/OSC2/CLKO pin
101 = INTRC oscillator; CLKO function on RA6/OSC2/CLKO pin and port I/O function on RA7/OSC1/CLKI pin
100 = INTRC oscillator; port I/O function on both RA6/OSC2/CLKO pin and RA7/OSC1/CLKI pin
011 = EXTCLK; port I/O function on RA6/OSC2/CLKO pin
010 = HS oscillator
001 = XT oscillator
000 = LP oscillator

**Note 1:** The erased (unprogrammed) value of the Configuration Word is 3FFFh.

| **Legend:** | | |
|---|---|---|
| R = Readable bit | P = Programmable bit | U = Unimplemented bit, read as '1' |
| -n = Value when device is unprogrammed | | u = Unchanged from programmed state |


© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

## 12.2 Reset

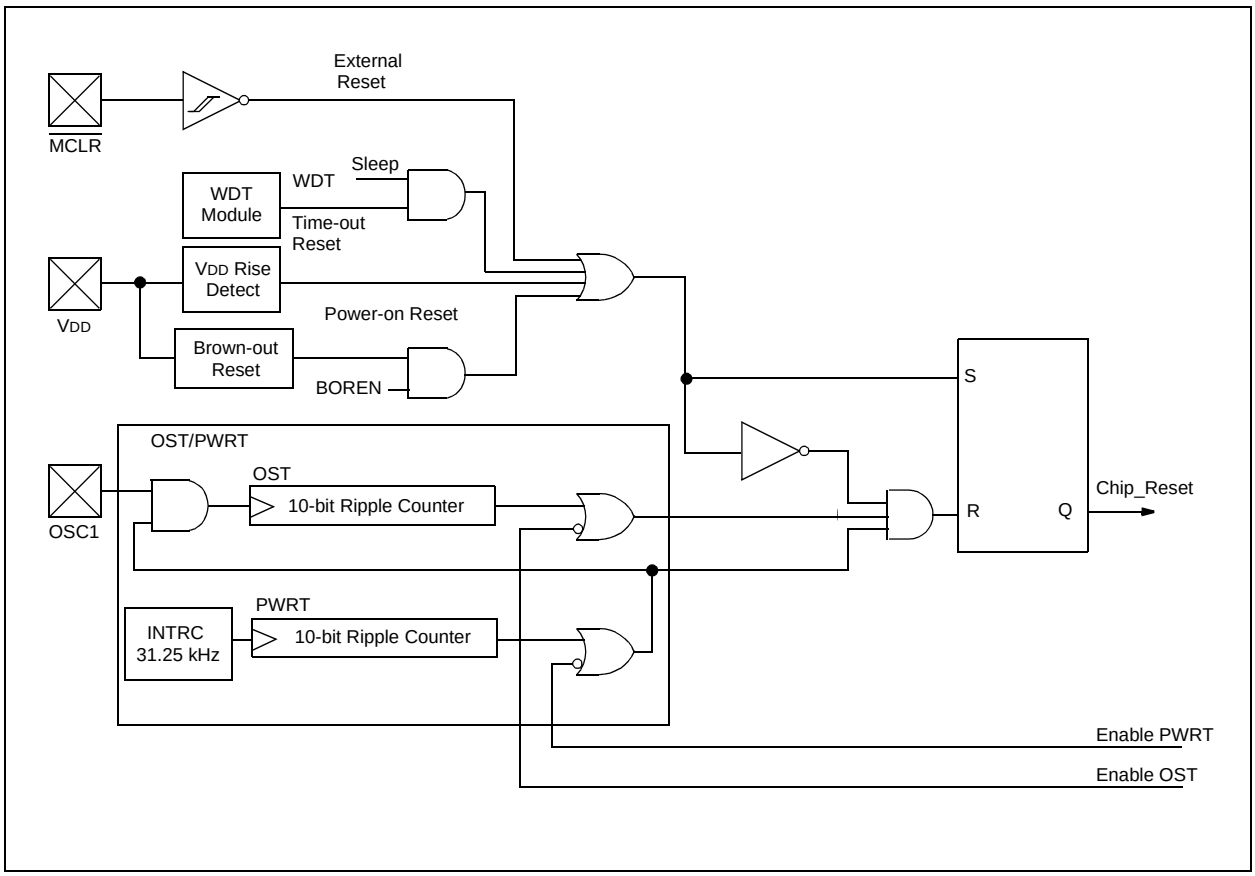
The PIC16F818/819 differentiates between various kinds of Reset:

- Power-on Reset (POR)
- MCLR Reset during normal operation
- MCLR Reset during Sleep
- WDT Reset during normal operation
- WDT wake-up during Sleep
- Brown-out Reset (BOR)

Some registers are not affected in any Reset condition. Their status is unknown on POR and unchanged in any other Reset. Most other registers are reset to a "Reset state" on Power-on Reset (POR), on the MCLR and WDT Reset, on MCLR Reset during Sleep and Brown-out Reset (BOR). They are not affected by a WDT wake-up which is viewed as the resumption of normal operation. The TO and PD bits are set or cleared differently in different Reset situations as indicated in Table 12-3. These bits are used in software to determine the nature of the Reset. Upon a POR, BOR or wake-up from Sleep, the CPU requires approximately 5-10 µs to become ready for code execution. This delay runs in parallel with any other timers. See Table 12-4 for a full description of Reset states of all registers.

A simplified block diagram of the on-chip Reset circuit is shown in Figure 12-1.

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






Downloaded from Arrow.com.

## 12.3 MCLR

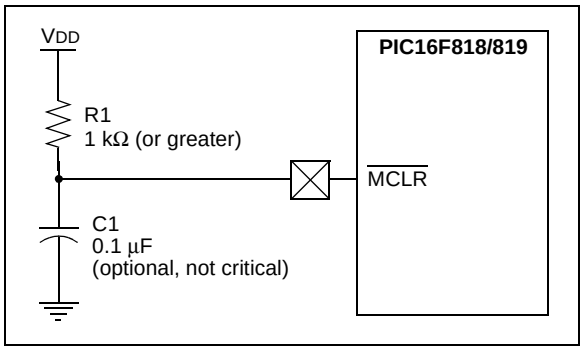
PIC16F818/819 device has a noise filter in the MCLR Reset path. The filter will detect and ignore small pulses.

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

The behavior of the ESD protection on the MCLR pin has been altered from previous devices of this family. Voltages applied to the pin that exceed its specification can result in both MCLR and excessive current beyond the device specification during the ESD event. For this reason, Microchip recommends that the MCLR pin no longer be tied directly to VDD. The use of an RC network, as shown in Figure 12-2, is suggested.

The RA5/MCLR/VPP pin can be configured for MCLR (default) or as an I/O pin (RA5). This is configured through the MCLRE bit in the Configuration Word register.

**FIGURE 12-2: RECOMMENDED MCLR CIRCUIT**




## 12.4 Power-on Reset (POR)

A Power-on Reset pulse is generated on-chip when VDD rise is detected (in the range of 1.2V-1.7V). To take advantage of the POR, tie the MCLR pin to VDD as described in **Section 12.3 "MCLR**". A maximum rise time for VDD is specified. See **Section 15.0 "Electrical Characteristics"** for details.

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. For more information, see Application Note *AN607, "Power-up Trouble Shooting"* (DS00607).

## 12.5 Power-up Timer (PWRT)

The Power-up Timer (PWRT) of the PIC16F818/819 is a counter that uses the INTRC oscillator as the clock input. This yields a count of 72 ms. While the PWRT is counting, the device is held in Reset.

The power-up time delay depends on the INTRC and will vary from chip-to-chip due to temperature and process variation. See DC parameter #33 for details.

The PWRT is enabled by clearing configuration bit, PWRTEN.

## 12.6 Oscillator Start-up Timer (OST)

The Oscillator Start-up Timer (OST) provides 1024 oscillator cycles (from OSC1 input) delay after the PWRT delay is over (if enabled). This helps to ensure that the crystal oscillator or resonator has started and stabilized.

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

## 12.7 Brown-out Reset (BOR)

The configuration bit, BOREN, can enable or disable the Brown-out Reset circuit. If VDD falls below VBOR (parameter #D005, about 4V) for longer than TBOR (parameter #35, about 100 µs), the brown-out situation will reset the device. If VDD falls below VBOR for less than TBOR, a Reset may not occur.

Once the brown-out occurs, the device will remain in Brown-out Reset until VDD rises above VBOR. The Power-up Timer (if enabled) will keep the device in Reset for TPWRT (parameter #33, about 72 ms). If VDD should fall below VBOR during TPWRT, the Brown-out Reset process will restart when VDD rises above VBOR with the Power-up Timer Reset. Unlike previous PIC16 devices, the PWRT is no longer automatically enabled when the Brown-out Reset circuit is enabled. The PWRTEN and BOREN configuration bits are independent of each other.

## 12.8 Time-out Sequence

On power-up, the time-out sequence is as follows: the PWRT delay starts (if enabled) when a POR occurs. Then, OST starts counting 1024 oscillator cycles when PWRT ends (LP, XT, HS). When the OST ends, the device comes out of Reset.

If MCLR is kept low long enough, all delays will expire. Bringing MCLR high will begin execution immediately. This is useful for testing purposes or to synchronize more than one PIC16F818/819 device operating in parallel.

Table 12-3 shows the Reset conditions for the Status, PCON and PC registers, while Table 12-4 shows the Reset conditions for all the registers.


© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

## 12.9 Power Control/Status Register (PCON)

The Power Control/Status register, PCON, has two bits to indicate the type of Reset that last occurred.

Bit 0 is Brown-out Reset Status bit, $\overline{BOR}$. Bit $\overline{BOR}$ is unknown on a Power-on Reset. It must then be set by the user and checked on subsequent Resets to see if bit $\overline{BOR}$ cleared, indicating a Brown-out Reset occurred. When the Brown-out Reset is disabled, the state of the $\overline{BOR}$ bit is unpredictable.

Bit 1 is Power-on Reset Status bit, $\overline{POR}$. It is cleared on a Power-on Reset and unaffected otherwise. The user must set this bit following a Power-on Reset.

**TABLE 12-1: TIME-OUT IN VARIOUS SITUATIONS**

| Oscillator Configuration | Power-up | | Brown-out Reset | | Wake-up from Sleep |
|---|---|---|---|---|---|
| | $\overline{PWRTE}$ = 0 | $\overline{PWRTE}$ = 1 | $\overline{PWRTE}$ = 0 | $\overline{PWRTE}$ = 1 | |
| XT, HS, LP | TPWRT + 1024 • TOSC | 1024 • TOSC | TPWRT + 1024 • TOSC | 1024 • TOSC | 1024 • TOSC |
| EXTRC, EXTCLK, INTRC | TPWRT | 5-10 µs(1) | TPWRT | 5-10 µs(1) | 5-10 µs(1) |

**Note 1:** CPU start-up is always invoked on POR, BOR and wake-up from Sleep.

**TABLE 12-2: STATUS BITS AND THEIR SIGNIFICANCE**

| $\overline{POR}$ | $\overline{BOR}$ | $\overline{TO}$ | $\overline{PD}$ | |
|---|---|---|---|---|
| 0 | x | 1 | 1 | Power-on Reset |
| 0 | x | 0 | x | Illegal, $\overline{TO}$ is set on $\overline{POR}$ |
| 0 | x | x | 0 | Illegal, $\overline{PD}$ is set on $\overline{POR}$ |
| 1 | 0 | 1 | 1 | Brown-out Reset |
| 1 | 1 | 0 | 1 | WDT Reset |
| 1 | 1 | 0 | 0 | WDT wake-up |
| 1 | 1 | u | u | $\overline{MCLR}$ Reset during normal operation |
| 1 | 1 | 1 | 0 | $\overline{MCLR}$ Reset during Sleep or interrupt wake-up from Sleep |

**Legend:** u = unchanged, x = unknown

**TABLE 12-3: RESET CONDITION FOR SPECIAL REGISTERS**

| Condition | Program Counter | Status Register | PCON Register |
|---|---|---|---|
| Power-on Reset | 000h | 0001 1xxx | ---- --0x |
| $\overline{MCLR}$ Reset during normal operation | 000h | 000u uuuu | ---- --uu |
| $\overline{MCLR}$ Reset during Sleep | 000h | 0001 0uuu | ---- --uu |
| WDT Reset | 000h | 0000 1uuu | ---- --uu |
| WDT wake-up | PC + 1 | uuu0 0uuu | ---- --uu |
| Brown-out Reset | 000h | 0001 1uuu | ---- --u0 |
| Interrupt wake-up from Sleep | PC + 1(1) | uuu1 0uuu | ---- --uu |

**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).



Downloaded from Arrow.com.

**TABLE 12-4: INITIALIZATION CONDITIONS FOR ALL REGISTERS**

| Register | Power-on Reset, Brown-out Reset | $\overline{\text{MCLR}}$ Reset, WDT Reset | Wake-up via WDT or Interrupt |
|---|---|---|---|
| W | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| INDF | N/A | N/A | N/A |
| TMR0 | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| PCL | 0000h | 0000h | PC + 1(2) |
| STATUS | 0001 1xxx | 000q quuu(3) | uuuq quuu(3) |
| FSR | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| PORTA | xxx0 0000 | uuu0 0000 | uuuu uuuu |
| PORTB | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| PCLATH | ---0 0000 | ---0 0000 | ---u uuuu |
| INTCON | 0000 000x | 0000 000u | uuuu uuuu(1) |
| PIR1 | -0-- 0000 | -0-- 0000 | -u-- uuuu(1) |
| PIR2 | ---0 ---- | ---0 ---- | ---u ----(1) |
| TMR1L | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| TMR1H | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| T1CON | --00 0000 | --uu uuuu | --uu uuuu |
| TMR2 | 0000 0000 | 0000 0000 | uuuu uuuu |
| T2CON | -000 0000 | -000 0000 | -uuu uuuu |
| SSPBUF | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| SSPCON | 0000 0000 | 0000 0000 | uuuu uuuu |
| CCPR1L | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| CCPR1H | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| CCP1CON | --00 0000 | --00 0000 | --uu uuuu |
| ADRESH | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| ADCON0 | 0000 00-0 | 0000 00-0 | uuuu uu-u |
| OPTION_REG | 1111 1111 | 1111 1111 | uuuu uuuu |
| TRISA | 1111 1111 | 1111 1111 | uuuu uuuu |
| TRISB | 1111 1111 | 1111 1111 | uuuu uuuu |
| PIE1 | -0-- 0000 | -0-- 0000 | -u-- uuuu |
| PIE2 | ---0 ---- | ---0 ---- | ---u ---- |
| PCON | ---- --qq | ---- --uu | ---- --uu |
| OSCCON | -000 -0-- | -000 -0-- | -uuu -u-- |
| OSCTUNE | --00 0000 | --00 0000 | --uu uuuu |
| PR2 | 1111 1111 | 1111 1111 | 1111 1111 |
| SSPADD | 0000 0000 | 0000 0000 | uuuu uuuu |
| SSPSTAT | 0000 0000 | 0000 0000 | uuuu uuuu |
| ADRESL | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| ADCON1 | 00-- 0000 | 00-- 0000 | uu-- uuuu |
| EEDATA | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| EEADR | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| EEDATH | --xx xxxx | --uu uuuu | --uu uuuu |
| EEADRH | ---- -xxx | ---- -uuu | ---- -uuu |
| EECON1 | x--x x000 | u--x u000 | u--u uuuu |
| EECON2 | ---- ---- | ---- ---- | ---- ---- |

**Legend:** u = unchanged, x = unknown, - = unimplemented bit, read as '0', q = value depends on condition, r = reserved, maintain clear

**Note 1:** One or more bits in INTCON, PIR1 and PR2 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 12-3 for Reset value for specific conditions.


© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

**FIGURE 12-3: TIME-OUT SEQUENCE ON POWER-UP ($\overline{MCLR}$ TIED TO VDD THROUGH PULL-UP RESISTOR)**

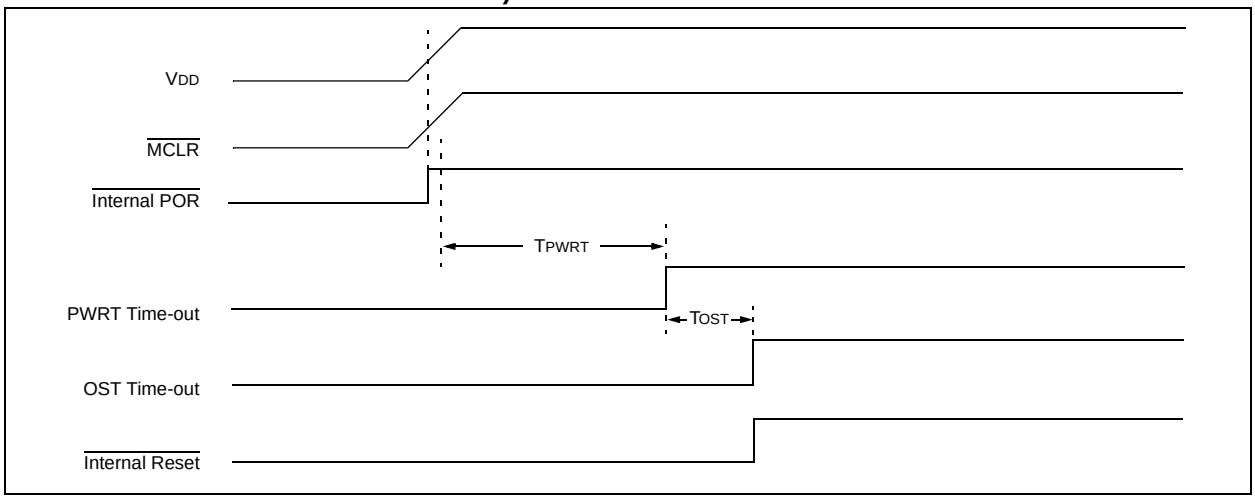



**FIGURE 12-4: TIME-OUT SEQUENCE ON POWER-UP ($\overline{MCLR}$ TIED TO VDD THROUGH RC NETWORK): CASE 1**

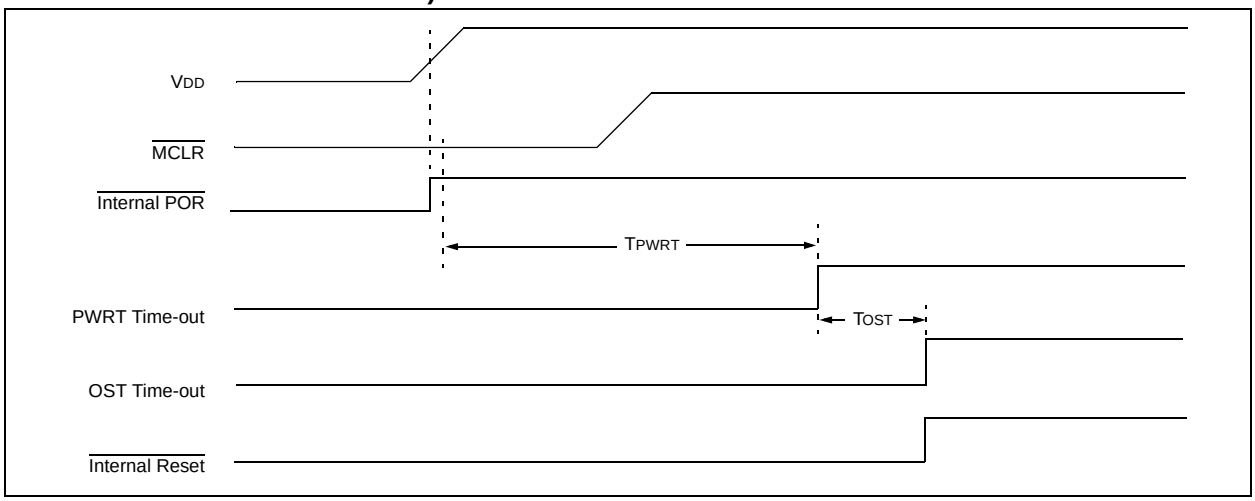
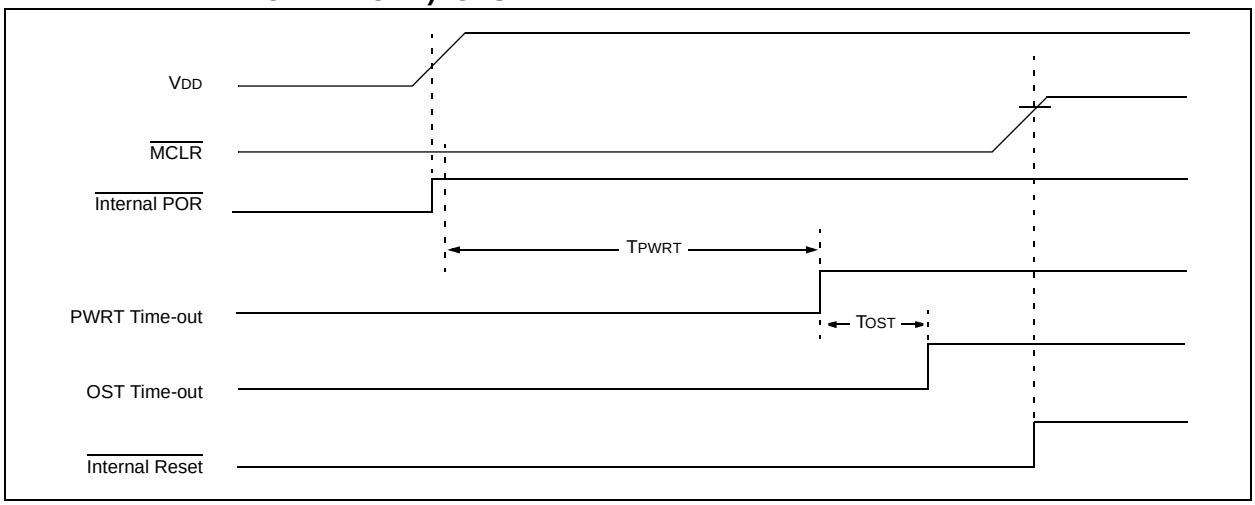


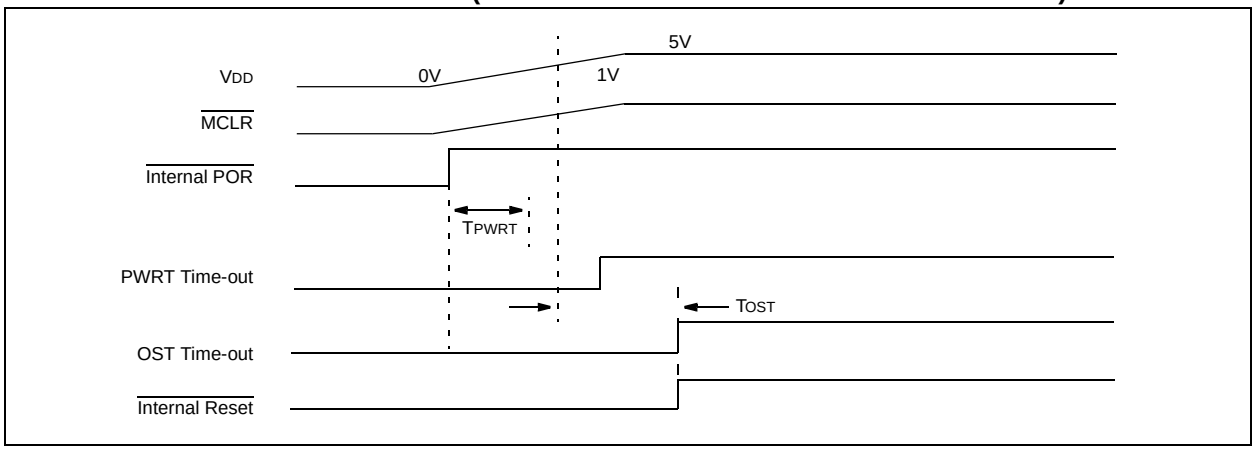

**FIGURE 12-5: TIME-OUT SEQUENCE ON POWER-UP ($\overline{MCLR}$ TIED TO VDD THROUGH RC NETWORK): CASE 2**






Downloaded from Arrow.com.

**FIGURE 12-6: SLOW RISE TIME (MCLR TIED TO VDD THROUGH RC NETWORK)**




## 12.10 Interrupts

The PIC16F818/819 has up to nine 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 (if set) all unmasked interrupts or disables (if cleared) all interrupts. 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.

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

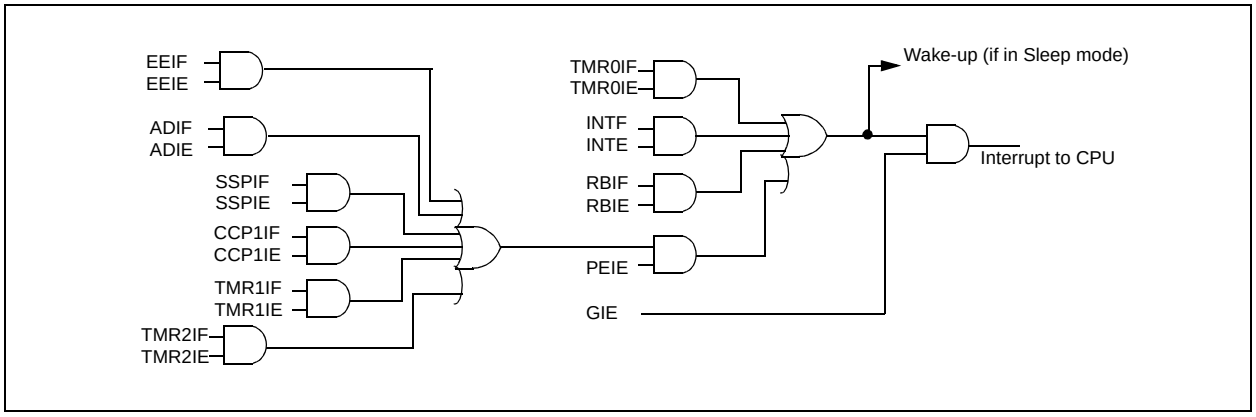
The RB0/INT pin interrupt, the RB port change interrupt and the TMR0 overflow interrupt flags are contained in the INTCON register.

The peripheral interrupt flags are contained in the Special Function Register, PIR1. The corresponding interrupt enable bits are contained in Special Function Register, PIE1 and the peripheral interrupt enable bit is contained in Special Function Register, INTCON.

When an interrupt is serviced, 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 on when the interrupt event occurs relative to the current Q cycle. 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, PEIE bit or the GIE bit.

**FIGURE 12-7: INTERRUPT LOGIC**





© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

### 12.10.1 INT INTERRUPT

External interrupt on the RB0/INT pin is edge triggered, either rising if bit INTEDG (OPTION_REG<6>) is set, or falling if the INTEDG 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 12.13 "Power-Down Mode (Sleep)"** for details on Sleep mode.

### 12.10.2 TMR0 INTERRUPT

An overflow (FFh → 00h) in the TMR0 register will set flag bit, TMR0IF (INTCON<2>). The interrupt can be enabled/disabled by setting/clearing enable bit, TMR0IE (INTCON<5>) (see **Section 6.0 "Timer0 Module"**).

### 12.10.3 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<3>). See **Section 3.2 "EECON1 and EECON2 Registers"**.

## 12.11 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, Status registers). This will have to be implemented in software as shown in Example 12-1.

For PIC16F818 devices, the upper 64 bytes of each bank are common. Temporary holding registers, W_TEMP and STATUS_TEMP, should be placed here. These 64 locations do not require banking and therefore, make it easier for context save and restore.

For PIC16F819 devices, the upper 16 bytes of each bank are common.

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

```
    MOVWF   W_TEMP              ;Copy W to TEMP register
    SWAPF   STATUS, W           ;Swap status to be saved into W
    CLRF    STATUS              ;bank 0, regardless of current bank, Clears IRP,RP1,RP0
    MOVWF   STATUS_TEMP         ;Save status to bank zero STATUS_TEMP register
    :
    :(ISR)                      ;Insert user code here
    :
    SWAPF   STATUS_TEMP, W      ;Swap STATUS_TEMP register into W
                                ;(sets bank to original state)
    MOVWF   STATUS              ;Move W into STATUS register
    SWAPF   W_TEMP, F           ;Swap W_TEMP
    SWAPF   W_TEMP, W           ;Swap W_TEMP into W
```



Downloaded from Arrow.com.

## 12.12 Watchdog Timer (WDT)

For PIC16F818/819 devices, the WDT is driven by the INTRC oscillator. When the WDT is enabled, the INTRC (31.25 kHz) oscillator is enabled. The nominal WDT period is 16 ms and has the same accuracy as the INTRC oscillator.

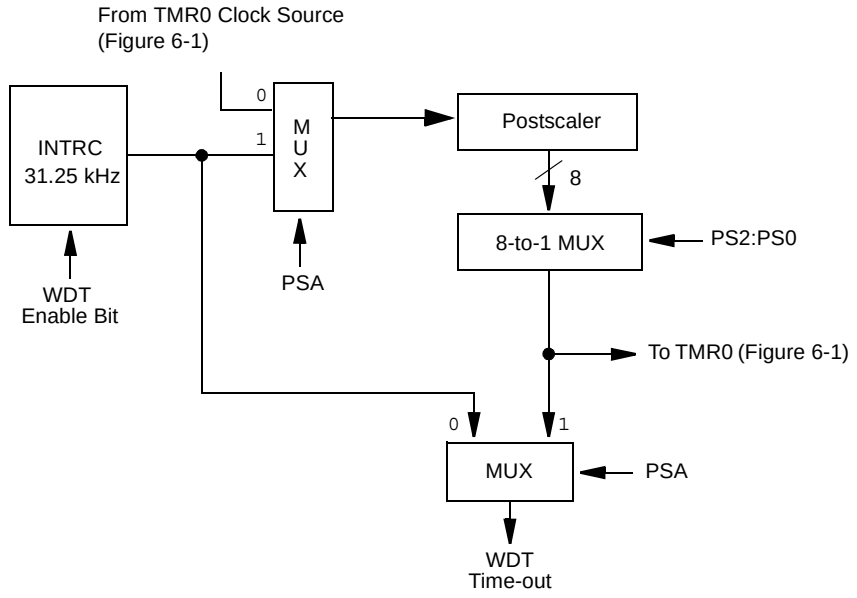
During normal operation, a WDT time-out generates a device Reset (Watchdog Timer Reset). If the device is in Sleep mode, a WDT time-out causes the device to wake-up and continue with normal operation (Watchdog Timer wake-up). The $\overline{TO}$ bit in the Status register will be cleared upon a Watchdog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

The WDT can be permanently disabled by clearing configuration bit, WDTEN (see **Section 12.1 "Configuration Bits"**).

WDT time-out period values may be found in **Section 15.0 "Electrical Characteristics"** under 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 1:** | 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. |
| **2:** | 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 12-8: WATCHDOG TIMER BLOCK DIAGRAM**




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

**TABLE 12-5: SUMMARY OF WATCHDOG TIMER REGISTERS**

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|
| 81h,181h | OPTION_REG | $\overline{RBPU}$ | INTEDG | T0CS | T0SE | PSA | PS2 | PS1 | PS0 |
| 2007h | Configuration bits(1) | LVP | BOREN | MCLRE | FOSC2 | $\overline{PWRTEN}$ | WDTEN | FOSC1 | FOSC0 |

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

**Note 1:** See Register 12-1 for operation of these bits.


© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

## 12.13 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 disable external clocks. Pull all I/O pins that are high-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 also be considered.

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

### 12.13.1 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 a peripheral interrupt.

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 the 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. CCP Capture mode interrupt.
3. Special event trigger (Timer1 in Asynchronous mode using an external clock).
4. SSP (Start/Stop) bit detect interrupt.
5. SSP transmit or receive in Slave mode (SPI/$I^2C$).
6. A/D conversion (when A/D clock source is RC).
7. EEPROM write operation completion.

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

When the SLEEP instruction is being executed, the next instruction (PC + 1) is prefetched. For the device to wake-up through an interrupt event, the corresponding interrupt enable bit must be set (enabled). Wake-up occurs regardless of the state of the GIE bit. If the GIE bit is clear (disabled), the device continues execution at the instruction after the SLEEP instruction. If the GIE bit is set (enabled), the device executes the instruction after the SLEEP instruction 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.

### 12.13.2 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 bit will not be cleared.
- If the interrupt occurs **during or after** the execution of a SLEEP instruction, the device will immediately wake-up from Sleep. The SLEEP instruction 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.

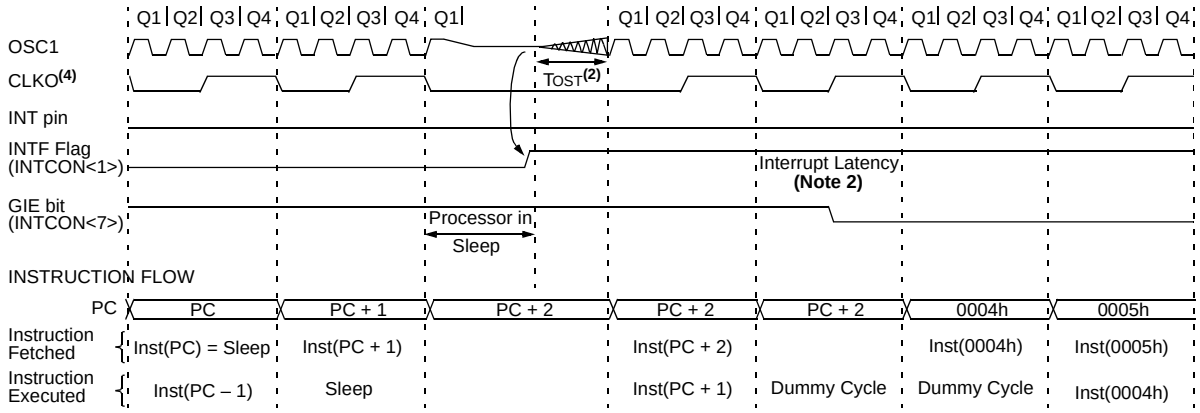
Even if the flag bits were checked before executing a SLEEP instruction, it may be possible for flag bits to become set before the SLEEP instruction completes. To determine whether a SLEEP instruction executed, test the PD bit. If the PD bit is set, the SLEEP instruction was executed as a NOP.

To ensure that the WDT is cleared, a CLRWDT instruction should be executed before a SLEEP instruction.



Downloaded from Arrow.com.

**FIGURE 12-9: WAKE-UP FROM SLEEP THROUGH INTERRUPT**




**Note 1:** XT, HS or LP Oscillator mode assumed.
**2:** TOST = 1024 TOSC (drawing not to scale). This delay will not be there for RC Oscillator mode.
**3:** GIE = 1 assumed. In this case, after wake-up, the processor jumps to the interrupt routine. If GIE = 0, execution will continue in-line.
**4:** CLKO is not available in these oscillator modes but shown here for timing reference.

## 12.14 In-Circuit Debugger

When the DEBUG bit in the Configuration Word is programmed to a '0', the In-Circuit Debugger functionality is enabled. This function allows simple debugging functions when used with MPLAB® ICD. When the microcontroller has this feature enabled, some of the resources are not available for general use. Table 12-6 shows which features are consumed by the background debugger.

**TABLE 12-6: DEBUGGER RESOURCES**

| | |
|---|---|
| I/O pins | RB6, RB7 |
| Stack | 1 level |
| Program Memory | Address 0000h must be NOP |
| | Last 100h words |
| Data Memory | 0x070 (0x0F0, 0x170, 0x1F0) 0x1EB-0x1EF |

To use the In-Circuit Debugger function of the microcontroller, the design must implement In-Circuit Serial Programming connections to MCLR/VPP, VDD, GND, RB7 and RB6. This will interface to the in-circuit debugger module available from Microchip or one of the third party development tool companies.

## 12.15 Program Verification/Code Protection

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

## 12.16 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 four Least Significant bits of the ID location are used.


© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

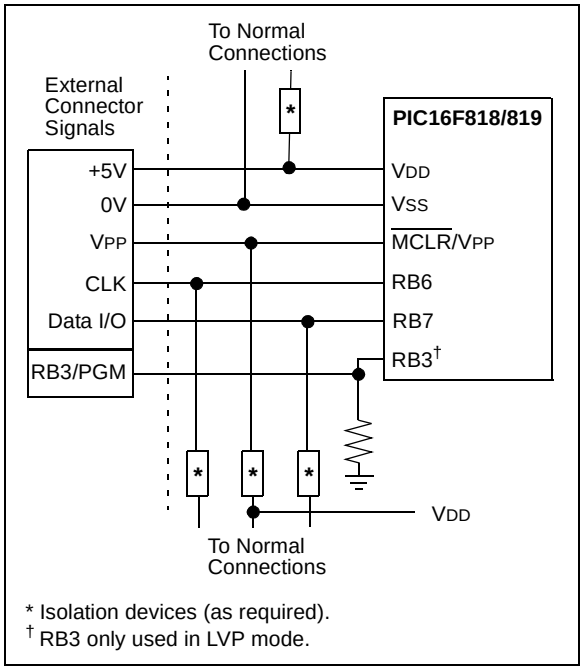
## 12.17 In-Circuit Serial Programming

PIC16F818/819 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 (see Figure 12-10 for an example). 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 more information on serial programming, please refer to the *"PIC16F818/819 Flash Memory Programming Specification"* (DS39603).

| **Note:** | The Timer1 oscillator shares the T1OSI and T1OSO pins with the PGD and PGC pins used for programming and debugging.<br><br>When using the Timer1 oscillator, In-Circuit Serial Programming™ (ICSP™) may not function correctly (high voltage or low voltage) or the In-Circuit Debugger (ICD) may not communicate with the controller. As a result of using either ICSP or ICD, the Timer1 crystal may be damaged.<br><br>If ICSP or ICD operations are required, the crystal should be disconnected from the circuit (disconnect either lead) or installed after programming. The oscillator loading capacitors may remain in-circuit during ICSP or ICD operation. |
|---|---|

**FIGURE 12-10: TYPICAL IN-CIRCUIT SERIAL PROGRAMMING CONNECTION**




* Isolation devices (as required).
† RB3 only used in LVP mode.



Downloaded from Arrow.com.

## 12.18 Low-Voltage ICSP Programming

The LVP bit of the Configuration Word register enables Low-Voltage ICSP Programming. This mode allows the microcontroller to be programmed via ICSP using a VDD source in the operating voltage range. This only means that VPP does not have to be brought to VIHH but can instead be left at the normal operating voltage. In this mode, the RB3/PGM pin is dedicated to the programming function and ceases to be a general purpose I/O pin.

If Low-Voltage Programming mode is not used, the LVP bit can be programmed to a '0' and RB3/PGM becomes a digital I/O pin. However, the LVP bit may only be programmed when Programming mode is entered with VIHH on MCLR. The LVP bit can only be changed when using high voltage on MCLR.

It should be noted that once the LVP bit is programmed to '0', only the High-Voltage Programming mode is available and only this mode can be used to program the device.

When using Low-Voltage ICSP, the part must be supplied at 4.5V to 5.5V if a bulk erase will be executed. This includes reprogramming of the code-protect bits from an ON state to an OFF state. For all other cases of Low-Voltage ICSP, the part may be programmed at the normal operating voltage. This means calibration values, unique user IDs or user code can be reprogrammed or added.

The following LVP steps assume the LVP bit is set in the Configuration Word register.

1. Apply VDD to the VDD pin.
2. Drive MCLR low.
3. Apply VDD to the RB3/PGM pin.
4. Apply VDD to the MCLR pin.
5. Follow with the associated programming steps.

**Note 1:** The High-Voltage Programming mode is always available, regardless of the state of the LVP bit, by applying VIHH to the MCLR pin.

**2:** While in Low-Voltage ICSP mode (LVP = 1), the RB3 pin can no longer be used as a general purpose I/O pin.

**3:** When using Low-Voltage ICSP Programming (LVP) and the pull-ups on PORTB are enabled, bit 3 in the TRISB register must be cleared to disable the pull-up on RB3 and ensure the proper operation of the device.

**4:** RB3 should not be allowed to float if LVP is enabled. An external pull-down device should be used to default the device to normal operating mode. If RB3 floats high, the PIC16F818/819 device will enter Programming mode.

**5:** LVP mode is enabled by default on all devices shipped from Microchip. It can be disabled by clearing the LVP bit in the Configuration Word register.

**6:** Disabling LVP will provide maximum compatibility to other PIC16CXXX devices.


© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

## 13.0 INSTRUCTION SET SUMMARY

The PIC16 instruction set is highly orthogonal and is comprised of three basic categories:

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

Each PIC16 instruction is a 14-bit word divided into an **opcode,** which specifies the instruction type and one or more **operands,** which further specify the operation of the instruction. The formats for each of the categories are presented in Figure 13-1, while the various opcode fields are summarized in Table 13-1.

Table 13-2 lists the instructions recognized by the MPASM™ assembler. A complete description of each instruction is also available in the *"PICmicro® Mid-Range MCU Family Reference Manual"* (DS33023).

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

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

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

For **literal and control** operations, 'k' represents an eight or eleven-bit constant or literal value

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

| | |
|---|---|
| **Note:** | To maintain upward compatibility with future PIC16F818/819 products, do not use the OPTION and TRIS instructions. |

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

## 13.1 READ-MODIFY-WRITE OPERATIONS

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

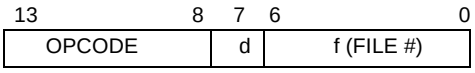
For example, a "CLRF PORTB" instruction will read PORTB, clear all the data bits, then write the result back to PORTB. This example would have the unintended result that the condition that sets the RBIF flag would be cleared.

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

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

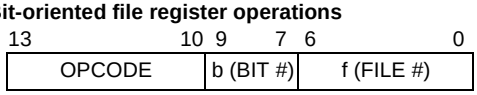
**FIGURE 13-1: GENERAL FORMAT FOR INSTRUCTIONS**

**Byte-oriented file register operations**




d = 0 for destination W
d = 1 for destination f
f = 7-bit file register address

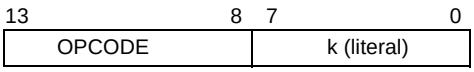
**Bit-oriented file register operations**




b = 3-bit bit address
f = 7-bit file register address

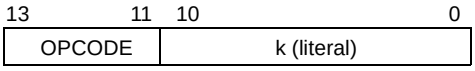
**Literal and control operations**

General




k = 8-bit immediate value

CALL and GOTO instructions only




k = 11-bit immediate value



Downloaded from Arrow.com.

**TABLE 13-2: PIC16F818/819 INSTRUCTION SET**

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

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

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

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

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


© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

## 13.2 Instruction Descriptions

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

| ADDWF | Add W and f |
|---|---|
| Syntax: | [ *label* ] ADDWF f,d |
| Operands: | 0 ≤ f ≤ 127<br>d ∈ [0,1] |
| Operation: | (W) + (f) → (destination) |
| Status Affected: | C, DC, Z |
| Description: | Add the contents of the W register with register 'f'. If 'd' = 0, the result is stored in the W register. If 'd' = 1, the result is stored back in register 'f'. |

| ANDLW | AND Literal with W |
|---|---|
| Syntax: | [ *label* ] ANDLW k |
| Operands: | 0 ≤ k ≤ 255 |
| Operation: | (W) .AND. (k) → (W) |
| Status Affected: | Z |
| Description: | The contents of W register are ANDed with the eight-bit literal 'k'. The result is placed in the W register. |

| ANDWF | AND W with f |
|---|---|
| Syntax: | [ *label* ] ANDWF f,d |
| Operands: | 0 ≤ f ≤ 127<br>d ∈ [0,1] |
| Operation: | (W) .AND. (f) → (destination) |
| Status Affected: | Z |
| Description: | AND the W register with register 'f'. If 'd' = 0, the result is stored in the W register. If 'd' = 1, the result is stored back in register 'f'. |

| BCF | Bit Clear f |
|---|---|
| Syntax: | [ *label* ] BCF f,b |
| Operands: | 0 ≤ f ≤ 127<br>0 ≤ b ≤ 7 |
| Operation: | 0 → (f<b>) |
| Status Affected: | None |
| Description: | Bit 'b' in register 'f' is cleared. |

| BSF | Bit Set f |
|---|---|
| Syntax: | [ *label* ] BSF f,b |
| Operands: | 0 ≤ f ≤ 127<br>0 ≤ b ≤ 7 |
| Operation: | 1 → (f<b>) |
| Status Affected: | None |
| Description: | Bit 'b' in register 'f' is set. |



Downloaded from Arrow.com.

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

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

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

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

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

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


© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

| COMF | Complement f |
|---|---|
| Syntax: | [ *label* ] COMF f,d |
| Operands: | 0 ≤ f ≤ 127<br>d ∈ [0,1] |
| Operation: | (f) → (destination) |
| Status Affected: | Z |
| Description: | The contents of register 'f' are complemented. If 'd' = 0, the result is stored in W. If 'd' = 1, the result is stored back in register 'f'. |

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

| DECF | Decrement f |
|---|---|
| Syntax: | [ *label* ] DECF f,d |
| Operands: | 0 ≤ f ≤ 127<br>d ∈ [0,1] |
| Operation: | (f) – 1 → (destination) |
| Status Affected: | Z |
| Description: | Decrement register 'f'. If 'd' = 0, the result is stored in the W register. If 'd' = 1, the result is stored back in register 'f'. |

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

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

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



Downloaded from Arrow.com.

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

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

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

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

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

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


© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

**RETFIE** **Return from Interrupt**

Syntax: [ *label* ] RETFIE

Operands: None

Operation: TOS → PC,
1 → GIE

Status Affected: None

**RETLW** **Return with Literal in W**

Syntax: [ *label* ] RETLW k

Operands: 0 ≤ k ≤ 255

Operation: k → (W);
TOS → PC

Status Affected: None

Description: The W register is loaded with the eight-bit literal ‘k’. The program counter is loaded from the top of the stack (the return address). This is a two-cycle instruction.

**RETURN** **Return from Subroutine**

Syntax: [ *label* ] RETURN

Operands: None

Operation: TOS → PC

Status Affected: None

Description: Return from subroutine. The stack is POPed and the top of the stack (TOS) is loaded into the program counter. This is a two-cycle instruction.

**RLF** **Rotate Left f through Carry**

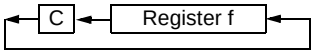
Syntax: [ *label* ] RLF f,d

Operands: 0 ≤ f ≤ 127
d ∈ [0,1]

Operation: See description below

Status Affected: C

Description: The contents of register ‘f’ are rotated one bit to the left through the Carry flag. If ‘d’ = 0, the result is placed in the W register. If ‘d’ = 1, the result is stored back in register ‘f’.




**RRF** **Rotate Right f through Carry**

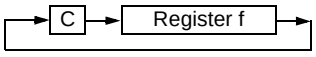
Syntax: [ *label* ] RRF f,d

Operands: 0 ≤ f ≤ 127
d ∈ [0,1]

Operation: See description below

Status Affected: C

Description: The contents of register ‘f’ are rotated one bit to the right through the Carry flag. If ‘d’ = 0, the result is placed in the W register. If ‘d’ = 1, the result is placed back in register ‘f’.




**SLEEP** **Enter Sleep mode**

Syntax: [ *label* ] SLEEP

Operands: None

Operation: 00h → WDT,
0 → WDT prescaler,
1 → $\overline{TO}$,
0 → $\overline{PD}$

Status Affected: $\overline{TO}$, $\overline{PD}$

Description: The Power-Down status bit, $\overline{PD}$, is cleared. Time-out status bit, $\overline{TO}$, is set. Watchdog Timer and its prescaler are cleared.
The processor is put into Sleep mode with the oscillator stopped.



Downloaded from Arrow.com.

**SUBLW** **Subtract W from Literal**

Syntax: [ *label* ] SUBLW k

Operands: $0 \le k \le 255$

Operation: $k - (W) \rightarrow (W)$

Status Affected: C, DC, Z

Description: The W register is subtracted (2's complement method) from the eight-bit literal 'k'. The result is placed in the W register.

**SUBWF** **Subtract W from f**

Syntax: [ *label* ] SUBWF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (f) – (W) → (destination)

Status Affected: C, DC, Z

Description: Subtract (2's complement method) W register from register 'f'. If 'd' = 0, the result is stored in the W register. If 'd' = 1, the result is stored back in register 'f'.

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

Syntax: [ *label* ] SWAPF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (f<3:0>) → (destination<7:4>),
(f<7:4>) → (destination<3:0>)

Status Affected: None

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

**XORLW** **Exclusive OR Literal with W**

Syntax: [ *label* ] XORLW k

Operands: $0 \le k \le 255$

Operation: (W) .XOR. k → (W)

Status Affected: Z

Description: The contents of the W register are XORed with the eight-bit literal 'k'. The result is placed in the W register.

**XORWF** **Exclusive OR W with f**

Syntax: [ *label* ] XORWF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (W) .XOR. (f) → (destination)

Status Affected: Z

Description: Exclusive OR the contents of the W register with register 'f'. If 'd' = 0, the result is stored in the W register. If 'd' = 1, the result is stored back in register 'f'.


© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

## 14.0 DEVELOPMENT SUPPORT

The PICmicro® microcontrollers are supported with a full range of hardware and software development tools:

- Integrated Development Environment
  - MPLAB® IDE Software
- Assemblers/Compilers/Linkers
  - MPASM™ Assembler
  - MPLAB 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
  - MPLAB PM3 Device 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
  - KEELOQ® Evaluation and Programming Tools
  - PICDEM MSC
  - microID® Developer Kits
  - CAN
  - PowerSmart® Developer Kits
  - Analog

### 14.1 MPLAB Integrated Development Environment Software

The MPLAB IDE software brings an ease of software development previously unseen in the 8/16-bit microcontroller market. The MPLAB IDE is a Windows® 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)
  - 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.

### 14.2 MPASM Assembler

The MPASM assembler is a full-featured, universal macro assembler for all PICmicro MCUs.

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

The MPASM assembler features include:

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



Downloaded from Arrow.com.

## 14.3 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.

## 14.4 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 precompiled libraries, using directives from a linker script.

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

The object linker/library features include:

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

## 14.5 MPLAB 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.

## 14.6 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

## 14.7 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.

## 14.8 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.


© 2003 Microchip Technology Inc.

Downloaded from Arrow.com.

## 14.9 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.

## 14.10 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 ICD 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.

## 14.11 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 MCUs and can be used to develop for 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.

## 14.12 PRO MATE II Universal Device Programmer

The PRO MATE II is a universal, CE compliant device programmer with programmable voltage verification at $V_{DDMIN}$ and $V_{DDMAX}$ 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.

## 14.13 MPLAB PM3 Device Programmer

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



Downloaded from Arrow.com.

## 14.14 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.

## 14.15 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.

## 14.16 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, web server with HTML, 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

## 14.17 PICDEM 2 Plus Demonstration Board

The PICDEM 2 Plus demonstration board supports many 18, 28 and 40-pin microcontrollers, including PIC16F87X and PIC18FXX2 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.

## 14.18 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.

## 14.19 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 supercapacitor 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, 2 x 16 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.


© 2003 Microchip Technology Inc.

Downloaded from Arrow.com.

## 14.20 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.

## 14.21 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/Demultiplexed 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.

## 14.22 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.

## 14.23 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.

## 14.24 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.

## 14.25 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 Selector Guide for the complete list of demonstration and evaluation kits.



Downloaded from Arrow.com.

**NOTES:**


© 2003 Microchip Technology Inc.

Downloaded from Arrow.com.

## 15.0 ELECTRICAL CHARACTERISTICS

### Absolute Maximum Ratings †

Ambient temperature under bias ..... -40°C to +125°C
Storage temperature ..... -65°C to +150°C
Voltage on any pin with respect to $V_{SS}$ (except $V_{DD}$ and $\overline{MCLR}$) ..... -0.3V to ($V_{DD}$ + 0.3V)
Voltage on $V_{DD}$ with respect to $V_{SS}$ ..... -0.3 to +7.5V
Voltage on $\overline{MCLR}$ with respect to $V_{SS}$ **(Note 2)** ..... -0.3 to +14V
Total power dissipation **(Note 1)** ..... 1W
Maximum current out of $V_{SS}$ pin ..... 200 mA
Maximum current into $V_{DD}$ pin ..... 200 mA
Input clamp current, $I_{IK}$ ($V_I$ < 0 or $V_I$ > $V_{DD}$) ..... ±20 mA
Output clamp current, $I_{OK}$ ($V_O$ < 0 or $V_O$ > $V_{DD}$) ..... ±20 mA
Maximum output current sunk by any I/O pin ..... 25 mA
Maximum output current sourced by any I/O pin ..... 25 mA
Maximum current sunk by PORTA ..... 100 mA
Maximum current sourced by PORTA ..... 100 mA
Maximum current sunk by PORTB ..... 100 mA
Maximum current sourced by PORTB ..... 100 mA

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

**2:** Voltage spikes at the $\overline{MCLR}$ pin may cause latch-up. A series resistor of greater than 1 kΩ should be used to pull $\overline{MCLR}$ to $V_{DD}$, rather than tying the pin directly to $V_{DD}$.

† 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.



Downloaded from Arrow.com.

**FIGURE 15-1: PIC16F818/819 VOLTAGE-FREQUENCY GRAPH (INDUSTRIAL, EXTENDED)**

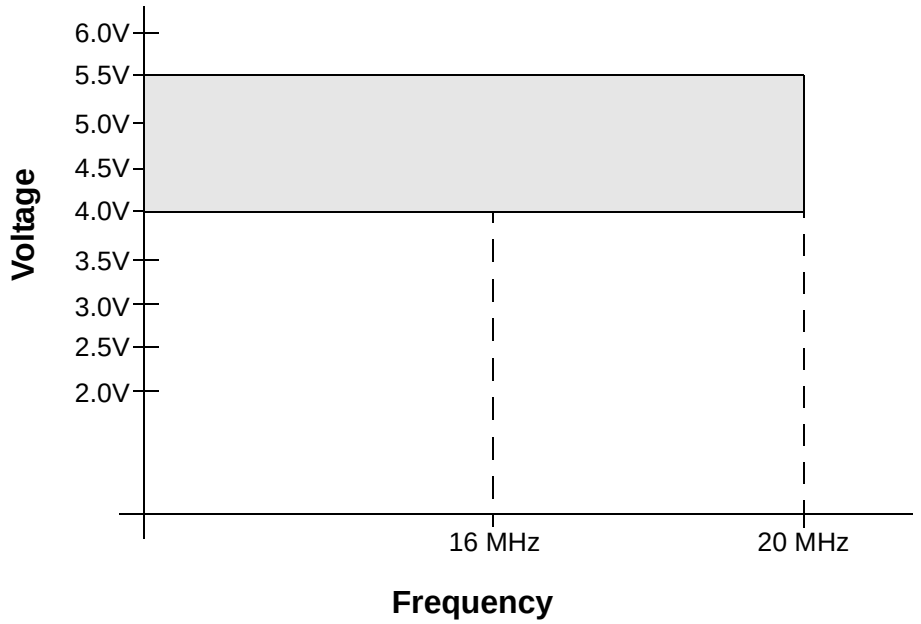
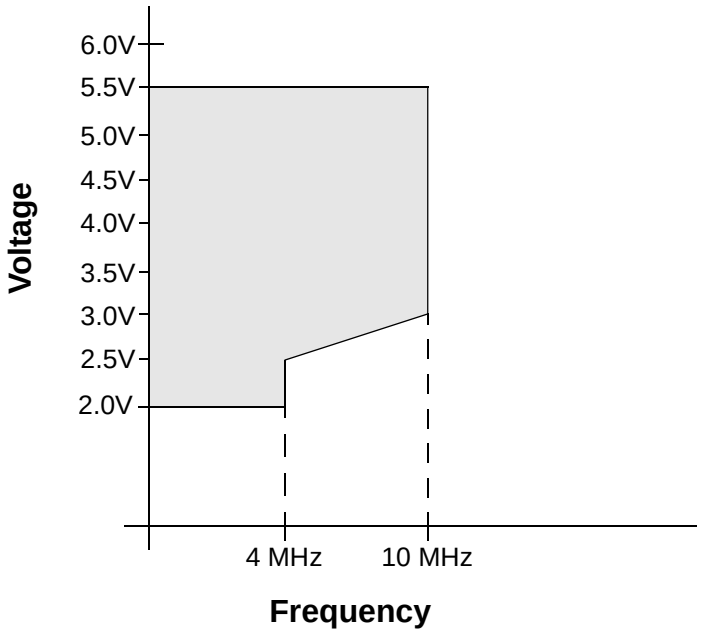



**FIGURE 15-2: PIC16LF818/819 VOLTAGE-FREQUENCY GRAPH (INDUSTRIAL)**




FMAX = (12 MHz/V) (VDDAPPMIN – 2.5V) + 4 MHz

**Note 1:** VDDAPPMIN is the minimum voltage of the PICmicro® device in the application.

**2:** FMAX has a maximum frequency of 10 MHz.


© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

## 15.1 DC Characteristics: Supply Voltage PIC16F818/819 (Industrial, Extended) PIC16LF818/819 (Industrial)

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **PIC16LF818/819** (Industrial) | | | **Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for industrial | | | | |
| **PIC16F818/819** (Industrial, Extended) | | | **Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for industrial<br>-40°C ≤ TA ≤ +125°C for extended | | | | |
| **Param No.** | **Symbol** | **Characteristic** | **Min** | **Typ** | **Max** | **Units** | **Conditions** |
| | VDD | **Supply Voltage** | | | | | |
| D001 | | PIC16LF818/819 | 2.0 | — | 5.5 | V | HS, XT, RC and LP Oscillator mode |
| D001 | | PIC16F818/819 | 4.0 | — | 5.5 | V | |
| D002 | VDR | **RAM Data Retention Voltage(1)** | 1.5 | — | — | V | |
| D003 | VPOR | **VDD Start Voltage** to ensure internal Power-on Reset signal | — | — | 0.7 | V | See **Section 12.4 "Power-on Reset (POR)"** for details |
| D004 | SVDD | **VDD Rise Rate** to ensure internal Power-on Reset signal | 0.05 | — | — | V/ms | See **Section 12.4 "Power-on Reset (POR)"** for details |
| | VBOR | **Brown-out Reset Voltage** | | | | | |
| D005 | | PIC16LF818/819 | 3.65 | — | 4.35 | V | |
| D005 | | PIC16F818/819 | 3.65 | — | 4.35 | V | FMAX = 14 MHz(2) |

**Legend:** Shading of rows is to assist in readability of the table.

**Note 1:** This is the limit to which VDD can be lowered in Sleep mode, or during a device Reset, without losing RAM data

**2:** When BOR is enabled, the device will operate correctly until the VBOR voltage trip point is reached.



Downloaded from Arrow.com.

## 15.2 DC Characteristics: Power-Down and Supply Current PIC16F818/819 (Industrial, Extended) PIC16LF818/819 (Industrial)

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **PIC16LF818/819** (Industrial) | | **Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for industrial | | | | | |
| **PIC16F818/819** (Industrial, Extended) | | **Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended | | | | | |
| **Param No.** | **Device** | **Typ** | **Max** | **Units** | **Conditions** | | |
| | **Power-Down Current (IPD)(1)** | | | | | | |
| | PIC16LF818/819 | 0.1 | 0.4 | µA | -40°C | VDD = 2.0V | |
| | | 0.1 | 0.4 | µA | +25°C | | |
| | | 0.4 | 1.5 | µA | +85°C | | |
| | PIC16LF818/819 | 0.3 | 0.5 | µA | -40°C | VDD = 3.0V | |
| | | 0.3 | 0.5 | µA | +25°C | | |
| | | 0.7 | 1.7 | µA | +85°C | | |
| | All devices | 0.6 | 1.0 | µA | -40°C | VDD = 5.0V | |
| | | 0.6 | 1.0 | µA | +25°C | | |
| | | 1.2 | 5.0 | µA | +85°C | | |
| | Extended devices | 6.0 | 28 | µA | +125°C | | |

**Legend:** Shading of rows is to assist in readability of the table.

**Note 1:** The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in high-impedance state and tied to VDD or VSS and all features that add delta current disabled (such as WDT, Timer1 Oscillator, BOR, etc.).

**2:** The supply current is mainly a function of operating voltage, frequency and mode. Other factors, such as I/O pin loading and switching rate, oscillator type and circuit, internal code execution pattern and temperature, also have an impact on the current consumption.
The test conditions for all IDD measurements in active operation mode are:
OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to VDD;
MCLR = VDD; WDT enabled/disabled as specified.

**3:** For RC oscillator configurations, current through REXT is not included. The current through the resistor can be estimated by the formula Ir = VDD/2REXT (mA) with REXT in kΩ.


© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

## 15.2 DC Characteristics: Power-Down and Supply Current PIC16F818/819 (Industrial, Extended) PIC16LF818/819 (Industrial) (Continued)

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **PIC16LF818/819** (Industrial) | | **Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for industrial | | | | | |
| **PIC16F818/819** (Industrial, Extended) | | **Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended | | | | | |
| **Param No.** | **Device** | **Typ** | **Max** | **Units** | **Conditions** | | |
| | **Supply Current (IDD)(2,3)** | | | | | | |
| | PIC16LF818/819 | 9 | 20 | µA | -40°C | VDD = 2.0V | FOSC = 32 kHz (LP Oscillator) |
| | | 7 | 15 | µA | +25°C | | |
| | | 7 | 15 | µA | +85°C | | |
| | PIC16LF818/819 | 16 | 30 | µA | -40°C | VDD = 3.0V | |
| | | 14 | 25 | µA | +25°C | | |
| | | 14 | 25 | µA | +85°C | | |
| | All devices | 32 | 40 | µA | -40°C | VDD = 5.0V | |
| | | 26 | 35 | µA | +25°C | | |
| | | 26 | 35 | µA | +85°C | | |
| | Extended devices | 35 | 53 | µA | +125°C | | |

**Legend:** Shading of rows is to assist in readability of the table.

**Note 1:** The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in high-impedance state and tied to VDD or VSS and all features that add delta current disabled (such as WDT, Timer1 Oscillator, BOR, etc.).

**2:** The supply current is mainly a function of operating voltage, frequency and mode. Other factors, such as I/O pin loading and switching rate, oscillator type and circuit, internal code execution pattern and temperature, also have an impact on the current consumption.
The test conditions for all IDD measurements in active operation mode are:
OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to VDD;
$\overline{MCLR}$ = VDD; WDT enabled/disabled as specified.

**3:** For RC oscillator configurations, current through REXT is not included. The current through the resistor can be estimated by the formula Ir = VDD/2REXT (mA) with REXT in kΩ.



Downloaded from Arrow.com.

## 15.2 DC Characteristics: Power-Down and Supply Current PIC16F818/819 (Industrial, Extended) PIC16LF818/819 (Industrial) (Continued)

| PIC16LF818/819 (Industrial) | Standard Operating Conditions (unless otherwise stated)<br>Operating temperature -40°C ≤ TA ≤ +85°C for industrial |
|---|---|
| **PIC16F818/819** (Industrial, Extended) | **Standard Operating Conditions (unless otherwise stated)**<br>Operating temperature -40°C ≤ TA ≤ +85°C for industrial<br>-40°C ≤ TA ≤ +125°C for extended |

| Param No. | Device | Typ | Max | Units | Conditions | | |
|---|---|---|---|---|---|---|---|
| | **Supply Current (IDD)(2,3)** | | | | | | |
| | PIC16LF818/819 | 72 | 95 | µA | -40°C | VDD = 2.0V | FOSC = 1 MHz (RC Oscillator)(3) |
| | | 76 | 90 | µA | +25°C | | |
| | | 76 | 90 | µA | +85°C | | |
| | PIC16LF818/819 | 138 | 175 | µA | -40°C | VDD = 3.0V | |
| | | 136 | 170 | µA | +25°C | | |
| | | 136 | 170 | µA | +85°C | | |
| | All devices | 310 | 380 | µA | -40°C | VDD = 5.0V | |
| | | 290 | 360 | µA | +25°C | | |
| | | 280 | 360 | µA | +85°C | | |
| | Extended devices | 350 | 500 | µA | +125°C | | |
| | PIC16LF818/819 | 270 | 315 | µA | -40°C | VDD = 2.0V | FOSC = 4 MHz (RC Oscillator)(3) |
| | | 280 | 310 | µA | +25°C | | |
| | | 285 | 310 | µA | +85°C | | |
| | PIC16LF818/819 | 460 | 610 | µA | -40°C | VDD = 3.0V | |
| | | 450 | 600 | µA | +25°C | | |
| | | 450 | 600 | µA | +85°C | | |
| | All devices | 900 | 1060 | µA | -40°C | VDD = 5.0V | |
| | | 890 | 1050 | µA | +25°C | | |
| | | 890 | 1050 | µA | +85°C | | |
| | Extended devices | .920 | 1.5 | mA | +125°C | | |

**Legend:** Shading of rows is to assist in readability of the table.

**Note 1:** The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in high-impedance state and tied to VDD or VSS and all features that add delta current disabled (such as WDT, Timer1 Oscillator, BOR, etc.).

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

The test conditions for all IDD measurements in active operation mode are:

OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to VDD;
MCLR = VDD; WDT enabled/disabled as specified.

**3:** For RC oscillator configurations, current through REXT is not included. The current through the resistor can be estimated by the formula Ir = VDD/2REXT (mA) with REXT in kΩ.


© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

## 15.2 DC Characteristics: Power-Down and Supply Current PIC16F818/819 (Industrial, Extended) PIC16LF818/819 (Industrial) (Continued)

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **PIC16LF818/819** (Industrial) | | **Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for industrial | | | | | |
| **PIC16F818/819** (Industrial, Extended) | | **Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended | | | | | |
| **Param No.** | **Device** | **Typ** | **Max** | **Units** | **Conditions** | | |
| | **Supply Current (IDD)(2,3)** | | | | | | |
| | All devices | 1.8 | 2.3 | mA | -40°C | VDD = 4.0V | FOSC = 20 MHz (HS Oscillator) |
| | | 1.6 | 2.2 | mA | +25°C | | |
| | | 1.3 | 2.2 | mA | +85°C | | |
| | All devices | 3.0 | 4.2 | mA | -40°C | VDD = 5.0V | |
| | | 2.5 | 4.0 | mA | +25°C | | |
| | | 2.5 | 4.0 | mA | +85°C | | |
| | Extended devices | 3.0 | 5.0 | mA | +125°C | | |

**Legend:** Shading of rows is to assist in readability of the table.

**Note 1:** The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in high-impedance state and tied to VDD or VSS and all features that add delta current disabled (such as WDT, Timer1 Oscillator, BOR, etc.).

**2:** The supply current is mainly a function of operating voltage, frequency and mode. Other factors, such as I/O pin loading and switching rate, oscillator type and circuit, internal code execution pattern and temperature, also have an impact on the current consumption.
The test conditions for all IDD measurements in active operation mode are:
OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to VDD;
$\overline{MCLR}$ = VDD; WDT enabled/disabled as specified.

**3:** For RC oscillator configurations, current through REXT is not included. The current through the resistor can be estimated by the formula Ir = VDD/2REXT (mA) with REXT in kΩ.



Downloaded from Arrow.com.

## 15.2 DC Characteristics: Power-Down and Supply Current PIC16F818/819 (Industrial, Extended) PIC16LF818/819 (Industrial) (Continued)

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **PIC16LF818/819** (Industrial) | | **Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for industrial | | | | | |
| **PIC16F818/819** (Industrial, Extended) | | **Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for industrial<br>-40°C ≤ TA ≤ +125°C for extended | | | | | |
| **Param No.** | **Device** | **Typ** | **Max** | **Units** | **Conditions** | | |
| | **Supply Current (IDD)(2,3)** | | | | | | |
| | PIC16LF818/819 | 8 | 20 | µA | -40°C | VDD = 2.0V | FOSC = 31.25 kHz (**RC_RUN** mode, Internal RC Oscillator) |
| | | 7 | 15 | µA | +25°C | | |
| | | 7 | 15 | µA | +85°C | | |
| | PIC16LF818/819 | 16 | 30 | µA | -40°C | VDD = 3.0V | |
| | | 14 | 25 | µA | +25°C | | |
| | | 14 | 25 | µA | +85°C | | |
| | All devices | 32 | 40 | µA | -40°C | VDD = 5.0V | |
| | | 29 | 35 | µA | +25°C | | |
| | | 29 | 35 | µA | +85°C | | |
| | Extended devices | 35 | 45 | µA | +125°C | | |
| | PIC16LF818/819 | 132 | 160 | µA | -40°C | VDD = 2.0V | FOSC = 1 MHz (**RC_RUN** mode, Internal RC Oscillator) |
| | | 126 | 155 | µA | +25°C | | |
| | | 126 | 155 | µA | +85°C | | |
| | PIC16LF818/819 | 260 | 310 | µA | -40°C | VDD = 3.0V | |
| | | 230 | 300 | µA | +25°C | | |
| | | 230 | 300 | µA | +85°C | | |
| | All devices | 560 | 690 | µA | -40°C | VDD = 5.0V | |
| | | 500 | 650 | µA | +25°C | | |
| | | 500 | 650 | µA | +85°C | | |
| | Extended devices | 570 | 710 | µA | +125°C | | |
| | PIC16LF818/819 | 310 | 420 | µA | -40°C | VDD = 2.0V | FOSC = 4 MHz (**RC_RUN** mode, Internal RC Oscillator) |
| | | 300 | 410 | µA | +25°C | | |
| | | 300 | 410 | µA | +85°C | | |
| | PIC16LF818/819 | 550 | 650 | µA | -40°C | VDD = 3.0V | |
| | | 530 | 620 | µA | +25°C | | |
| | | 530 | 620 | µA | +85°C | | |
| | All devices | 1.2 | 1.5 | mA | -40°C | VDD = 5.0V | |
| | | 1.1 | 1.4 | mA | +25°C | | |
| | | 1.1 | 1.4 | mA | +85°C | | |
| | Extended devices | 1.3 | 1.6 | mA | +125°C | | |

**Legend:** Shading of rows is to assist in readability of the table.

**Note 1:** The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in high-impedance state and tied to VDD or VSS and all features that add delta current disabled (such as WDT, Timer1 Oscillator, BOR, etc.).

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

The test conditions for all IDD measurements in active operation mode are:

OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to VDD;

$\overline{MCLR}$ = VDD; WDT enabled/disabled as specified.

**3:** For RC oscillator configurations, current through REXT is not included. The current through the resistor can be estimated by the formula Ir = VDD/2REXT (mA) with REXT in kΩ.


© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

## 15.2 DC Characteristics: Power-Down and Supply Current PIC16F818/819 (Industrial, Extended) PIC16LF818/819 (Industrial) (Continued)

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **PIC16LF818/819** (Industrial) | | **Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for industrial | | | | | |
| **PIC16F818/819** (Industrial, Extended) | | **Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for industrial<br>-40°C ≤ TA ≤ +125°C for extended | | | | | |
| **Param No.** | **Device** | **Typ** | **Max** | **Units** | **Conditions** | | |
| | **Supply Current (IDD)(2,3)** | | | | | | |
| | PIC16LF818/819 | .950 | 1.3 | mA | -40°C | VDD = 3.0V | FOSC = 8 MHz (**RC_RUN** mode, Internal RC Oscillator) |
| | | .930 | 1.2 | mA | +25°C | | |
| | | .930 | 1.2 | mA | +85°C | | |
| | All devices | 1.8 | 3.0 | mA | -40°C | VDD = 5.0V | |
| | | 1.7 | 2.8 | mA | +25°C | | |
| | | 1.7 | 2.8 | mA | +85°C | | |
| | Extended devices | 2.0 | 4.0 | mA | +125°C | | |

**Legend:** Shading of rows is to assist in readability of the table.

**Note 1:** The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in high-impedance state and tied to VDD or VSS and all features that add delta current disabled (such as WDT, Timer1 Oscillator, BOR, etc.).

**2:** The supply current is mainly a function of operating voltage, frequency and mode. Other factors, such as I/O pin loading and switching rate, oscillator type and circuit, internal code execution pattern and temperature, also have an impact on the current consumption.
The test conditions for all IDD measurements in active operation mode are:
OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to VDD;
$\overline{MCLR}$ = VDD; WDT enabled/disabled as specified.

**3:** For RC oscillator configurations, current through REXT is not included. The current through the resistor can be estimated by the formula Ir = VDD/2REXT (mA) with REXT in kΩ.



Downloaded from Arrow.com.

## 15.2 DC Characteristics: Power-Down and Supply Current PIC16F818/819 (Industrial, Extended) PIC16LF818/819 (Industrial) (Continued)

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **PIC16LF818/819** (Industrial) | | **Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for industrial | | | | | |
| **PIC16F818/819** (Industrial, Extended) | | **Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended | | | | | |
| **Param No.** | **Device** | **Typ** | **Max** | **Units** | **Conditions** | | |
| D022 (ΔIWDT) | **Module Differential Currents (ΔIWDT, ΔIBOR, ΔILVD, ΔIOSCB, ΔIAD)** | | | | | | |
| | **Watchdog Timer** | 1.5 | 3.8 | µA | -40°C | VDD = 2.0V | |
| | | 2.2 | 3.8 | µA | +25°C | | |
| | | 2.7 | 4.0 | µA | +85°C | | |
| | | 2.3 | 4.6 | µA | -40°C | VDD = 3.0V | |
| | | 2.7 | 4.6 | µA | +25°C | | |
| | | 3.1 | 4.8 | µA | +85°C | | |
| | | 3.0 | 10.0 | µA | -40°C | VDD = 5.0V | |
| | | 3.3 | 10.0 | µA | +25°C | | |
| | | 3.9 | 13.0 | µA | +85°C | | |
| | Extended Devices | 5.0 | 21.0 | µA | +125°C | | |
| D022A (ΔIBOR) | **Brown-out Reset** | 40 | 60 | µA | -40°C to +85°C | VDD = 5.0V | |
| D025 (ΔIOSCB) | **Timer1 Oscillator** | 1.7 | 2.3 | µA | -40°C | VDD = 2.0V | 32 kHz on Timer1 |
| | | 1.8 | 2.3 | µA | +25°C | | |
| | | 2.0 | 2.3 | µA | +85°C | | |
| | | 2.2 | 3.8 | µA | -40°C | VDD = 3.0V | |
| | | 2.6 | 3.8 | µA | +25°C | | |
| | | 2.9 | 3.8 | µA | +85°C | | |
| | | 3.0 | 6.0 | µA | -40°C | VDD = 5.0V | |
| | | 3.2 | 6.0 | µA | +25°C | | |
| | | 3.4 | 7.0 | µA | +85°C | | |
| D026 (ΔIAD) | **A/D Converter** | 0.001 | 2.0 | µA | -40°C to +85°C | VDD = 2.0V | A/D on, Sleep, not converting |
| | | 0.001 | 2.0 | µA | -40°C to +85°C | VDD = 3.0V | |
| | | 0.003 | 2.0 | µA | -40°C to +85°C | VDD = 5.0V | |
| | Extended Devices | 4.0 | 8.0 | µA | -40°C to +125°C | | |

**Legend:** Shading of rows is to assist in readability of the table.

**Note 1:** The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in high-impedance state and tied to VDD or VSS and all features that add delta current disabled (such as WDT, Timer1 Oscillator, BOR, etc.).

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

The test conditions for all IDD measurements in active operation mode are:

OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to VDD;
MCLR = VDD; WDT enabled/disabled as specified.

**3:** For RC oscillator configurations, current through REXT is not included. The current through the resistor can be estimated by the formula Ir = VDD/2REXT (mA) with REXT in kΩ.


© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

## 15.3 DC Characteristics: Internal RC Accuracy
## PIC16F818/819, PIC16F818/819 TSL (Industrial, Extended)
## PIC16LF818/819, PIC16LF818/819 TSL (Industrial)

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **PIC16LF818/819(3)**<br>**PIC16LF818/819 TSL(3)**<br>(Industrial) | | **Standard Operating Conditions (unless otherwise stated)**<br>Operating temperature -40°C ≤ TA ≤ +85°C for industrial | | | | | |
| **PIC16F818/819(3)**<br>**PIC16F818/819 TSL(3)**<br>(Industrial, Extended) | | **Standard Operating Conditions (unless otherwise stated)**<br>Operating temperature -40°C ≤ TA ≤ +85°C for industrial<br>-40°C ≤ TA ≤ +125°C for extended | | | | | |
| **Param No.** | **Device** | **Min** | **Typ** | **Max** | **Units** | **Conditions** | |
| | **INTOSC Accuracy @ Freq = 8 MHz, 4 MHz, 2 MHz, 1 MHz, 500 kHz, 250 kHz, 125 kHz(1)** | | | | | | |
| | PIC16LF818/819 | -5 | ±1 | 5 | % | +25°C | VDD = 2.7-3.3V |
| | | -25 | — | 25 | % | -10°C to +85°C | |
| | | -30 | — | 30 | % | -40°C to +85°C | |
| | PIC16F818/819(4) | -5 | ±1 | 5 | % | +25°C | VDD = 4.5-5.5V |
| | | -25 | — | 25 | % | -10°C to +85°C | |
| | | -30 | — | 30 | % | -40°C to +85°C | |
| | | -35 | — | 35 | % | -40°C to +125°C | |
| | PIC16LF818/819 TSL | -2 | ±1 | 2 | % | +25°C | VDD = 2.7-3.3V |
| | | -5 | — | 5 | % | -10°C to +85°C | |
| | | -10 | — | 10 | % | -40°C to +85°C | |
| | PIC16F818/819 TSL(5) | -2 | ±1 | 2 | % | +25°C | VDD = 4.5-5.5V |
| | | -5 | — | 5 | % | -10°C to +85°C | |
| | | -10 | — | 10 | % | -40°C to +85°C | |
| | | -15 | — | 15 | % | -40°C to +125°C | |
| | **INTRC Accuracy @ Freq = 31 kHz(2)** | | | | | | |
| | PIC16LF818/819 | 26.562 | — | 35.938 | kHz | -40°C to +85°C | VDD = 2.7-3.3V |
| | PIC16F818/819(4) | 26.562 | — | 35.938 | kHz | -40°C to +85°C | VDD = 4.5-5.5V |
| | PIC16LF818/819 TSL | 26.562 | — | 35.938 | kHz | -40°C to +85°C | VDD = 2.7-3.3V |
| | PIC16F818/819 TSL(5) | 26.562 | — | 35.938 | kHz | -40°C to +85°C | VDD = 4.5-5.5V |

**Legend:** Shading of rows is to assist in readability of the table.

**Note 1:** Frequency calibrated at 25°C. OSCTUNE register can be used to compensate for temperature drift.

**2:** INTRC frequency after calibration.

**3:** The only specification difference between a non-TSL device and a TSL device is the internal RC oscillator specifications listed above. All other specifications are maintained.

**4:** Example part number for the specifications listed above: PIC16F818-I/SS (PIC16F818 device, Industrial temperature, SSOP package).

**5:** Example part number for the specifications listed above: PIC16F818-I/SSTSL (PIC16F818 device, Industrial temperature, SSOP package).



Downloaded from Arrow.com.

## 15.4 DC Characteristics: PIC16F818/819 (Industrial, Extended) PIC16LF818/819 (Industrial)

| DC CHARACTERISTICS | | | Standard Operating Conditions (unless otherwise stated)<br>Operating temperature -40°C ≤ TA ≤ +85°C for industrial<br>-40°C ≤ TA ≤ +125°C for extended<br>Operating voltage VDD range as described in **Section 15.1 "DC Characteristics: Supply Voltage"**. | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Sym** | **Characteristic** | **Min** | **Typ†** | **Max** | **Units** | **Conditions** |
| | VIL | **Input Low Voltage** | | | | | |
| | | I/O ports: | | | | | |
| D030 | | with TTL buffer | VSS | — | 0.15 VDD | V | For entire VDD range |
| D030A | | | VSS | — | 0.8V | V | 4.5V ≤ VDD ≤ 5.5V |
| D031 | | with Schmitt Trigger buffer | VSS | — | 0.2 VDD | V | |
| D032 | | $\overline{MCLR}$, OSC1 (in RC mode) | VSS | — | 0.2 VDD | V | **(Note 1)** |
| D033 | | OSC1 (in XT and LP mode) | VSS | — | 0.3V | V | |
| | | OSC1 (in HS mode) | VSS | — | 0.3 VDD | V | |
| | | Ports RB1 and RB4: | | | | | |
| D034 | | with Schmitt Trigger buffer | VSS | — | 0.3 VDD | V | For entire VDD range |
| | VIH | **Input High Voltage** | | | | | |
| | | I/O ports: | | | | | |
| D040 | | with TTL buffer | 2.0 | — | VDD | V | 4.5V ≤ VDD ≤ 5.5V |
| D040A | | | 0.25 VDD + 0.8V | — | VDD | V | For entire VDD range |
| D041 | | with Schmitt Trigger buffer | 0.8 VDD | — | VDD | V | For entire VDD range |
| D042 | | $\overline{MCLR}$ | 0.8 VDD | — | VDD | V | |
| D042A | | OSC1 (in XT and LP mode) | 1.6V | — | VDD | V | |
| | | OSC1 (in HS mode) | 0.7 VDD | — | VDD | V | |
| D043 | | OSC1 (in RC mode) | 0.9 VDD | — | VDD | V | **(Note 1)** |
| | | Ports RB1 and RB4: | | | | | |
| D044 | | with Schmitt Trigger buffer | 0.7 VDD | — | VDD | V | For entire VDD range |
| D070 | IPURB | **PORTB Weak Pull-up Current** | 50 | 250 | 400 | µA | VDD = 5V, VPIN = VSS |
| | IIL | **Input Leakage Current (Notes 2, 3)** | | | | | |
| D060 | | I/O ports | — | — | ±1 | µA | Vss ≤ VPIN ≤ VDD, pin at high-impedance |
| D061 | | $\overline{MCLR}$ | — | — | ±5 | µA | Vss ≤ VPIN ≤ VDD |
| D063 | | OSC1 | — | — | ±5 | µA | Vss ≤ VPIN ≤ VDD, XT, HS and LP oscillator configuration |

† 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:** In RC oscillator configuration, the OSC1/CLKI pin is a Schmitt Trigger input. It is not recommended that the PIC16F818/819 be driven with external clock in RC mode.

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

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


© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

## 15.4 DC Characteristics: PIC16F818/819 (Industrial, Extended) PIC16LF818/819 (Industrial) (Continued)

| DC CHARACTERISTICS | | | Standard Operating Conditions (unless otherwise stated)<br>Operating temperature -40°C ≤ TA ≤ +85°C for industrial<br>-40°C ≤ TA ≤ +125°C for extended<br>Operating voltage VDD range as described in **Section 15.1 "DC Characteristics: Supply Voltage"**. | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Sym** | **Characteristic** | **Min** | **Typ†** | **Max** | **Units** | **Conditions** |
| | VOL | **Output Low Voltage** | | | | | |
| D080 | | I/O ports | — | — | 0.6 | V | IOL = 8.5 mA, VDD = 4.5V, -40°C to +125°C |
| D083 | | OSC2/CLKO (RC oscillator config) | — | — | 0.6 | V | IOL = 1.6 mA, VDD = 4.5V, -40°C to +125°C |
| | VOH | **Output High Voltage** | | | | | |
| D090 | | I/O ports **(Note 3)** | VDD – 0.7 | — | — | V | IOH = -3.0 mA, VDD = 4.5V, -40°C to +125°C |
| D092 | | OSC2/CLKO (RC oscillator config) | VDD – 0.7 | — | — | V | IOH = -1.3 mA, VDD = 4.5V, -40°C to +125°C |
| | | **Capacitive Loading Specs on Output Pins** | | | | | |
| D100 | COSC2 | OSC2 pin | — | — | 15 | pF | In XT, HS and LP modes when external clock is used to drive OSC1 |
| D101 | CIO | All I/O pins and OSC2 (in RC mode) | — | — | 50 | pF | |
| D102 | CB | SCL, SDA in $I^2C$™ mode | — | — | 400 | pF | |
| | | **Data EEPROM Memory** | | | | | |
| D120 | ED | Endurance | 100K | 1M | — | E/W | -40°C to +85°C |
| | | | 10K | 100K | — | E/W | +85°C to +125°C |
| D121 | VDRW | VDD for read/write | VMIN | — | 5.5 | V | Using EECON to read/write, VMIN = min. operating voltage |
| D122 | TDEW | Erase/write cycle time | — | 4 | 8 | ms | |
| | | **Program Flash Memory** | | | | | |
| D130 | EP | Endurance | 10K | 100K | — | E/W | -40°C to +85°C |
| | | | 1K | 10K | — | E/W | +85°C to +125°C |
| D131 | VPR | VDD for read | VMIN | — | 5.5 | V | |
| D132A | | VDD for erase/write | VMIN | — | 5.5 | V | Using EECON to read/write, VMIN = min. operating voltage |
| D133 | TPE | Erase cycle time | — | 2 | 4 | ms | |
| D134 | TPW | Write cycle time | — | 2 | 4 | ms | |

† 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:** In RC oscillator configuration, the OSC1/CLKI pin is a Schmitt Trigger input. It is not recommended that the PIC16F818/819 be driven with external clock in RC mode.

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

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



Downloaded from Arrow.com.

## 15.5 Timing Parameter Symbology

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

1. TppS2ppS
2. TppS
3. TCC:ST (I²C specifications only)
4. Ts (I²C specifications only)

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

Lowercase letters (pp) and their meanings:

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

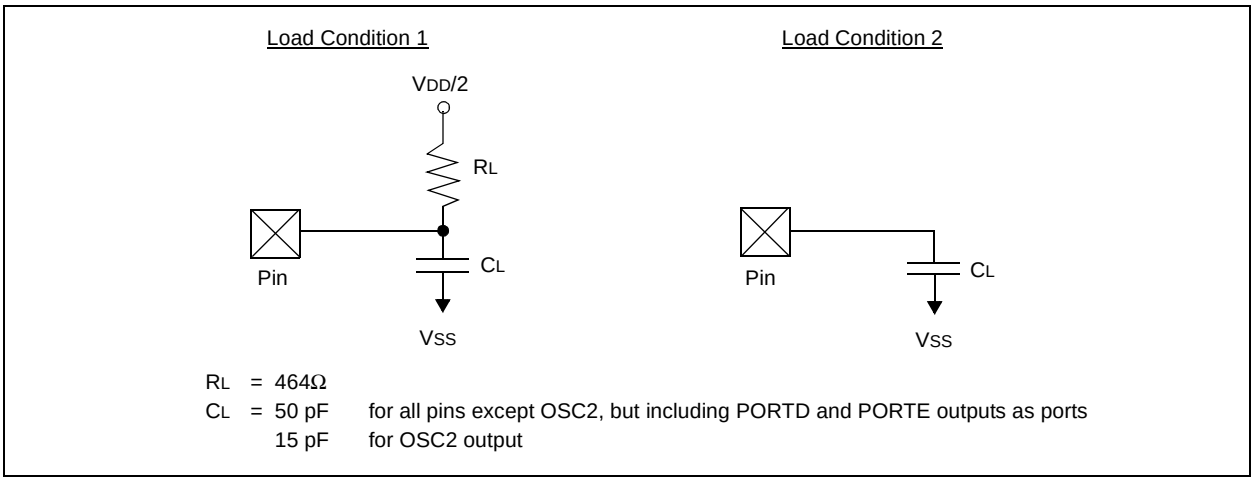
Uppercase letters and their meanings:

| | | | |
|---|---|---|---|
| **S** | | | |
| F | Fall | P | Period |
| H | High | R | Rise |
| I | Invalid (High-impedance) | V | Valid |
| L | Low | Z | High-impedance |
| **I²C only** | | | |
| AA | output access | High | High |
| BUF | Bus free | Low | Low |

TCC:ST (I²C specifications only)

| | | | |
|---|---|---|---|
| **CC** | | | |
| HD | Hold | SU | Setup |
| **ST** | | | |
| DAT | DATA input hold | STO | Stop condition |
| STA | Start condition | | |

**FIGURE 15-3: LOAD CONDITIONS**




RL = 464Ω
CL = 50 pF for all pins except OSC2, but including PORTD and PORTE outputs as ports
15 pF for OSC2 output


© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

**FIGURE 15-4: EXTERNAL CLOCK TIMING**

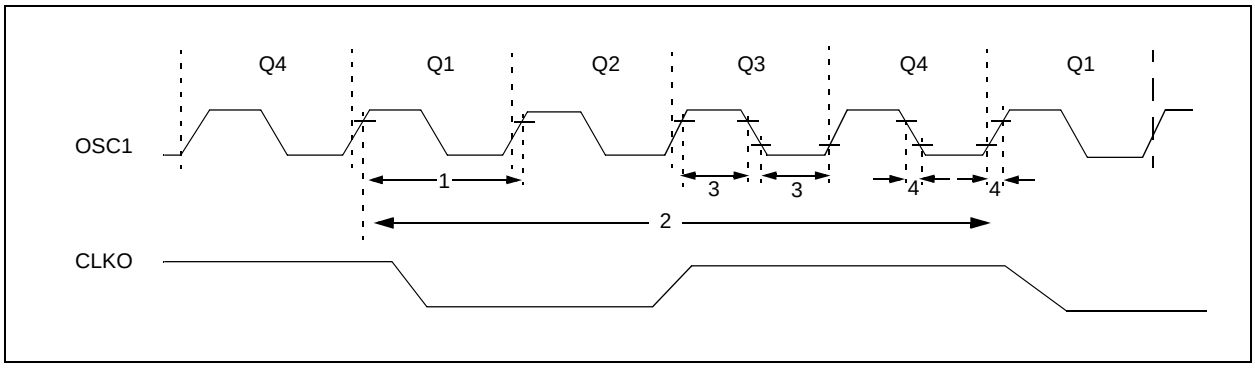



**TABLE 15-1: EXTERNAL CLOCK TIMING REQUIREMENTS**

| Param No. | Sym | Characteristic | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|
| | FOSC | External CLKI Frequency **(Note 1)** | DC | — | 1 | MHz | XT and RC Oscillator mode |
| | | | DC | — | 20 | MHz | HS Oscillator mode |
| | | | DC | — | 32 | kHz | LP Oscillator mode |
| | | Oscillator Frequency **(Note 1)** | DC | — | 4 | MHz | RC Oscillator mode |
| | | | 0.1 | — | 4 | MHz | XT Oscillator mode |
| | | | 4 | — | 20 | MHz | HS Oscillator mode |
| | | | 5 | — | 200 | kHz | LP Oscillator mode |
| 1 | TOSC | External CLKI Period **(Note 1)** | 1000 | — | — | ns | XT and RC Oscillator mode |
| | | | 50 | — | — | ns | HS Oscillator mode |
| | | | 5 | — | — | ms | LP Oscillator mode |
| | | Oscillator Period **(Note 1)** | 250 | — | — | ns | RC Oscillator mode |
| | | | 250 | — | 10,000 | ns | XT Oscillator mode |
| | | | 50 | — | 250 | ns | HS Oscillator mode |
| | | | 5 | — | — | ms | LP Oscillator mode |
| 2 | TCY | Instruction Cycle Time **(Note 1)** | 200 | TCY | DC | ns | TCY = 4/FOSC |
| 3 | TOSL, TOSH | External Clock in (OSC1) High or Low Time | 500 | — | — | ns | XT Oscillator |
| | | | 2.5 | — | — | ms | LP Oscillator |
| | | | 15 | — | — | ns | HS Oscillator |
| 4 | TOSR, TOSF | External Clock in (OSC1) Rise or Fall Time | — | — | 25 | ns | XT Oscillator |
| | | | — | — | 50 | ns | LP Oscillator |
| | | | — | — | 15 | ns | HS Oscillator |

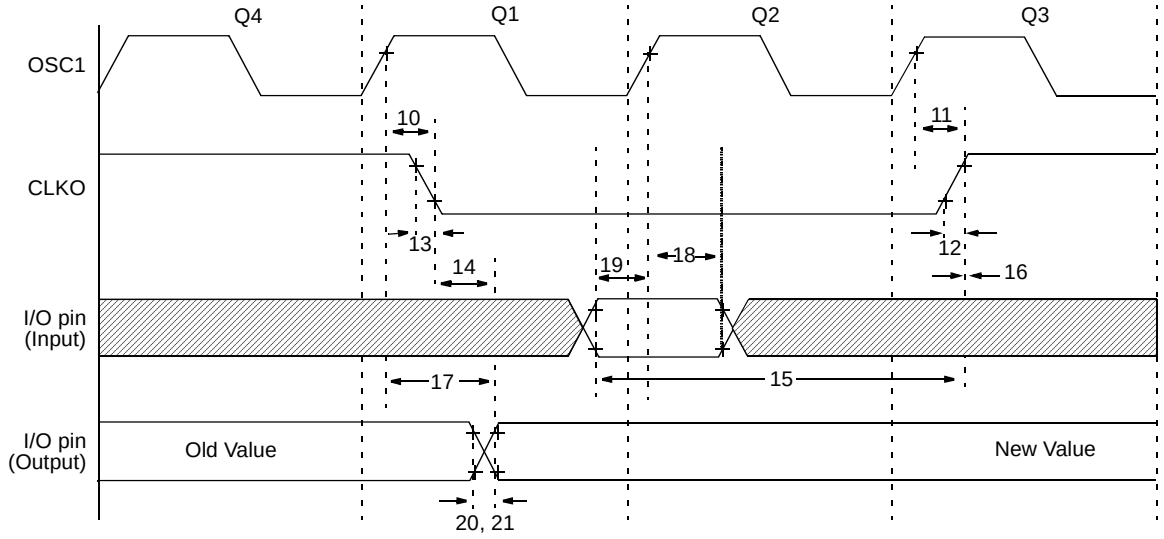
† Data in “Typ” column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**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/CLKI pin. When an external clock input is used, the “max.” cycle time limit is “DC” (no clock) for all devices.



Downloaded from Arrow.com.

**FIGURE 15-5: CLKO AND I/O TIMING**




**Note:** Refer to Figure 15-3 for load conditions.

**TABLE 15-2: CLKO AND I/O TIMING REQUIREMENTS**

| Param No. | Symbol | Characteristic | | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|---|
| 10* | TosH2ckL | OSC1 ↑ to CLKO ↓ | | — | 75 | 200 | ns | **(Note 1)** |
| 11* | TosH2ckH | OSC1 ↑ to CLKO ↑ | | — | 75 | 200 | ns | **(Note 1)** |
| 12* | TckR | CLKO Rise Time | | — | 35 | 100 | ns | **(Note 1)** |
| 13* | TckF | CLKO Fall Time | | — | 35 | 100 | ns | **(Note 1)** |
| 14* | TckL2ioV | CLKO ↓ to Port Out Valid | | — | — | 0.5 Tcy + 20 | ns | **(Note 1)** |
| 15* | TioV2ckH | Port In Valid before CLKO ↑ | | Tosc + 200 | — | — | ns | **(Note 1)** |
| 16* | TckH2ioI | Port In Hold after CLKO ↑ | | 0 | — | — | ns | **(Note 1)** |
| 17* | TosH2ioV | OSC1 ↑ (Q1 cycle) to Port Out Valid | | — | 100 | 255 | ns | |
| 18* | TosH2ioI | OSC1 ↑ (Q2 cycle) to Port Input Invalid (I/O in hold time) | PIC16**F**818/819 | 100 | — | — | ns | |
| | | | PIC16**LF**818/819 | 200 | — | — | ns | |
| 19* | TioV2osH | Port Input Valid to OSC1 ↑ (I/O in setup time) | | 0 | — | — | ns | |
| 20* | TioR | Port Output Rise Time | PIC16**F**818/819 | — | 10 | 40 | ns | |
| | | | PIC16**LF**818/819 | — | — | 145 | ns | |
| 21* | TioF | Port Output Fall Time | PIC16**F**818/819 | — | 10 | 40 | ns | |
| | | | PIC16**LF**818/819 | — | — | 145 | ns | |
| 22††* | Tinp | INT pin High or Low Time | | Tcy | — | — | ns | |
| 23††* | Trbp | RB7:RB4 Change INT High or Low Time | | Tcy | — | — | ns | |

* 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.

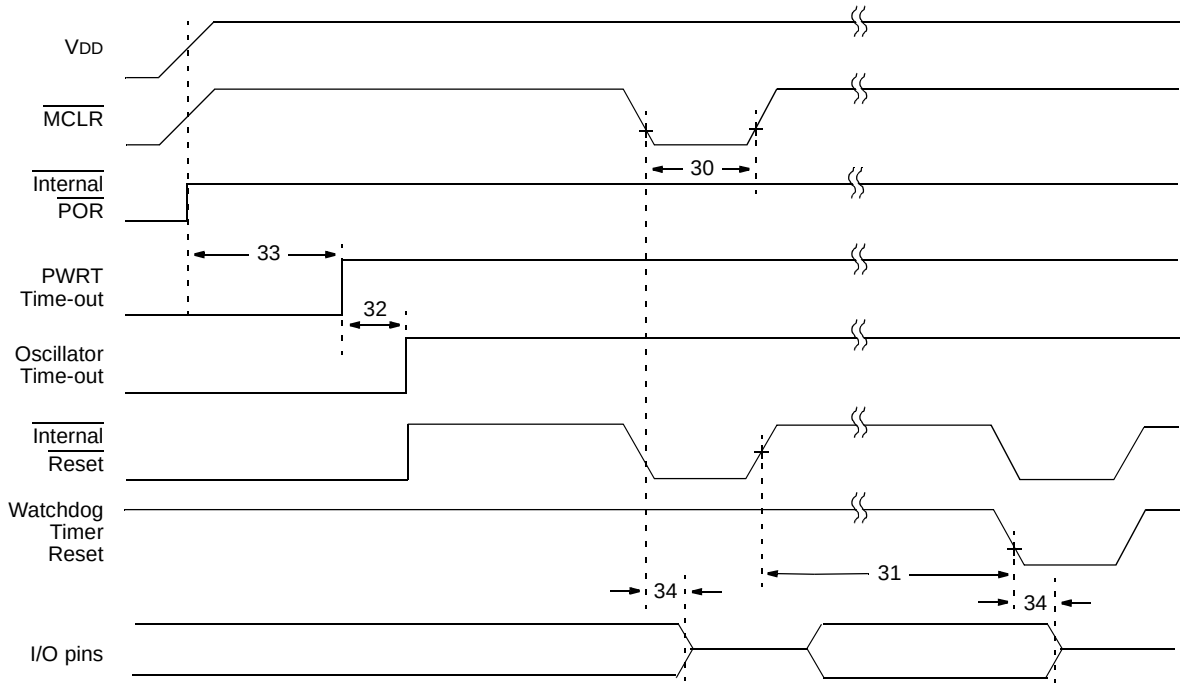
†† These parameters are asynchronous events, not related to any internal clock edges.

**Note 1:** Measurements are taken in RC mode, where CLKO output is 4 x Tosc.


© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

**FIGURE 15-6: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING**




**Note:** Refer to Figure 15-3 for load conditions.

**FIGURE 15-7: BROWN-OUT RESET TIMING**

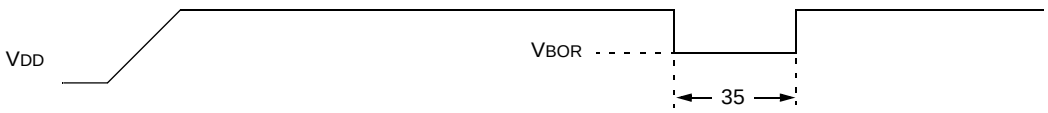



**TABLE 15-3: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER, POWER-UP TIMER AND BROWN-OUT RESET REQUIREMENTS**

| Param No. | Symbol | Characteristic | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|
| 30 | TMCL | MCLR Pulse Width (Low) | 2 | — | — | µs | VDD = 5V, -40°C to +85°C |
| 31* | TWDT | Watchdog Timer Time-out Period (no prescaler) | 13.6 | 16 | 18.4 | ms | VDD = 5V, -40°C to +85°C |
| 32 | TOST | Oscillation Start-up Timer Period | — | 1024 TOSC | — | — | TOSC = OSC1 period |
| 33* | TPWRT | Power-up Timer Period | 61.2 | 72 | 82.8 | ms | VDD = 5V, -40°C to +85°C |
| 34 | TIOZ | I/O High-Impedance from MCLR Low or Watchdog Timer Reset | — | — | 2.1 | µs | |
| 35 | TBOR | Brown-out Reset Pulse Width | 100 | — | — | µs | VDD ≤ VBOR (D005) |

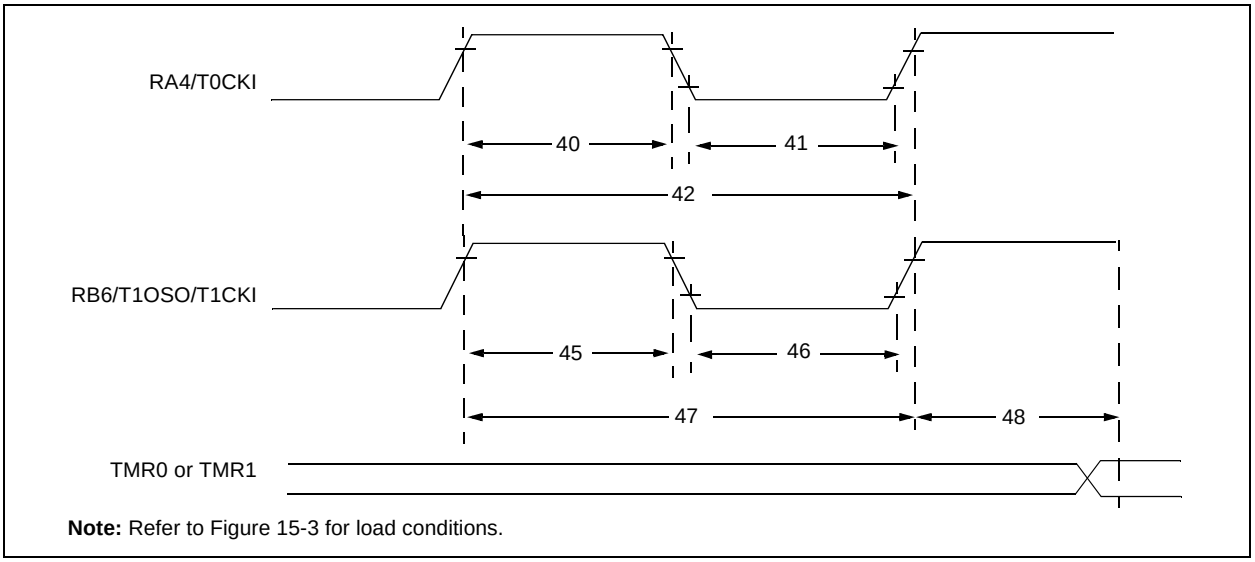
* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.



Downloaded from Arrow.com.

**FIGURE 15-8: TIMER0 AND TIMER1 EXTERNAL CLOCK TIMINGS**




**Note:** Refer to Figure 15-3 for load conditions.

**TABLE 15-4: TIMER0 AND TIMER1 EXTERNAL CLOCK REQUIREMENTS**

| Param No. | Symbol | Characteristic | | | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|---|---|
| 40* | TT0H | T0CKI High Pulse Width | | No Prescaler | 0.5 TCY + 20 | — | — | ns | Must also meet parameter 42 |
| | | | | With Prescaler | 10 | — | — | ns | |
| 41* | TT0L | T0CKI Low Pulse Width | | No Prescaler | 0.5 TCY + 20 | — | — | ns | Must also meet parameter 42 |
| | | | | With Prescaler | 10 | — | — | ns | |
| 42* | TT0P | T0CKI Period | | No Prescaler | TCY + 40 | — | — | ns | |
| | | | | With Prescaler | Greater of: 20 or TCY + 40 / N | — | — | ns | N = prescale value (2, 4, ..., 256) |
| 45* | TT1H | T1CKI High Time | Synchronous, Prescaler = 1 | | 0.5 TCY + 20 | — | — | ns | Must also meet parameter 47 |
| | | | Synchronous, Prescaler = 2,4,8 | PIC16**F**818/819 | 15 | — | — | ns | |
| | | | | PIC16**LF**818/819 | 25 | — | — | ns | |
| | | | Asynchronous | PIC16**F**818/819 | 30 | — | — | ns | |
| | | | | PIC16**LF**818/819 | 50 | — | — | ns | |
| 46* | TT1L | T1CKI Low Time | Synchronous, Prescaler = 1 | | 0.5 TCY + 20 | — | — | ns | Must also meet parameter 47 |
| | | | Synchronous, Prescaler = 2,4,8 | PIC16**F**818/819 | 15 | — | — | ns | |
| | | | | PIC16**LF**818/819 | 25 | — | — | ns | |
| | | | Asynchronous | PIC16**F**818/819 | 30 | — | — | ns | |
| | | | | PIC16**LF**818/819 | 50 | — | — | ns | |
| 47* | TT1P | T1CKI Input Period | Synchronous | PIC16**F**818/819 | Greater of: 30 or TCY + 40 / N | — | — | ns | N = prescale value (1, 2, 4, 8) |
| | | | | PIC16**LF**818/819 | Greater of: 50 or TCY + 40 / N | | | | N = prescale value (1, 2, 4, 8) |
| | | | Asynchronous | PIC16**F**818/819 | 60 | — | — | ns | |
| | | | | PIC16**LF**818/819 | 100 | — | — | ns | |
| | FT1 | Timer1 Oscillator Input Frequency Range (Oscillator enabled by setting bit T1OSCEN) | | | DC | — | 32.768 | kHz | |
| 48 | TCKEZTMR1 | Delay from External Clock Edge to Timer Increment | | | 2 TOSC | — | 7 TOSC | — | |

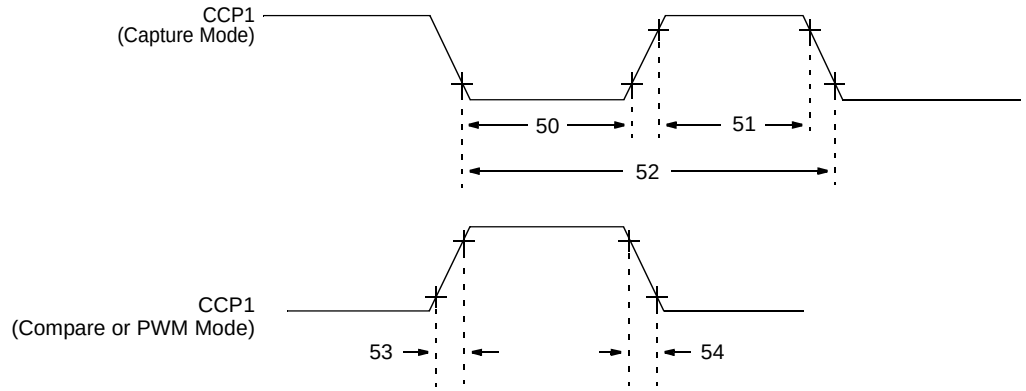
* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.


© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

**FIGURE 15-9: CAPTURE/COMPARE/PWM TIMINGS (CCP1)**




**Note:** Refer to Figure 15-3 for load conditions.

**TABLE 15-5: CAPTURE/COMPARE/PWM REQUIREMENTS (CCP1)**

| Param No. | Symbol | Characteristic | | | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|---|---|
| 50* | TccL | CCP1 Input Low Time | No Prescaler | | 0.5 TCY + 20 | — | — | ns | |
| | | | With Prescaler | PIC16**F**818/819 | 10 | — | — | ns | |
| | | | | PIC16**LF**818/819 | 20 | — | — | ns | |
| 51* | TccH | CCP1 Input High Time | No Prescaler | | 0.5 TCY + 20 | — | — | ns | |
| | | | With Prescaler | PIC16**F**818/819 | 10 | — | — | ns | |
| | | | | PIC16**LF**818/819 | 20 | — | — | ns | |
| 52* | TccP | CCP1 Input Period | | | $\frac{3\ T_{CY} + 40}{N}$ | — | — | ns | N = prescale value (1,4 or 16) |
| 53* | TccR | CCP1 Output Rise Time | | PIC16**F**818/819 | — | 10 | 25 | ns | |
| | | | | PIC16**LF**818/819 | — | 25 | 50 | ns | |
| 54* | TccF | CCP1 Output Fall Time | | PIC16**F**818/819 | — | 10 | 25 | ns | |
| | | | | PIC16**LF**818/819 | — | 25 | 45 | ns | |

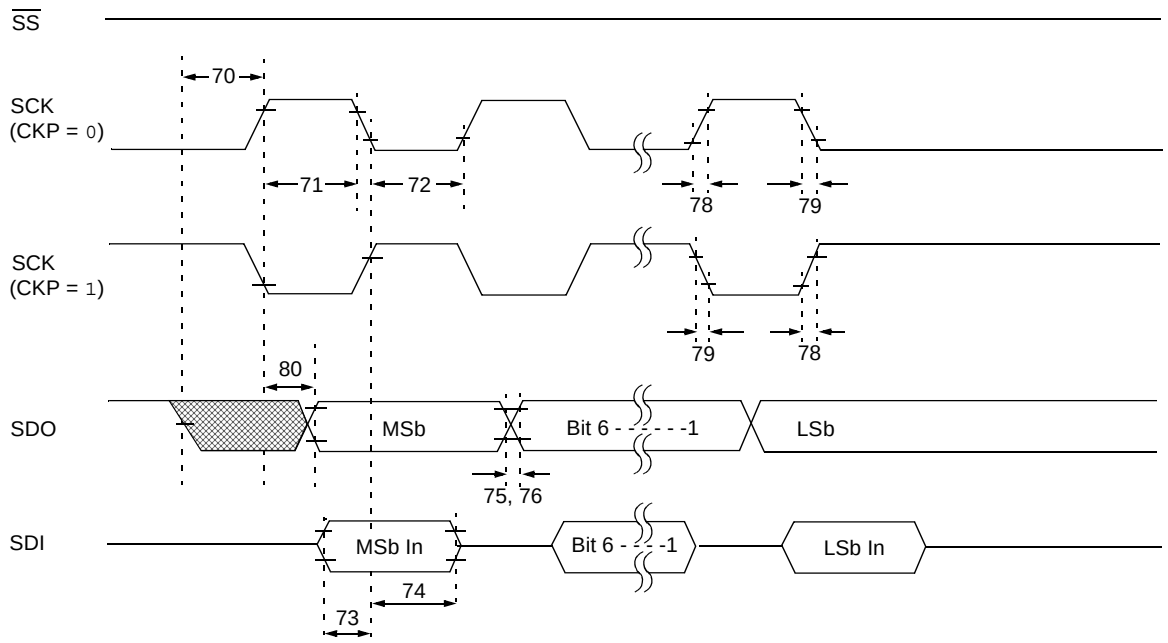
* These parameters are characterized but not tested.

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



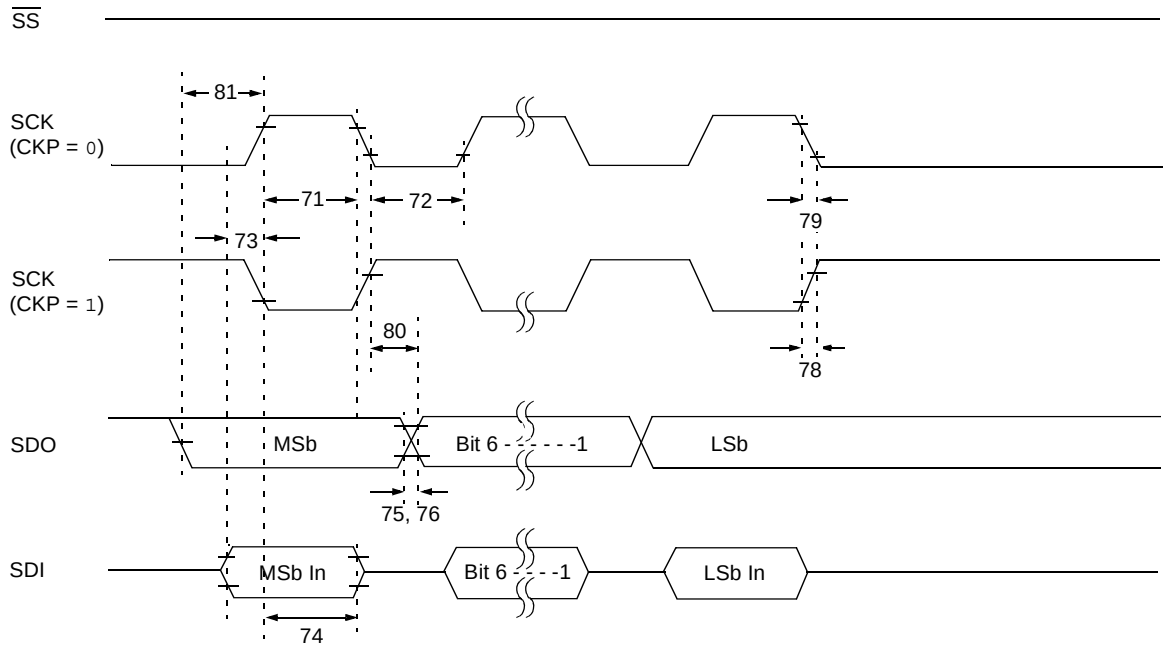
Downloaded from Arrow.com.

**FIGURE 15-10: SPI™ MASTER MODE TIMING (CKE = 0, SMP = 0)**




**Note:** Refer to Figure 15-3 for load conditions.

**FIGURE 15-11: SPI™ MASTER MODE TIMING (CKE = 1, SMP = 1)**

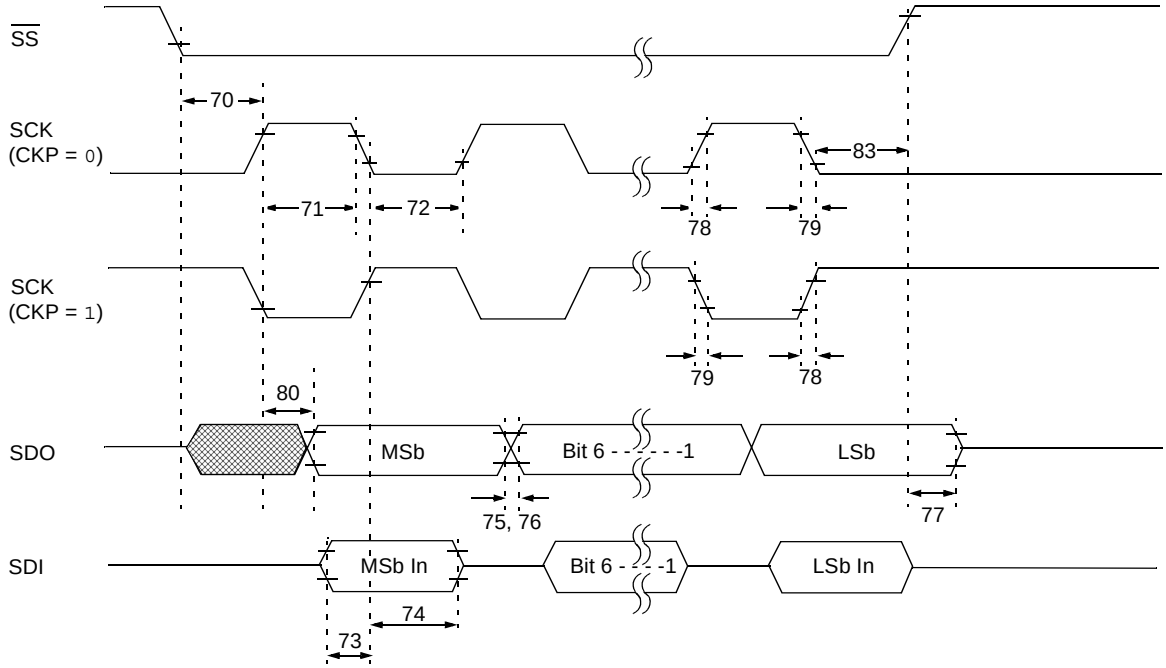



**Note:** Refer to Figure 15-3 for load conditions.


© 2004 Microchip Technology Inc.

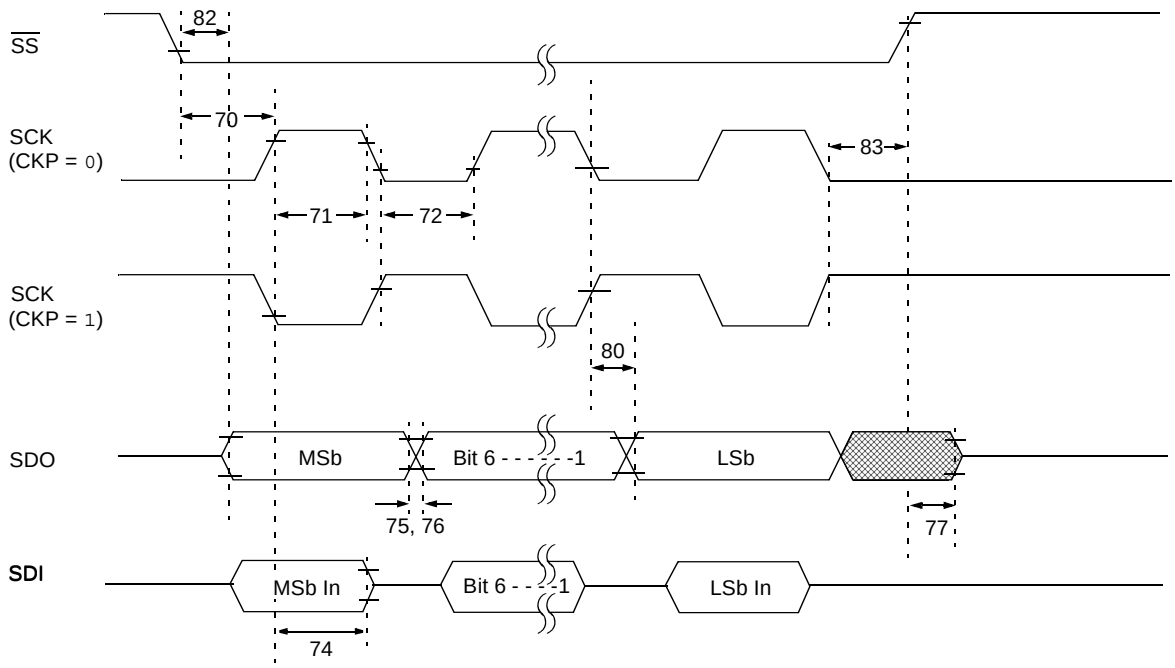
Downloaded from Arrow.com.

**FIGURE 15-12: SPI™ SLAVE MODE TIMING (CKE = 0)**




**Note:** Refer to Figure 15-3 for load conditions.

**FIGURE 15-13: SPI™ SLAVE MODE TIMING (CKE = 1)**




**Note:** Refer to Figure 15-3 for load conditions.



Downloaded from Arrow.com.

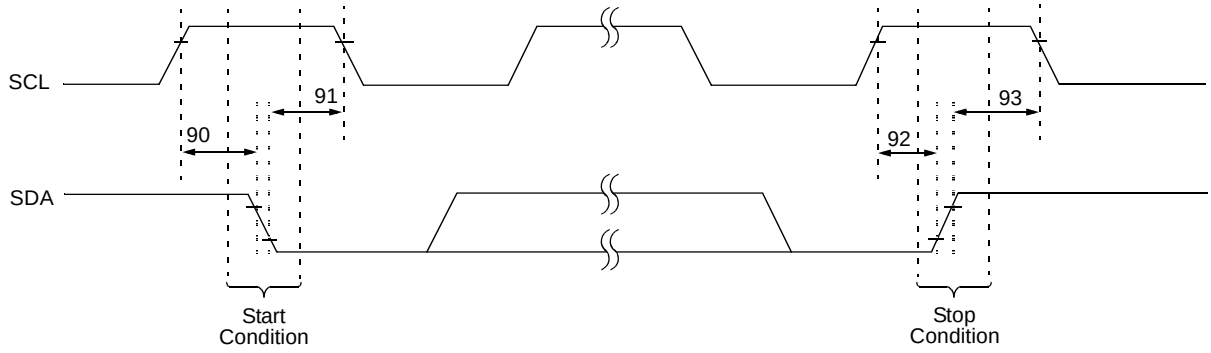
**TABLE 15-6: SPI™ MODE REQUIREMENTS**

| Param No. | Symbol | Characteristic | | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|---|
| 70* | TssL2scH, TssL2scL | $\overline{SS}$ ↓ to SCK ↓ or SCK ↑ Input | | Tcy | — | — | ns | |
| 71* | TscH | SCK Input High Time (Slave mode) | | Tcy + 20 | — | — | ns | |
| 72* | TscL | SCK Input Low Time (Slave mode) | | Tcy + 20 | — | — | ns | |
| 73* | TdiV2scH, TdiV2scL | Setup Time of SDI Data Input to SCK Edge | | 100 | — | — | ns | |
| 74* | TscH2diL, TscL2diL | Hold Time of SDI Data Input to SCK Edge | | 100 | — | — | ns | |
| 75* | TdoR | SDO Data Output Rise Time | PIC16**F**818/819<br>PIC16**LF**818/819 | —<br>— | 10<br>25 | 25<br>50 | ns<br>ns | |
| 76* | TdoF | SDO Data Output Fall Time | | — | 10 | 25 | ns | |
| 77* | TssH2doZ | $\overline{SS}$ ↑ to SDO Output High-Impedance | | 10 | — | 50 | ns | |
| 78* | TscR | SCK Output Rise Time (Master mode) | PIC16**F**818/819<br>PIC16**LF**818/819 | —<br>— | 10<br>25 | 25<br>50 | ns<br>ns | |
| 79* | TscF | SCK Output Fall Time (Master mode) | | — | 10 | 25 | ns | |
| 80* | TscH2doV, TscL2doV | SDO Data Output Valid after SCK Edge | PIC16**F**818/819<br>PIC16**LF**818/819 | —<br>— | —<br>— | 50<br>145 | ns<br>ns | |
| 81* | TdoV2scH, TdoV2scL | SDO Data Output Setup to SCK Edge | | Tcy | — | — | ns | |
| 82* | TssL2doV | SDO Data Output Valid after $\overline{SS}$ ↓ Edge | | — | — | 50 | ns | |
| 83* | TscH2ssH, TscL2ssH | $\overline{SS}$ ↑ after SCK Edge | | 1.5 Tcy + 40 | — | — | ns | |

* 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 15-14: I²C™ BUS START/STOP BITS TIMING**




**Note**: Refer to Figure 15-3 for load conditions.


© 2004 Microchip Technology Inc.

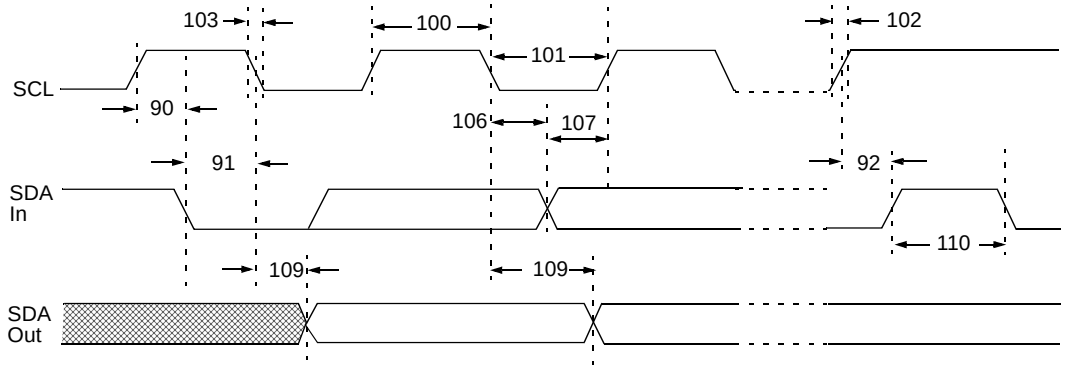
Downloaded from Arrow.com.

**TABLE 15-7: I²C™ BUS START/STOP BITS REQUIREMENTS**

| Param No. | Symbol | Characteristic | | Min | Typ | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|---|
| 90* | TSU:STA | Start Condition Setup Time | 100 kHz mode | 4700 | — | — | ns | Only relevant for Repeated Start condition |
| | | | 400 kHz mode | 600 | — | — | | |
| 91* | THD:STA | Start Condition Hold Time | 100 kHz mode | 4000 | — | — | ns | After this period, the first clock pulse is generated |
| | | | 400 kHz mode | 600 | — | — | | |
| 92* | TSU:STO | Stop Condition Setup Time | 100 kHz mode | 4700 | — | — | ns | |
| | | | 400 kHz mode | 600 | — | — | | |
| 93 | THD:STO | Stop Condition Hold Time | 100 kHz mode | 4000 | — | — | ns | |
| | | | 400 kHz mode | 600 | — | — | | |

* These parameters are characterized but not tested.

**FIGURE 15-15: I²C™ BUS DATA TIMING**




**Note:** Refer to Figure 15-3 for load conditions.



Downloaded from Arrow.com.

**TABLE 15-8: I²C™ BUS DATA REQUIREMENTS**

| Param. No. | Symbol | Characteristic | | Min | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|
| 100* | THIGH | Clock High Time | 100 kHz mode | 4.0 | — | µs | |
| | | | 400 kHz mode | 0.6 | — | µs | |
| | | | SSP Module | 1.5 TCY | — | | |
| 101* | TLOW | Clock Low Time | 100 kHz mode | 4.7 | — | µs | |
| | | | 400 kHz mode | 1.3 | — | µs | |
| | | | SSP Module | 1.5 TCY | — | | |
| 102* | TR | SDA and SCL Rise Time | 100 kHz mode | — | 1000 | ns | |
| | | | 400 kHz mode | 20 + 0.1 CB | 300 | ns | CB is specified to be from 10-400 pF |
| 103* | TF | SDA and SCL Fall Time | 100 kHz mode | — | 300 | ns | |
| | | | 400 kHz mode | 20 + 0.1 CB | 300 | ns | CB is specified to be from 10-400 pF |
| 90* | TSU:STA | Start Condition Setup Time | 100 kHz mode | 4.7 | — | µs | Only relevant for Repeated Start condition |
| | | | 400 kHz mode | 0.6 | — | µs | |
| 91* | THD:STA | Start Condition Hold Time | 100 kHz mode | 4.0 | — | µs | After this period, the first clock pulse is generated |
| | | | 400 kHz mode | 0.6 | — | µs | |
| 106* | THD:DAT | Data Input Hold Time | 100 kHz mode | 0 | — | ns | |
| | | | 400 kHz mode | 0 | 0.9 | µs | |
| 107* | TSU:DAT | Data Input Setup Time | 100 kHz mode | 250 | — | ns | **(Note 2)** |
| | | | 400 kHz mode | 100 | — | ns | |
| 92* | TSU:STO | Stop Condition Setup Time | 100 kHz mode | 4.7 | — | µs | |
| | | | 400 kHz mode | 0.6 | — | µs | |
| 109* | TAA | Output Valid from Clock | 100 kHz mode | — | 3500 | ns | **(Note 1)** |
| | | | 400 kHz mode | — | — | ns | |
| 110* | TBUF | Bus Free Time | 100 kHz mode | 4.7 | — | µs | Time the bus must be free before a new transmission can start |
| | | | 400 kHz mode | 1.3 | — | µs | |
| | CB | Bus Capacitive Loading | | — | 400 | pF | |

* These parameters are characterized but not tested.

**Note 1:** As a transmitter, the device must provide this internal minimum delay time to bridge the undefined region (min. 300 ns) of the falling edge of SCL to avoid unintended generation of Start or Stop conditions.

**2:** A Fast mode (400 kHz) I²C™ bus device can be used in a Standard mode (100 kHz) I²C bus system but the requirement, TSU:DAT ≥ 250 ns, must then be met. This will automatically be the case if the device does not stretch the LOW period of the SCL signal. If such a device does stretch the LOW period of the SCL signal, it must output the next data bit to the SDA line, TR max. + TSU:DAT = 1000 + 250 = 1250 ns (according to the Standard mode I²C bus specification), before the SCL line is released.


© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

**TABLE 15-9: A/D CONVERTER CHARACTERISTICS: PIC16F818/819 (INDUSTRIAL, EXTENDED) PIC16LF818/819 (INDUSTRIAL)**

| Param No. | Sym | Characteristic | | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|---|
| A01 | NR | Resolution | | — | — | 10-bits | bit | VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF |
| A03 | EIL | Integral Linearity Error | | — | — | <±1 | LSb | VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF |
| A04 | EDL | Differential Linearity Error | | — | — | <±1 | LSb | VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF |
| A06 | EOFF | Offset Error | | — | — | <±2 | LSb | VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF |
| A07 | EGN | Gain Error | | — | — | <±1 | LSb | VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF |
| A10 | — | Monotonicity | | — | guaranteed(3) | — | — | VSS ≤ VAIN ≤ VREF |
| A20 | VREF | Reference Voltage (VREF+ – VREF-) | | 2.0 | — | VDD + 0.3 | V | |
| A21 | VREF+ | Reference Voltage High | | AVDD – 2.5V | | AVDD + 0.3V | V | |
| A22 | VREF- | Reference Voltage Low | | AVSS – 0.3V | | VREF+ – 2.0V | V | |
| A25 | VAIN | Analog Input Voltage | | VSS – 0.3V | — | VREF + 0.3V | V | |
| A30 | ZAIN | Recommended Impedance of Analog Voltage Source | | — | — | 2.5 | kΩ | **(Note 4)** |
| A40 | IAD | A/D Conversion Current (VDD) | PIC16**F**818/819 | — | 220 | — | µA | Average current consumption when A/D is on **(Note 1)** |
| | | | PIC16**LF**818/819 | — | 90 | — | µA | |
| A50 | IREF | VREF Input Current **(Note 2)** | | — | — | 5 | µA | During VAIN acquisition. Based on differential of VHOLD to VAIN to charge CHOLD, see **Section 11.1 "A/D Acquisition Requirements"**. |
| | | | | — | — | 150 | µA | During A/D conversion cycle |

† 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.

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

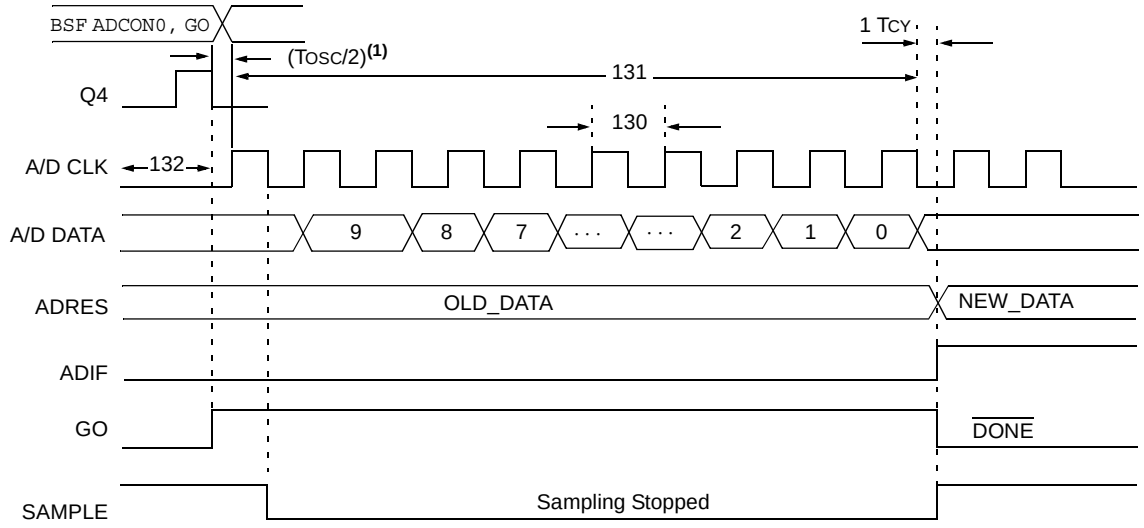
**3:** The A/D conversion result never decreases with an increase in the input voltage and has no missing codes.

**4:** Maximum allowed impedance for analog voltage source is 10 kΩ. This requires higher acquisition time.



Downloaded from Arrow.com.

**FIGURE 15-16: A/D CONVERSION TIMING**




**Note:** 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 15-10: A/D CONVERSION REQUIREMENTS**

| Param No. | Symbol | Characteristic | | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|---|
| 130 | TAD | A/D Clock Period | PIC16**F**818/819 | 1.6 | — | — | µs | TOSC based, VREF ≥ 3.0V |
| | | | PIC16**LF**818/819 | 3.0 | — | — | µs | TOSC based, VREF ≥ 2.0V |
| | | | PIC16**F**818/819 | 2.0 | 4.0 | 6.0 | µs | A/D RC mode |
| | | | PIC16**LF**818/819 | 3.0 | 6.0 | 9.0 | µs | A/D RC mode |
| 131 | TCNV | Conversion Time (not including S/H time) **(Note 1)** | | | — | 12 | TAD | |
| 132 | TACQ | Acquisition Time | | **(Note 2)** | 40 | — | µs | |
| | | | | 10* | — | — | µs | The minimum time is the amplifier settling time. This may be used if the "new" input voltage has not changed by more than 1 LSb (i.e., 5.0 mV @ 5.12V) from the last sampled voltage (as stated on CHOLD). |
| 134 | TGO | Q4 to A/D Clock Start | | — | TOSC/2 § | — | — | If the A/D clock source is selected as RC, a time of TCY is added before the A/D clock starts. This allows the SLEEP instruction to be executed. |

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

§ This specification ensured by design.

**Note 1:** ADRES register may be read on the following TCY cycle.

**2:** See **Section 11.1 "A/D Acquisition Requirements"** for minimum conditions.


© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

## 16.0 DC AND AC CHARACTERISTICS GRAPHS AND TABLES

| **Note:** | The graphs and tables provided following this note are a statistical summary based on a limited number of samples and are provided for informational purposes only. The performance characteristics listed herein are not tested or guaranteed. In some graphs or tables, the data presented may be outside the specified operating range (e.g., outside specified power supply range) and therefore, outside the warranted range. |
|---|---|

"Typical" represents the mean of the distribution at 25°C. "Maximum" or "minimum" represents (mean + 3σ) or (mean – 3σ) respectively, where σ is a standard deviation, over the whole temperature range.

**FIGURE 16-1: TYPICAL IDD vs. FOSC OVER VDD (HS MODE)**

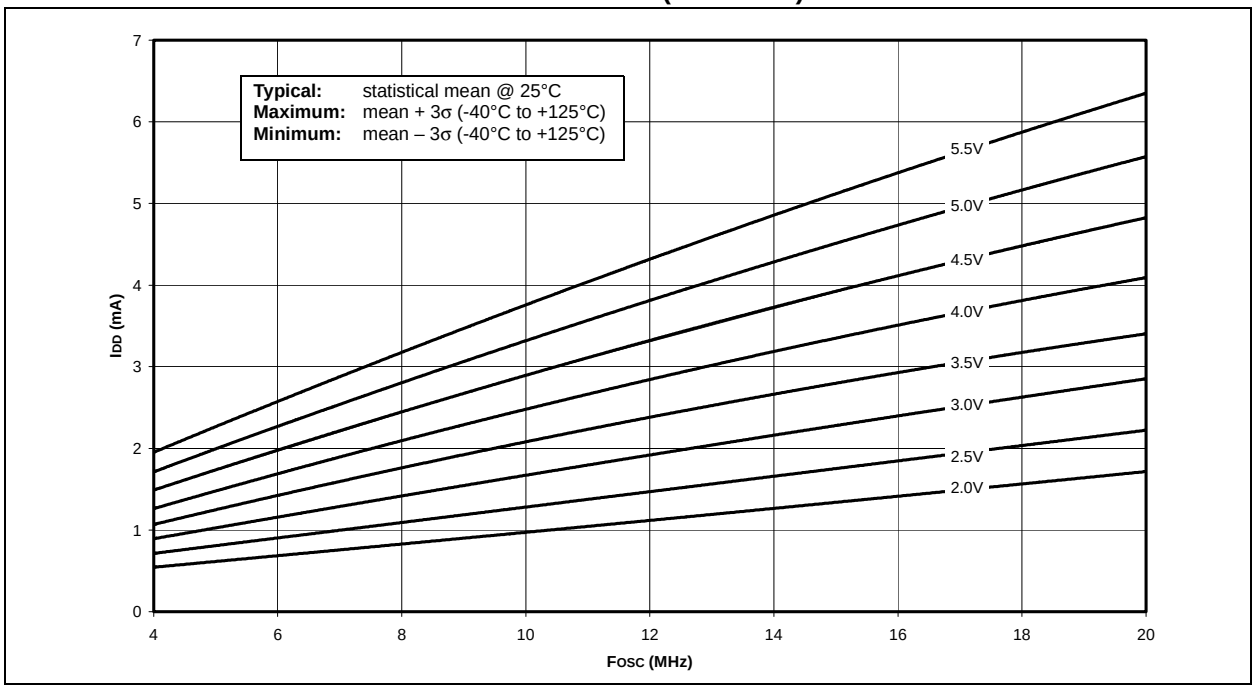
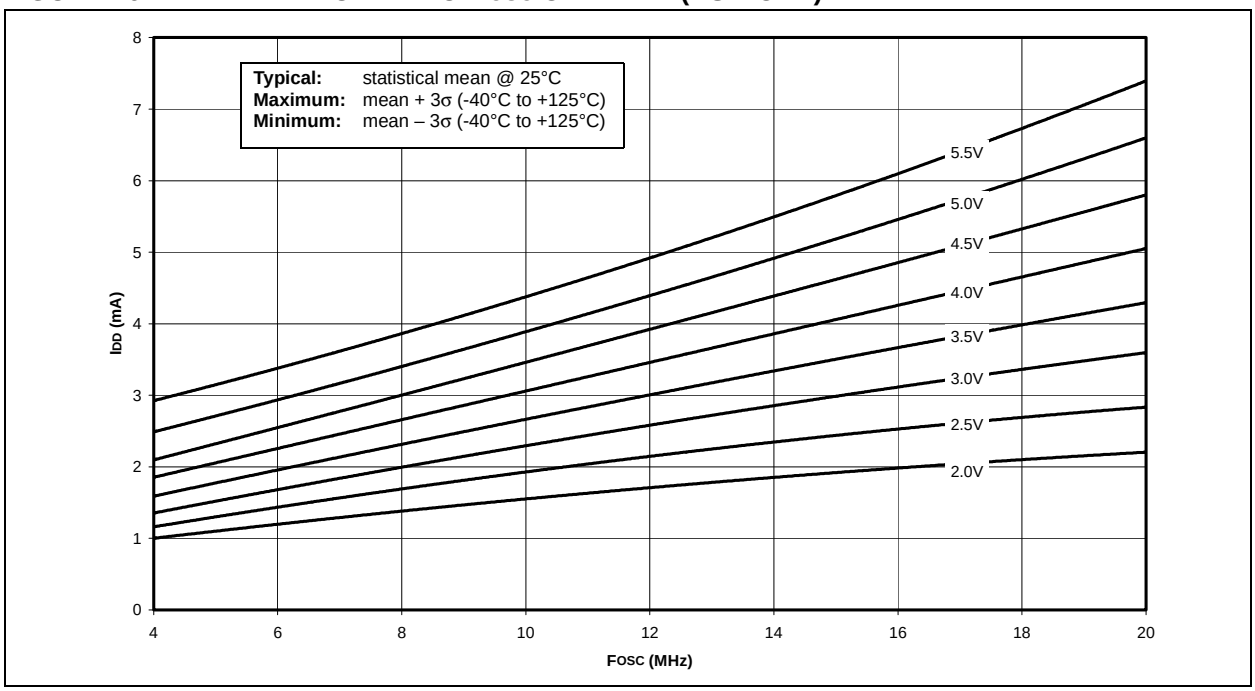



**FIGURE 16-2: MAXIMUM IDD vs. FOSC OVER VDD (HS MODE)**






Downloaded from Arrow.com.

**FIGURE 16-3: TYPICAL IDD vs. FOSC OVER VDD (XT MODE)**

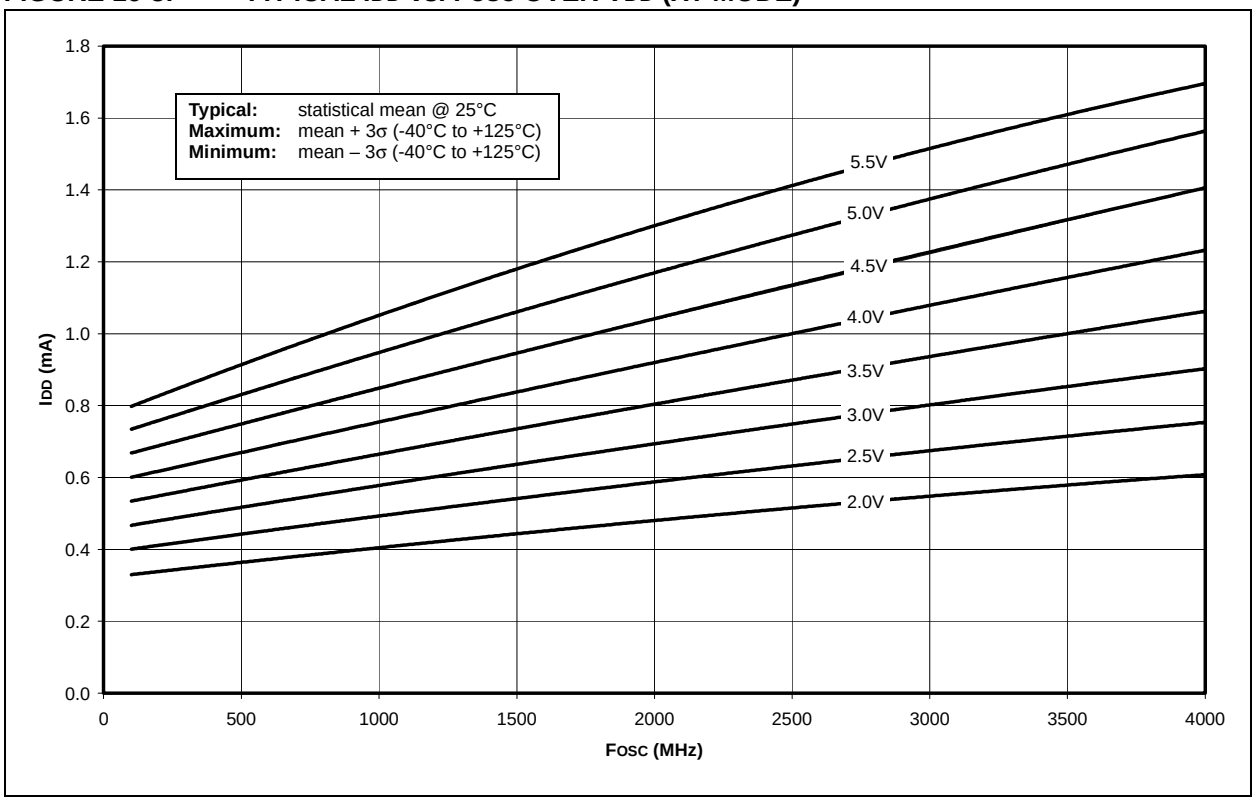
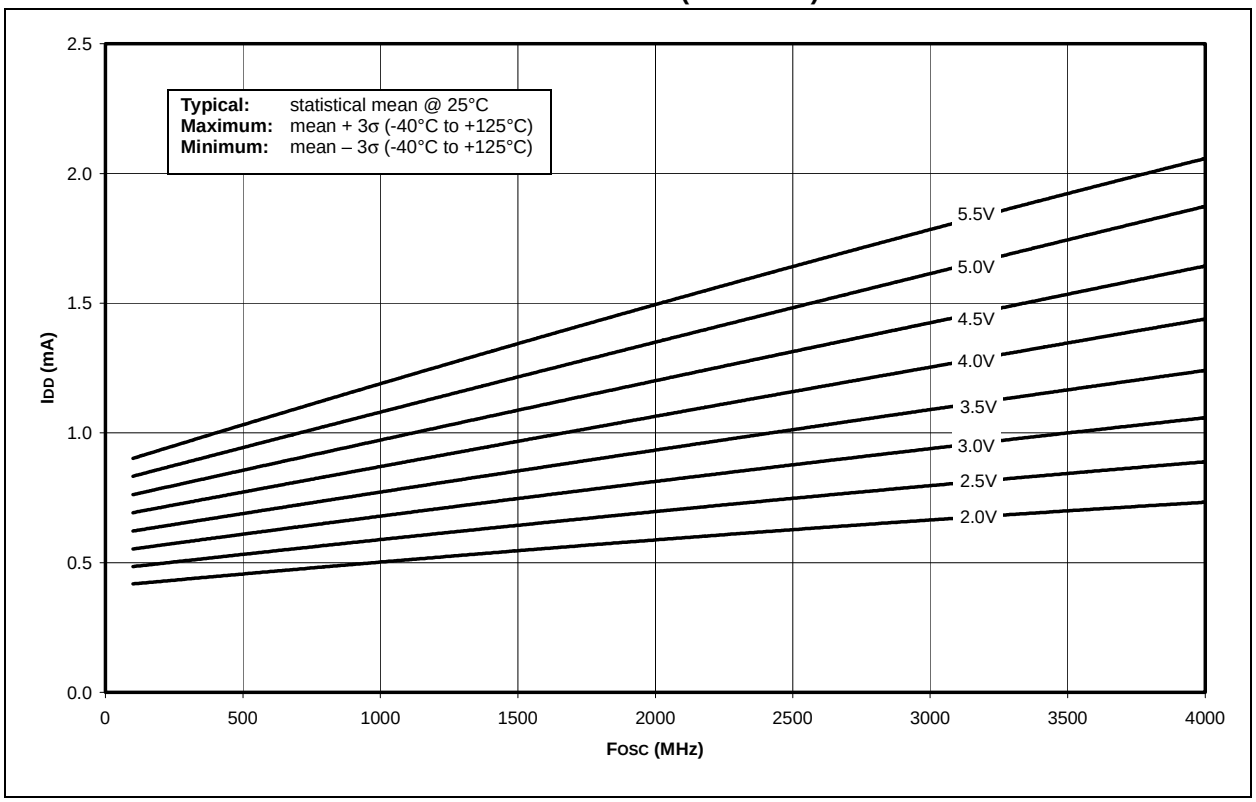



**FIGURE 16-4: MAXIMUM IDD vs. FOSC OVER VDD (XT MODE)**





© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

**FIGURE 16-5: TYPICAL IDD vs. FOSC OVER VDD (LP MODE)**

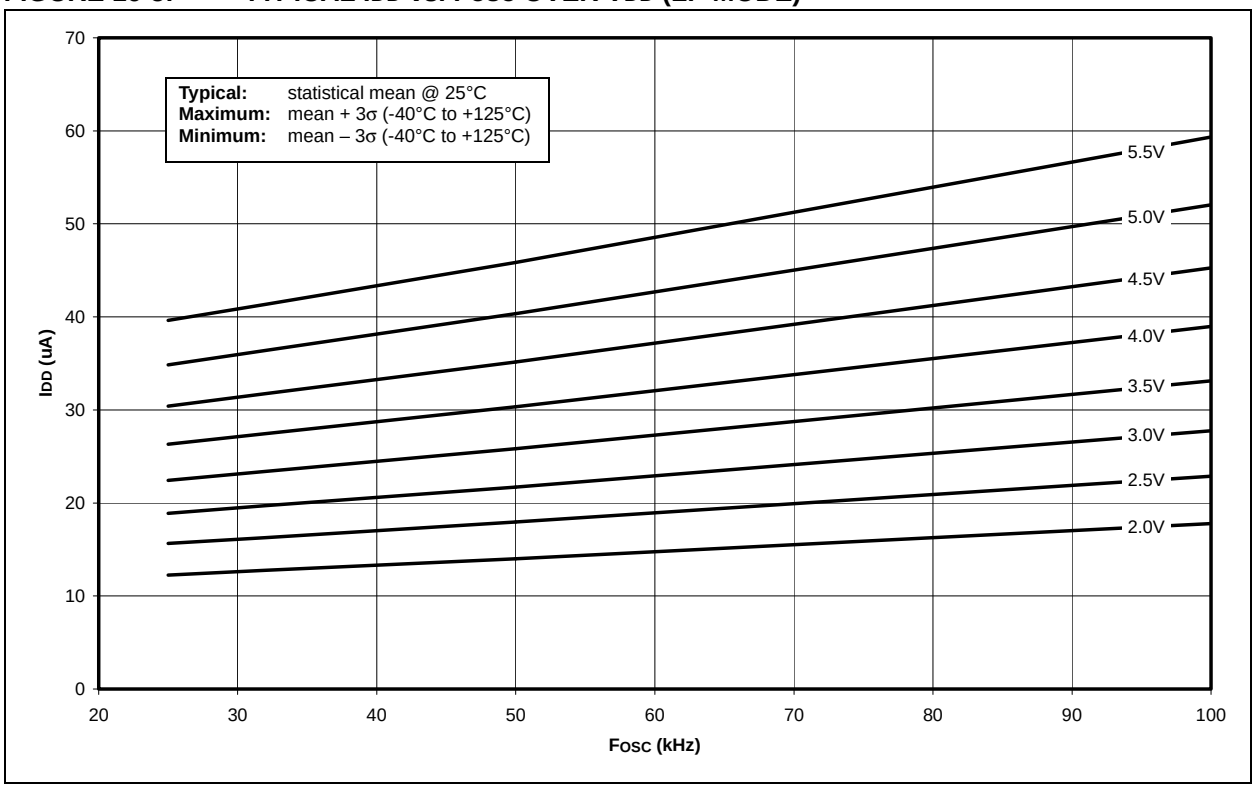
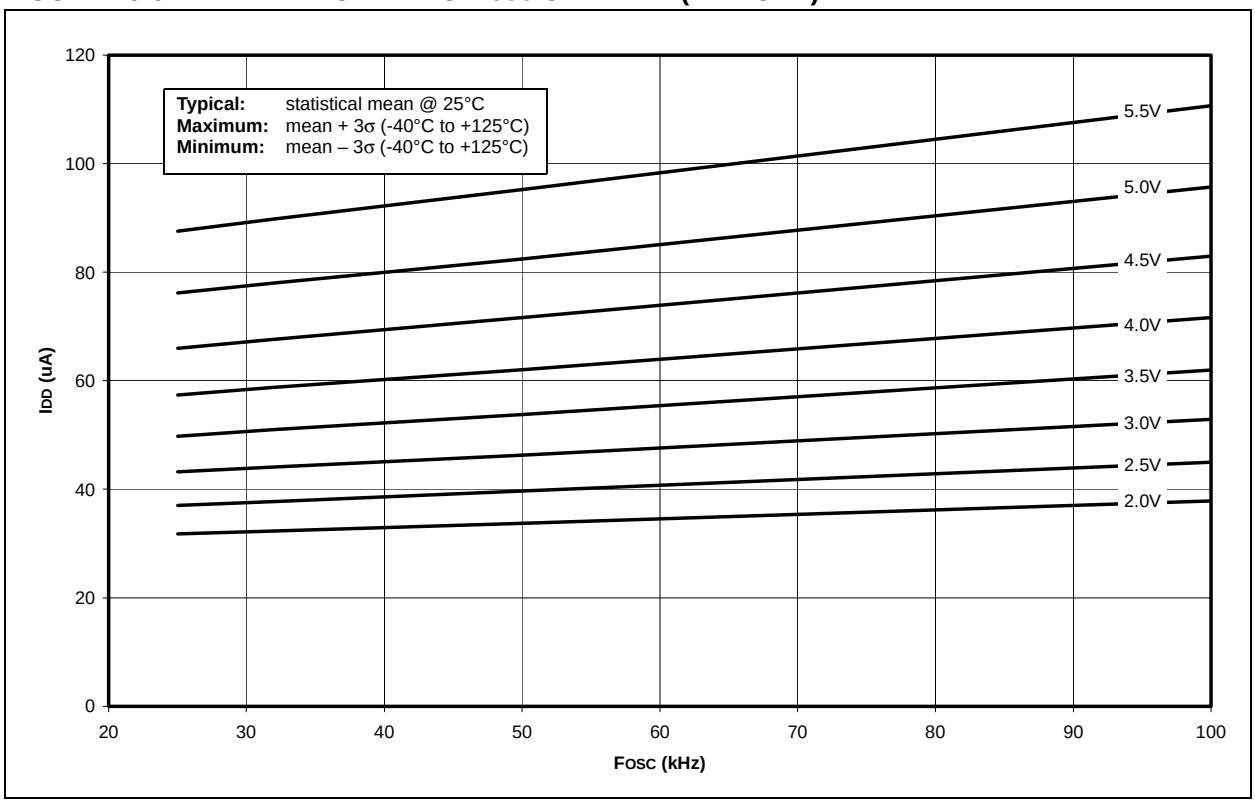



**FIGURE 16-6: MAXIMUM IDD vs. FOSC OVER VDD (LP MODE)**






Downloaded from Arrow.com.

**FIGURE 16-7: TYPICAL IDD vs. VDD, -40°C TO +125°C, 1 MHz TO 8 MHz (RC_RUN MODE, ALL PERIPHERALS DISABLED)**

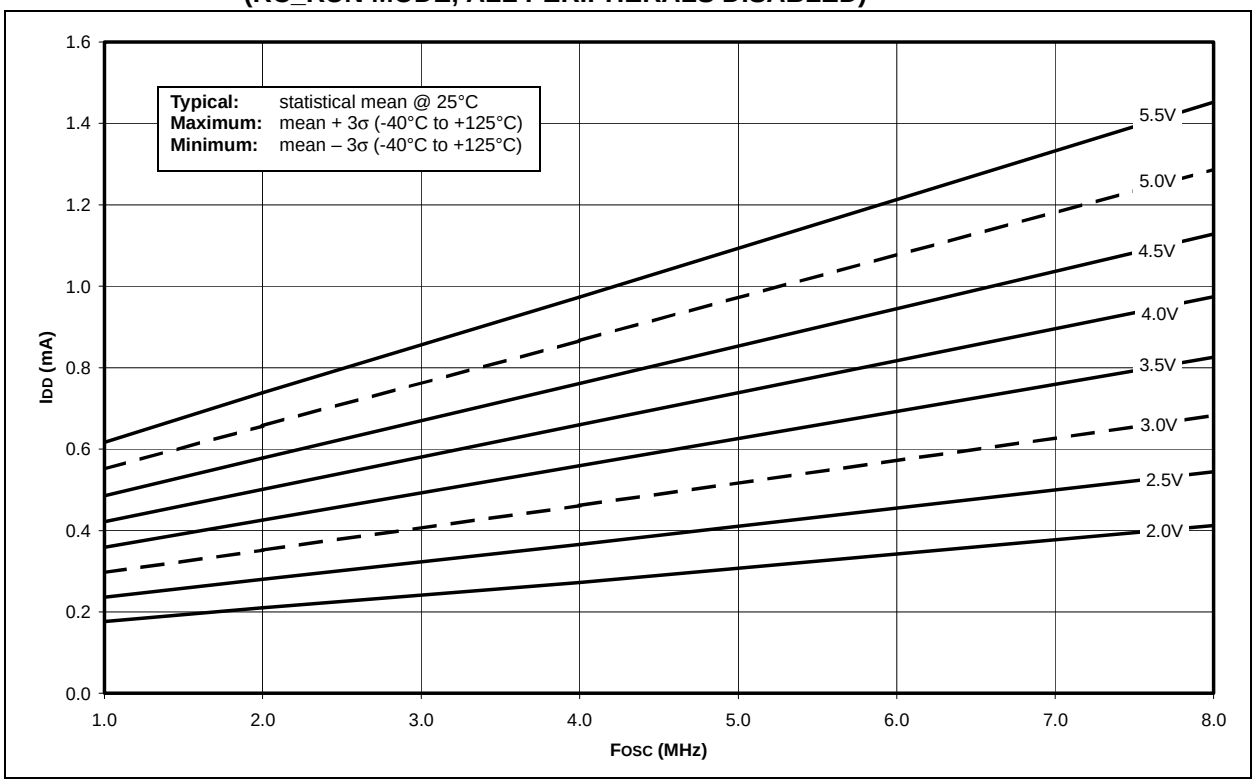
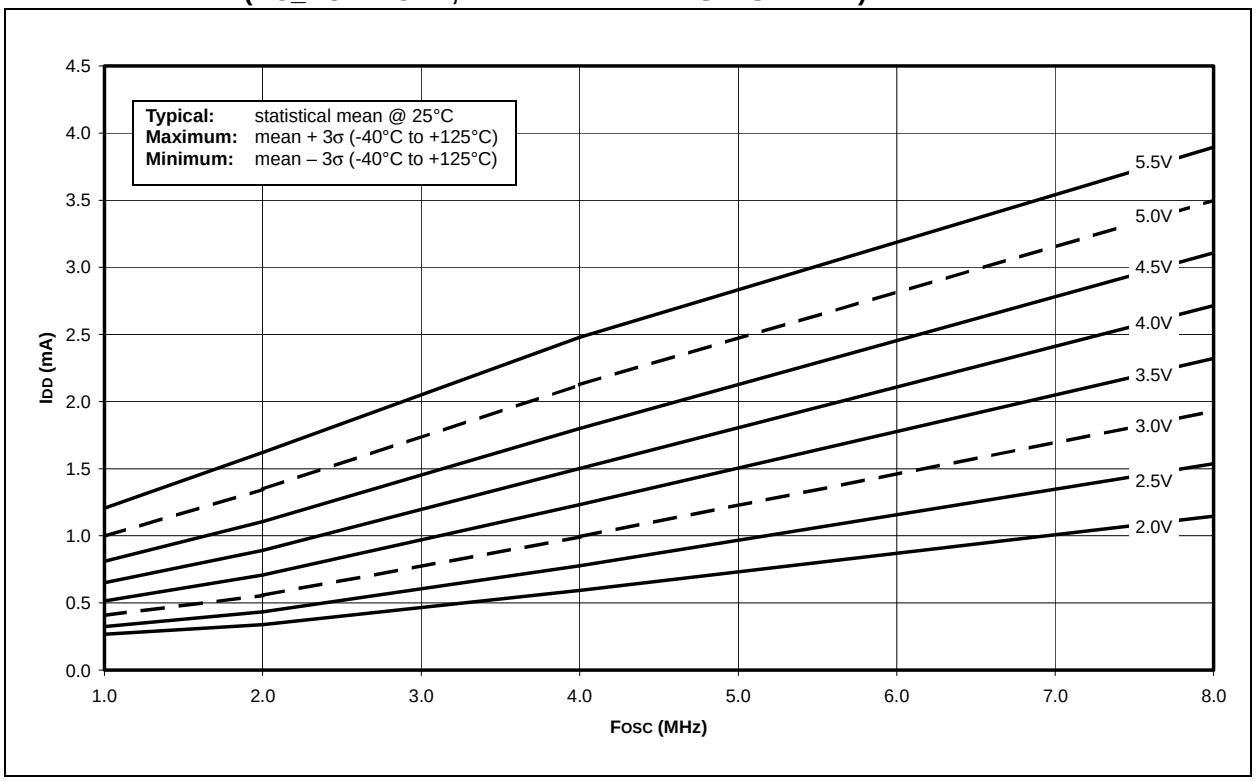



**FIGURE 16-8: MAXIMUM IDD vs. VDD, -40°C TO +125°C, 1 MHz TO 8 MHz (RC_RUN MODE, ALL PERIPHERALS DISABLED)**





© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

**FIGURE 16-9: IPD vs. VDD, -40°C TO +125°C (SLEEP MODE, ALL PERIPHERALS DISABLED)**

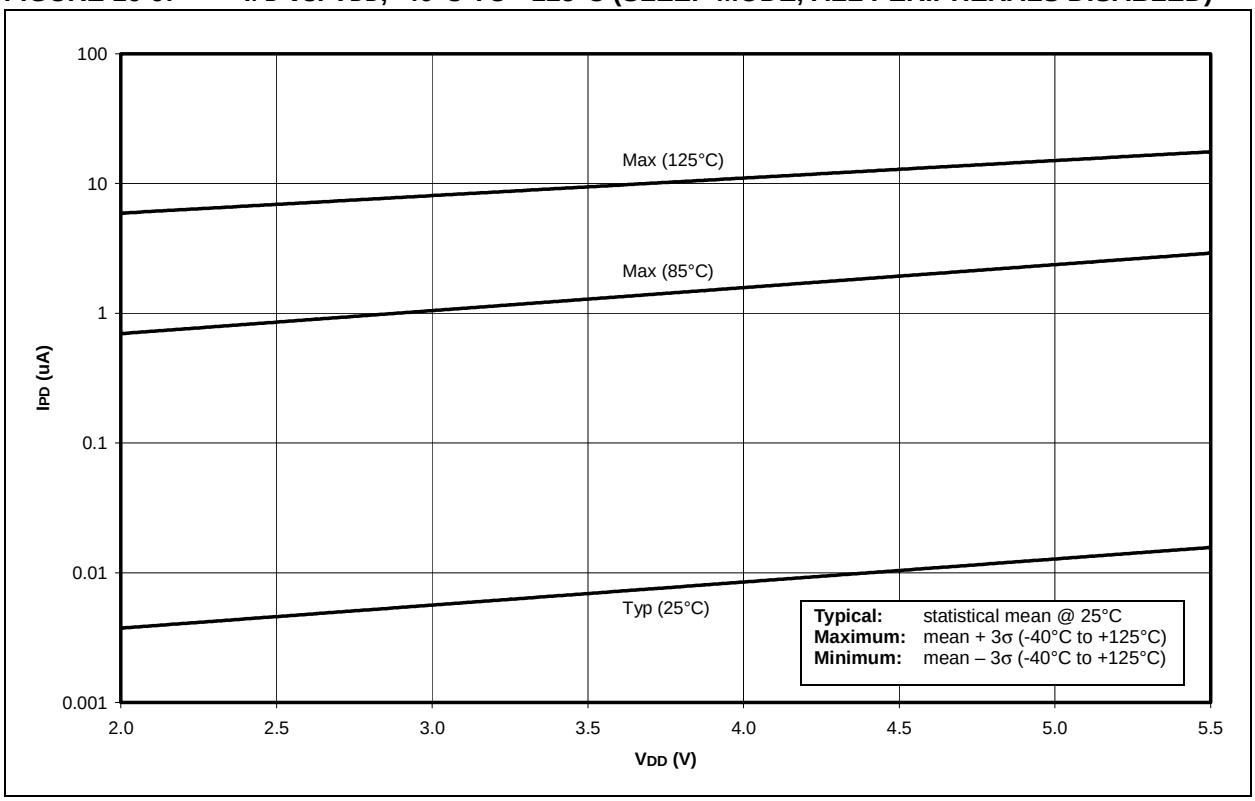
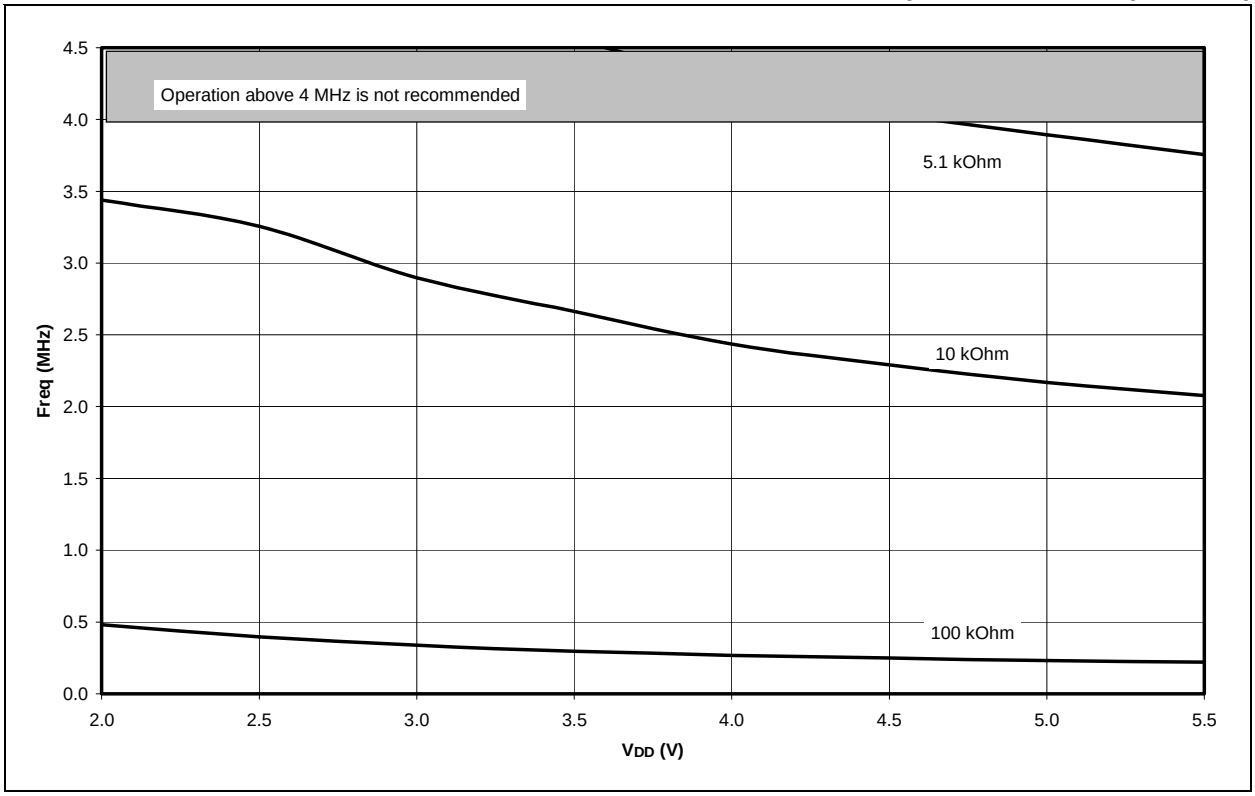



**FIGURE 16-10: AVERAGE FOSC vs. VDD FOR VARIOUS VALUES OF R (RC MODE, C = 20 pF, +25°C)**






Downloaded from Arrow.com.

**FIGURE 16-11: AVERAGE FOSC vs. VDD FOR VARIOUS VALUES OF R (RC MODE, C = 100 pF, +25°C)**

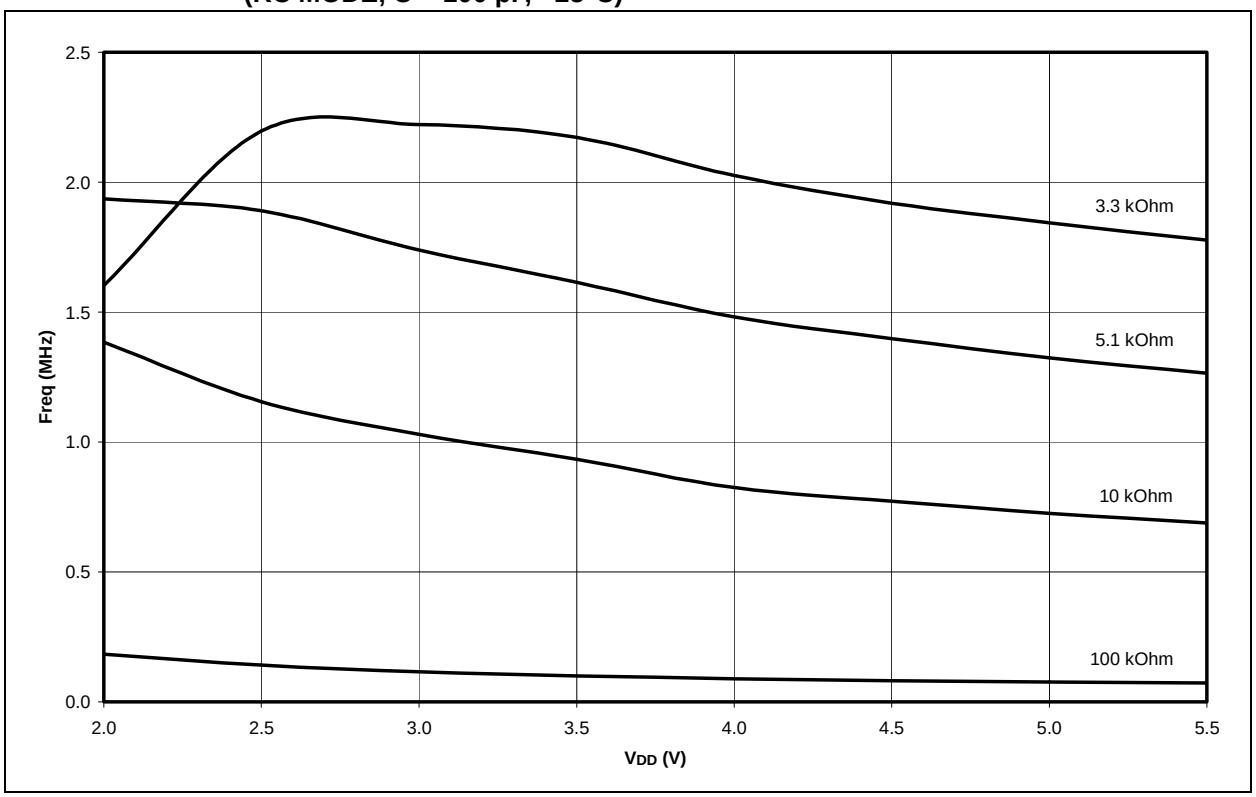
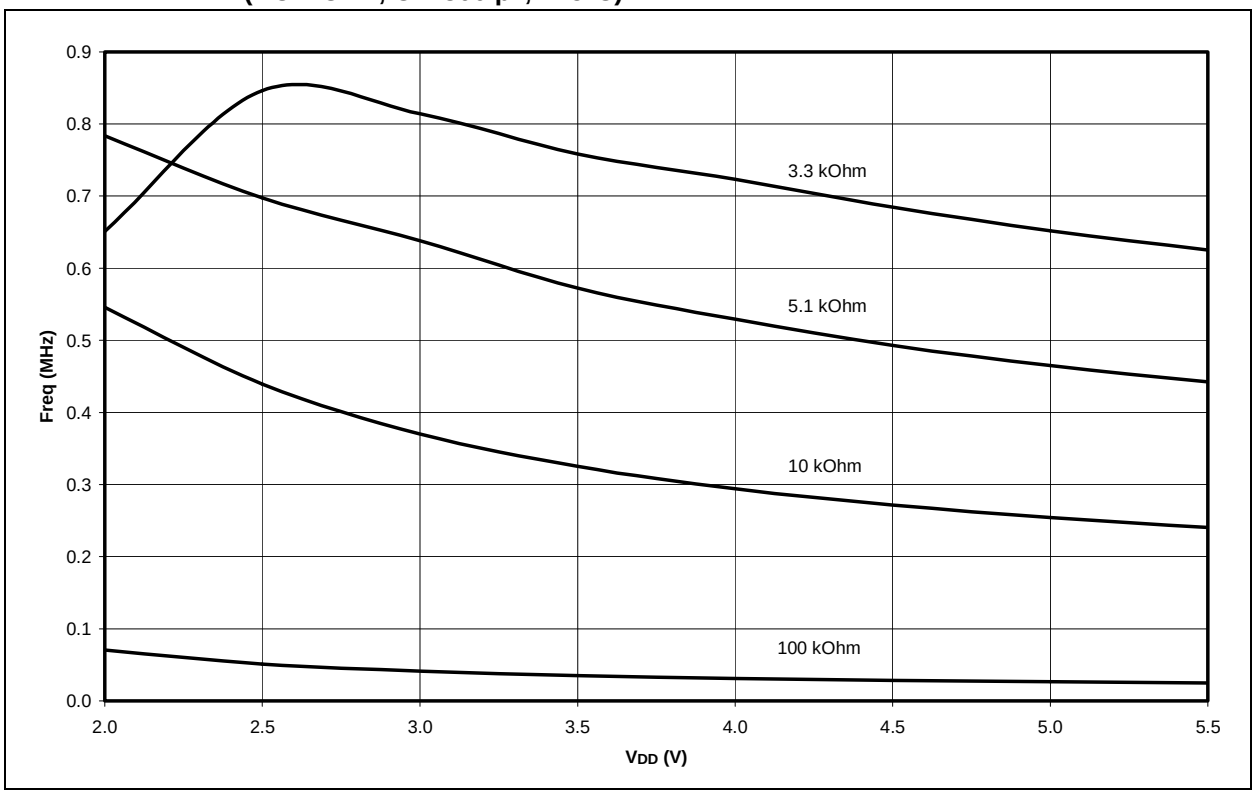



**FIGURE 16-12: AVERAGE FOSC vs. VDD FOR VARIOUS VALUES OF R (RC MODE, C = 300 pF, +25°C)**





© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

**FIGURE 16-13: ΔIPD TIMER1 OSCILLATOR, -10°C TO +70°C (SLEEP MODE, TMR1 COUNTER DISABLED)**

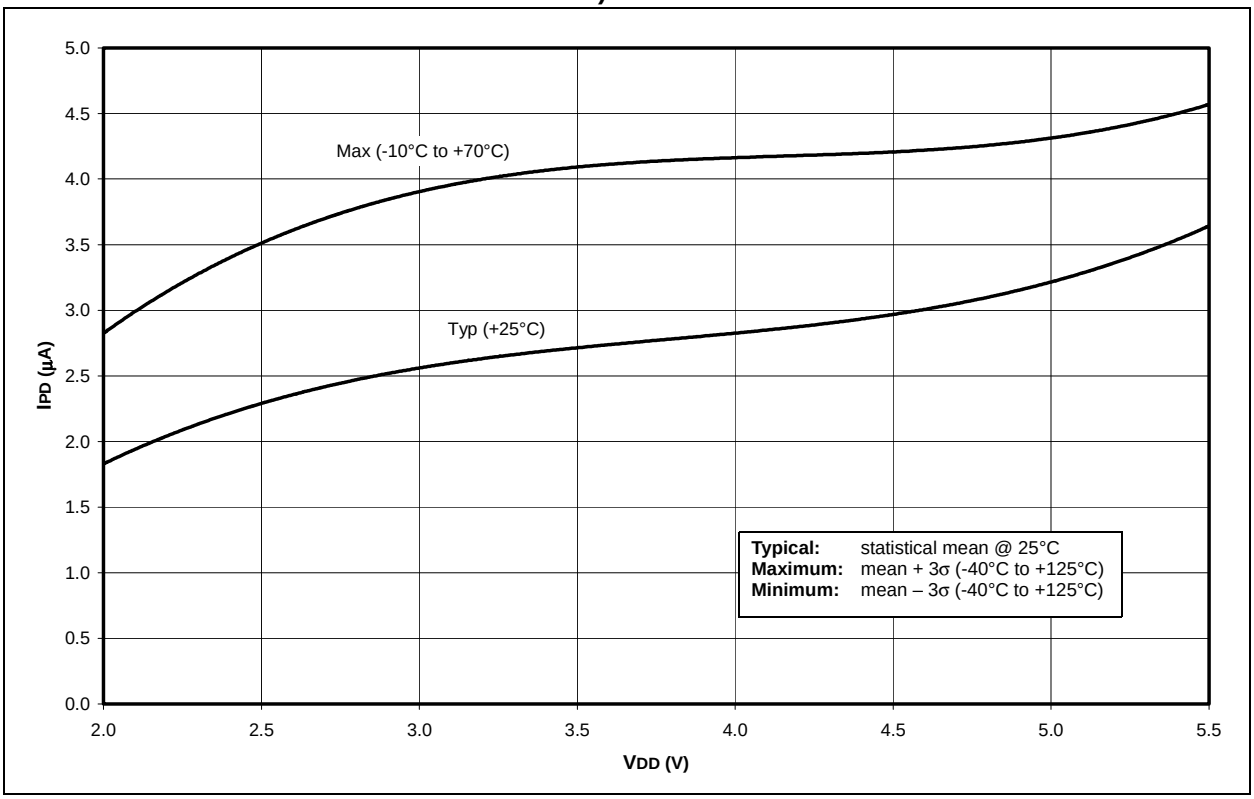
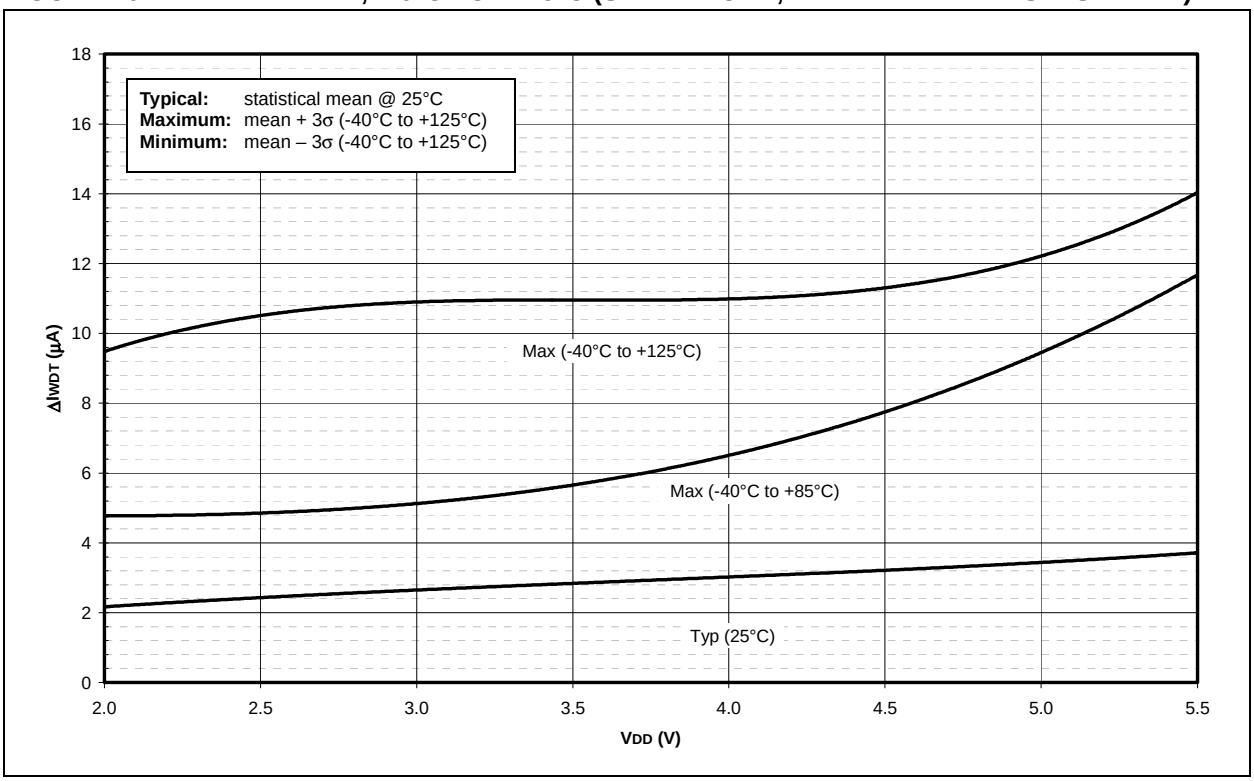



**FIGURE 16-14: ΔIPD WDT, -40°C TO +125°C (SLEEP MODE, ALL PERIPHERALS DISABLED)**






Downloaded from Arrow.com.

**FIGURE 16-15: ΔIPD BOR vs. VDD, -40°C TO +125°C (SLEEP MODE, BOR ENABLED AT 2.00V-2.16V)**

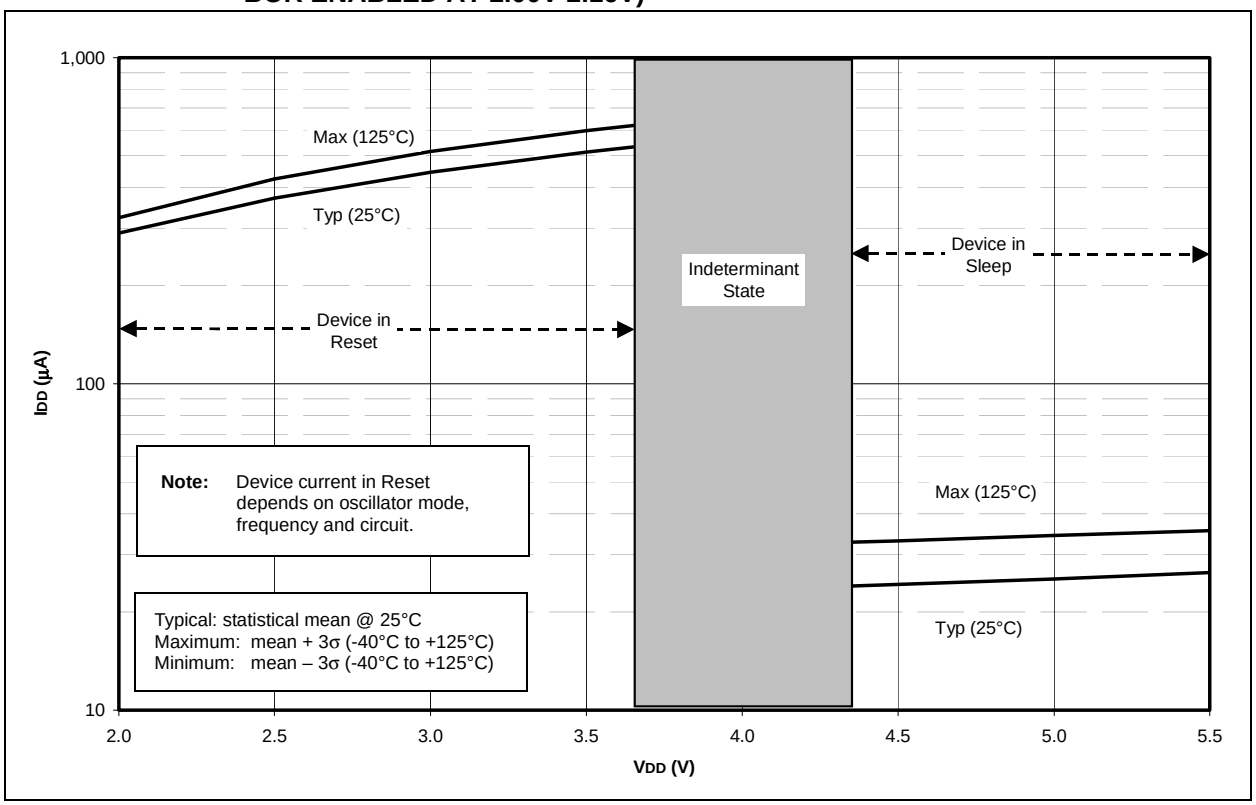
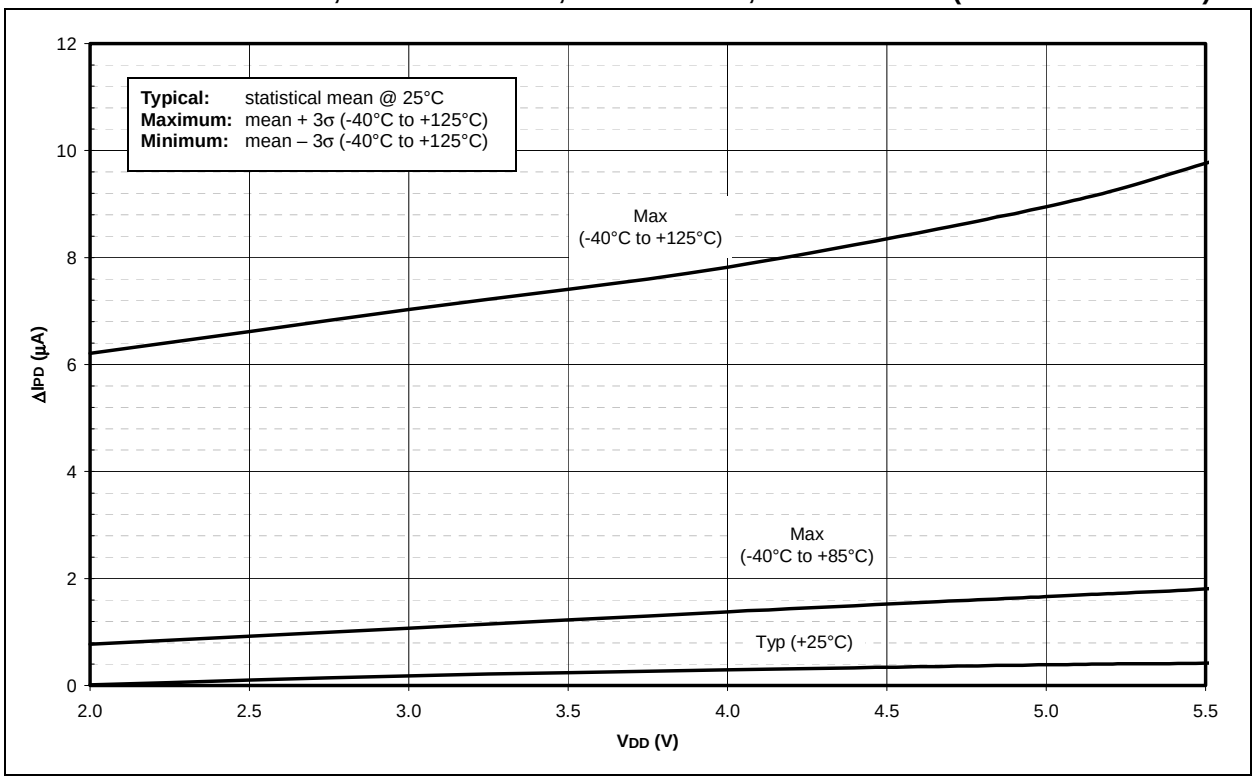



**FIGURE 16-16: IPD A/D, -40°C TO +125°C, SLEEP MODE, A/D ENABLED (NOT CONVERTING)**





© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

**FIGURE 16-17: TYPICAL, MINIMUM AND MAXIMUM $V_{OH}$ vs. $I_{OH}$ ($V_{DD}$ = 5V, -40°C TO +125°C)**

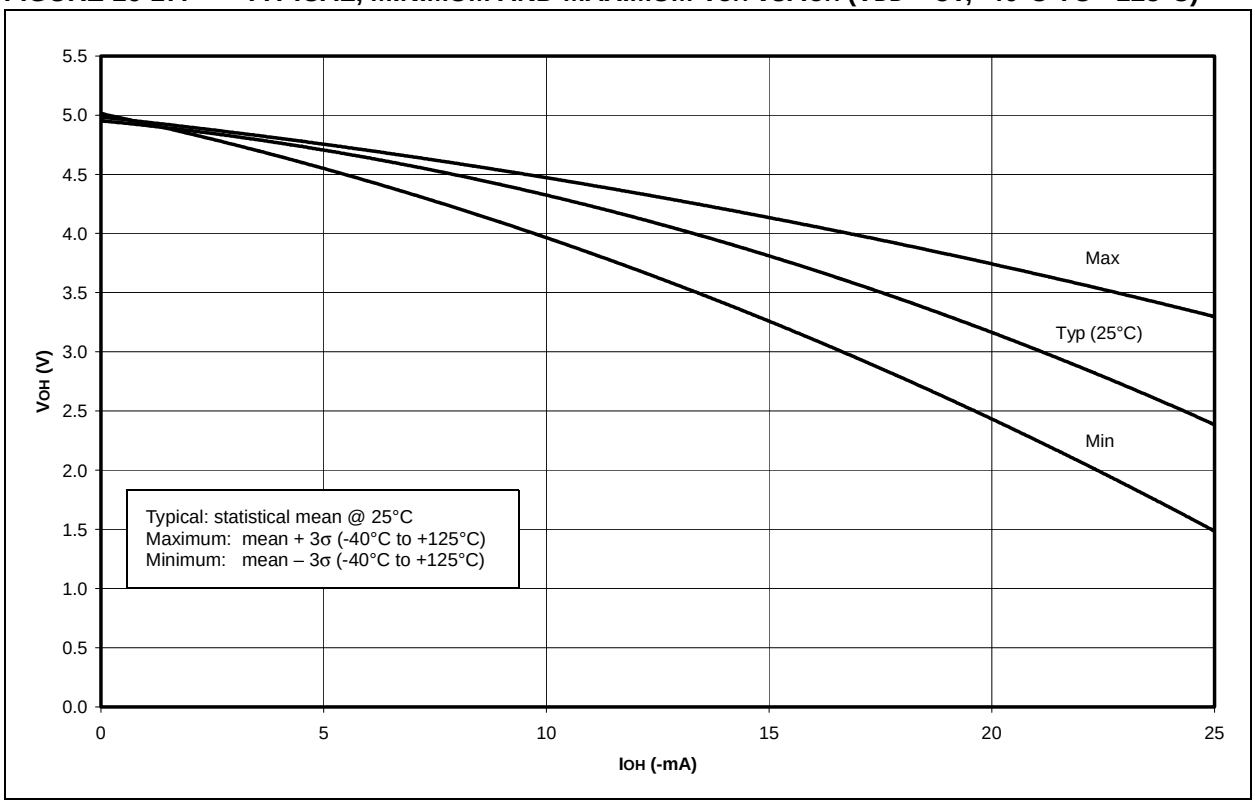
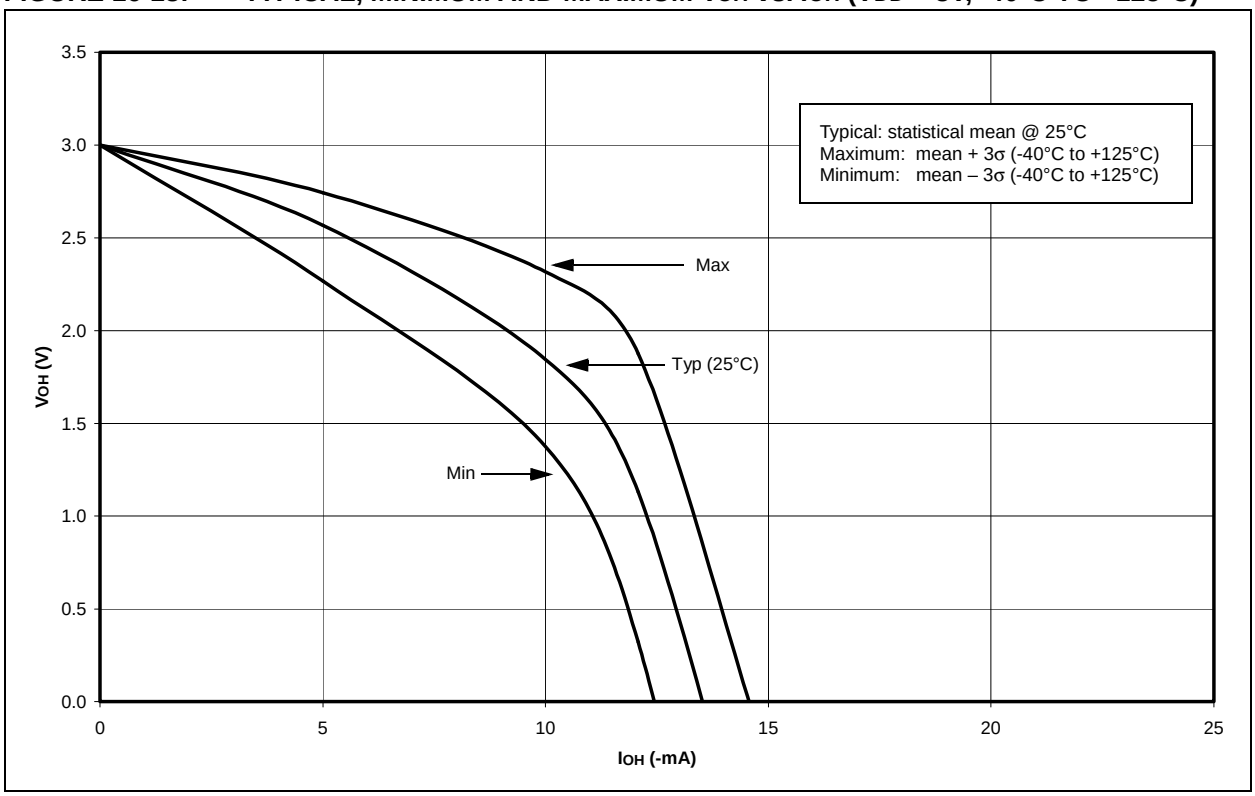



**FIGURE 16-18: TYPICAL, MINIMUM AND MAXIMUM $V_{OH}$ vs. $I_{OH}$ ($V_{DD}$ = 3V, -40°C TO +125°C)**






Downloaded from Arrow.com.

**FIGURE 16-19: TYPICAL, MINIMUM AND MAXIMUM VOL vs. IOL (VDD = 5V, -40°C TO +125°C)**

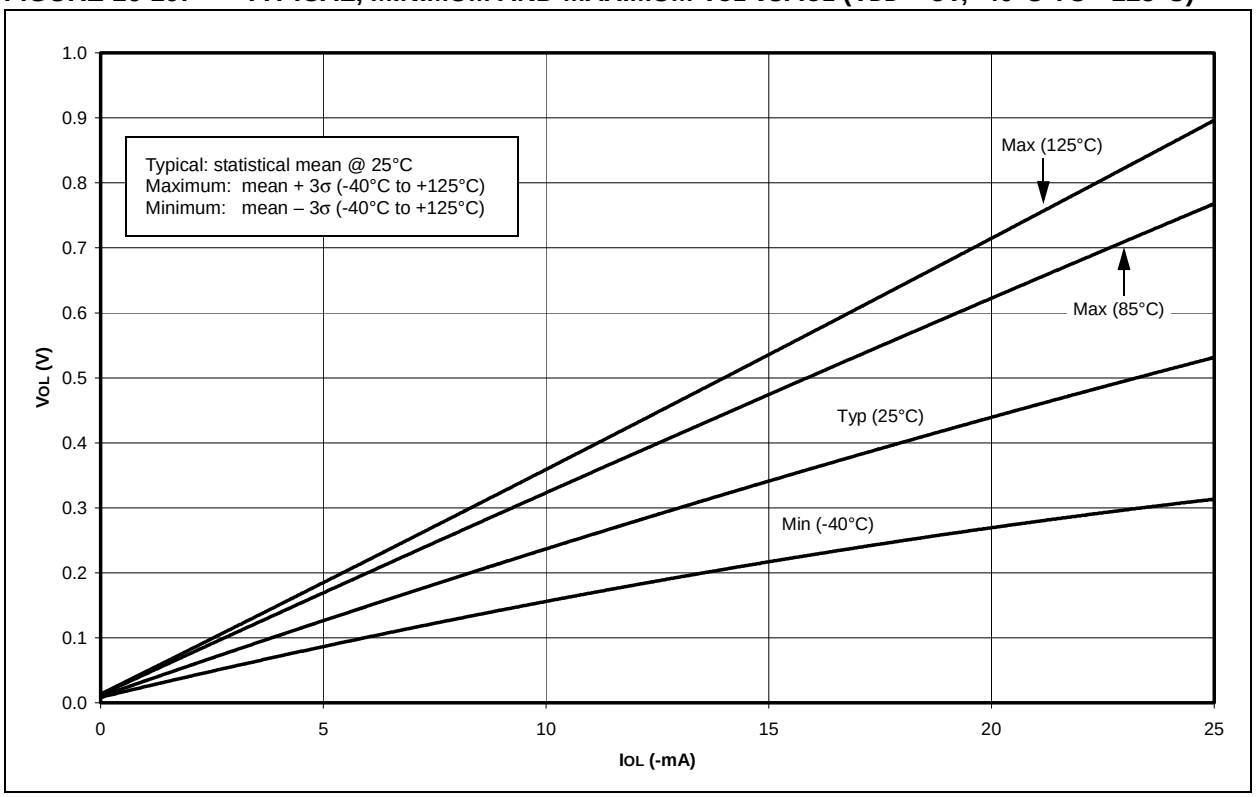
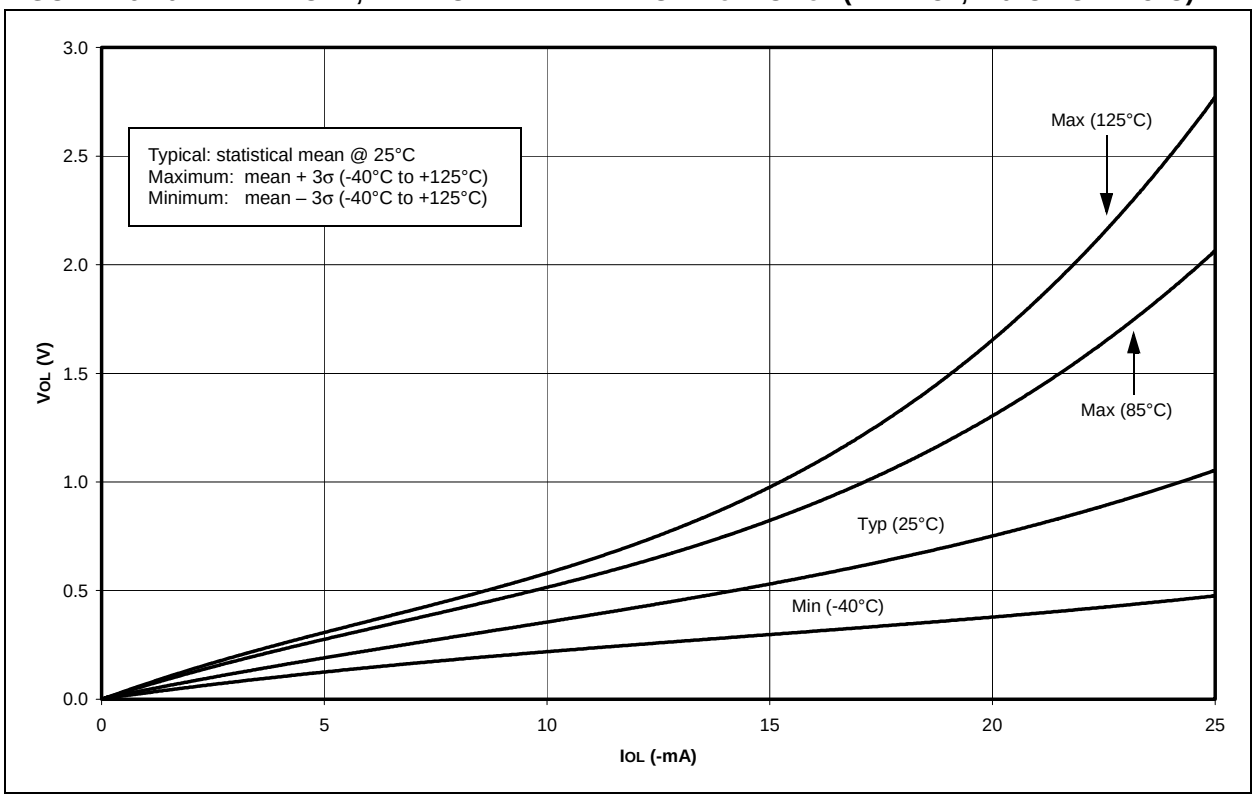



**FIGURE 16-20: TYPICAL, MINIMUM AND MAXIMUM VOL vs. IOL (VDD = 3V, -40°C TO +125°C)**





© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

**FIGURE 16-21: MINIMUM AND MAXIMUM VIN vs. VDD (TTL INPUT, -40°C TO +125°C)**

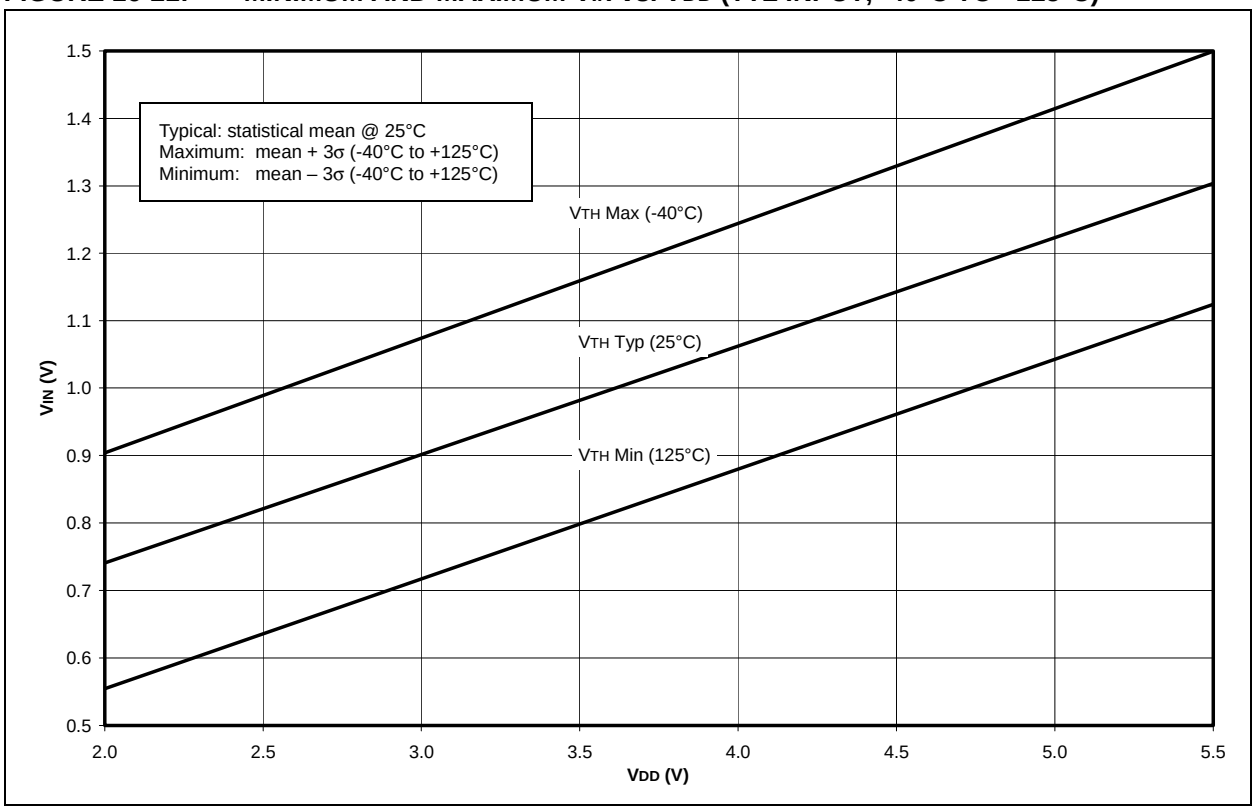
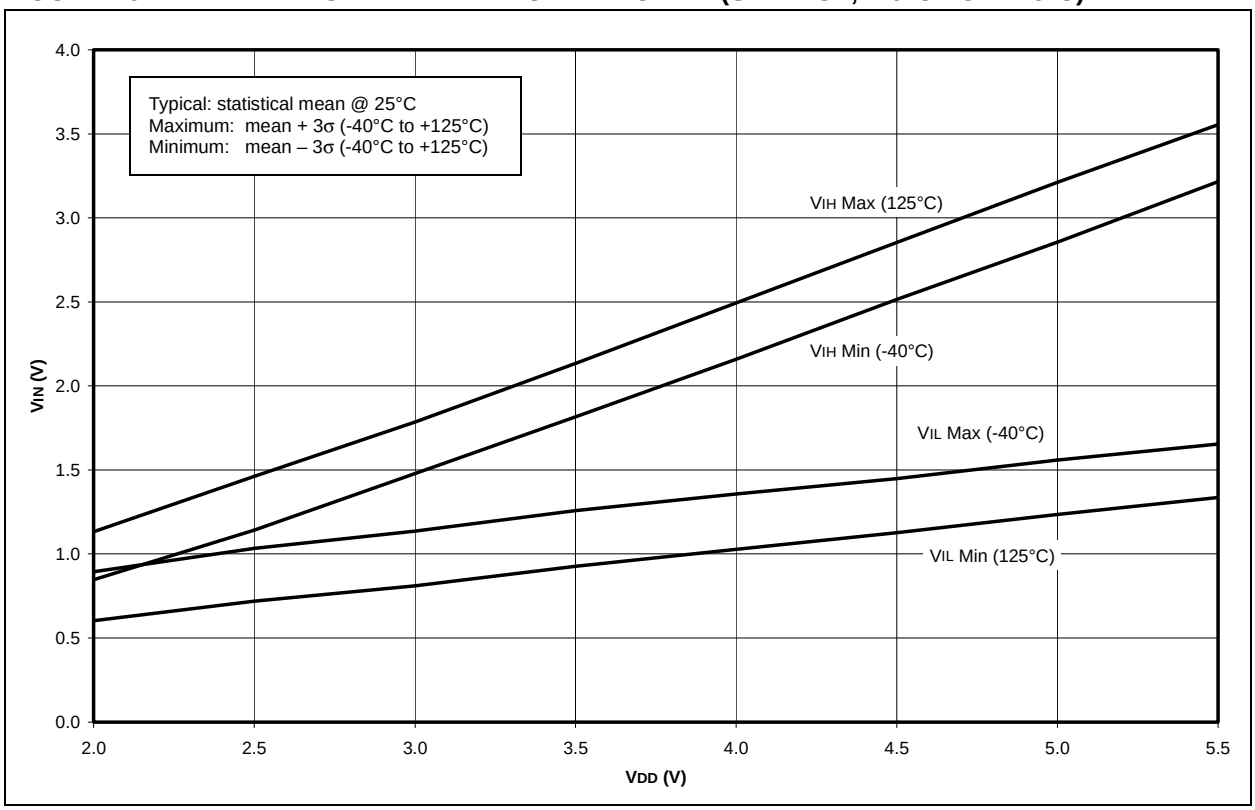



**FIGURE 16-22: MINIMUM AND MAXIMUM VIN vs. VDD (ST INPUT, -40°C TO +125°C)**






Downloaded from Arrow.com.

**FIGURE 16-23: MINIMUM AND MAXIMUM VIN vs. VDD (I2C™ INPUT, -40°C TO +125°C)**

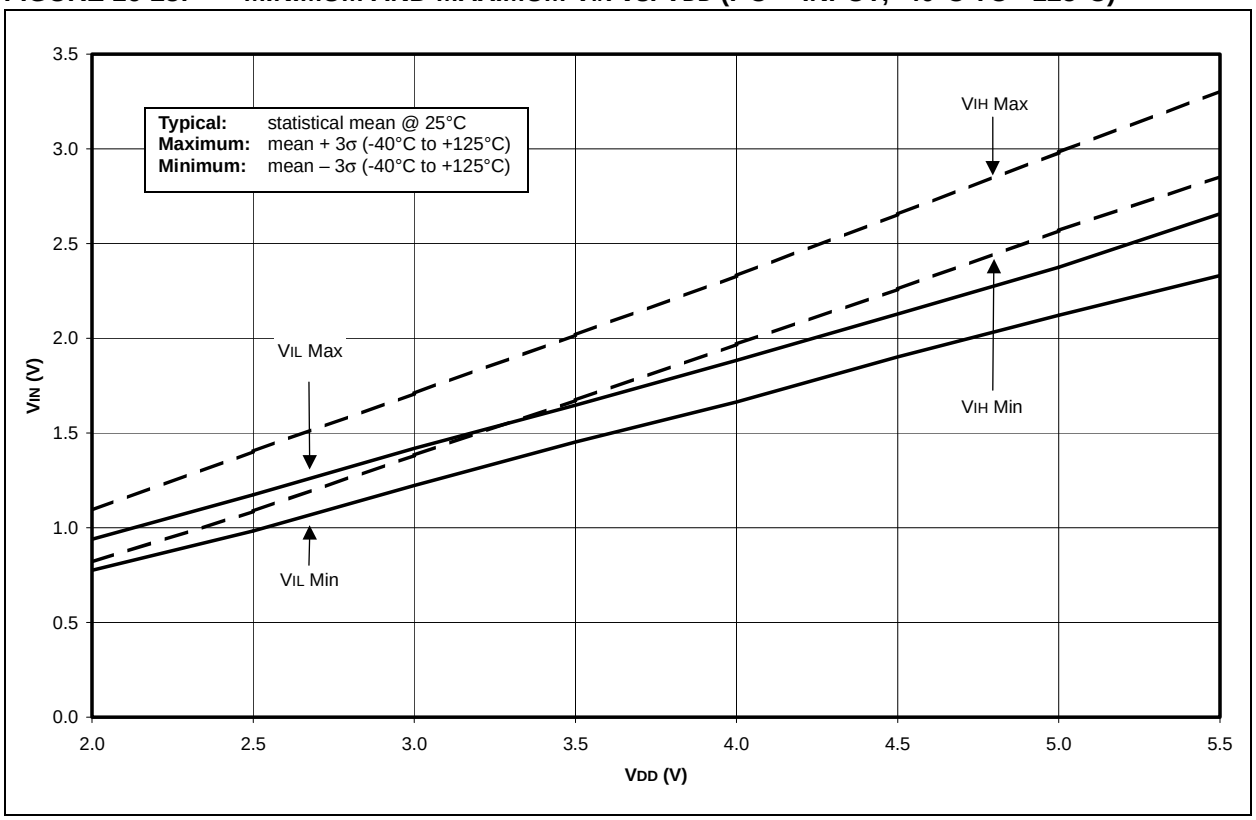
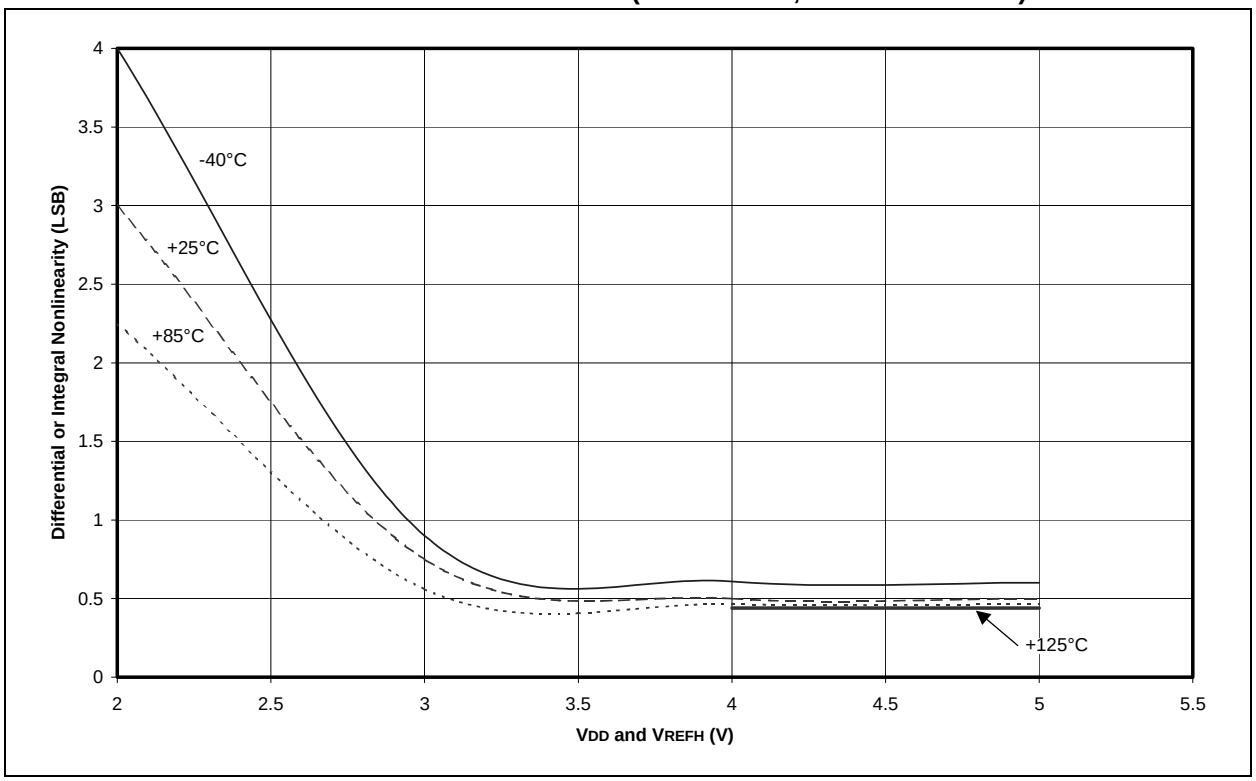



**FIGURE 16-24: A/D NONLINEARITY vs. VREFH (VDD = VREFH, -40°C TO +125°C)**

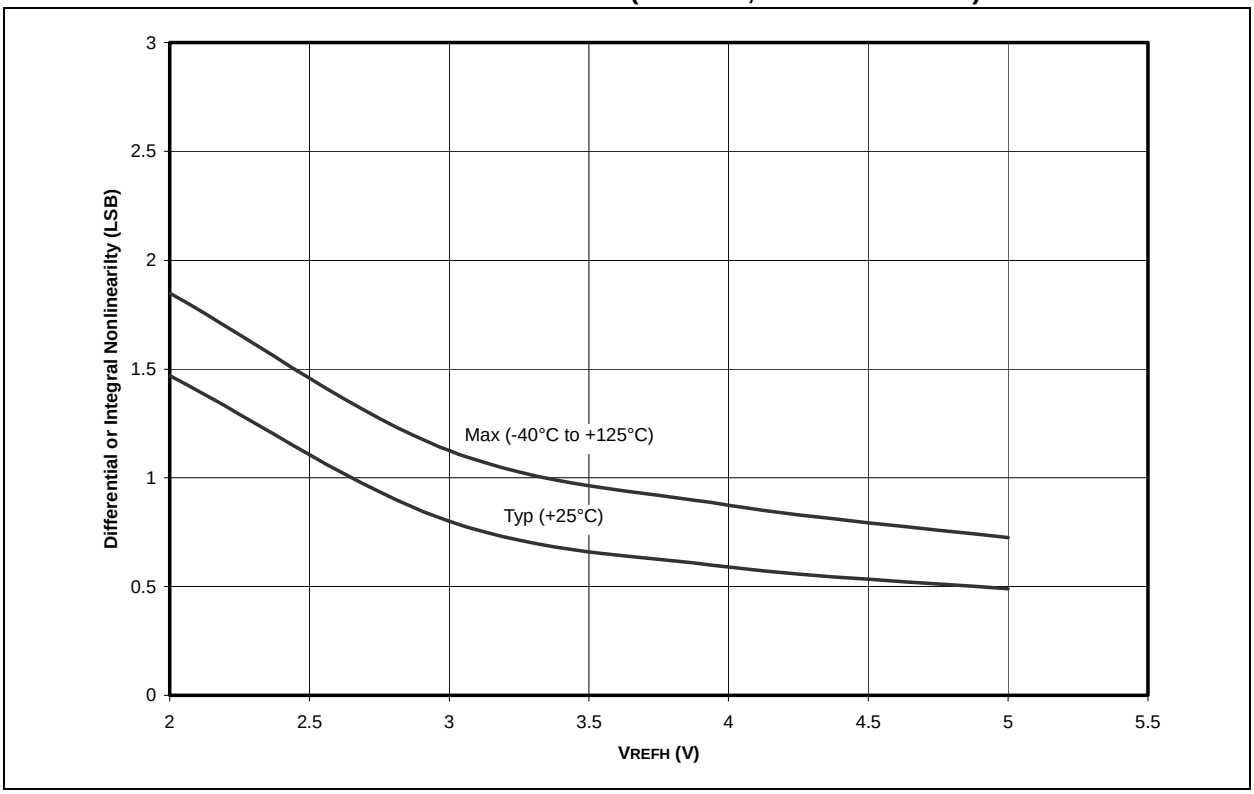




© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

**FIGURE 16-25: A/D NONLINEARITY vs. VREFH (VDD = 5V, -40°C TO +125°C)**






Downloaded from Arrow.com.

**NOTES:**

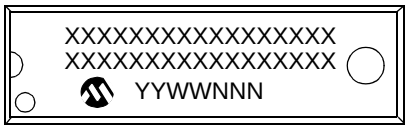

© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

## 17.0 PACKAGING INFORMATION

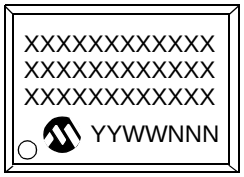
### 17.1 Package Marking Information

18-Lead PDIP

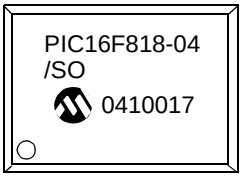



Example

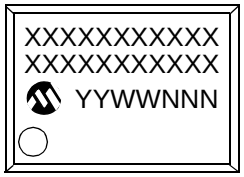
18-Lead SOIC



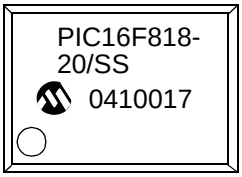

Example




20-Lead SSOP




Example




28-Lead QFN

Example

| | | |
|---|---|---|
| **Legend:** | XX...X | Customer specific information* |
| | Y | Year code (last digit of calendar year) |
| | YY | Year code (last 2 digits of calendar year) |
| | WW | Week code (week of January 1 is week ‘01’) |
| | NNN | Alphanumeric traceability code |

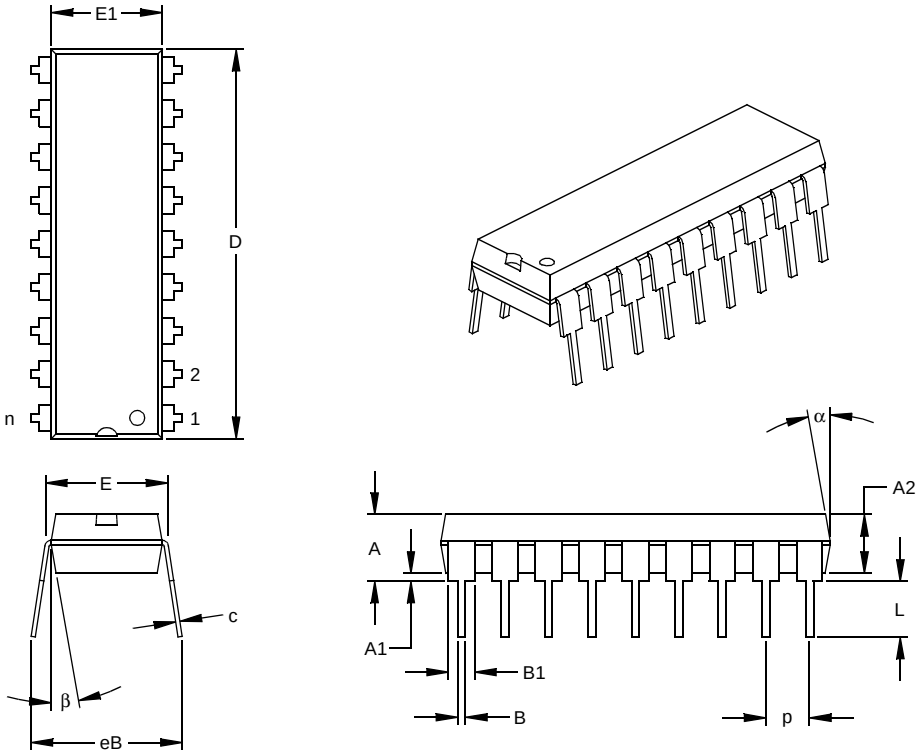
**Note**: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line thus limiting the number of available characters for customer specific information.

***** 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.



Downloaded from Arrow.com.

## 18-Lead Plastic Dual In-line (P) – 300 mil Body (PDIP)




| Units | | INCHES* | | | MILLIMETERS | | |
|---|---|---|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX | MIN | NOM | MAX |
| Number of Pins | n | | 18 | | | 18 | |
| Pitch | p | | .100 | | | 2.54 | |
| Top to Seating Plane | A | .140 | .155 | .170 | 3.56 | 3.94 | 4.32 |
| Molded Package Thickness | A2 | .115 | .130 | .145 | 2.92 | 3.30 | 3.68 |
| Base to Seating Plane | A1 | .015 | | | 0.38 | | |
| Shoulder to Shoulder Width | E | .300 | .313 | .325 | 7.62 | 7.94 | 8.26 |
| Molded Package Width | E1 | .240 | .250 | .260 | 6.10 | 6.35 | 6.60 |
| Overall Length | D | .890 | .898 | .905 | 22.61 | 22.80 | 22.99 |
| Tip to Seating Plane | L | .125 | .130 | .135 | 3.18 | 3.30 | 3.43 |
| Lead Thickness | c | .008 | .012 | .015 | 0.20 | 0.29 | 0.38 |
| Upper Lead Width | B1 | .045 | .058 | .070 | 1.14 | 1.46 | 1.78 |
| Lower Lead Width | B | .014 | .018 | .022 | 0.36 | 0.46 | 0.56 |
| Overall Row Spacing § | eB | .310 | .370 | .430 | 7.87 | 9.40 | 10.92 |
| Mold Draft Angle Top | α | 5 | 10 | 15 | 5 | 10 | 15 |
| Mold Draft Angle Bottom | β | 5 | 10 | 15 | 5 | 10 | 15 |

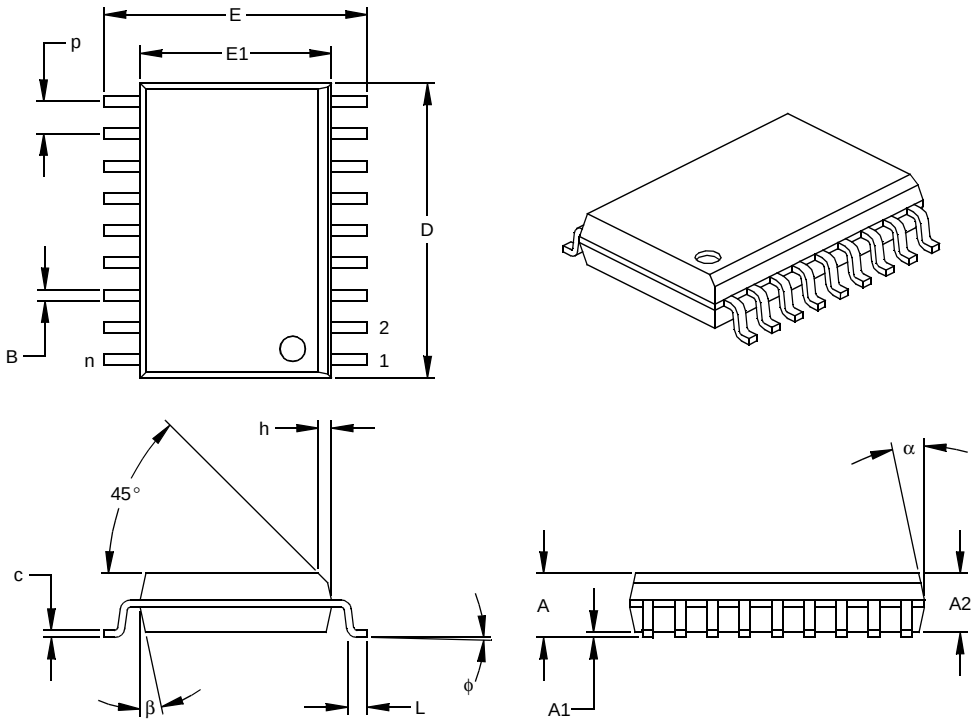
* Controlling Parameter
§ Significant Characteristic

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


© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

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




| Units | | INCHES* | | | MILLIMETERS | | |
|---|---|---|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX | MIN | NOM | MAX |
| Number of Pins | n | | 18 | | | 18 | |
| Pitch | p | | .050 | | | 1.27 | |
| Overall Height | A | .093 | .099 | .104 | 2.36 | 2.50 | 2.64 |
| Molded Package Thickness | A2 | .088 | .091 | .094 | 2.24 | 2.31 | 2.39 |
| Standoff § | A1 | .004 | .008 | .012 | 0.10 | 0.20 | 0.30 |
| Overall Width | E | .394 | .407 | .420 | 10.01 | 10.34 | 10.67 |
| Molded Package Width | E1 | .291 | .295 | .299 | 7.39 | 7.49 | 7.59 |
| Overall Length | D | .446 | .454 | .462 | 11.33 | 11.53 | 11.73 |
| Chamfer Distance | h | .010 | .020 | .029 | 0.25 | 0.50 | 0.74 |
| Foot Length | L | .016 | .033 | .050 | 0.41 | 0.84 | 1.27 |
| Foot Angle | ϕ | 0 | 4 | 8 | 0 | 4 | 8 |
| Lead Thickness | c | .009 | .011 | .012 | 0.23 | 0.27 | 0.30 |
| Lead Width | B | .014 | .017 | .020 | 0.36 | 0.42 | 0.51 |
| Mold Draft Angle Top | α | 0 | 12 | 15 | 0 | 12 | 15 |
| Mold Draft Angle Bottom | β | 0 | 12 | 15 | 0 | 12 | 15 |

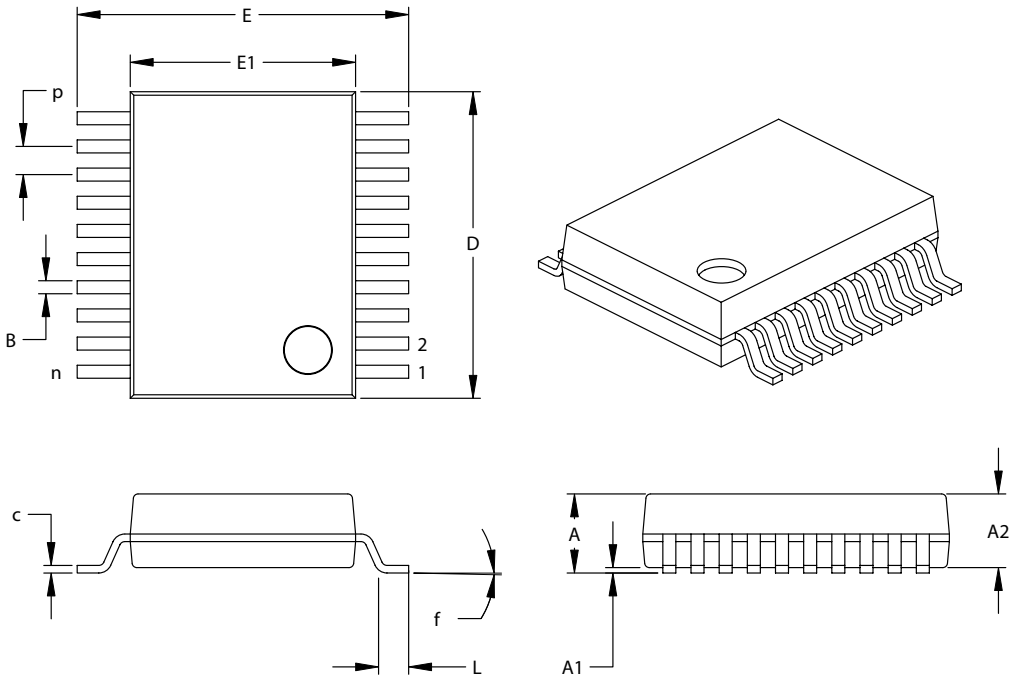
* Controlling Parameter
§ Significant Characteristic

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-013
Drawing No. C04-051



Downloaded from Arrow.com.

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




| Units | | INCHES | | | MILLIMETERS* | | |
|---|---|---|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX | MIN | NOM | MAX |
| Number of Pins | n | | 20 | | | 20 | |
| Pitch | p | | .026 | | | 0.65 | |
| Overall Height | A | - | - | .079 | - | - | 2.00 |
| Molded Package Thickness | A2 | .065 | .069 | .073 | 1.65 | 1.75 | 1.85 |
| Standoff | A1 | .002 | - | - | 0.05 | - | - |
| Overall Width | E | .291 | .307 | .323 | 7.40 | 7.80 | 8.20 |
| Molded Package Width | E1 | .197 | .209 | .220 | 5.00 | 5.30 | 5.60 |
| Overall Length | D | .272 | .283 | .289 | .295 | 7.20 | 7.50 |
| Foot Length | L | .022 | .030 | .037 | 0.55 | 0.75 | 0.95 |
| Lead Thickness | c | .004 | - | .010 | 0.09 | - | 0.25 |
| Foot Angle | f | 0° | 4° | 8° | 0° | 4° | 8° |
| Lead Width | B | .009 | - | .015 | 0.22 | - | 0.38 |

*Controlling Parameter

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

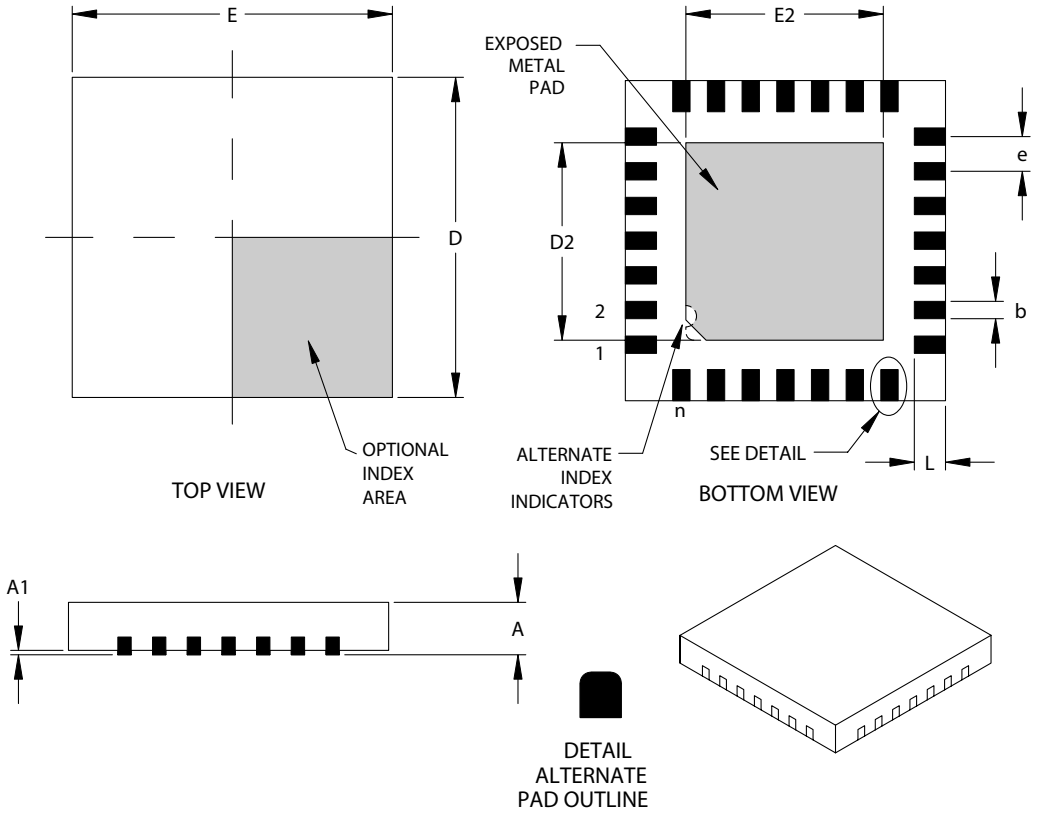
Revised 11/03/03



© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

## 28-Lead Plastic Quad Flat No Lead Package (ML) 6x6 mm Body (QFN) – With 0.55 mm Contact Length (Saw Singulated)




| Units | | INCHES | | | MILLIMETERS* | | |
|---|---|---|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX | MIN | NOM | MAX |
| Number of Pins | n | | 28 | | | 28 | |
| Pitch | e | | .026 BSC | | | 0.65 BSC | |
| Overall Height | A | .031 | .035 | .039 | 0.80 | 0.90 | 1.00 |
| Standoff | A1 | .000 | .001 | .002 | 0.00 | 0.02 | 0.05 |
| Contact Thickness | A3 | | .008 REF | | | 0.20 REF | |
| Overall Width | E | .232 | .236 | .240 | 5.90 | 6.00 | 6.10 |
| Exposed Pad Width | E2 | .140 | .146 | .152 | 3.55 | 3.70 | 3.85 |
| Overall Length | D | .232 | .236 | .240 | 5.90 | 6.00 | 6.10 |
| Exposed Pad Length | D2 | .140 | .146 | .152 | 3.55 | 3.70 | 3.85 |
| Contact Width | b | .009 | .011 | .013 | 0.23 | 0.28 | 0.33 |
| Contact Length | L | .020 | .024 | .028 | 0.50 | 0.60 | 0.70 |

*Controlling Parameter

Notes:

JEDEC equivalent: MO-220

Drawing No. C04-105

Revised 05-24-04



Downloaded from Arrow.com.

**NOTES:**


© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

## APPENDIX A: REVISION HISTORY

### Revision A (May 2002)

Original version of this data sheet.

### Revision B (August 2002)

Added INTRC section. PWRT and BOR are independent of each other. Revised program memory text and code routine. Added QFN package. Modified PORTB diagrams.

### Revision C (November 2002)

Added various new feature descriptions. Added internal RC oscillator specifications. Added low-power Timer1 specifications and RTC application example.

### Revision D (November 2003)

Updated IRCF bit modification information and changed the INTOSC stabilization delay from 1 ms to 4 ms in **Section 4.0 "Oscillator Configurations"**. Updated **Section 12.17 "In-Circuit Serial Programming"** to clarify LVP programming. In **Section 15.0 "Electrical Characteristics"**, the DC Characteristics (**Section 15.2** and **Section 15.3**) have been updated to include the Typ, Min and Max values and Table 15-1 **"External Clock Timing Requirements"** has been updated.

### Revision E (September 2004)

This revision includes the DC and AC Characteristics Graphs and Tables. The Electrical Specifications in **Section 16.0 "DC and AC Characteristics Graphs and Tables"** have been updated and there have been minor corrections to the data sheet text.

## APPENDIX B: DEVICE DIFFERENCES

The differences between the devices in this data sheet are listed in Table B-1.

**TABLE B-1: DIFFERENCES BETWEEN THE PIC16F818 AND PIC16F819**

| Features | PIC16F818 | PIC16F819 |
|---|---|---|
| Flash Program Memory (14-bit words) | 1K | 2K |
| Data Memory (bytes) | 128 | 256 |
| EEPROM Data Memory (bytes) | 128 | 256 |



Downloaded from Arrow.com.

**NOTES:**


© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

# INDEX


## A

A/D
- Acquisition Requirements ... 84
- ADIF Bit ... 83
- Analog-to-Digital Converter ... 81
- Associated Registers ... 87
- Calculating Acquisition Time ... 84
- Configuring Analog Port Pins ... 85
- Configuring the Interrupt ... 83
- Configuring the Module ... 83
- Conversion Clock ... 85
- Conversion Requirements ... 142
- Conversions ... 86
- Converter Characteristics ... 141
- Delays ... 84
- Effects of a Reset ... 87
- GO/DONE Bit ... 83
- Internal Sampling Switch (Rss) Impedance ... 84
- Operation During Sleep ... 87
- Result Registers ... 86
- Source Impedance ... 84
- Time Delays ... 84
- Use of the CCP Trigger ... 87

Absolute Maximum Ratings ... 117
ACK ... 77
ADCON0 Register ... 81
ADCON1 Register ... 81
ADRESH Register ... 13, 81
ADRESH, ADRESL Register Pair ... 83
ADRESL Register ... 14, 81
Application Notes
- AN556 (Implementing a Table Read) ... 23
- AN578 (Use of the SSP Module in the $I^2C$ Multi-Master Environment) ... 71
- AN607 (Power-up Trouble Shooting) ... 92

Assembler
- MPASM Assembler ... 111

## B

BF Bit ... 77
Block Diagrams
- A/D ... 83
- Analog Input Model ... 84
- Capture Mode Operation ... 66
- Compare Mode Operation ... 67
- In-Circuit Serial Programming Connections ... 101
- Interrupt Logic ... 96
- On-Chip Reset Circuit ... 91
- PIC16F818/819 ... 6
- PWM ... 68
- RA0/AN0:RA1/AN1 Pins ... 40
- RA2/AN2/VREF- Pin ... 40
- RA3/AN3/VREF+ Pin ... 40
- RA4/AN4/T0CKI Pin ... 40
- RA5/MCLR/VPP Pin ... 41
- RA6/OSC2/CLKO Pin ... 41
- RA7/OSC1/CLKI Pin ... 42
- RB0 Pin ... 45
- RB1 Pin ... 46
- RB2 Pin ... 47
- RB3 Pin ... 48
- RB4 Pin ... 49
- RB5 Pin ... 50
- RB6 Pin ... 51
- RB7 Pin ... 52
- Recommended MCLR Circuit ... 92
- SSP in $I^2C$ Mode ... 76
- SSP in SPI Mode ... 74
- System Clock ... 38
- Timer0/WDT Prescaler ... 53
- Timer1 ... 58
- Timer2 ... 63
- Watchdog Timer (WDT) ... 98

BOR. *See* Brown-out Reset.
Brown-out Reset (BOR) ... 89, 91, 92, 93, 94

## C

C Compilers
- MPLAB C17 ... 112
- MPLAB C18 ... 112
- MPLAB C30 ... 112

Capture/Compare/PWM (CCP) ... 65
- Capture Mode ... 66
  - CCP Prescaler ... 66
  - Pin Configuration ... 66
  - Software Interrupt ... 66
  - Timer1 Mode Selection ... 66
- Capture, Compare and Timer1
  - Associated Registers ... 67
- CCP1IF ... 66
- CCPR1 ... 66
- CCPR1H:CCPR1L ... 66
- Compare Mode ... 67
  - Pin Configuration ... 67
  - Software Interrupt Mode ... 67
  - Special Event Trigger ... 67
  - Special Event Trigger Output of CCP1 ... 67
  - Timer1 Mode Selection ... 67
- PWM and Timer2
  - Associated Registers ... 69
- PWM Mode ... 68
  - Duty Cycle ... 68
  - Example Frequencies/Resolutions ... 69
  - Period ... 68
  - Setup for Operation ... 69
- Timer Resources ... 65

CCP1M0 Bit ... 65
CCP1M1 Bit ... 65
CCP1M2 Bit ... 65
CCP1M3 Bit ... 65
CCP1X Bit ... 65
CCP1Y Bit ... 65
CCPR1H Register ... 65
CCPR1L Register ... 65
Code Examples
- Changing Between Capture Prescalers ... 66
- Changing Prescaler Assignment from Timer0 to WDT ... 55
- Changing Prescaler Assignment from WDT to Timer0 ... 55
- Clearing RAM Using Indirect Addressing ... 23
- Erasing a Flash Program Memory Row ... 29
- Implementing a Real-Time Clock Using a Timer1 Interrupt Service ... 62
- Initializing PORTA ... 39
- Reading a 16-Bit Free Running Timer ... 59




Downloaded from Arrow.com.

Reading Data EEPROM ..... 27
Reading Flash Program Memory ..... 28
Saving Status and W Registers in RAM ..... 97
Writing a 16-Bit Free Running Timer ..... 59
Writing to Data EEPROM ..... 27
Writing to Flash Program Memory ..... 31
Code Protection ..... 89, 100
Computed GOTO ..... 23
Configuration Bits ..... 89
Crystal Oscillator and Ceramic Resonators ..... 33

**D**

Data EEPROM Memory ..... 25
Associated Registers ..... 32
EEADR Register ..... 25
EEADRH Register ..... 25
EECON1 Register ..... 25
EECON2 Register ..... 25
EEDATA Register ..... 25
EEDATH Register ..... 25
Operation During Code-Protect ..... 32
Protection Against Spurious Writes ..... 32
Reading ..... 27
Write Interrupt Enable Flag (EEIF Bit) ..... 25
Writing ..... 27
Data Memory
Special Function Registers ..... 13
DC and AC Characteristics
Graphs and Tables ..... 143
DC Characteristics
Internal RC Accuracy ..... 127
PIC16F818/819, PIC16LF818/819 ..... 128
Power-Down and Supply Current ..... 120
Supply Voltage ..... 119
Demonstration Boards
PICDEM 1 ..... 114
PICDEM 17 ..... 115
PICDEM 18R ..... 115
PICDEM 2 Plus ..... 114
PICDEM 3 ..... 114
PICDEM 4 ..... 114
PICDEM LIN ..... 115
PICDEM USB ..... 115
PICDEM.net Internet/Ethernet ..... 114
Development Support ..... 111
Device Differences ..... 163
Device Overview ..... 5
Direct Addressing ..... 24

**E**

EEADR Register ..... 25
EEADRH Register ..... 25
EECON1 Register ..... 25
EECON2 Register ..... 25
EEDATA Register ..... 25
EEDATH Register ..... 25
Electrical Characteristics ..... 117
Endurance ..... 1
Errata ..... 3
Evaluation and Programming Tools ..... 115
External Clock Input ..... 34
External Interrupt Input (RB0/INT).
*See* Interrupt Sources.

**F**

Flash Program Memory ..... 25
Associated Registers ..... 32
EEADR Register ..... 25
EEADRH Register ..... 25
EECON1 Register ..... 25
EECON2 Register ..... 25
EEDATA Register ..... 25
EEDATH Register ..... 25
Erasing ..... 28
Reading ..... 28
Writing ..... 30
FSR Register ..... 13, 14, 15, 23

**G**

General Purpose Register File ..... 10

**I**

I/O Ports ..... 39
$I^2C$
Associated Registers ..... 79
Master Mode Operation ..... 79
Mode ..... 76
Mode Selection ..... 76
Multi-Master Mode Operation ..... 79
Slave Mode ..... 77
Addressing ..... 77
Reception ..... 77
SCL, SDA Pins ..... 77
Transmission ..... 77
ID Locations ..... 89, 100
In-Circuit Debugger ..... 100
In-Circuit Serial Programming ..... 89
In-Circuit Serial Programming (ICSP) ..... 101
INDF Register ..... 14, 15, 23
Indirect Addressing ..... 23, 24
Instruction Format ..... 103
Instruction Set ..... 103
ADDLW ..... 105
ADDWF ..... 105
ANDLW ..... 105
ANDWF ..... 105
BCF ..... 105
BSF ..... 105
BTFSC ..... 106
BTFSS ..... 106
CALL ..... 106
CLRF ..... 106
CLRW ..... 106
CLRWDT ..... 106
COMF ..... 107
DECF ..... 107
DECFSZ ..... 107
Descriptions ..... 105
GOTO ..... 107
INCF ..... 107
INCFSZ ..... 107
IORLW ..... 108
IORWF ..... 108
MOVF ..... 108
MOVLW ..... 108
MOVWF ..... 108



© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

NOP ..... 108
Read-Modify-Write Operations ..... 103
RETFIE ..... 109
RETLW ..... 109
RETURN ..... 109
RLF ..... 109
RRF ..... 109
SLEEP ..... 109
SUBLW ..... 110
SUBWF ..... 110
Summary Table ..... 104
SWAPF ..... 110
XORLW ..... 110
XORWF ..... 110
INT Interrupt (RB0/INT). *See* Interrupt Sources.
INTCON Register ..... 15
GIE Bit ..... 18
INTE Bit ..... 18
INTF Bit ..... 18
RBIF Bit ..... 18
TMR0IE Bit ..... 18
Internal Oscillator Block ..... 35
INTRC Modes ..... 35
Interrupt Sources ..... 89, 96
RB0/INT Pin, External ..... 97
TMR0 Overflow ..... 97
Interrupts
RB7:RB4 Port Change ..... 43
Synchronous Serial Port Interrupt ..... 20
Interrupts, Context Saving During ..... 97
Interrupts, Enable Bits
Global Interrupt Enable (GIE Bit) ..... 96
Interrupt-on-Change (RB7:RB4)
Enable (RBIE Bit) ..... 97
RB0/INT Enable (INTE Bit) ..... 18
TMR0 Overflow Enable (TMR0IE Bit) ..... 18
Interrupts, Enable bits
Global Interrupt Enable (GIE Bit) ..... 18
Interrupts, Flag Bits
Interrupt-on-Change (RB7:RB4) Flag
(RBIF Bit) ..... 18, 97
RB0/INT Flag (INTF Bit) ..... 18
TMR0 Overflow Flag (TMR0IF Bit) ..... 97
INTRC Modes
Adjustment ..... 36


## L


Loading of PC ..... 23
Low-Voltage ICSP Programming ..... 102


## M


Master Clear ($\overline{MCLR}$)
$\overline{MCLR}$ Reset, Normal Operation ..... 91, 93, 94
$\overline{MCLR}$ Reset, Sleep ..... 91, 93, 94
Operation and ESD Protection ..... 92
Memory Organization ..... 9
Data Memory ..... 10
Program Memory ..... 9
MPLAB ASM30 Assembler, Linker, Librarian ..... 112
MPLAB ICD 2 In-Circuit Debugger ..... 113
MPLAB ICE 2000 High-Performance
Universal In-Circuit Emulator ..... 113
MPLAB ICE 4000 High-Performance
Universal In-Circuit Emulator ..... 113
MPLAB Integrated Development
Environment Software ..... 111
MPLAB PM3 Device Programmer ..... 113
MPLINK Object Linker/
MPLIB Object Librarian ..... 112


## O


Opcode Field Descriptions ..... 103
OPTION_REG Register ..... 15
INTEDG Bit ..... 17, 54
PS2:PS0 Bits ..... 17
$\overline{PSA}$ Bit ..... 17
$\overline{RBPU}$ Bit ..... 17, 54
T0CS Bit ..... 17
T0SE Bit ..... 17
Oscillator Configuration ..... 33
ECIO ..... 33
EXTCLK ..... 93
EXTRC ..... 93
HS ..... 33, 93
INTIO1 ..... 33
INTIO2 ..... 33
INTRC ..... 93
LP ..... 33, 93
RC ..... 33, 35
RCIO ..... 33
XT ..... 33, 93
Oscillator Control Register ..... 37
Modifying IRCF Bits ..... 37
Clock Transition Sequence ..... 37
Oscillator Start-up Timer (OST) ..... 89, 92
Oscillator, WDT ..... 98


## P


Packaging Information ..... 157
Marking ..... 157
PCFG0 Bit ..... 82
PCFG1 Bit ..... 82
PCFG2 Bit ..... 82
PCFG3 Bit ..... 82
PCL Register ..... 13, 14, 15, 23
PCLATH Register ..... 13, 14, 15, 23
PCON Register ..... 93
$\overline{POR}$ Bit ..... 22
PICkit 1 Flash Starter Kit ..... 115
PICSTART Plus Development
Programmer ..... 114
Pinout Descriptions
PIC16F818/819 ..... 7
Pointer, FSR ..... 23
POP ..... 23
POR. *See* Power-on Reset.
PORTA ..... 7
Associated Register Summary ..... 39
Functions ..... 39
PORTA Register ..... 39
TRISA Register ..... 39
PORTA Register ..... 13




Downloaded from Arrow.com.

PORTB ..... 8
Associated Register Summary ..... 44
Functions ..... 44
PORTB Register ..... 43
Pull-up Enable (RBPU Bit) ..... 17, 54
RB0/INT Edge Select (INTEDG Bit) ..... 17, 54
RB0/INT Pin, External ..... 97
RB7:RB4 Interrupt-on-Change ..... 97
RB7:RB4 Interrupt-on-Change Enable (RBIE Bit) ..... 97
RB7:RB4 Interrupt-on-Change Flag (RBIF Bit) ..... 18, 97
TRISB Register ..... 43
PORTB Register ..... 13, 15
Postscaler, WDT
Assignment (PSA Bit) ..... 17
Rate Select (PS2:PS0 Bits) ..... 17
Power-Down Mode. *See* Sleep.
Power-on Reset (POR) ..... 89, 91, 92, 93, 94
POR Status (POR Bit) ..... 22
Power Control (PCON) Register ..... 93
Power-Down (PD Bit) ..... 91
Time-out (TO Bit) ..... 16, 91
Power-up Timer (PWRT) ..... 89, 92
PR2 Register ..... 63
Prescaler, Timer0
Assignment (PSA Bit) ..... 17
Rate Select (PS2:PS0 Bits) ..... 17
PRO MATE II Universal Device Programmer ..... 113
Program Counter
Reset Conditions ..... 93
Program Memory
Interrupt Vector ..... 9
Map and Stack
PIC16F818 ..... 9
PIC16F819 ..... 9
Reset Vector ..... 9
Program Verification ..... 100
PUSH ..... 23

## R

R/W Bit ..... 77
RA0/AN0 Pin ..... 7
RA1/AN1 Pin ..... 7
RA2/AN2/VREF- Pin ..... 7
RA3/AN3/VREF+ Pin ..... 7
RA4/AN4/T0CKI Pin ..... 7
RA5/MCLR/VPP Pin ..... 7
RA6/OSC2/CLKO Pin ..... 7
RA7/OSC1/CLKI Pin ..... 7
RB0/INT Pin ..... 8
RB1/SDI/SDA Pin ..... 8
RB2/SDO/CCP1 Pin ..... 8
RB3/CCP1/PGM Pin ..... 8
RB4/SCK/SCL Pin ..... 8
RB5/SS Pin ..... 8
RB6/T1OSO/T1CKI/PGC Pin ..... 8
RB7/T1OSI/PGD Pin ..... 8
RBIF Bit ..... 43
RCIO Oscillator Mode ..... 35
Receive Overflow Indicator Bit, SSPOV ..... 73
Register File Map
PIC16F818 ..... 11
PIC16F819 ..... 12
Registers
ADCON0 (A/D Control 0) ..... 81
ADCON1 (A/D Control 1) ..... 82
CCP1CON (Capture/Compare/PWM Control 1) ..... 65
Configuration Word ..... 90
EECON1 (Data EEPROM Access Control 1) ..... 26
Initialization Conditions (table) ..... 94
INTCON (Interrupt Control) ..... 18
OPTION_REG (Option) ..... 17, 54
OSCCON (Oscillator Control) ..... 38
OSCTUNE (Oscillator Tuning) ..... 36
PCON (Power Control) ..... 22
PIE1 (Peripheral Interrupt Enable 1) ..... 19
PIE2 (Peripheral Interrupt Enable 2) ..... 21
PIR1 (Peripheral Interrupt Request (Flag) 1) ..... 20
PIR2 (Peripheral Interrupt Request (Flag) 2) ..... 21
SSPCON (Synchronous Serial Port Control 1) ..... 73
SSPSTAT (Synchronous Serial Port Status) ..... 72
Status ..... 16
T1CON (Timer1 Control) ..... 57
T2CON (Timer2 Control) ..... 64
Reset ..... 89, 91
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 All Registers ..... 94
Reset Conditions for PCON Register ..... 93
Reset Conditions for Program Counter ..... 93
Reset Conditions for Status Register ..... 93
WDT Reset. *See* Watchdog Timer (WDT).
Revision History ..... 163
RP0 Bit ..... 10
RP1 Bit ..... 10

## S

Sales and Support ..... 172
SCL Clock ..... 77
Sleep ..... 89, 91, 99
Software Simulator (MPLAB SIM) ..... 112
Software Simulator (MPLAB SIM30) ..... 112
Special Event Trigger ..... 87
Special Features of the CPU ..... 89
Special Function Register Summary ..... 13
Special Function Registers ..... 13
SPI Mode
Associated Registers ..... 74
Serial Clock ..... 71
Serial Data In ..... 71
Serial Data Out ..... 71
Slave Select ..... 71
SSP
ACK ..... 77
I2C
I2C Operation ..... 76
SSPADD Register ..... 14
SSPIF ..... 20
SSPOV ..... 73
SSPOV Bit ..... 77


© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

SSPSTAT Register ..... 14
Stack ..... 23
Overflow ..... 23
Underflow ..... 23
Status Register ..... 13, 15
DC Bit ..... 16
IRP Bit ..... 16
PD Bit ..... 91
TO Bit ..... 16, 91
Z Bit ..... 16
Synchronous Serial Port (SSP) ..... 71
Overview ..... 71
SPI Mode ..... 71
Synchronous Serial Port Interrupt ..... 20

## T

T1CKPS0 Bit ..... 57
T1CKPS1 Bit ..... 57
T1OSCEN Bit ..... 57
T1SYNC Bit ..... 57
T2CKPS0 Bit ..... 64
T2CKPS1 Bit ..... 64
Tad ..... 85
Time-out Sequence ..... 92
Timer0 ..... 53
Associated Registers ..... 55
Clock Source Edge Select (T0SE Bit) ..... 17
Clock Source Select (T0CS Bit) ..... 17
External Clock ..... 54
Interrupt ..... 53
Operation ..... 53
Overflow Enable (TMR0IE Bit) ..... 18
Overflow Flag (TMR0IF Bit) ..... 97
Overflow Interrupt ..... 97
Prescaler ..... 54
T0CKI ..... 54
Timer1 ..... 57
Associated Registers ..... 62
Capacitor Selection ..... 60
Counter Operation ..... 58
Operation ..... 57
Operation in Asynchronous Counter Mode ..... 59
Operation in Synchronized Counter Mode ..... 58
Operation in Timer Mode ..... 58
Oscillator ..... 60
Oscillator Layout Considerations ..... 60
Prescaler ..... 61
Resetting Register Pair (TMR1H, TMR1L) ..... 61
Resetting Using a CCP Trigger Output ..... 61
TMR1H ..... 59
TMR1L ..... 59
Use as a Real-Time Clock ..... 61
Timer2 ..... 63
Associated Registers ..... 64
Output ..... 63
Postscaler ..... 63
Prescaler ..... 63
Prescaler and Postscaler ..... 63
Timing Diagrams
A/D Conversion ..... 142
Brown-out Reset ..... 133
Capture/Compare/PWM (CCP1) ..... 135
CLKO and I/O ..... 132
External Clock ..... 131
I²C Bus Data ..... 139
I²C Bus Start/Stop Bits ..... 138
I²C Reception (7-Bit Address) ..... 78
I²C Transmission (7-Bit Address) ..... 78
PWM Output ..... 68
Reset, Watchdog Timer, Oscillator Start-up Timer and Power-up Timer ..... 133
Slow Rise Time (MCLR Tied to VDD Through RC Network) ..... 96
SPI Master Mode ..... 75
SPI Master Mode (CKE = 0, SMP = 0) ..... 136
SPI Master Mode (CKE = 1, SMP = 1) ..... 136
SPI Slave Mode (CKE = 0) ..... 75, 137
SPI Slave Mode (CKE = 1) ..... 75, 137
Time-out Sequence on Power-up (MCLR Tied to VDD Through Pull-up Resistor) ..... 95
Time-out Sequence on Power-up (MCLR Tied to VDD Through RC Network): Case 1 ..... 95
Time-out Sequence on Power-up (MCLR Tied to VDD Through RC Network): Case 2 ..... 95
Timer0 and Timer1 External Clock ..... 134
Timer1 Incrementing Edge ..... 58
Wake-up from Sleep through Interrupt ..... 100
Timing Parameter Symbology ..... 130
Timing Requirements
External Clock ..... 131
TMR0 Register ..... 15
TMR1CS Bit ..... 57
TMR1H Register ..... 13
TMR1L Register ..... 13
TMR1ON Bit ..... 57
TMR2 Register ..... 13
TMR2ON Bit ..... 64
TOUTPS0 Bit ..... 64
TOUTPS1 Bit ..... 64
TOUTPS2 Bit ..... 64
TOUTPS3 Bit ..... 64
TRISA Register ..... 14
TRISB Register ..... 14, 15

## V

Vdd Pin ..... 8
Vss Pin ..... 8

## W

Wake-up from Sleep ..... 89, 99
Interrupts ..... 93, 94
MCLR Reset ..... 94
WDT Reset ..... 94
Wake-up Using Interrupts ..... 99
Watchdog Timer (WDT) ..... 89, 98
Associated Registers ..... 98
Enable (WDTEN Bit) ..... 98
INTRC Oscillator ..... 98
Postscaler. *See* Postscaler, WDT.
Programming Considerations ..... 98
Time-out Period ..... 98
WDT Reset, Normal Operation ..... 91, 93, 94
WDT Reset, Sleep ..... 91, 94
WDT Wake-up ..... 93
WCOL ..... 73
Write Collision Detect Bit, WCOL ..... 73
WWW, On-Line Support ..... 3



Downloaded from Arrow.com.

**NOTES:**


© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

## 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



Downloaded from Arrow.com.

## READER RESPONSE

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

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

To: Technical Publications Manager  Total Pages Sent _____

RE: Reader Response

From: Name _____

Company _____

Address _____

City / State / ZIP / Country _____

Telephone: (_____) _____ - _____  FAX: (_____) _____ - _____

Application (optional):

Would you like a reply? ___Y ___N

Device: PIC16F818/819  Literature Number: DS39598E

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?


© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.

## PIC16F818/819 PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

| PART NO. | X | /XX | XXX |
|---|---|---|---|
| Device | Temperature Range | Package | Pattern |

| | |
|---|---|
| Device | PIC16F818: Standard VDD range<br>PIC16F818T: (Tape and Reel)<br>PIC16LF818: Extended VDD range |
| Temperature Range | - = 0°C to +70°C<br>I = -40°C to +85°C (Industrial)<br>E = -40°C to +125°C (Extended) |
| Package | P = PDIP<br>SO = SOIC<br>SS = SSOP<br>ML = QFN |
| Pattern | QTP, SQTP, ROM Code (factory specified) or Special Requirements. Blank for OTP and Windowed devices. |

**Examples:**

a) PIC16LF818-I/P = Industrial temp., PDIP package, Extended VDD limits.

b) PIC16F818-I/SO = Industrial temp., SOIC package, normal VDD limits.

**Note 1:** F = CMOS Flash
LF = Low-Power CMOS Flash

**2:** T = in tape and reel – SOIC, SSOP packages only.



Downloaded from Arrow.com.

# WORLDWIDE SALES AND SERVICE

## AMERICAS

**Corporate Office**
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http:\\support.microchip.com
Web Address:
www.microchip.com

**Atlanta**
Alpharetta, GA
Tel: 770-640-0034
Fax: 770-640-0307

**Boston**
Westford, MA
Tel: 978-692-3848
Fax: 978-692-3821

**Chicago**
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

**Dallas**
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

**Detroit**
Farmington Hills, MI
Tel: 248-538-2250
Fax: 248-538-2260

**Kokomo**
Kokomo, IN
Tel: 765-864-8360
Fax: 765-864-8387

**Los Angeles**
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

**San Jose**
Mountain View, CA
Tel: 650-215-1444
Fax: 650-961-0286

**Toronto**
Mississauga, Ontario,
Canada
Tel: 905-673-0699
Fax: 905-673-6509

## ASIA/PACIFIC

**Australia - Sydney**
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

**China - Beijing**
Tel: 86-10-8528-2100
Fax: 86-10-8528-2104

**China - Chengdu**
Tel: 86-28-8676-6200
Fax: 86-28-8676-6599

**China - Fuzhou**
Tel: 86-591-750-3506
Fax: 86-591-750-3521

**China - Hong Kong SAR**
Tel: 852-2401-1200
Fax: 852-2401-3431

**China - Shanghai**
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

**China - Shenyang**
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

**China - Shenzhen**
Tel: 86-755-8203-2660
Fax: 86-755-8203-1760

**China - Shunde**
Tel: 86-757-2839-5507
Fax: 86-757-2839-5571

**China - Qingdao**
Tel: 86-532-502-7355
Fax: 86-532-502-7205

## ASIA/PACIFIC

**India - Bangalore**
Tel: 91-80-2229-0061
Fax: 91-80-2229-0062

**India - New Delhi**
Tel: 91-11-5160-8632
Fax: 91-11-5160-8632

**Japan - Kanagawa**
Tel: 81-45-471- 6166
Fax: 81-45-471-6122

**Korea - Seoul**
Tel: 82-2-554-7200
Fax: 82-2-558-5932 or
82-2-558-5934

**Singapore**
Tel: 65-6334-8870
Fax: 65-6334-8850

**Taiwan - Kaohsiung**
Tel: 886-7-536-4818
Fax: 886-7-536-4803

**Taiwan - Taipei**
Tel: 886-2-2500-6610
Fax: 886-2-2508-0102

**Taiwan - Hsinchu**
Tel: 886-3-572-9526
Fax: 886-3-572-6459

## EUROPE

**Austria - Weis**
Tel: 43-7242-2244-399
Fax: 43-7242-2244-393

**Denmark - Ballerup**
Tel: 45-4420-9895
Fax: 45-4420-9910

**France - Massy**
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

**Germany - Ismaning**
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

**Italy - Milan**
Tel: 39-0331-742611
Fax: 39-0331-466781

**Netherlands - Drunen**
Tel: 31-416-690399
Fax: 31-416-690340

**England - Berkshire**
Tel: 44-118-921-5869
Fax: 44-118-921-5820

09/27/04


© 2004 Microchip Technology Inc.

Downloaded from Arrow.com.