



# **PIC16F87/88**

## **Data Sheet**

18/20/28-Pin Enhanced Flash  
Microcontrollers with  
nanoWatt Technology

---

---

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

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

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

---

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in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates.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, Migratable Memory, 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, 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.

© 2005, 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.*

## 18/20/28-Pin Enhanced Flash MCUs with nanoWatt Technology

### Low-Power Features:

- Power-Managed modes:
  - Primary Run: RC oscillator, 76  $\mu$ A, 1 MHz, 2V
  - RC\_RUN: 7  $\mu$ A, 31.25 kHz, 2V
  - SEC\_RUN: 9  $\mu$ A, 32 kHz, 2V
  - Sleep: 0.1  $\mu$ A, 2V
- Timer1 Oscillator: 1.8  $\mu$ A, 32 kHz, 2V
- Watchdog Timer: 2.2  $\mu$ A, 2V
- Two-Speed Oscillator Start-up

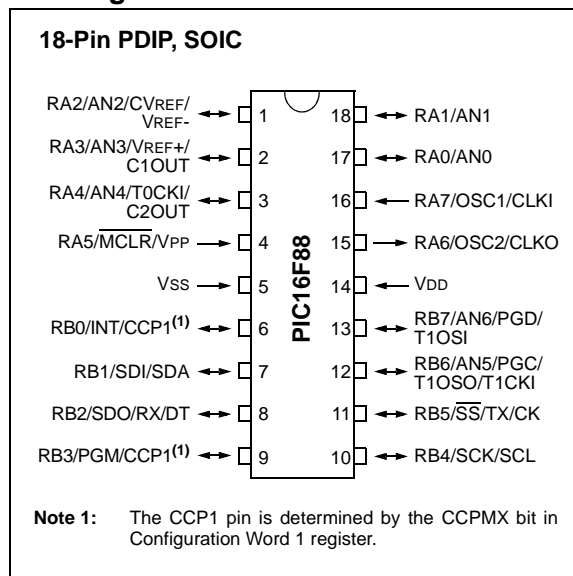
### Oscillators:

- 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:

- 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, 7-channel Analog-to-Digital Converter
- Synchronous Serial Port (SSP) with SPI™ (Master/Slave) and I<sup>2</sup>C™ (Slave)
- Addressable Universal Synchronous Asynchronous Receiver Transmitter (AUSART/SCI) with 9-bit address detection:
  - RS-232 operation using internal oscillator (no external crystal required)
- Dual Analog Comparator module:
  - Programmable on-chip voltage reference
  - Programmable input multiplexing from device inputs and internal voltage reference
  - Comparator outputs are externally accessible

### Pin Diagram



### Special Microcontroller Features:

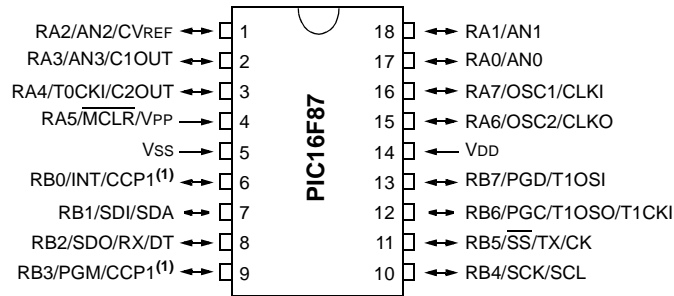
- 100,000 erase/write cycles Enhanced Flash program memory typical
- 1,000,000 typical erase/write cycles EEPROM data memory typical
- EEPROM Data Retention: > 40 years
- In-Circuit Serial Programming™ (ICSP™) via two pins
- Processor read/write access to program memory
- Low-Voltage Programming
- In-Circuit Debugging via two pins
- Extended Watchdog Timer (WDT):
  - Programmable period from 1 ms to 268s
- Wide operating voltage range: 2.0V to 5.5V

Device	Program Memory		Data Memory		I/O Pins	10-bit A/D (ch)	CCP (PWM)	AUSART	Comparators	SSP	Timers 8/16-bit
	Flash (bytes)	# Single-Word Instructions	SRAM (bytes)	EEPROM (bytes)							
PIC16F87	7168	4096	368	256	16	N/A	1	Y	2	Y	2/1
PIC16F88	7168	4096	368	256	16	1	1	Y	2	Y	2/1

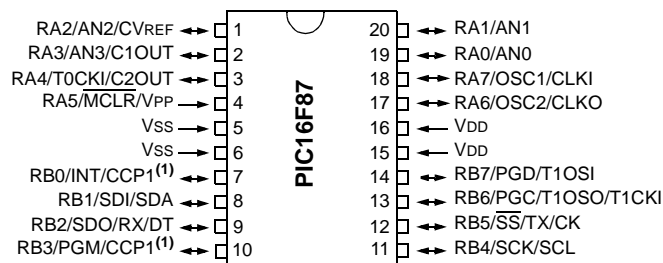
# PIC16F87/88

## Pin Diagrams

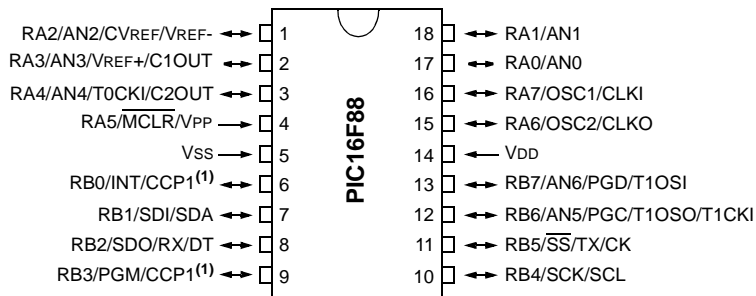
### 18-Pin PDIP, SOIC



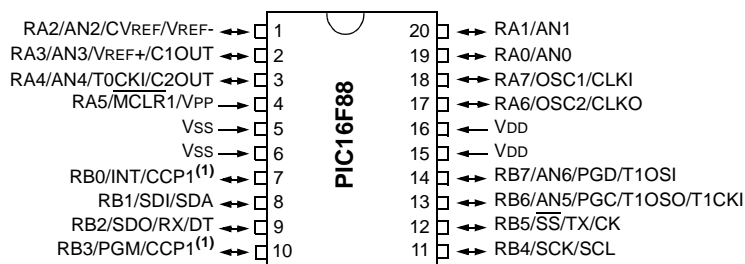
### 20-Pin SSOP



### 18-Pin PDIP, SOIC

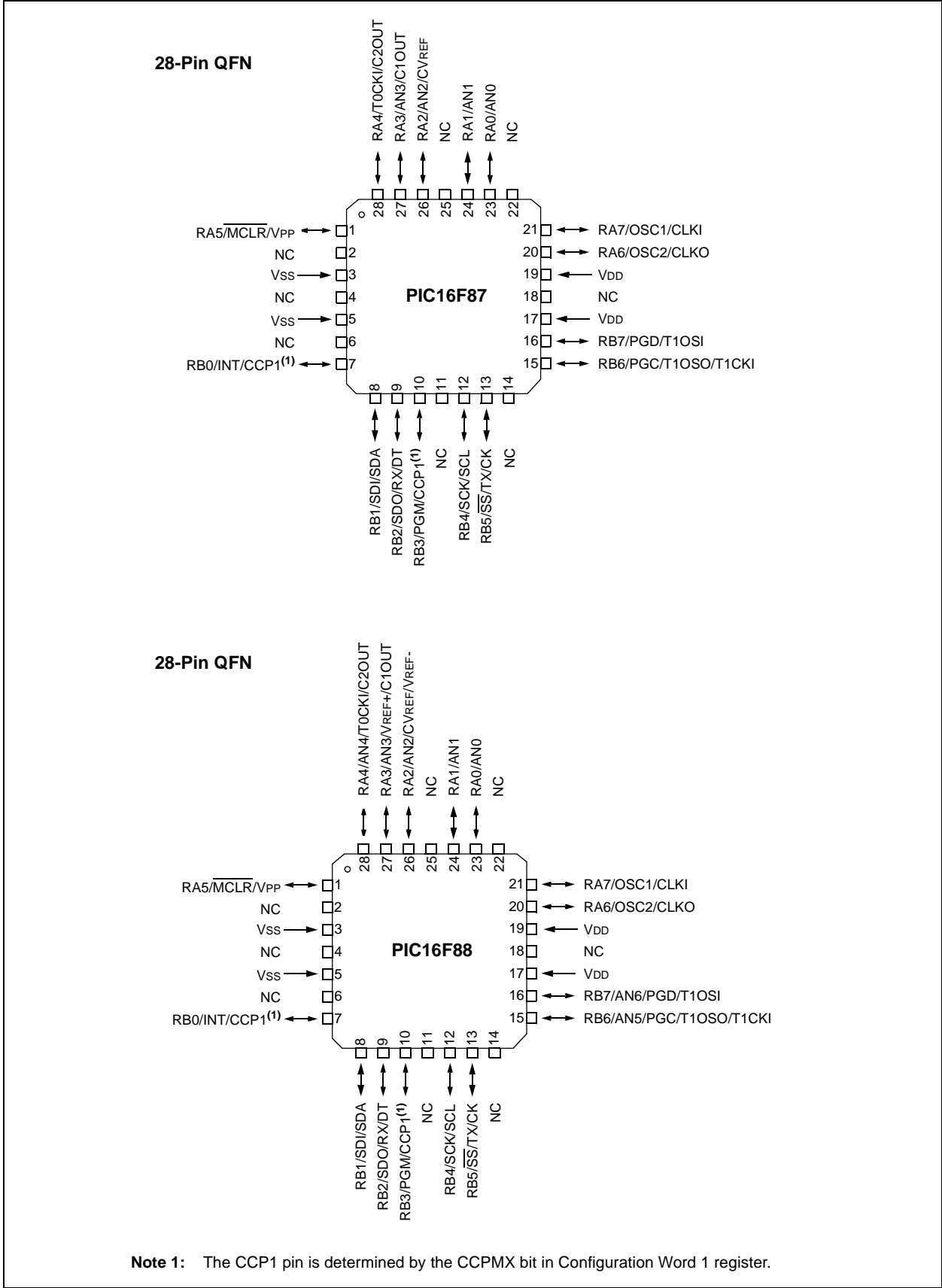


### 20-Pin SSOP



**Note 1:** The CCP1 pin is determined by the CCPMX bit in Configuration Word 1 register.

## Pin Diagrams (Cont'd)



# PIC16F87/88

## Table of Contents

1.0	Device Overview .....	5
2.0	Memory Organization .....	11
3.0	Data EEPROM and Flash Program Memory .....	27
4.0	Oscillator Configurations .....	35
5.0	I/O Ports .....	51
6.0	Timer0 Module .....	67
7.0	Timer1 Module .....	71
8.0	Timer2 Module .....	79
9.0	Capture/Compare/PWM (CCP) Module .....	81
10.0	Synchronous Serial Port (SSP) Module .....	87
11.0	Addressable Universal Synchronous Asynchronous Receiver Transmitter (AUSART) .....	97
12.0	Analog-to-Digital Converter (A/D) Module .....	113
13.0	Comparator Module .....	121
14.0	Comparator Voltage Reference Module .....	127
15.0	Special Features of the CPU .....	129
16.0	Instruction Set Summary .....	149
17.0	Development Support .....	157
18.0	Electrical Characteristics .....	163
19.0	DC and AC Characteristics Graphs and Tables .....	193
20.0	Packaging Information .....	207
	Appendix A: Revision History .....	213
	Appendix B: Device Differences .....	213
	Index .....	215
	The Microchip Web Site .....	223
	Customer Change Notification Service .....	223
	Customer Support .....	223
	Reader Response .....	224
	PIC16F87/88 Product Identification System .....	225

## 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](mailto: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](http://www.microchip.com) to receive the most current information on all of our products.

## 1.0 DEVICE OVERVIEW

This document contains device specific information for the operation of the PIC16F87/88 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. This 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 PIC16F87/88 belongs to the Mid-Range family of the PICmicro® devices. Block diagrams of the devices are shown in Figure 1-1 and Figure 1-2. These devices contain features that are new to the PIC16 product line:

- Low-power modes: RC\_RUN allows the core and peripherals to be clocked from the INTRC, while SEC\_RUN allows the core and peripherals to be clocked from the low-power Timer1. Refer to **Section 4.7 “Power-Managed Modes”** for further details.
- 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 a primary or secondary clock source. Refer to **Section 4.5 “Internal Oscillator Block”** 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 7.0 “Timer1 Module”** for further details.
- Extended Watchdog Timer (WDT) that can have a programmable period from 1 ms to 268s. The WDT has its own 16-bit prescaler. Refer to **Section 15.12 “Watchdog Timer (WDT)”** for further details.
- Two-Speed Start-up: When the oscillator is configured for LP, XT or HS Oscillator mode, this feature will clock the device from the INTRC while the oscillator is warming up. This, in turn, will enable almost immediate code execution. Refer to **Section 15.12.3 “Two-Speed Clock Start-up Mode”** for further details.
- Fail-Safe Clock Monitor: This feature will allow the device to continue operation if the primary or secondary clock source fails by switching over to the INTRC.
- The A/D module has a new register for PIC16 devices named ANSEL. This register allows easier configuration of analog or digital I/O pins.

**TABLE 1-1: AVAILABLE MEMORY IN PIC16F87/88 DEVICES**

Device	Program Flash	Data Memory	Data EEPROM
PIC16F87/88	4K x 14	368 x 8	256 x 8

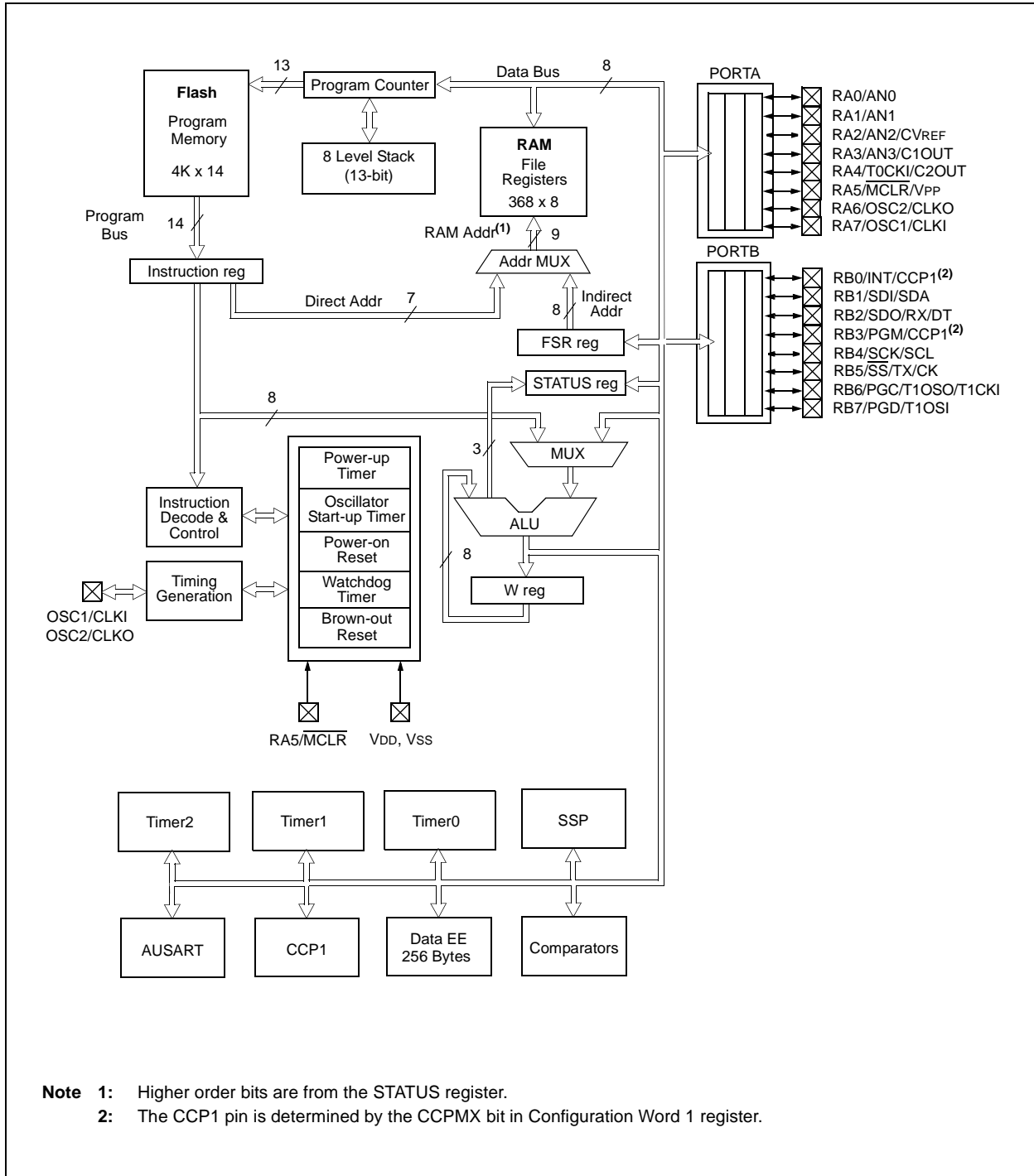
There are 16 I/O pins that are user configurable on a pin-to-pin basis. Some pins are multiplexed with other device functions. These functions include:

- External Interrupt
- Change on PORTB Interrupt
- Timer0 Clock Input
- Low-Power Timer1 Clock/Oscillator
- Capture/Compare/PWM
- 10-bit, 7-channel A/D Converter (PIC16F88 only)
- SPI™/I<sup>2</sup>C™
- Two Analog Comparators
- AUSART
- MCLR (RA5) can be configured as an input

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

# PIC16F87/88

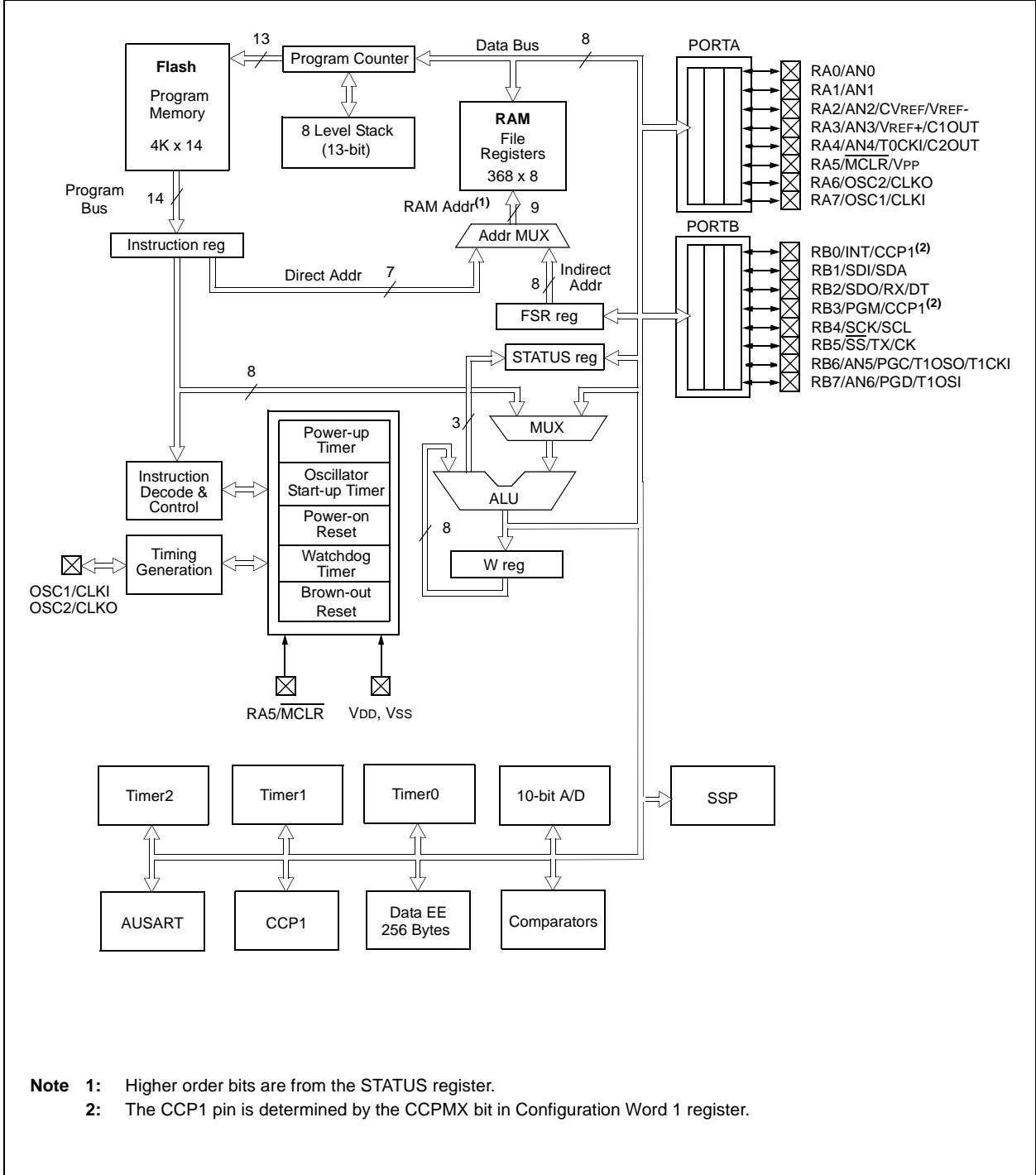
FIGURE 1-1: PIC16F87 DEVICE BLOCK DIAGRAM



- Note 1:** Higher order bits are from the STATUS register.  
**Note 2:** The CCP1 pin is determined by the CCPMX bit in Configuration Word 1 register.



**FIGURE 1-2: PIC16F88 DEVICE BLOCK DIAGRAM**



- Note 1:** Higher order bits are from the STATUS register.
- Note 2:** The CCP1 pin is determined by the CCPMX bit in Configuration Word 1 register.

# PIC16F87/88

**TABLE 1-2: PIC16F87/88 PINOUT DESCRIPTION**

Pin Name	PDIP/ SOIC Pin#	SSOP Pin#	QFN Pin#	I/O/P Type	Buffer Type	Description
RA0/AN0 RA0 AN0	17	19	23	I/O I	TTL Analog	PORTA is a bidirectional I/O port.  Bidirectional I/O pin. Analog input channel 0.
RA1/AN1 RA1 AN1	18	20	24	I/O I	TTL Analog	Bidirectional I/O pin. Analog input channel 1.
RA2/AN2/CVREF/VREF- RA2 AN2 CVREF VREF-(4)	1	1	26	I/O I O I	TTL Analog Analog	Bidirectional I/O pin. Analog input channel 2. Comparator VREF output. A/D reference voltage (Low) input.
RA3/AN3/VREF+/C1OUT RA3 AN3 VREF+(4) C1OUT	2	2	27	I/O I I O	TTL Analog Analog	Bidirectional I/O pin. Analog input channel 3. A/D reference voltage (High) input. Comparator 1 output.
RA4/AN4/T0CKI/C2OUT RA4 AN4(4) T0CKI C2OUT	3	3	28	I/O I I O	ST Analog ST	Bidirectional I/O pin. Analog input channel 4. Clock input to the TMR0 timer/counter. Comparator 2 output.
RA5/MCLR/VPP RA5 MCLR  VPP	4	4	1	I I  P	ST ST  -	Input pin. Master Clear (Reset). Input/programming voltage input. This pin is an active-low Reset to the device. Programming voltage input.
RA6/OSC2/CLKO RA6 OSC2  CLKO	15	17	20	I/O O  O	ST -  -	Bidirectional I/O pin. Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode. 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 RA7 OSC1 CLKI	16	18	21	I/O I I	ST ST/CMOS(3) -	Bidirectional I/O pin. Oscillator crystal input. 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.  
 4: PIC16F88 devices only.  
 5: The CCP1 pin is determined by the CCPMX bit in Configuration Word 1 register.

**TABLE 1-2: PIC16F87/88 PINOUT DESCRIPTION (CONTINUED)**

Pin Name	PDIP/ SOIC Pin#	SSOP Pin#	QFN Pin#	I/O/P Type	Buffer Type	Description
RB0/INT/CCP1 <sup>(5)</sup> RB0 INT CCP1	6	7	7	I/O I I/O	TTL ST <sup>(1)</sup> ST	PORTB is a bidirectional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs.  Bidirectional I/O pin. External interrupt pin. Capture input, Compare output, PWM output.
RB1/SDI/SDA RB1 SDI SDA	7	8	8	I/O I I/O	TTL ST ST	Bidirectional I/O pin. SPI™ data in. I <sup>2</sup> C™ data.
RB2/SDO/RX/DT RB2 SDO RX DT	8	9	9	I/O O I I/O	TTL ST I I/O	Bidirectional I/O pin. SPI data out. AUSART asynchronous receive. AUSART synchronous detect.
RB3/PGM/CCP1 <sup>(5)</sup> RB3 PGM CCP1	9	10	10	I/O I/O I	TTL ST ST	Bidirectional I/O pin. Low-Voltage ICSP™ Programming enable pin. Capture input, Compare output, PWM output.
RB4/SCK/SCL RB4 SCK SCL	10	11	12	I/O I/O I	TTL ST ST	Bidirectional I/O pin. Interrupt-on-change pin. Synchronous serial clock input/output for SPI. Synchronous serial clock Input for I <sup>2</sup> C.
RB5/ $\overline{SS}$ /TX/CK RB5 $\overline{SS}$ TX CK	11	12	13	I/O I O I/O	TTL TTL O I/O	Bidirectional I/O pin. Interrupt-on-change pin. Slave select for SPI in Slave mode. AUSART asynchronous transmit. AUSART synchronous clock.
RB6/AN5/PGC/T1OSO/ T1CKI RB6 AN5 <sup>(4)</sup> PGC T1OSO T1CKI	12	13	15	I/O I I/O O I	TTL I ST <sup>(2)</sup> ST ST	Bidirectional I/O pin. Interrupt-on-change pin. Analog input channel 5. In-Circuit Debugger and programming clock pin. Timer1 oscillator output. Timer1 external clock input.
RB7/AN6/PGD/T1OSI RB7 AN6 <sup>(4)</sup> PGD T1OSI	13	14	16	I/O I I I	TTL I ST <sup>(2)</sup> ST	Bidirectional I/O pin. Interrupt-on-change pin. Analog input channel 6. In-Circuit Debugger and ICSP programming data pin. Timer1 oscillator input.
V <sub>SS</sub>	5	5, 6	3, 5	P	–	Ground reference for logic and I/O pins.
V <sub>DD</sub>	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.  
**4:** PIC16F88 devices only.  
**5:** The CCP1 pin is determined by the CCPMX bit in Configuration Word 1 register.

# PIC16F87/88

---

NOTES:

## 2.0 MEMORY ORGANIZATION

There are two memory blocks in the PIC16F87/88 devices. 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 PIC16F87/88 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”**.

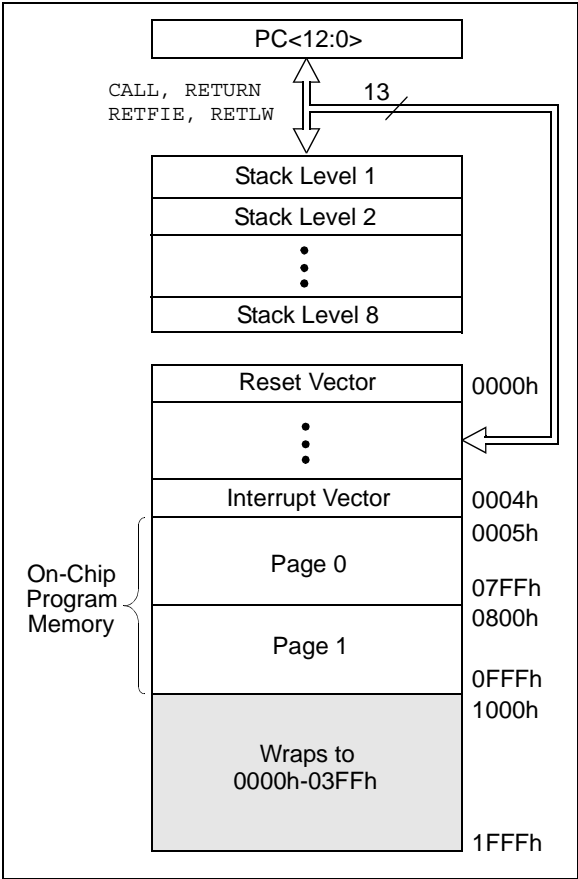
Additional information on device memory may be found in the “PICmicro® Mid-Range MCU Family Reference Manual” (DS33023).

### 2.1 Program Memory Organization

The PIC16F87/88 devices have a 13-bit program counter capable of addressing an 8K x 14 program memory space. For the PIC16F87/88, the first 4K x 14 (0000h-0FFFh) is physically implemented (see Figure 2-1). 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: PIC16F87/88**



### 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 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.

# PIC16F87/88

## 2.2.1 GENERAL PURPOSE REGISTER FILE

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

**FIGURE 2-2: PIC16F87 REGISTER FILE MAP**

File Address		File Address		File Address		File Address	
Indirect addr. <sup>(*)</sup>	00h	Indirect addr. <sup>(*)</sup>	80h	Indirect addr. <sup>(*)</sup>	100h	Indirect addr. <sup>(*)</sup>	180h
TMR0	01h	OPTION_REG	81h	TMR0	101h	OPTION_REG	181h
PCL	02h	PCL	82h	PCL	102h	PCL	182h
STATUS	03h	STATUS	83h	STATUS	103h	STATUS	183h
FSR	04h	FSR	84h	FSR	104h	FSR	184h
PORTA	05h	TRISA	85h	WDTCON	105h		185h
PORTB	06h	TRISB	86h	PORTB	106h	TRISB	186h
	07h		87h		107h		187h
	08h		88h		108h		188h
	09h		89h		109h		189h
PCLATH	0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah
INTCON	0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh
PIR1	0Ch	PIE1	8Ch	EEDATA	10Ch	EECON1	18Ch
PIR2	0Dh	PIE2	8Dh	EEADR	10Dh	EECON2	18Dh
TMR1L	0Eh	PCON	8Eh	EEDATH	10Eh	Reserved <sup>(1)</sup>	18Eh
TMR1H	0Fh	OSCCON	8Fh	EEADRH	10Fh	Reserved <sup>(1)</sup>	18Fh
T1CON	10h	OSCTUNE	90h		110h		190h
TMR2	11h		91h	General Purpose Register 16 Bytes		General Purpose Register 16 Bytes	
T2CON	12h	PR2	92h				
SSPBUF	13h	SSPADD	93h				
SSPCON	14h	SSPSTAT	94h				
CCPR1L	15h		95h				
CCPR1H	16h		96h				
CCP1CON	17h		97h				
RCSTA	18h	TXSTA	98h				
TXREG	19h	SPBRG	99h				
RCREG	1Ah		9Ah				
	1Bh		9Bh				
	1Ch	CMCON	9Ch				
	1Dh	CVRCON	9Dh				
	1Eh		9Eh				
	1Fh		9Fh				
General Purpose Register 96 Bytes	20h	General Purpose Register 80 Bytes	A0h				
			EFh F0h	120h	1A0h		
	7Fh	accesses 70h-7Fh	FFh	accesses 70h-7Fh	16Fh 170h	accesses 70h-7Fh	1EFh 1F0h
Bank 0		Bank 1		Bank 2		Bank 3	

Unimplemented data memory locations, read as '0'.  
 \* Not a physical register.

**Note 1:** This register is reserved, maintain this register clear.

**FIGURE 2-3: PIC16F88 REGISTER FILE MAP**

File Address		File Address		File Address		File Address	
Indirect addr. (*)	00h	Indirect addr. (*)	80h	Indirect addr. (*)	100h	Indirect addr. (*)	180h
TMR0	01h	OPTION_REG	81h	TMR0	101h	OPTION_REG	181h
PCL	02h	PCL	82h	PCL	102h	PCL	182h
STATUS	03h	STATUS	83h	STATUS	103h	STATUS	183h
FSR	04h	FSR	84h	FSR	104h	FSR	184h
PORTA	05h	TRISA	85h	WDTCON	105h		185h
PORTB	06h	TRISB	86h	PORTB	106h	TRISB	186h
	07h		87h		107h		187h
	08h		88h		108h		188h
	09h		89h		109h		189h
PCLATH	0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah
INTCON	0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh
PIR1	0Ch	PIE1	8Ch	EEDATA	10Ch	EECON1	18Ch
PIR2	0Dh	PIE2	8Dh	EEADR	10Dh	EECON2	18Dh
TMR1L	0Eh	PCON	8Eh	EEDATH	10Eh	Reserved <sup>(1)</sup>	18Eh
TMR1H	0Fh	OSCCON	8Fh	EEADRH	10Fh	Reserved <sup>(1)</sup>	18Fh
T1CON	10h	OSCTUNE	90h		110h		190h
TMR2	11h		91h	General Purpose Register 16 Bytes		General Purpose Register 16 Bytes	
T2CON	12h	PR2	92h				
SSPBUF	13h	SSPADD	93h				
SSPCON	14h	SSPSTAT	94h				
CCPR1L	15h		95h				
CCPR1H	16h		96h				
CCP1CON	17h		97h				
RCSTA	18h	TXSTA	98h				
TXREG	19h	SPBRG	99h				
RCREG	1Ah		9Ah				
	1Bh	ANSEL	9Bh				
	1Ch	CMCON	9Ch				
	1Dh	CVRCON	9Dh				
ADRESH	1Eh	ADRESL	9Eh				
ADCON0	1Fh	ADCON1	9Fh				
General Purpose Register 96 Bytes	20h	General Purpose Register 80 Bytes	A0h				
				120h	1A0h		
				16Fh	1EFh		
				170h	1F0h		
		accesses 70h-7Fh		accesses 70h-7Fh		accesses 70h-7Fh	
	7Fh		FFh		17Fh		1FFh

Bank 0
Bank 1
Bank 2
Bank 3

Unimplemented data memory locations, read as '0'.

\* Not a physical register.

**Note 1:** This register is reserved, maintain this register clear.

# PIC16F87/88

## 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
<b>Bank 0</b>											
00h <sup>(2)</sup>	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	26, 135
01h	TMR0	Timer0 Module Register								xxxx xxxx	69
02h <sup>(2)</sup>	PCL	Program Counter (PC) Least Significant Byte								0000 0000	
03h <sup>(2)</sup>	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxxx	17
04h <sup>(2)</sup>	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	135
05h	PORTA	PORTA Data Latch when written; PORTA pins when read (PIC16F87) PORTA Data Latch when written; PORTA pins when read (PIC16F88)								xxxx 0000 xxxx0 0000	52
06h	PORTB	PORTB Data Latch when written; PORTB pins when read (PIC16F87) PORTB Data Latch when written; PORTB pins when read (PIC16F88)								xxxx xxxx 00xx xxxx	58
07h	—	Unimplemented								—	—
08h	—	Unimplemented								—	—
09h	—	Unimplemented								—	—
0Ah <sup>(1,2)</sup>	PCLATH	—	—	—	Write Buffer for the Upper 5 bits of the Program Counter				---0 0000	135	
0Bh <sup>(2)</sup>	INTCON	GIE	PEIE	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	19, 69, 77
0Ch	PIR1	—	ADIF <sup>(4)</sup>	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	-000 0000	21, 77
0Dh	PIR2	OSFIF	CMIF	—	EEIF	—	—	—	—	00-0 ----	23, 34
0Eh	TMR1L	Holding Register for the Least Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	77, 83
0Fh	TMR1H	Holding Register for the Most Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	77, 83
10h	T1CON	—	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{T1SYNC}$	TMR1CS	TMR1ON	-000 0000	72, 83
11h	TMR2	Timer2 Module Register								0000 0000	80, 85
12h	T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	80, 85
13h	SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register								xxxx xxxx	90, 95
14h	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	89, 95
15h	CCPR1L	Capture/Compare/PWM Register 1 (LSB)								xxxx xxxx	83, 85
16h	CCPR1H	Capture/Compare/PWM Register 1 (MSB)								xxxx xxxx	83, 85
17h	CCP1CON	—	—	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	81, 83
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	98, 99
19h	TXREG	AUSART Transmit Data Register								0000 0000	103
1Ah	RCREG	AUSART Receive Data Register								0000 0000	105
1Bh	—	Unimplemented								—	—
1Ch	—	Unimplemented								—	—
1Dh	—	Unimplemented								—	—
1Eh	ADRESH <sup>(4)</sup>	A/D Result Register High Byte								xxxx xxxx	120
1Fh	ADCON0 <sup>(4)</sup>	ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	—	ADON	0000 00-0	114, 120

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

**2:** These registers can be addressed from any bank.

**3:** RA5 is an input only; the state of the TRISA5 bit has no effect and will always read '1'.

**4:** PIC16F88 device only.



**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		
<b>Bank 1</b>													
80h <sup>(2)</sup>	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	26, 135		
81h	OPTION_REG	RBP $\bar{U}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	18, 69		
82h <sup>(2)</sup>	PCL	Program Counter (PC) Least Significant Byte								0000 0000	135		
83h <sup>(2)</sup>	STATUS	IRP	RP1	RP0	$\bar{T}O$	$\bar{P}D$	Z	DC	C	0001 1xxx	17		
84h <sup>(2)</sup>	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	135		
85h	TRISA	TRISA7	TRISA6	TRISA5 <sup>(3)</sup>	PORTA Data Direction Register (TRISA<4:0>)							1111 1111	52, 126
86h	TRISB	PORTB Data Direction Register								1111 1111	58, 85		
87h	—	Unimplemented								—	—		
88h	—	Unimplemented								—	—		
89h	—	Unimplemented								—	—		
8Ah <sup>(1,2)</sup>	PCLATH	—	—	—	Write Buffer for the Upper 5 bits of the Program Counter					---0 0000	135		
8Bh <sup>(2)</sup>	INTCON	GIE	PEIE	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBF	0000 000x	19, 69, 77		
8Ch	PIE1	—	ADIE <sup>(4)</sup>	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	-000 0000	20, 80		
8Dh	PIE2	OSFIE	CMIE	—	EEIE	—	—	—	—	00-0 ----	22, 34		
8Eh	PCON	—	—	—	—	—	—	$\bar{P}OR$	$\bar{B}OR$	---- -00q	24		
8Fh	OSCCON	—	IRCF2	IRCF1	IRCF0	OSTS	IOFS	SCS1	SCS0	-000 0000	40		
90h	OSCTUNE	—	—	TUN5	TUN4	TUN3	TUN2	TUN1	TUN0	--00 0000	38		
91h	—	Unimplemented								—	—		
92h	PR2	Timer2 Period Register								1111 1111	80, 85		
93h	SSPADD	Synchronous Serial Port (I <sup>2</sup> C™ mode) Address Register								0000 0000	95		
94h	SSPSTAT	SMP	CKE	D/ $\bar{A}$	P	S	R/ $\bar{W}$	UA	BF	0000 0000	88, 95		
95h	—	Unimplemented								—	—		
96h	—	Unimplemented								—	—		
97h	—	Unimplemented								—	—		
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	97, 99		
99h	SPBRG	Baud Rate Generator Register								0000 0000	99, 103		
9Ah	—	Unimplemented								—	—		
9Bh	ANSEL <sup>(4)</sup>	—	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0	-111 1111	120		
9Ch	CMCON	C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0	0000 0111	121, 126, 128		
9Dh	CVRCON	CVREN	CVROE	CVRR	—	CVR3	CVR2	CVR1	CVR0	000- 0000	126, 128		
9Eh	ADRESL <sup>(4)</sup>	A/D Result Register Low Byte								xxxx xxxx	120		
9Fh	ADCON1 <sup>(4)</sup>	ADFM	ADCS2	VCFG1	VCFG0	—	—	—	—	0000 ----	52, 115, 120		

**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:** The upper byte of the program counter is not directly accessible. PCLATH is a holding register for PC<12:8>, whose contents are transferred to the upper byte of the program counter.
- 2:** These registers can be addressed from any bank.
- 3:** RA5 is an input only; the state of the TRISA5 bit has no effect and will always read '1'.
- 4:** PIC16F88 device only.

# PIC16F87/88

**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
<b>Bank 2</b>											
100h <sup>(2)</sup>	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	26, 135
101h	TMR0	Timer0 Module Register								xxxx xxxx	69
102h <sup>(2)</sup>	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	135
103h <sup>(2)</sup>	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	17
104h <sup>(2)</sup>	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	135
105h	WDTCN	—	—	—	WDTPS3	WDTPS2	WDTPS1	WDTPS0	SWDTEN	---0 1000	142
106h	PORTB	PORTB Data Latch when written; PORTB pins when read (PIC16F87) PORTB Data Latch when written; PORTB pins when read (PIC16F88)								xxxx xxxx 00xx xxxxx	58
107h	—	Unimplemented								—	—
108h	—	Unimplemented								—	—
109h	—	Unimplemented								—	—
10Ah <sup>(1,2)</sup>	PCLATH	—	—	—	Write Buffer for the Upper 5 bits of the Program Counter				---0 0000	135	
10Bh <sup>(2)</sup>	INTCON	GIE	PEIE	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	19, 69, 77
10Ch	EEDATA	EEPROM/Flash Data Register Low Byte								xxxx xxxx	34
10Dh	EEADR	EEPROM/Flash Address Register Low Byte								xxxx xxxx	34
10Eh	EEDATH	—	—	EEPROM/Flash Data Register High Byte				—xx xxxxx	34		
10Fh	EEADRH	—	—	—	—	EEPROM/Flash Address Register High Byte				--- xxxxx	34
<b>Bank 3</b>											
180h <sup>(2)</sup>	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	135
181h	OPTION_REG	$\overline{RBP}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	18, 69
182h <sup>(2)</sup>	PCL	Program Counter (PC) Least Significant Byte								0000 0000	135
183h <sup>(2)</sup>	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	17
184h <sup>(2)</sup>	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	135
185h	—	Unimplemented								—	—
186h	TRISB	PORTB Data Direction Register								1111 1111	58, 83
187h	—	Unimplemented								—	—
188h	—	Unimplemented								—	—
189h	—	Unimplemented								—	—
18Ah <sup>(1,2)</sup>	PCLATH	—	—	—	Write Buffer for the Upper 5 bits of the Program Counter				---0 0000	135	
18Bh <sup>(2)</sup>	INTCON	GIE	PEIE	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	19, 69, 77
18Ch	EECON1	EEPGD	—	—	FREE	WRERR	WREN	WR	RD	x--x x000	28, 34
18Dh	EECON2	EEPROM Control Register 2 (not a physical register)								---- ----	34
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:** The upper byte of the program counter is not directly accessible. PCLATH is a holding register for PC<12:8>, whose contents are transferred to the upper byte of the program counter.

**2:** These registers can be addressed from any bank.

**3:** RA5 is an input only; the state of the TRISA5 bit has no effect and will always read '1'.

**4:** PIC16F88 device only.

## 2.2.2.1 STATUS Register

The STATUS register, shown in Register 2-1, contains the arithmetic status of the ALU, the Reset status 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  $\overline{\text{TO}}$  and  $\overline{\text{PD}}$  bits are not writable. Therefore, the result of an instruction with the STATUS register as destination may be different than intended.

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

It is recommended, therefore, that only `BCF`, `BSF`, `SWAPF` and `MOVWF` instructions are used to alter the STATUS register because these instructions do not affect the Z, C or DC bits from the STATUS register. For other instructions not affecting any Status bits, see **Section 16.0 "Instruction Set Summary"**.

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

### REGISTER 2-1: STATUS: ARITHMETIC 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	$\overline{\text{TO}}$	$\overline{\text{PD}}$	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  **$\overline{\text{TO}}$ :** Time-out bit  
 1 = After power-up, `CLRWDT` instruction or `SLEEP` instruction  
 0 = A WDT time-out occurred
- bit 3  **$\overline{\text{PD}}$ :** Power-Down bit  
 1 = After power-up or by the `CLRWDT` instruction  
 0 = By execution of the `SLEEP` instruction
- bit 2 **Z:** Zero bit  
 1 = The result of an arithmetic or logic operation is zero  
 0 = The result of an arithmetic or logic operation is not zero
- bit 1 **DC:** Digit carry/borrow bit (`ADDWF`, `ADDLW`, `SUBLW` and `SUBWF` instructions)<sup>(1)</sup>  
 1 = A carry-out from the 4th low-order bit of the result occurred  
 0 = No carry-out from the 4th low-order bit of the result
- bit 0 **C:** Carry/borrow bit (`ADDWF`, `ADDLW`, `SUBLW` and `SUBWF` instructions)<sup>(1,2)</sup>  
 1 = A carry-out from the Most Significant bit of the result occurred  
 0 = No carry-out from the Most Significant bit of the result occurred

- Note 1:** For borrow, the polarity is reversed. A subtraction is executed by adding the two's complement of the second operand.
- 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

# PIC16F87/88

## 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. Although the prescaler can be assigned to either the WDT or Timer0, but not both, a new divide counter is implemented in the WDT circuit to give multiple WDT time-out selections. This allows TMR0 and WDT to each have their own scaler. Refer to **Section 15.12 “Watchdog Timer (WDT)”** for further details.

### REGISTER 2-2: OPTION\_REG: OPTION CONTROL 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
RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0

bit 7

bit 0

- bit 7 **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 RA4/T0CKI/C2OUT pin  
0 = Internal instruction cycle clock (CLKO)
- bit 4 **T0SE:** TMR0 Source Edge Select bit  
1 = Increment on high-to-low transition on RA4/T0CKI/C2OUT pin  
0 = Increment on low-to-high transition on RA4/T0CKI/C2OUT pin
- bit 3 **PSA:** Prescaler Assignment bit  
1 = Prescaler is assigned to the WDT  
0 = Prescaler is assigned to the Timer0 module
- bit 2-0 **PS<2:0>:** Prescaler Rate Select bits

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

**Legend:**

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

## 2.2.2.3 INTCON Register

The INTCON register is a readable and writable register 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 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	INT0IE	RBIE	TMR0IF	INT0IF	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 **INT0IE:** 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 **INT0IF:** 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

# PIC16F87/88

## 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	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	ADIE <sup>(1)</sup>	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE
bit 7							bit 0

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

bit 6 **ADIE:** A/D Converter Interrupt Enable bit<sup>(1)</sup>

1 = Enabled  
0 = Disabled

**Note 1:** This bit is only implemented on the PIC16F88. The bit will read '0' on the PIC16F87.

bit 5 **RCIE:** AUSART Receive Interrupt Enable bit

1 = Enabled  
0 = Disabled

bit 4 **TXIE:** AUSART Transmit Interrupt Enable bit

1 = Enabled  
0 = Disabled

bit 3 **SSPIE:** Synchronous Serial Port (SSP) Interrupt Enable bit

1 = Enabled  
0 = Disabled

bit 2 **CCP1IE:** CCP1 Interrupt Enable bit

1 = Enabled  
0 = Disabled

bit 1 **TMR2IE:** TMR2 to PR2 Match Interrupt Enable bit

1 = Enabled  
0 = Disabled

bit 0 **TMR1IE:** TMR1 Overflow Interrupt Enable bit

1 = Enabled  
0 = Disabled

#### Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

## 2.2.2.5 PIR1 Register

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 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	R-0	R-0	R-0	R/W-0	R/W-0	R/W-0
	—	ADIF <sup>(1)</sup>	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF
bit 7								bit 0

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

bit 6 **ADIF:** A/D Converter Interrupt Flag bit<sup>(1)</sup>

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

**Note 1:** This bit is only implemented on the PIC16F88. The bit will read '0' on the PIC16F87.

bit 5 **RCIF:** AUSART Receive Interrupt Flag bit

1 = The AUSART receive buffer is full (cleared by reading RCREG)  
0 = The AUSART receive buffer is not full

bit 4 **TXIF:** AUSART Transmit Interrupt Flag bit

1 = The AUSART transmit buffer is empty (cleared by writing to TXREG)  
0 = The AUSART transmit buffer is full

bit 3 **SSPIF:** Synchronous Serial Port (SSP) Interrupt Flag bit

1 = The transmission/reception is complete (must be cleared in software)  
0 = Waiting to transmit/receive

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 Interrupt Flag bit

1 = A 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 = The TMR1 register overflowed (must be cleared in software)  
0 = The 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

# PIC16F87/88

## 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)

R/W-0	R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0
OSFIE	CMIE	—	EEIE	—	—	—	—
bit 7							bit 0

bit 7 **OSFIE:** Oscillator Fail Interrupt Enable bit

1 = Enabled  
0 = Disabled

bit 6 **CMIE:** Comparator Interrupt Enable bit

1 = Enabled  
0 = Disabled

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

bit 4 **EEIE:** EEPROM Write Operation Interrupt Enable bit

1 = Enabled  
0 = Disabled

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 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)

	R/W-0	R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0
	OSFIF	CMIF	—	EEIF	—	—	—	—
bit 7								bit 0

- bit 7 **OSFIF:** Oscillator Fail Interrupt Flag bit  
 1 = System oscillator failed, clock input has changed to INTRC (must be cleared in software)  
 0 = System clock operating
- bit 6 **CMIF:** Comparator Interrupt Flag bit  
 1 = Comparator input has changed (must be cleared in software)  
 0 = Comparator input has not changed
- bit 5 **Unimplemented:** Read as '0'
- bit 4 **EEIF:** EEPROM Write Operation Interrupt Flag bit  
 1 = The write operation completed (must be cleared in software)  
 0 = The write operation is not complete or has not been started
- 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

# PIC16F87/88

## 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 enable bit, GIE (INTCON<7>). User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

**Note:**  $\overline{\text{BOR}}$  is unknown on Power-on Reset. It must then be set by the user and checked on subsequent Resets to see if  $\overline{\text{BOR}}$  is clear, indicating a brown-out has occurred. The  $\overline{\text{BOR}}$  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).

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.

### 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
	—	—	—	—	—	—	$\overline{\text{POR}}$	$\overline{\text{BOR}}$
bit 7								bit 0

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

bit 1     **POR:** Power-on Reset Status bit

1 = No Power-on Reset occurred

0 = A Power-on Reset occurred (must be set in software after a Power-on Reset occurs)

bit 0     **BOR:** Brown-out Reset Status bit

1 = No Brown-out Reset occurred

0 = A Brown-out Reset occurred (must be set in software after a 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

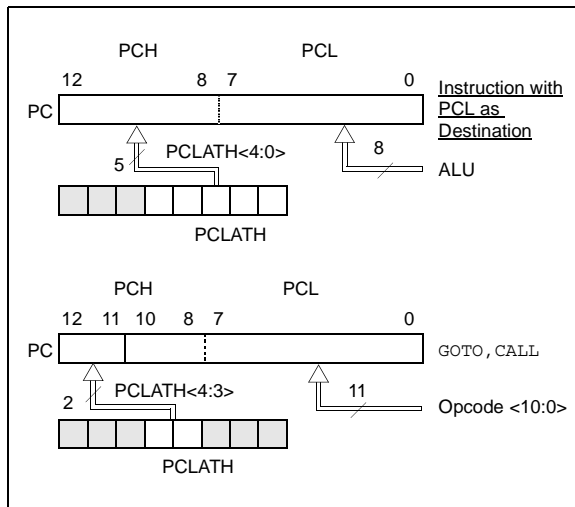
'0' = Bit is cleared

x = Bit is unknown

## 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 2-4 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 2-4: 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".

### 2.3.2 STACK

The PIC16F87/88 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 Program Memory Paging

All PIC16F87/88 devices are capable of addressing a continuous 8K word block of program memory. The CALL and GOTO instructions provide only 11 bits of address to allow branching within any 2K program memory page. When doing a CALL or GOTO instruction, the upper 2 bits of the address are provided by PCLATH<4:3>. When doing a CALL or GOTO instruction, the user must ensure that the page select bits are programmed so that the desired program memory page is addressed. If a return from a CALL instruction (or interrupt) is executed, the entire 13-bit PC is popped off the stack. Therefore, manipulation of the PCLATH<4:3> bits is not required for the RETURN instructions (which POPs the address from the stack).

**Note:** The contents of the PCLATH register are unchanged after a RETURN or RETFIE instruction is executed. The user must rewrite the contents of the PCLATH register for any subsequent subroutine calls or GOTO instructions.

Example 2-1 shows the calling of a subroutine in page 1 of the program memory. This example assumes that PCLATH is saved and restored by the Interrupt Service Routine (if interrupts are used).

### EXAMPLE 2-1: CALL OF A SUBROUTINE IN PAGE 1 FROM PAGE 0

```

ORG 0x500
BCF PCLATH, 4
BSF PCLATH, 3 ;Select page 1
                ;(800h-FFFh)
CALL SUB1_P1 ;Call subroutine in
:                ;page 1 (800h-FFFh)
:
ORG 0x900 ;page 1 (800h-FFFh)
SUB1_P1
:                ;called subroutine
                ;page 1 (800h-FFFh)
:
RETURN ;return to
        ;Call subroutine
        ;in page 0
        ;(000h-7FFh)
    
```

# PIC16F87/88

## 2.5 Indirect Addressing, INDF and FSR Registers

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

Indirect addressing is possible by using the INDF register. Any instruction using the INDF register actually accesses the register pointed to by the File Select Register, FSR. Reading the INDF register itself, indirectly (FSR = 0) will read 00h. Writing to the INDF register indirectly results in a no operation (although status bits may be affected). An effective 9-bit address is obtained by concatenating the 8-bit FSR register and the IRP bit (STATUS<7>), as shown in Figure 2-5.

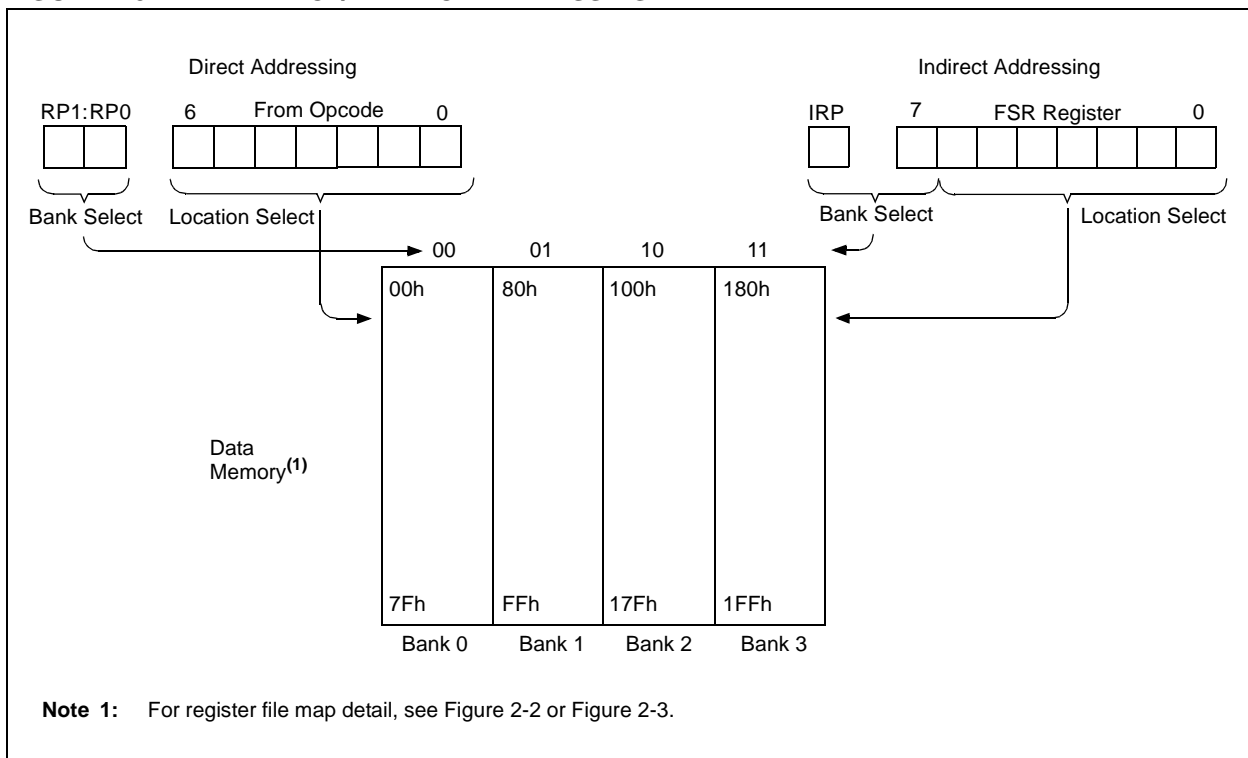
A simple program to clear RAM locations 20h-2Fh using indirect addressing is shown in Example 2-2.

### EXAMPLE 2-2: INDIRECT ADDRESSING

```

MOV LW 0x20 ;initialize pointer
MOV WF FSR ;to RAM
NEXT   CLR F INDF ;clear INDF register
      INC F FSR, F ;inc pointer
      BT FSS FSR, 4 ;all done?
      GOTO NEXT ;no clear next
CONTINUE
      : ;yes continue
    
```

FIGURE 2-5: DIRECT/INDIRECT ADDRESSING



## 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. The PIC16F87/88 devices have 256 bytes of data EEPROM with an address range from 00h to 0FFh. 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. The PIC16F87/88 devices have 4K words of program Flash with an address range from 0000h to 0FFFh. 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  $\overline{\text{MCLR}}$ , 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.

# PIC16F87/88

## 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
- 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	U = Unimplemented bit, read as '0'	S = Set only
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

## 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 clear, to indicate the end of the program cycle.

### EXAMPLE 3-2: DATA EEPROM WRITE

```
BANKSEL EECON1     ; Select Bank of
                ; EECON1
BTFSCL 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          ;
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
```

# PIC16F87/88

## 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 microcontroller 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 setup 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.



## 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     ADDRH, W    ;
        MOVWF   EEADRH      ; 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)

```

# PIC16F87/88

## 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 15-1). 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  $EEP GD$  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  $EECON1$

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).

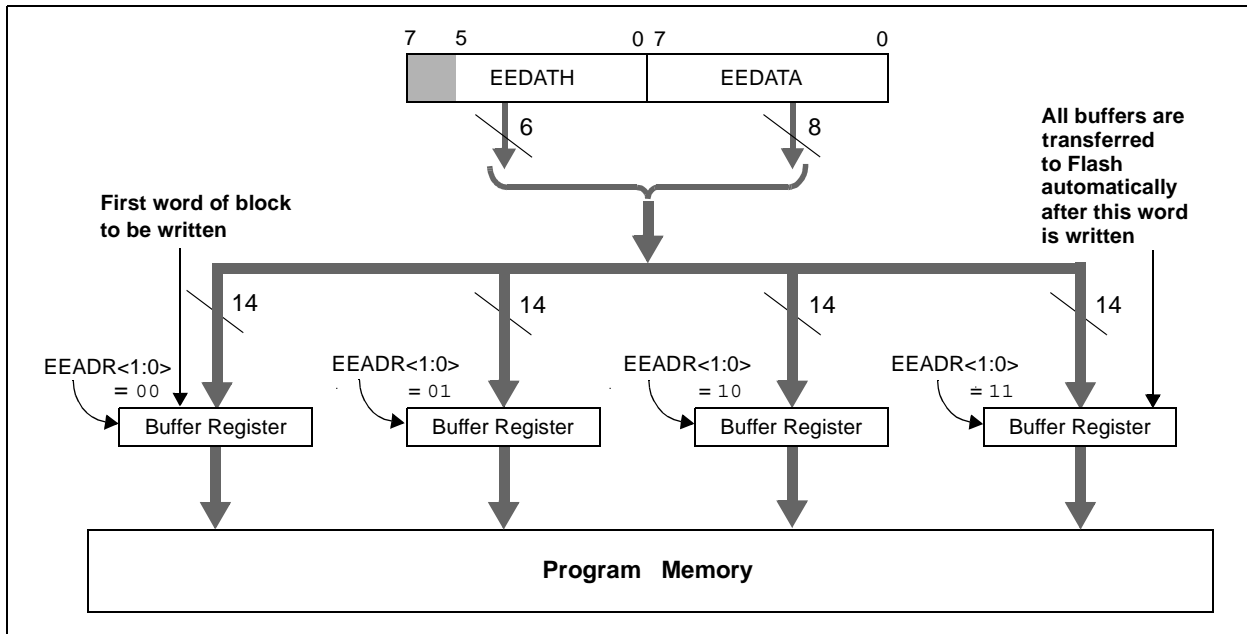
There are 4 buffer register words and all four locations **MUST** be written to with correct data.

After the “ $BSF\ EECON1, WR$ ” instruction, if  $EEADR \neq \text{xxxxxx}11$ , 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 1 cycle.

After the “ $BSF\ EECON1, WR$ ” instruction, if  $EEADR = \text{xxxxxx}11$ , 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 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



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
        NOP                         ;instructions here are ignored as processor

        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
    
```

# PIC16F87/88

## 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 15.1 “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				---- xxxx	---- uuuu
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	OSFIF	CMIF	—	EEIF	—	—	—	—	00-0 ----	00-0 ----
8Dh	PIE2	OSFIE	CMIE	—	EEIE	—	—	—	—	00-0 ----	00-0 ----

**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.

## 4.0 OSCILLATOR CONFIGURATIONS

### 4.1 Oscillator Types

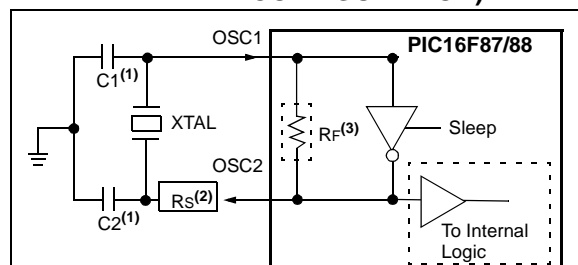
The PIC16F87/88 can be operated in eight different oscillator modes. The user can program three configuration bits (FOSC2:FOSC0) to select one of these eight modes (modes 5-8 are new PIC16 oscillator configurations):

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 PIC16F87/88 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 OSCILLATOR CONFIGURATION)**



- Note1:** See Table 4-1 for typical values of C1 and C2.
- 2:** A series resistor ( $R_s$ ) may be required for AT strip cut crystals.
  - 3:**  $R_f$  varies with the crystal chosen (typically between 2 M $\Omega$  to 10 M $\Omega$ ).

**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
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.

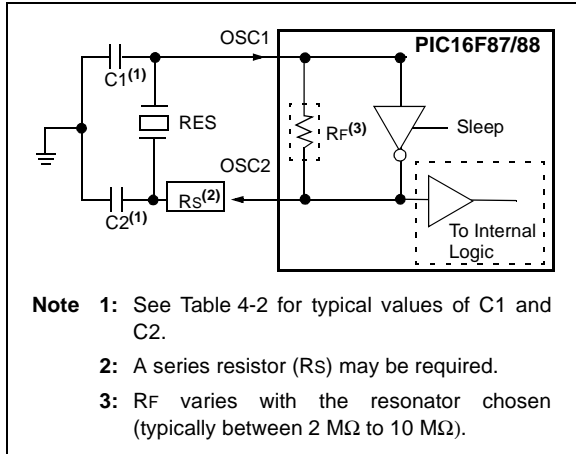
Different capacitor values may be required to produce acceptable oscillator operation. The user should test the performance of the oscillator over the expected VDD and temperature range for the application.

See the notes following this table for additional information.

- Note 1:** Higher capacitance increases the stability of 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:**  $R_s$  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.

# PIC16F87/88

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



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

Typical Capacitor Values Used:			
Mode	Freq	OSC1	OSC2
XT	455 kHz	56 pF	56 pF
	2.0 MHz	47 pF	47 pF
	4.0 MHz	33 pF	33 pF
HS	8.0 MHz	27 pF	27 pF
	16.0 MHz	22 pF	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.

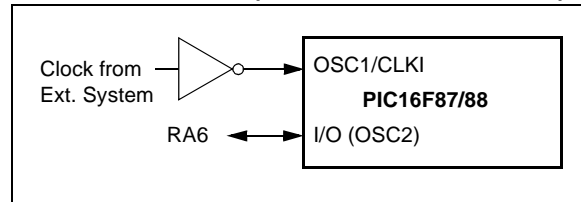
**Note:** When using resonators with frequencies above 3.5 MHz, the use of HS mode, rather than XT mode, is recommended. HS mode may be used at any VDD for which the controller is rated. If HS is selected, it is possible that the gain of the oscillator will overdrive the resonator. Therefore, a series 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)**

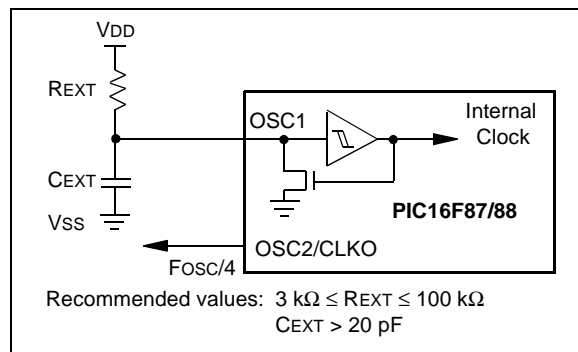


## 4.4 RC Oscillator

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 (R<sub>EXT</sub>) and capacitor (C<sub>EXT</sub>) values and the operating temperature. In addition to this, the oscillator frequency will vary from unit to unit due to normal manufacturing variation. Furthermore, the difference in lead frame capacitance between package types will also affect the oscillation frequency, especially for low C<sub>EXT</sub> values. The user also needs to take into account variation due to tolerance of external R and C components used. Figure 4-4 shows how the R/C combination is connected.

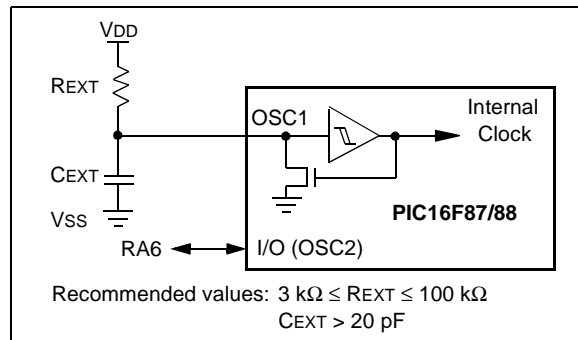
In the RC Oscillator mode, the oscillator frequency divided by 4 is available on the OSC2 pin. This signal may be used for test purposes or to synchronize other logic.

**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 PIC16F87/88 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 six 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
- Two-Speed Start-up
- Fail-Safe Clock Monitor

These features are discussed in greater detail in **Section 15.0 “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 (page 40).

**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).

# PIC16F87/88

## 4.5.1 INTRC MODES

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

- In INTIO1 mode, the OSC2 pin outputs  $F_{osc}/4$ , 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.

## 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  $\pm 12.5\%$ .

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  $8 * 32 \mu s = 256 \mu s$ ); 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



## 4.6 Clock Sources and Oscillator Switching

The PIC16F87/88 devices include a feature that allows the system clock source to be switched from the main oscillator to an alternate low-frequency clock source. PIC16F87/88 devices offer three alternate clock sources. When enabled, these give additional options for switching to the various power-managed operating modes.

Essentially, there are three clock sources for these devices:

- Primary oscillators
- Secondary oscillators
- Internal oscillator block (INTRC)

The **primary oscillators** include the External Crystal and Resonator modes, the External RC modes, the External Clock mode and the internal oscillator block. The particular mode is defined on POR by the contents of Configuration Word 1. The details of these modes are covered earlier in this chapter.

The **secondary oscillators** are those external sources not connected to the OSC1 or OSC2 pins. These sources may continue to operate even after the controller is placed in a power-managed mode.

PIC16F87/88 devices offer the Timer1 oscillator as a secondary oscillator. This oscillator continues to run when a SLEEP instruction is executed and is often the time base for functions such as a real-time clock.

Most often, a 32.768 kHz watch crystal is connected between the RB6/T1OSO and RB7/T1OSI pins. Like the LP mode oscillator circuit, loading capacitors are also connected from each pin to ground. The Timer1 oscillator is discussed in greater detail in **Section 7.6 “Timer1 Oscillator”**.

In addition to being a primary clock source, the **internal oscillator block** is available as a power-managed mode clock source. The 31.25 kHz INTRC source is also used as the clock source for several special features, such as the WDT, Fail-Safe Clock Monitor, Power-up Timer and Two-Speed Start-up.

The clock sources for the PIC16F87/88 devices are shown in Figure 4-6. See **Section 7.0 “Timer1 Module”** for further details of the Timer1 oscillator. See **Section 15.1 “Configuration Bits”** for Configuration register details.

### 4.6.1 OSCCON REGISTER

The OSCCON register (Register 4-2) controls several aspects of the system clock's operation, both in full power operation and in power-managed modes.

The System Clock Select bits, SCS1:SCS0, select the clock source that is used when the device is operating in power-managed modes. When the bits are cleared (SCS<1:0> = 00), the system clock source comes from the main oscillator that is selected by the

FOSC2:FOSC0 configuration bits in Configuration Word 1 register. When the bits are set in any other manner, the system clock source is provided by the Timer1 oscillator (SCS1:SCS0 = 01) or from the internal oscillator block (SCS1:SCS0 = 10). After a Reset, SCS<1:0> are always set to '00'.

**Note:** The instruction to immediately follow the modification of SCS<1:0> will have an instruction time (T<sub>CY</sub>) based on the previous clock source. This should be taken into consideration when developing time dependant code.

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.

The OSTS and IOFS bits indicate the status of the primary oscillator and INTOSC source; these bits are set when their respective oscillators are stable. In particular, OSTS indicates that the Oscillator Start-up Timer has timed out.

### 4.6.2 CLOCK SWITCHING

Clock switching will occur for the following reasons:

- The FCMEN (CONFIG2<0>) bit is set, the device is running from the primary oscillator and the primary oscillator fails. The clock source will be the internal RC oscillator.
- The FCMEN bit is set, the device is running from the T1OSC and T1OSC fails. The clock source will be the internal RC oscillator.
- Following a wake-up due to a Reset or a POR, when the device is configured for Two-Speed Start-up mode, switching will occur between the INTRC and the system clock defined by the FOSC<2:0> bits.
- A wake-up from Sleep occurs due to an interrupt or WDT wake-up and Two-Speed Start-up is enabled. If the primary clock is XT, HS or LP, the clock will switch between the INTRC and the primary system clock after 1024 clocks (OST) and 8 clocks of the primary oscillator. This is conditional upon the SCS bits being set equal to '00'.
- SCS bits are modified from their original value.
- IRCF bits are modified from their original value.

**Note:** Because the SCS bits are cleared on any Reset, no clock switching will occur on a Reset unless the Two-Speed Start-up is enabled and the primary clock is XT, HS or LP. The device will wait for the primary clock to become stable before execution begins (Two-Speed Start-up disabled).

# PIC16F87/88

## 4.6.3 CLOCK TRANSITION AND WDT

When clock switching is performed, the Watchdog Timer is disabled because the Watchdog ripple counter is used as the Oscillator Start-up Timer.

Once the clock transition is complete (i.e., new oscillator selection switch has occurred), the Watchdog counter is re-enabled with the counter reset. This allows the user to synchronize the Watchdog Timer to the start of execution at the new clock frequency.

**Note:** The OST is only used when switching to XT, HS and LP Oscillator modes.

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

U-0	R/W-0	R/W-0	R/W-0	R-0	R/W-0	R/W-0	R/W-0		
bit 7	—	IRCF2	IRCF1	IRCF0	OSTS <sup>(1)</sup>	IOFS	SCS1	SCS0	bit 0

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

bit 6-4 **IRCF<2:0>:** Internal RC Oscillator Frequency Select bits

- 000 = 31.25 kHz
- 001 = 125 kHz
- 010 = 250 kHz
- 011 = 500 kHz
- 100 = 1 MHz
- 101 = 2 MHz
- 110 = 4 MHz
- 111 = 8 MHz

bit 3 **OSTS:** Oscillator Start-up Time-out Status bit<sup>(1)</sup>

- 1 = Device is running from the primary system clock
- 0 = Device is running from T1OSC or INTRC as a secondary system clock

**Note 1:** Bit resets to '0' with Two-Speed Start-up mode and LP, XT or HS selected as the oscillator mode.

bit 2 **IOFS:** INTOSC Frequency Stable bit

- 1 = Frequency is stable
- 0 = Frequency is not stable

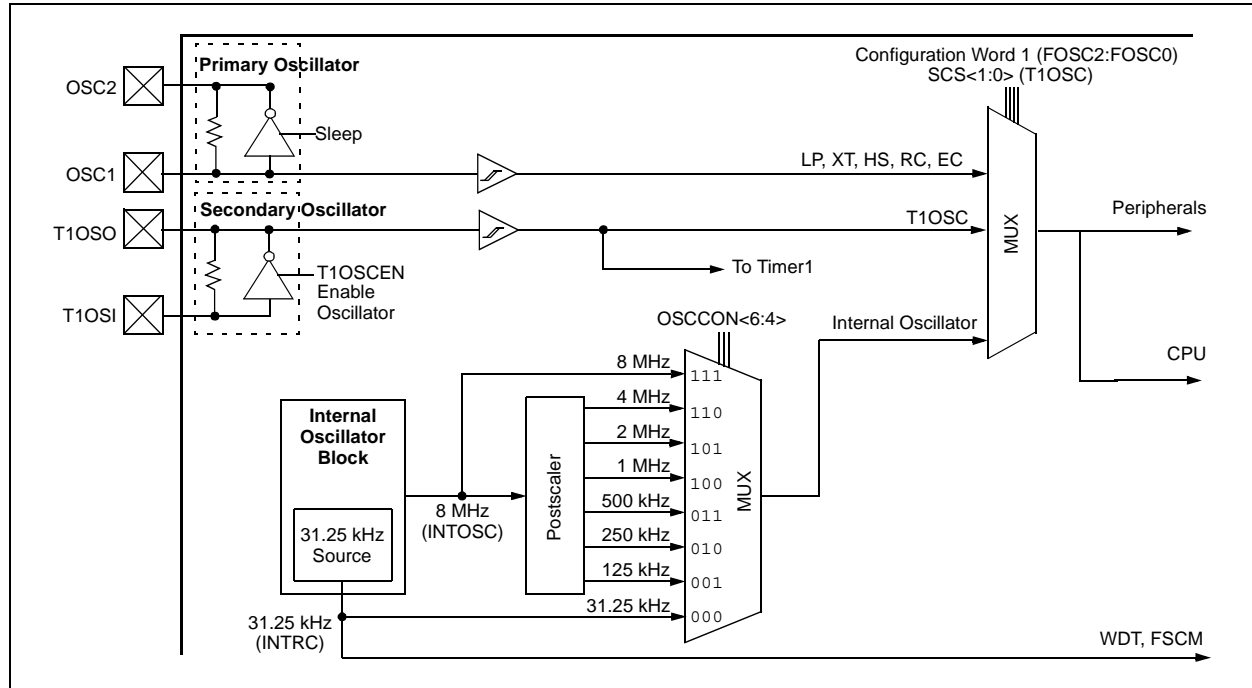
bit 1-0 **SCS<1:0>:** Oscillator Mode Select bits

- 00 = Oscillator mode defined by FOSC<2:0>
- 01 = T1OSC is used for system clock
- 10 = Internal RC is used for system clock
- 11 = 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

**FIGURE 4-6: PIC16F87/88 CLOCK DIAGRAM**



#### 4.6.4 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).

#### 4.6.5 CLOCK TRANSITION SEQUENCE

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 CLK0 is held low.
  3. The clock switching circuitry then waits for eight falling edges of requested clock, after which it switches CLK0 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 CLK0 is held low.
  3. The clock switching circuitry then waits for eight falling edges of requested clock, after which it switches CLK0 to this new clock source.
  4. Oscillator switchover is complete.

# PIC16F87/88

- 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.

## 4.6.6 OSCILLATOR DELAY UPON POWER-UP, WAKE-UP AND CLOCK SWITCHING

Table 4-3 shows the different delays invoked for various clock switching sequences. It also shows the delays invoked for POR and wake-up.

**TABLE 4-3: OSCILLATOR DELAY EXAMPLES**

Clock Switch		Frequency	Oscillator Delay	Comments
From	To			
Sleep/POR	INTRC T1OSC	31.25 kHz 32.768 kHz	CPU Start-up <sup>(1)</sup>	Following a wake-up from Sleep mode or POR, CPU start-up is invoked to allow the CPU to become ready for code execution.
	INTOSC/ INTOSC Postscaler	125 kHz-8 MHz	4 ms (approx.) and CPU Start-up <sup>(1)</sup>	
INTRC/Sleep	EC, RC	DC – 20 MHz		
INTRC (31.25 kHz)	EC, RC	DC – 20 MHz		
Sleep	LP, XT, HS	32.768 kHz-20 MHz	1024 Clock Cycles (OST)	Following a change from INTRC, an OST of 1024 cycles must occur.
INTRC (31.25 kHz)	INTOSC/ INTOSC Postscaler	125 kHz-8 MHz	4 ms (approx.)	Refer to <b>Section 4.6.4 “Modifying the IRCF Bits”</b> for further details.

**Note 1:** The 5-10 μs start-up delay is based on a 1 MHz system clock.

## 4.7 Power-Managed Modes

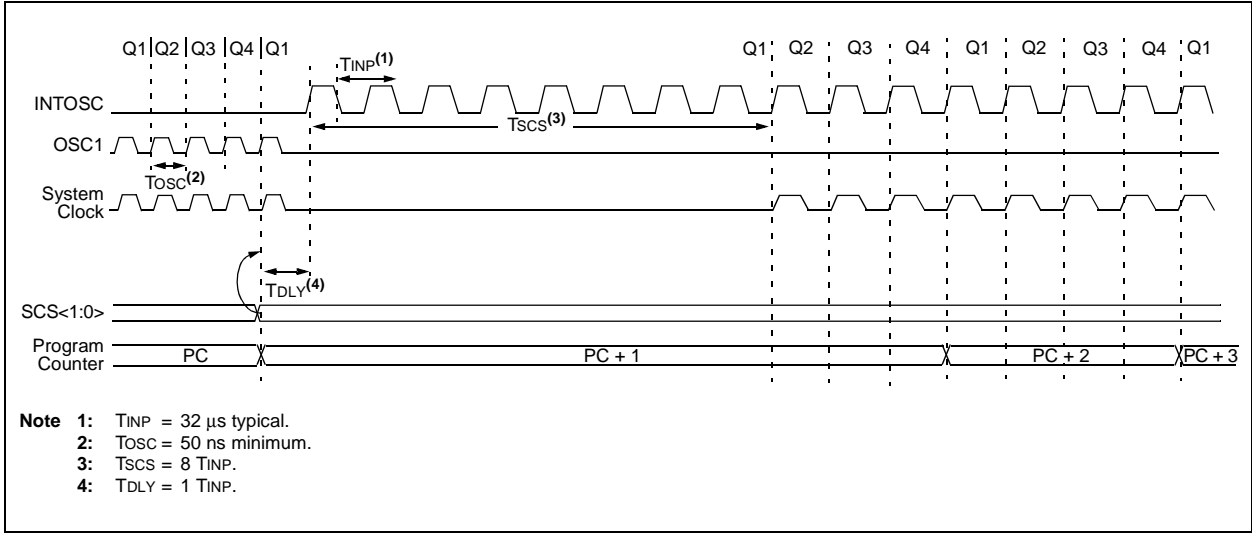
### 4.7.1 RC\_RUN MODE

When SCS bits are configured to run from the INTRC, a clock transition is generated if the system clock is not already using the INTRC. The event will clear the OSTS bit, switch the system clock from the primary system clock (if SCS<1:0> = 00) determined by the value contained in the configuration bits, or from the T1OSC (if SCS<1:0> = 01) to the INTRC clock option and shut down the primary system clock to conserve power. Clock switching will not occur if the primary system clock is already configured as INTRC.

If the system clock does not come from the INTRC (31.25 kHz) when the SCS bits are changed and the IRCF bits in the OSCCON register are configured for a frequency other than INTRC, the frequency may not be stable immediately. The IOFS bit (OSCCON<2>) will be set when the INTOSC or postscaler frequency is stable, after 4 ms (approx.).

After a clock switch has been executed, the OSTS bit is cleared, indicating a low-power mode and the device does not run from the primary system clock. The internal Q clocks are held in the Q1 state until eight falling edge clocks are counted on the INTRC oscillator. After the eight clock periods have transpired, the clock input to the Q clocks is released and operation resumes (see Figure 4-7).

**FIGURE 4-7: TIMING DIAGRAM FOR XT, HS, LP, EC AND EXTRC TO RC\_RUN MODE**



# PIC16F87/88

## 4.7.2 SEC\_RUN MODE

The core and peripherals can be configured to be clocked by T1OSC using a 32.768 kHz crystal. The crystal must be connected to the T1OSO and T1OSI pins. This is the same configuration as the low-power timer circuit (see **Section 7.6 “Timer1 Oscillator”**). When SCS bits are configured to run from T1OSC, a clock transition is generated. It will clear the OSTS bit, switch the system clock from either the primary system clock or INTRC, depending on the value of SCS<1:0> and FOSC<2:0>, to the external low-power Timer1 oscillator input (T1OSC) and shut down the primary system clock to conserve power.

After a clock switch has been executed, the internal Q clocks are held in the Q1 state until eight falling edge clocks are counted on the T1OSC. After the eight clock periods have transpired, the clock input to the Q clocks is released and operation resumes (see Figure 4-8). In addition, T1RUN (In T1CON) is set to indicate that T1OSC is being used as the system clock.

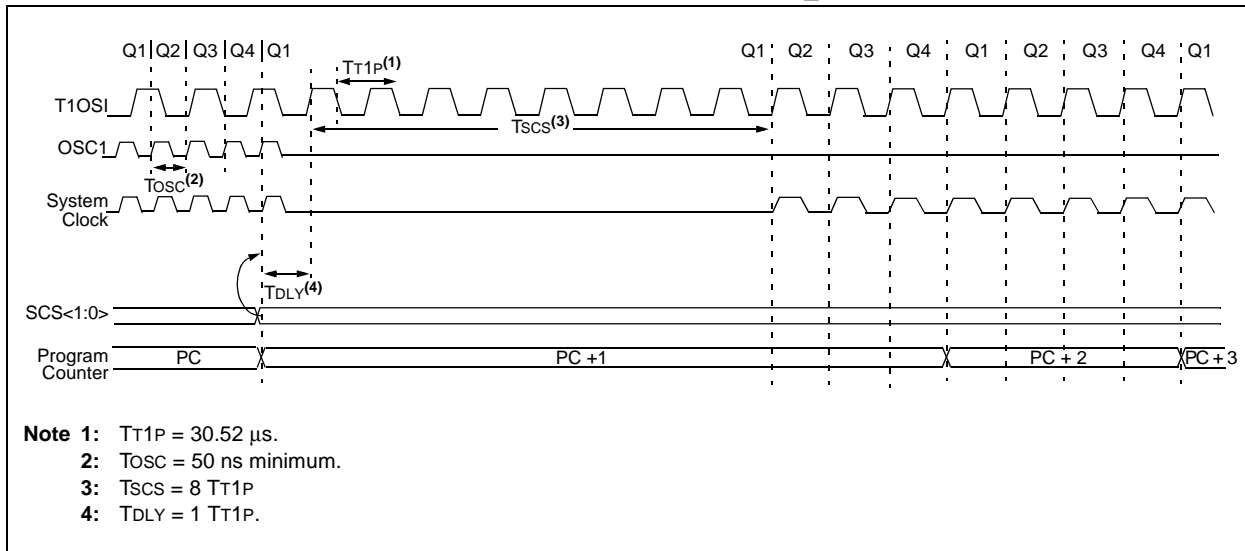
**Note 1:** The T1OSCEN bit must be enabled and it is the user’s responsibility to ensure T1OSC is stable before clock switching to the T1OSC input clock can occur.

**2:** When T1OSCEN = 0, the following possible effects result.

Original SCS<1:0>	Modified SCS<1:0>	Final SCS<1:0>
00	01	00 – no change
00	11	10 – INTRC
10	11	10 – no change
10	01	00 – Oscillator defined by FOSC<2:0>

A clock switching event will occur if the final state of the SCS bits is different from the original.

**FIGURE 4-8: TIMING DIAGRAM FOR SWITCHING TO SEC\_RUN MODE**



## 4.7.3 SEC\_RUN/RC\_RUN TO PRIMARY CLOCK SOURCE

When switching from a SEC\_RUN or RC\_RUN mode back to the primary system clock, following a change of SCS<1:0> to '00', the sequence of events that takes place will depend upon the value of the FOSC bits in the Configuration register. If the primary clock source is configured as a crystal (HS, XT or LP), then the transition will take place after 1024 clock cycles. This is necessary because the crystal oscillator has been powered down until the time of the transition. In order to provide the system with a reliable clock when the changeover has occurred, the clock will not be released to the changeover circuit until the 1024 count has expired.

During the oscillator start-up time, the system clock comes from the current system clock. Instruction execution and/or peripheral operation continues using the currently selected oscillator as the CPU clock source, until the necessary clock count has expired, to ensure that the primary system clock is stable.

To know when the OST has expired, the OSTS bit should be monitored. OSTS = 1 indicates that the Oscillator Start-up Timer has timed out and the system clock comes from the primary clock source.

Following the oscillator start-up time, the internal Q clocks are held in the Q1 state until eight falling edge clocks are counted from the primary system clock. The clock input to the Q clocks is then released and operation resumes with the primary system clock determined by the FOSC bits (see Figure 4-10).

When in SEC\_RUN mode, the act of clearing the T1OSCEN bit in the T1CON register will cause SCS<0> to be cleared, which causes the SCS<1:0> bits to revert to '00' or '10' depending on what SCS<1> is. Although the T1OSCEN bit was cleared, T1OSC will be enabled and instruction execution will continue until the OST time-out for the main system clock is complete. At that time, the system clock will switch from the T1OSC to the primary clock or the INTRC. Following this, the T1 oscillator will be shut down.

**Note:** If the primary system clock is either RC or EC, an internal delay timer (5-10  $\mu$ s) will suspend operation after exiting Secondary Clock mode to allow the CPU to become ready for code execution.

### 4.7.3.1 Returning to Primary Clock Source Sequence

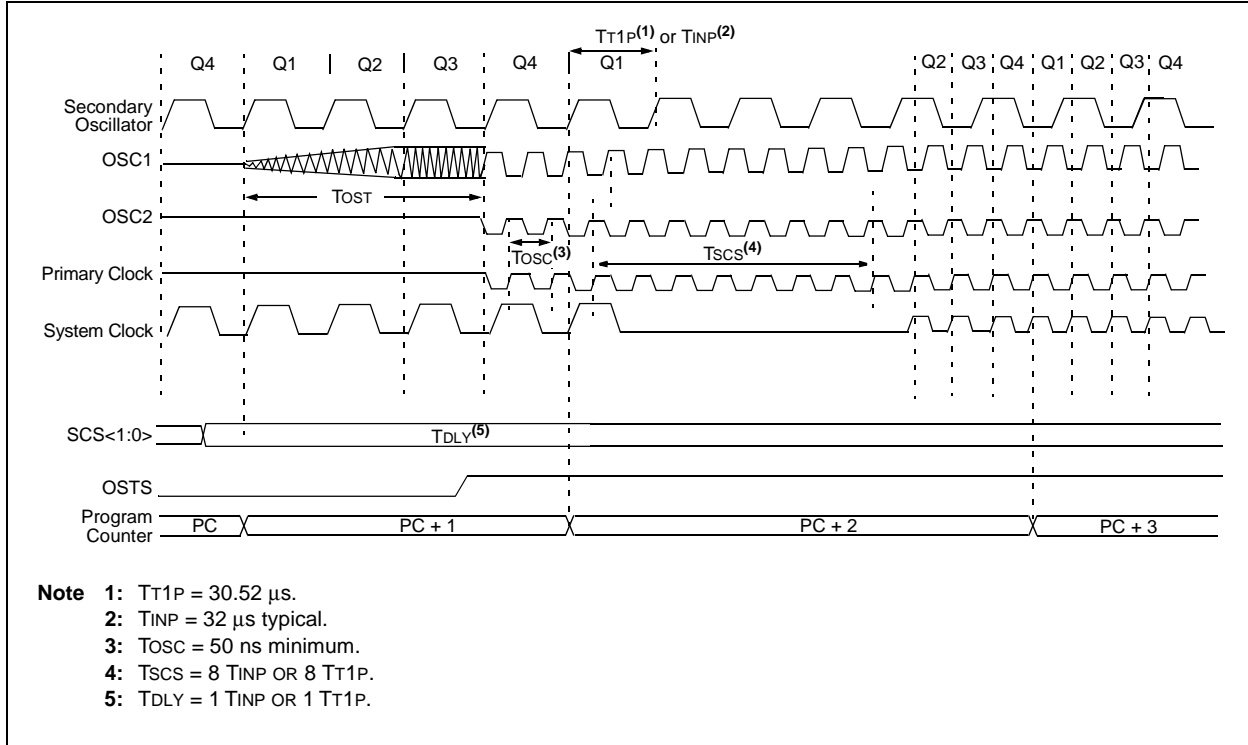
Changing back to the primary oscillator from SEC\_RUN or RC\_RUN can be accomplished by either changing SCS<1:0> to '00', or clearing the T1OSCEN bit in the T1CON register (if T1OSC was the secondary clock).

The sequence of events that follows is the same for both modes:

1. If the primary system clock is configured as EC, RC or INTRC, then the OST time-out is skipped. Skip to step 3.
2. If the primary system clock is configured as an external oscillator (HS, XT, LP), then the OST will be active, waiting for 1024 clocks of the primary system clock.
3. On the following Q1, the device holds the system clock in Q1.
4. The device stays in Q1 while eight falling edges of the primary system clock are counted.
5. Once the eight counts transpire, the device begins to run from the primary oscillator.
6. If the secondary clock was INTRC and the primary is not INTRC, the INTRC will be shut down to save current providing that the INTRC is not being used for any other function, such as WDT or Fail-Safe Clock monitoring.
7. If the secondary clock was T1OSC, the T1OSC will continue to run if T1OSCEN is still set; otherwise, the T1 oscillator will be shut down.

# PIC16F87/88

**FIGURE 4-9: TIMING FOR TRANSITION BETWEEN SEC\_RUN/RC\_RUN AND PRIMARY CLOCK**





## 4.7.3.2 Returning to Primary Oscillator with a Reset

A Reset will clear SCS<1:0> back to '00'. The sequence for starting the primary oscillator following a Reset is the same for all forms of Reset, including POR. There is no transition sequence from the alternate system clock to the primary system clock on a Reset condition. Instead, the device will reset the state of the OSCCON register and default to the primary system clock. The sequence of events that takes place after this will depend upon the value of the FOSC bits in the Configuration register. If the external oscillator is configured as a crystal (HS, XT or LP), the CPU will be held in the Q1 state until 1024 clock cycles have transpired on the primary clock. This is necessary because the crystal oscillator has been powered down until the time of the transition.

During the oscillator start-up time, instruction execution and/or peripheral operation is suspended.

**Note:** If Two-Speed Clock Start-up mode is enabled, the INTRC will act as the system clock until the OST timer has timed out.

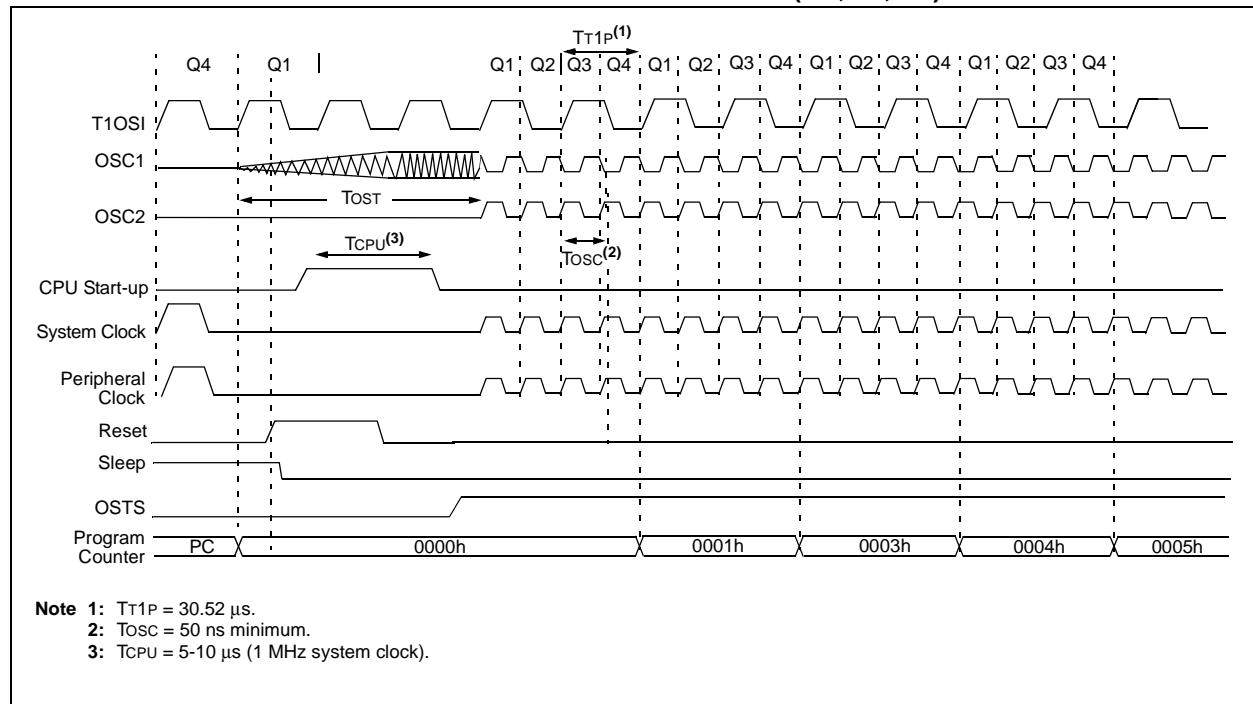
If the primary system clock is either RC, EC or INTRC, the CPU will begin operating on the first Q1 cycle following the wake-up event. This means that there is

no oscillator start-up time required because the primary clock is already stable; however, there is a delay between the wake-up event and the following Q2. An internal delay timer of 5-10  $\mu$ s will suspend operation after the Reset to allow the CPU to become ready for code execution. The CPU and peripheral clock will be held in the first Q1.

The sequence of events is as follows:

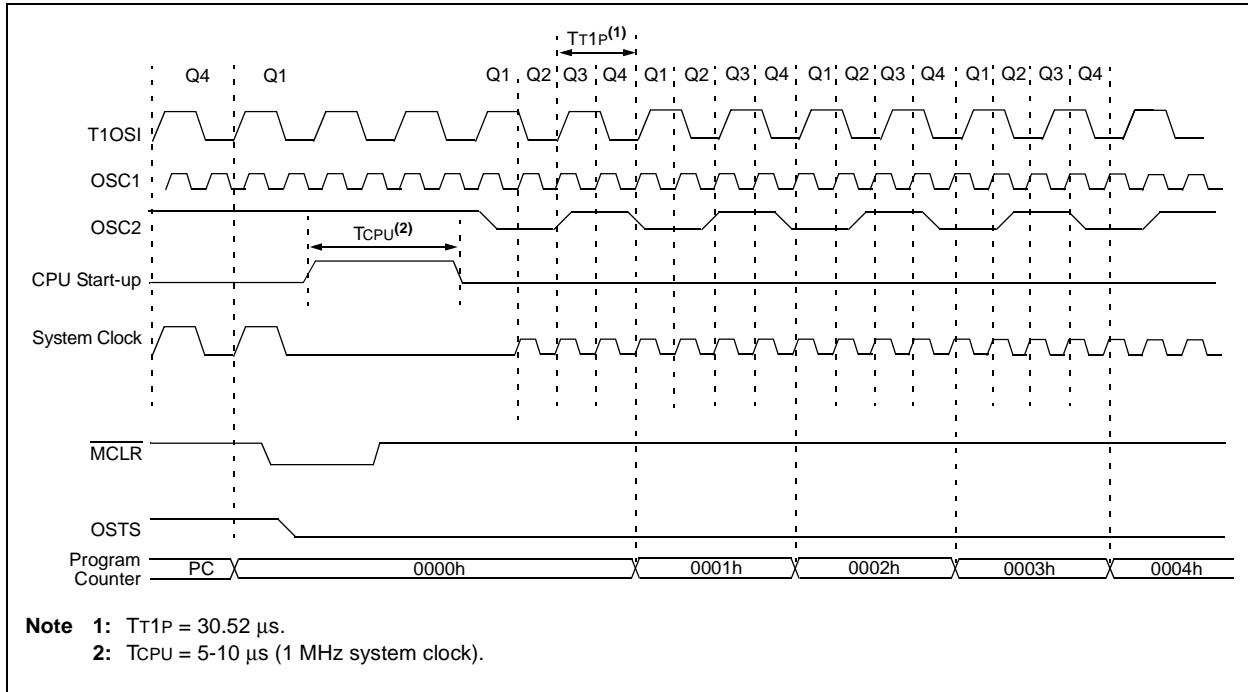
1. A device Reset is asserted from one of many sources (WDT, BOR, MCLR, etc.).
2. The device resets and the CPU start-up timer is enabled if in Sleep mode. The device is held in Reset until the CPU start-up time-out is complete.
3. If the primary system clock is configured as an external oscillator (HS, XT, LP), then the OST will be active waiting for 1024 clocks of the primary system clock. While waiting for the OST, the device will be held in Reset. The OST and CPU start-up timers run in parallel.
4. After both the CPU start-up and OST timers have timed out, the device will wait for one additional clock cycle and instruction execution will begin.

**FIGURE 4-10: PRIMARY SYSTEM CLOCK AFTER RESET (HS, XT, LP)**



# PIC16F87/88

**FIGURE 4-11: PRIMARY SYSTEM CLOCK AFTER RESET (EC, RC, INTRC)**



**TABLE 4-4: CLOCK SWITCHING MODES**

Current System Clock	SCS Bits <1:0> Modified to:	Delay	OSTS Bit	IOFS Bit	T1RUN Bit	New System Clock	Comments
LP, XT, HS, T1OSC, EC, RC	10 (INTRC) FOSC<2:0> = LP, XT or HS	8 Clocks of INTRC	0	1 <sup>(1)</sup>	0	INTRC or INTOSC or INTOSC Postscaler	The internal RC oscillator frequency is dependant upon the IRCF bits.
LP, XT, HS, INTRC, EC, RC	01 (T1OSC) FOSC<2:0> = LP, XT or HS	8 Clocks of T1OSC	0	N/A	1	T1OSC	T1OSCEN bit must be enabled.
INTRC T1OSC	00 FOSC<2:0> = EC or FOSC<2:0> = RC	8 Clocks of EC or RC	1	N/A	0	EC or RC	
INTRC T1OSC	00 FOSC<2:0> = LP, XT, HS	1024 Clocks (OST) + 8 Clocks of LP, XT, HS	1	N/A	0	LP, XT, HS	During the 1024 clocks, program execution is clocked from the secondary oscillator until the primary oscillator becomes stable.
LP, XT, HS	00 (Due to Reset) LP, XT, HS	1024 Clocks (OST)	1	N/A	0	LP, XT, HS	When a Reset occurs, there is no clock transition sequence. Instruction execution and/or peripheral operation is suspended unless Two-Speed Start-up mode is enabled, after which the INTRC will act as the system clock until the OST timer has expired.

**Note 1:** If the new clock source is the INTOSC or INTOSC postscaler, then the IOFS bit will be set 4 ms (approx.) after the clock change.

# PIC16F87/88

---

## 4.7.4 EXITING SLEEP WITH AN INTERRUPT

Any interrupt, such as WDT or INT0, will cause the part to leave the Sleep mode.

The SCS bits are unaffected by a `SLEEP` command and are the same before and after entering and leaving Sleep. The clock source used after an exit from Sleep is determined by the SCS bits.

### 4.7.4.1 Sequence of Events

#### If `SCS<1:0> = 00`:

1. The device is held in Sleep until the CPU start-up time-out is complete.
2. If the primary system clock is configured as an external oscillator (HS, XT, LP), then the OST will be active waiting for 1024 clocks of the primary system clock. While waiting for the OST, the device will be held in Sleep unless Two-Speed Start-up is enabled. The OST and CPU start-up timers run in parallel. Refer to **Section 15.12.3 “Two-Speed Clock Start-up Mode”** for details on Two-Speed Start-up.
3. After both the CPU start-up and OST timers have timed out, the device will exit Sleep and begin instruction execution with the primary clock defined by the FOSC bits.

#### If `SCS<1:0> = 01` or `10`:

1. The device is held in Sleep until the CPU start-up time-out is complete.
2. After the CPU start-up timer has timed out, the device will exit Sleep and begin instruction execution with the selected oscillator mode.

**Note:** If a user changes `SCS<1:0>` just before entering Sleep mode, the system clock used when exiting Sleep mode could be different than the system clock used when entering Sleep mode.

As an example, if `SCS<1:0> = 01` and T1OSC is the system clock and the following instructions are executed:

```
BCF      OSCCON, SCS0
SLEEP
```

then a clock change event is executed. If the primary oscillator is XT, LP or HS, the core will continue to run off T1OSC and execute the `SLEEP` command.

When Sleep is exited, the part will resume operation with the primary oscillator after the OST has expired.

## 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. On PIC16F88 devices, it is also multiplexed with an analog input to become the RA4/AN4/T0CKI/C2OUT pin. The RA4/AN4/T0CKI/C2OUT 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 comparator outputs. On PIC16F88 devices, pins RA<3:2> are also multiplexed with the 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
CLRWF  PORTA      ; Initialize PORTA by
                  ; clearing output
                  ; data latches
BANKSEL ANSEL      ; Select Bank of ANSEL
MOVLW  0x00      ; Configure all pins
MOVWF  ANSEL      ; 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/CVREF/VREF-( <sup>2</sup> )	bit 2	TTL	Input/output, analog input, VREF- or comparator VREF output.
RA3/AN3/VREF+( <sup>2</sup> )/C1OUT	bit 3	TTL	Input/output, analog input, VREF+ or comparator output.
RA4/AN4( <sup>2</sup> )/T0CKI/C2OUT	bit 4	ST	Input/output, analog input, TMR0 external input or comparator output.
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( <sup>1</sup> )	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.

**2:** PIC16F88 only.

# PIC16F87/88

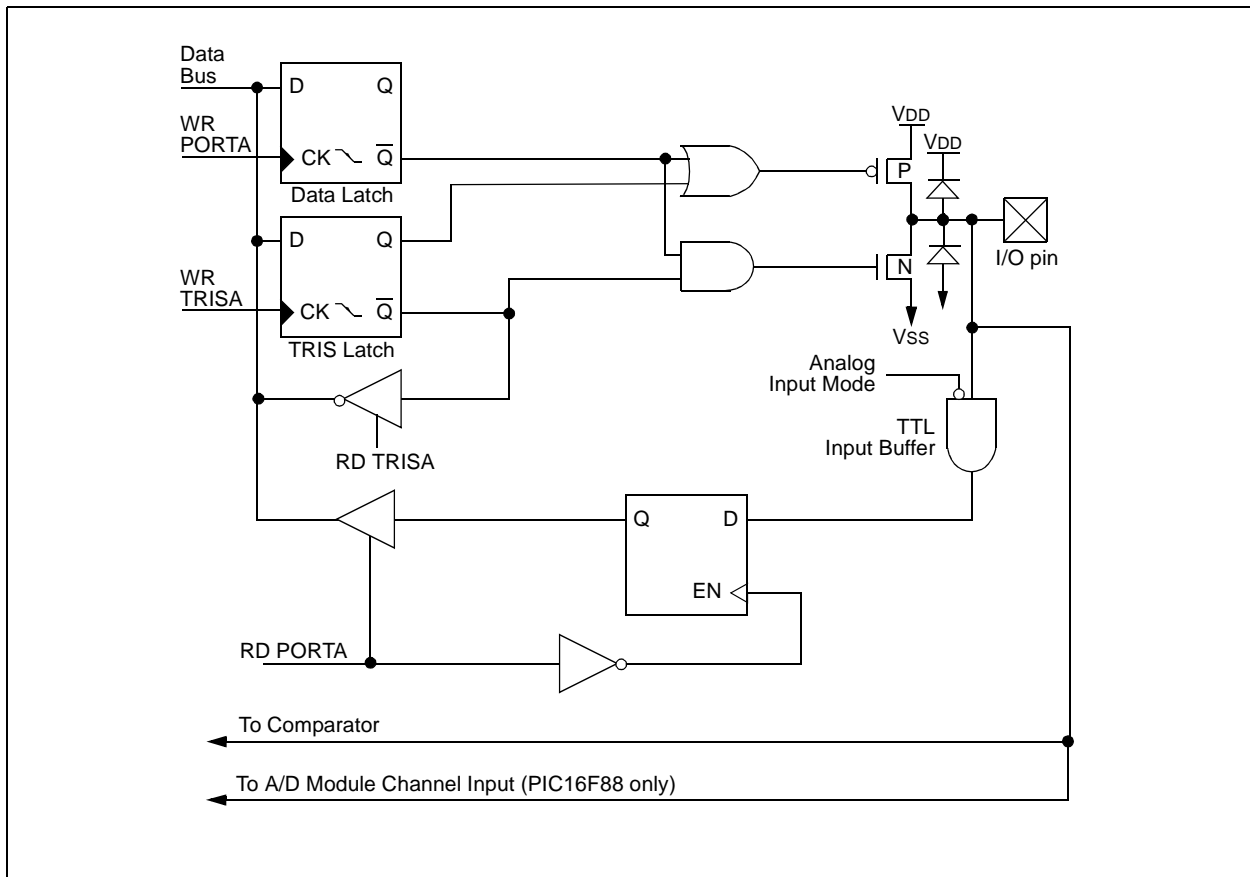
**TABLE 5-2: SUMMARY OF REGISTERS ASSOCIATED WITH PORTA**

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	xxxx 0000 <sup>(1)</sup> xxx0 0000 <sup>(2)</sup>	uuuu 0000 <sup>(1)</sup> uuu0 0000 <sup>(2)</sup>
85h	TRISA	TRISA7	TRISA6	TRISA5 <sup>(3)</sup>	PORTA Data Direction Register					1111 1111	1111 1111
9Fh	ADCON1	ADFM	ADCS2	VCFG1	VCFG0	—	—	—	—	0000 ----	0000 ----
9Bh	ANSEL <sup>(4)</sup>	—	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0	-111 1111	-111 1111

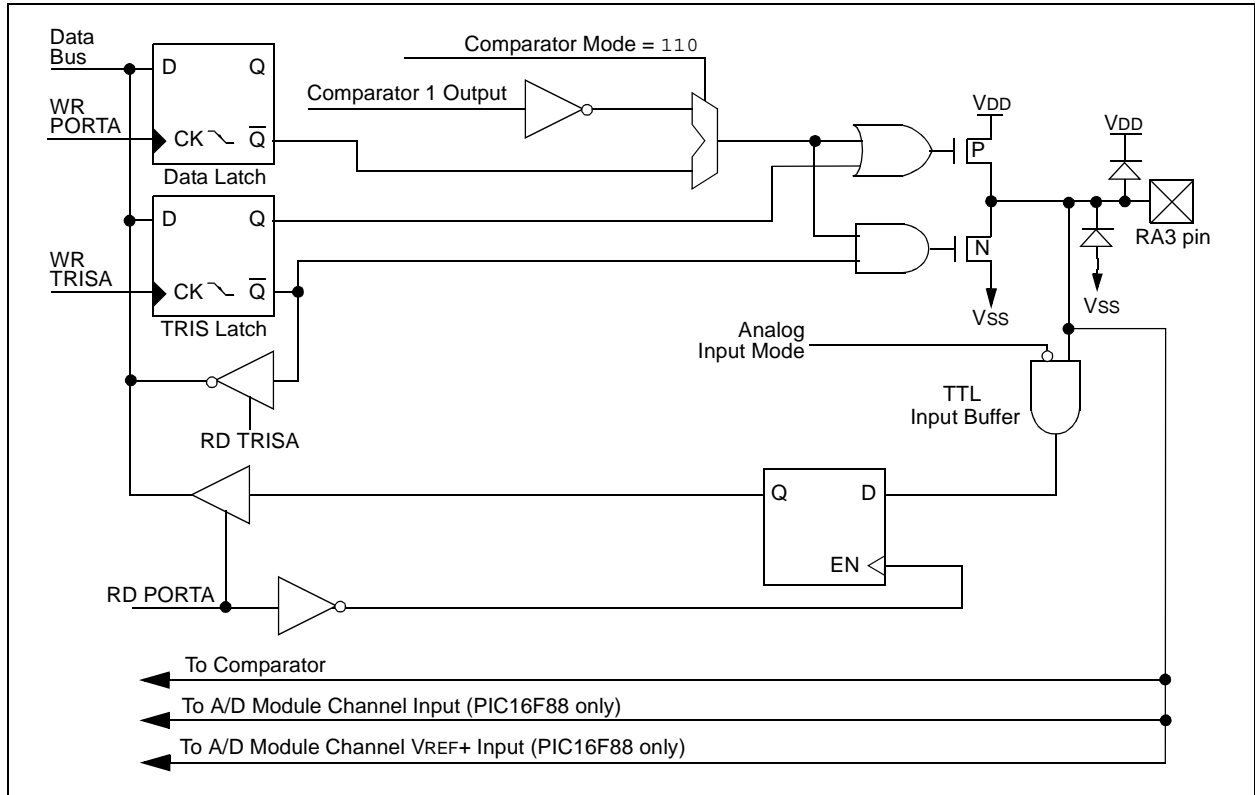
**Legend:** x = unknown, u = unchanged, - = unimplemented locations read as '0'. Shaded cells are not used by PORTA.

- Note 1:** This value applies only to the PIC16F87.  
**Note 2:** This value applies only to the PIC16F88.  
**Note 3:** Pin 5 is an input only; the state of the TRISA5 bit has no effect and will always read '1'.  
**Note 4:** PIC16F88 device only.

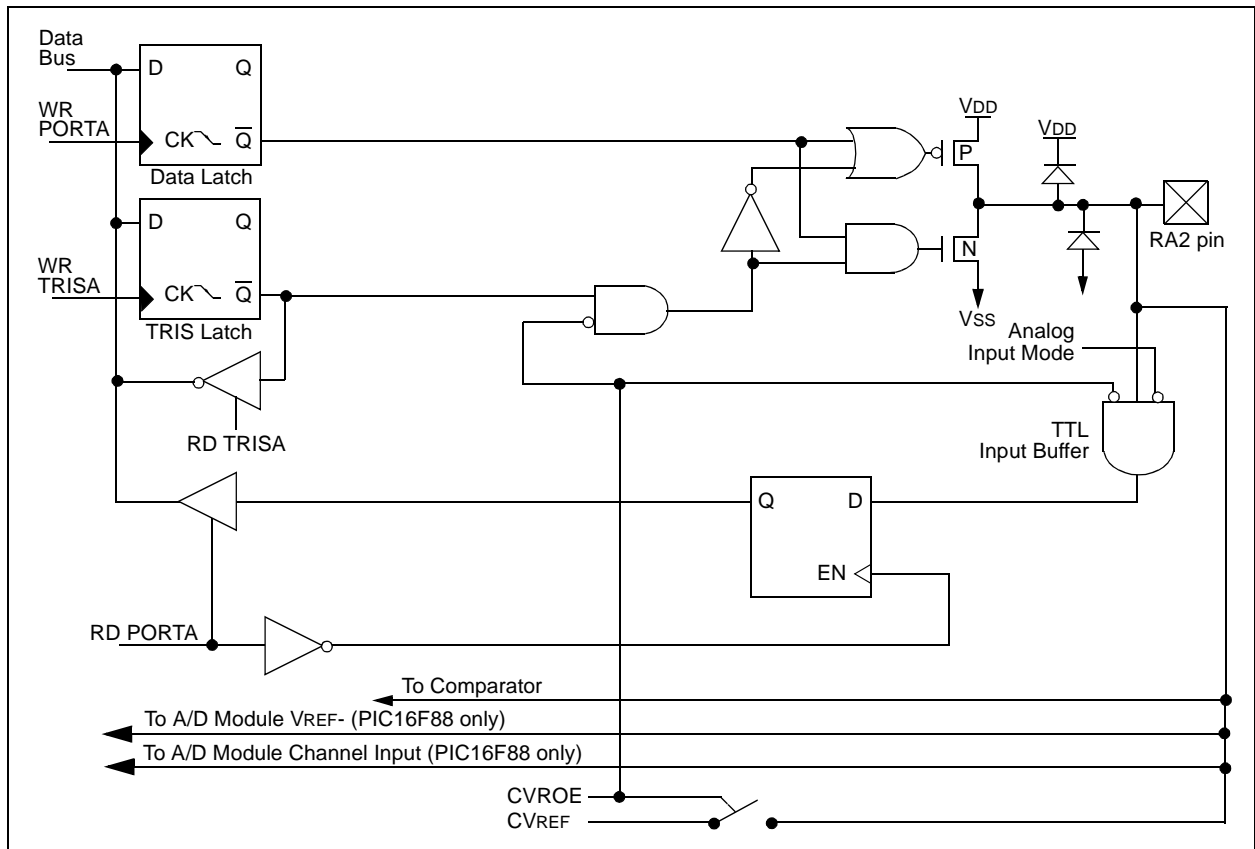
**FIGURE 5-1: BLOCK DIAGRAM OF RA0/AN0:RA1/AN1 PINS**



**FIGURE 5-2: BLOCK DIAGRAM OF RA3/AN3/VREF+/C1OUT PIN**

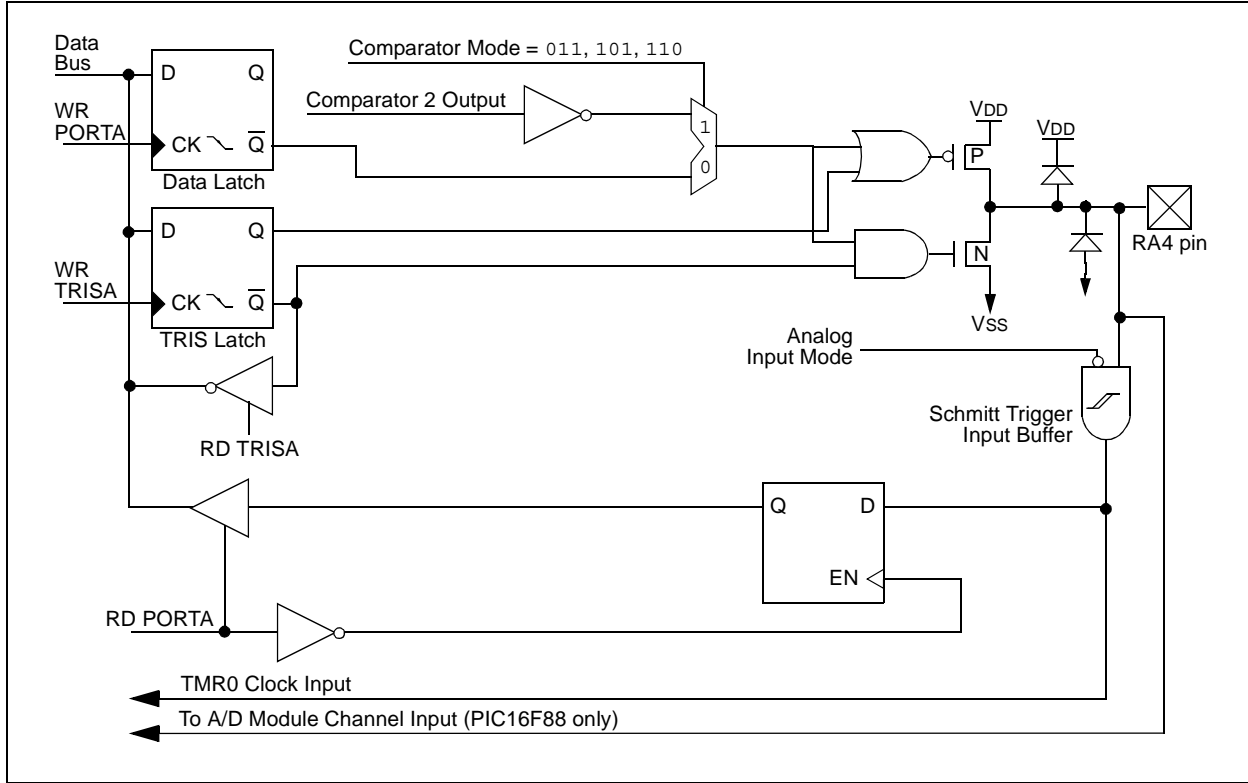


**FIGURE 5-3: BLOCK DIAGRAM OF RA2/AN2/CVREF/VREF- PIN**



# PIC16F87/88

**FIGURE 5-4: BLOCK DIAGRAM OF RA4/AN4/T0CKI/C2OUT PIN**



**FIGURE 5-5: BLOCK DIAGRAM OF RA5/MCLR/VPP PIN**

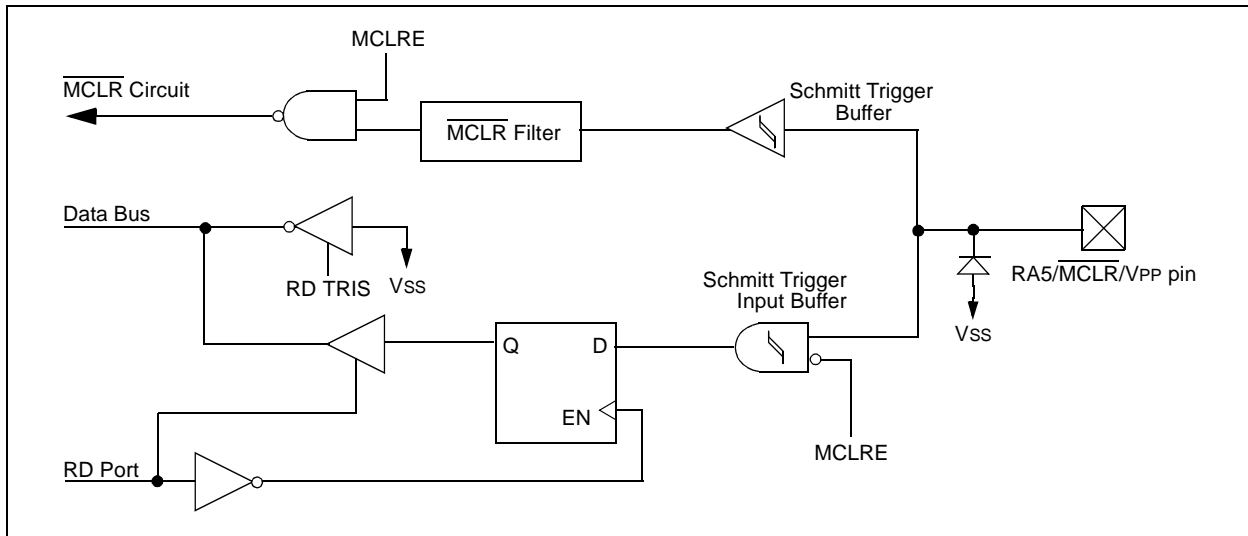
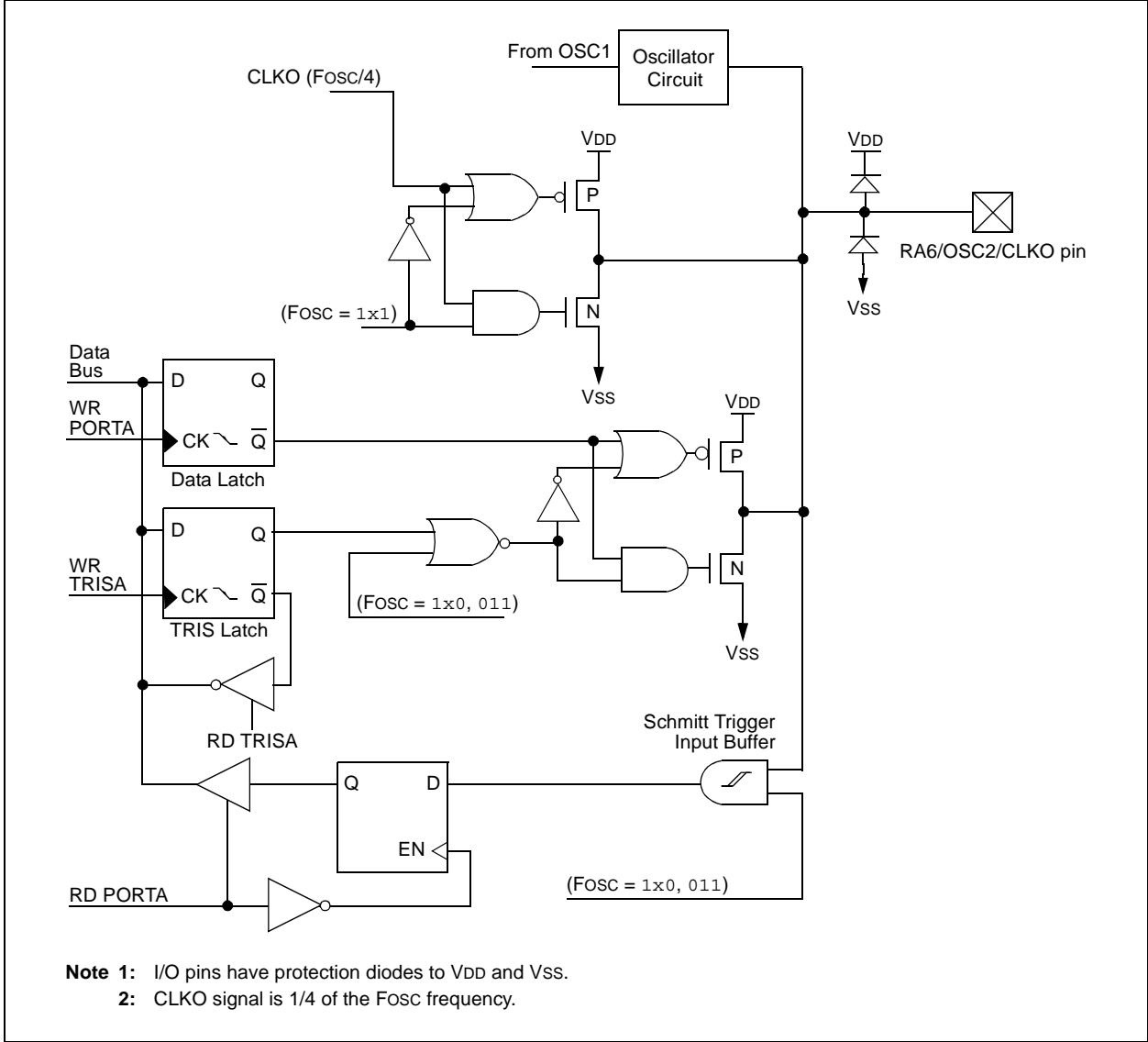


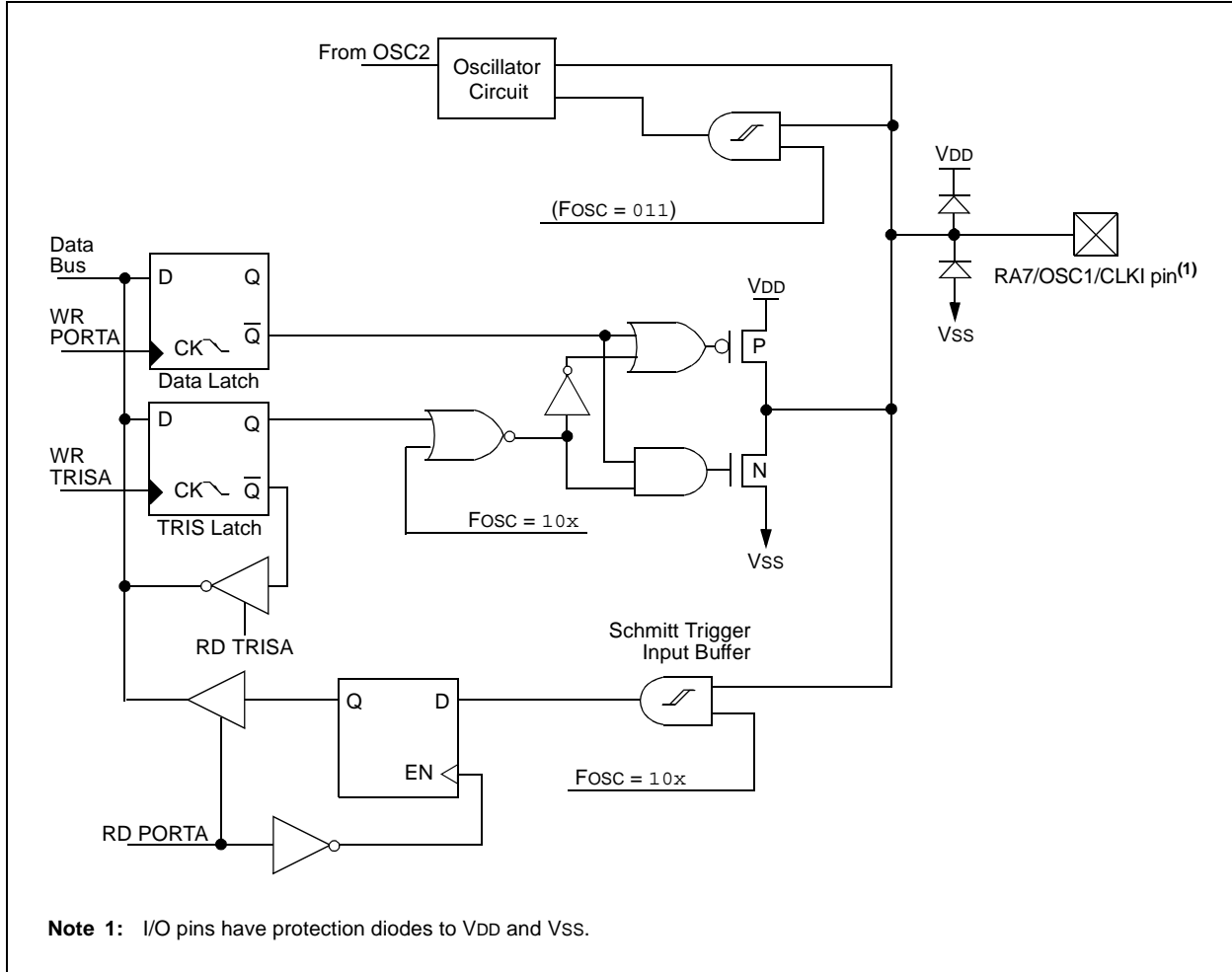


FIGURE 5-6: BLOCK DIAGRAM OF RA6/OSC2/CLKO PIN



# PIC16F87/88

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



## 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 OR'ed together to generate the RB Port Change Interrupt with Flag bit RBIF (INTCON<0>).

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

- 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.

# PIC16F87/88

**TABLE 5-3: PORTB FUNCTIONS**

Name	Bit#	Buffer	Function
RB0/INT/CCP1 <sup>(7)</sup>	bit 0	TTL/ST <sup>(1)</sup>	Input/output pin or external interrupt input. Capture input/Compare output/PWM output pin. Internal software programmable weak pull-up.
RB1/SDI/SDA	bit 1	TTL/ST <sup>(5)</sup>	Input/output pin, SPI™ data input pin or I <sup>2</sup> C™ data I/O pin. Internal software programmable weak pull-up.
RB2/SDO/RX/DT	bit 2	TTL/ST <sup>(4)</sup>	Input/output pin, SPI data output pin. AUSART asynchronous receive or synchronous data. Internal software programmable weak pull-up.
RB3/PGM/CCP1 <sup>(3,7)</sup>	bit 3	TTL/ST <sup>(2)</sup>	Input/output pin, programming in LVP mode or Capture input/Compare output/PWM output pin. Internal software programmable weak pull-up.
RB4/SCK/SCL	bit 4	TTL/ST <sup>(5)</sup>	Input/output pin or SPI and I <sup>2</sup> C clock pin (with interrupt-on-change). Internal software programmable weak pull-up.
RB5/SS/TX/CK	bit 5	TTL	Input/output pin or SPI slave select pin (with interrupt-on-change). AUSART asynchronous transmit or synchronous clock. Internal software programmable weak pull-up.
RB6/AN5 <sup>(6)</sup> /PGC/ T1OSO/T1CKI	bit 6	TTL/ST <sup>(2)</sup>	Input/output pin, analog input <sup>(6)</sup> , serial programming clock (with interrupt-on-change), Timer1 oscillator output pin or Timer1 clock input pin. Internal software programmable weak pull-up.
RB7/AN6 <sup>(6)</sup> /PGD/ T1OSI	bit 7	TTL/ST <sup>(2)</sup>	Input/output pin, analog input <sup>(6)</sup> , serial programming data (with interrupt-on-change) or Timer1 oscillator input pin. 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<sup>2</sup>C mode.

**6:** PIC16F88 only.

**7:** The CCP1 pin is determined by the CCPMX bit in Configuration Word 1 register.

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

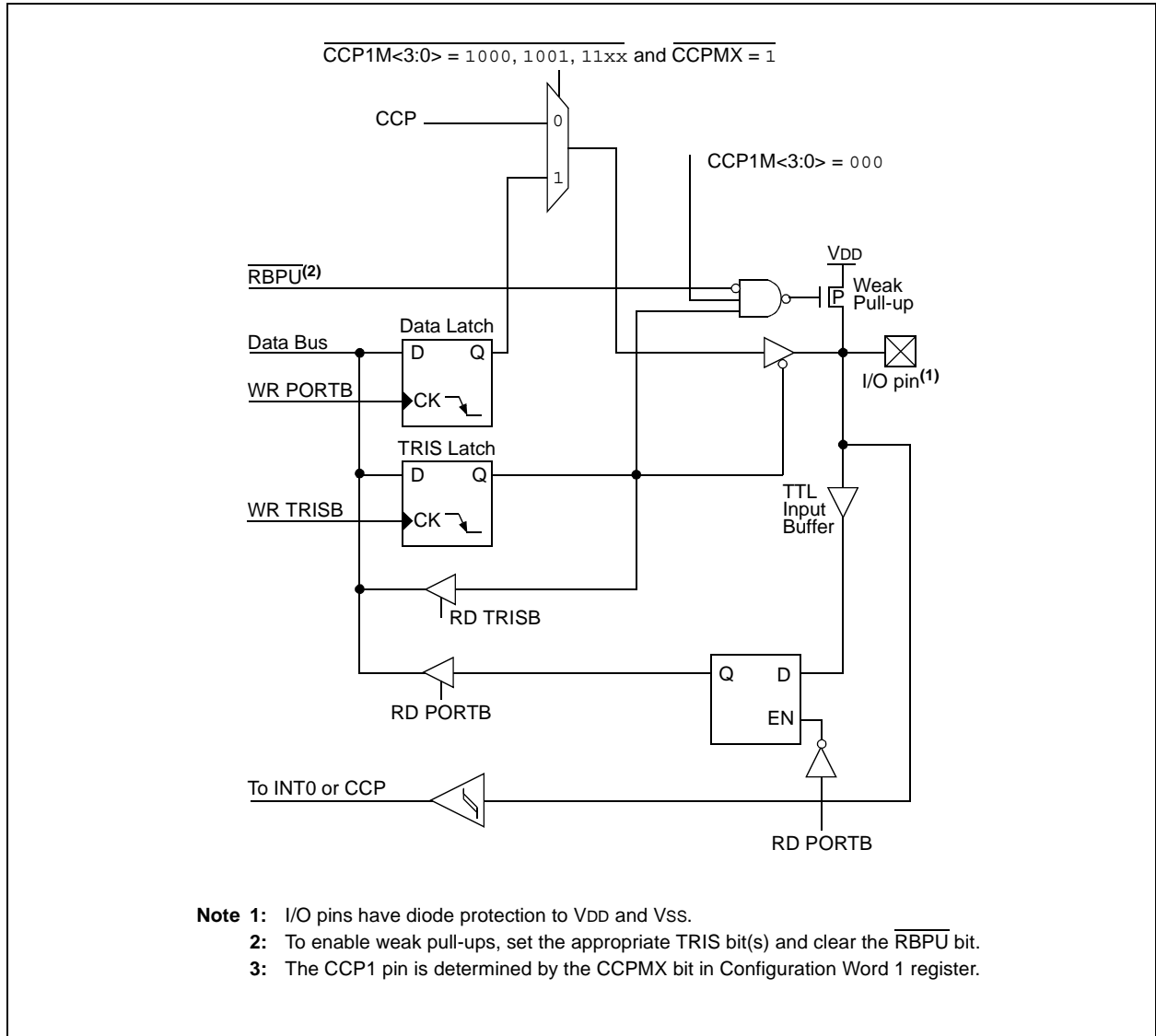
Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
06h, 106h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx xxxx <sup>(1)</sup> 00xx xxxx <sup>(2)</sup>	uuuu uuuu <sup>(1)</sup> 00uu uuuu <sup>(2)</sup>
86h, 186h	TRISB	PORTB Data Direction Register								1111 1111	1111 1111
81h, 181h	OPTION_REG	RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
9Bh	ANSEL <sup>(2)</sup>	—	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0	-111 1111	-111 1111

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

**Note 1:** This value applies only to the PIC16F87.

**2:** This value applies only to the PIC16F88.

**FIGURE 5-8: BLOCK DIAGRAM OF RB0/INT/CCP1<sup>(3)</sup> PIN**



# PIC16F87/88

FIGURE 5-9: BLOCK DIAGRAM OF RB1/SDI/SDA PIN

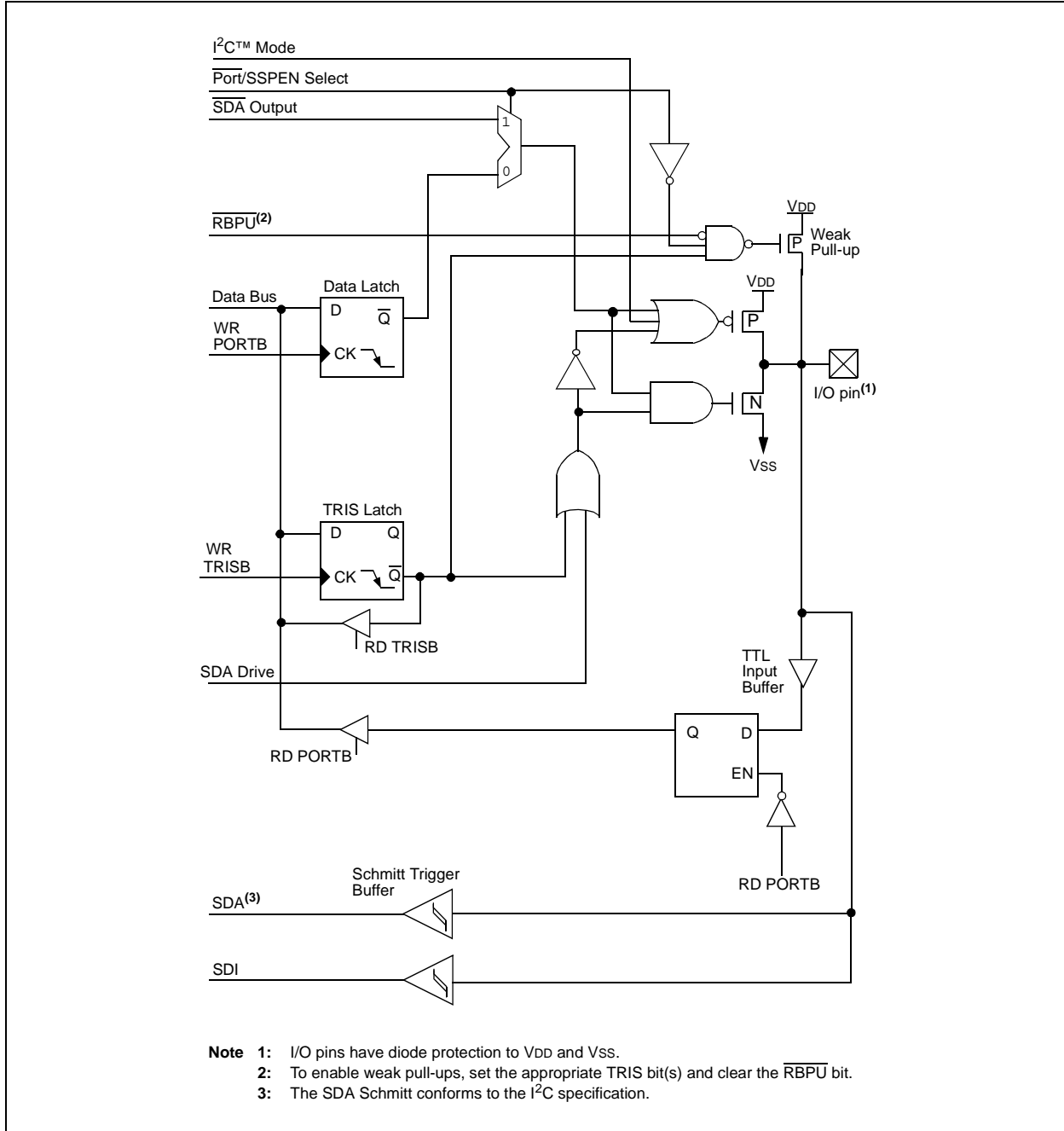
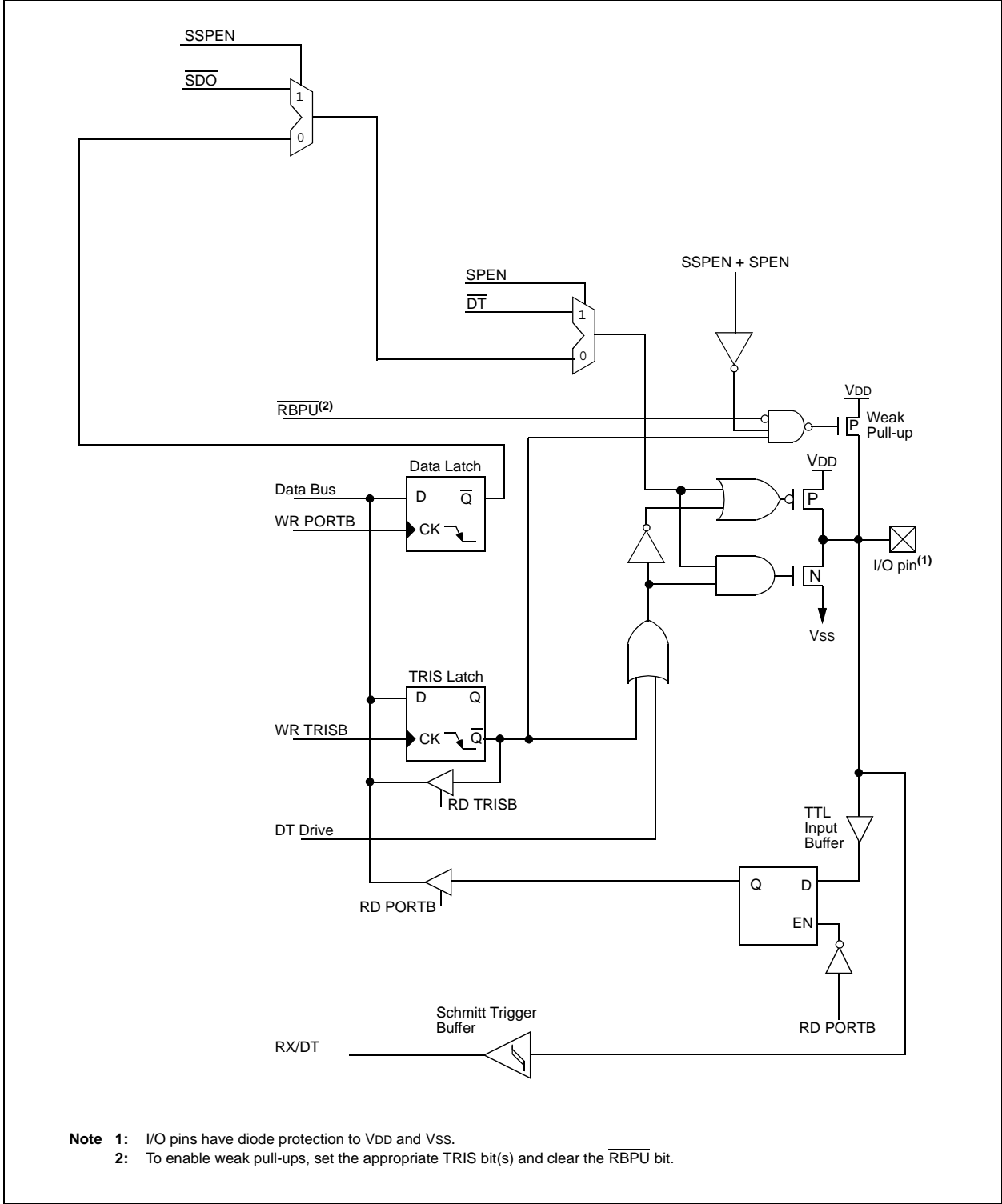


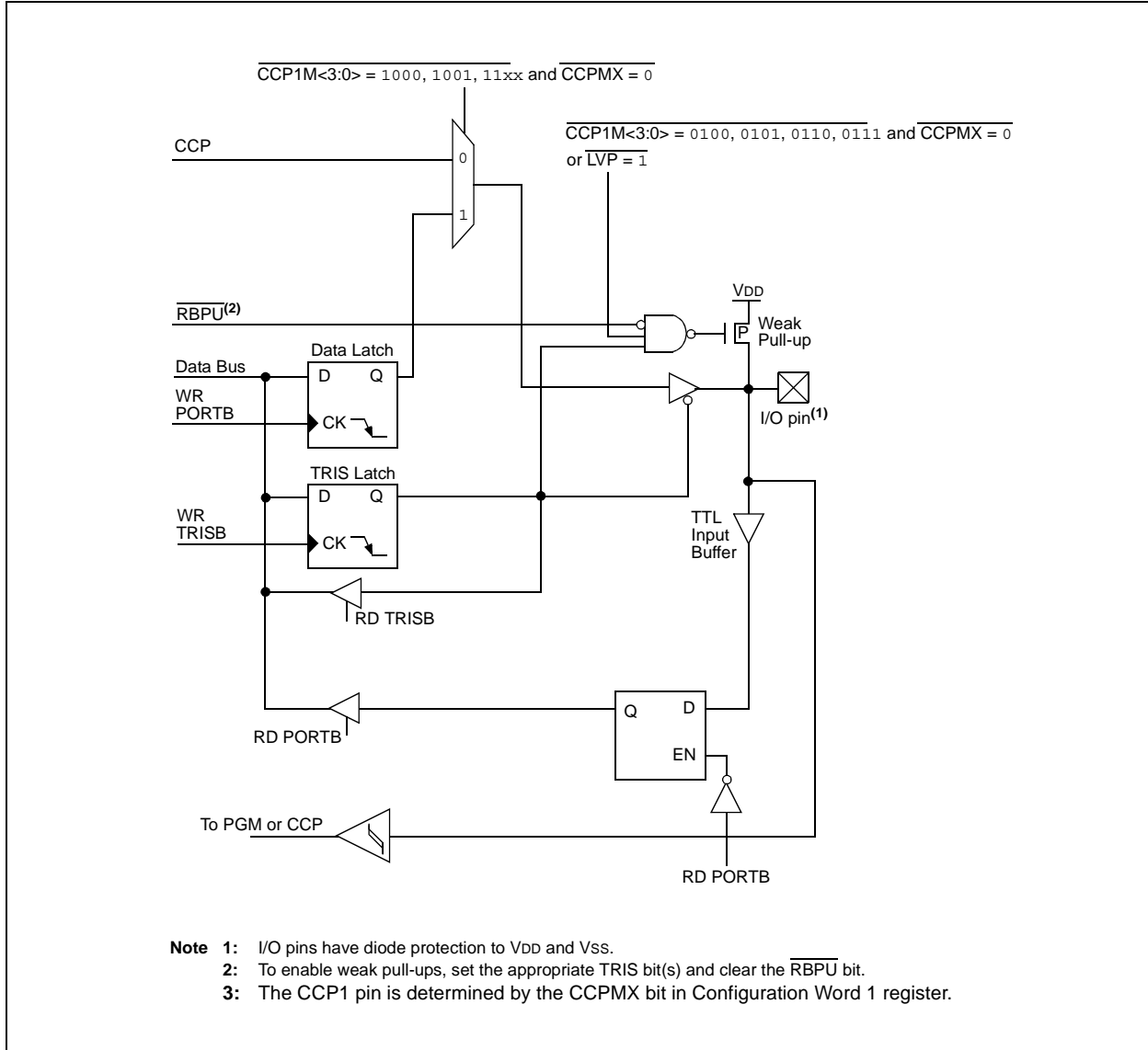
FIGURE 5-10: BLOCK DIAGRAM OF RB2/SDO/RX/DT PIN



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

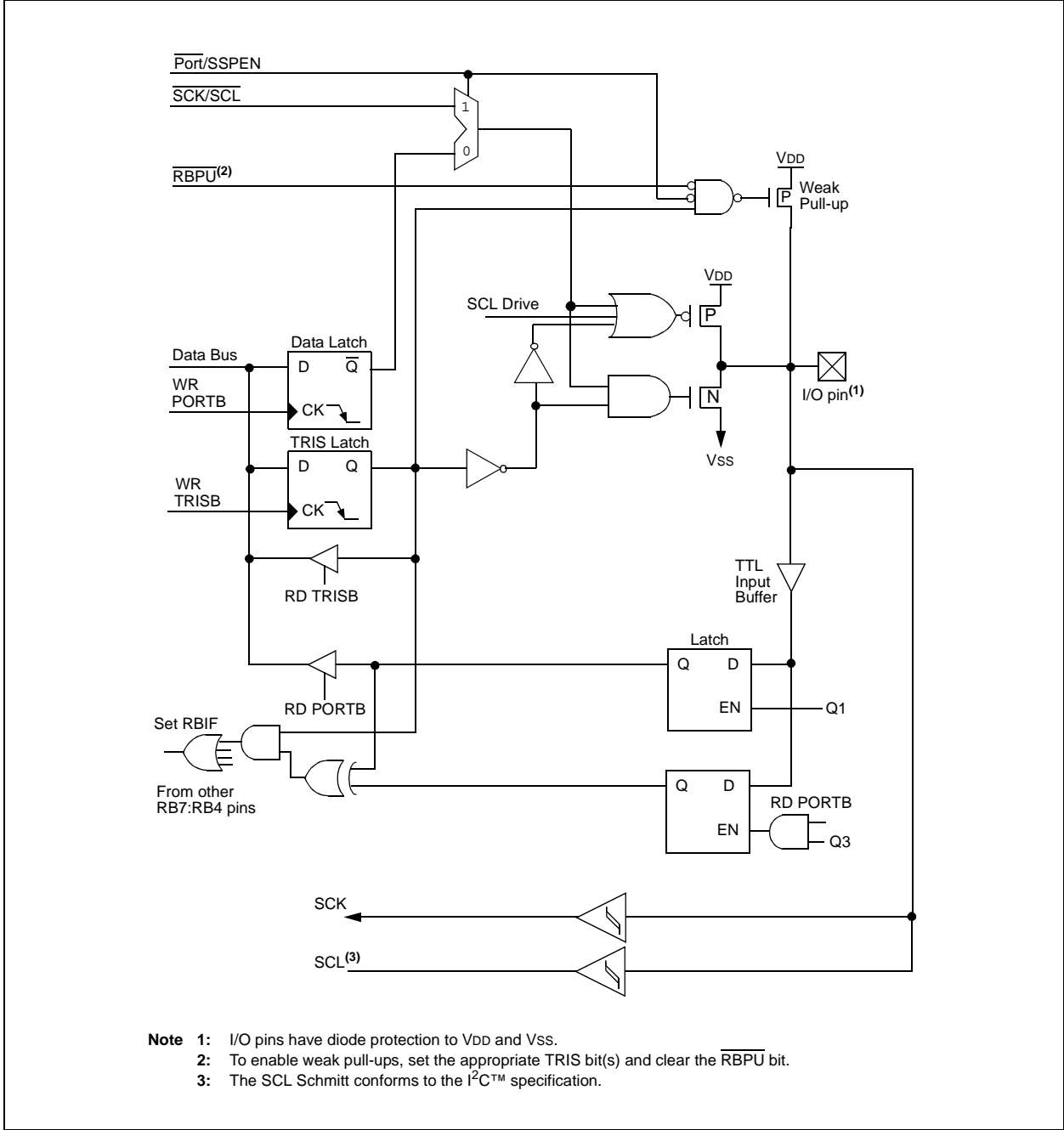
# PIC16F87/88

FIGURE 5-11: BLOCK DIAGRAM OF RB3/PGM/CCP1<sup>(3)</sup> PIN





**FIGURE 5-12: BLOCK DIAGRAM OF RB4/SCK/SCL PIN**



# PIC16F87/88

FIGURE 5-13: BLOCK DIAGRAM OF RB5/ $\overline{SS}$ /TX/CK PIN

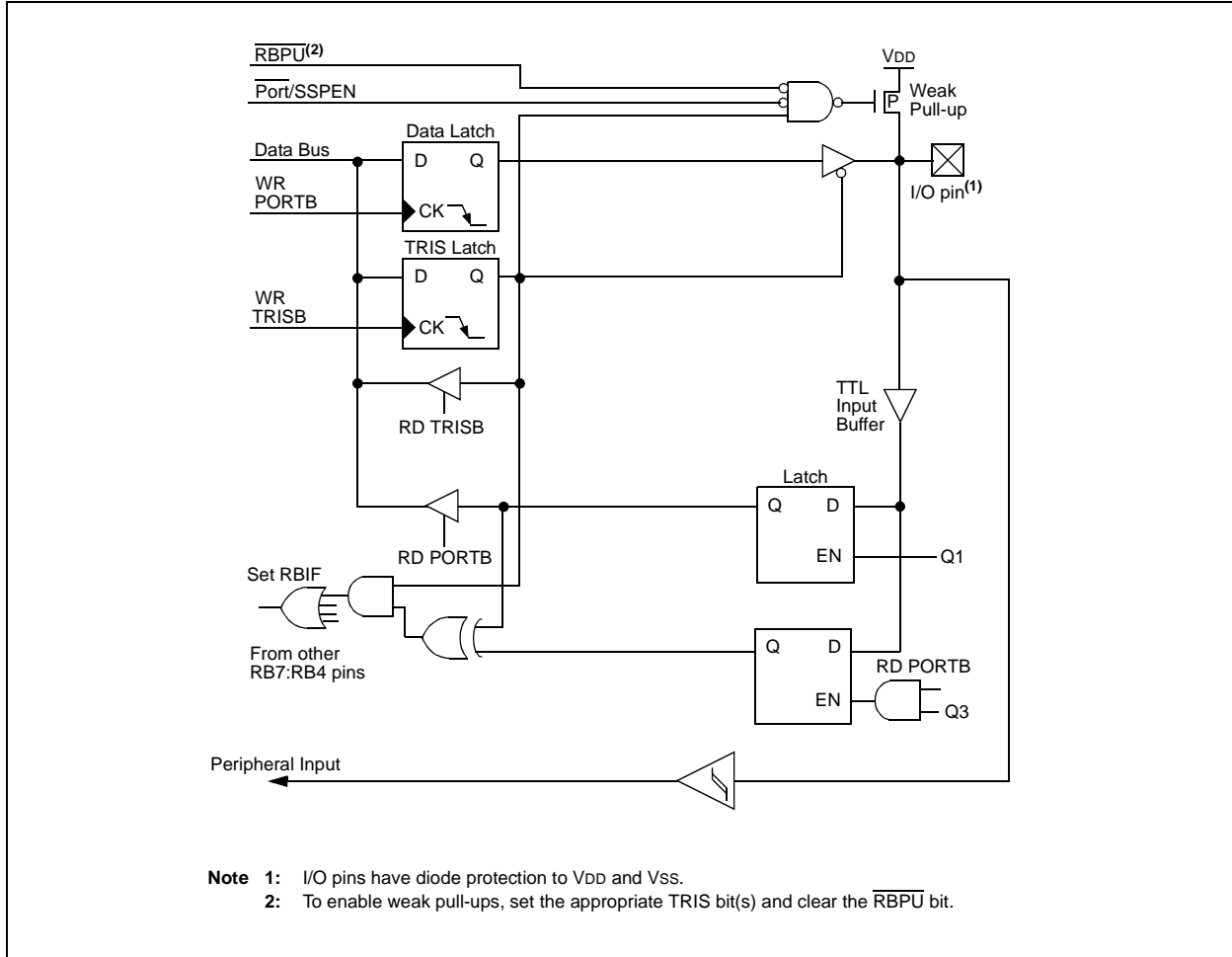
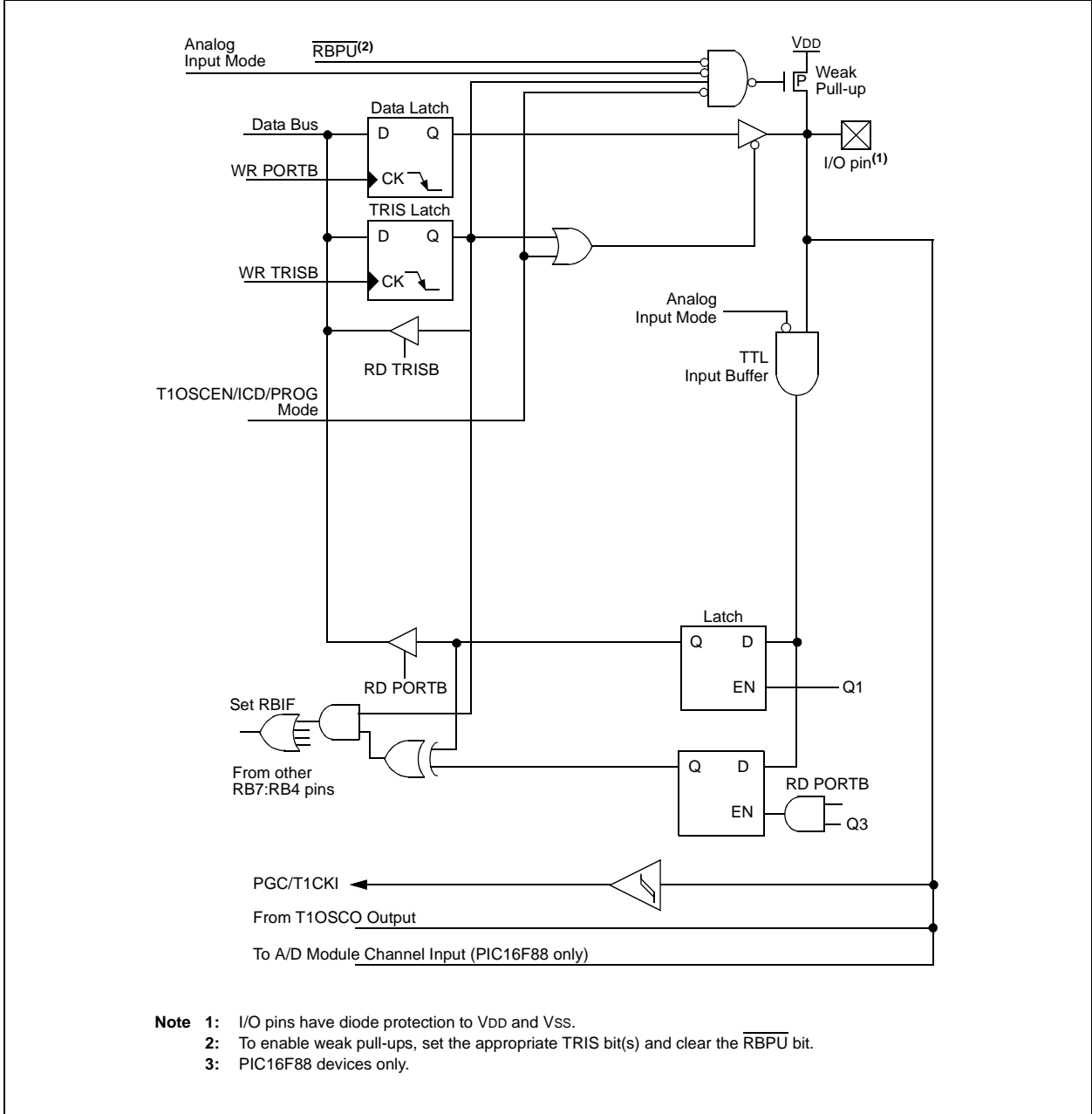


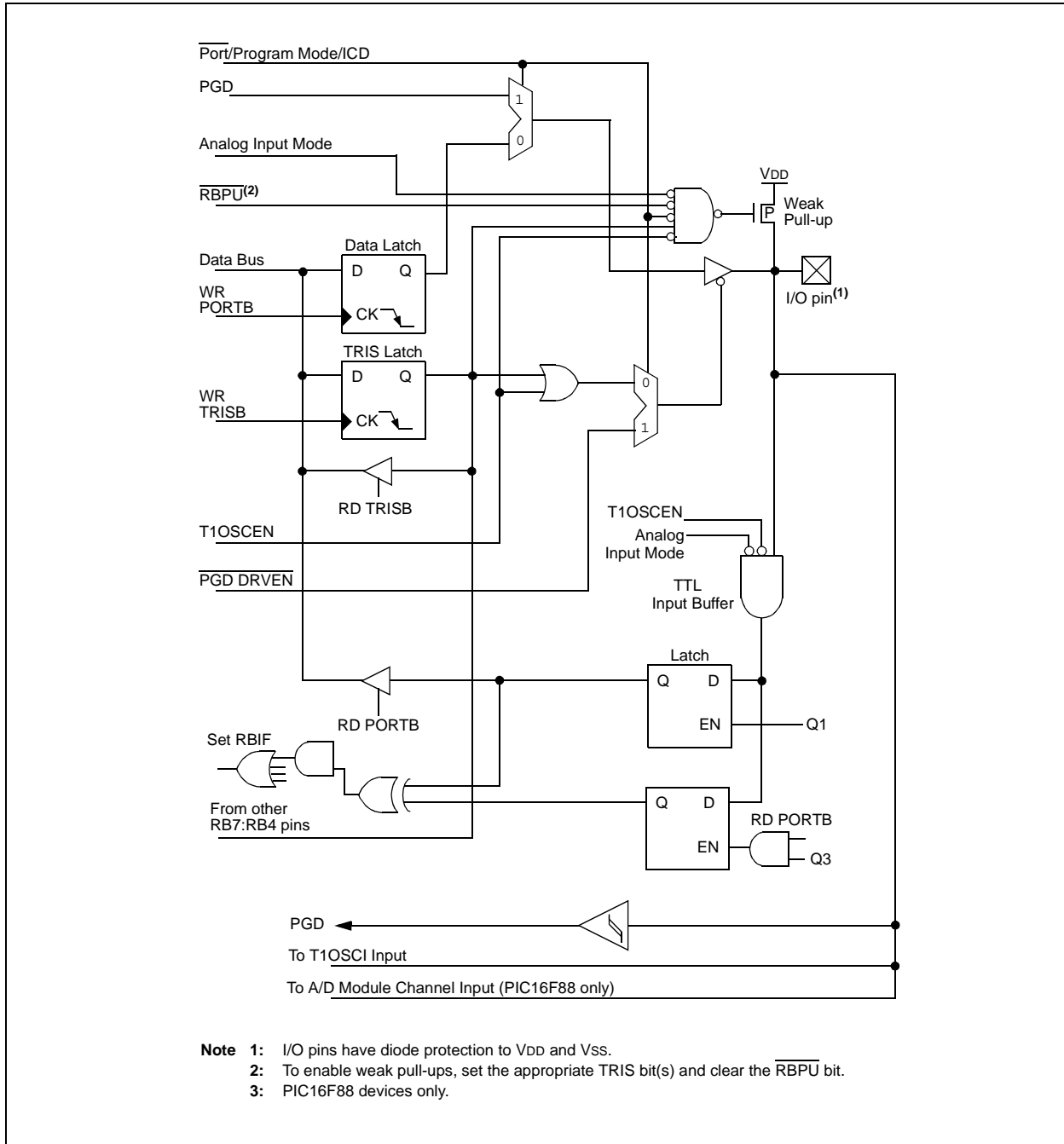
FIGURE 5-14: BLOCK DIAGRAM OF RB6/AN5<sup>(3)</sup>/PGC/T1OSO/T1CKI PIN



- Note 1:** I/O pins have diode protection to VDD and VSS.  
**Note 2:** To enable weak pull-ups, set the appropriate TRIS bit(s) and clear the  $\overline{RBPV}$  bit.  
**Note 3:** PIC16F88 devices only.

# PIC16F87/88

FIGURE 5-15: BLOCK DIAGRAM OF RB7/AN6<sup>(3)</sup>/PGD/T1OSI PIN



## 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.

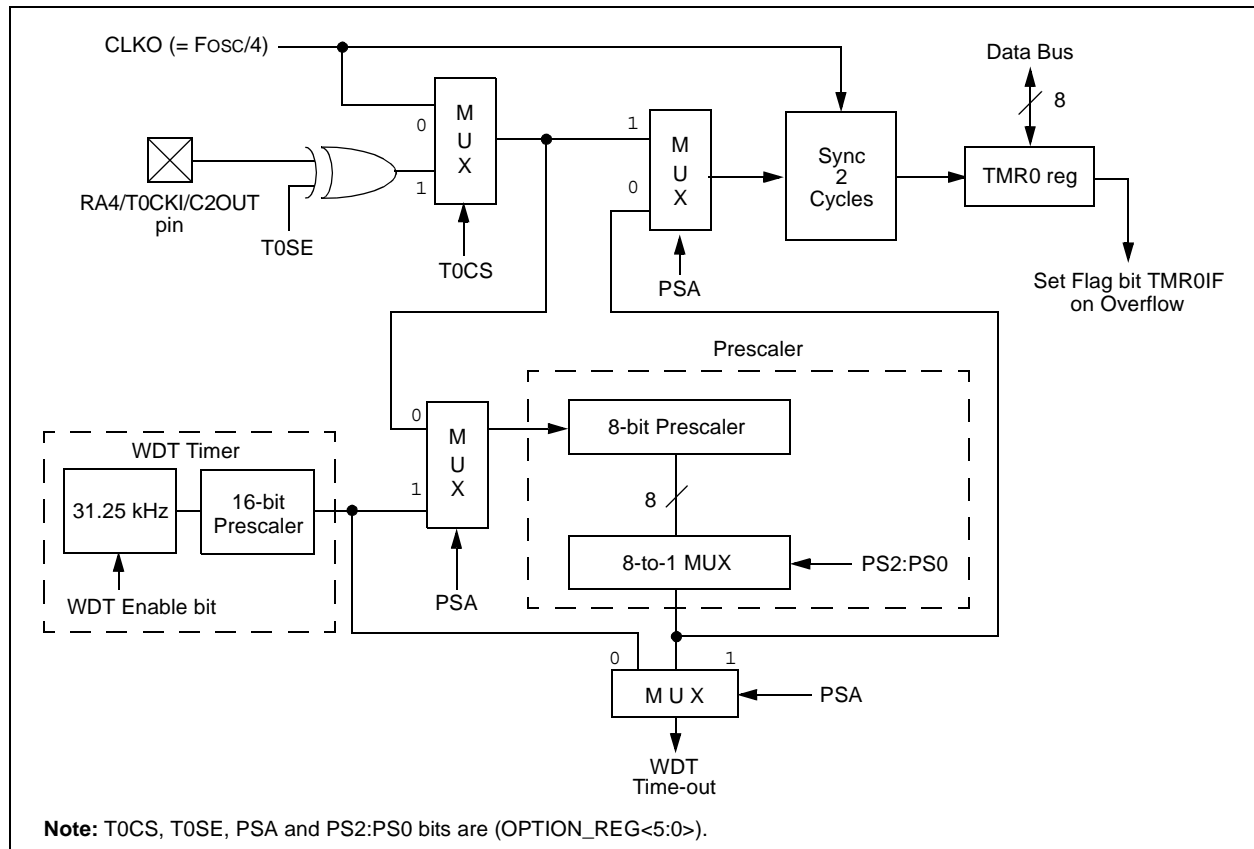
Counter mode is selected by setting bit T0CS (OPTION\_REG<5>). In Counter mode, Timer0 will increment, either on every rising or falling edge of pin RA4/TOCKI/C2OUT. The incrementing edge is determined by the Timer0 Source Edge Select bit, T0SE (OPTION\_REG<4>). Clearing bit T0SE selects the rising edge. 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**



# PIC16F87/88

## 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 the prescaler cannot be used by the Watchdog Timer and vice versa. This prescaler is not readable or writable (see Figure 6-1).

**Note:** Although the prescaler can be assigned to either the WDT or Timer0, but not both, a new divide counter is implemented in the WDT circuit to give multiple WDT time-out selections. This allows TMR0 and WDT to each have their own scaler. Refer to **Section 15.12 “Watchdog Timer (WDT)”** for further details.

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

- bit 7 **RBPU:** PORTB Pull-up Enable bit
- bit 6 **INTEDG:** Interrupt Edge Select bit
- 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 **PS<2:0>:** 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.

## EXAMPLE 6-1: 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	INT0IE	RBIE	TMR0IF	INT0IF	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. Shaded cells are not used by Timer0.

# PIC16F87/88

---

NOTES:



## 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>).

The Timer1 oscillator can be used as a secondary clock source in low-power modes. When the T1RUN bit is set along with SCS<1:0> = 01, the Timer1 oscillator is providing the system clock. If the Fail-Safe Clock Monitor is enabled and the Timer1 oscillator fails while providing the system clock, polling the T1RUN bit will indicate whether the clock is being provided by the Timer1 oscillator or another source.

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/PGC/T1OSO/T1CKI and RB7/PGD/T1OSI 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).

# PIC16F87/88

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

U-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{T1SYNC}$	TMR1CS	TMR1ON
bit 7							bit 0

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

bit 6 **T1RUN:** Timer1 System Clock Status bit  
 1 = System clock is derived from Timer1 oscillator  
 0 = System clock is derived from another source

bit 5-4 **T1CKPS<1:0>:** 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 **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/AN5<sup>(1)</sup>/PGC/T1OSO/T1CKI (on the rising edge)  
 0 = Internal clock (Fosc/4)

**Note 1:** Available on PIC16F88 devices only.

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

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

## 7.2 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  $F_{osc}/4$ . The synchronize control bit,  $\overline{T1SYNC}$  (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.

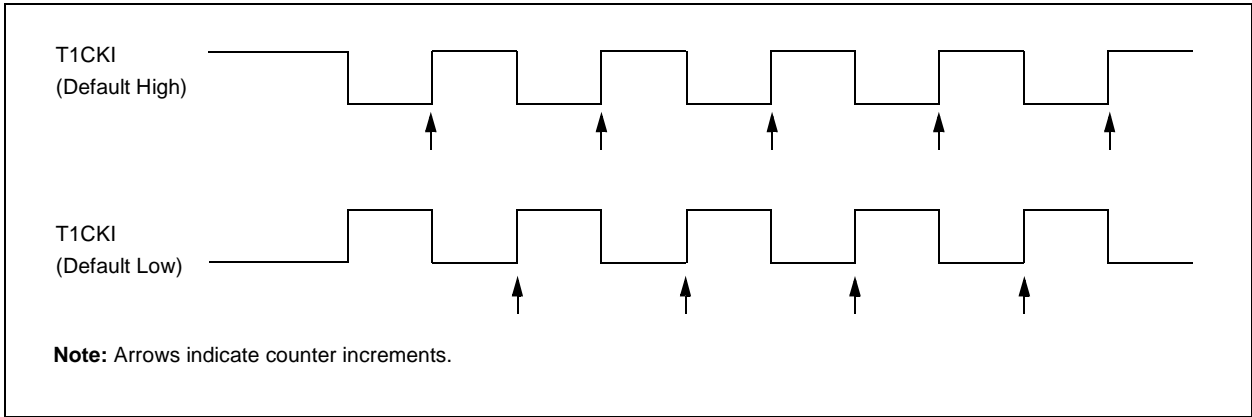
## 7.4 Timer1 Operation in Synchronized Counter Mode

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

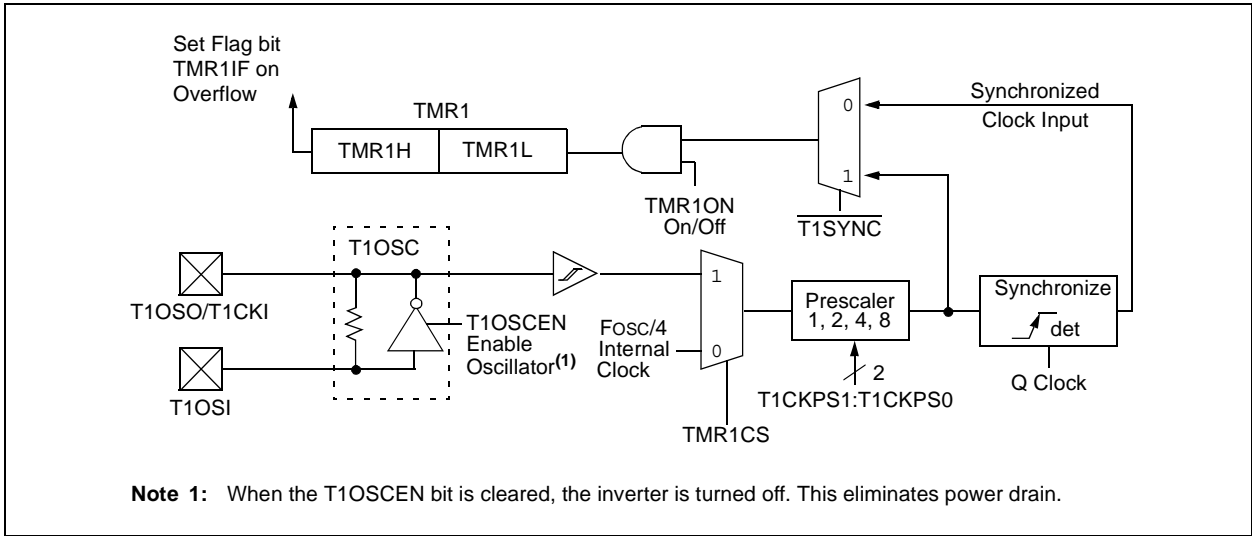
If  $\overline{T1SYNC}$  is cleared, then the external clock input is synchronized with internal phase clocks. The synchronization is done after the prescaler stage. The prescaler stage is an asynchronous ripple counter.

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**



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



# PIC16F87/88

## 7.5 Timer1 Operation in Asynchronous Counter Mode

If control bit  $\overline{T1SYNC}$  (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 (see **Section 7.5.1 "Reading and Writing Timer1 in Asynchronous Counter Mode"**).

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  TMR1L, 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
MOVWF  TMPL        ; Re-enable the Interrupt (if required)
CONTINUE    ; Continue with your code
```

## 7.6 Timer1 Oscillator

A crystal oscillator circuit is built 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 all power-managed modes. 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.

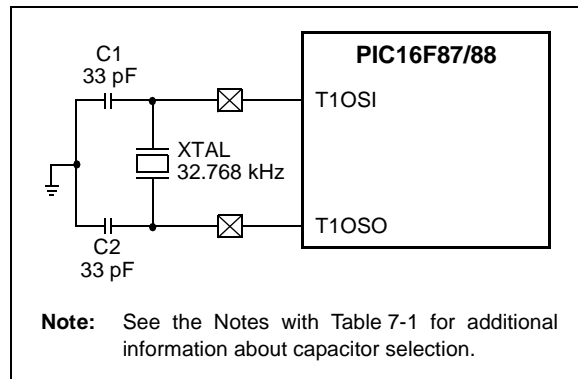
The user must provide a software time delay to ensure proper oscillator start-up.

**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**



**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.

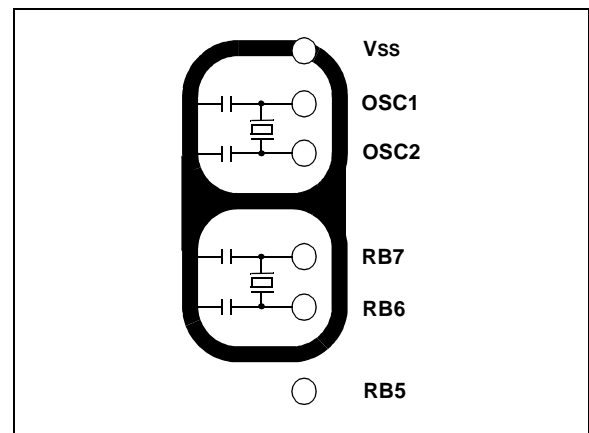
## 7.7 Timer1 Oscillator Layout Considerations

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**



# PIC16F87/88

---

## 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).

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

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

In the event that a write to Timer1 coincides with a special event trigger from 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 to 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.

## 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	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u
0Ch	PIR1	—	ADIF <sup>(1)</sup>	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	-000 0000	-000 0000
8Ch	PIE1	—	ADIE <sup>(1)</sup>	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	-000 0000	-000 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	—	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON	-000 0000	-uuu uuuu

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

**Note 1:** This bit is only implemented on the PIC16F88. The bit will read '0' on the PIC16F87.

# PIC16F87/88

---

NOTES:



## 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 ( $F_{osc}/4$ ) 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:

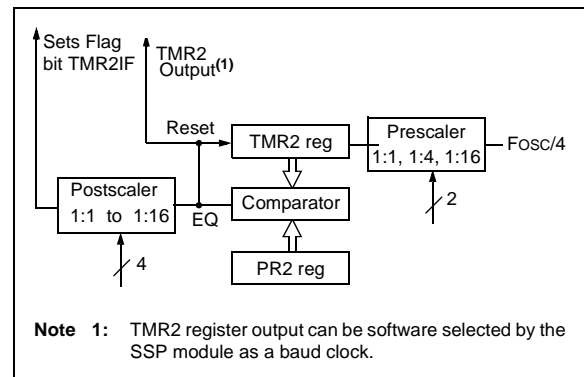
- A write to the TMR2 register
- A write to the T2CON register
- Any device Reset (Power-on Reset,  $\overline{MCLR}$ , WDT Reset or Brown-out Reset)

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 (SSP) which optionally uses it to generate a shift clock.

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



# PIC16F87/88

## 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 **TOUTPS<3:0>:** 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 **T2CKPS<1:0>:** Timer2 Clock Prescale Select bits  
 00 = Prescaler is 1  
 01 = Prescaler is 4  
 1x = Prescaler is 16

<b>Legend:</b>			
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	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u
0Ch	PIR1	—	ADIF <sup>(1)</sup>	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	-000 0000	-000 0000
8Ch	PIE1	—	ADIE <sup>(1)</sup>	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	-000 0000	-000 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.

**Note 1:** This bit is only implemented on the PIC16F88. The bit will read '0' on the PIC16F87.

## 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 RB0 or RB3. This selection is set in bit 12 (CCPMX) of the Configuration Word.

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 CCPR1L.

bit 3-0 **CCP1M<3:0>:** 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

# PIC16F87/88

## 9.1 Capture Mode

In Capture mode, CCP1H:CCP1L 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

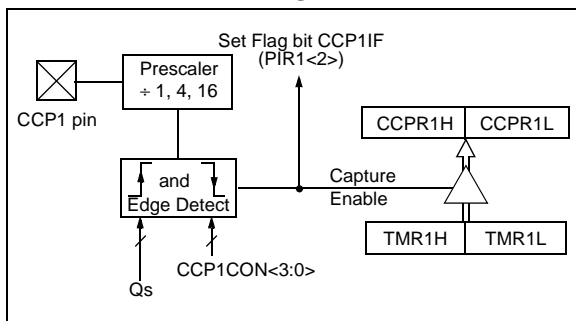
An event is selected by control bits CCP1M3:CCP1M0 (CCP1CON<3:0>). When a capture is made, the interrupt request flag bit, CCP1IF (PIR1<2>), is set. It must be cleared in software. If another capture occurs before the value in register CCPR1 is read, the old captured value is overwritten by the new captured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

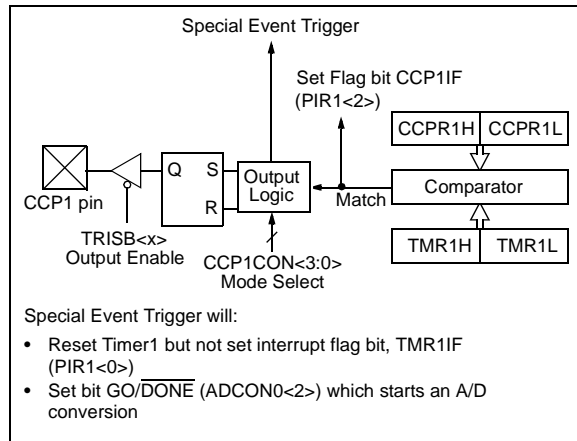
## 9.2 Compare Mode

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 (0 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	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u
0Ch	PIR1	—	ADIF <sup>(1)</sup>	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	-000 0000	-000 0000
8Ch	PIE1	—	ADIE <sup>(1)</sup>	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	-000 0000	-000 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	—	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON	-000 0000	-uuu 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.

**Note 1:** This bit is only implemented on the PIC16F88. The bit will read '0' on the PIC16F87.

# PIC16F87/88

## 9.3 PWM Mode

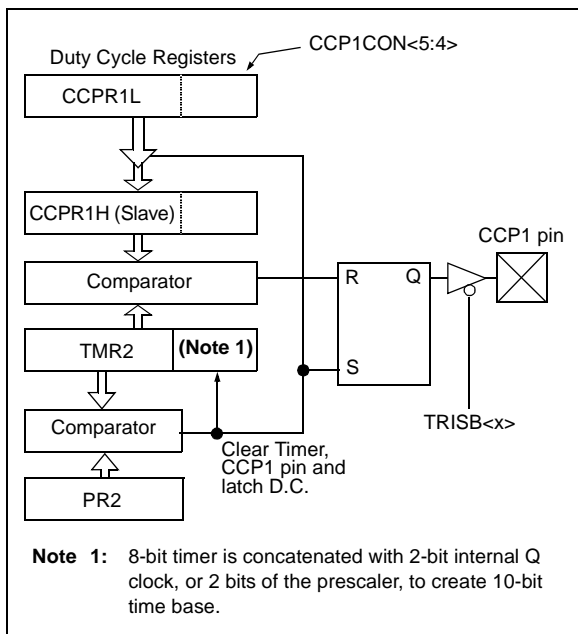
In Pulse-Width Modulation (PWM) mode, the CCP1 pin produces up to a 10-bit resolution PWM output. Since the CCP1 pin is multiplexed with the PORTB data latch, the TRISB<x> bit must be cleared to make the CCP1 pin an output.

**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.

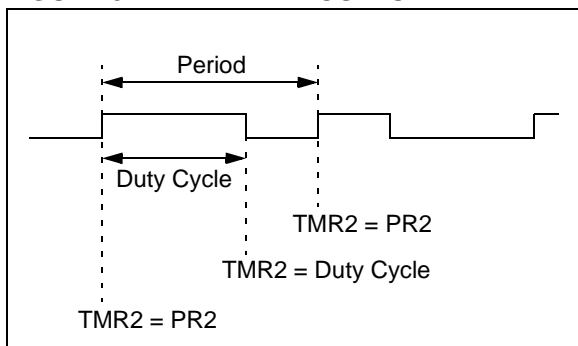
For a step-by-step procedure on how to set up the CCP module for PWM operation, see **Section 9.3.3 “Setup for PWM Operation”**.

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



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> bits contain 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.

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

**EQUATION 9-3:**

$$\text{Resolution} = \frac{\log\left(\frac{F_{\text{OSC}}}{F_{\text{PWM}}}\right)}{\log(2)} \text{ bits}$$

**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 (0 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	6.6

**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	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u
0Ch	PIR1	—	ADIF <sup>(1)</sup>	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	-000 0000	-000 0000
8Ch	PIE1	—	ADIE <sup>(1)</sup>	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	-000 0000	-000 0000
86h	TRISB	PORTB Data Direction Register								1111 1111	1111 1111
11h	TMR2	Timer2 Module Register								0000 0000	0000 0000
92h	PR2	Timer2 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.

**Note 1:** This bit is only implemented on the PIC16F88. The bit will read '0' on the PIC16F87.

# PIC16F87/88

---

NOTES:



## 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<sup>2</sup>C™)

An overview of I<sup>2</sup>C 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<sup>2</sup>C™ 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/RX/DT
- 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/ $\overline{SS}$ /TX/CK

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>).

# PIC16F87/88

## 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 $\bar{A}$	P <sup>(1)</sup>	S <sup>(1)</sup>	R $\bar{W}$	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<sup>2</sup>C 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>).
- bit 5 **D $\bar{A}$ :** Data/Address bit (I<sup>2</sup>C mode only)  
In I<sup>2</sup>C 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<sup>(1)</sup> (I<sup>2</sup>C mode only)  
 1 = Indicates that a Stop bit has been detected last  
 0 = Stop bit was not detected last
- bit 3 **S:** Start bit<sup>(1)</sup> (I<sup>2</sup>C 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 $\bar{W}$ :** Read/Write Information bit (I<sup>2</sup>C mode only)  
 Holds the R $\bar{W}$  bit information following the last address match and is only valid from address match to the next Start bit, Stop bit or ACK bit.  
 1 = Read  
 0 = Write
- bit 1 **UA:** Update Address bit (10-bit I<sup>2</sup>C 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<sup>2</sup>C modes):  
 1 = Receive complete, SSPBUF is full  
 0 = Receive not complete, SSPBUF is empty  
Transmit (in I<sup>2</sup>C 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

## REGISTER 10-2: SSPCON: SYNCHRONOUS SERIAL PORT CONTROL REGISTER (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 <sup>(1)</sup>	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<sup>2</sup>C 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<sup>(1)</sup>

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<sup>2</sup>C 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<sup>2</sup>C 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<sup>2</sup>C Slave mode, 7-bit address

0111 = I<sup>2</sup>C Slave mode, 10-bit address

1011 = I<sup>2</sup>C Firmware Controlled Master mode (Slave Idle)

1110 = I<sup>2</sup>C Slave mode, 7-bit address with Start and Stop bit interrupts enabled

1111 = I<sup>2</sup>C 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

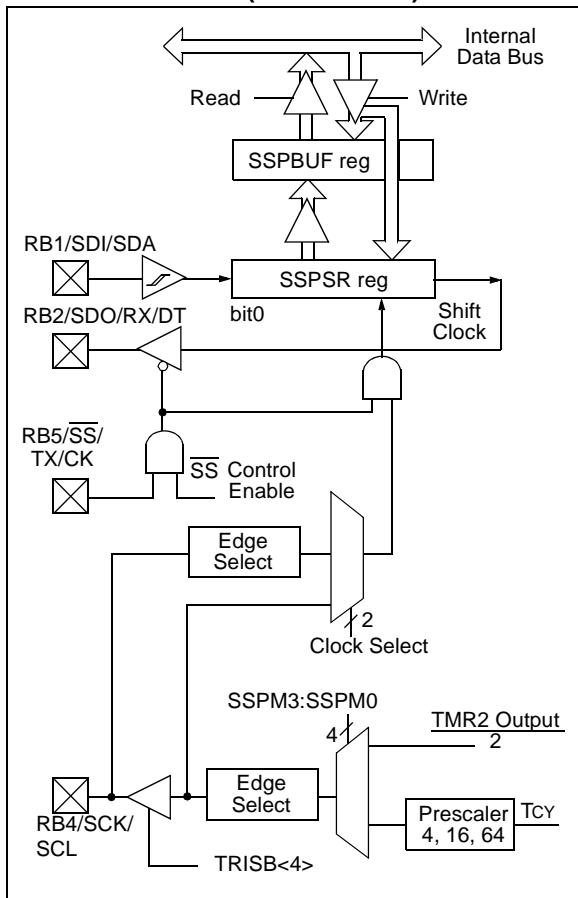
‘1’ = Bit is set

‘0’ = Bit is cleared

x = Bit is unknown

# PIC16F87/88

**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 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
- SS must have TRISB<5> set

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

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

**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	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u
0Ch	PIR1	—	ADIF <sup>(1)</sup>	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	-000 0000	-000 0000
8Ch	PIE1	—	ADIE <sup>(1)</sup>	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	-000 0000	-000 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/A	P	S	R/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.

**Note 1:** This bit is only implemented on the PIC16F88. The bit will read '0' on the PIC16F87.

FIGURE 10-2: SPI™ MODE TIMING (MASTER MODE)

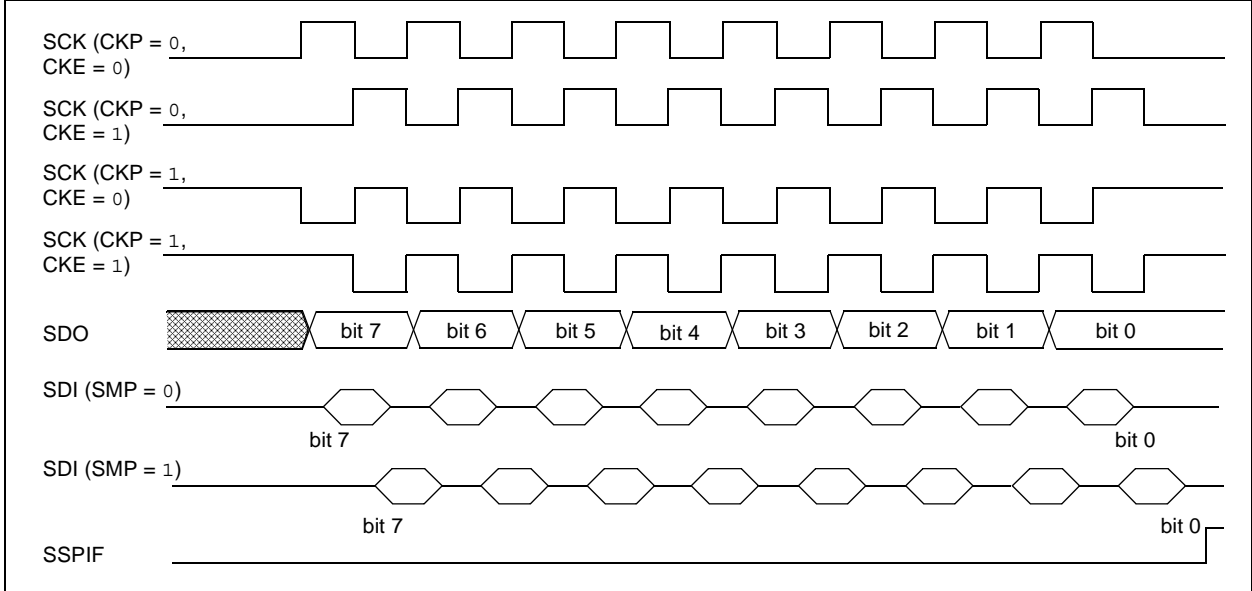


FIGURE 10-3: SPI™ MODE TIMING (SLAVE MODE WITH CKE = 0)

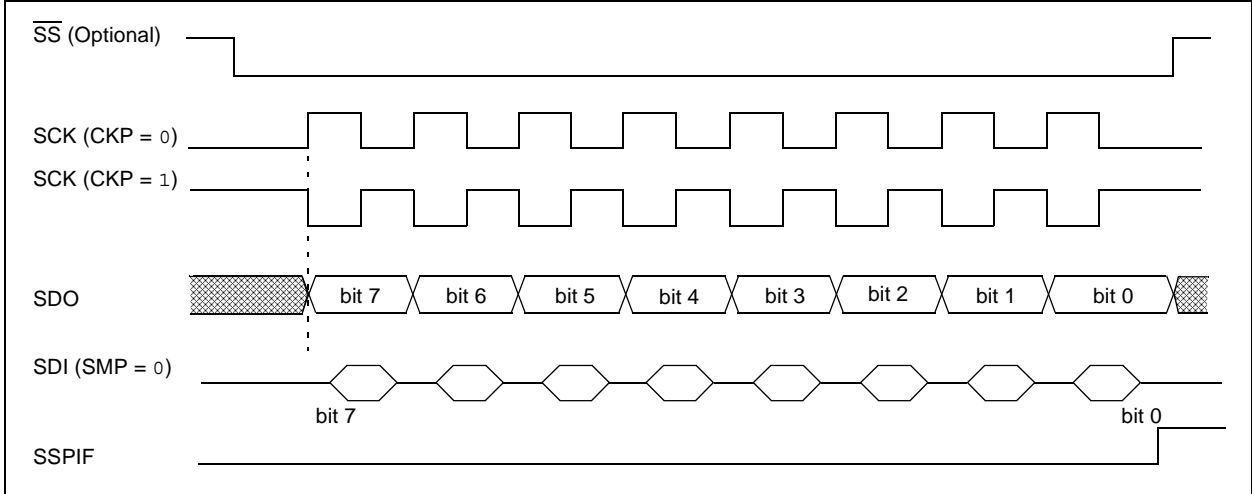
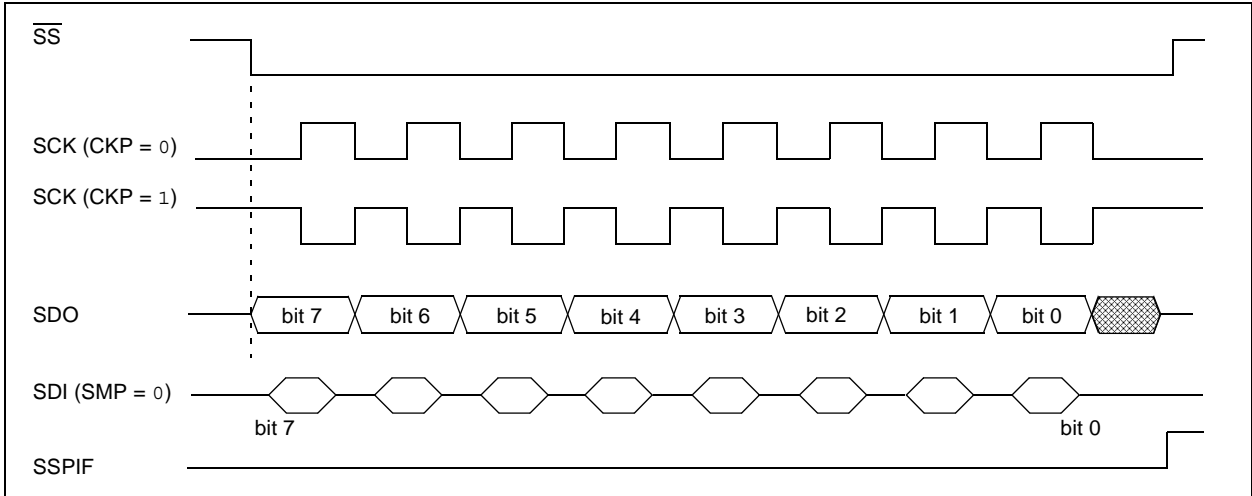


FIGURE 10-4: SPI™ MODE TIMING (SLAVE MODE WITH CKE = 1)



# PIC16F87/88

## 10.3 SSP I<sup>2</sup>C Mode Operation

The SSP module in I<sup>2</sup>C 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 I<sup>2</sup>C Slave mode, the TRIS bits (TRISx [SDA, SCL]) corresponding to the I<sup>2</sup>C pins must be set to '1'. If any TRIS bits (TRISx<7:0>) of the port containing the I<sup>2</sup>C pins (PORTx [SDA, SCL]) are changed in software during I<sup>2</sup>C communication using a Read-Modify-Write instruction (BSF, BCF), then the I<sup>2</sup>C mode may stop functioning properly and I<sup>2</sup>C communication may suspend. Do not change any of the TRISx bits (TRIS bits of the port containing the I<sup>2</sup>C pins) using the instruction BSF or BCF during I<sup>2</sup>C communication. If it is absolutely necessary to change the TRISx bits during communication, the following method can be used:

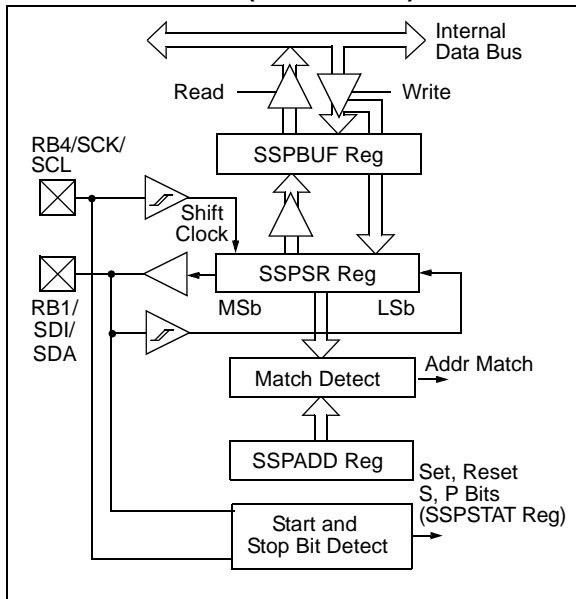
### EXAMPLE 10-1:

```
MOVWF  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 (I<sup>2</sup>C™ MODE)**



The SSP module has five registers for I<sup>2</sup>C operation:

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

The SSPCON register allows control of the I<sup>2</sup>C operation. Four mode selection bits (SSPCON<3:0>) allow one of the following I<sup>2</sup>C modes to be selected:

- I<sup>2</sup>C Slave mode (7-bit address)
- I<sup>2</sup>C Slave mode (10-bit address)
- I<sup>2</sup>C Slave mode (7-bit address) with Start and Stop bit interrupts enabled to support Firmware Controlled Master mode
- I<sup>2</sup>C Slave mode (10-bit address) with Start and Stop bit interrupts enabled to support Firmware Controlled Master mode
- I<sup>2</sup>C Firmware Controlled Master mode operation with Start and Stop bit interrupts enabled; slave is Idle

Selection of any I<sup>2</sup>C 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 I<sup>2</sup>C module.

Additional information on SSP I<sup>2</sup>C operation may be found in the "PICmicro® Mid-Range MCU Family Reference Manual" (DS33023).

## 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 ( $\overline{\text{ACK}}$ ) 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<sup>2</sup>C 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  $\overline{\text{ACK}}$  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  $\overline{\text{R}/\overline{\text{W}}}$  (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 mode is as follows, with steps 7-9 for slave transmitter:

1. Receive first (high) byte of address (bits SSPIF, BF and 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  $\overline{\text{R}/\overline{\text{W}}}$  bit of the address byte is clear and an address match occurs, the  $\overline{\text{R}/\overline{\text{W}}}$  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 ( $\overline{\text{ACK}}$ ) 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  $\overline{\text{R}/\overline{\text{W}}}$  bit of the incoming address byte is set and an address match occurs, the  $\overline{\text{R}/\overline{\text{W}}}$  bit of the SSPSTAT register is set. The received address is loaded into the SSPBUF register. The  $\overline{\text{ACK}}$  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).

# PIC16F87/88

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  $\overline{\text{ACK}}$  pulse from the master receiver is latched on the rising edge of the ninth SCL input pulse. If the SDA line was high (not ACK), then

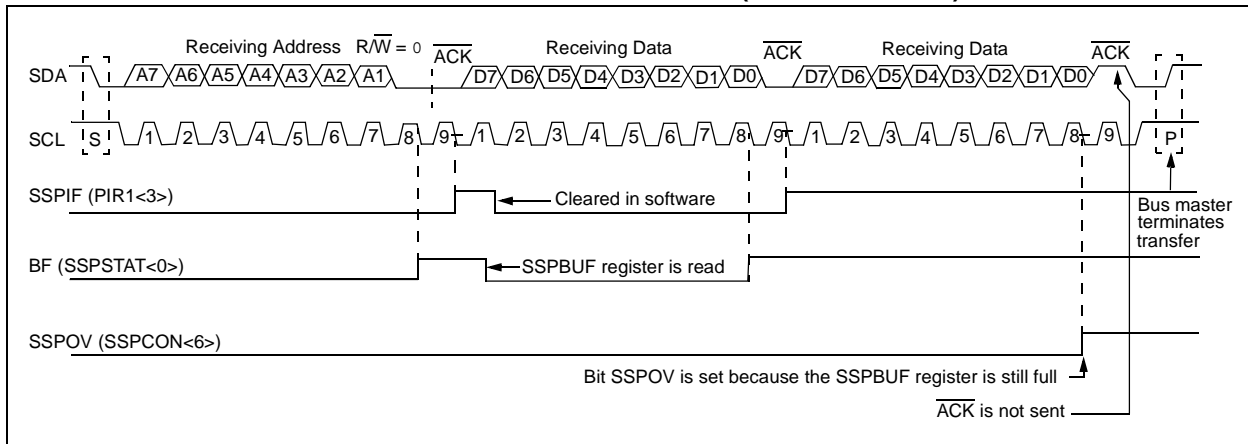
the data transfer is complete. When the  $\overline{\text{ACK}}$  is latched by the slave device, the slave logic is reset (resets SSPSTAT register) and the slave device then monitors for another occurrence of the Start bit. If the SDA line was low (ACK), the transmit data must be loaded into the SSPBUF 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**

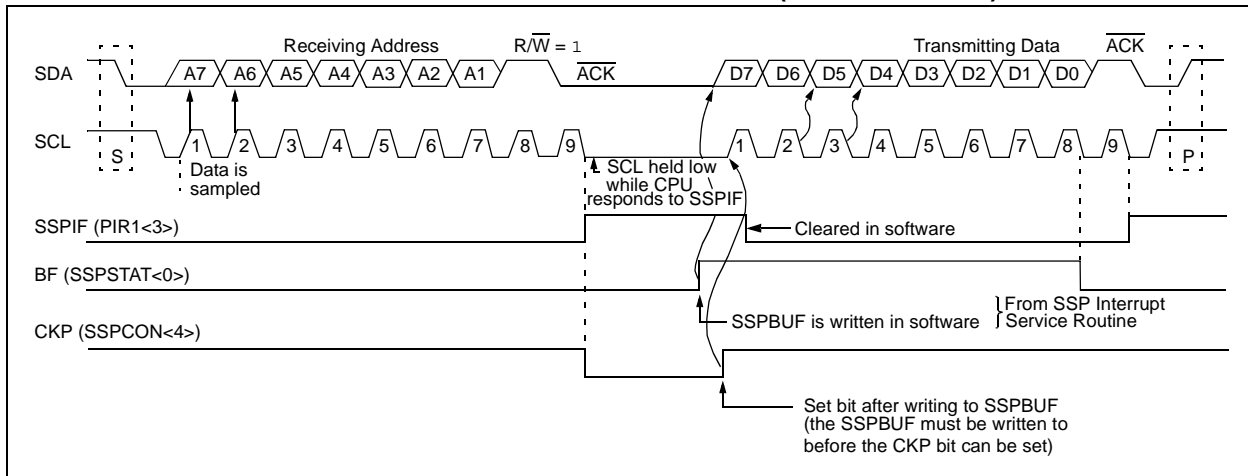
Status Bits as Data Transfer is Received		SSPSR → SSPBUF	Generate $\overline{\text{ACK}}$ Pulse	Set SSPIF Bit (SSP Interrupt Occurs if Enabled)
BF	SSPOV			
0	0	Yes	Yes	Yes
1	0	No	No	Yes
1	1	No	No	Yes
0	1	No	No	Yes

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

**FIGURE 10-6: I<sup>2</sup>C™ WAVEFORMS FOR RECEPTION (7-BIT ADDRESS)**



**FIGURE 10-7: I<sup>2</sup>C™ WAVEFORMS FOR TRANSMISSION (7-BIT ADDRESS)**





## 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<sup>2</sup>C 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<sup>2</sup>C 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 Application Note AN554, "Software Implementation of I<sup>2</sup>C™ Bus Master".

## 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<sup>2</sup>C 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  $\overline{\text{ACK}}$  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 Application Note AN578, "Use of the SSP Module in the of I<sup>2</sup>C™ Multi-Master Environment".

**TABLE 10-3: REGISTERS ASSOCIATED WITH I<sup>2</sup>C™ 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	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u
0Ch	PIR1	—	ADIF <sup>(1)</sup>	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	-000 0000	-000 0000
8Ch	PIE1	—	ADIE <sup>(1)</sup>	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	-000 0000	-000 0000
13h	SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register								xxxxx xxxxx	uuuu uuuu
93h	SSPADD	Synchronous Serial Port (I <sup>2</sup> C mode) Address Register								0000 0000	0000 0000
14h	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	0000 0000
94h	SSPSTAT	SMP <sup>(2)</sup>	CKE <sup>(2)</sup>	D $\overline{\text{A}}$	P	S	R $\overline{\text{W}}$	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:** This bit is only implemented on the PIC16F88. The bit will read '0' on the PIC16F87.  
**2:** Maintain these bits clear in I<sup>2</sup>C™ mode.

# PIC16F87/88

---

NOTES:

## 11.0 ADDRESSABLE UNIVERSAL SYNCHRONOUS ASYNCHRONOUS RECEIVER TRANSMITTER (AUSART)

The Addressable Universal Synchronous Asynchronous Receiver Transmitter (AUSART) module is one of the two serial I/O modules. (AUSART is also known as a Serial Communications Interface or SCI.) The AUSART can be configured as a full-duplex asynchronous system that can communicate with peripheral devices, such as CRT terminals and personal computers, or it can be configured as a half-duplex synchronous system that can communicate with peripheral devices, such as A/D or D/A integrated circuits, serial EEPROMs, etc.

The AUSART can be configured in the following modes:

- Asynchronous (full-duplex)
- Synchronous – Master (half-duplex)
- Synchronous – Slave (half-duplex)

Bit SPEN (RCSTA<7>) and bits TRISB<5,2> have to be set in order to configure pins, RB5/ $\overline{SS}$ /TX/CK and RB2/SDO/RX/DT, as the Addressable Universal Synchronous Asynchronous Receiver Transmitter.

The AUSART module also has a multi-processor communication capability, using 9-bit address detection.

### REGISTER 11-1: TXSTA: TRANSMIT STATUS AND CONTROL REGISTER (ADDRESS 98h)

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R-1	R/W-0
CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D
						bit 7	bit 0

- bit 7 **CSRC:** Clock Source Select bit  
Asynchronous mode:  
 Don't care.  
Synchronous mode:  
 1 = Master mode (clock generated internally from BRG)  
 0 = Slave mode (clock from external source)
- bit 6 **TX9:** 9-bit Transmit Enable bit  
 1 = Selects 9-bit transmission  
 0 = Selects 8-bit transmission
- bit 5 **TXEN:** Transmit Enable bit  
 1 = Transmit enabled  
 0 = Transmit disabled  
**Note:** SREN/CREN overrides TXEN in Sync mode.
- bit 4 **SYNC:** AUSART Mode Select bit  
 1 = Synchronous mode  
 0 = Asynchronous mode
- bit 3 **Unimplemented:** Read as '0'
- bit 2 **BRGH:** High Baud Rate Select bit  
Asynchronous mode:  
 1 = High speed  
 0 = Low speed  
Synchronous mode:  
 Unused in this mode.
- bit 1 **TRMT:** Transmit Shift Register Status bit  
 1 = TSR empty  
 0 = TSR full
- bit 0 **TX9D:** 9th bit of Transmit Data, can be Parity bit

#### Legend:

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

# PIC16F87/88

## REGISTER 11-2: RCSTA: RECEIVE STATUS AND CONTROL REGISTER (ADDRESS 18h)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R-0	R-x
SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
bit 7						bit 0	

- bit 7     **SPEN:** Serial Port Enable bit  
 1 = Serial port enabled (configures RB2/SDO/RX/DT and RB5/ $\overline{SS}$ /TX/CK pins as serial port pins)  
 0 = Serial port disabled
- bit 6     **RX9:** 9-bit Receive Enable bit  
 1 = Selects 9-bit reception  
 0 = Selects 8-bit reception
- bit 5     **SREN:** Single Receive Enable bit  
Asynchronous mode:  
 Don't care.  
Synchronous mode – Master:  
 1 = Enables single receive  
 0 = Disables single receive  
 This bit is cleared after reception is complete.  
Synchronous mode – Slave:  
 Don't care.
- bit 4     **CREN:** Continuous Receive Enable bit  
Asynchronous mode:  
 1 = Enables continuous receive  
 0 = Disables continuous receive  
Synchronous mode:  
 1 = Enables continuous receive until enable bit CREN is cleared (CREN overrides SREN)  
 0 = Disables continuous receive
- bit 3     **ADDEN:** Address Detect Enable bit  
Asynchronous mode 9-bit (RX9 = 1):  
 1 = Enables address detection, enables interrupt and load of the receive buffer when RSR<8> is set  
 0 = Disables address detection, all bytes are received and ninth bit can be used as parity bit
- bit 2     **FERR:** Framing Error bit  
 1 = Framing error (can be updated by reading RCREG register and receive next valid byte)  
 0 = No framing error
- bit 1     **OERR:** Overrun Error bit  
 1 = Overrun error (can be cleared by clearing bit CREN)  
 0 = No overrun error
- bit 0     **RX9D:** 9th bit of Received Data (can be Parity bit, but must be calculated by user firmware)

**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

## 11.1 AUSART Baud Rate Generator (BRG)

The BRG supports both the Asynchronous and Synchronous modes of the AUSART. It is a dedicated 8-bit Baud Rate Generator. The SPBRG register controls the period of a free running 8-bit timer. In Asynchronous mode, bit BRGH (TXSTA<2>) also controls the baud rate. In Synchronous mode, bit BRGH is ignored. Table 11-1 shows the formula for computation of the baud rate for different AUSART modes which only apply in Master mode (internal clock).

Given the desired baud rate and FOSC, the nearest integer value for the SPBRG register can be calculated using the formula in Table 11-1. From this, the error in baud rate can be determined.

It may be advantageous to use the high baud rate (BRGH = 1) even for slower baud clocks. This is because the  $FOSC/(16(X + 1))$  equation can reduce the baud rate error in some cases.

Writing a new value to the SPBRG register causes the BRG timer to be reset (or cleared). This ensures the BRG does not wait for a timer overflow before outputting the new baud rate.

### 11.1.1 AUSART AND INTRC OPERATION

The PIC16F87/88 has an 8 MHz INTRC that can be used as the system clock, thereby eliminating the need for external components to provide the clock source. When the INTRC provides the system clock, the AUSART module will also use the INTRC as its system clock. Table 11-1 shows some of the INTRC frequencies that can be used to generate the AUSART module's baud rate.

### 11.1.2 LOW-POWER MODE OPERATION

The system clock is used to generate the desired baud rate; however, when a low-power mode is entered, the low-power clock source may be operating at a different frequency than in full power execution. In Sleep mode, no clocks are present. This may require the value in SPBRG to be adjusted.

### 11.1.3 SAMPLING

The data on the RB2/SDO/RX/DT pin is sampled three times by a majority detect circuit to determine if a high or a low level is present at the RX pin.

**TABLE 11-1: BAUD RATE FORMULA**

SYNC	BRGH = 0 (Low Speed)	BRGH = 1 (High Speed)
0	(Asynchronous) Baud Rate = $FOSC/(64(X + 1))$	Baud Rate = $FOSC/(16(X + 1))$
1	(Synchronous) Baud Rate = $FOSC/(4(X + 1))$	N/A

**Legend:** X = value in SPBRG (0 to 255)

**TABLE 11-2: REGISTERS ASSOCIATED WITH BAUD RATE GENERATOR**

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	
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010	
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x	
99h	SPBRG	Baud Rate Generator Register									0000 0000	0000 0000

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

# PIC16F87/88

**TABLE 11-3: BAUD RATES FOR ASYNCHRONOUS MODE (BRGH = 0)**

BAUD RATE (K)	Fosc = 20 MHz			Fosc = 16 MHz			Fosc = 10 MHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	—	—	—	—	—	—	—	—	—
1.2	1.221	+1.75	255	1.202	+0.17	207	1.202	+0.17	129
2.4	2.404	+0.17	129	2.404	+0.17	103	2.404	+0.17	64
9.6	9.766	+1.73	31	9.615	+0.16	25	9.766	+1.73	15
19.2	19.531	+1.72	15	19.231	+0.16	12	19.531	+1.72	7
28.8	31.250	+8.51	9	27.778	-3.55	8	31.250	+8.51	4
33.6	34.722	+3.34	8	35.714	+6.29	6	31.250	-6.99	4
57.6	62.500	+8.51	4	62.500	+8.51	3	52.083	-9.58	2
HIGH	1.221	—	255	0.977	—	255	0.610	—	255
LOW	312.500	—	0	250.000	—	0	156.250	—	0

BAUD RATE (K)	Fosc = 4 MHz			Fosc = 3.6864 MHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	0.300	0	207	0.3	0	191
1.2	1.202	+0.17	51	1.2	0	47
2.4	2.404	+0.17	25	2.4	0	23
9.6	8.929	+6.99	6	9.6	0	5
19.2	20.833	+8.51	2	19.2	0	2
28.8	31.250	+8.51	1	28.8	0	1
33.6	—	—	—	—	—	—
57.6	62.500	+8.51	0	57.6	0	0
HIGH	0.244	—	255	0.225	—	255
LOW	62.500	—	0	57.6	—	0

**TABLE 11-4: BAUD RATES FOR ASYNCHRONOUS MODE (BRGH = 1)**

BAUD RATE (K)	Fosc = 20 MHz			Fosc = 16 MHz			Fosc = 10 MHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	—	—	—	—	—	—	—	—	—
1.2	—	—	—	—	—	—	—	—	—
2.4	—	—	—	—	—	—	2.441	+1.71	255
9.6	9.615	+0.16	129	9.615	+0.16	103	9.615	+0.16	64
19.2	19.231	+0.16	64	19.231	+0.16	51	19.531	+1.72	31
28.8	29.070	+0.94	42	29.412	+2.13	33	28.409	-1.36	21
33.6	33.784	+0.55	36	33.333	-0.79	29	32.895	-2.10	18
57.6	59.524	+3.34	20	58.824	+2.13	16	56.818	-1.36	10
HIGH	4.883	—	255	3.906	—	255	2.441	—	255
LOW	1250.000	—	0	1000.000	—	0	625.000	—	0

BAUD RATE (K)	Fosc = 4 MHz			Fosc = 3.6864 MHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	—	—	—	—	—	—
1.2	1.202	+0.17	207	1.2	0	191
2.4	2.404	+0.17	103	2.4	0	95
9.6	9.615	+0.16	25	9.6	0	23
19.2	19.231	+0.16	12	19.2	0	11
28.8	27.798	-3.55	8	28.8	0	7
33.6	35.714	+6.29	6	32.9	-2.04	6
57.6	62.500	+8.51	3	57.6	0	3
HIGH	0.977	—	255	0.9	—	255
LOW	250.000	—	0	230.4	—	0

**TABLE 11-5: INTRC BAUD RATES FOR ASYNCHRONOUS MODE (BRGH = 0)**

BAUD RATE (K)	Fosc = 8 MHz			Fosc = 4 MHz			Fosc = 2 MHz			Fosc = 1 MHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	NA	—	—	0.300	0	207	0.300	0	103	0.300	0	51
1.2	1.202	+0.16	103	1.202	+0.16	51	1.202	+0.16	25	1.202	+0.16	12
2.4	2.404	+0.16	51	2.404	+0.16	25	2.404	+0.16	12	2.232	-6.99	6
9.6	9.615	+0.16	12	8.929	-6.99	6	10.417	+8.51	2	NA	—	—
19.2	17.857	-6.99	6	20.833	+8.51	2	NA	—	—	NA	—	—
28.8	31.250	+8.51	3	31.250	+8.51	1	31.250	+8.51	0	NA	—	—
38.4	41.667	+8.51	2	NA	—	—	NA	—	—	NA	—	—
57.6	62.500	+8.51	1	62.500	8.51	0	NA	—	—	NA	—	—

**TABLE 11-6: INTRC BAUD RATES FOR ASYNCHRONOUS MODE (BRGH = 1)**

BAUD RATE (K)	Fosc = 8 MHz			Fosc = 4 MHz			Fosc = 2 MHz			Fosc = 1 MHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	NA	—	—	NA	—	—	NA	—	—	0.300	0	207
1.2	NA	—	—	1.202	+0.16	207	1.202	+0.16	103	1.202	+0.16	51
2.4	2.404	+0.16	207	2.404	+0.16	103	2.404	+0.16	51	2.404	+0.16	25
9.6	9.615	+0.16	51	9.615	+0.16	25	9.615	+0.16	12	8.929	-6.99	6
19.2	19.231	+0.16	25	19.231	+0.16	12	17.857	-6.99	6	20.833	+8.51	2
28.8	29.412	+2.12	16	27.778	-3.55	8	31.250	+8.51	3	31.250	+8.51	1
38.4	38.462	+0.16	12	35.714	-6.99	6	41.667	+8.51	2	NA	—	—
57.6	55.556	-3.55	8	62.500	+8.51	3	62.500	+8.51	1	62.500	+8.51	0

# PIC16F87/88

## 11.2 AUSART Asynchronous Mode

In this mode, the AUSART uses standard Non-Return-to-Zero (NRZ) format (one Start bit, eight or nine data bits and one Stop bit). The most common data format is 8 bits. An on-chip, dedicated, 8-bit Baud Rate Generator can be used to derive standard baud rate frequencies from the oscillator. The AUSART transmits and receives the LSb first. The transmitter and receiver are functionally independent, but use the same data format and baud rate. The Baud Rate Generator produces a clock, either x16 or x64 of the bit shift rate, depending on bit BRGH (TXSTA<2>). Parity is not supported by the hardware, but can be implemented in software (and stored as the ninth data bit). Asynchronous mode is stopped during Sleep.

Asynchronous mode is selected by clearing bit SYNC (TXSTA<4>).

The AUSART Asynchronous module consists of the following important elements:

- Baud Rate Generator
- Sampling Circuit
- Asynchronous Transmitter
- Asynchronous Receiver

### 11.2.1 AUSART ASYNCHRONOUS TRANSMITTER

The AUSART transmitter block diagram is shown in Figure 11-1. The heart of the transmitter is the Transmit (Serial) Shift Register (TSR). The Shift register obtains its data from the Read/Write Transmit Buffer register, TXREG. The TXREG register is loaded with data in software. The TSR register is not loaded until the Stop bit has been transmitted from the previous load. As soon as the Stop bit is transmitted, the TSR is loaded with new data from the TXREG register (if available). Once the TXREG register transfers the data to the TSR register (occurs in one Tcy), the TXREG register is empty and flag bit, TXIF (PIR1<4>), is set. This

interrupt can be enabled/disabled by setting/clearing enable bit, TXIE (PIE1<4>). Flag bit TXIF will be set, regardless of the state of enable bit TXIE and cannot be cleared in software. It will reset only when new data is loaded into the TXREG register. While flag bit TXIF indicates the status of the TXREG register, another bit, TRMT (TXSTA<1>), shows the status of the TSR register. Status bit TRMT is a read-only bit which is set when the TSR register is empty. No interrupt logic is tied to this bit, so the user has to poll this bit in order to determine if the TSR register is empty.

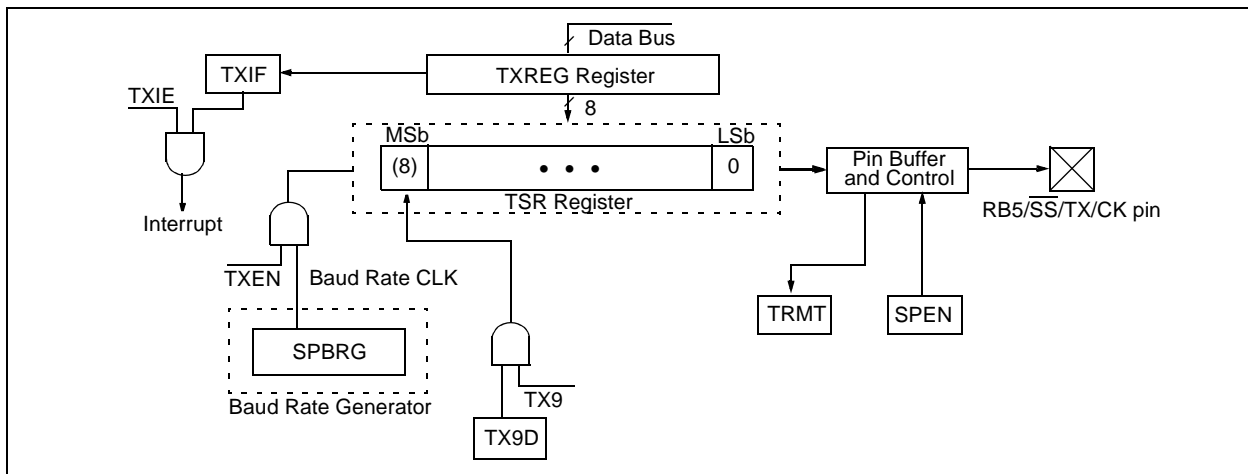
**Note 1:** The TSR register is not mapped in data memory, so it is not available to the user.

**2:** Flag bit TXIF is set when enable bit TXEN is set. TXIF is cleared by loading TXREG.

Transmission is enabled by setting enable bit TXEN (TXSTA<5>). The actual transmission will not occur until the TXREG register has been loaded with data and the Baud Rate Generator (BRG) has produced a shift clock (Figure 11-2). The transmission can also be started by first loading the TXREG register and then setting enable bit TXEN. Normally, when transmission is first started, the TSR register is empty. At that point, transfer to the TXREG register will result in an immediate transfer to TSR, resulting in an empty TXREG. A back-to-back transfer is thus possible (Figure 11-3). Clearing enable bit TXEN during a transmission will cause the transmission to be aborted and will reset the transmitter. As a result, the RB5/SS/TX/CK pin will revert to high-impedance.

In order to select 9-bit transmission, transmit bit, TX9 (TXSTA<6>), should be set and the ninth bit should be written to TX9D (TXSTA<0>). The ninth bit must be written before writing the 8-bit data to the TXREG register. This is because a data write to the TXREG register can result in an immediate transfer of the data to the TSR register (if the TSR is empty). In such a case, an incorrect ninth data bit may be loaded in the TSR register.

FIGURE 11-1: AUSART TRANSMIT BLOCK DIAGRAM

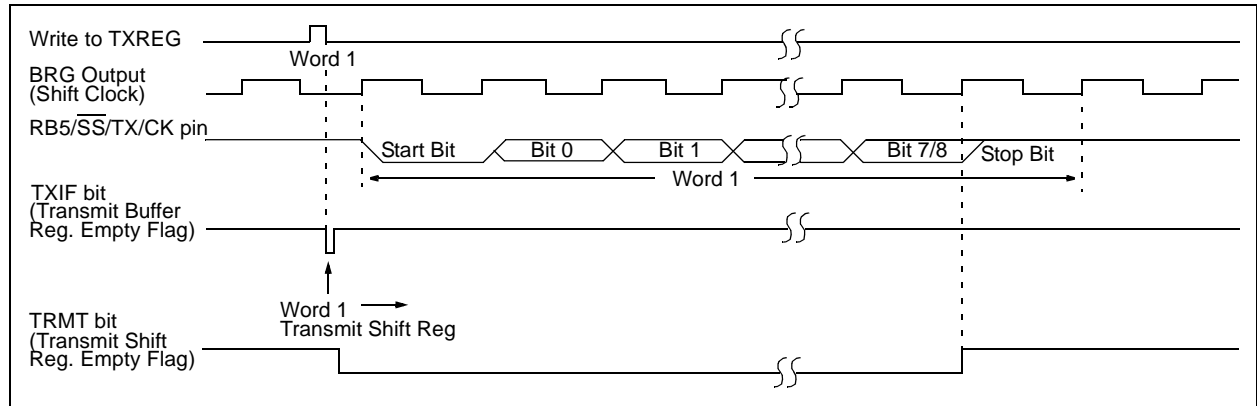




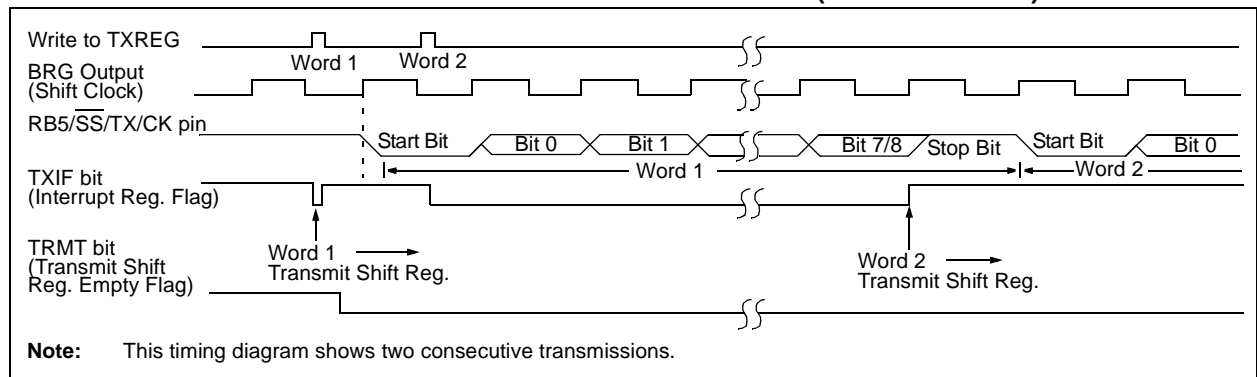
When setting up an asynchronous transmission, follow these steps:

1. Initialize the SPBRG register for the appropriate baud rate. If a high-speed baud rate is desired, set bit BRGH (Section 11.1 “AUSART Baud Rate Generator (BRG)”).
2. Enable the asynchronous serial port by clearing bit SYNC and setting bit SPEN.
3. If interrupts are desired, then set enable bit TXIE.
4. If 9-bit transmission is desired, then set transmit bit TX9.
5. Enable the transmission by setting bit TXEN which will also set bit TXIF.
6. If 9-bit transmission is selected, the ninth bit should be loaded in bit TX9D.
7. Load data to the TXREG register (starts transmission).
8. If using interrupts, ensure that GIE and PEIE (bits 7 and 6) of the INTCON register are set.

**FIGURE 11-2: ASYNCHRONOUS MASTER TRANSMISSION**



**FIGURE 11-3: ASYNCHRONOUS MASTER TRANSMISSION (BACK TO BACK)**



**TABLE 11-7: REGISTERS ASSOCIATED WITH ASYNCHRONOUS TRANSMISSION**

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	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u
0Ch	PIR1	—	ADIF <sup>(1)</sup>	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	-000 0000	-000 0000
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
19h	TXREG	AUSART Transmit Data Register								0000 0000	0000 0000
8Ch	PIE1	—	ADIE <sup>(1)</sup>	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	-000 0000	-000 0000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

**Legend:** x = unknown, - = unimplemented locations read as '0'. Shaded cells are not used for asynchronous transmission.  
**Note 1:** This bit is only implemented on the PIC16F88. The bit will read '0' on the PIC16F87.

# PIC16F87/88

## 11.2.2 AUSART ASYNCHRONOUS RECEIVER

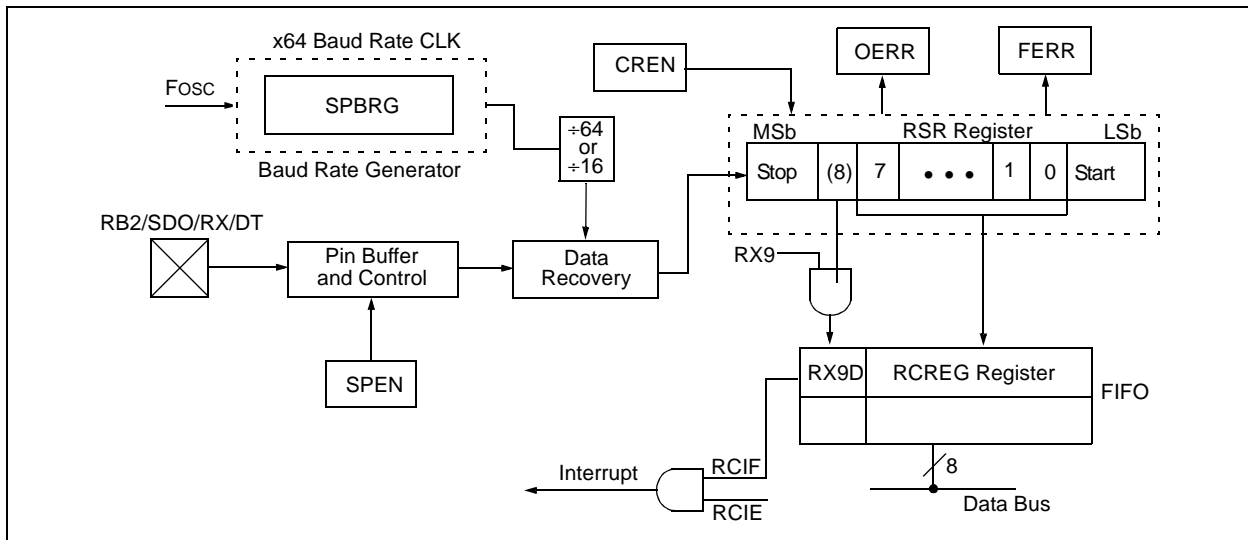
The receiver block diagram is shown in Figure 11-4. The data is received on the RB2/SDO/RX/DT pin and drives the data recovery block. The data recovery block is actually a high-speed shifter, operating at x16 times the baud rate; whereas, the main receive serial shifter operates at the bit rate or at Fosc.

Once Asynchronous mode is selected, reception is enabled by setting bit CREN (RCSTA<4>).

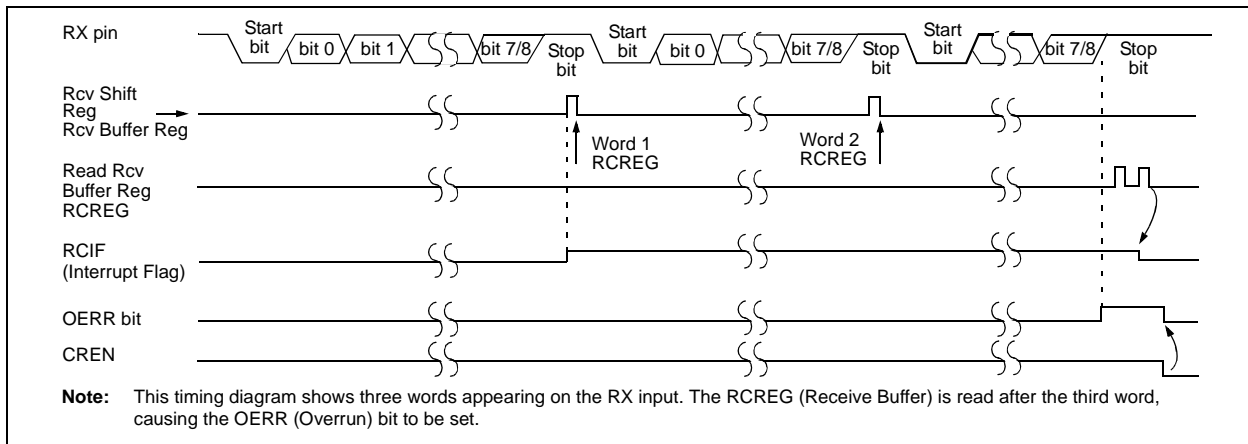
The heart of the receiver is the Receive (Serial) Shift Register (RSR). After sampling the Stop bit, the received data in the RSR is transferred to the RCREG register (if it is empty). If the transfer is complete, flag bit, RCIF (PIR1<5>), is set. The actual interrupt can be enabled/disabled by setting/clearing enable bit RCIE (PIE1<5>). Flag bit RCIF is a read-only bit which is cleared by the hardware. It is cleared when the RCREG register has been read and is empty. The RCREG is a double-buffered register (i.e., it is a two-deep FIFO). It

is possible for two bytes of data to be received and transferred to the RCREG FIFO and a third byte to begin shifting to the RSR register. On the detection of the Stop bit of the third byte, if the RCREG register is still full, the Overrun Error bit, OERR (RCSTA<1>), will be set. The word in the RSR will be lost. The RCREG register can be read twice to retrieve the two bytes in the FIFO. Overrun bit OERR has to be cleared in software. This is done by resetting the receive logic (CREN is cleared and then set). If bit OERR is set, transfers from the RSR register to the RCREG register are inhibited and no further data will be received. It is, therefore, essential to clear error bit OERR if it is set. Framing Error bit, FERR (RCSTA<2>), is set if a Stop bit is detected as clear. Bit FERR and the 9th receive bit are buffered the same way as the receive data. Reading the RCREG will load bits RX9D and FERR with new values; therefore, it is essential for the user to read the RCSTA register, before reading the RCREG register, in order not to lose the old FERR and RX9D information.

**FIGURE 11-4: AUSART RECEIVE BLOCK DIAGRAM**



**FIGURE 11-5: ASYNCHRONOUS RECEPTION**



When setting up an asynchronous reception, follow these steps:

1. Initialize the SPBRG register for the appropriate baud rate. If a high-speed baud rate is desired, set bit BRGH (Section 11.1 “AUSART Baud Rate Generator (BRG)”).
2. Enable the asynchronous serial port by clearing bit SYNC and setting bit SPEN.
3. If interrupts are desired, then set enable bit RCIE.
4. If 9-bit reception is desired, then set bit RX9.
5. Enable the reception by setting bit CREN.
6. Flag bit RCIF will be set when reception is complete and an interrupt will be generated if enable bit RCIE is set.
7. Read the RCSTA register to get the ninth bit (if enabled) and determine if any error occurred during reception.
8. Read the 8-bit received data by reading the RCREG register.
9. If any error occurred, clear the error by clearing enable bit CREN.
10. If using interrupts, ensure that GIE and PEIE (bits 7 and 6) of the INTCON register are set.

**TABLE 11-8: REGISTERS ASSOCIATED WITH ASYNCHRONOUS RECEPTION**

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	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u
0Ch	PIR1	—	ADIF <sup>(1)</sup>	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	-000 0000	-000 0000
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
1Ah	RCREG	AUSART Receive Data Register								0000 0000	0000 0000
8Ch	PIE1	—	ADIE <sup>(1)</sup>	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	-000 0000	-000 0000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

**Legend:** x = unknown, - = unimplemented locations read as '0'. Shaded cells are not used for asynchronous reception.

**Note 1:** This bit is only implemented on the PIC16F88. The bit will read '0' on the PIC16F87.

# PIC16F87/88

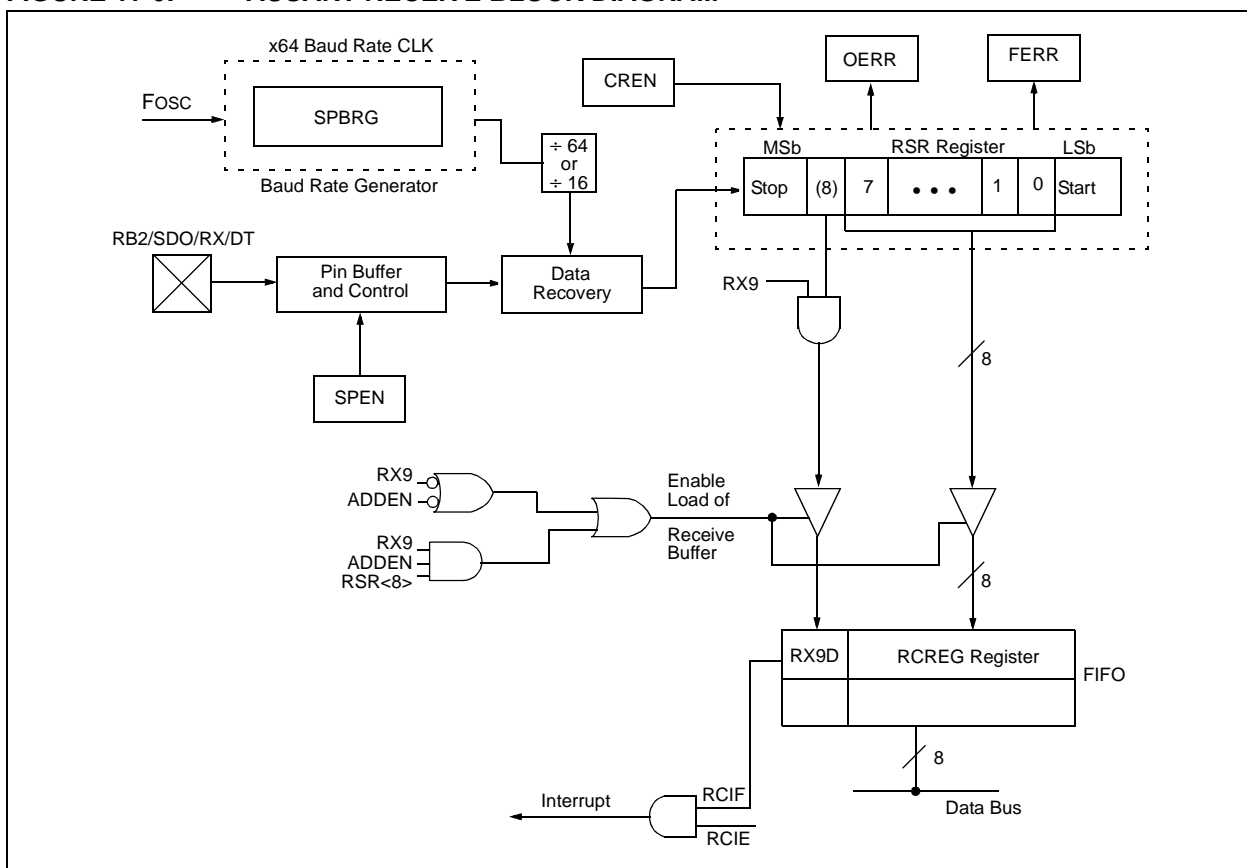
## 11.2.3 SETTING UP 9-BIT MODE WITH ADDRESS DETECT

When setting up an asynchronous reception with address detect enabled:

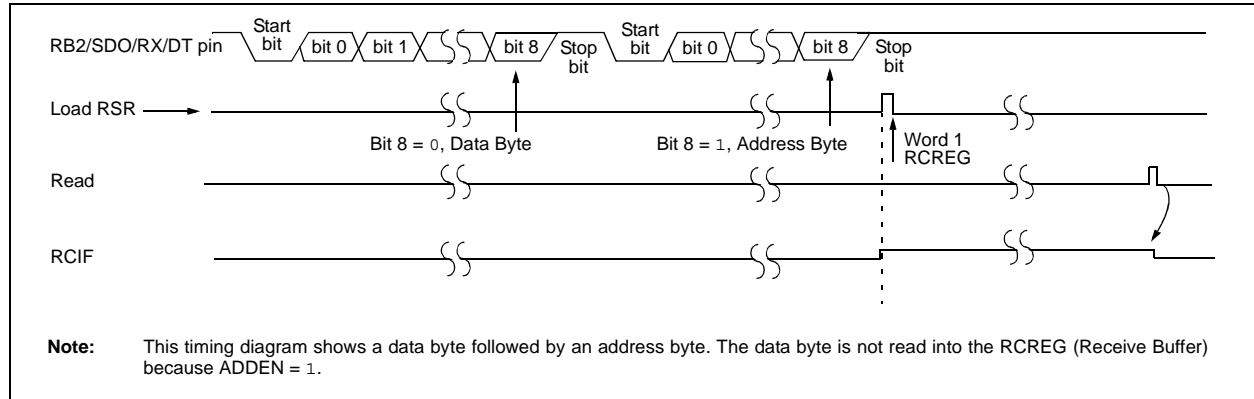
- Initialize the SPBRG register for the appropriate baud rate. If a high-speed baud rate is desired, set bit BRGH.
- Enable the asynchronous serial port by clearing bit SYNC and setting bit SPEN.
- If interrupts are desired, then set enable bit RCIE.
- Set bit RX9 to enable 9-bit reception.
- Set ADDEN to enable address detect.
- Enable the reception by setting enable bit CREN.

- Flag bit RCIF will be set when reception is complete and an interrupt will be generated if enable bit RCIE was set.
- Read the RCSTA register to get the ninth bit and determine if any error occurred during reception.
- Read the 8-bit received data by reading the RCREG register to determine if the device is being addressed.
- If any error occurred, clear the error by clearing enable bit CREN.
- If the device has been addressed, clear the ADDEN bit to allow data bytes and address bytes to be read into the receive buffer and interrupt the CPU.

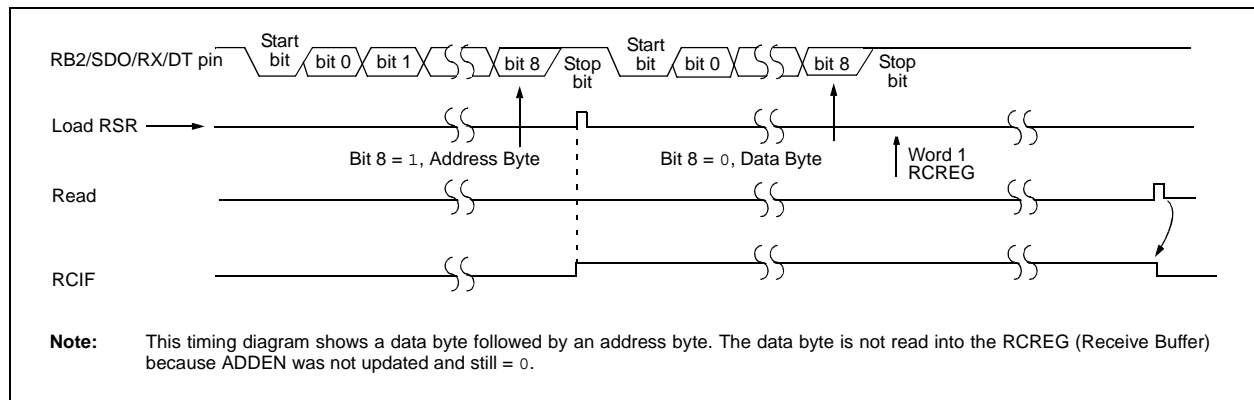
**FIGURE 11-6: AUSART RECEIVE BLOCK DIAGRAM**



**FIGURE 11-7: ASYNCHRONOUS RECEPTION WITH ADDRESS DETECT**



**FIGURE 11-8: ASYNCHRONOUS RECEPTION WITH ADDRESS BYTE FIRST**



**TABLE 11-9: REGISTERS ASSOCIATED WITH ASYNCHRONOUS RECEPTION**

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	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u
0Ch	PIR1	—	ADIF <sup>(1)</sup>	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	-000 0000	-000 0000
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
1Ah	RCREG	AUSART Receive Data Register								0000 0000	0000 0000
8Ch	PIE1	—	ADIE <sup>(1)</sup>	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	-000 0000	-000 0000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

**Legend:** x = unknown, - = unimplemented locations read as '0'. Shaded cells are not used for asynchronous reception.

**Note 1:** This bit is only implemented on the PIC16F88. The bit will read '0' on the PIC16F87.

## 11.3 AUSART Synchronous Master Mode

In Synchronous Master mode, the data is transmitted in a half-duplex manner (i.e., transmission and reception do not occur at the same time). When transmitting data, the reception is inhibited and vice versa. Synchronous mode is entered by setting bit SYNC (TXSTA<4>). In addition, enable bit SPEN (RCSTA<7>) is set in order to configure the RB5/SS/TX/CK and RB2/SDO/RX/DT I/O pins to CK (clock) and DT (data) lines, respectively. The Master mode indicates that the processor transmits the master clock on the CK line. The Master mode is entered by setting bit CSRC (TXSTA<7>).

### 11.3.1 AUSART SYNCHRONOUS MASTER TRANSMISSION

The AUSART transmitter block diagram is shown in Figure 11-6. The heart of the transmitter is the Transmit (Serial) Shift Register (TSR). The Shift register obtains its data from the Read/Write Transmit Buffer register, TXREG. The TXREG register is loaded with data in software. The TSR register is not loaded until the last bit has been transmitted from the previous load. As soon as the last bit is transmitted, the TSR is loaded with new data from the TXREG (if available). Once the TXREG register transfers the data to the TSR register (occurs in one T<sub>CYCLE</sub>), the TXREG is empty and interrupt bit TXIF (PIR1<4>) is set. The interrupt can be enabled/disabled by setting/clearing enable bit TXIE (PIE1<4>). Flag bit TXIF will be set, regardless of the state of enable bit TXIE and cannot be cleared in software. It will reset only when new data is loaded into the TXREG register. While flag bit TXIF indicates the status of the TXREG register, another bit, TRMT (TXSTA<1>), shows the status of the TSR register. TRMT is a read-only bit which is set when the TSR is empty. No interrupt logic is tied to this bit, so the user has to poll this bit in order to determine if the TSR register is empty. The TSR is not mapped in data memory, so it is not available to the user.

Transmission is enabled by setting enable bit TXEN (TXSTA<5>). The actual transmission will not occur until the TXREG register has been loaded with data. The first data bit will be shifted out on the next available rising edge of the clock on the CK line. Data out is stable around the falling edge of the synchronous clock (Figure 11-9). The transmission can also be started by first loading the TXREG register and then setting bit TXEN (Figure 11-10). This is advantageous when slow baud rates are selected, since the BRG is kept in Reset when bits TXEN, CREN and SREN are clear. Setting enable bit TXEN will start the BRG, creating a shift clock immediately. Normally, when transmission is first started, the TSR register is empty, so a transfer to the TXREG register will result in an immediate transfer to TSR, resulting in an empty TXREG. Back-to-back transfers are possible.

Clearing enable bit TXEN during a transmission will cause the transmission to be aborted and will reset the transmitter. The DT and CK pins will revert to high-impedance. If either bit CREN or bit SREN is set during a transmission, the transmission is aborted and the DT pin reverts to a high-impedance state (for a reception). The CK pin will remain an output if bit CSRC is set (internal clock). The transmitter logic, however, is not reset, although it is disconnected from the pins. In order to reset the transmitter, the user has to clear bit TXEN. If bit SREN is set (to interrupt an on-going transmission and receive a single word), then after the single word is received, bit SREN will be cleared and the serial port will revert back to transmitting, since bit TXEN is still set. The DT line will immediately switch from High-Impedance Receive mode to transmit and start driving. To avoid this, bit TXEN should be cleared.

In order to select 9-bit transmission, the TX9 (TXSTA<6>) bit should be set and the ninth bit should be written to bit TX9D (TXSTA<0>). The ninth bit must be written before writing the 8-bit data to the TXREG register. This is because a data write to the TXREG can result in an immediate transfer of the data to the TSR register (if the TSR is empty). If the TSR was empty and the TXREG was written before writing the “new” TX9D, the “present” value of bit TX9D is loaded.

Steps to follow when setting up a synchronous master transmission:

1. Initialize the SPBRG register for the appropriate baud rate (**Section 11.1 “AUSART Baud Rate Generator (BRG)”**).
2. Enable the synchronous master serial port by setting bits SYNC, SPEN and CSRC.
3. If interrupts are desired, set enable bit TXIE.
4. If 9-bit transmission is desired, set bit TX9.
5. Enable the transmission by setting bit TXEN.
6. If 9-bit transmission is selected, the ninth bit should be loaded in bit TX9D.
7. Start transmission by loading data to the TXREG register.
8. If using interrupts, ensure that GIE and PEIE (bits 7 and 6) of the INTCON register are set.

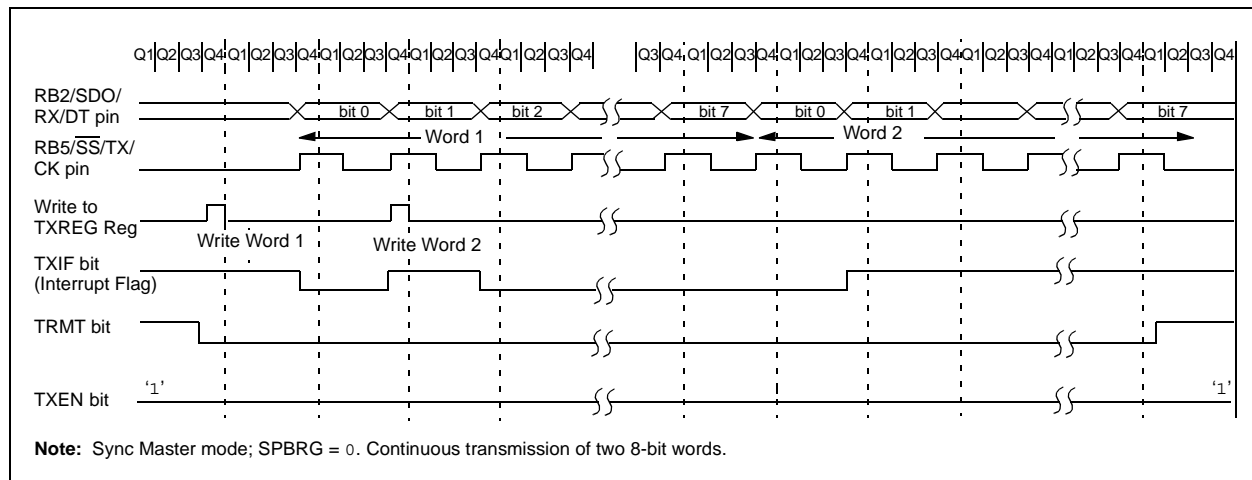
**TABLE 11-10: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER TRANSMISSION**

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	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u
0Ch	PIR1	—	ADIF <sup>(1)</sup>	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	-000 0000	-000 0000
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
19h	TXREG	AUSART Transmit Data Register								0000 0000	0000 0000
8Ch	PIE1	—	ADIE <sup>(1)</sup>	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	-000 0000	-000 0000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

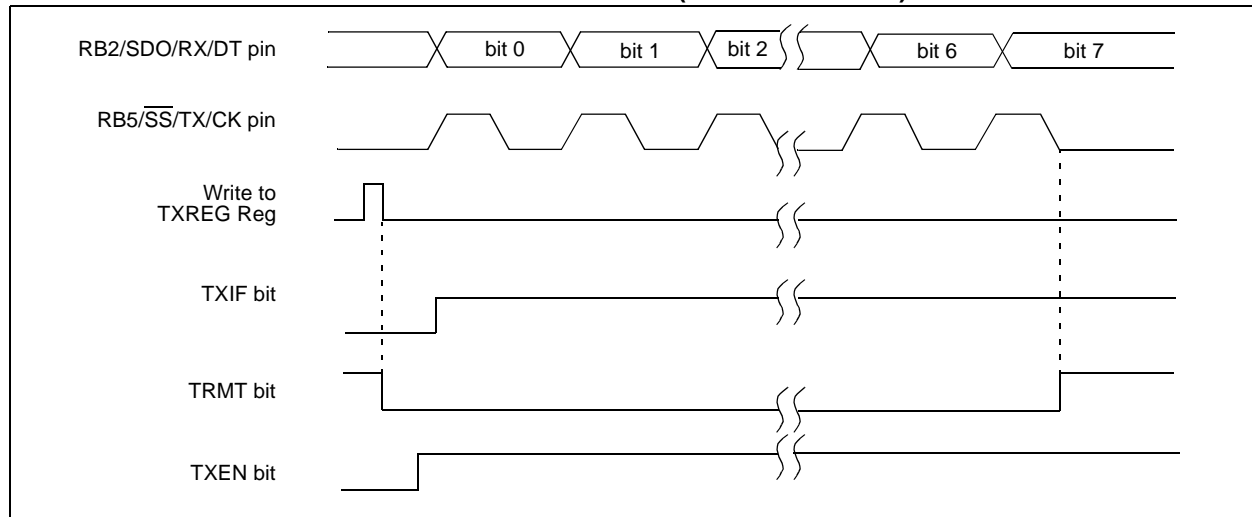
**Legend:** x = unknown, - = unimplemented, read as '0'. Shaded cells are not used for synchronous master transmission.

**Note 1:** This bit is only implemented on the PIC16F88. The bit will read '0' on the PIC16F87.

**FIGURE 11-9: SYNCHRONOUS TRANSMISSION**



**FIGURE 11-10: SYNCHRONOUS TRANSMISSION (THROUGH TXEN)**



# PIC16F87/88

## 11.3.2 AUSART SYNCHRONOUS MASTER RECEPTION

Once Synchronous mode is selected, reception is enabled by setting either enable bit SREN (RCSTA<5>), or enable bit CREN (RCSTA<4>). Data is sampled on the RB2/SDO/RX/DT pin on the falling edge of the clock. If enable bit SREN is set, then only a single word is received. If enable bit CREN is set, the reception is continuous until CREN is cleared. If both bits are set, CREN takes precedence.

After clocking the last bit, the received data in the Receive Shift Register (RSR) is transferred to the RCREG register (if it is empty). When the transfer is complete, interrupt flag bit, RCIF (PIR1<5>), is set. The actual interrupt can be enabled/disabled by setting/clearing enable bit RCIE (PIE1<5>).

Flag bit RCIF is a read-only bit which is reset by the hardware. In this case, it is reset when the RCREG register has been read and is empty. The RCREG is a double-buffered register (i.e., it is a two-deep FIFO). It is possible for two bytes of data to be received and transferred to the RCREG FIFO and a third byte to begin shifting into the RSR register. On the clocking of the last bit of the third byte, if the RCREG register is still full, then Overrun Error bit, OERR (RCSTA<1>), is set. The word in the RSR will be lost. The RCREG register can be read twice to retrieve the two bytes in the FIFO. Bit OERR has to be cleared in software (by clearing bit CREN). If bit OERR is set, transfers from the RSR to the RCREG are inhibited, so it is essential to clear bit OERR if it is set. The ninth receive bit is buffered the same way as the

receive data. Reading the RCREG register will load bit RX9D with a new value, therefore, it is essential for the user to read the RCSTA register, before reading RCREG, in order not to lose the old RX9D information.

When setting up a synchronous master reception:

1. Initialize the SPBRG register for the appropriate baud rate (**Section 11.1 “AUSART Baud Rate Generator (BRG)”**).
2. Enable the synchronous master serial port by setting bits SYNC, SPEN and CSRC.
3. Ensure bits CREN and SREN are clear.
4. If interrupts are desired, then set enable bit RCIE.
5. If 9-bit reception is desired, then set bit RX9.
6. If a single reception is required, set bit SREN. For continuous reception, set bit CREN.
7. Interrupt flag bit, RCIF, will be set when reception is complete and an interrupt will be generated if enable bit, RCIE, was set.
8. Read the RCSTA register to get the ninth bit (if enabled) and determine if any error occurred during reception.
9. Read the 8-bit received data by reading the RCREG register.
10. If any error occurred, clear the error by clearing bit CREN.
11. If using interrupts, ensure that GIE and PEIE (bits 7 and 6) of the INTCON register are set.

**TABLE 11-11: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER RECEPTION**

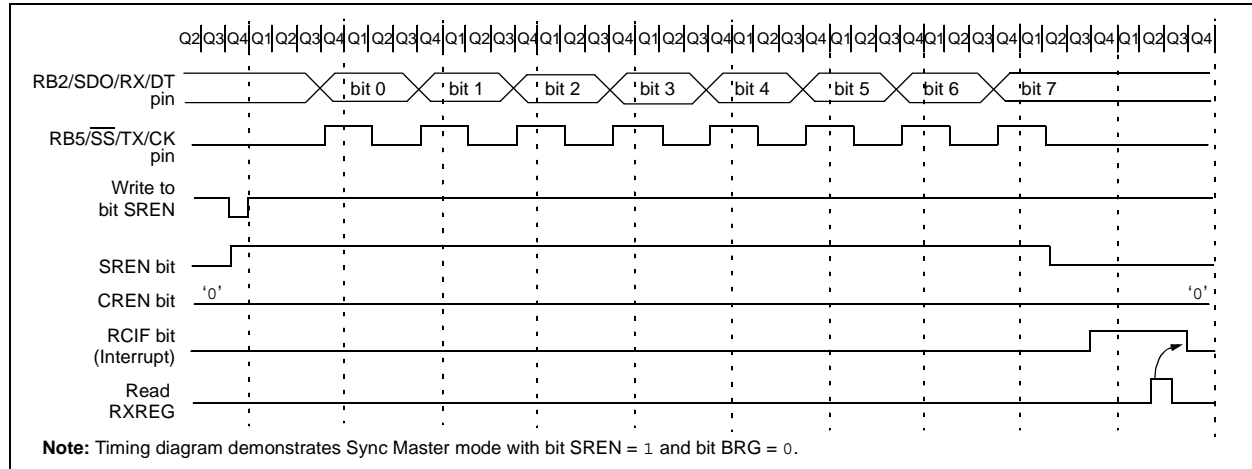
Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other Resets
0Bh, 8Bh, 10Bh, 18Bh	INTCON	GIE	PEIE	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u
0Ch	PIR1	—	ADIF <sup>(1)</sup>	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	-000 0000	-000 0000
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
1Ah	RCREG	AUSART Receive Data Register								0000 0000	0000 0000
8Ch	PIE1	—	ADIE <sup>(1)</sup>	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	-000 0000	-000 0000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

**Legend:** x = unknown, - = unimplemented, read as '0'. Shaded cells are not used for synchronous master reception.

**Note 1:** This bit is only implemented on the PIC16F88. The bit will read '0' on the PIC16F87.



**FIGURE 11-11: SYNCHRONOUS RECEPTION (MASTER MODE, SREN)**



## 11.4 AUSART Synchronous Slave Mode

Synchronous Slave mode differs from the Master mode in the fact that the shift clock is supplied externally at the RB5/SS/TX/CK pin (instead of being supplied internally in Master mode). This allows the device to transfer or receive data while in Sleep mode. Slave mode is entered by clearing bit CSRC (TXSTA<7>).

### 11.4.1 AUSART SYNCHRONOUS SLAVE TRANSMIT

The operation of the Synchronous Master and Slave modes is identical, except in the case of the Sleep mode.

If two words are written to the TXREG and then the SLEEP instruction is executed, the following will occur:

- The first word will immediately transfer to the TSR register and transmit.
- The second word will remain in the TXREG register.
- Flag bit TXIF will not be set.
- When the first word has been shifted out of TSR, the TXREG register will transfer the second word to the TSR and flag bit TXIF will now be set.

- If enable bit TXIE is set, the interrupt will wake the chip from Sleep and if the global interrupt is enabled, the program will branch to the interrupt vector (0004h).

When setting up a synchronous slave transmission, follow these steps:

- Enable the synchronous slave serial port by setting bits SYNC and SPEN and clearing bit CSRC.
- Clear bits CREN and SREN.
- If interrupts are desired, then set enable bit TXIE.
- If 9-bit transmission is desired, then set bit TX9.
- Enable the transmission by setting enable bit TXEN.
- If 9-bit transmission is selected, the ninth bit should be loaded in bit TX9D.
- Start transmission by loading data to the TXREG register.
- If using interrupts, ensure that GIE and PEIE (bits 7 and 6) of the INTCON register are set.

**TABLE 11-12: REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE TRANSMISSION**

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	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u
0Ch	PIR1	—	ADIF <sup>(1)</sup>	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	-000 0000	-000 0000
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
19h	TXREG	AUSART Transmit Data Register								0000 0000	0000 0000
8Ch	PIE1	—	ADIE <sup>(1)</sup>	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	-000 0000	-000 0000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

**Legend:** x = unknown, - = unimplemented, read as '0'. Shaded cells are not used for synchronous slave transmission.

**Note 1:** This bit is only implemented on the PIC16F88. The bit will read '0' on the PIC16F87.

# PIC16F87/88

## 11.4.2 AUSART SYNCHRONOUS SLAVE RECEPTION

The operation of the Synchronous Master and Slave modes is identical, except in the case of the Sleep mode. Bit SREN is a “don’t care” in Slave mode.

If receive is enabled by setting bit CREN prior to the SLEEP instruction, then a word may be received during Sleep. On completely receiving the word, the RSR register will transfer the data to the RCREG register and if enable bit RCIE bit is set, the interrupt generated will wake the chip from Sleep. If the global interrupt is enabled, the program will branch to the interrupt vector (0004h).

When setting up a synchronous slave reception, follow these steps:

1. Enable the synchronous master serial port by setting bits SYNC and SPEN and clearing bit CSRC.
2. If interrupts are desired, set enable bit RCIE.
3. If 9-bit reception is desired, set bit RX9.
4. To enable reception, set enable bit CREN.
5. Flag bit RCIF will be set when reception is complete and an interrupt will be generated if enable bit RCIE was set.
6. Read the RCSTA register to get the ninth bit (if enabled) and determine if any error occurred during reception.
7. Read the 8-bit received data by reading the RCREG register.
8. If any error occurred, clear the error by clearing bit CREN.
9. If using interrupts, ensure that GIE and PEIE (bits 7 and 6) of the INTCON register are set.

**TABLE 11-13: REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE RECEPTION**

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	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u
0Ch	PIR1	—	ADIF <sup>(1)</sup>	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	-000 0000	-000 0000
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
1Ah	RCREG	AUSART Receive Data Register								0000 0000	0000 0000
8Ch	PIE1	—	ADIE <sup>(1)</sup>	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	-000 0000	-000 0000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

**Legend:** x = unknown, - = unimplemented, read as ‘0’. Shaded cells are not used for synchronous slave reception.

**Note 1:** This bit is only implemented on the PIC16F88. The bit will read ‘0’ on the PIC16F87.

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

The Analog-to-Digital (A/D) converter module has seven inputs for 18/20 pin devices (PIC16F88 devices only).

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, VREF- (RA2) or VREF+ (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 five 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)
- Analog Select Register (ANSEL)

The ADCON0 register, shown in Register 12-2, controls the operation of the A/D module. The ANSEL register, shown in Register 12-1 and the ADCON1 register, shown in Register 12-3, configure the functions of the port pins. The port pins can be configured as analog inputs (RA3/RA2 can also be voltage references) or as digital I/O.

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

### REGISTER 12-1: ANSEL: ANALOG SELECT REGISTER (ADDRESS 9Bh) PIC16F88 DEVICES ONLY

U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0
bit 7							bit 0

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

bit 6-0 **ANS<6:0>:** Analog Input Select bits

Bits select input function on corresponding AN<6:0> pins.

1 = Analog I/O<sup>(1,2)</sup>

0 = Digital I/O

**Note 1:** Setting a pin to an analog input disables the digital input buffer. The corresponding TRIS bit should be set to input mode when using pins as analog inputs. Only AN2 is an analog I/O, all other ANx pins are analog inputs.

**2:** See the block diagrams for the analog I/O pins to see how ANSEL interacts with the CHS bits of the ADCON0 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

# PIC16F87/88

## REGISTER 12-2: **ADCON0: A/D CONTROL REGISTER (ADDRESS 1Fh) PIC16F88 DEVICES ONLY**

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/DONE	—	ADON
bit 7							bit 0

bit 7-6 **ADCS<1:0>**: 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 **CHS<2:0>**: 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)

101 = Channel 5 (RB6/AN5)

110 = Channel 6 (RB7/AN6)

bit 2 **GO/DONE**: A/D Conversion Status bit

If ADON = 1:

1 = A/D conversion in progress (setting this bit starts the A/D conversion)

0 = A/D conversion not in progress (this bit is automatically cleared by hardware when the A/D conversion is complete)

bit 1 **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

## REGISTER 12-3: ADCON1: A/D CONTROL REGISTER 1 (ADDRESS 9Fh) PIC16F88 DEVICES ONLY

R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0
ADFM	ADCS2	VCFG1	VCFG0	—	—	—	—
bit 7				bit 0			

bit 7 **ADFM:** A/D Result Format Select bit

1 = Right justified. Six Most Significant bits of ADRESH are read as '0'.  
 0 = Left justified. Six 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 **VCFG<1:0>:** A/D Voltage Reference Configuration bits

Logic State	VREF+	VREF-
00	AVDD	AVSS
01	AVDD	VREF-
10	VREF+	AVSS
11	VREF+	VREF-

**Note:** The ANSEL bits for AN3 and AN2 inputs must be configured as analog inputs for the VREF+ and VREF- external pins to be used.

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

# PIC16F87/88

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 12-1.

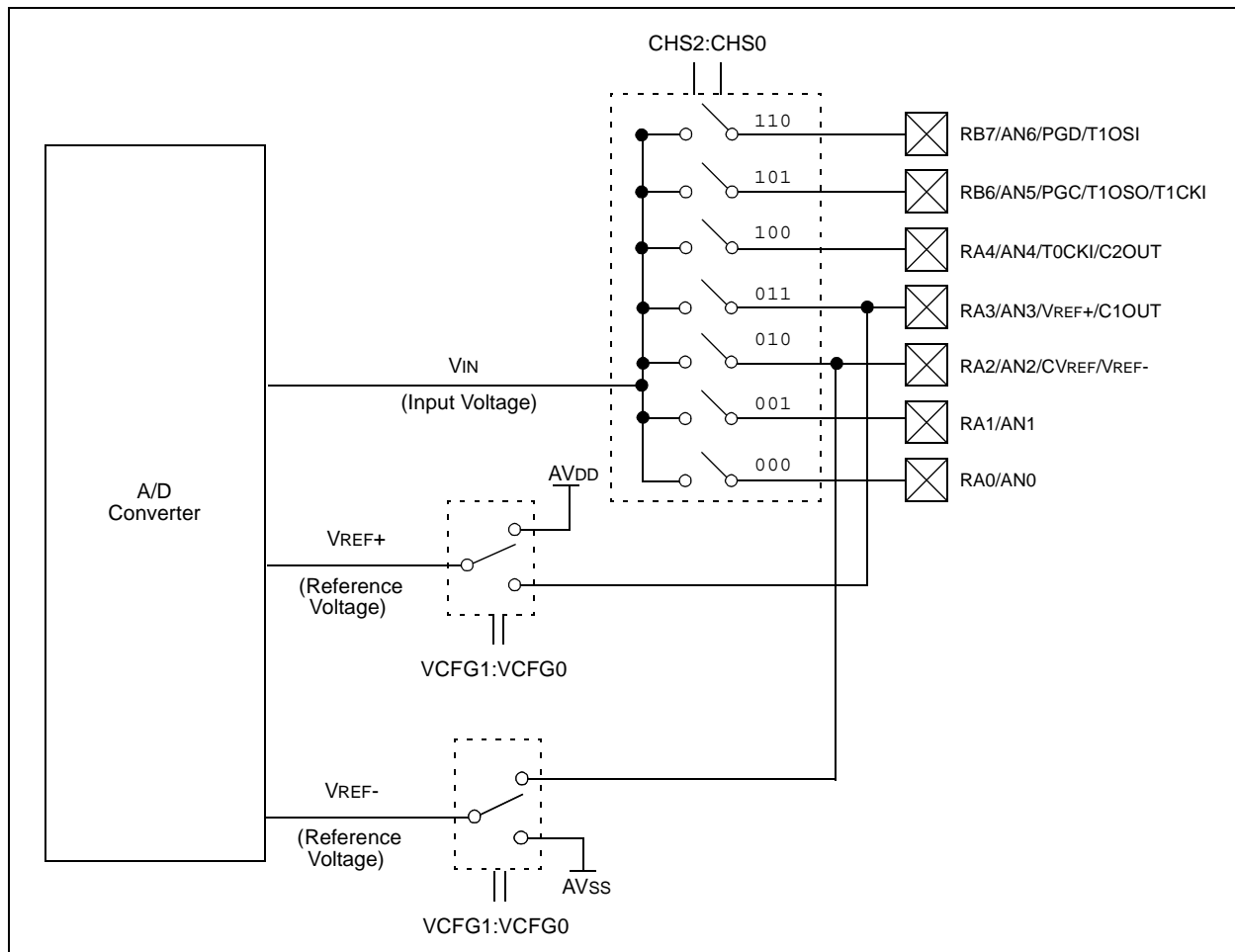
After the A/D module has been configured as desired, the selected channel must be acquired before the conversion is started. The analog input channels must have their corresponding TRIS bits selected as inputs.

To determine sample time, see **Section 12.1 “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/digital I/O (ANSEL)
  - Configure voltage reference (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 PEIE 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 12-1: A/D BLOCK DIAGRAM**



## 12.1 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 12-2. 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 12-2. **The maximum recommended impedance for analog sources is 10 kΩ.** As the impedance is decreased, the

acquisition time may be decreased. After the analog input channel is selected (changed), this acquisition must be done before the conversion can be started.

To calculate the minimum acquisition time, Equation 12-1 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 12-1: ACQUISITION TIME

$$\begin{aligned}
 TACQ &= \text{Amplifier Settling Time} + \text{Hold Capacitor Charging Time} + \text{Temperature Coefficient} \\
 &= TAMP + TC + TCOFF \\
 &= 2 \mu\text{s} + TC + [(\text{Temperature} - 25^\circ\text{C})(0.05 \mu\text{s}/^\circ\text{C})] \\
 TC &= CHOLD (RIC + RSS + RS) \ln(1/2047) \\
 &= -120 \text{ pF} (1 \text{ k}\Omega + 7 \text{ k}\Omega + 10 \text{ k}\Omega) \ln(0.0004885) \\
 &= 16.47 \mu\text{s} \\
 TACQ &= 2 \mu\text{s} + 16.47 \mu\text{s} + [(50^\circ\text{C} - 25^\circ\text{C})(0.05 \mu\text{s}/^\circ\text{C})] \\
 &= 19.72 \mu\text{s}
 \end{aligned}$$

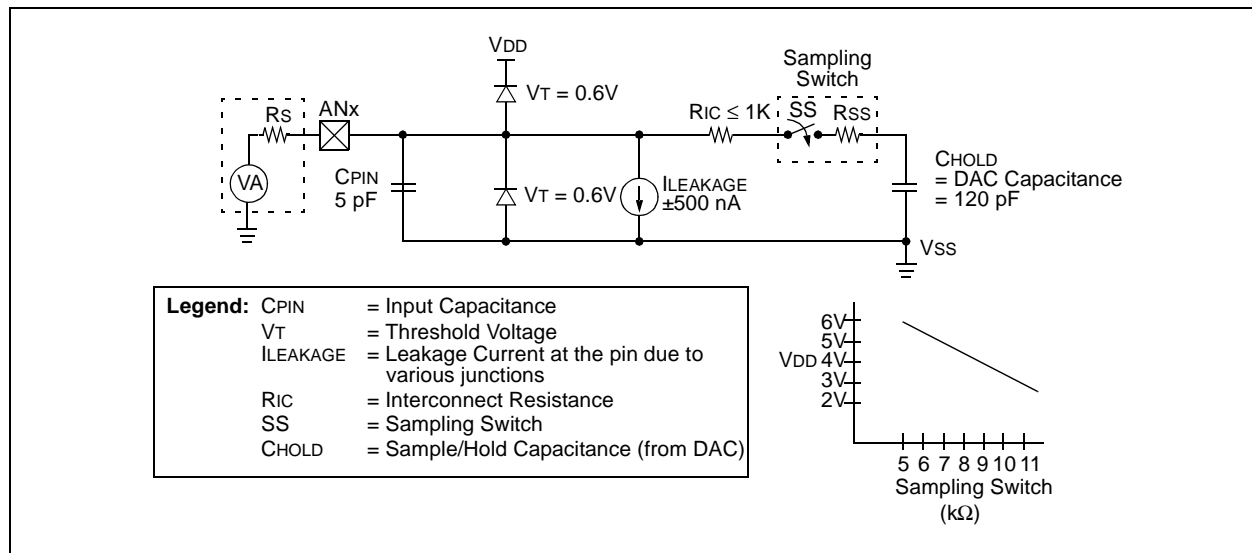
**Note 1:** The reference voltage (VREF) has no effect on the equation, since it cancels itself out.

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

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

**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 12-2: ANALOG INPUT MODEL**



# PIC16F87/88

## 12.2 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  $\mu$ s)

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  $\mu$ s and not greater than 6.4  $\mu$ s.

Table 12-1 shows the resultant TAD times derived from the device operating frequencies and the A/D clock source selected.

## 12.3 Operation in Power-Managed Modes

The selection of the automatic acquisition time and A/D conversion clock is determined in part by the clock source and frequency while in a power-managed mode.

If the A/D is expected to operate while the device is in a power-managed mode, the ADCS2:ADCS0 bits in ADCON0 and ADCON1 should be updated in accordance with the power-managed mode clock that will be used. After the power-managed mode is entered (either of the power-managed Run modes), an A/D acquisition or conversion may be started. Once an acquisition or conversion is started, the device should continue to be clocked by the same power-managed mode clock source until the conversion has been completed.

If the power-managed mode clock frequency is less than 1 MHz, the A/D RC clock source should be selected.

**TABLE 12-1: TAD vs. MAXIMUM DEVICE OPERATING FREQUENCIES – STANDARD DEVICES (C)**

AD Clock Source (TAD)			Maximum Device Frequency
Operation	ADCS<2>	ADCS<1:0>	Max.
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	<b>(Note 1)</b>

**Note 1:** The RC source has a typical TAD time of 4  $\mu$ s, but can vary between 2-6  $\mu$ s.

**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 18.0 “Electrical Characteristics”**.



## 12.4 Configuring Analog Port Pins

The ADCON1, ANSEL, TRISA and TRISB 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 RA4:RA0 and RB7:RB6 pins), may cause the input buffer to consume current out of the device specification.

## 12.5 A/D Conversions

Clearing the GO/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/DONE bit can then be set to start the conversion.

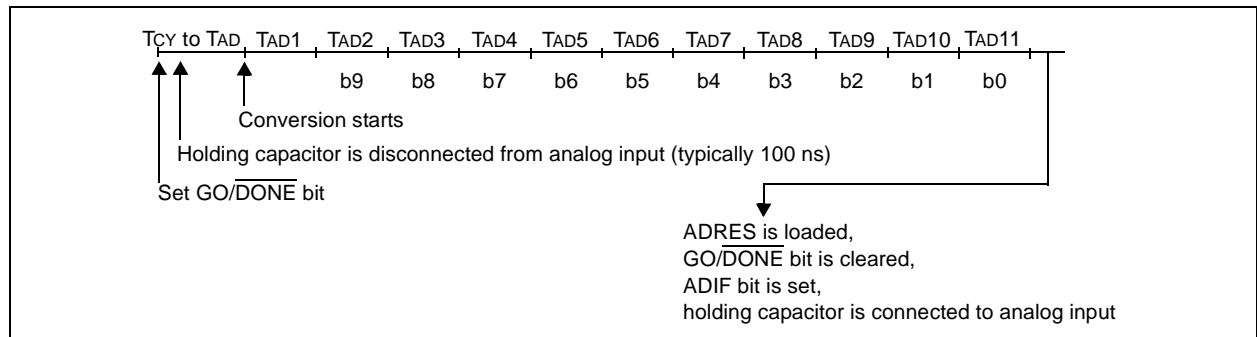
In Figure 12-3, after the GO/DONE bit is set, the first time segment has a minimum of T<sub>CY</sub> and a maximum of TAD.

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

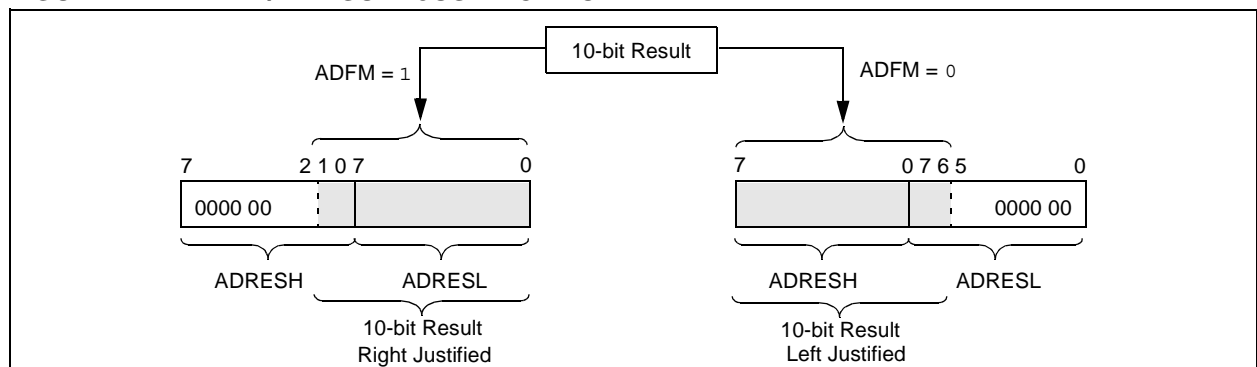
### 12.5.1 A/D RESULT REGISTERS

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 12-4 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 12-3: A/D CONVERSION TAD CYCLES**



**FIGURE 12-4: A/D RESULT JUSTIFICATION**



# PIC16F87/88

## 12.6 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 registers. 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.

## 12.7 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.

## 12.8 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 12-2: 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	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u
0Ch	PIR1	—	ADIF <sup>(1)</sup>	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	-000 0000	-000 0000
8Ch	PIE1	—	ADIE <sup>(1)</sup>	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	-000 0000	-000 0000
1Eh	ADRESH <sup>(2)</sup>	A/D Result Register High Byte								xxxx xxxx	uuuu uuuu
9Eh	ADRESL <sup>(2)</sup>	A/D Result Register Low Byte								xxxx xxxx	uuuu uuuu
1Fh	ADCON0 <sup>(2)</sup>	ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	—	ADON	0000 00-0	0000 00-0
9Fh	ADCON1 <sup>(2)</sup>	ADFM	ADCS2	VCFG1	VCFG0	—	—	—	—	0000 ----	0000 ----
9Bh	ANSEL <sup>(2)</sup>	—	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0	-111 1111	-111 1111
05h	PORTA (PIC16F87) (PIC16F88)	RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0	xxxx 0000 xxx0 0000	uuuu 0000 uuu0 0000
05h, 106h	PORTB (PIC16F87) (PIC16F88)	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx xxxx 00xx xxxx	uuuu uuuu 00uu uuuu
85h	TRISA	TRISA7	TRISA6	TRISA5 <sup>(3)</sup>	PORTA Data Direction Register (TRISA<4:0>)					1111 1111	1111 1111
86h, 186h	TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	1111 1111	1111 1111

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

**Note 1:** This bit is only implemented on the PIC16F88. The bit will read ‘0’ on the PIC16F87.

**2:** PIC16F88 only.

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

## 13.0 COMPARATOR MODULE

The comparator module contains two analog comparators. The inputs to the comparators are multiplexed with I/O port pins RA0 through RA3, while the outputs are multiplexed to pins RA3 and RA4. The on-chip Voltage Reference (**Section 14.0 “Comparator Voltage Reference Module”**) can also be an input to the comparators.

The CMCON register (Register 13-1) controls the comparator input and output multiplexors. A block diagram of the various comparator configurations is shown in Figure 13-1.

**REGISTER 13-1: CMCON: COMPARATOR MODULE CONTROL REGISTER (ADDRESS 9Ch)**

R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-1	R/W-1	R/W-1
C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0
bit 7							bit 0

- bit 7 **C2OUT:** Comparator 2 Output bit  
When C2INV = 0:  
 1 = C2 VIN+ > C2 VIN-  
 0 = C2 VIN+ < C2 VIN-  
When C2INV = 1:  
 1 = C2 VIN+ < C2 VIN-  
 0 = C2 VIN+ > C2 VIN-
- bit 6 **C1OUT:** Comparator 1 Output bit  
When C1INV = 0:  
 1 = C1 VIN+ > C1 VIN-  
 0 = C1 VIN+ < C1 VIN-  
When C1INV = 1:  
 1 = C1 VIN+ < C1 VIN-  
 0 = C1 VIN+ > C1 VIN-
- bit 5 **C2INV:** Comparator 2 Output Inversion bit  
 1 = C2 output inverted  
 0 = C2 output not inverted
- bit 4 **C1INV:** Comparator 1 Output Inversion bit  
 1 = C1 output inverted  
 0 = C1 output not inverted
- bit 3 **CIS:** Comparator Input Switch bit  
When CM2:CM0 = 001:  
 1 = C1 VIN- connects to RA3  
 0 = C1 VIN- connects to RA0  
When CM2:CM0 = 010:  
 1 = C1 VIN- connects to RA3  
     C2 VIN- connects to RA2  
 0 = C1 VIN- connects to RA0  
     C2 VIN- connects to RA1
- bit 2-0 **CM<2:0>:** Comparator Mode bits

**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

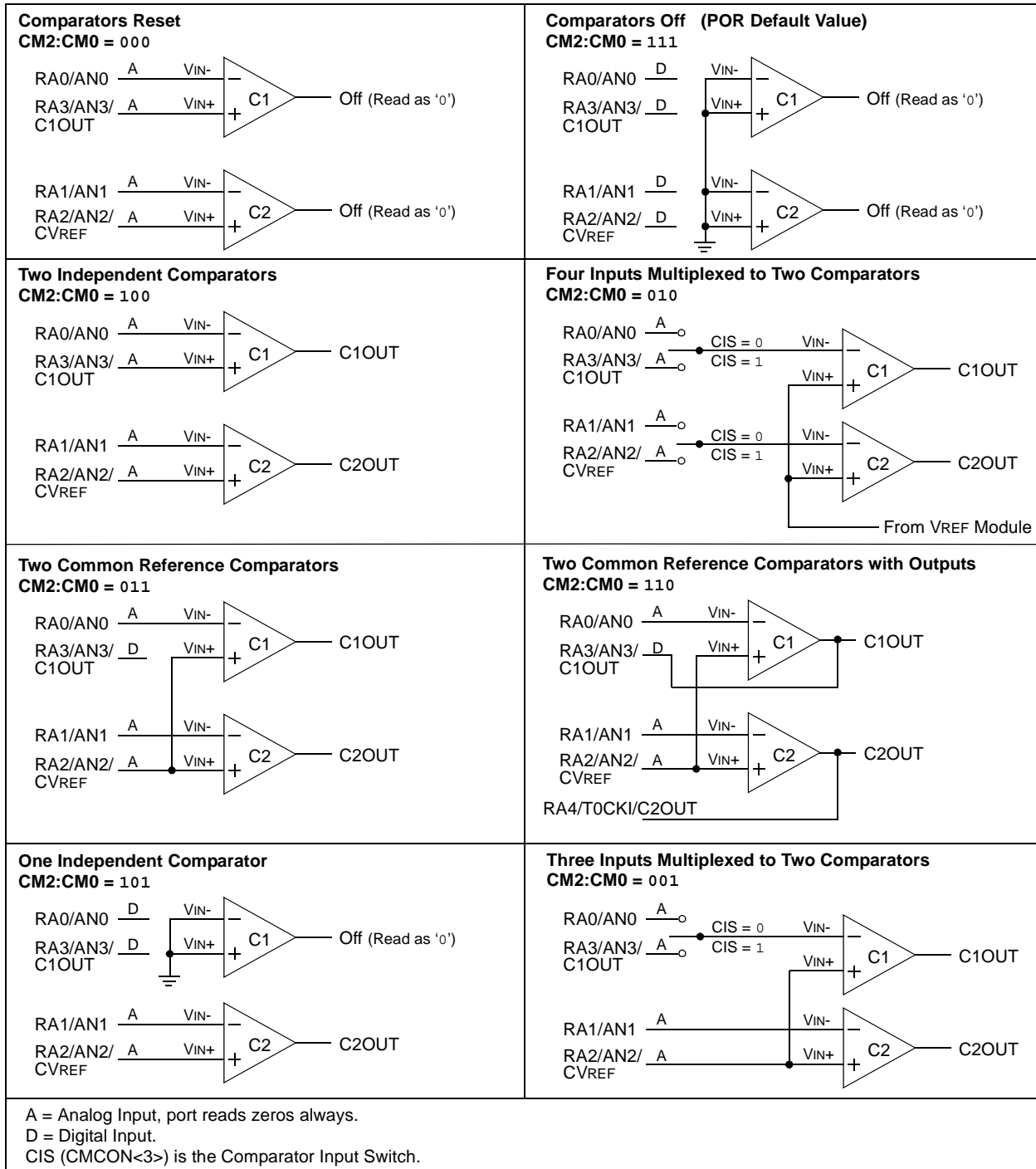
# PIC16F87/88

## 13.1 Comparator Configuration

There are eight modes of operation for the comparators. The CMCON register is used to select these modes. Figure 13-1 shows the eight possible modes. The TRISA register controls the data direction of the comparator pins for each mode. If the Comparator mode is changed, the comparator output level may not be valid for the specified mode change delay shown in Section 18.0 “Electrical Characteristics”.

**Note:** Comparator interrupts should be disabled during a Comparator mode change; otherwise, a false interrupt may occur.

**FIGURE 13-1: COMPARATOR I/O OPERATING MODES**



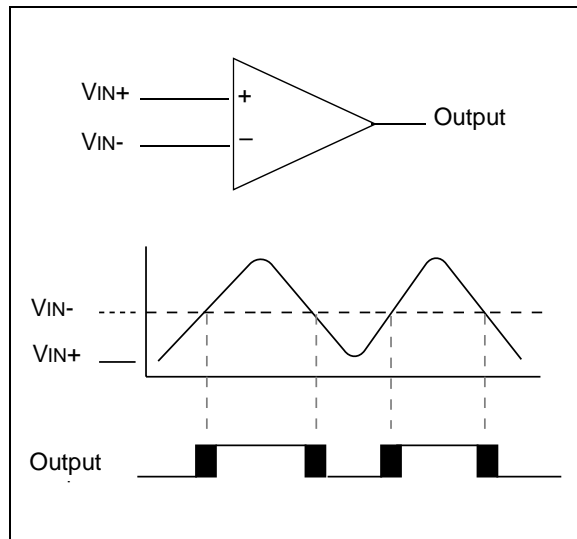
## 13.2 Comparator Operation

A single comparator is shown in Figure 13-2, along with the relationship between the analog input levels and the digital output. When the analog input at VIN+ is less than the analog input VIN-, the output of the comparator is a digital low level. When the analog input at VIN+ is greater than the analog input VIN-, the output of the comparator is a digital high level. The shaded areas of the output of the comparator in Figure 13-2 represent the uncertainty due to input offsets and response time.

## 13.3 Comparator Reference

An external or internal reference signal may be used depending on the comparator operating mode. The analog signal present at VIN- is compared to the signal at VIN+ and the digital output of the comparator is adjusted accordingly (Figure 13-2).

**FIGURE 13-2: SINGLE COMPARATOR**



### 13.3.1 EXTERNAL REFERENCE SIGNAL

When external voltage references are used, the comparator module can be configured to have the comparators operate from the same, or different reference sources. However, threshold detector applications may require the same reference. The reference signal must be between VSS and VDD and can be applied to either pin of the comparator(s).

### 13.3.2 INTERNAL REFERENCE SIGNAL

The comparator module also allows the selection of an internally generated voltage reference for the comparators. **Section 14.0 “Comparator Voltage Reference Module”** contains a detailed description of the Comparator Voltage Reference module that provides this signal. The internal reference signal is used when comparators are in mode CM<2:0> = 010 (Figure 13-1). In this mode, the internal voltage reference is applied to the VIN+ pin of both comparators.

## 13.4 Comparator Response Time

Response time is the minimum time, after selecting a new reference voltage or input source, before the comparator output has a valid level. If the internal reference is changed, the maximum delay of the internal voltage reference must be considered when using the comparator outputs. Otherwise, the maximum delay of the comparators should be used (**Section 18.0 “Electrical Characteristics”**).

## 13.5 Comparator Outputs

The comparator outputs are read through the CMCON register. These bits are read-only. The comparator outputs may also be directly output to the RA3 and RA4 I/O pins. When enabled, multiplexors in the output path of each pin will switch and the output of each pin will be the unsynchronized output of the comparator. The uncertainty of each of the comparators is related to the input offset voltage and the response time given in the specifications. Figure 13-3 shows the comparator output block diagram.

The TRISA bits will still function as an output enable/disable for the RA3 and RA4 pins while in this mode.

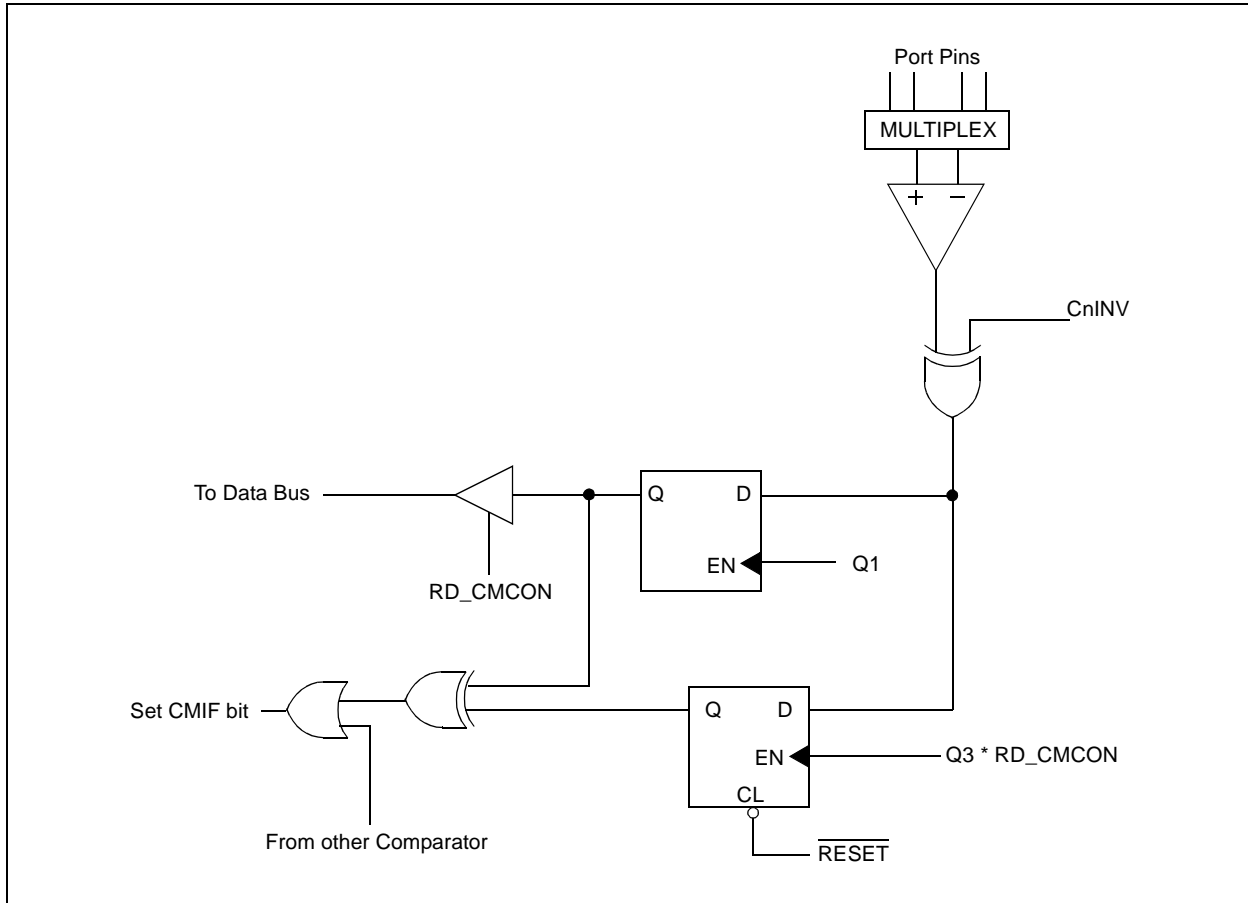
The polarity of the comparator outputs can be changed using the C2INV and C1INV bits (CMCON<5:4>).

**Note 1:** When reading the Port register, all pins configured as analog inputs will read as '0'. Pins configured as digital inputs will convert an analog input, according to the Schmitt Trigger input specification.

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

# PIC16F87/88

FIGURE 13-3: COMPARATOR OUTPUT BLOCK DIAGRAM



## 13.6 Comparator Interrupts

The comparator interrupt flag is set whenever there is a change in the output value of either comparator. Software will need to maintain information about the status of the output bits, as read from  $CMCON\langle 7:6 \rangle$ , to determine the actual change that occurred. The CMIF bit (PIR2 register) is the Comparator Interrupt Flag. The CMIF bit must be reset by clearing it ('0'). Since it is also possible to write a '1' to this register, a simulated interrupt may be initiated.

The CMIE bit (PIE2 register) and the PEIE bit (INTCON register) must be set to enable the interrupt. In addition, the GIE bit must also be set. If any of these bits are clear, the interrupt is not enabled, though the CMIF bit will still be set if an interrupt condition occurs.

**Note:** If a change in the CMCON register (C1OUT or C2OUT) should occur when a read operation is being executed (start of the Q2 cycle), then the CMIF (PIR2 register) interrupt flag may not get set.

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

- Any read or write of CMCON will end the mismatch condition.
- Clear flag bit CMIF.

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

## 13.7 Comparator Operation During Sleep

When a comparator is active and the device is placed in Sleep mode, the comparator remains active and the interrupt is functional, if enabled. This interrupt will wake-up the device from Sleep mode when enabled. While the comparator is powered up, higher Sleep currents than shown in the power-down current specification will occur. Each operational comparator will consume additional current, as shown in the comparator specifications. To minimize power consumption while in Sleep mode, turn off the comparators,  $CM<2:0> = 111$ , before entering Sleep. If the device wakes up from Sleep, the contents of the CMCON register are not affected.

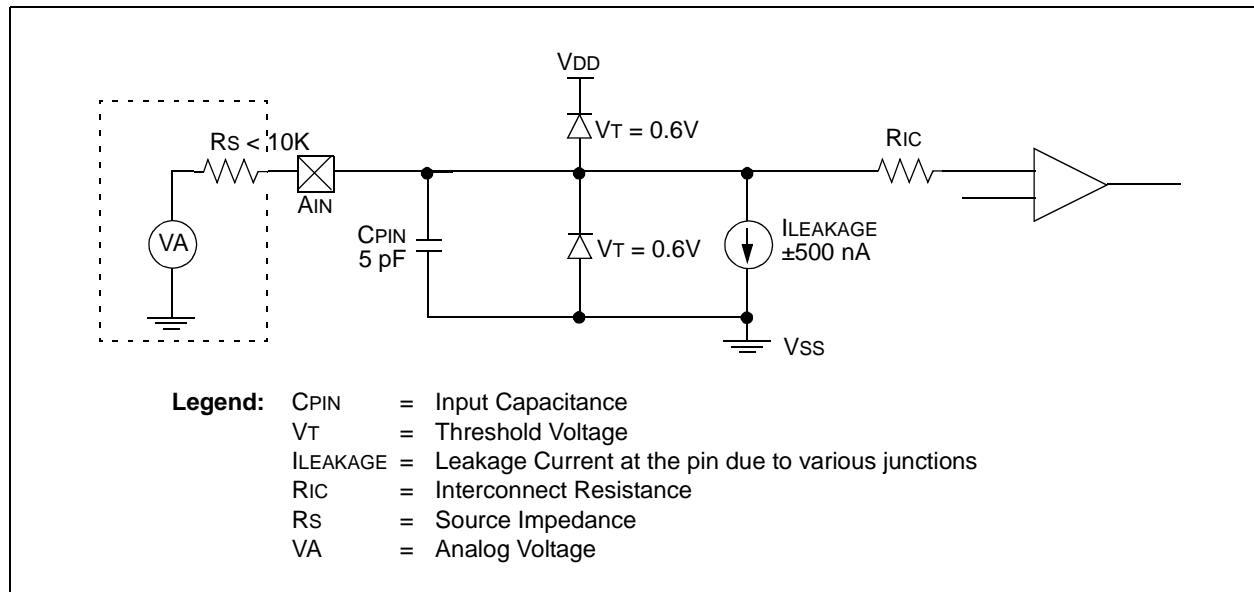
## 13.8 Effects of a Reset

A device Reset forces the CMCON register to its Reset state, causing the comparator module to be in the Comparator Off mode,  $CM<2:0> = 111$ .

## 13.9 Analog Input Connection Considerations

A simplified circuit for an analog input is shown in Figure 13-4. Since the analog pins are connected to a digital output, they have reverse biased diodes to VDD and VSS. The analog input, therefore, must be between VSS and VDD. If the input voltage deviates from this range by more than 0.6V in either direction, one of the diodes is forward biased and a latch-up condition may occur. A maximum source impedance of 10 k $\Omega$  is recommended for the analog sources. Any external component connected to an analog input pin, such as a capacitor or a Zener diode, should have very little leakage current.

**FIGURE 13-4: ANALOG INPUT MODEL**



# PIC16F87/88

**TABLE 13-1: REGISTERS ASSOCIATED WITH THE COMPARATOR MODULE**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR	Value on all other Resets
9Ch	CMCON	C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0	0000 0111	0000 0111
9Dh	CVRCON	CVREN	CVROE	CVRR	—	CVR3	CVR2	CVR1	CVR0	000- 0000	000- 0000
0Bh, 8Bh, 10Bh, 18Bh	INTCON	GIE	PEIE	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u
0Dh	PIR2	OSFIF	CMIF	—	EEIF	—	—	—	—	00-0 ----	00-0 ----
8Dh	PIE2	OSFIE	CMIE	—	EEIE	—	—	—	—	00-0 ----	00-0 ----
05h	PORTA (PIC16F87) (PIC16F88)	RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0	xxxx 0000 xxx0 0000	uuuu 0000 uuu0 0000
85h	TRISA	TRISA7	TRISA6	TRISA5 <sup>(1)</sup>	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	1111 1111	1111 1111

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

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



## 14.0 COMPARATOR VOLTAGE REFERENCE MODULE

The comparator voltage reference generator is a 16-tap resistor ladder network that provides a fixed voltage reference when the comparators are in mode '010'. A programmable register controls the function of the reference generator. Register 14-1 lists the bit functions of the CVRCON register.

As shown in Figure 14-1, the resistor ladder is segmented to provide two ranges of CVREF values and has a power-down function to conserve power when the reference is not being used. The comparator reference

supply voltage (also referred to as CVRSRC) comes directly from VDD. It should be noted, however, that the voltage at the top of the ladder is  $CVRSRC - VSAT$ , where VSAT is the saturation voltage of the power switch transistor. This reference will only be as accurate as the values of CVRSRC and VSAT.

The output of the reference generator may be connected to the RA2/AN2/CVREF/VREF- pin (VREF- is available on the PIC16F88 device only). This can be used as a simple D/A function by the user if a very high-impedance load is used. The primary purpose of this function is to provide a test path for testing the reference generator function.

### REGISTER 14-1: CVRCON: COMPARATOR VOLTAGE REFERENCE CONTROL REGISTER (ADDRESS 9Dh)

R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
CVREN	CVROE	CVRR	—	CVR3	CVR2	CVR1	CVR0
bit 7							bit 0

bit 7 **CVREN:** Comparator Voltage Reference Enable bit

1 = CVREF circuit powered on  
0 = CVREF circuit powered down

bit 6 **CVROE:** Comparator VREF Output Enable bit

1 = CVREF voltage level is output on the RA2/AN2/CVREF/VREF- pin<sup>(1)</sup>  
0 = CVREF voltage level is disconnected from the RA2/AN2/CVREF/VREF- pin<sup>(1)</sup>

bit 5 **CVRR:** Comparator VREF Range Selection bit<sup>(1)</sup>

1 = 0.00 CVRSRC to 0.625 CVRSRC with CVRSRC/24 step size  
0 = 0.25 CVRSRC to 0.72 CVRSRC with CVRSRC/32 step size

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

bit 3-0 **CVR<3:0>:** Comparator VREF Value Selection  $0 \leq VR3:VR0 \leq 15$  bits<sup>(1)</sup>

When CVRR = 1:

$$CVREF = (VR<3:0>/24) \cdot (CVRSRC)$$

When CVRR = 0:

$$CVREF = 1/4 \cdot (CVRSRC) + (VR3:VR0/32) \cdot (CVRSRC)$$

**Note 1:** VREF is available on the PIC16F88 device only.

#### Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

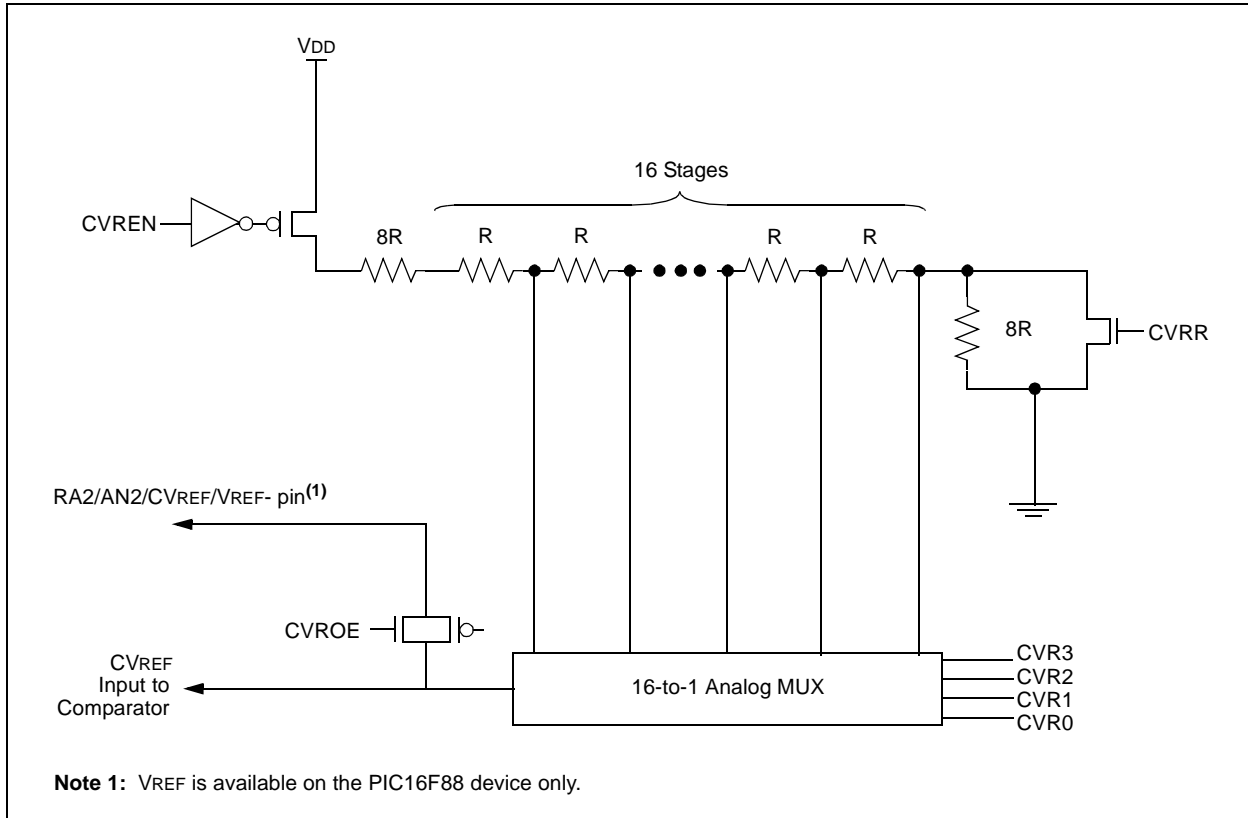
'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

# PIC16F87/88

**FIGURE 14-1: COMPARATOR VOLTAGE REFERENCE BLOCK DIAGRAM**



**TABLE 14-1: REGISTERS ASSOCIATED WITH COMPARATOR VOLTAGE REFERENCE**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR	Value on all other Resets
9Dh	CVRCON	CVREN	CVROE	CVRR	—	CVR3	CVR2	CVR1	CVR0	000- 0000	000- 0000
9Ch	CMCON	C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0	0000 0111	0000 0111

**Legend:** x = unknown, u = unchanged, - = unimplemented, read as '0'. Shaded cells are not used with the comparator voltage reference.

## 15.0 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)
- Two-Speed Start-up
- Fail-Safe Clock Monitor
- Sleep
- Code Protection
- ID Locations
- In-Circuit Serial Programming™ (ICSP™)

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.

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

## 15.1 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 locations 2007h and 2008h.

The user will note that address 2007h is beyond the user program memory space which can be accessed only during programming.

# PIC16F87/88

## REGISTER 15-1: CONFIG1: CONFIGURATION WORD 1 REGISTER (ADDRESS 2007h)

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	R/P-1
CP	CCPMX	DEBUG	WRT1	WRT0	CPD	LVP	BOREN	MCLR	FOSC2	PWRTEN	WDTEN	FOSC1	FOSC0	
bit 13													bit 0	

- bit 13 **CP:** Flash Program Memory Code Protection bits  
 1 = Code protection off  
 0 = 0000h to 0FFFh code-protected (all protected)
- bit 12 **CCPMX:** CCP1 Pin Selection bit  
 1 = CCP1 function on RB0  
 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 **WRT<1:0>:** Flash Program Memory Write Enable bits  
 11 = Write protection off  
 10 = 0000h to 00FFh write-protected, 0100h to 0FFFh may be modified by EECON control  
 01 = 0000h to 07FFh write-protected, 0800h to 0FFFh may be modified by EECON control  
 00 = 0000h to 0FFFh write-protected
- bit 8 **CPD:** Data EE Memory Code Protection bit  
 1 = Code protection off  
 0 = Data EE memory code-protected
- bit 7 **LVP:** Low-Voltage Programming Enable bit  
 1 = RB3/PGM pin has PGM function, Low-Voltage Programming enabled  
 0 = RB3 is digital I/O, 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 **MCLR:** RA5/ $\overline{\text{MCLR}}$ /VPP Pin Function Select bit  
 1 = RA5/ $\overline{\text{MCLR}}$ /VPP pin function is  $\overline{\text{MCLR}}$   
 0 = RA5/ $\overline{\text{MCLR}}$ /VPP pin function is digital I/O,  $\overline{\text{MCLR}}$  internally tied to VDD
- bit 3 **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 **FOSC<2:0>:** Oscillator Selection bits  
 111 = EXTRC oscillator; CLKO function on RA6/OSC2/CLKO  
 110 = EXTRC oscillator; port I/O function on RA6/OSC2/CLKO  
 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 = ECIO; port I/O function on RA6/OSC2/CLKO  
 010 = HS oscillator  
 001 = XT oscillator  
 000 = LP oscillator

### Legend:

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

## REGISTER 15-2: CONFIG2: CONFIGURATION WORD 2 REGISTER (ADDRESS 2008h)

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1	R/P-1	R/P-1
—	—	—	—	—	—	—	—	—	—	—	—	IESO	FCMEN
bit 13												bit 0	

bit 13-2 **Unimplemented:** Read as '1'

bit 1 **IESO:** Internal External Switchover bit  
1 = Internal External Switchover mode enabled  
0 = Internal External Switchover mode disabled

bit 0 **FCMEN:** Fail-Safe Clock Monitor Enable bit  
1 = Fail-Safe Clock Monitor enabled  
0 = Fail-Safe Clock Monitor disabled

<b>Legend:</b>			
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

# PIC16F87/88

## 15.2 Reset

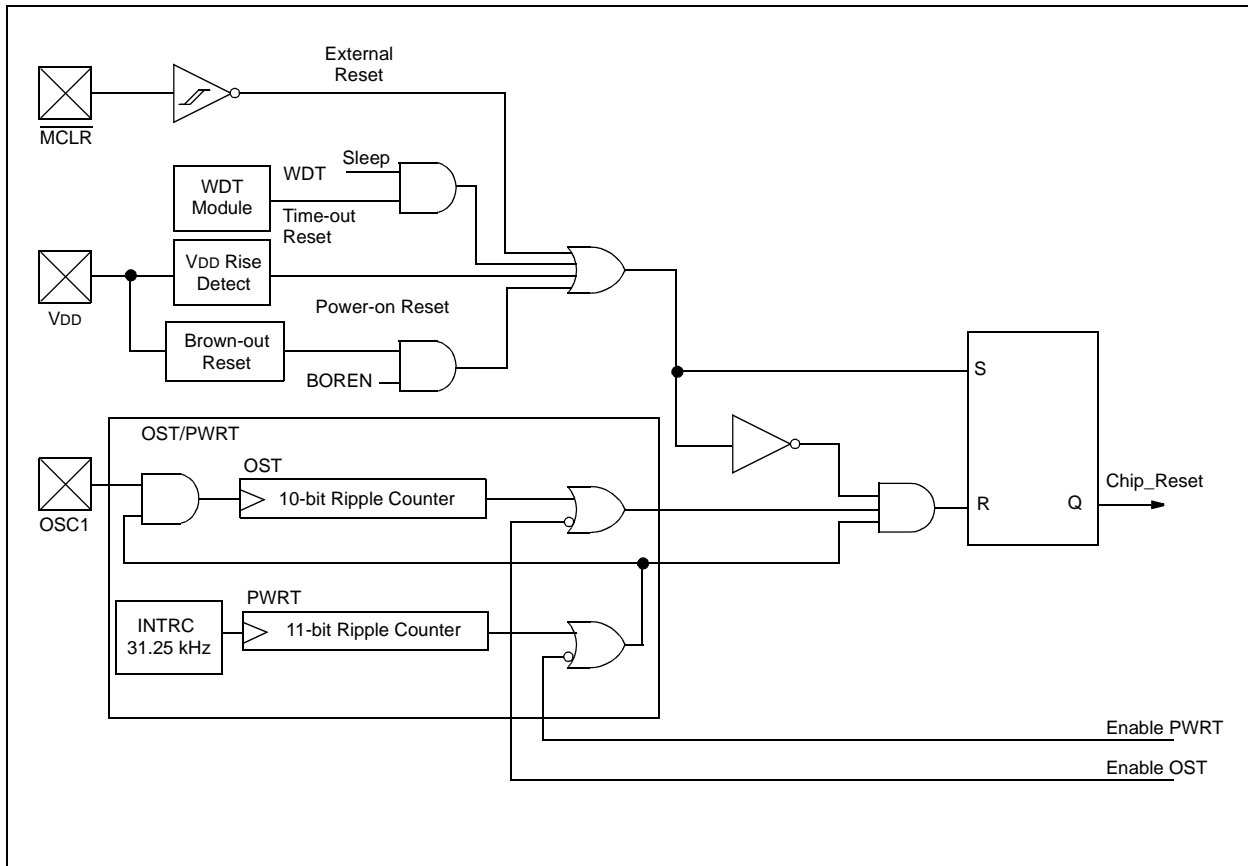
The PIC16F87/88 differentiates between various kinds of Reset:

- Power-on Reset (POR)
- $\overline{\text{MCLR}}$  Reset during normal operation
- $\overline{\text{MCLR}}$  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  $\overline{\text{MCLR}}$  and WDT Reset, on  $\overline{\text{MCLR}}$  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  $\overline{\text{TO}}$  and  $\overline{\text{PD}}$  bits are set or cleared differently in different Reset situations, as indicated in Table 15-3. 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  $\mu\text{s}$  to become ready for code execution. This delay runs in parallel with any other timers. See Table 15-4 for a full description of Reset states of all registers.

A simplified block diagram of the On-Chip Reset Circuit is shown in Figure 15-1.

**FIGURE 15-1: SIMPLIFIED BLOCK DIAGRAM OF ON-CHIP RESET CIRCUIT**



## 15.3 MCLR

PIC16F87/88 devices have a noise filter in the MCLR Reset path. The filter will detect and ignore small pulses.

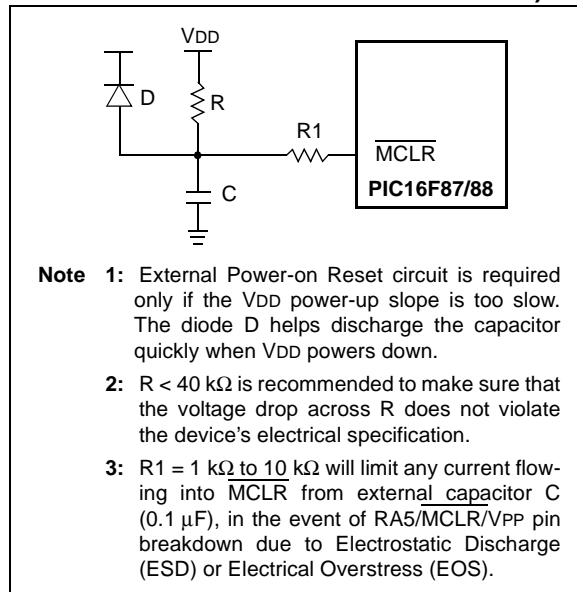
It should be noted that a WDT Reset does not drive MCLR pin low.

The behavior of the ESD protection on the MCLR pin has been altered from previous devices of this family. Voltages applied to the pin that exceed its specification can result in both MCLR and excessive current beyond the device specification during the ESD event. The circuit, as shown in Figure 15-2, is suggested.

**Note:** For this reason, Microchip recommends that the MCLR pin no longer be tied directly to VDD.

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 Configuration Word 1.

**FIGURE 15-2: EXTERNAL POWER-ON RESET CIRCUIT (FOR SLOW VDD POWER-UP)**



## 15.4 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 15.3 "MCLR". A maximum rise time for VDD is specified. See Section 18.0 "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).

## 15.5 Power-up Timer (PWRT)

The Power-up Timer (PWRT) of the PIC16F87/88 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 PWRTEEN.

## 15.6 Oscillator Start-up Timer (OST)

The Oscillator Start-up Timer (OST) provides a 1024 oscillator cycle (from OSC1 input) delay after the PWRT delay is over (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.

## 15.7 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  $\mu\text{s}$ ), 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 PWRTEEN and BOREN configuration bits are independent of each other.

# PIC16F87/88

## 15.8 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  $\overline{\text{MCLR}}$  is kept low long enough, all delays will expire. Bringing  $\overline{\text{MCLR}}$  high will begin execution immediately. This is useful for testing purposes, or to synchronize more than one PIC16F87/88 device operating in parallel.

Table 15-3 shows the Reset conditions for the STATUS, PCON and PC registers, while Table 15-4 shows the Reset conditions for all the registers.

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

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

**TABLE 15-1: TIME-OUT IN VARIOUS SITUATIONS**

Oscillator Configuration	Power-up		Brown-out Reset		Wake-up from Sleep
	$\overline{\text{PWRTE}} = 0$	$\overline{\text{PWRTE}} = 1$	$\overline{\text{PWRTE}} = 0$	$\overline{\text{PWRTE}} = 1$	
XT, HS, LP	$\text{TPWRT} + 1024 \cdot \text{TOSC}$	$1024 \cdot \text{TOSC}$	$\text{TPWRT} + 1024 \cdot \text{TOSC}$	$1024 \cdot \text{TOSC}$	$1024 \cdot \text{TOSC}$
EXTRC, INTRC	TPWRT	$5\text{-}10 \mu\text{s}^{(1)}$	TPWRT	$5\text{-}10 \mu\text{s}^{(1)}$	$5\text{-}10 \mu\text{s}^{(1)}$
T1OSC	—	—	—	—	$5\text{-}10 \mu\text{s}^{(1)}$

**Note 1:** CPU start-up is always invoked on POR, BOR and wake-up from Sleep. The 5-10  $\mu\text{s}$  delay is based on a 1 MHz system clock.

**TABLE 15-2: STATUS BITS AND THEIR SIGNIFICANCE**

$\overline{\text{POR}}$	$\overline{\text{BOR}}$	$\overline{\text{TO}}$	$\overline{\text{PD}}$	
0	x	1	1	Power-on Reset
0	x	0	x	Illegal, $\overline{\text{TO}}$ is set on $\overline{\text{POR}}$
0	x	x	0	Illegal, $\overline{\text{PD}}$ is set on $\overline{\text{POR}}$
1	0	1	1	Brown-out Reset
1	1	0	1	WDT Reset
1	1	0	0	WDT Wake-up
1	1	u	u	$\overline{\text{MCLR}}$ Reset during Normal Operation
1	1	1	0	$\overline{\text{MCLR}}$ Reset during Sleep or Interrupt Wake-up from Sleep

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



**TABLE 15-3: RESET CONDITION FOR SPECIAL REGISTERS**

Condition	Program Counter	STATUS Register	PCON Register
Power-on Reset	000h	0001 1xxx	---- --0x
MCLR Reset during normal operation	000h	000u uuuu	---- --uu
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 <sup>(1)</sup>	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).

**TABLE 15-4: INITIALIZATION CONDITIONS FOR ALL REGISTERS**

Register	Power-on Reset, Brown-out Reset	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 <sup>(2)</sup>
STATUS	0001 1xxx	000q quuu <sup>(3)</sup>	uuuq quuu <sup>(3)</sup>
FSR	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTA (PIC16F87)	xxxx 0000	uuuu 0000	uuuu uuuu
PORTA (PIC16F88)	xxx0 0000	uuu0 0000	uuuu uuuu
PORTB (PIC16F87)	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTB (PIC16F87)	00xx xxxx	00uu uuuu	uuuu uuuu
PCLATH	---0 0000	---0 0000	---u uuuu
INTCON	0000 000x	0000 000u	uuuu uuuu <sup>(1)</sup>
PIR1	-000 0000	-000 0000	-uuu uuuu <sup>(1)</sup>
PIR2	00-0 ----	00-0 ----	uu-u ---- <sup>(1)</sup>
TMR1L	xxxx xxxx	uuuu uuuu	uuuu uuuu
TMR1H	xxxx xxxx	uuuu uuuu	uuuu uuuu
T1CON	-000 0000	-uuu uuuu	-uuu 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
RCSTA	0000 000x	0000 000x	uuuu uuuu

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

**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 15-3 for Reset value for specific condition.

# PIC16F87/88

**TABLE 15-4: INITIALIZATION CONDITIONS FOR ALL REGISTERS (CONTINUED)**

Register	Power-on Reset, Brown-out Reset	MCLR Reset, WDT Reset	Wake-up via WDT or Interrupt
TXREG	0000 0000	0000 0000	uuuu uuuu
RCREG	0000 0000	0000 0000	uuuu 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	-000 0000	-000 0000	-uuu uuuu
PIE2	00-0 ----	00-0 ----	uu-u ----
PCON	---- -0q	---- -uu	---- -uu
OSCCON	-000 0000	-000 0000	-uuu uuuu
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
TXSTA	0000 -010	0000 -010	uuuu -u1u
SPBRG	0000 0000	0000 0000	uuuu uuuu
ANSEL	-111 1111	-111 1111	-111 1111
CMCON	0000 0111	0000 0111	uuuu u111
CVRCON	000- 0000	000- 0000	uuu- uuuu
WDTCON	---0 1000	---0 1000	---u uuuu
ADRESL	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADCON1	0000 ----	0000 ----	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	---- ----	---- ----	---- ----

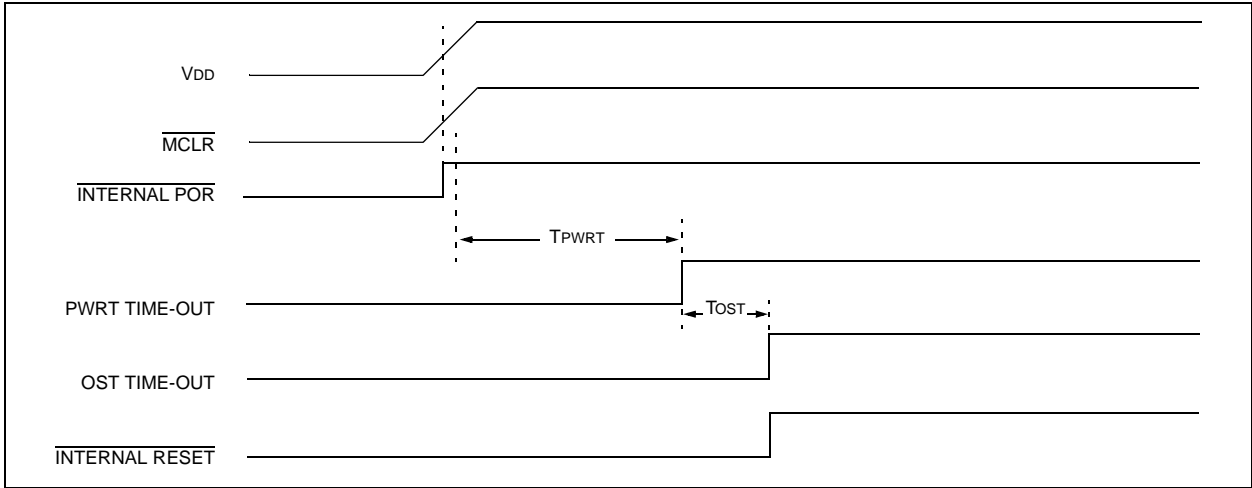
**Legend:** u = unchanged, x = unknown, - = unimplemented bit, read as '0', q = value depends on condition

**Note 1:** One or more bits in INTCON, PIR1 and PR2 will be affected (to cause wake-up).

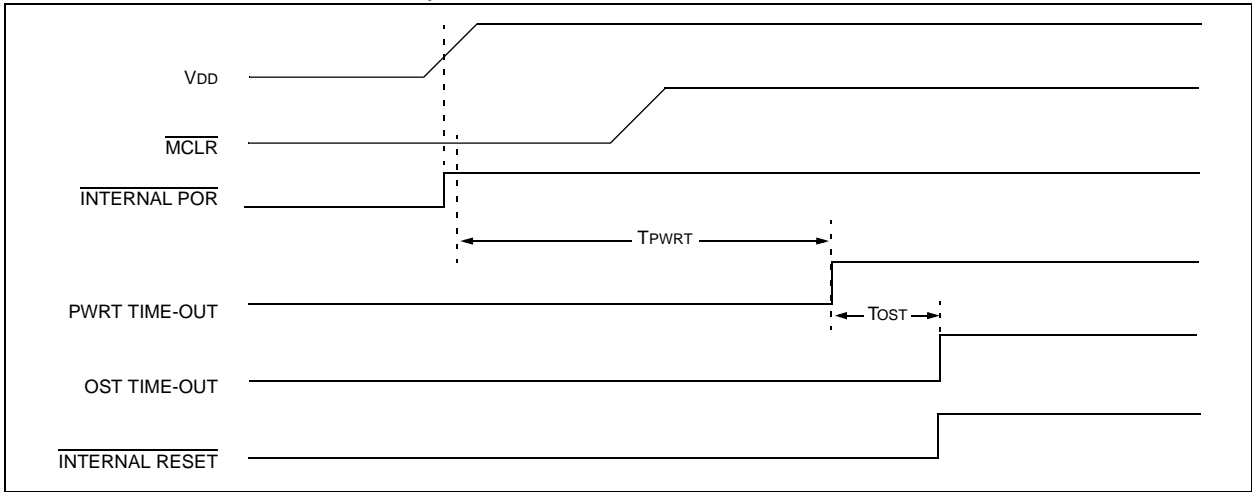
**2:** When the wake-up is due to an interrupt and the GIE bit is set, the PC is loaded with the interrupt vector (0004h).

**3:** See Table 15-3 for Reset value for specific condition.

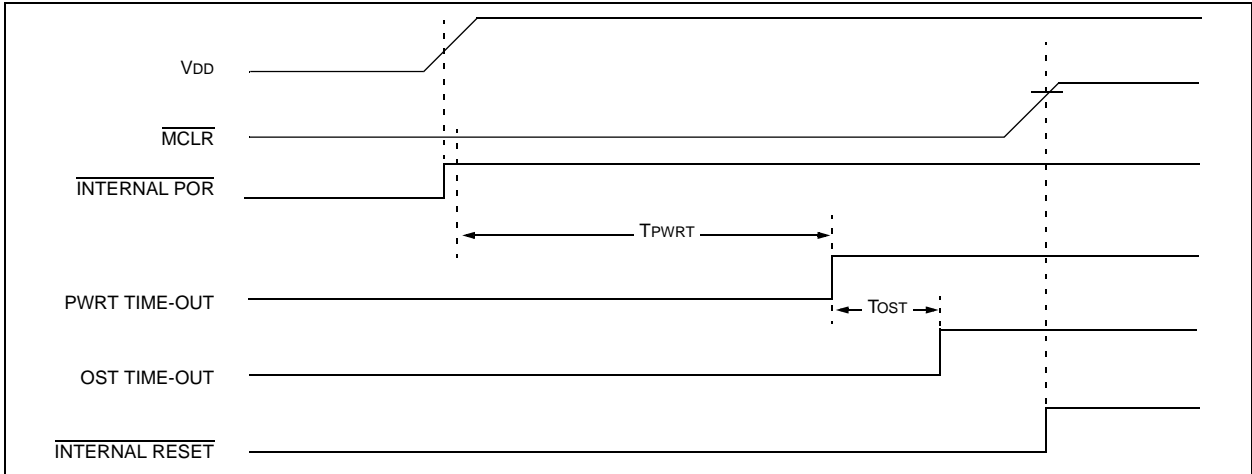
**FIGURE 15-3: TIME-OUT SEQUENCE ON POWER-UP ( $\overline{\text{MCLR}}$  TIED TO  $V_{\text{DD}}$  THROUGH PULL-UP RESISTOR)**



**FIGURE 15-4: TIME-OUT SEQUENCE ON POWER-UP ( $\overline{\text{MCLR}}$  TIED TO  $V_{\text{DD}}$  THROUGH RC NETWORK): CASE 1**

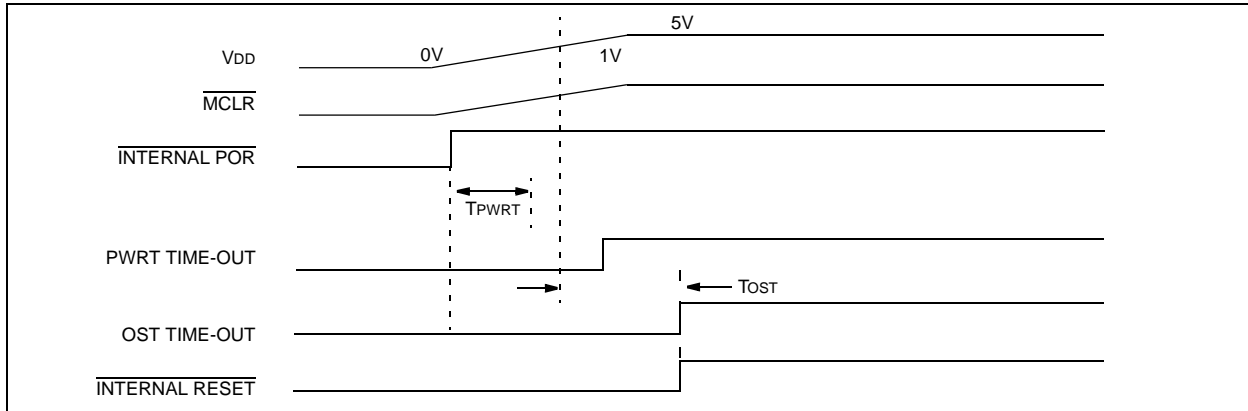


**FIGURE 15-5: TIME-OUT SEQUENCE ON POWER-UP ( $\overline{\text{MCLR}}$  TIED TO  $V_{\text{DD}}$  THROUGH RC NETWORK): CASE 2**



# PIC16F87/88

FIGURE 15-6: SLOW RISE TIME ( $\overline{\text{MCLR}}$  TIED TO  $V_{DD}$  THROUGH RC NETWORK)



## 15.10 Interrupts

The PIC16F87/88 has up to 12 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.

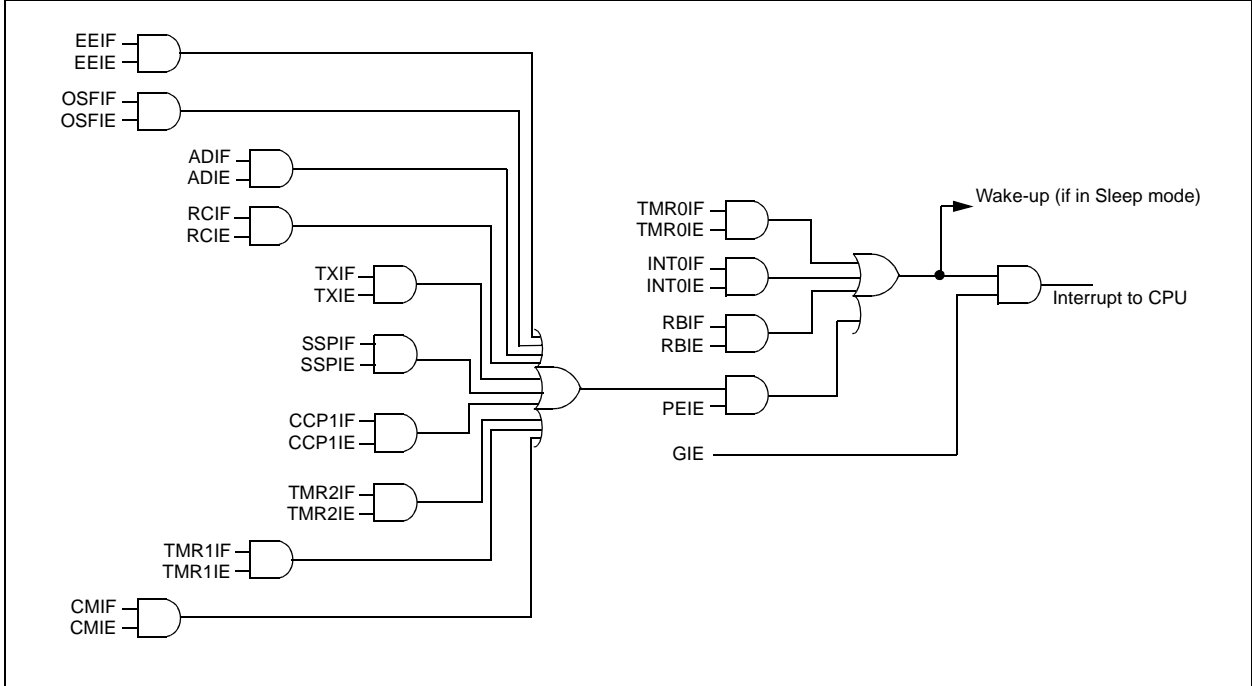
The RB0/INT pin interrupt, the RB port change interrupt and the TMR0 overflow interrupt flags are contained in the INTCON register.

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 15-7: INTERRUPT LOGIC



# PIC16F87/88

## 15.10.1 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, INT0IF (INTCON<1>), is set. This interrupt can be disabled by clearing enable bit INT0IE (INTCON<4>). Flag bit INT0IF 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 INT0IE 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 15.13 “Power-Down Mode (Sleep)”** for details on Sleep mode.

## 15.10.2 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”**.

## 15.10.3 PORTB INTCON CHANGE

An input change on PORTB<7:4> sets flag bit RBIF (INTCON<0>). The interrupt can be enabled/disabled by setting/clearing enable bit RBIE (INTCON<4>), see **Section 3.2 “EECON1 and EECON2 Registers”**.

## 15.11 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).

Since the upper 16 bytes of each bank are common in the PIC16F87/88 devices, temporary holding registers W\_TEMP, STATUS\_TEMP and PCLATH\_TEMP should be placed in here. These 16 locations don't require banking and therefore, make it easier for context save and restore. The same code shown in Example 15-1 can be used.

### EXAMPLE 15-1: SAVING STATUS, W AND PCLATH 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
MOVF   PCLATH, W   ;Only required if using page 1
MOVWF  PCLATH_TEMP ;Save PCLATH into W
CLRF   PCLATH      ;Page zero, regardless of current page
:
:(ISR)              ;(Insert user code here)
:
MOVF   PCLATH_TEMP, W ;Restore PCLATH
MOVWF  PCLATH       ;Move W into PCLATH
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
```

## 15.12 Watchdog Timer (WDT)

For PIC16F87/88 devices, the WDT has been modified from previous PIC16 devices. The new WDT is code and functionally backward compatible with previous PIC16 WDT modules and allows the user to have a scaler value for the WDT and TMR0 at the same time. In addition, the WDT time-out value can be extended to 268 seconds, using the prescaler with the postscaler when PSA is set to '1'.

### 15.12.1 WDT OSCILLATOR

The WDT derives its time base from the 31.25 kHz INTRC. The value of WDTCON is '---0 1000' on all Resets. This gives a nominal time base of 16.38 ms, which is compatible with the time base generated with previous PIC16 microcontroller versions.

**Note:** When the OST is invoked, the WDT is held in Reset because the WDT ripple counter is used by the OST to perform the oscillator delay count. When the OST count has expired, the WDT will begin counting (if enabled).

A new prescaler has been added to the path between the internal RC and the multiplexors used to select the path for the WDT. This prescaler is 16 bits and can be programmed to divide the internal RC by 32 to 65536, giving the time base used for the WDT a nominal range of 1 ms to 2.097s.

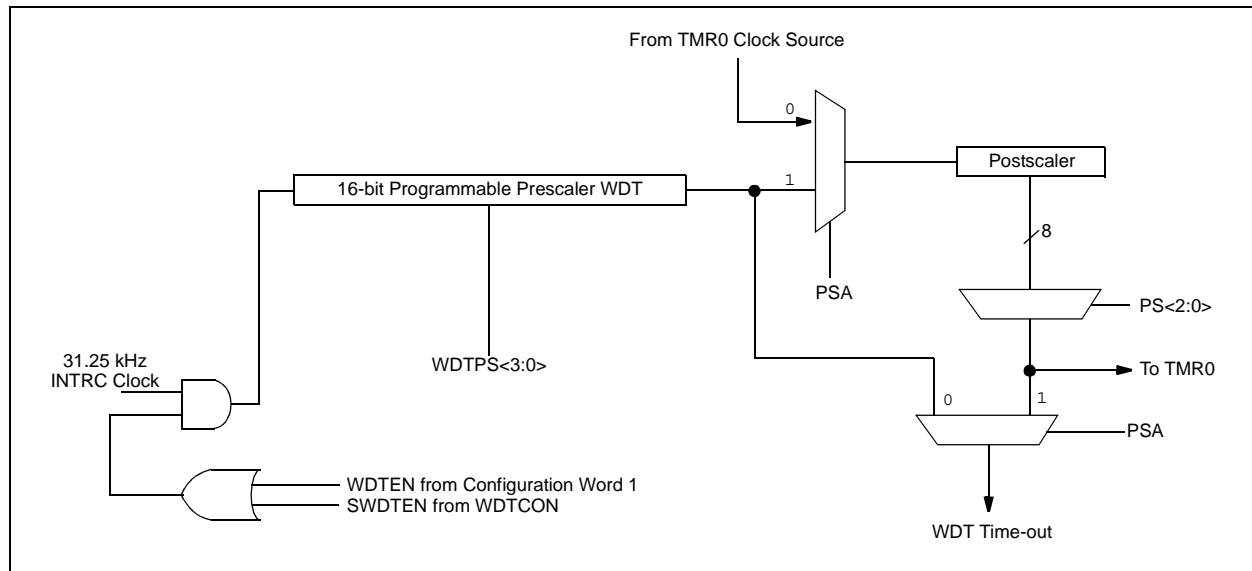
### 15.12.2 WDT CONTROL

The WDTEN bit is located in Configuration Word 1 and when this bit is set, the WDT runs continuously.

The SWDTEN bit is in the WDTCON register. When the WDTEN bit in the Configuration Word 1 register is set, the SWDTEN bit has no effect. If WDTEN is clear, then the SWDTEN bit can be used to enable and disable the WDT. Setting the bit will enable it and clearing the bit will disable it.

The PSA and PS<2:0> bits (OPTION\_REG register) have the same function as in previous versions of the PIC16 family of microcontrollers.

**FIGURE 15-8: WATCHDOG TIMER BLOCK DIAGRAM**



**TABLE 15-5: PRESCALER/POSTSCALER BIT STATUS**

Conditions	Prescaler	Postscaler (PSA = 1)
WDTEN = 0	Cleared	Cleared
CLRWDT command		
Oscillator fail detected		
Exit Sleep + System Clock = T1OSC, EXTRC, INTRC, ECIO		
Exit Sleep + System Clock = XT, HS, LP	Cleared at end of OST	Cleared at end of OST

# PIC16F87/88

## REGISTER 15-3: WDTCON: WATCHDOG CONTROL REGISTER (ADDRESS 105h)

U-0	U-0	U-0	R/W-0	R/W-1	R/W-0	R/W-0	R/W-0	
—	—	—	WDTPS3	WDTPS2	WDTPS1	WDTPS0	SWDTEN <sup>(1)</sup>	
bit 7								bit 0

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

bit 4-1 **WDTPS<3:0>:** Watchdog Timer Period Select bits

Bit Value	Prescale Rate
0000	= 1:32
0001	= 1:64
0010	= 1:128
0011	= 1:256
0100	= 1:512
0101	= 1:1024
0110	= 1:2048
0111	= 1:4096
1000	= 1:8192
1001	= 1:16394
1010	= 1:32768
1011	= 1:65536

bit 0 **SWDTEN:** Software Enable/Disable for Watchdog Timer bit<sup>(1)</sup>

1 = WDT is turned on

0 = WDT is turned off

**Note 1:** If WDTEEN configuration bit = 1, then WDT is always enabled, irrespective of this control bit. If WDTEEN configuration bit = 0, then it is possible to turn WDT on/off with this control bit.

### Legend:

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

**TABLE 15-6: 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	$\overline{\text{RBPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
2007h	Configuration bits	LVP	BOREN	MCLRE	FOSC2	$\overline{\text{PWRTEEN}}$	WDTEN	FOSC1	FOSC0
105h	WDTCON	—	—	—	WDTPS3	$\overline{\text{WDTPS2}}$	WSTPS1	WDTPS0	SWDTEN

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

**Note 1:** See Register 15-1 for operation of these bits.



## 15.12.3 TWO-SPEED CLOCK START-UP MODE

Two-Speed Start-up mode minimizes the latency between oscillator start-up and code execution that may be selected with the IESO (Internal/External Switchover) bit in Configuration Word 2. This mode is achieved by initially using the INTRC for code execution until the primary oscillator is stable.

If this mode is enabled and any of the following conditions exist, the system will begin execution with the INTRC oscillator. This results in almost immediate code execution with a minimum of delay.

- POR and after the Power-up Timer has expired (if  $PWRTEN = 0$ );
- or following a wake-up from Sleep;
- or a Reset when running from T1OSC or INTRC (after a Reset,  $SCS<1:0>$  are always set to '00').

**Note:** Following any Reset, the IRCF bits are zeroed and the frequency selection is forced to 31.25 kHz. The user can modify the IRCF bits to select a higher internal oscillator frequency.

If the primary oscillator is configured to be anything other than XT, LP or HS, then Two-Speed Start-up mode is disabled because the primary oscillator will not require any time to become stable after POR, or an exit from Sleep.

If the IRCF bits of the OSCCON register are configured to a non-zero value prior to entering Sleep mode, the system clock frequency will come from the output of the INTOSC. The IOFS bit in the OSCCON register will be clear until the INTOSC is stable. This will allow the user to determine when the internal oscillator can be used for time critical applications.

Checking the state of the OST<sub>S</sub> bit will confirm whether the primary clock configuration is engaged. If not, the OST<sub>S</sub> bit will remain clear.

When the device is auto-configured in INTRC mode following a POR or wake-up from Sleep, the rules for entering other oscillator modes still apply, meaning the  $SCS<1:0>$  bits in OSCCON can be modified before the OST time-out has occurred. This would allow the application to wake-up from Sleep, perform a few instructions using the INTRC as the clock source and go back to Sleep without waiting for the primary oscillator to become stable.

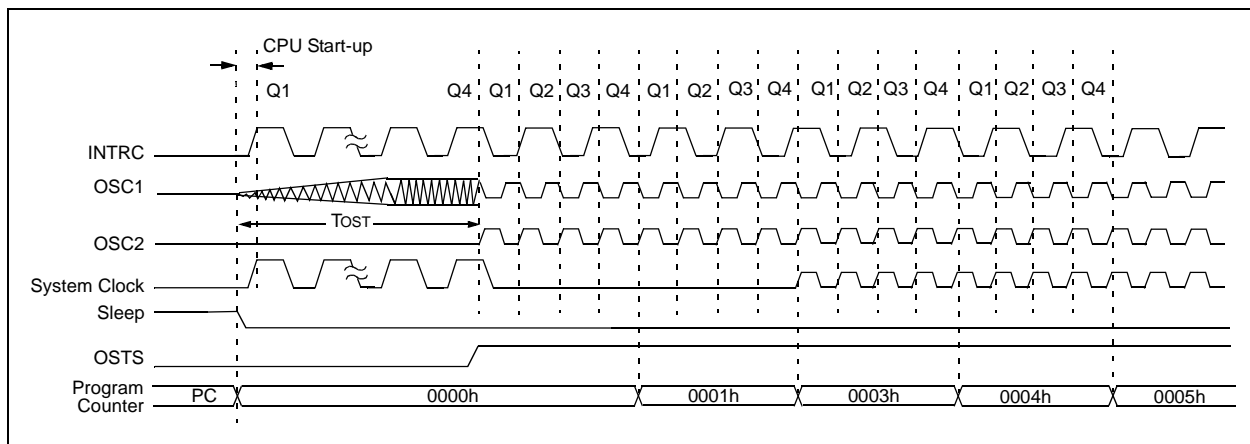
**Note:** Executing a SLEEP instruction will abort the oscillator start-up time and will cause the OST<sub>S</sub> bit to remain clear.

### 15.12.3.1 Two-Speed Start-up Mode Sequence

1. Wake-up from Sleep, Reset or POR.
2. OSCCON bits configured to run from INTRC (31.25 kHz).
3. Instructions begin execution by INTRC (31.25 kHz).
4. OST enabled to count 1024 clock cycles.
5. OST timed out, wait for falling edge of INTRC.
6. OST<sub>S</sub> is set.
7. System clock held low for eight falling edges of new clock (LP, XT or HS).
8. System clock is switched to primary source (LP, XT or HS).

The software may read the OST<sub>S</sub> bit to determine when the switchover takes place so that any software timing edges can be adjusted.

**FIGURE 15-9: TWO-SPEED START-UP MODE**

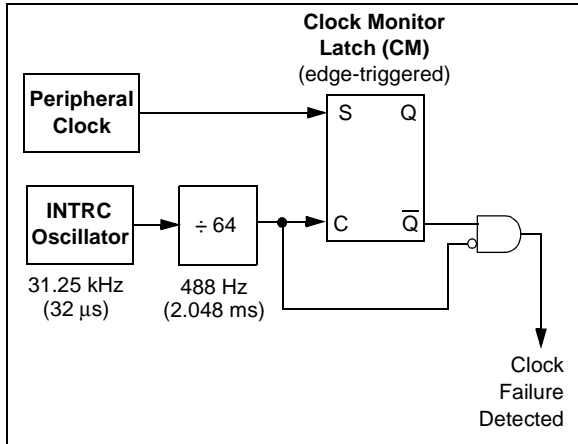


# PIC16F87/88

## 15.12.4 FAIL-SAFE OPTION

The Fail-Safe Clock Monitor (FSCM) is designed to allow the device to continue to operate even in the event of an oscillator failure.

**FIGURE 15-10: FSCM BLOCK DIAGRAM**



The FSCM function is enabled by setting the FCMEN bit in Configuration Word 2.

In the event of an oscillator failure, the FSCM will generate an oscillator fail interrupt and will switch the system clock over to the internal oscillator. The system will continue to come from the internal oscillator until the fail-safe condition is exited. The fail-safe condition is exited with either a Reset, the execution of a `SLEEP` instruction or a write to the `OSCCON` register.

The frequency of the internal oscillator will depend upon the value contained in the `IRCF` bits. Another clock source can be selected via the `IRCF` and the `SCS` bits of the `OSCCON` register.

The FSCM sample clock is generated by dividing the `INTRC` clock by 64. This will allow enough time between FSCM sample clocks for a system clock edge to occur.

On the rising edge of the postscaled clock, the monitoring latch (`CM = 0`) will be cleared. On a falling edge of the primary or secondary system clock, the monitoring latch will be set (`CM = 1`). In the event that a falling edge of the postscaled clock occurs and the monitoring latch is not set, a clock failure has been detected.

While in Fail-Safe mode, a Reset will exit the fail-safe condition. If the primary clock source is configured for a crystal, the `OST` timer will wait for the 1024 clock cycles for the `OST` time-out and the device will continue running from the internal oscillator until the `OST` is complete. A `SLEEP` instruction, or a write to the `SCS` bits (where `SCS` bits do not = 00), can be performed to put the device into a low-power mode.

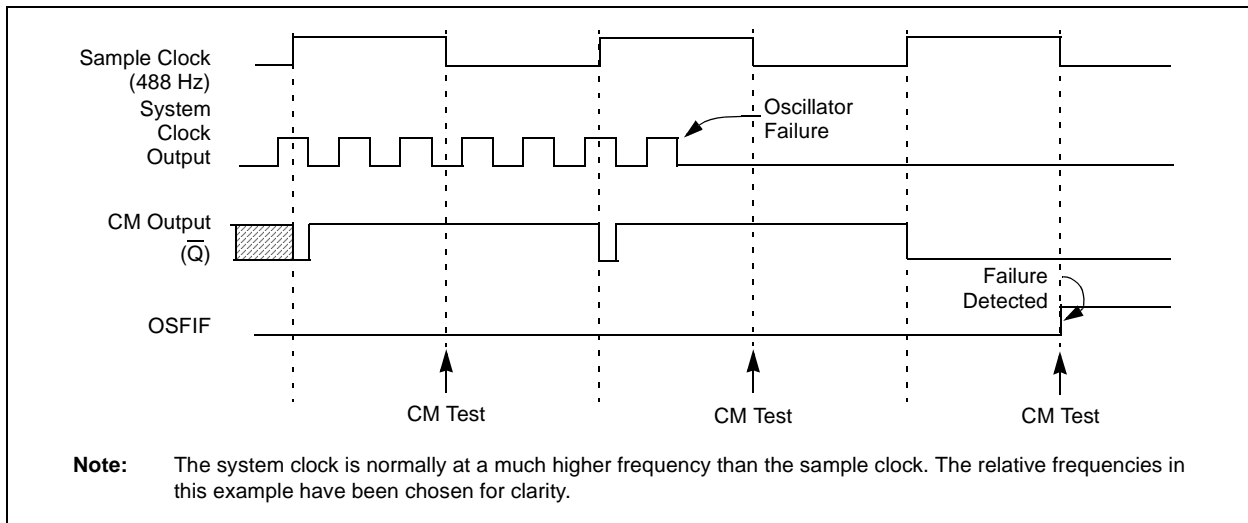
**Note:** Two-Speed Start-up mode is automatically enabled when the fail-safe option is enabled.

If Reset occurs while in Fail-Safe mode and the primary clock source is `EC` or `RC`, then the device will immediately switch back to `EC` or `RC` mode.

### 15.12.4.1 Fail-Safe in Low-Power Mode

A write to the `OSCCON` register, or `SLEEP` instruction, will end the fail-safe condition. The system clock will default to the source selected by the `SCS` bits, which is either `T1OSC`, `INTRC` or none (Sleep mode). However, the FSCM will continue to monitor the system clock. If the secondary clock fails, the device will immediately switch to the internal oscillator clock. If `OSFIE` is set, an interrupt will be generated.

**FIGURE 15-11: FSCM TIMING DIAGRAM**



## 15.12.4.2 FSCM and the Watchdog Timer

When a clock failure is detected, SCS<1:0> will be forced to '10' which will reset the WDT (if enabled).

## 15.12.4.3 POR or Wake From Sleep

The FSCM is designed to detect oscillator failure at any point after the device has exited Power-on Reset (POR) or low-power Sleep mode. When the primary system clock is EC, RC or INTRC modes, monitoring can begin immediately following these events.

For Oscillator modes involving a crystal or resonator (HS, LP or XT), the situation is somewhat different. Since the oscillator may require a start-up time considerably longer than the FSCM sample clock time, a false clock failure may be detected. To prevent this, the internal oscillator block is automatically configured as the system clock and functions until the primary clock is stable (the OST timer has timed out). This is identical to Two-Speed Start-up mode. Once the primary clock is stable, the INTRC returns to its role as the FSCM source.

**Note:** The same logic that prevents false oscillator failure interrupts on port or wake from Sleep, will also prevent the detection of the oscillator's failure to start at all following these events. This can be avoided by monitoring the OSTS bit and using a timing routine to determine if the oscillator is taking too long to start. Even so, no oscillator failure interrupt will be flagged.

## 15.12.4.4 Example Fail-Safe Conditions

### 1. CONDITIONS:

The device is clocked from a crystal, crystal operation fails and then Sleep mode is entered.

OSTS = 0

SCS = 00

OSFIF = 1

### USER ACTION:

Sleep mode will exit the fail-safe condition. Therefore, if the user code did not handle the detected fail-safe prior to the SLEEP command, then upon wake-up, the device will try to start the crystal that failed and a fail-safe condition will not be detected. Monitoring the OSTS bit will determine if the crystal is operating. The user should not enter Sleep mode without handling the fail-safe condition first.

### 2. CONDITIONS:

After a POR (Power-on Reset), the device is running in Two-Speed Start-up mode. The crystal fails before the OST has expired. If a crystal fails during the OST period, a fail-safe condition will not be detected (OSFIF will not get set).

OSTS = 0

SCS = 00

OSFIF = 0

### USER ACTION:

Check the OSTS bit. If it's clear and the OST should have expired at this point, then the user can assume the crystal has failed. The user should change the SCS bit to cause a clock switch which will also release the 10-bit ripple counter for WDT operation (if enabled).

### 3. CONDITIONS:

The device is clocked from a crystal during normal operation and it fails.

OSTS = 0

SCS = 00

OSFIF = 1

### USER ACTION:

Clear the OSFIF bit. Configure the SCS bits for a clock switch and the fail-safe condition will be cleared. Later, if the user decides to, the crystal can be retried for operation. If this is done, the OSTS bit should be monitored to determine if the crystal operates.

## 15.13 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  $\overline{PD}$  bit (STATUS<3>) is cleared, the  $\overline{TO}$  (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  $\overline{MCLR}$  pin must be at a logic high level (VIHMC).

# PIC16F87/88

## 15.13.1 WAKE-UP FROM SLEEP

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

1. External Reset input on  $\overline{\text{MCLR}}$  pin.
2. Watchdog Timer wake-up (if WDT was enabled).
3. Interrupt from INT pin, RB port change or a peripheral interrupt.

External  $\overline{\text{MCLR}}$  Reset will cause a device Reset. All other events are considered a continuation of program execution and cause a “wake-up”. The  $\overline{\text{TO}}$  and  $\overline{\text{PD}}$  bits in the STATUS register can be used to determine the cause of the device Reset. The  $\overline{\text{PD}}$  bit, which is set on power-up, is cleared when Sleep is invoked. The  $\overline{\text{TO}}$  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<sup>2</sup>C).
6. A/D conversion (when A/D clock source is RC).
7. EEPROM write operation completion.
8. Comparator output changes state.
9. AUSART RX or TX (Synchronous Slave mode).

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

When the SLEEP instruction is being executed, the next instruction (PC + 1) is 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.

## 15.13.2 WAKE-UP USING INTERRUPTS

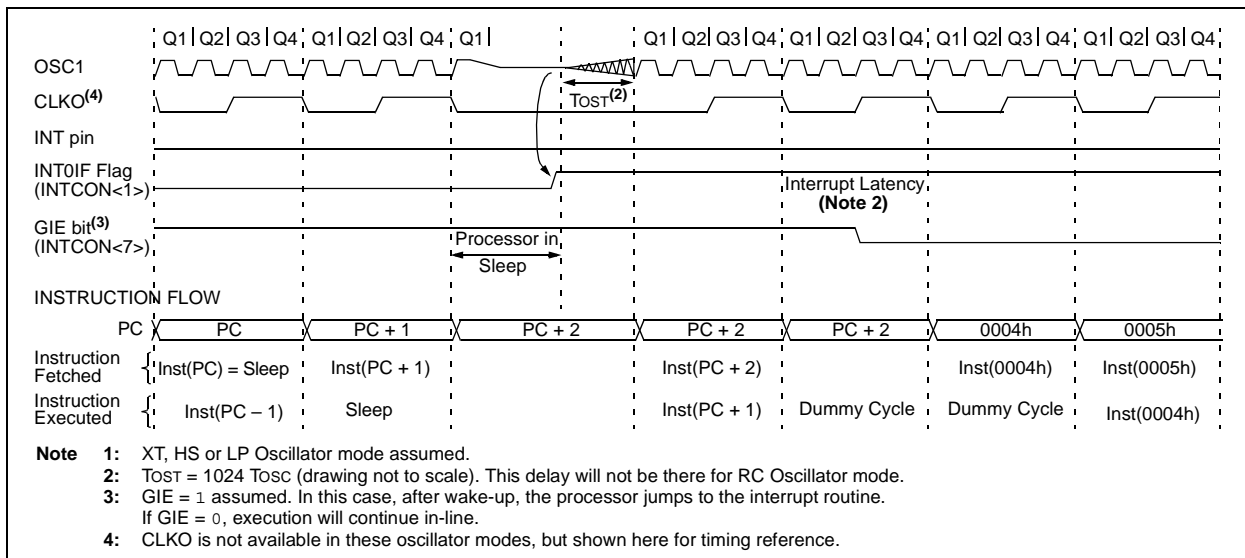
When global interrupts are disabled (GIE cleared) and any interrupt source has both its interrupt enable bit and interrupt flag bit set, one of the following will occur:

- If the interrupt occurs **before** the execution of a SLEEP instruction, the SLEEP instruction will complete as a NOP. Therefore, the WDT and WDT prescaler and postscaler (if enabled) will **not** be cleared, the  $\overline{\text{TO}}$  bit will not be set and the  $\overline{\text{PD}}$  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 prescaler and postscaler (if enabled) will be cleared, the  $\overline{\text{TO}}$  bit will be set and the  $\overline{\text{PD}}$  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  $\overline{\text{PD}}$  bit. If the  $\overline{\text{PD}}$  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.

**FIGURE 15-12: WAKE-UP FROM SLEEP THROUGH INTERRUPT<sup>(1)</sup>**



## 15.14 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 15-7 shows which features are consumed by the background debugger.

**TABLE 15-7: 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 RA5/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.

## 15.15 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.

## 15.16 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.

## 15.17 In-Circuit Serial Programming

PIC16F87/88 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 15-13 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.

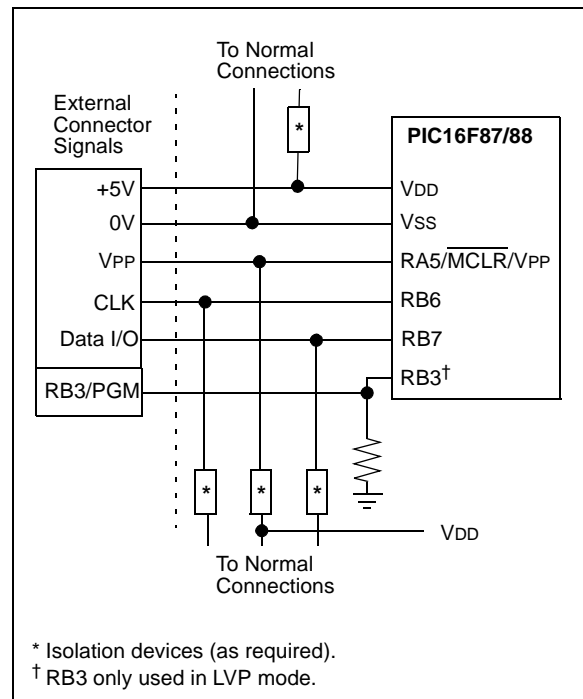
For more information on serial programming, please refer to the "PIC16F87/88 Flash Memory Programming Specification" (DS39607).

**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 15-13: TYPICAL IN-CIRCUIT SERIAL PROGRAMMING CONNECTION**



# PIC16F87/88

## 15.18 Low-Voltage ICSP Programming

The LVP bit of the Configuration Word enables Low-Voltage ICSP Programming. This mode allows the microcontroller to be programmed via ICSP using a  $V_{DD}$  source in the operating voltage range. This only means that  $V_{PP}$  does not have to be brought to  $V_{IH}$ , 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  $V_{IH}$  on  $\overline{MCLR}$ . The LVP bit can only be changed when using high voltage on  $\overline{MCLR}$ .

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 register.

1. Apply  $V_{DD}$  to the VDD pin.
2. Drive  $\overline{MCLR}$  low.
3. Apply  $V_{DD}$  to the RB3/PGM pin.
4. Apply  $V_{DD}$  to the  $\overline{MCLR}$  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  $V_{IH}$  to the  $\overline{MCLR}$  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 PIC16F87/88 devices 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 CONFIG1 register.
- 6: Disabling LVP will provide maximum compatibility to other PIC16CXXX devices.

## 16.0 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 16-1, while the various opcode fields are summarized in Table 16-1.

Table 16-2 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 PIC16F87/88 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.

### 16.1 Read-Modify-Write Operations

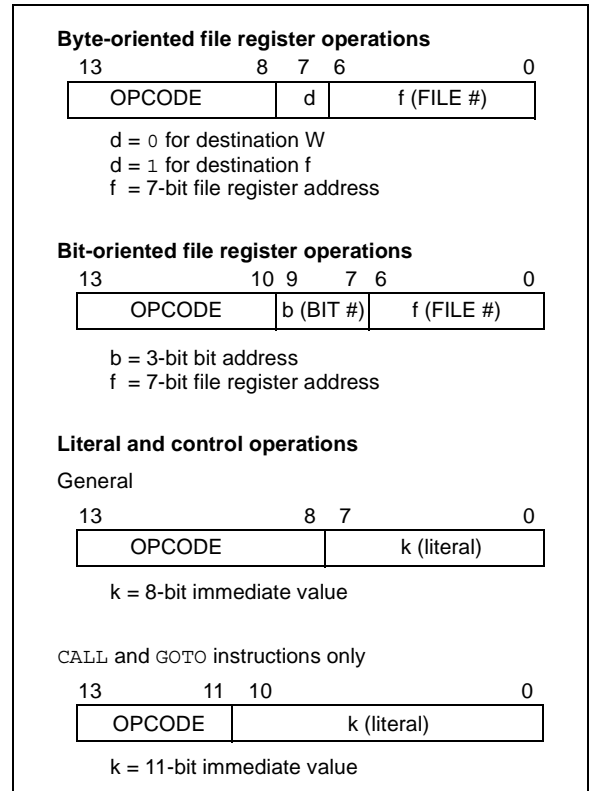
Any instruction that specifies a file register as part of the instruction performs a Read-Modify-Write (RMW) operation. The register is read, the data is modified and the result is stored according to either the instruction, or the destination designator ‘d’. A read operation is performed on a register 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 16-1: 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). The assembler will generate code with x = 0. It is the recommended form of use for compatibility with all Microchip software tools.
d	Destination select; d = 0: store result in W, d = 1: store result in file register f. Default is d = 1.
PC	Program Counter
TO	Time-out bit
PD	Power-Down bit

**FIGURE 16-1: GENERAL FORMAT FOR INSTRUCTIONS**



# PIC16F87/88

**TABLE 16-2: PIC16F87/88 INSTRUCTION SET**

Mnemonic, Operands	Description	Cycles	14-Bit Opcode		Status Affected	Notes
			MSb	LSb		
<b>BYTE-ORIENTED FILE REGISTER OPERATIONS</b>						
<b>ADDWF</b>	f, d Add W and f	1	00	0111	dfff ffff	C,DC,Z 1,2
<b>ANDWF</b>	f, d AND W with f	1	00	0101	dfff ffff	Z 1,2
<b>CLRF</b>	f Clear f	1	00	0001	1fff ffff	Z 2
<b>CLRW</b>	- Clear W	1	00	0001	0xxx xxxx	Z
<b>COMF</b>	f, d Complement f	1	00	1001	dfff ffff	Z 1,2
<b>DECf</b>	f, d Decrement f	1	00	0011	dfff ffff	Z 1,2
<b>DECFSZ</b>	f, d Decrement f, Skip if 0	1(2)	00	1011	dfff ffff	1,2,3
<b>INCF</b>	f, d Increment f	1	00	1010	dfff ffff	Z 1,2
<b>INCFSZ</b>	f, d Increment f, Skip if 0	1(2)	00	1111	dfff ffff	1,2,3
<b>IORWF</b>	f, d Inclusive OR W with f	1	00	0100	dfff ffff	Z 1,2
<b>MOVF</b>	f, d Move f	1	00	1000	dfff ffff	Z 1,2
<b>MOVWF</b>	f Move W to f	1	00	0000	1fff ffff	
<b>NOP</b>	- No Operation	1	00	0000	0xx0 0000	
<b>RLF</b>	f, d Rotate Left f through Carry	1	00	1101	dfff ffff	C 1,2
<b>RRF</b>	f, d Rotate Right f through Carry	1	00	1100	dfff ffff	C 1,2
<b>SUBWF</b>	f, d Subtract W from f	1	00	0010	dfff ffff	C,DC,Z 1,2
<b>SWAPF</b>	f, d Swap nibbles in f	1	00	1110	dfff ffff	1,2
<b>XORWF</b>	f, d Exclusive OR W with f	1	00	0110	dfff ffff	Z 1,2
<b>BIT-ORIENTED FILE REGISTER OPERATIONS</b>						
<b>BCF</b>	f, b Bit Clear f	1	01	00bb	bfff ffff	1,2
<b>BSF</b>	f, b Bit Set f	1	01	01bb	bfff ffff	1,2
<b>BTFSC</b>	f, b Bit Test f, Skip if Clear	1 (2)	01	10bb	bfff ffff	3
<b>BTFSS</b>	f, b Bit Test f, Skip if Set	1 (2)	01	11bb	bfff ffff	3
<b>LITERAL AND CONTROL OPERATIONS</b>						
<b>ADDLW</b>	k Add literal and W	1	11	111x	kkkk kkkk	C,DC,Z
<b>ANDLW</b>	k AND literal with W	1	11	1001	kkkk kkkk	Z
<b>CALL</b>	k Call subroutine	2	10	0kkk	kkkk kkkk	
<b>CLRWDt</b>	- Clear Watchdog Timer	1	00	0000	0110 0100	$\overline{TO}, \overline{PD}$
<b>GOTO</b>	k Go to address	2	10	1kkk	kkkk kkkk	
<b>IORLW</b>	k Inclusive OR literal with W	1	11	1000	kkkk kkkk	Z
<b>MOVLW</b>	k Move literal to W	1	11	00xx	kkkk kkkk	
<b>RETFIE</b>	- Return from interrupt	2	00	0000	0000 1001	
<b>RETLW</b>	k Return with literal in W	2	11	01xx	kkkk kkkk	
<b>RETURN</b>	- Return from Subroutine	2	00	0000	0000 1000	
<b>SLEEP</b>	- Go into Standby mode	1	00	0000	0110 0011	$\overline{TO}, \overline{PD}$
<b>SUBLW</b>	k Subtract W from literal	1	11	110x	kkkk kkkk	C,DC,Z
<b>XORLW</b>	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'.
- Note 2:** If this instruction is executed on the TMR0 register (and where applicable, d = 1), the prescaler will be cleared if assigned to the Timer0 module.
- Note 3:** If 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).



## 16.2 Instruction Descriptions

### **ADDLW**      **Add Literal and W**

Syntax:            [ *label* ] ADDLW   k

Operands:         $0 \leq k \leq 255$

Operation:         $(W) + k \rightarrow (W)$

Status Affected: C, DC, Z

Description:      The contents of the W register are added to the eight-bit literal 'k' and the result is placed in the W register.

### **ANDWF**        **AND W with f**

Syntax:            [ *label* ] ANDWF   f,d

Operands:         $0 \leq f \leq 127$   
 $d \in [0,1]$

Operation:         $(W) .AND. (f) \rightarrow (\text{destination})$

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'.

### **ADDWF**        **Add W and f**

Syntax:            [ *label* ] ADDWF   f,d

Operands:         $0 \leq f \leq 127$   
 $d \in [0,1]$

Operation:         $(W) + (f) \rightarrow (\text{destination})$

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'.

### **BCF**            **Bit Clear f**

Syntax:            [ *label* ] BCF    f,b

Operands:         $0 \leq f \leq 127$   
 $0 \leq b \leq 7$

Operation:         $0 \rightarrow (f<b>)$

Status Affected: None

Description:      Bit 'b' in register 'f' is cleared.

### **ANDLW**        **AND Literal with W**

Syntax:            [ *label* ] ANDLW   k

Operands:         $0 \leq k \leq 255$

Operation:         $(W) .AND. (k) \rightarrow (W)$

Status Affected: Z

Description:      The contents of W register are AND'ed with the eight-bit literal 'k'. The result is placed in the W register.

### **BSF**            **Bit Set f**

Syntax:            [ *label* ] BSF    f,b

Operands:         $0 \leq f \leq 127$   
 $0 \leq b \leq 7$

Operation:         $1 \rightarrow (f<b>)$

Status Affected: None

Description:      Bit 'b' in register 'f' is set.

# PIC16F87/88

---

<b>BTFSS</b>	<b>Bit Test f, Skip if Set</b>
Syntax:	[ <i>label</i> ] BTFSS f,b
Operands:	$0 \leq f \leq 127$ $0 \leq b < 7$
Operation:	skip if (f<b>) = 1
Status Affected:	None
Description:	If bit 'b' in register 'f' = 0, the next instruction is executed. If bit 'b' = 1, then the next instruction is discarded and a NOP is executed instead, making this a 2 Tcy instruction.

<b>CLRF</b>	<b>Clear f</b>
Syntax:	[ <i>label</i> ] CLRF f
Operands:	$0 \leq f \leq 127$
Operation:	00h → (f), 1 → Z
Status Affected:	Z
Description:	The contents of register 'f' are cleared and the Z bit is set.

<b>BTFSC</b>	<b>Bit Test, Skip if Clear</b>
Syntax:	[ <i>label</i> ] BTFSC f,b
Operands:	$0 \leq f \leq 127$ $0 \leq b \leq 7$
Operation:	skip if (f<b>) = 0
Status Affected:	None
Description:	If bit 'b' in register 'f' = 1, the next instruction is executed. If bit 'b', in register 'f', = 0, the next instruction is discarded and a NOP is executed instead, making this a 2 Tcy instruction.

<b>CLRW</b>	<b>Clear W</b>
Syntax:	[ <i>label</i> ] CLRW
Operands:	None
Operation:	00h → (W), 1 → Z
Status Affected:	Z
Description:	W register is cleared. Zero bit (Z) is set.

<b>CALL</b>	<b>Call Subroutine</b>
Syntax:	[ <i>label</i> ] CALL k
Operands:	$0 \leq k \leq 2047$
Operation:	(PC) + 1 → TOS, k → PC<10:0>, (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.

<b>CLRWDT</b>	<b>Clear Watchdog Timer</b>
Syntax:	[ <i>label</i> ] CLRWDT
Operands:	None
Operation:	00h → WDT, 0 → WDT prescaler, 1 → $\overline{TO}$ , 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.

<b>COMF</b>	<b>Complement f</b>
Syntax:	[ <i>label</i> ] COMF f,d
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$
Operation:	$(f) \rightarrow (\text{destination})$
Status Affected:	Z
Description:	The contents of register 'f' are complemented. If 'd' = 0, the result is stored in W. If 'd' = 1, the result is stored back in register 'f'.

<b>GOTO</b>	<b>Unconditional Branch</b>
Syntax:	[ <i>label</i> ] GOTO k
Operands:	$0 \leq k \leq 2047$
Operation:	$k \rightarrow \text{PC}\langle 10:0 \rangle$ , $\text{PCLATH}\langle 4:3 \rangle \rightarrow \text{PC}\langle 12:11 \rangle$
Status Affected:	None
Description:	GOTO is an unconditional branch. The eleven-bit immediate value is loaded into PC bits $\langle 10:0 \rangle$ . The upper bits of PC are loaded from PCLATH $\langle 4:3 \rangle$ . GOTO is a two-cycle instruction.

<b>DECF</b>	<b>Decrement f</b>
Syntax:	[ <i>label</i> ] DECF f,d
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$
Operation:	$(f) - 1 \rightarrow (\text{destination})$
Status Affected:	Z
Description:	Decrement register 'f'. If 'd' = 0, the result is stored in the W register. If 'd' = 1, the result is stored back in register 'f'.

<b>INCF</b>	<b>Increment f</b>
Syntax:	[ <i>label</i> ] INCF f,d
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$
Operation:	$(f) + 1 \rightarrow (\text{destination})$
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'.

<b>DECFSZ</b>	<b>Decrement f, Skip if 0</b>
Syntax:	[ <i>label</i> ] DECFSZ f,d
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$
Operation:	$(f) - 1 \rightarrow (\text{destination})$ ; 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'. 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.

<b>INCFSZ</b>	<b>Increment f, Skip if 0</b>
Syntax:	[ <i>label</i> ] INCFSZ f,d
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$
Operation:	$(f) + 1 \rightarrow (\text{destination})$ ; 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'. 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.

# PIC16F87/88

---

---

**IORLW**                    **Inclusive OR Literal with W**

---

Syntax:                    [ *label* ] IORLW k

Operands:                 $0 \leq k \leq 255$

Operation:                (W) .OR. k  $\rightarrow$  (W)

Status Affected:        Z

Description:              The contents of the W register are OR'ed with the eight-bit literal 'k'. The result is placed in the W register.

**MOVLW**                    **Move Literal to W**

---

Syntax:                    [ *label* ] MOVLW k

Operands:                 $0 \leq k \leq 255$

Operation:                k  $\rightarrow$  (W)

Status Affected:        None

Description:              The eight-bit literal 'k' is loaded into W register. The don't cares will assemble as '0's.

**IORWF**                    **Inclusive OR W with f**

---

Syntax:                    [ *label* ] IORWF f,d

Operands:                 $0 \leq f \leq 127$   
d  $\in$  [0,1]

Operation:                (W) .OR. (f)  $\rightarrow$  (destination)

Status Affected:        Z

Description:              Inclusive OR the W register with register 'f'. If 'd' = 0, the result is placed in the W register. If 'd' = 1, the result is placed back in register 'f'.

**MOVWF**                    **Move W to f**

---

Syntax:                    [ *label* ] MOVWF f

Operands:                 $0 \leq f \leq 127$

Operation:                (W)  $\rightarrow$  (f)

Status Affected:        None

Description:              Move data from W register to register 'f'.

**MOVF**                    **Move f**

---

Syntax:                    [ *label* ] MOVF f,d

Operands:                 $0 \leq f \leq 127$   
d  $\in$  [0,1]

Operation:                (f)  $\rightarrow$  (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.

**NOP**                    **No Operation**

---

Syntax:                    [ *label* ] NOP

Operands:                None

Operation:                No operation

Status Affected:        None

Description:              No operation.

**RETFIE**      **Return from Interrupt**

---

Syntax:      [ *label* ] RETFIE

Operands:    None

Operation:    TOS → PC,  
                  1 → GIE

Status Affected: None

**RLF**            **Rotate Left f through Carry**

---

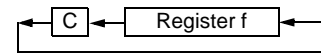
Syntax:      [ *label* ] RLF f,d

Operands:     $0 \leq f \leq 127$   
                   $d \in [0,1]$

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'.



**RETLW**        **Return with Literal in W**

---

Syntax:      [ *label* ] RETLW k

Operands:     $0 \leq k \leq 255$

Operation:     $k \rightarrow (W)$ ;  
                  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.

**RRF**            **Rotate Right f through Carry**

---

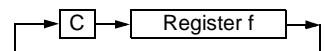
Syntax:      [ *label* ] RRF f,d

Operands:     $0 \leq f \leq 127$   
                   $d \in [0,1]$

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'.



**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.

**SLEEP**         **Sleep**

---

Syntax:      [ *label* ] SLEEP

Operands:    None

Operation:    00h → WDT,  
                  0 → WDT prescaler,  
                  1 →  $\overline{TO}$ ,  
                  0 → 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.

# PIC16F87/88

---

---

**SUBLW**      **Subtract W from Literal**

---

Syntax:      [ *label* ]    SUBLW   k

Operands:     $0 \leq k \leq 255$

Operation:     $k - (W) \rightarrow (W)$

Status Affected:    C, DC, Z

Description:    The W register is subtracted (two's complement method) from the eight-bit literal 'k'. The result is placed in the W register.

**XORLW**      **Exclusive OR Literal with W**

---

Syntax:      [ *label* ]    XORLW   k

Operands:     $0 \leq k \leq 255$

Operation:     $(W) .XOR. k \rightarrow (W)$

Status Affected:    Z

Description:    The contents of the W register are XOR'ed with the eight-bit literal 'k'. The result is placed in the W register.

**SUBWF**      **Subtract W from f**

---

Syntax:      [ *label* ]    SUBWF   f,d

Operands:     $0 \leq f \leq 127$   
                   $d \in [0,1]$

Operation:     $(f) - (W) \rightarrow (\text{destination})$

Status Affected:    C, DC, Z

Description:    Subtract (two'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'.

**XORWF**      **Exclusive OR W with f**

---

Syntax:      [ *label* ]    XORWF   f,d

Operands:     $0 \leq f \leq 127$   
                   $d \in [0,1]$

Operation:     $(W) .XOR. (f) \rightarrow (\text{destination})$

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'.

**SWAPF**      **Swap Nibbles in f**

---

Syntax:      [ *label* ]    SWAPF   f,d

Operands:     $0 \leq f \leq 127$   
                   $d \in [0,1]$

Operation:     $(f<3:0>) \rightarrow (\text{destination}<7:4>),$   
                   $(f<7:4>) \rightarrow (\text{destination}<3:0>)$

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'.

## 17.0 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

## 17.1 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.

## 17.2 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

## 17.3 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.

## 17.4 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/librarian 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

## 17.5 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.

## 17.6 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 its object file. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. Notable features of the assembler include:

- Support for the entire dsPIC30F instruction set
- Support for fixed-point and floating-point data
- Command line interface
- Rich directive set
- Flexible macro language
- MPLAB IDE compatibility

## 17.7 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.

## 17.8 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.



## 17.9 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.

## 17.10 MPLAB ICE 4000 High-Performance Universal In-Circuit Emulator

The MPLAB ICE 4000 universal in-circuit emulator is intended to provide the product development engineer with a complete microcontroller design tool set for high-end PICmicro microcontrollers. Software control of the MPLAB ICE in-circuit emulator is provided by the MPLAB Integrated Development Environment, which allows editing, building, downloading and source debugging from a single environment.

The MPLAB ICE 4000 is a premium emulator system, providing the features of MPLAB ICE 2000, but with increased emulation memory and high-speed performance for dsPIC30F and PIC18XXXX devices. Its advanced emulator features include complex triggering and timing, up to 2 Mb of emulation memory and the ability to view variables in real-time.

The MPLAB ICE 4000 in-circuit emulator system has been designed as a real-time emulation system with advanced features that are typically found on more expensive development tools. The PC platform and Microsoft Windows 32-bit operating system were chosen to best make these features available in a simple, unified application.

## 17.11 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.

## 17.12 PRO MATE II Universal Device Programmer

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

## 17.13 MPLAB PM3 Device Programmer

The MPLAB PM3 is a universal, CE compliant device programmer with programmable voltage verification at VDDMIN and VDDMAX for maximum reliability. It features a large LCD display (128 x 64) for menus and error messages and a modular detachable socket assembly to support various package types. The ICSP™ cable assembly is included as a standard item. In Stand-Alone mode, the MPLAB PM3 device programmer can read, verify and program PICmicro devices without a PC connection. It can also set code protection in this mode. 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.

# PIC16F87/88

---

## 17.14 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.

## 17.15 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.

## 17.16 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

## 17.17 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.

## 17.18 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.

## 17.19 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.

## 17.20 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.

## 17.21 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.

## 17.22 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.

## 17.23 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.

## 17.24 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.

## 17.25 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 rLab™ 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.

# PIC16F87/88

---

NOTES:

## 18.0 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 VSS (except VDD and $\overline{\text{MCLR}}$ ) .....	-0.3V to (VDD + 0.3V)
Voltage on VDD with respect to VSS .....	-0.3 to +7.5V
Voltage on $\overline{\text{MCLR}}$ with respect to VSS ( <b>Note 2</b> ) .....	-0.3 to +14V
Total power dissipation ( <b>Note 1</b> ) .....	1W
Maximum current out of VSS pin .....	200 mA
Maximum current into VDD pin .....	200 mA
Input clamp current, I <sub>IK</sub> (V <sub>I</sub> < 0 or V <sub>I</sub> > VDD) .....	±20 mA
Output clamp current, I <sub>OK</sub> (V <sub>O</sub> < 0 or V <sub>O</sub> > VDD) .....	±20 mA
Maximum output current sunk by any I/O pin .....	25 mA
Maximum output current sourced by any I/O pin .....	25 mA
Maximum current sunk by PORTA .....	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:  $P_{dis} = VDD \times \{I_{DD} - \sum I_{OH}\} + \sum \{(VDD - V_{OH}) \times I_{OH}\} + \sum (V_{OL} \times I_{OL})$

**2:** Voltage spikes at the  $\overline{\text{MCLR}}$  pin may cause latch-up. A series resistor of greater than 1 kΩ should be used to pull  $\overline{\text{MCLR}}$  to VDD, rather than tying the pin directly to VDD.

† 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.

# PIC16F87/88

FIGURE 18-1: PIC16F87/88 VOLTAGE-FREQUENCY GRAPH (INDUSTRIAL, EXTENDED)

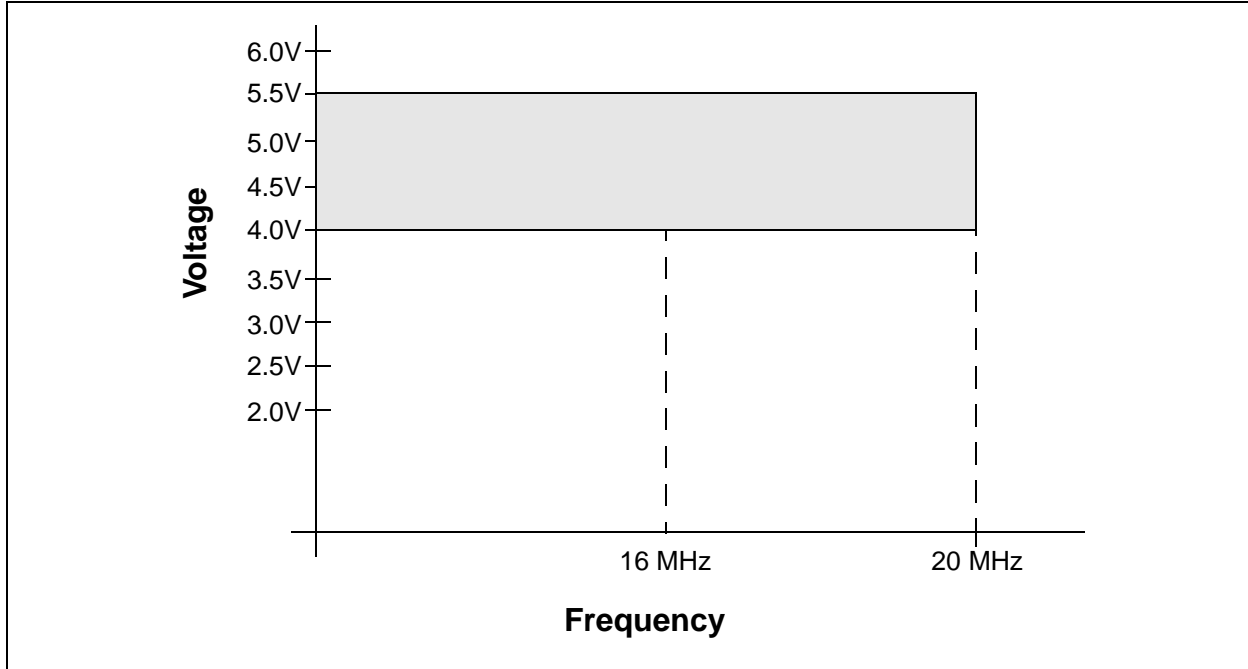
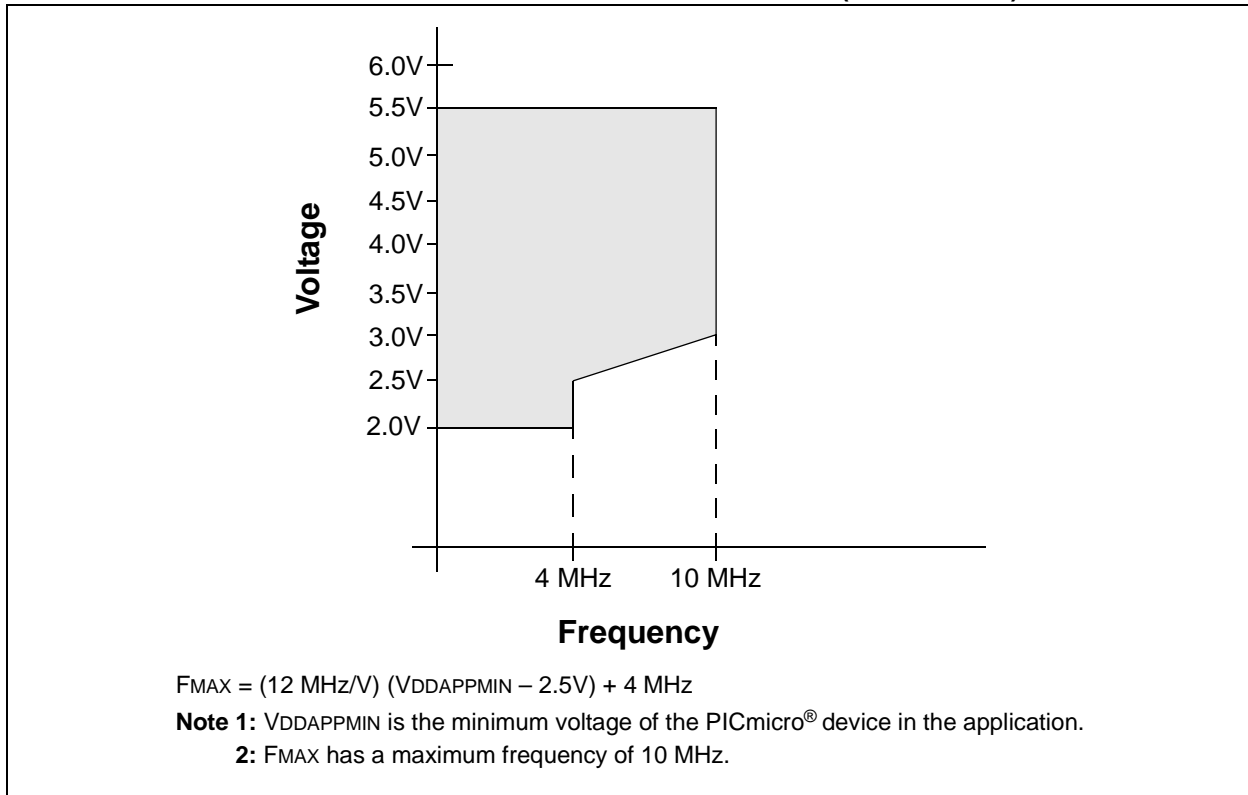


FIGURE 18-2: PIC16LF87/88 VOLTAGE-FREQUENCY GRAPH (INDUSTRIAL)



## 18.1 DC Characteristics: Supply Voltage PIC16F87/88 (Industrial, Extended) PIC16LF87/88 (Industrial)

PIC16LF87/88 (Industrial)		Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial					
PIC16F87/88 (Industrial, Extended)		Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ for extended					
Param No.	Symbol	Characteristic	Min	Typ	Max	Units	Conditions
D001	VDD	<b>Supply Voltage</b>					
		PIC16LF87/88	2.0	—	5.5	V	HS, XT, RC and LP Oscillator mode
D001		PIC16F87/88	4.0	—	5.5	V	
D002	VDR	<b>RAM Data Retention Voltage<sup>(1)</sup></b>	1.5	—	—	V	
D003	VPOR	<b>VDD Start Voltage</b> to ensure internal Power-on Reset signal	—	—	0.7	V	See Section 15.4 “Power-on Reset (POR)” for details
D004	SVDD	<b>VDD Rise Rate</b> to ensure internal Power-on Reset signal	0.05	—	—	V/ms	See Section 15.4 “Power-on Reset (POR)” for details
D005	VBOR	<b>Brown-out Reset Voltage</b>					
		PIC16LF87/88	3.65	—	4.35	V	
D005		PIC16F87/88	3.65	—	4.35	V	F <sub>MAX</sub> = 14 MHz <sup>(2)</sup>

**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.

# PIC16F87/88

## 18.2 DC Characteristics: Power-Down and Supply Current PIC16F87/88 (Industrial, Extended) PIC16LF87/88 (Industrial)

<b>PIC16LF87/88</b> (Industrial)		<b>Standard Operating Conditions (unless otherwise stated)</b> Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial				
<b>PIC16F87/88</b> (Industrial, Extended)		<b>Standard Operating Conditions (unless otherwise stated)</b> Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ for extended				
Param No.	Device	Typ	Max	Units	Conditions	
<b>Power-Down Current (<math>I_{PD}</math>)<sup>(1)</sup></b>						
	PIC16LF87/88	0.1	0.4	$\mu\text{A}$	$-40^{\circ}\text{C}$	$V_{DD} = 2.0\text{V}$
		0.1	0.4	$\mu\text{A}$	$+25^{\circ}\text{C}$	
		0.4	1.5	$\mu\text{A}$	$+85^{\circ}\text{C}$	
	PIC16LF87/88	0.3	0.5	$\mu\text{A}$	$-40^{\circ}\text{C}$	$V_{DD} = 3.0\text{V}$
		0.3	0.5	$\mu\text{A}$	$+25^{\circ}\text{C}$	
		0.7	1.7	$\mu\text{A}$	$+85^{\circ}\text{C}$	
	All devices	0.6	1.0	$\mu\text{A}$	$-40^{\circ}\text{C}$	$V_{DD} = 5.0\text{V}$
		0.6	1.0	$\mu\text{A}$	$+25^{\circ}\text{C}$	
		1.2	5.0	$\mu\text{A}$	$+85^{\circ}\text{C}$	
Extended devices	6	28	$\mu\text{A}$	$+125^{\circ}\text{C}$		

**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  $V_{DD}$  or  $V_{SS}$  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  $I_{DD}$  measurements in active operation mode are:  
OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to  $V_{DD}$ ;  
MCLR =  $V_{DD}$ ; 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  $I_r = V_{DD}/2R_{EXT}$  (mA) with REXT in  $k\Omega$ .



## 18.2 DC Characteristics: Power-Down and Supply Current PIC16F87/88 (Industrial, Extended) PIC16LF87/88 (Industrial) (Continued)

PIC16LF87/88 (Industrial)		Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial					
PIC16F87/88 (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		
<b>Supply Current (IDD)<sup>(2,3)</sup></b>							
	PIC16LF87/88	9	20	μA	-40°C	VDD = 2.0V	FOSC = 32 kHz (LP Oscillator)
		7	15	μA	+25°C		
		7	15	μA	+85°C		
	PIC16LF87/88	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;  
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  $I_r = V_{DD}/2R_{EXT}$  (mA) with REXT in kΩ.

# PIC16F87/88

## 18.2 DC Characteristics: Power-Down and Supply Current PIC16F87/88 (Industrial, Extended) PIC16LF87/88 (Industrial) (Continued)

PIC16LF87/88 (Industrial)		Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial					
PIC16F87/88 (Industrial, Extended)		Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ for extended					
Param No.	Device	Typ	Max	Units	Conditions		
<b>Supply Current (<math>I_{DD}</math>)<sup>(2,3)</sup></b>							
PIC16LF87/88		72	95	$\mu\text{A}$	$-40^{\circ}\text{C}$	$V_{DD} = 2.0\text{V}$	Fosc = 1 MHz (RC Oscillator) <sup>(3)</sup>
		76	90	$\mu\text{A}$	$+25^{\circ}\text{C}$		
		76	90	$\mu\text{A}$	$+85^{\circ}\text{C}$		
PIC16LF87/88		138	175	$\mu\text{A}$	$-40^{\circ}\text{C}$	$V_{DD} = 3.0\text{V}$	
		136	170	$\mu\text{A}$	$+25^{\circ}\text{C}$		
		136	170	$\mu\text{A}$	$+85^{\circ}\text{C}$		
All devices		310	380	$\mu\text{A}$	$-40^{\circ}\text{C}$	$V_{DD} = 5.0\text{V}$	
		290	360	$\mu\text{A}$	$+25^{\circ}\text{C}$		
		280	360	$\mu\text{A}$	$+85^{\circ}\text{C}$		
Extended devices		330	500	$\mu\text{A}$	$125^{\circ}\text{C}$		
PIC16LF87/88		270	335	$\mu\text{A}$	$-40^{\circ}\text{C}$	$V_{DD} = 2.0\text{V}$	Fosc = 4 MHz (RC Oscillator) <sup>(3)</sup>
		280	330	$\mu\text{A}$	$+25^{\circ}\text{C}$		
		285	330	$\mu\text{A}$	$+85^{\circ}\text{C}$		
PIC16LF87/88		460	610	$\mu\text{A}$	$-40^{\circ}\text{C}$	$V_{DD} = 3.0\text{V}$	
		450	600	$\mu\text{A}$	$+25^{\circ}\text{C}$		
		450	600	$\mu\text{A}$	$+85^{\circ}\text{C}$		
All devices		900	1060	$\mu\text{A}$	$-40^{\circ}\text{C}$	$V_{DD} = 5.0\text{V}$	
		890	1050	$\mu\text{A}$	$+25^{\circ}\text{C}$		
		890	1050	$\mu\text{A}$	$+85^{\circ}\text{C}$		
Extended devices		.920	1.5	$\text{mA}$	$+125^{\circ}\text{C}$		

**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  $V_{DD}$  or  $V_{SS}$  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  $I_{DD}$  measurements in active operation mode are:  
 $\text{OSC1}$  = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to  $V_{DD}$ ;  
 $\text{MCLR}$  =  $V_{DD}$ ; WDT enabled/disabled as specified.
- 3:** For RC oscillator configurations, current through  $R_{EXT}$  is not included. The current through the resistor can be estimated by the formula  $I_r = V_{DD}/2R_{EXT}$  (mA) with  $R_{EXT}$  in  $\text{k}\Omega$ .

## 18.2 DC Characteristics: Power-Down and Supply Current PIC16F87/88 (Industrial, Extended) PIC16LF87/88 (Industrial) (Continued)

<b>PIC16LF87/88</b> (Industrial)		<b>Standard Operating Conditions (unless otherwise stated)</b> Operating temperature -40°C ≤ TA ≤ +85°C for industrial					
<b>PIC16F87/88</b> (Industrial, Extended)		<b>Standard Operating Conditions (unless otherwise stated)</b> Operating temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended					
Param No.	Device	Typ	Max	Units	Conditions		
<b>Supply Current (IDD)<sup>(2,3)</sup></b>							
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	+85°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  $I_r = V_{DD}/2R_{EXT}$  (mA) with REXT in kΩ.

# PIC16F87/88

## 18.2 DC Characteristics: Power-Down and Supply Current PIC16F87/88 (Industrial, Extended) PIC16LF87/88 (Industrial) (Continued)

PIC16LF87/88 (Industrial)		Standard Operating Conditions (unless otherwise stated)					
		Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial					
PIC16F87/88 (Industrial, Extended)		Standard Operating Conditions (unless otherwise stated)					
		Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ for extended					
Param No.	Device	Typ	Max	Units	Conditions		
<b>Supply Current (<math>I_{DD}</math>)<sup>(2,3)</sup></b>							
	PIC16LF87/88	8	20	$\mu\text{A}$	$-40^{\circ}\text{C}$	$V_{DD} = 2.0\text{V}$	FOSC = 31.25 kHz (RC_RUN mode, Internal RC Oscillator)
		7	15	$\mu\text{A}$	$+25^{\circ}\text{C}$		
		7	15	$\mu\text{A}$	$+85^{\circ}\text{C}$		
	PIC16LF87/88	16	30	$\mu\text{A}$	$-40^{\circ}\text{C}$	$V_{DD} = 3.0\text{V}$	
		14	25	$\mu\text{A}$	$+25^{\circ}\text{C}$		
		14	25	$\mu\text{A}$	$+85^{\circ}\text{C}$		
	All devices	32	40	$\mu\text{A}$	$-40^{\circ}\text{C}$	$V_{DD} = 5.0\text{V}$	
		29	35	$\mu\text{A}$	$+25^{\circ}\text{C}$		
		29	35	$\mu\text{A}$	$+85^{\circ}\text{C}$		
	Extended devices	35	45	$\mu\text{A}$	$+125^{\circ}\text{C}$		
	PIC16LF87/88	132	160	$\mu\text{A}$	$-40^{\circ}\text{C}$	$V_{DD} = 2.0\text{V}$	FOSC = 1 MHz (RC_RUN mode, Internal RC Oscillator)
		126	155	$\mu\text{A}$	$+25^{\circ}\text{C}$		
		126	155	$\mu\text{A}$	$+85^{\circ}\text{C}$		
	PIC16LF87/88	260	310	$\mu\text{A}$	$-40^{\circ}\text{C}$	$V_{DD} = 3.0\text{V}$	
		230	300	$\mu\text{A}$	$+25^{\circ}\text{C}$		
		230	300	$\mu\text{A}$	$+85^{\circ}\text{C}$		
	All devices	560	690	$\mu\text{A}$	$-40^{\circ}\text{C}$	$V_{DD} = 5.0\text{V}$	
		500	650	$\mu\text{A}$	$+25^{\circ}\text{C}$		
		500	650	$\mu\text{A}$	$+85^{\circ}\text{C}$		
	Extended devices	570	710	$\mu\text{A}$	$+125^{\circ}\text{C}$		

**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  $V_{DD}$  or  $V_{SS}$  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  $I_{DD}$  measurements in active operation mode are:  
 $\text{OSC1}$  = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to  $V_{DD}$ ;  
 $\text{MCLR}$  =  $V_{DD}$ ; WDT enabled/disabled as specified.
- 3:** For RC oscillator configurations, current through  $R_{EXT}$  is not included. The current through the resistor can be estimated by the formula  $I_r = V_{DD}/2R_{EXT}$  (mA) with  $R_{EXT}$  in k $\Omega$ .

## 18.2 DC Characteristics: Power-Down and Supply Current PIC16F87/88 (Industrial, Extended) PIC16LF87/88 (Industrial) (Continued)

PIC16LF87/88 (Industrial)		Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial				
PIC16F87/88 (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	
<b>Supply Current (IDD)<sup>(2,3)</sup></b>						
	PIC16LF87/88	310	420	μA	-40°C	VDD = 2.0V
		300	410	μA	+25°C	
		300	410	μA	+85°C	
	PIC16LF87/88	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;  
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  $I_r = V_{DD}/2R_{EXT}$  (mA) with REXT in kΩ.

# PIC16F87/88

## 18.2 DC Characteristics: Power-Down and Supply Current PIC16F87/88 (Industrial, Extended) PIC16LF87/88 (Industrial) (Continued)

PIC16LF87/88 (Industrial)		Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial						
PIC16F87/88 (Industrial, Extended)		Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ for extended						
Param No.	Device	Typ	Max	Units	Conditions			
<b>Supply Current (I<sub>DD</sub>)<sup>(2,3)</sup></b>								
	PIC16LF87/88	.950	1.3	mA	-40°C	V <sub>DD</sub> = 3.0V	F <sub>OSC</sub> = 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	V <sub>DD</sub> = 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			
PIC16LF87/88		9	13	μA	-10°C	V <sub>DD</sub> = 2.0V		F <sub>OSC</sub> = 32 kHz (SEC_RUN mode, Timer1 as clock)
		9	14	μA	+25°C			
		11	16	μA	+70°C			
PIC16LF87/88		12	34	μA	-10°C	V <sub>DD</sub> = 3.0V		
		12	31	μA	+25°C			
		14	28	μA	+70°C			
All devices		20	72	μA	-10°C	V <sub>DD</sub> = 5.0V		
		20	65	μA	+25°C			
		25	59	μA	+70°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 V<sub>DD</sub> or V<sub>SS</sub> 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 I<sub>DD</sub> measurements in active operation mode are:  
OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to V<sub>DD</sub>;  
MCLR = V<sub>DD</sub>; WDT enabled/disabled as specified.
- 3:** For RC oscillator configurations, current through R<sub>EXT</sub> is not included. The current through the resistor can be estimated by the formula  $I_r = V_{DD}/2R_{EXT}$  (mA) with R<sub>EXT</sub> in kΩ.

## 18.2 DC Characteristics: Power-Down and Supply Current PIC16F87/88 (Industrial, Extended) PIC16LF87/88 (Industrial) (Continued)

PIC16LF87/88 (Industrial)		Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial					
PIC16F87/88 (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 ( $\Delta I_{WDT}$ )	<b>Module Differential Currents (<math>\Delta I_{WDT}</math>, <math>\Delta I_{BOR}</math>, <math>\Delta I_{LVD}</math>, <math>\Delta I_{OSCB}</math>, <math>\Delta I_{AD}</math>)</b>						
	<b>Watchdog Timer</b>	1.5	3.8	$\mu A$	-40°C	VDD = 2.0V	
		2.2	3.8	$\mu A$	+25°C		
		2.7	4.0	$\mu A$	+85°C		
		2.3	4.6	$\mu A$	-40°C	VDD = 3.0V	
		2.7	4.6	$\mu A$	+25°C		
		3.1	4.8	$\mu A$	+85°C		
		3.0	10.0	$\mu A$	-40°C	VDD = 5.0V	
		3.3	10.0	$\mu A$	+25°C		
		3.9	13.0	$\mu A$	+85°C		
		Extended devices	5.0	21.0	$\mu A$	+125°C	
D022A ( $\Delta I_{BOR}$ )	<b>Brown-out Reset</b>	40	60	$\mu A$	-40°C to +85°C	VDD = 5.0V	
D025 ( $\Delta I_{OSCB}$ )	<b>Timer1 Oscillator</b>	1.7	2.3	$\mu A$	-40°C	VDD = 2.0V	
		1.8	2.3	$\mu A$	+25°C		
		2.0	2.3	$\mu A$	+85°C		
		2.2	3.8	$\mu A$	-40°C	VDD = 3.0V	
		2.6	3.8	$\mu A$	+25°C		
		2.9	3.8	$\mu A$	+85°C		
		3.0	6.0	$\mu A$	-40°C	VDD = 5.0V	
		3.2	6.0	$\mu A$	+25°C		
		3.4	7.0	$\mu A$	+85°C		
D026 ( $\Delta I_{AD}$ )	<b>A/D Converter</b>	0.001	2.0	$\mu A$	-40°C to +85°C	VDD = 2.0V	
		0.001	2.0	$\mu A$	-40°C to +85°C		VDD = 3.0V
		0.003	2.0	$\mu A$	-40°C to +85°C		VDD = 5.0V
		Extended devices	4.0	8.0	$\mu A$		

**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  $I_r = V_{DD}/2R_{EXT}$  (mA) with REXT in k $\Omega$ .

# PIC16F87/88

## 18.3 DC Characteristics: Internal RC Accuracy PIC16F87/88 (Industrial, Extended) PIC16LF87/88 (Industrial)

<b>PIC16LF87/88</b> (Industrial)		<b>Standard Operating Conditions (unless otherwise stated)</b> Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial					
<b>PIC16F87/88</b> (Industrial, Extended)		<b>Standard Operating Conditions (unless otherwise stated)</b> Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ for extended					
Param No.	Device	Min	Typ	Max	Units	Conditions	
<b>INTOSC Accuracy @ Freq = 8 MHz, 4 MHz, 2 MHz, 1 MHz, 500 kHz, 250 kHz, 125 kHz<sup>(1)</sup></b>							
	PIC16LF87/88	-2	$\pm 1$	2	%	+25°C	VDD = 2.7-3.3V
		-5	—	5	%	-10°C to +85°C	
		-10	—	10	%	-40°C to +85°C	
	PIC16F87/88	-2	$\pm 1$	2	%	25°C	VDD = 4.5-5.5V
		-5	—	5	%	-10°C to +85°C	
		-10	—	10	%	-40°C to +85°C	
	Extended devices	-15	—	15	%	-40°C to +125°C	VDD = 4.5-5.5V
<b>INTRC Accuracy @ Freq = 31 kHz<sup>(2)</sup></b>							
	PIC16LF87/88	26.562	—	35.938	kHz	-40°C to +85°C	VDD = 2.7-3.3V
	PIC16F87/88	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.  
**Note 2:** INTRC frequency after calibration.



## 18.4 DC Characteristics: PIC16F87/88 (Industrial, Extended) PIC16LF87/88 (Industrial)

DC CHARACTERISTICS			Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ for extended Operating voltage $V_{DD}$ range as described in DC Specification, Section 18.1 "DC Characteristics: Supply Voltage".				
Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
	$V_{IL}$	<b>Input Low Voltage</b>					
D030		I/O ports: with TTL buffer	$V_{SS}$	—	$0.15 V_{DD}$	V	For entire $V_{DD}$ range
D030A			$V_{SS}$	—	0.8V	V	$4.5V \leq V_{DD} \leq 5.5V$
D031		with Schmitt Trigger buffer	$V_{SS}$	—	$0.2 V_{DD}$	V	<b>(Note 1)</b>
D032		$\overline{\text{MCLR}}$ , OSC1 (in RC mode)	$V_{SS}$	—	$0.2 V_{DD}$	V	
D033		OSC1 (in XT and LP mode) OSC1 (in HS mode)	$V_{SS}$	—	0.3V $0.3 V_{DD}$	V	
D034		Ports RB1 and RB4: with Schmitt Trigger buffer	$V_{SS}$	—	$0.3 V_{DD}$	V	For entire $V_{DD}$ range
	$V_{IH}$	<b>Input High Voltage</b>					
D040		I/O ports: with TTL buffer	2.0	—	$V_{DD}$	V	$4.5V \leq V_{DD} \leq 5.5V$
D040A			$0.25 V_{DD} + 0.8V$	—	$V_{DD}$	V	For entire $V_{DD}$ range
D041		with Schmitt Trigger buffer	$0.8 V_{DD}$	—	$V_{DD}$	V	For entire $V_{DD}$ range
D042		$\overline{\text{MCLR}}$	$0.8 V_{DD}$	—	$V_{DD}$	V	<b>(Note 1)</b>
D042A		OSC1 (in XT and LP mode) OSC1 (in HS mode)	1.6V $0.7 V_{DD}$	—	$V_{DD}$	V	
D043		OSC1 (in RC mode)	$0.9 V_{DD}$	—	$V_{DD}$	V	
D044		Ports RB1 and RB4: with Schmitt Trigger buffer	$0.7 V_{DD}$	—	$V_{DD}$	V	For entire $V_{DD}$ range
D070		$I_{PURB}$	<b>PORTB Weak Pull-up Current</b>	50	250	400	$\mu\text{A}$
	$I_{IL}$	<b>Input Leakage Current (Notes 2, 3)</b>					
D060		I/O ports	—	—	$\pm 1$	$\mu\text{A}$	$V_{SS} \leq V_{PIN} \leq V_{DD}$ , pin at high-impedance
D061		$\overline{\text{MCLR}}$	—	—	$\pm 5$	$\mu\text{A}$	$V_{SS} \leq V_{PIN} \leq V_{DD}$
D063		OSC1	—	—	$\pm 5$	$\mu\text{A}$	$V_{SS} \leq V_{PIN} \leq V_{DD}$ , XT, HS and LP oscillator configuration

\* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

- Note 1:** In RC oscillator configuration, the OSC1/CLKI pin is a Schmitt Trigger input. It is not recommended that the PIC16F87/88 be driven with external clock in RC mode.
- 2:** The leakage current on the  $\overline{\text{MCLR}}$  pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.
- 3:** Negative current is defined as current sourced by the pin.

# PIC16F87/88

## 18.4 DC Characteristics: PIC16F87/88 (Industrial, Extended) PIC16LF87/88 (Industrial) (Continued)

DC CHARACTERISTICS		Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ for extended Operating voltage $V_{DD}$ range as described in DC Specification, Section 18.1 "DC Characteristics: Supply Voltage".					
Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
		<b>Output Low Voltage</b>					
D080	VOL	I/O ports	—	—	0.6	V	$I_{OL} = 8.5 \text{ mA}$ , $V_{DD} = 4.5\text{V}$ , $-40^{\circ}\text{C}$ to $+125^{\circ}\text{C}$
D083		OSC2/CLKO (RC oscillator configuration)	—	—	0.6	V	$I_{OL} = 1.6 \text{ mA}$ , $V_{DD} = 4.5\text{V}$ , $-40^{\circ}\text{C}$ to $+125^{\circ}\text{C}$
		<b>Output High Voltage</b>					
D090	VOH	I/O ports (Note 3)	$V_{DD} - 0.7$	—	—	V	$I_{OH} = -3.0 \text{ mA}$ , $V_{DD} = 4.5\text{V}$ , $-40^{\circ}\text{C}$ to $+125^{\circ}\text{C}$
D092		OSC2/CLKO (RC oscillator configuration)	$V_{DD} - 0.7$	—	—	V	$I_{OH} = -1.3 \text{ mA}$ , $V_{DD} = 4.5\text{V}$ , $-40^{\circ}\text{C}$ to $+125^{\circ}\text{C}$
		<b>Capacitive Loading Specs on Output Pins</b>					
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 <sup>2</sup> C™ mode	—	—	400	pF	
		<b>Data EEPROM Memory</b>					
D120	ED	Endurance	100K 10K	1M 100K	— —	E/W E/W	$-40^{\circ}\text{C}$ to $85^{\circ}\text{C}$ $+85^{\circ}\text{C}$ to $+125^{\circ}\text{C}$
D121	VDRW	$V_{DD}$ for Read/Write	$V_{MIN}$	—	5.5	V	Using EECON to read/write, $V_{MIN}$ = min. operating voltage
D122	TDEW	Erase/Write Cycle Time	—	4	8	ms	
		<b>Program Flash Memory</b>					
D130	EP	Endurance	10K 1K	100K 10K	— —	E/W E/W	$-40^{\circ}\text{C}$ to $85^{\circ}\text{C}$ $+85^{\circ}\text{C}$ to $+125^{\circ}\text{C}$
D131	VPR	$V_{DD}$ for Read	$V_{MIN}$	—	5.5	V	
D132A		$V_{DD}$ for Erase/Write	$V_{MIN}$	—	5.5	V	Using EECON to read/write, $V_{MIN}$ = min. operating voltage
D133	TPE	Erase Cycle Time	—	2	4	ms	
D134	TPW	Write Cycle Time	—	2	4	ms	

\* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

- Note 1:** In RC oscillator configuration, the OSC1/CLKI pin is a Schmitt Trigger input. It is not recommended that the PIC16F87/88 be driven with external clock in RC mode.
- 2:** The leakage current on the  $\overline{\text{MCLR}}$  pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.
- 3:** Negative current is defined as current sourced by the pin.

**TABLE 18-1: COMPARATOR SPECIFICATIONS**

Operating Conditions: $3.0V < V_{DD} < 5.5V$ , $-40^{\circ}C < T_A < +85^{\circ}C$ , unless otherwise stated							
Param No.	Sym	Characteristics	Min	Typ	Max	Units	Comments
D300	V <sub>IOFF</sub>	Input Offset Voltage	—	±5.0	±10	mV	
D301	V <sub>ICM</sub>	Input Common Mode Voltage*	0	—	$V_{DD} - 1.5$	V	
D302	CMRR	Common Mode Rejection Ratio*	55	—	—	dB	
300 300A	T <sub>RESP</sub>	Response Time <sup>(1)*</sup>	—	150	400 600	ns ns	PIC16F87/88 PIC16LF87/88
301	T <sub>MC2OV</sub>	Comparator Mode Change to Output Valid*	—	—	10	μs	

\* These parameters are characterized but not tested.

**Note 1:** Response time measured with one comparator input at  $(V_{DD} - 1.5)/2$  while the other input transitions from  $V_{SS}$  to  $V_{DD}$ .

**TABLE 18-2: VOLTAGE REFERENCE SPECIFICATIONS**

Operating Conditions: $3.0V < V_{DD} < 5.5V$ , $-40^{\circ}C < T_A < +85^{\circ}C$ , unless otherwise stated							
Spec No.	Sym	Characteristics	Min	Typ	Max	Units	Comments
D310	V <sub>RES</sub>	Resolution	$V_{DD}/24$	—	$V_{DD}/32$	LSb	
D311	V <sub>RAA</sub>	Absolute Accuracy	—	—	1/2 1/2	LSb LSb	Low Range (CVRR = 1) High Range (CVRR = 0)
D312	V <sub>RUR</sub>	Unit Resistor Value (R)*	—	2k	—	Ω	
310	T <sub>SET</sub>	Settling Time <sup>(1)*</sup>	—	—	10	μs	

\* These parameters are characterized but not tested.

**Note 1:** Settling time measured while  $CVRR = 1$  and  $CVR<3:0>$  transitions from '0000' to '1111'.

# PIC16F87/88

## 18.5 Timing Parameter Symbology

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

- |             |           |  |
|-------------|-----------|--|
| 1. TppS2ppS | 3. TCC:ST | (I <sup>2</sup> C specifications only) |
| 2. TppS     | 4. Ts     | (I <sup>2</sup> C specifications only) |

<b>T</b>		
F	Frequency	T Time

Lowercase letters (pp) and their meanings:

<b>pp</b>			
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	$\overline{WR}$

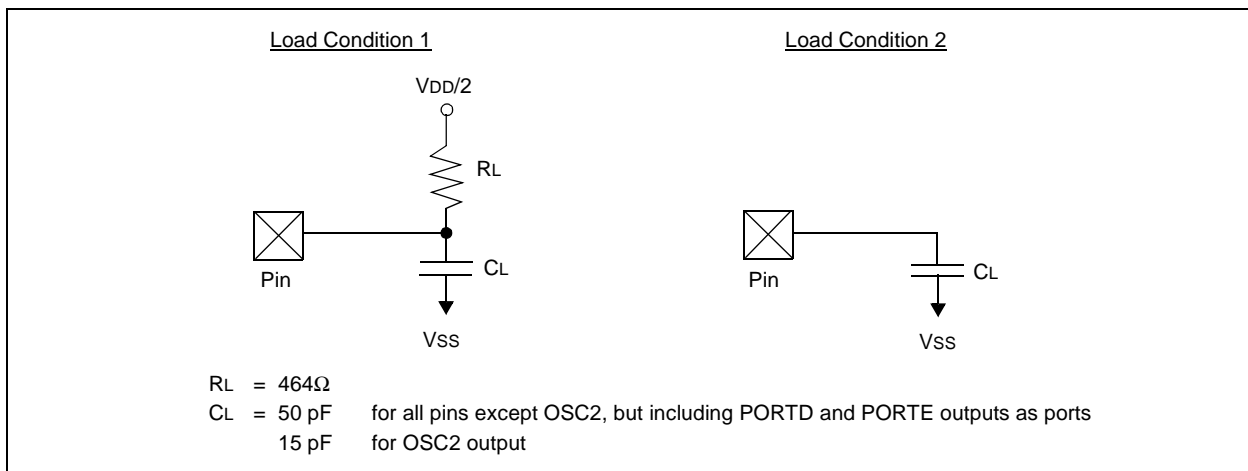
Uppercase letters and their meanings:

<b>S</b>			
F	Fall	P	Period
H	High	R	Rise
I	Invalid (High-impedance)	V	Valid
L	Low	Z	High-impedance
<b>I<sup>2</sup>C only</b>			
AA	output access	High	High
BUF	Bus free	Low	Low

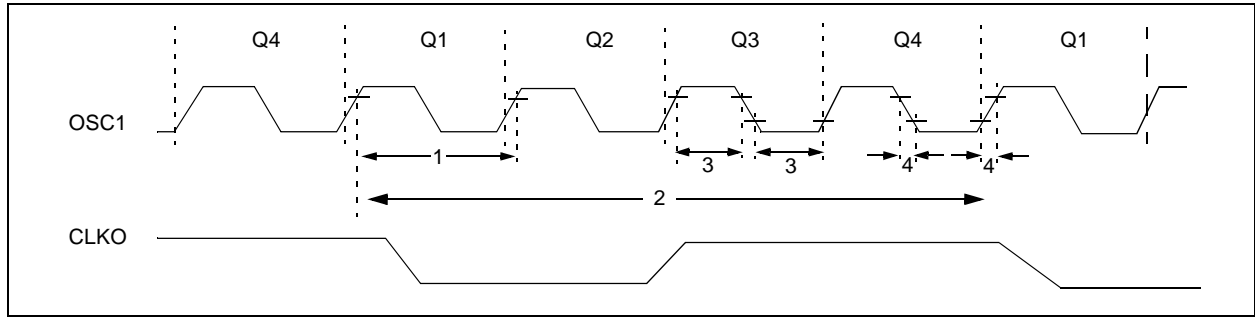
TCC:ST (I<sup>2</sup>C specifications only)

<b>CC</b>			
HD	Hold	SU	Setup
<b>ST</b>			
DAT	DATA input hold	STO	Stop condition
STA	Start condition		

**FIGURE 18-3: LOAD CONDITIONS**



**FIGURE 18-4: EXTERNAL CLOCK TIMING**



**TABLE 18-3: EXTERNAL CLOCK TIMING REQUIREMENTS**

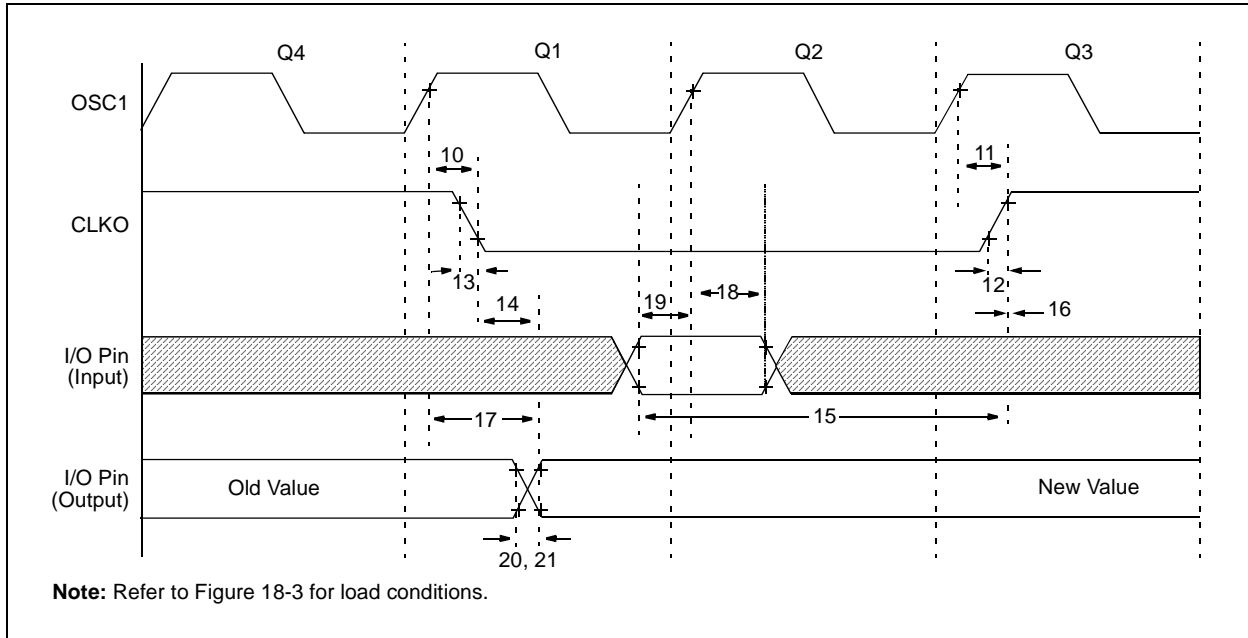
Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
	FOSC	External CLKI Frequency (Note 1)	DC	—	1	MHz	XT and RC Oscillator mode
			DC	—	20	MHz	HS Oscillator mode
			DC	—	32	kHz	LP Oscillator mode
	Oscillator Frequency (Note 1)	DC	—	4	MHz	RC Oscillator mode	
		0.1	—	4	MHz	XT Oscillator mode	
		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 modes
			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.

# PIC16F87/88

**FIGURE 18-5: CLKO AND I/O TIMING**



**TABLE 18-4: 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)	PIC16F87/88	100	—	—	ns	
			PIC16LF87/88	200	—	—	ns	
19*	TioV2osH	Port Input Valid to OSC1 ↑ (I/O in setup time)	0	—	—	ns		
20*	TioR	Port Output Rise Time	PIC16F87/88	—	10	40	ns	
			PIC16LF87/88	—	—	145	ns	
21*	TioF	Port Output Fall Time	PIC16F87/88	—	10	40	ns	
			PIC16LF87/88	—	—	145	ns	
22††*	TINP	INT Pin High or Low Time	TCY	—	—	ns		
23††*	TRBP	RB7:RB4 Change INT High or Low Time	TCY	—	—	ns		

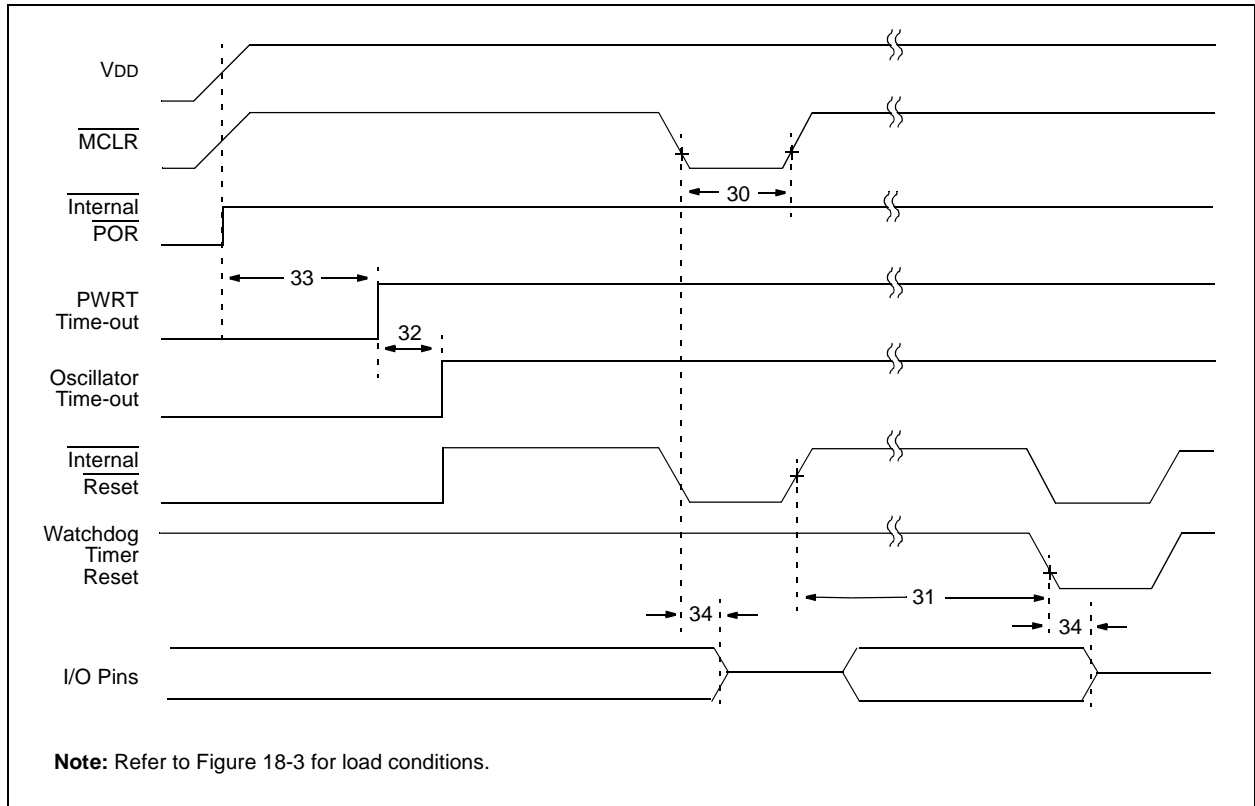
\* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

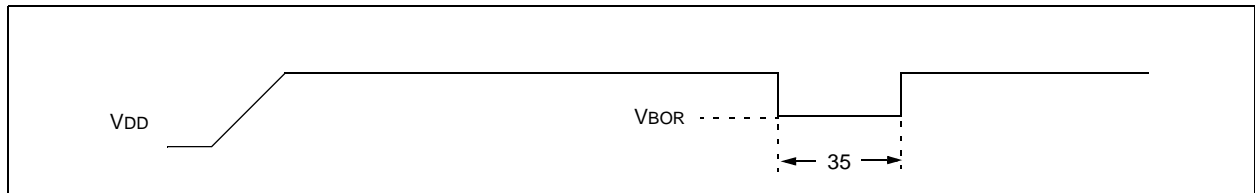
†† These parameters are asynchronous events not related to any internal clock edges.

**Note 1:** Measurements are taken in RC mode where CLKO output is 4 x Tosc.

**FIGURE 18-6: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING**



**FIGURE 18-7: BROWN-OUT RESET TIMING**



**TABLE 18-5: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER, POWER-UP TIMER AND BROWN-OUT RESET REQUIREMENTS**

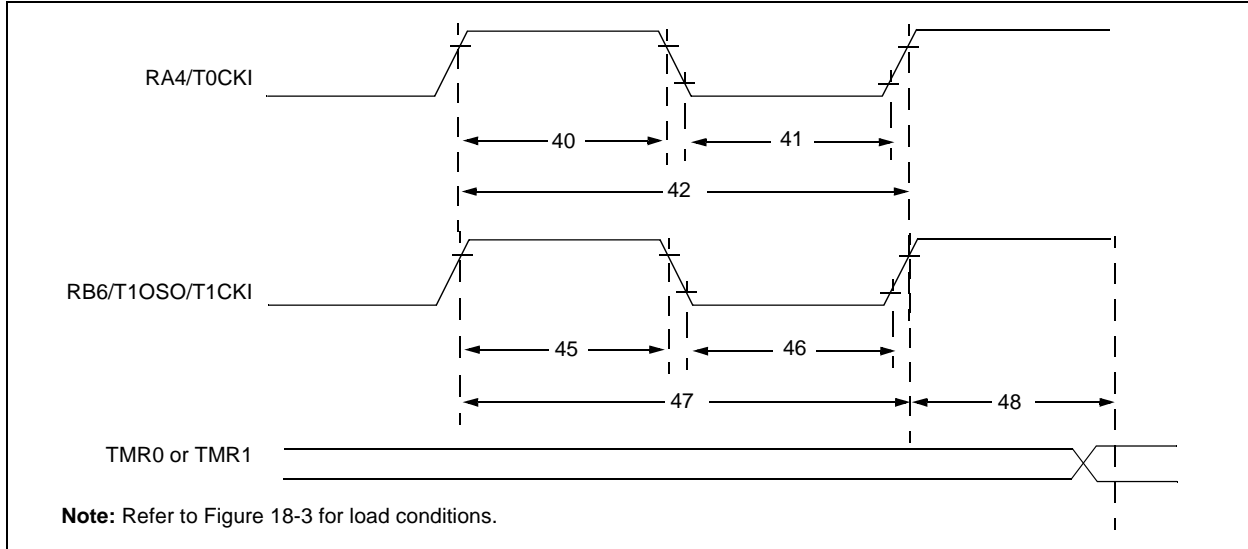
Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
30	TmCL	MCLR Pulse Width (Low)	2	—	—	μs	VDD = 5V, -40°C to +85°C
31*	TWDT	Watchdog Timer Time-out Period (16-bit prescaler = 0100 and no postscaler)	13.6	16	18.4	ms	VDD = 5V, -40°C to +85°C
32	TOST	Oscillation Start-up Timer Period	—	1024 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.

# PIC16F87/88

**FIGURE 18-8: TIMER0 AND TIMER1 EXTERNAL CLOCK TIMINGS**



**TABLE 18-6: TIMER0 AND TIMER1 EXTERNAL CLOCK REQUIREMENTS**

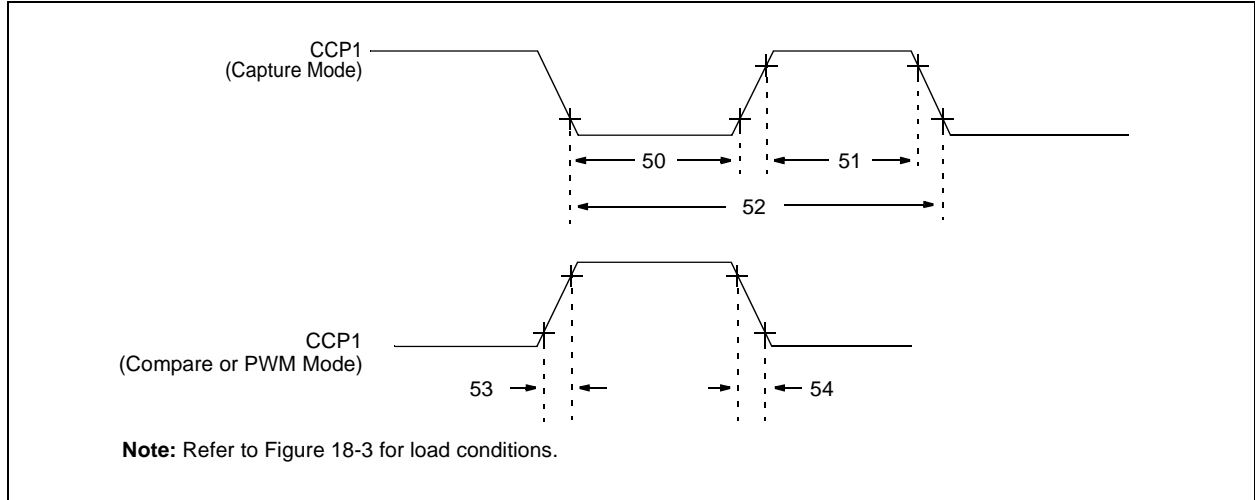
Param No.	Symbol	Characteristic		Min	Typ†	Max	Units	Conditions	
40*	Tt0H	T0CKI High Pulse Width	No Prescaler	$0.5 T_{CY} + 20$	—	—	ns	Must also meet parameter 42	
			With Prescaler	10	—	—	ns		
41*	Tt0L	T0CKI Low Pulse Width	No Prescaler	$0.5 T_{CY} + 20$	—	—	ns	Must also meet parameter 42	
			With Prescaler	10	—	—	ns		
42*	Tt0P	T0CKI Period	No Prescaler	$T_{CY} + 40$	—	—	ns		
			With Prescaler	Greater of: $20$ or $\frac{T_{CY} + 40}{N}$	—	—	ns		N = prescale value (2, 4, ..., 256)
45*	Tt1H	T1CKI High Time	Synchronous, Prescaler = 1	$0.5 T_{CY} + 20$	—	—	ns	Must also meet parameter 47	
			Synchronous, Prescaler = 2, 4, 8	PIC16F87/88	15	—	—		ns
				PIC16LF87/88	25	—	—		ns
			Asynchronous	PIC16F87/88	30	—	—		ns
PIC16LF87/88	50	—		—	ns				
46*	Tt1L	T1CKI Low Time	Synchronous, Prescaler = 1	$0.5 T_{CY} + 20$	—	—	ns	Must also meet parameter 47	
			Synchronous, Prescaler = 2, 4, 8	PIC16F87/88	15	—	—		ns
				PIC16LF87/88	25	—	—		ns
			Asynchronous	PIC16F87/88	30	—	—		ns
PIC16LF87/88	50	—		—	ns				
47*	Tt1P	T1CKI Input Period	Synchronous	PIC16F87/88	Greater of: $30$ or $\frac{T_{CY} + 40}{N}$	—	—	ns	N = prescale value (1, 2, 4, 8)
				PIC16LF87/88	Greater of: $50$ or $\frac{T_{CY} + 40}{N}$	—	—	ns	N = prescale value (1, 2, 4, 8)
			Asynchronous	PIC16F87/88	60	—	—	ns	
				PIC16LF87/88	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		$2 T_{OSC}$	—	$7 T_{OSC}$	—		

\* 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 18-9: CAPTURE/COMPARE/PWM TIMINGS (CCP1)**



**TABLE 18-7: CAPTURE/COMPARE/PWM REQUIREMENTS (CCP1)**

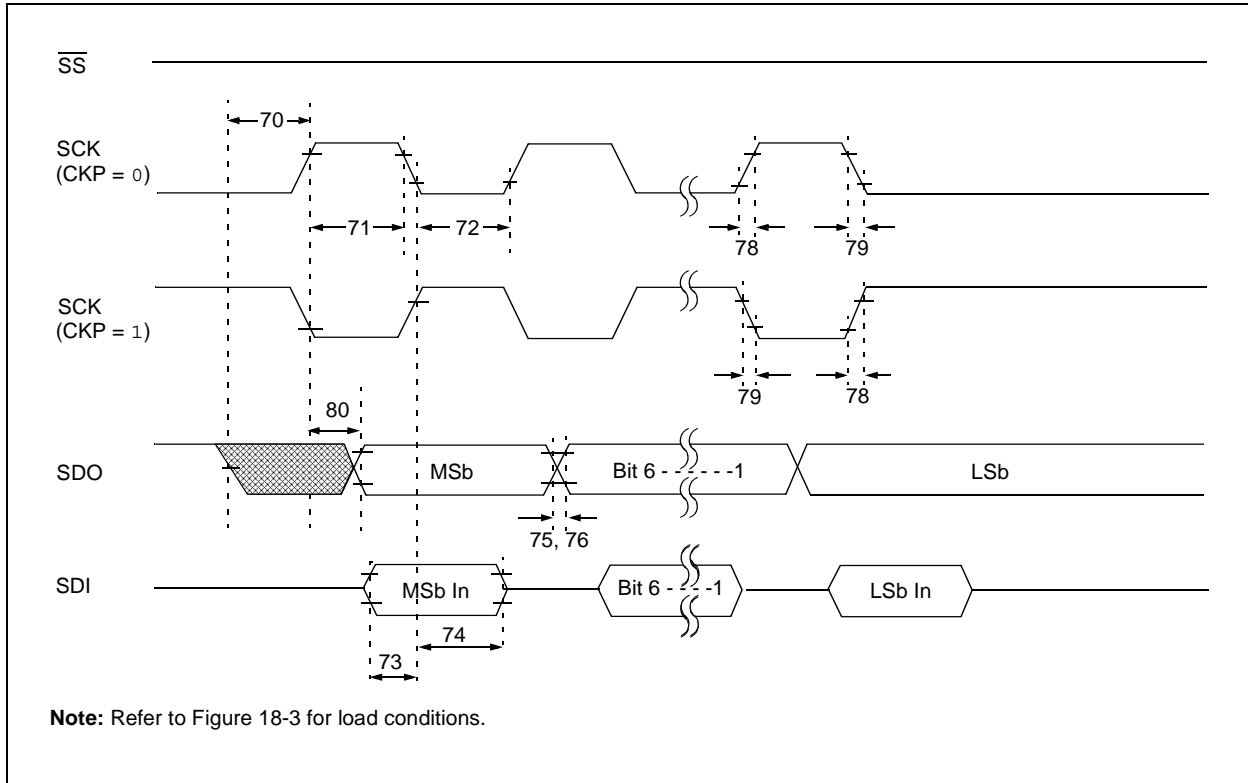
Param No.	Symbol	Characteristic		Min	Typ†	Max	Units	Conditions	
50*	TccL	CCP1 Input Low Time	No Prescaler	$0.5 T_{CY} + 20$	—	—	ns		
			With Prescaler	PIC16F87/88	10	—	—		ns
				PIC16LF87/88	20	—	—		ns
51*	TccH	CCP1 Input High Time	No Prescaler	$0.5 T_{CY} + 20$	—	—	ns		
			With Prescaler	PIC16F87/88	10	—	—		ns
				PIC16LF87/88	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	PIC16F87/88	—	10	25	ns		
			PIC16LF87/88	—	25	50	ns		
54*	TccF	CCP1 Output Fall Time	PIC16F87/88	—	10	25	ns		
			PIC16LF87/88	—	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.

# PIC16F87/88

**FIGURE 18-10: SPI™ MASTER MODE TIMING (CKE = 0, SMP = 0)**



**FIGURE 18-11: SPI™ MASTER MODE TIMING (CKE = 1, SMP = 1)**

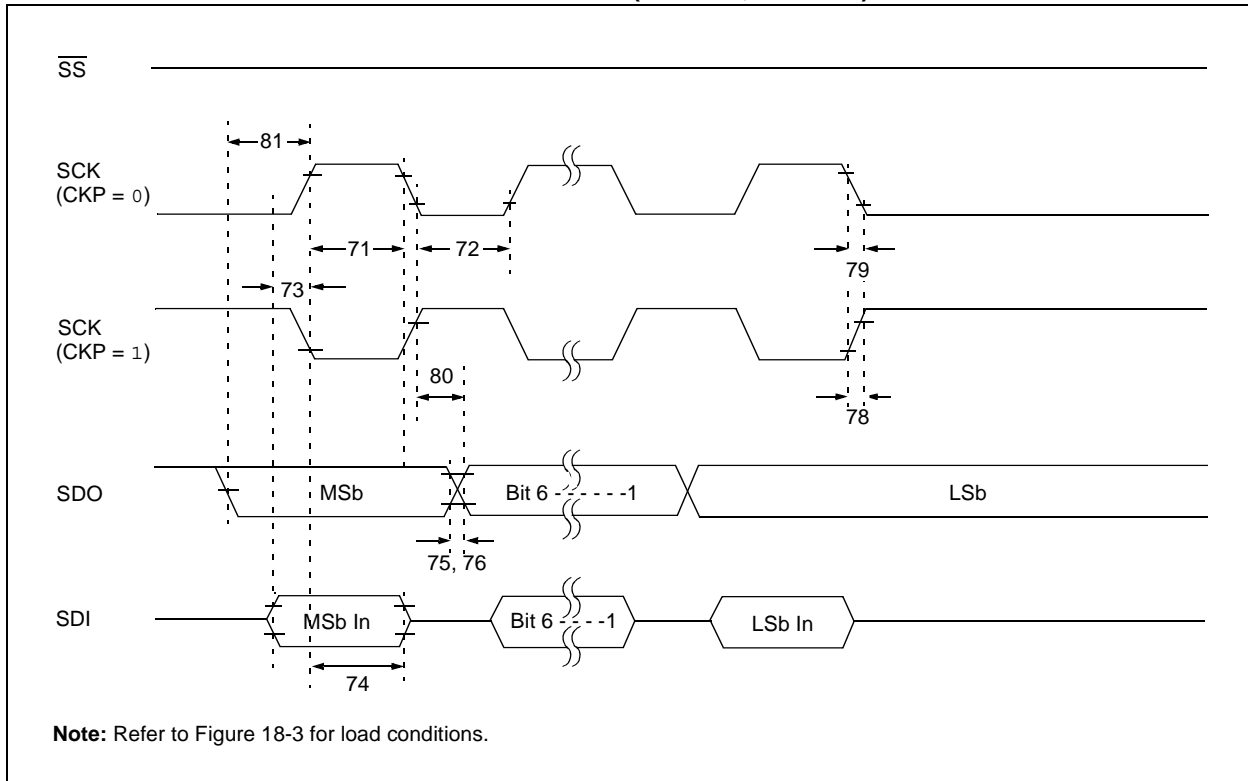


FIGURE 18-12: SPI™ SLAVE MODE TIMING (CKE = 0)

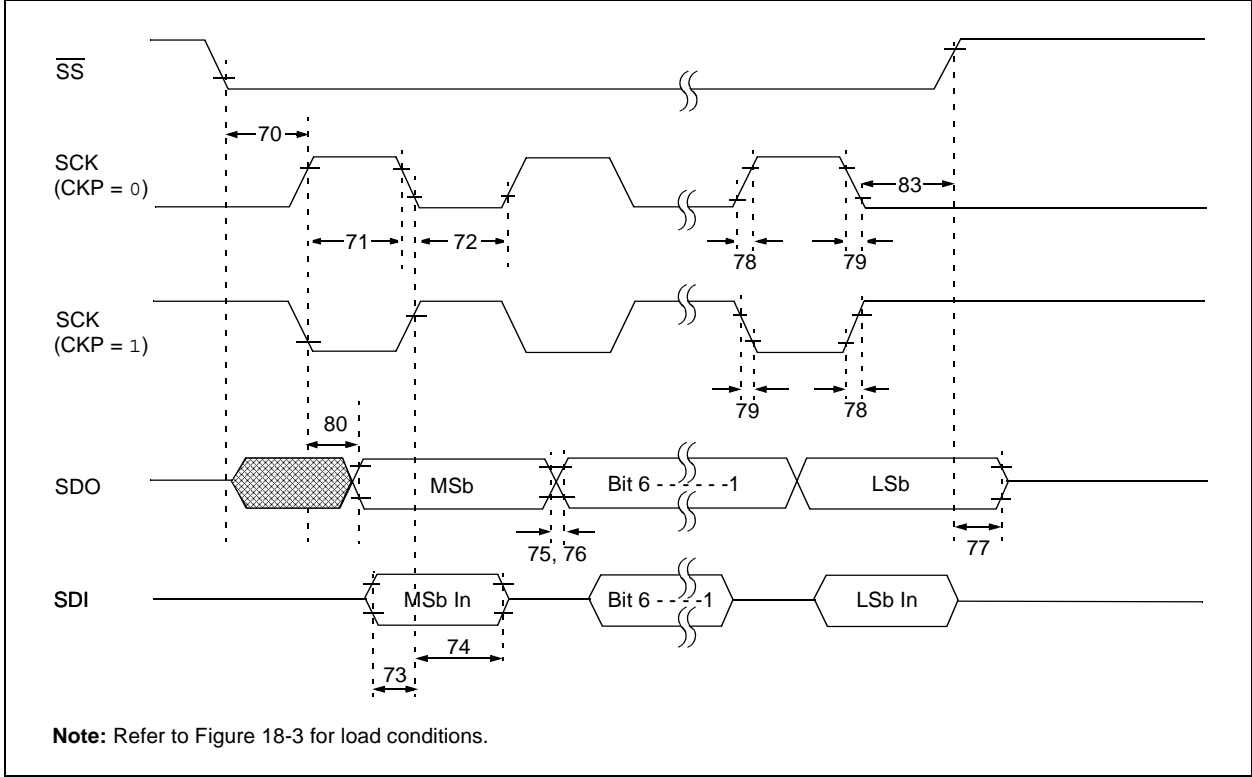
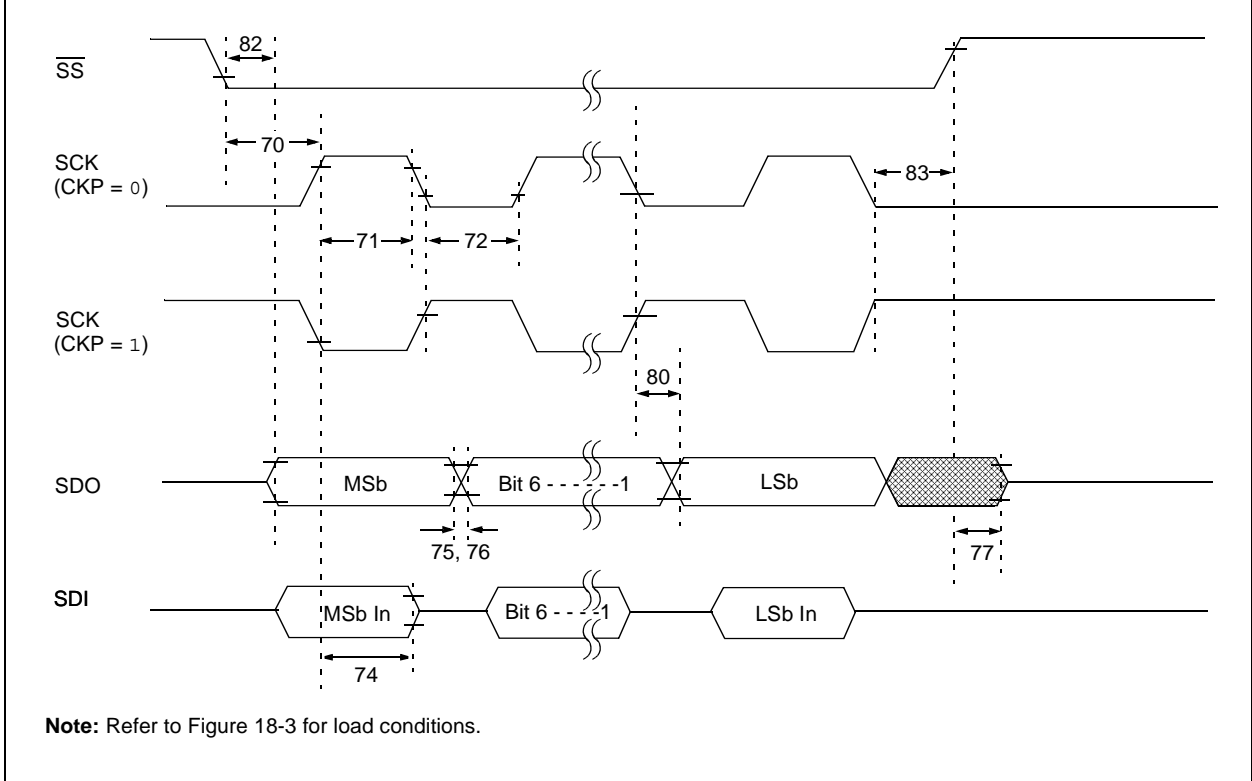


FIGURE 18-13: SPI™ SLAVE MODE TIMING (CKE = 1)



# PIC16F87/88

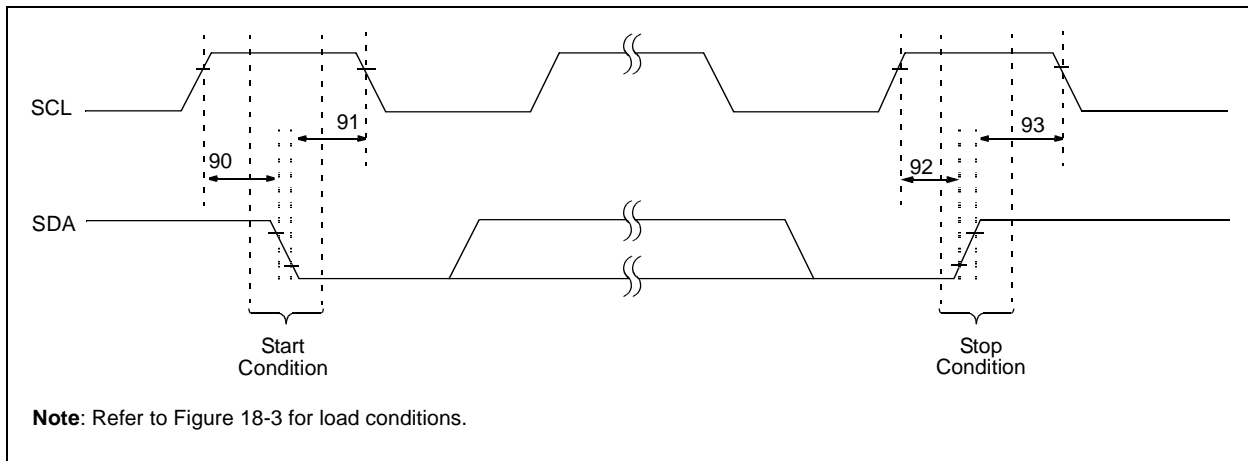
**TABLE 18-8: SPI™ MODE REQUIREMENTS**

Param No.	Symbol	Characteristic	Min	Typ†	Max	Units	Conditions
70*	TssL2scH, TssL2scL	$\overline{SS}$ ↓ to SCK ↓ or SCK ↑ Input	T <sub>CY</sub>	—	—	ns	
71*	TscH	SCK Input High Time (Slave mode)	T <sub>CY</sub> + 20	—	—	ns	
72*	TscL	SCK Input Low Time (Slave mode)	T <sub>CY</sub> + 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	—	10	25	ns	PIC16F87/88 PIC16LF87/88
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)	—	10	25	ns	PIC16F87/88 PIC16LF87/88
79*	TscF	SCK Output Fall Time (Master mode)	—	10	25	ns	
80*	Tsch2doV, TscL2doV	SDO Data Output Valid after SCK Edge	—	—	50	ns	PIC16F87/88 PIC16LF87/88
81*	TdoV2scH, TdoV2scL	SDO Data Output Setup to SCK Edge	T <sub>CY</sub>	—	—	ns	
82*	TssL2doV	SDO Data Output Valid after $\overline{SS}$ ↓ Edge	—	—	50	ns	
83*	Tsch2ssH, TscL2ssH	$\overline{SS}$ ↑ after SCK Edge	1.5 T <sub>CY</sub> + 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 18-14: I<sup>2</sup>C™ BUS START/STOP BITS TIMING**

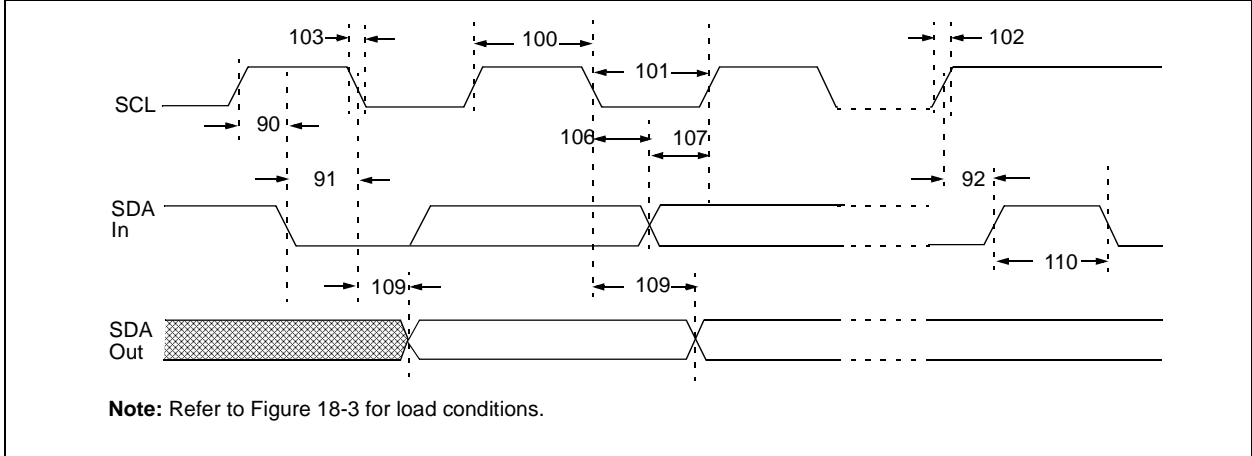


**TABLE 18-9: I<sup>2</sup>C™ BUS START/STOP BITS REQUIREMENTS**

Param No.	Symbol	Characteristic	Min	Typ	Max	Units	Conditions	
90*	TSU:STA	Start Condition	100 kHz mode	4700	—	—	ns	Only relevant for Repeated Start condition
		Setup Time	400 kHz mode	600	—	—		
91*	THD:STA	Start Condition	100 kHz mode	4000	—	—	ns	After this period, the first clock pulse is generated
		Hold Time	400 kHz mode	600	—	—		
92*	TSU:STO	Stop Condition	100 kHz mode	4700	—	—	ns	
		Setup Time	400 kHz mode	600	—	—		
93	THD:STO	Stop Condition	100 kHz mode	4000	—	—	ns	
		Hold Time	400 kHz mode	600	—	—		

\* These parameters are characterized but not tested.

**FIGURE 18-15: I<sup>2</sup>C™ BUS DATA TIMING**



# PIC16F87/88

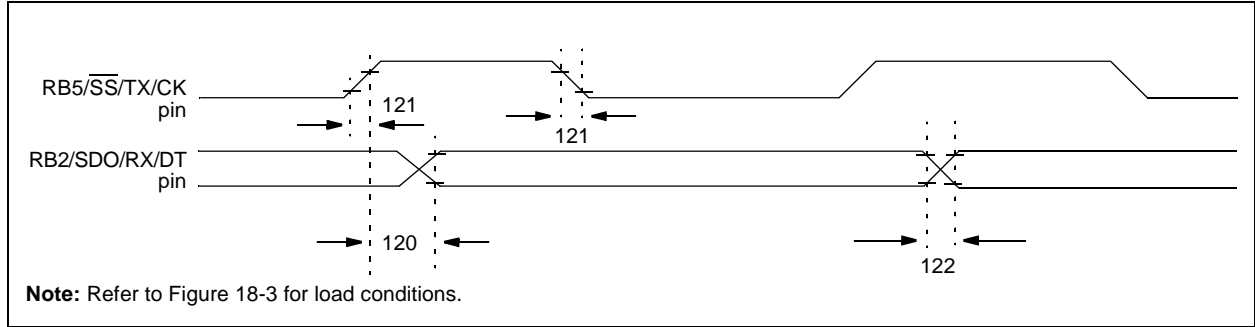
**TABLE 18-10: I<sup>2</sup>C™ BUS DATA REQUIREMENTS**

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	<b>(Note 2)</b>
			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	<b>(Note 1)</b>
			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) I<sup>2</sup>C™ bus device can be used in a Standard mode (100 kHz) I<sup>2</sup>C 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 I<sup>2</sup>C bus specification), before the SCL line is released.

**FIGURE 18-16: AUSART SYNCHRONOUS TRANSMISSION (MASTER/SLAVE) TIMING**

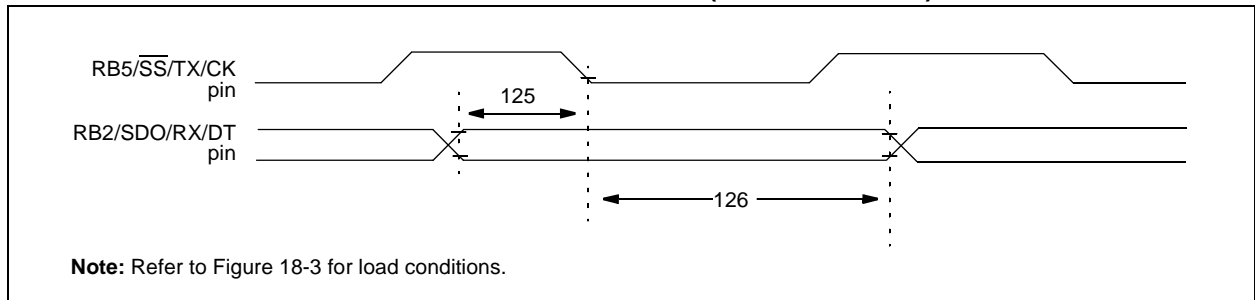


**TABLE 18-11: AUSART SYNCHRONOUS TRANSMISSION REQUIREMENTS**

Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
120	TckH2dtV	SYNC XMIT (MASTER & SLAVE) Clock High to Data Out Valid	PIC16F87/88	—	—	80	ns
		PIC16LF87/88	—	—	100	ns	
121	Tckrf	Clock Out Rise Time and Fall Time (Master mode)	PIC16F87/88	—	—	45	ns
			PIC16LF87/88	—	—	50	ns
122	Tdtrf	Data Out Rise Time and Fall Time	PIC16F87/88	—	—	45	ns
			PIC16LF87/88	—	—	50	ns

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**FIGURE 18-17: AUSART SYNCHRONOUS RECEIVE (MASTER/SLAVE) TIMING**



**TABLE 18-12: AUSART SYNCHRONOUS RECEIVE REQUIREMENTS**

Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
125	TdtV2ckL	SYNC RCV (MASTER & SLAVE) Data Setup before CK ↓ (DT setup time)	15	—	—	ns	
		126	TckL2dtl	Data Hold after CK ↓ (DT hold time)	15	—	—

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

# PIC16F87/88

**TABLE 18-13: A/D CONVERTER CHARACTERISTICS: PIC16F87/88 (INDUSTRIAL, EXTENDED)  
PIC16LF87/88 (INDUSTRIAL)**

Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions	
A01	NR	Resolution	—	—	10-bit	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	E0FF	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 <sup>(3)</sup>	—	—	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	<b>(Note 4)</b>	
A40	IAD	A/D Conversion Current (VDD)	PIC16F87/88	—	220	—	μA	Average current consumption when A/D is on <b>(Note 1)</b>
			PIC16LF87/88	—	90	—	μA	
A50	IREF	VREF Input Current <b>(Note 2)</b>	—	—	5	μA	During VAIN acquisition. Based on differential of VHOLD to VAIN to charge CHOLD, see <b>Section 12.1 “A/D Acquisition Requirements”</b> . During A/D conversion cycle	
			—	—	150	μA		

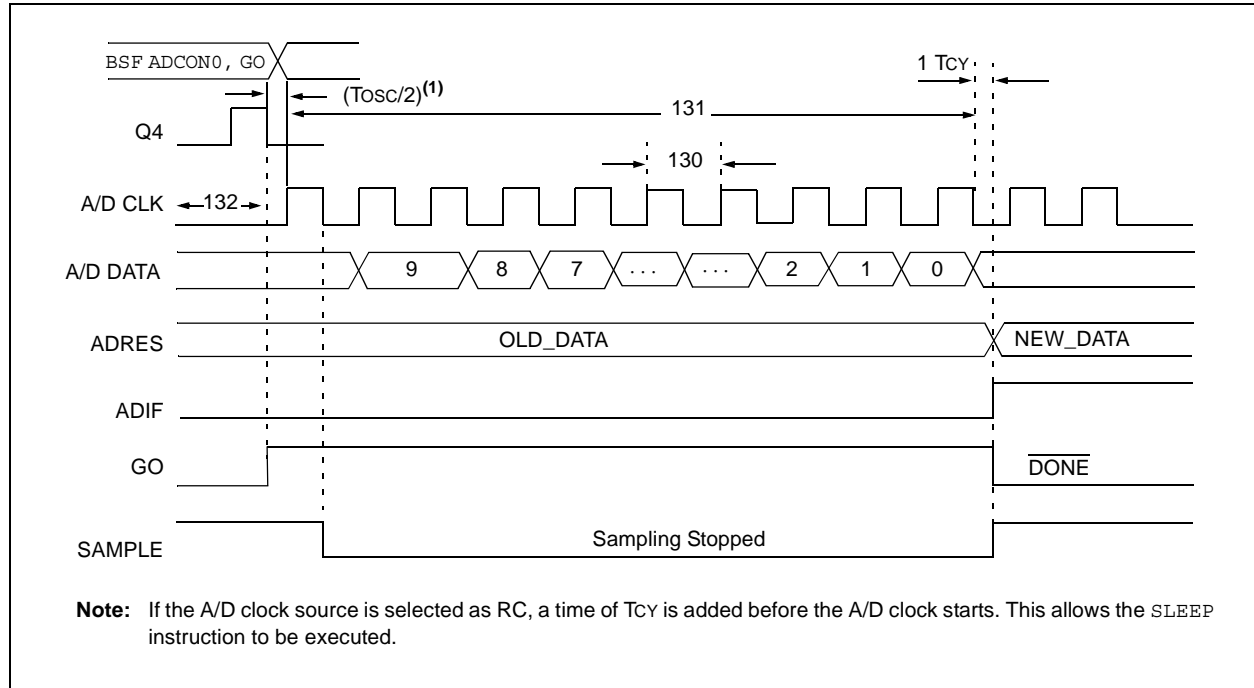
\* These parameters are characterized but not tested.

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

- Note 1:** When A/D is off, it will not consume any current other than minor leakage current. The power-down current specification 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.



**FIGURE 18-18: A/D CONVERSION TIMING**



**TABLE 18-14: A/D CONVERSION REQUIREMENTS**

Param No.	Symbol	Characteristic	Min	Typ†	Max	Units	Conditions	
130	TAD	A/D Clock Period	PIC16F87/88	1.6	—	—	$\mu\text{s}$	TOSC based, $V_{REF} \geq 3.0\text{V}$
			PIC16LF87/88	3.0	—	—	$\mu\text{s}$	TOSC based, $V_{REF} \geq 2.0\text{V}$
			PIC16F87/88	2.0	4.0	6.0	$\mu\text{s}$	A/D RC mode
			PIC16LF87/88	3.0	6.0	9.0	$\mu\text{s}$	A/D RC mode
131	TCNV	Conversion Time (not including S/H time) <b>(Note 1)</b>		—	12	TAD		
132	TACQ	Acquisition Time	<b>(Note 2)</b>	40	—	$\mu\text{s}$	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).	
			10*	—	—	$\mu\text{s}$		
134	TGO	Q4 to A/D Clock Start	—	$T_{OSC}/2$ §	—	—	If the A/D clock source is selected as RC, a time of $T_{CY}$ is added before the A/D clock starts. This allows the <code>SLEEP</code> 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 registers may be read on the following  $T_{CY}$  cycle.

**Note 2:** See **Section 12.1 “A/D Acquisition Requirements”** for minimum conditions.

# PIC16F87/88

---

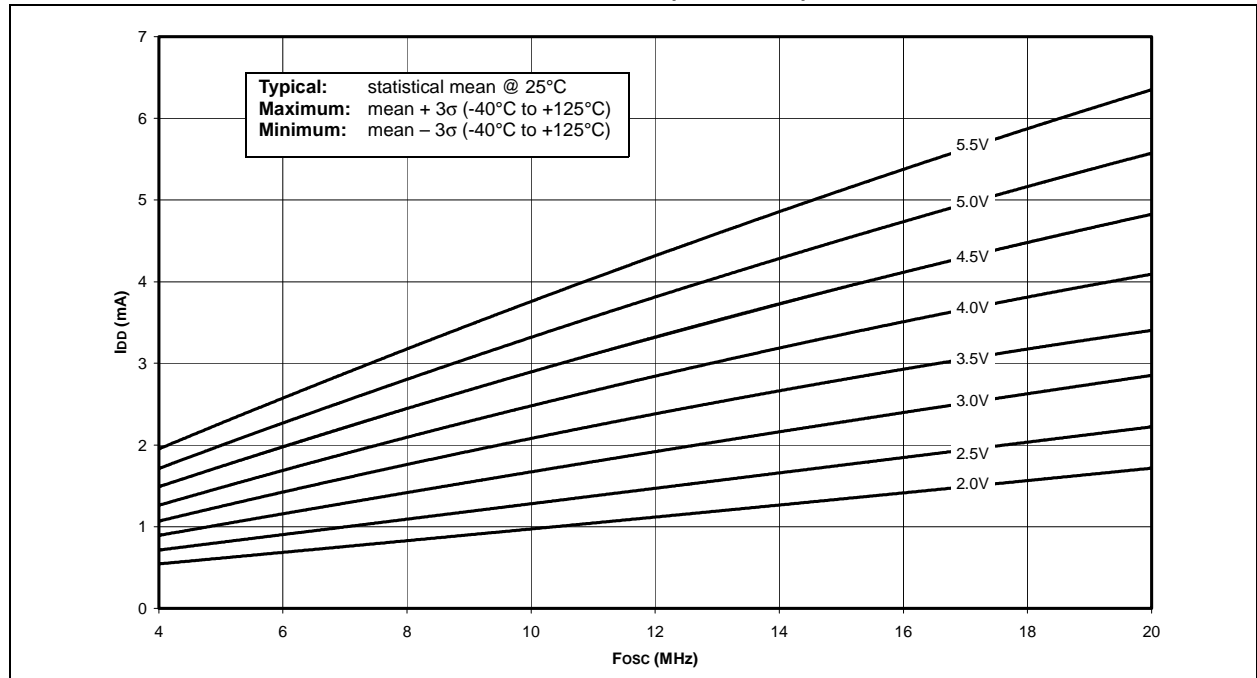
NOTES:

## 19.0 DC AND AC CHARACTERISTICS GRAPHS AND TABLES

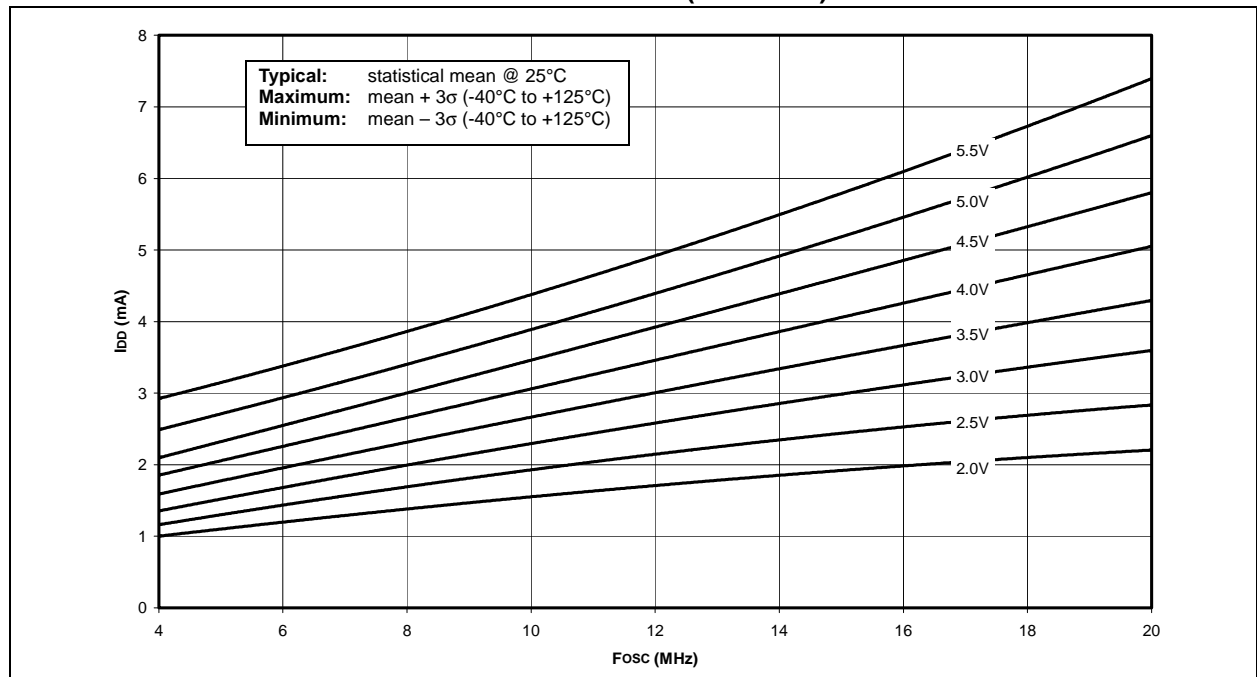
**Note:** The graphs and tables provided following this note are a statistical summary based on a limited number of samples and are provided for informational purposes only. The performance characteristics listed herein are not tested or guaranteed. In some graphs or tables, the data presented may be 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 19-1: TYPICAL I<sub>DD</sub> vs. Fosc OVER V<sub>DD</sub> (HS MODE)**



**FIGURE 19-2: MAXIMUM I<sub>DD</sub> vs. Fosc OVER V<sub>DD</sub> (HS MODE)**



# PIC16F87/88

FIGURE 19-3: TYPICAL  $I_{DD}$  vs.  $F_{osc}$  OVER  $V_{DD}$  (XT MODE)

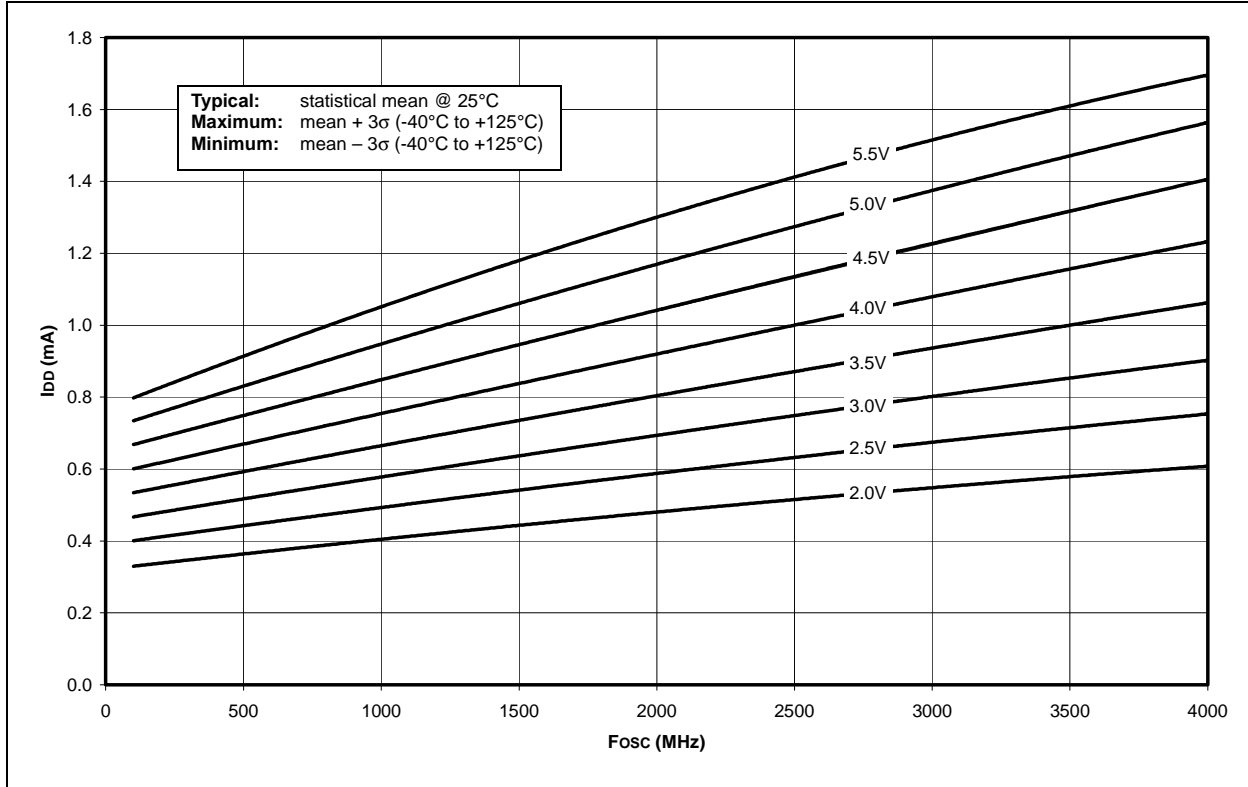
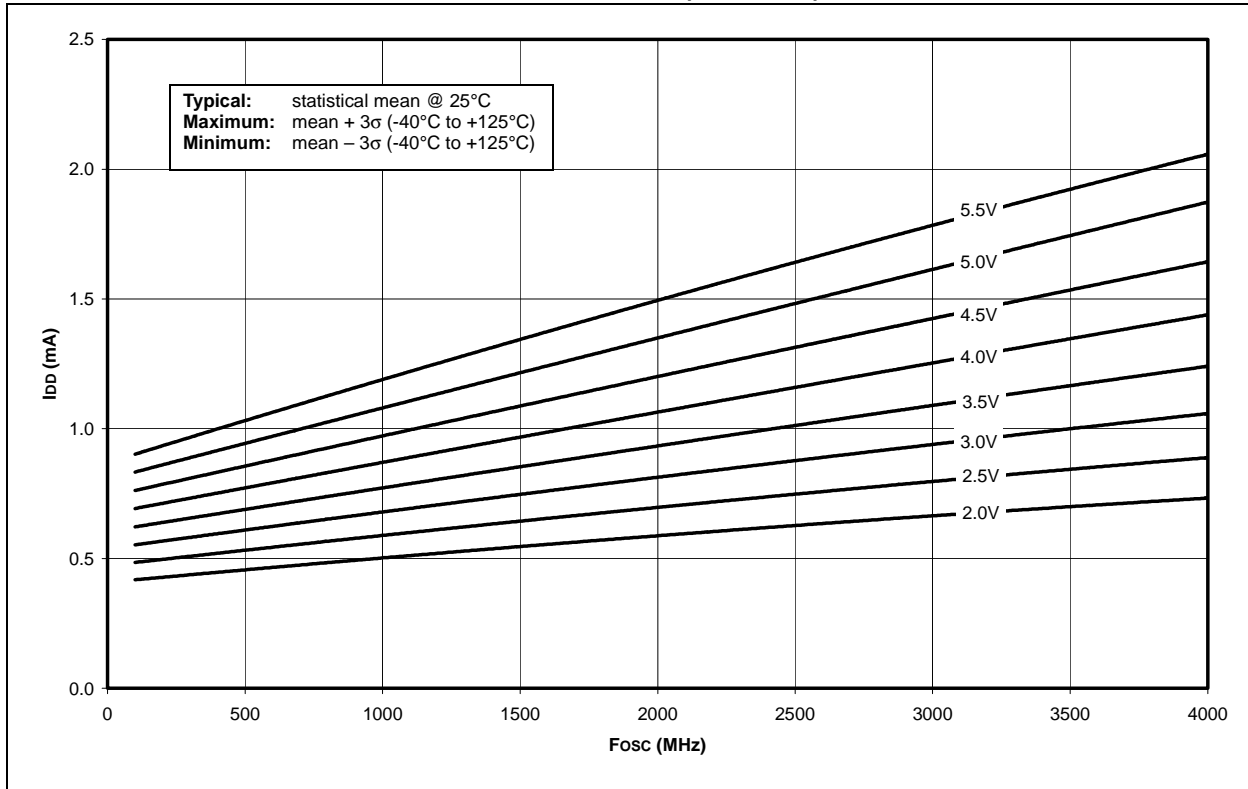
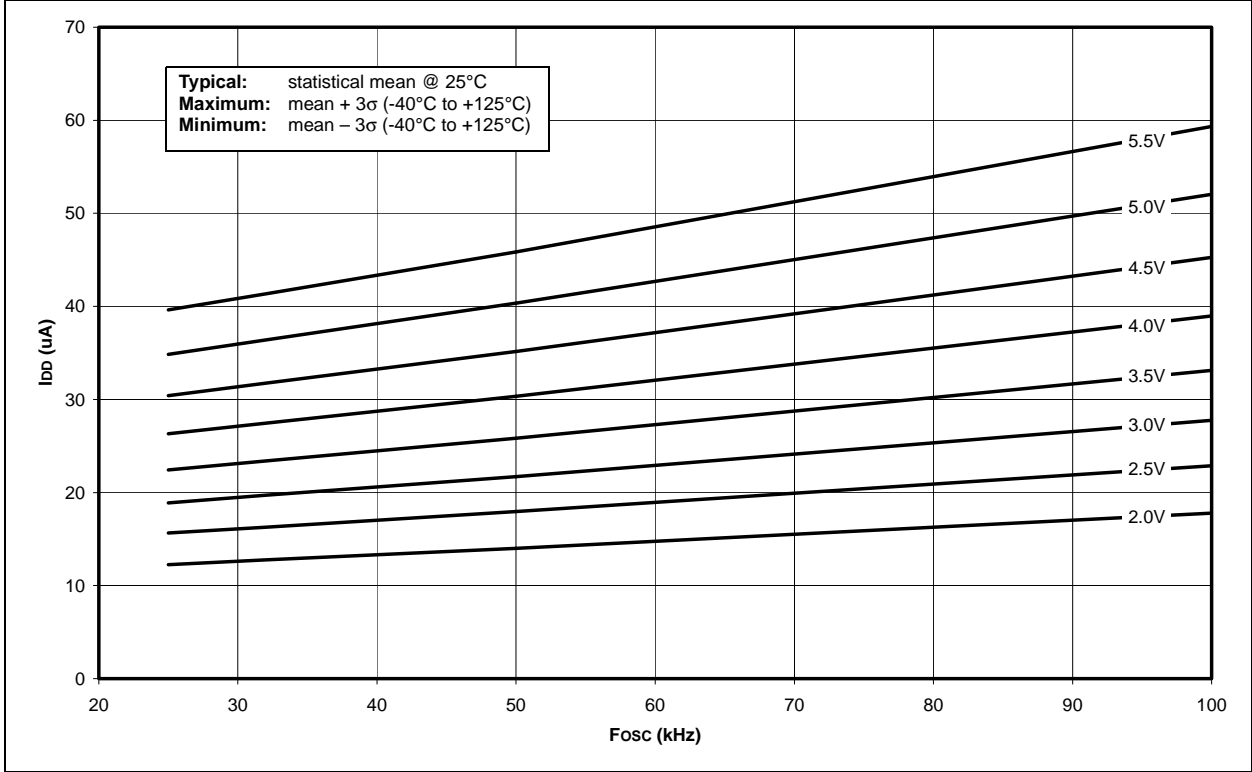


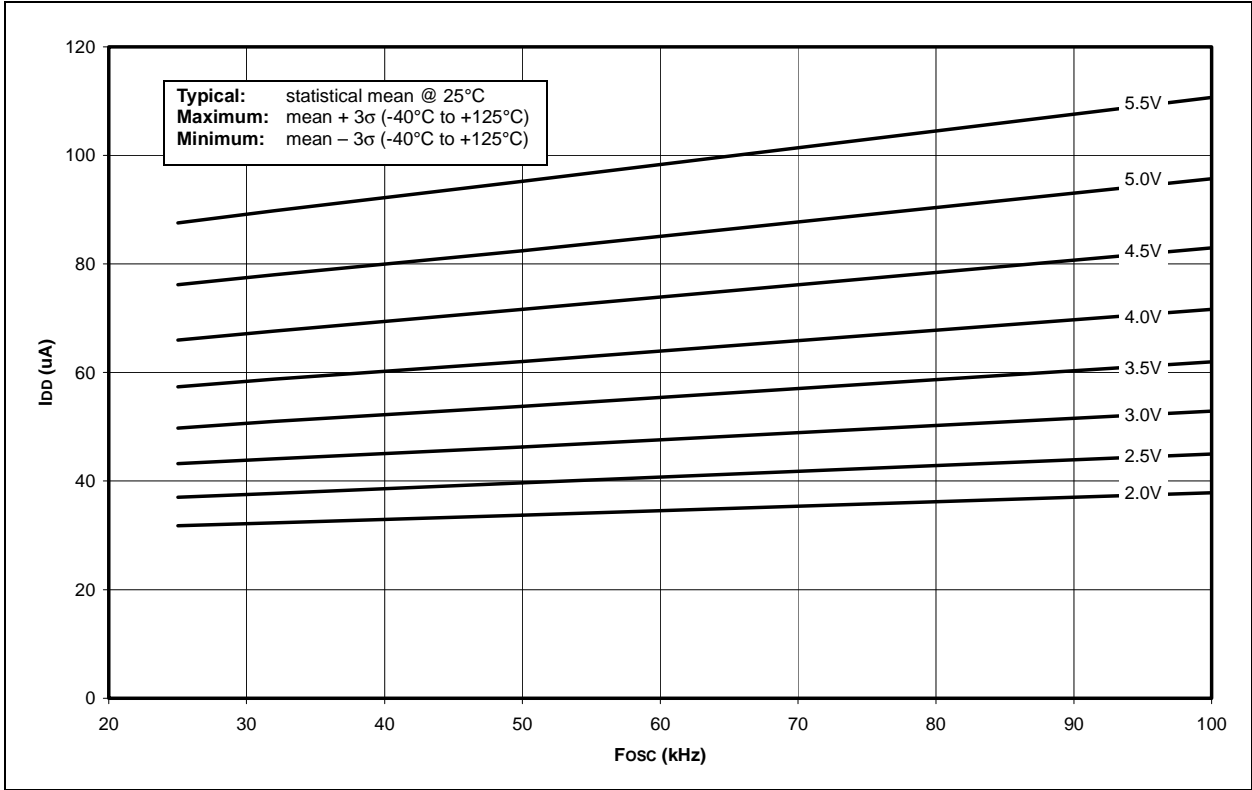
FIGURE 19-4: MAXIMUM  $I_{DD}$  vs.  $F_{osc}$  OVER  $V_{DD}$  (XT MODE)



**FIGURE 19-5: TYPICAL  $I_{DD}$  vs.  $F_{OSC}$  OVER  $V_{DD}$  (LP MODE)**

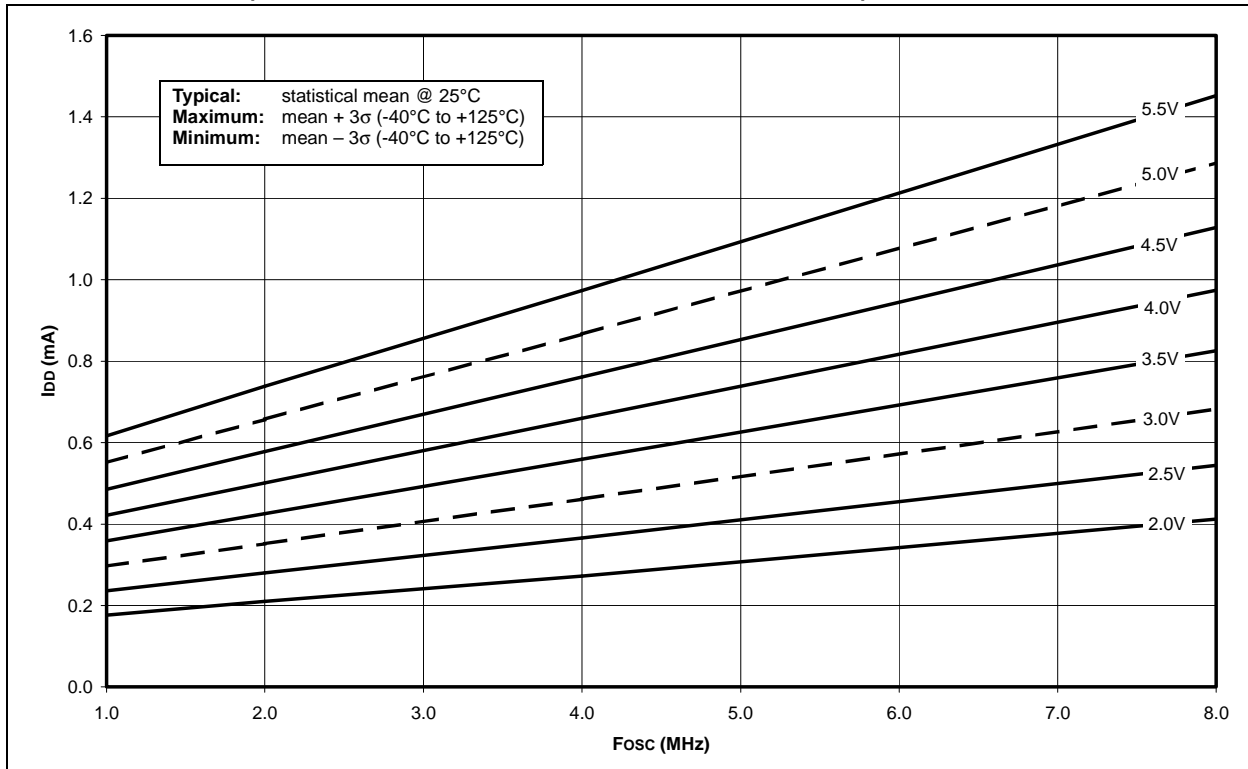


**FIGURE 19-6: MAXIMUM  $I_{DD}$  vs.  $F_{OSC}$  OVER  $V_{DD}$  (LP MODE)**

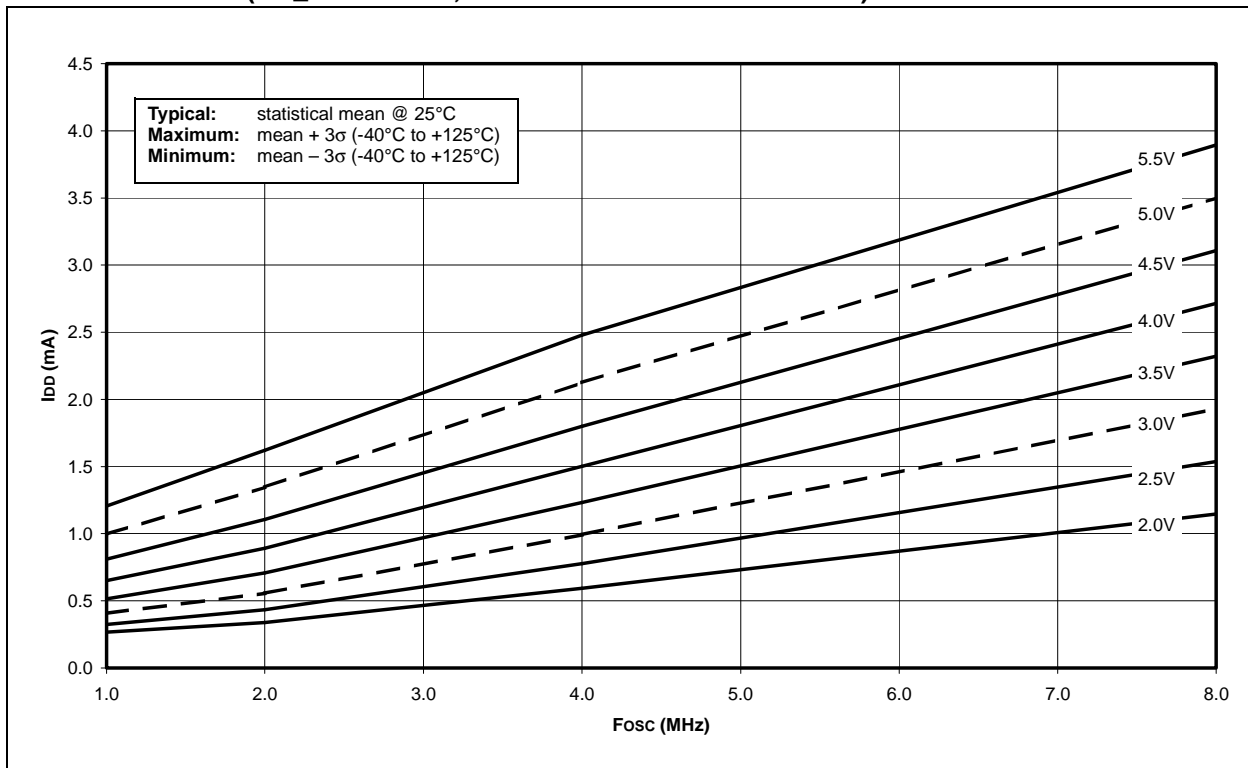


# PIC16F87/88

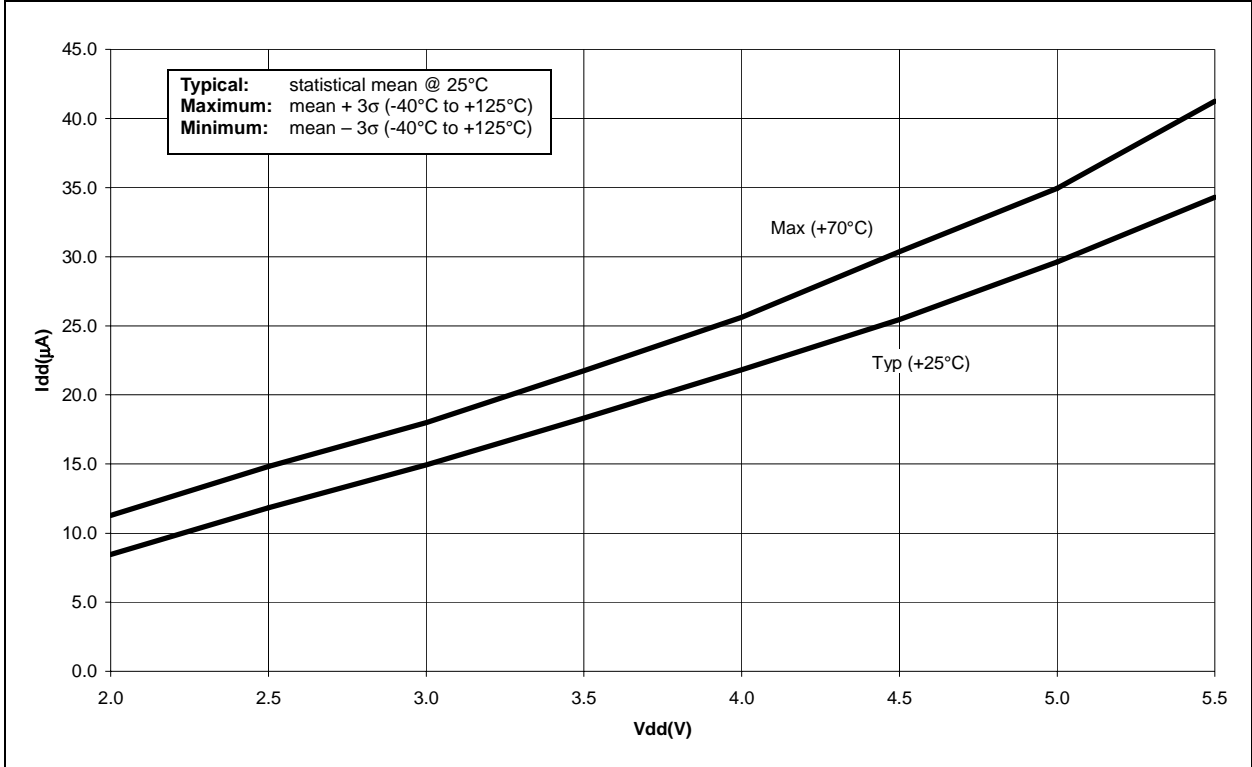
**FIGURE 19-7: TYPICAL  $I_{DD}$  vs.  $V_{DD}$ , -40°C TO +125°C, 1 MHz TO 8 MHz (RC\_RUN MODE, ALL PERIPHERALS DISABLED)**



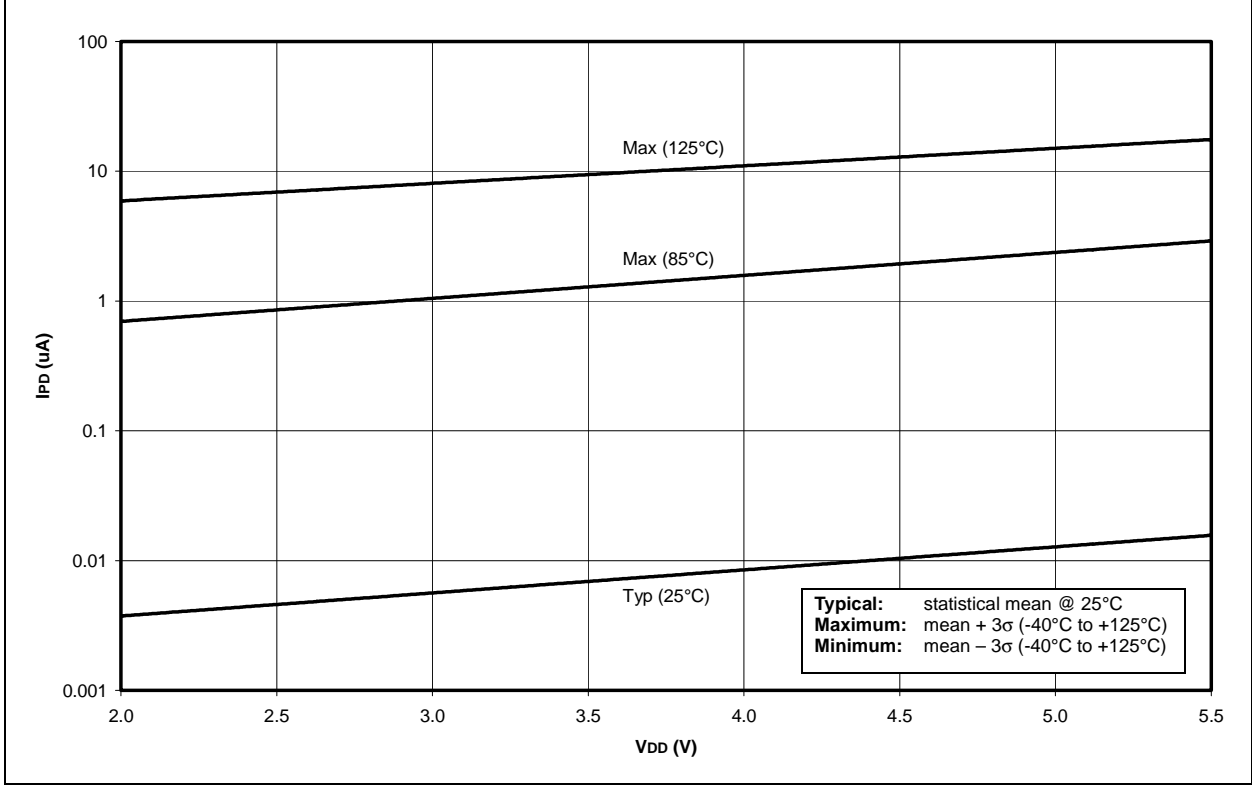
**FIGURE 19-8: MAXIMUM  $I_{DD}$  vs.  $V_{DD}$ , -40°C TO +125°C, 1 MHz TO 8 MHz (RC\_RUN MODE, ALL PERIPHERALS DISABLED)**



**FIGURE 19-9: I<sub>DD</sub> vs. V<sub>DD</sub>, SEC\_RUN MODE, -10°C TO +125°C, 32.768 kHz  
(XTAL 2 x 22 pF, ALL PERIPHERALS DISABLED)**

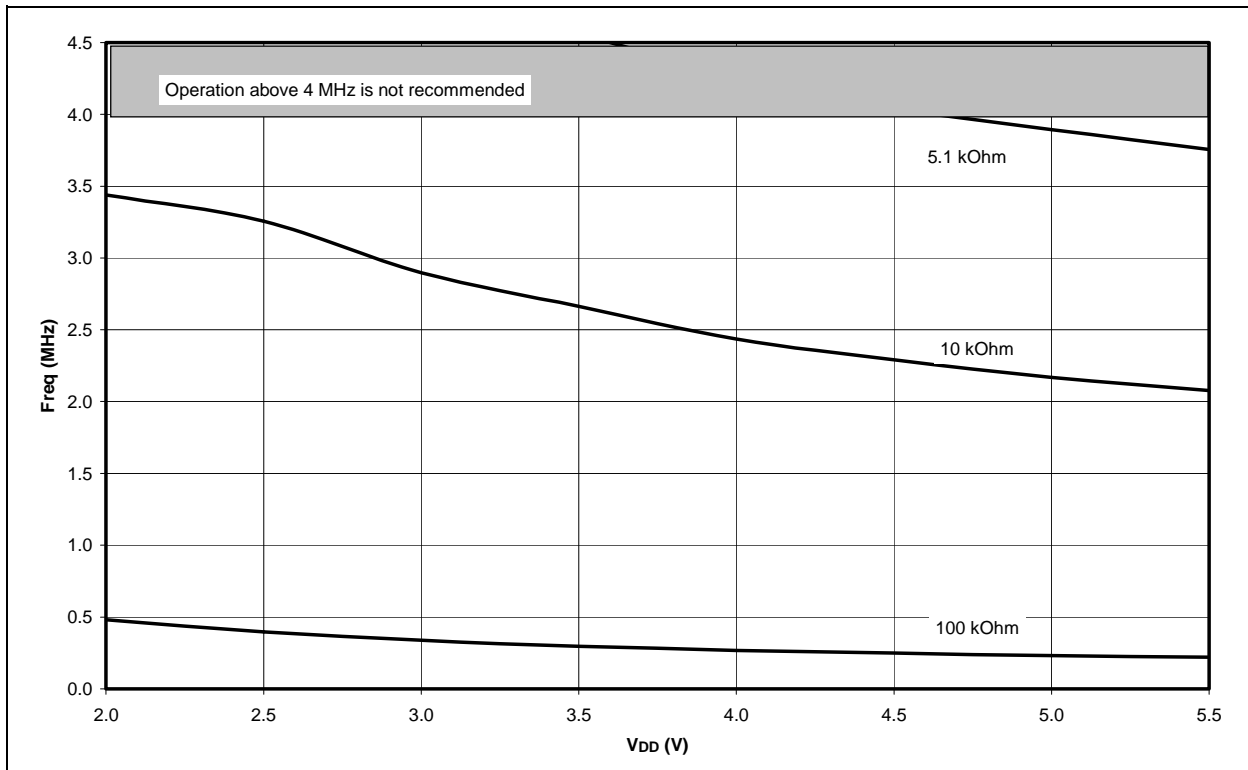


**FIGURE 19-10: I<sub>PD</sub> vs. V<sub>DD</sub>, -40°C TO +125°C (SLEEP MODE, ALL PERIPHERALS DISABLED)**



# PIC16F87/88

**FIGURE 19-11: AVERAGE  $F_{osc}$  vs.  $V_{DD}$  FOR VARIOUS VALUES OF R (RC MODE, C = 20 pF, +25°C)**



**FIGURE 19-12: AVERAGE  $F_{osc}$  vs.  $V_{DD}$  FOR VARIOUS VALUES OF R (RC MODE, C = 100 pF, +25°C)**

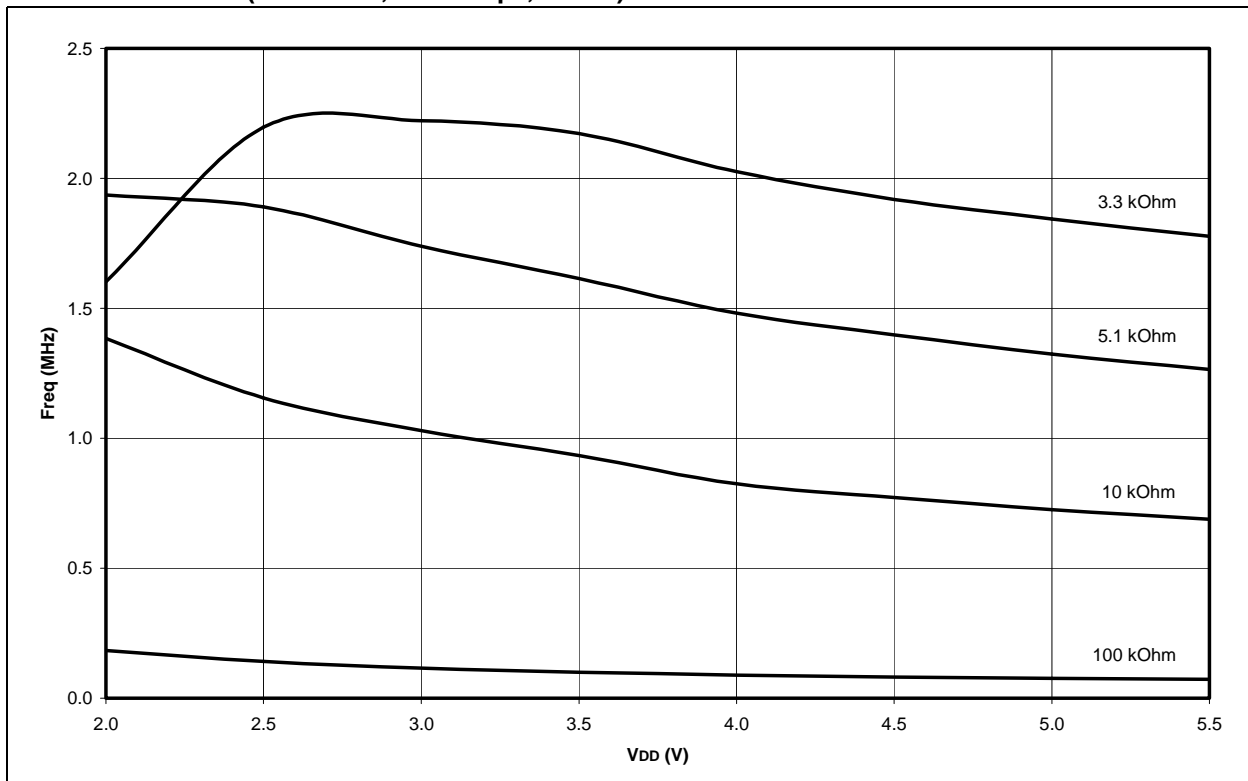




FIGURE 19-13: AVERAGE  $F_{osc}$  vs.  $V_{DD}$  FOR VARIOUS VALUES OF R (RC MODE, C = 300 pF, +25°C)

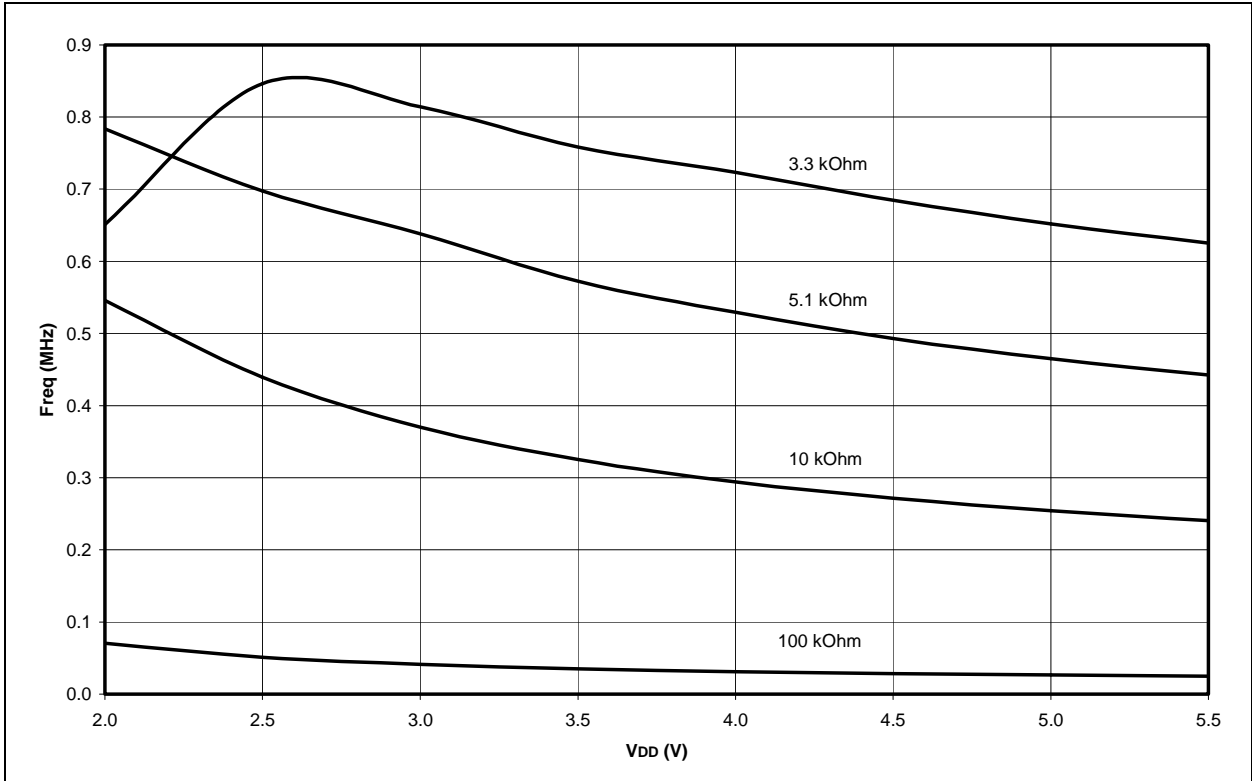
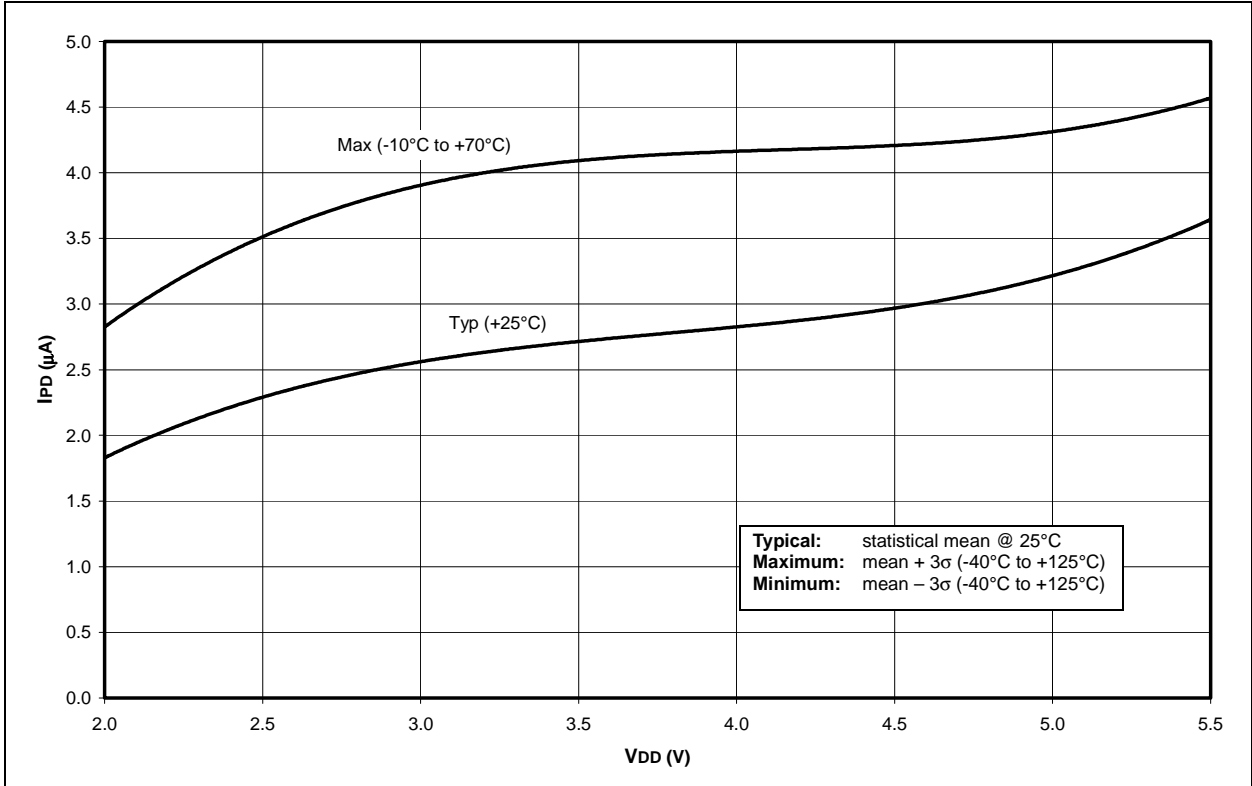
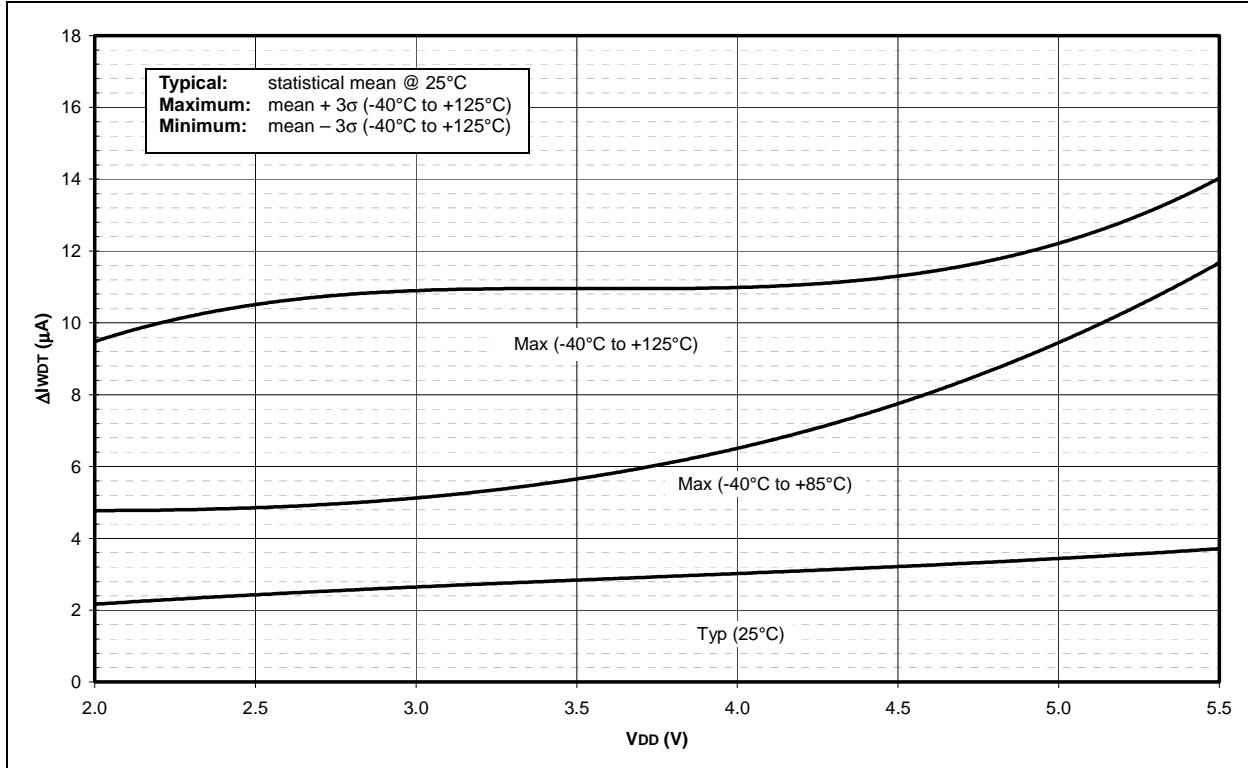


FIGURE 19-14:  $\Delta I_{PD}$  TIMER1 OSCILLATOR, -10°C TO +70°C (SLEEP MODE, TMR1 COUNTER DISABLED)

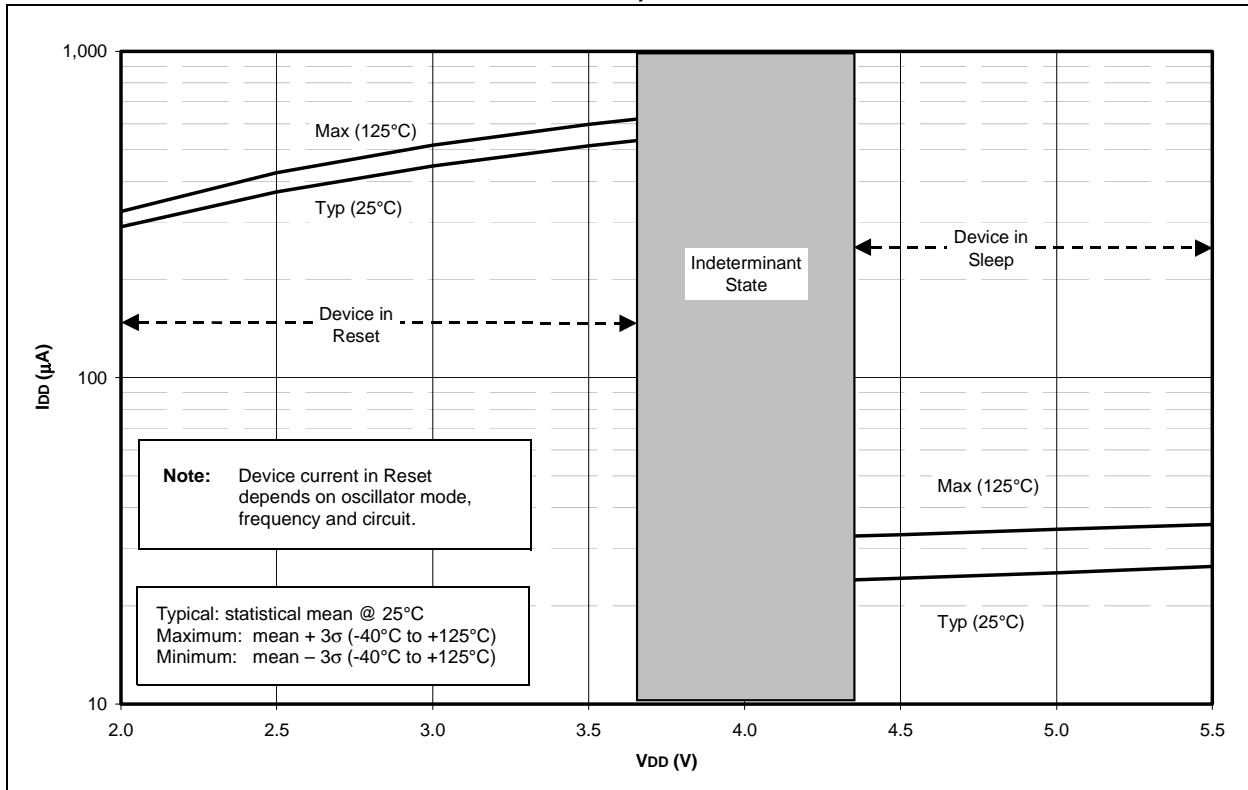


# PIC16F87/88

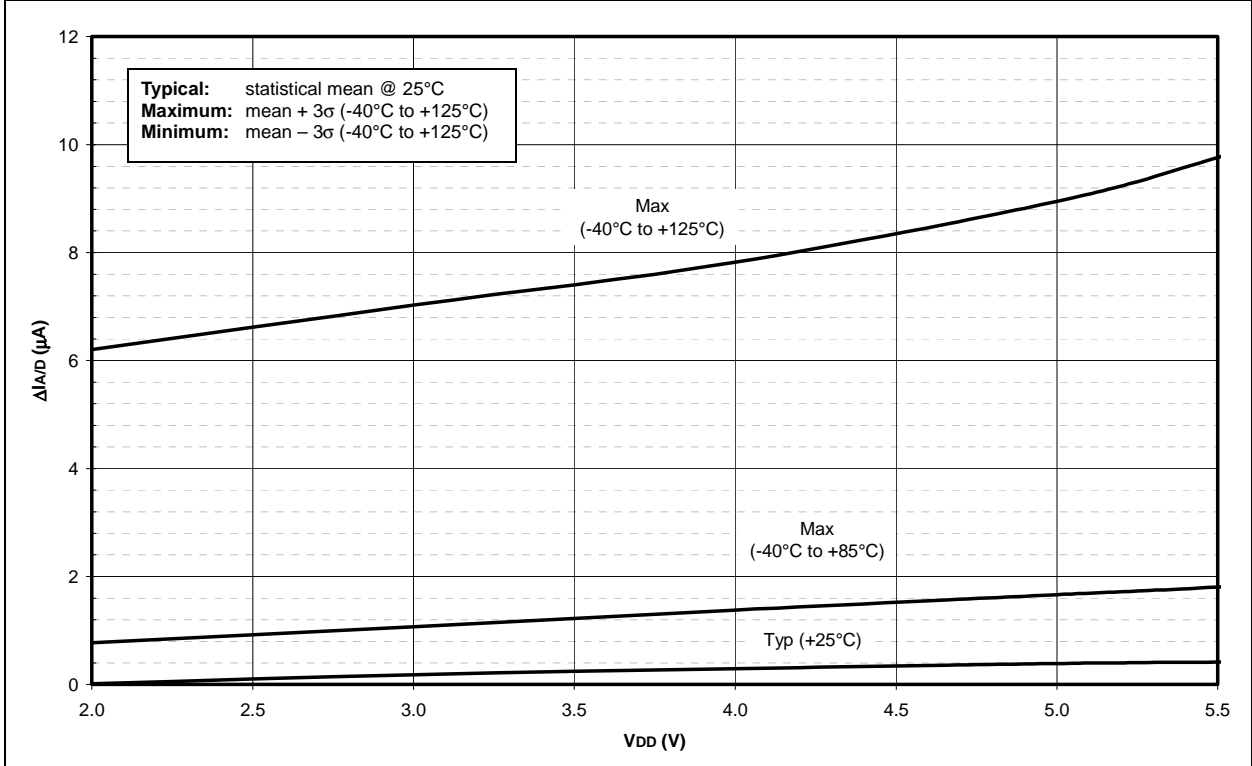
**FIGURE 19-15:  $\Delta I_{PD}$  WDT, -40°C TO +125°C (SLEEP MODE, ALL PERIPHERALS DISABLED)**



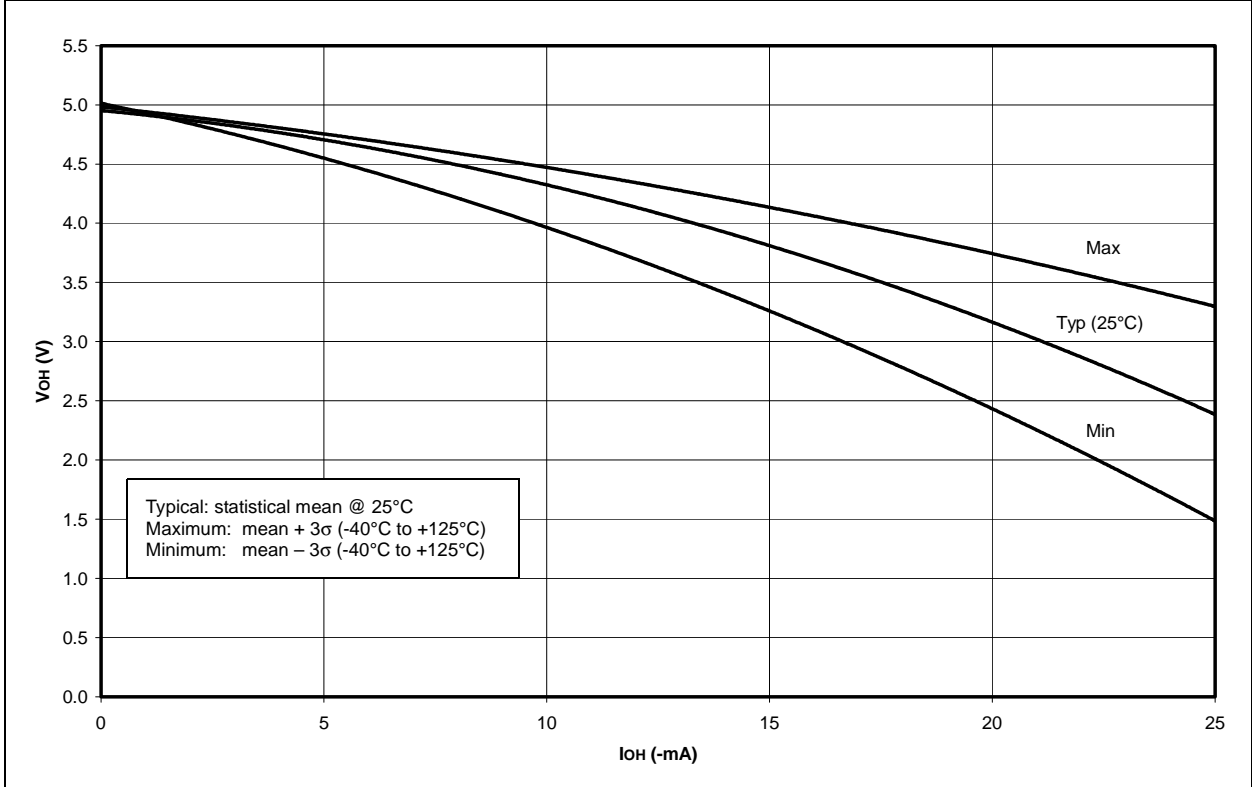
**FIGURE 19-16:  $\Delta I_{PD}$  BOR vs. V<sub>DD</sub>, -40°C TO +125°C (SLEEP MODE, BOR ENABLED AT 2.00V-2.16V)**



**FIGURE 19-17: I<sub>PD</sub> A/D, -40°C TO +125°C, SLEEP MODE, A/D ENABLED (NOT CONVERTING)**

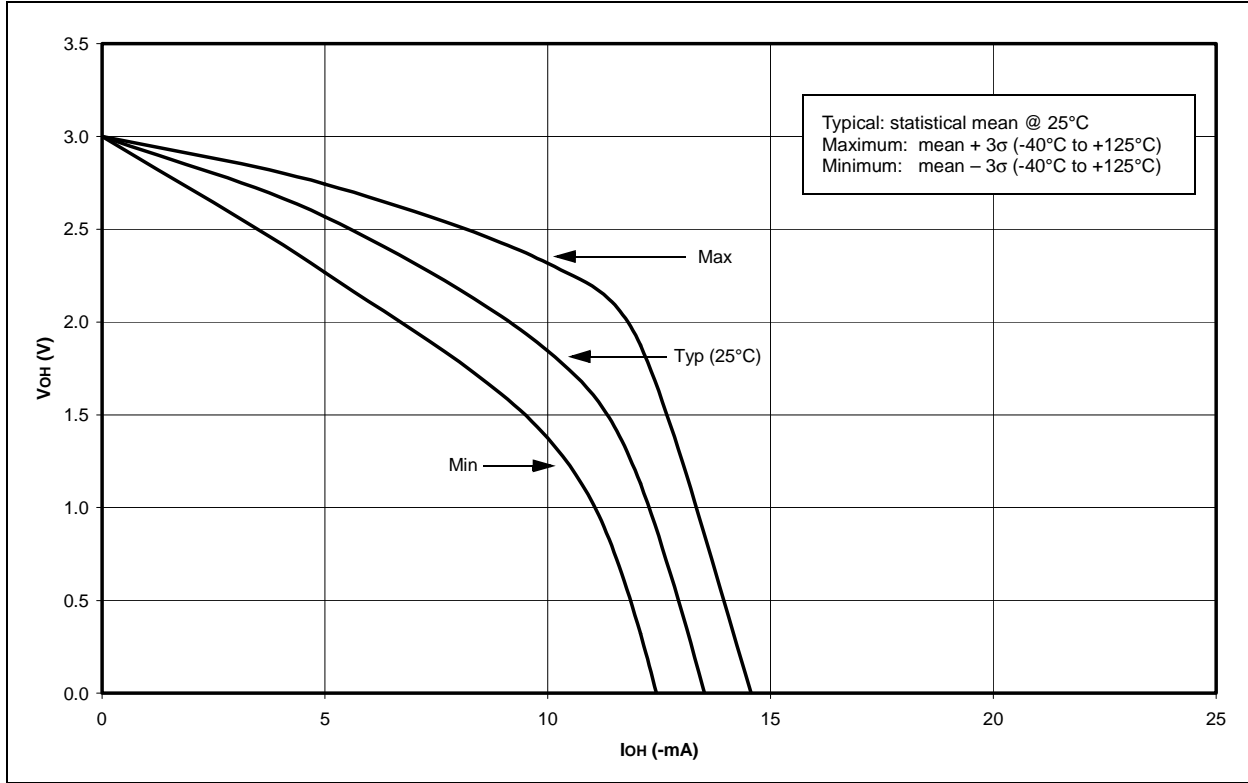


**FIGURE 19-18: TYPICAL, MINIMUM AND MAXIMUM V<sub>OH</sub> vs. I<sub>OH</sub> (V<sub>DD</sub> = 5V, -40°C TO +125°C)**

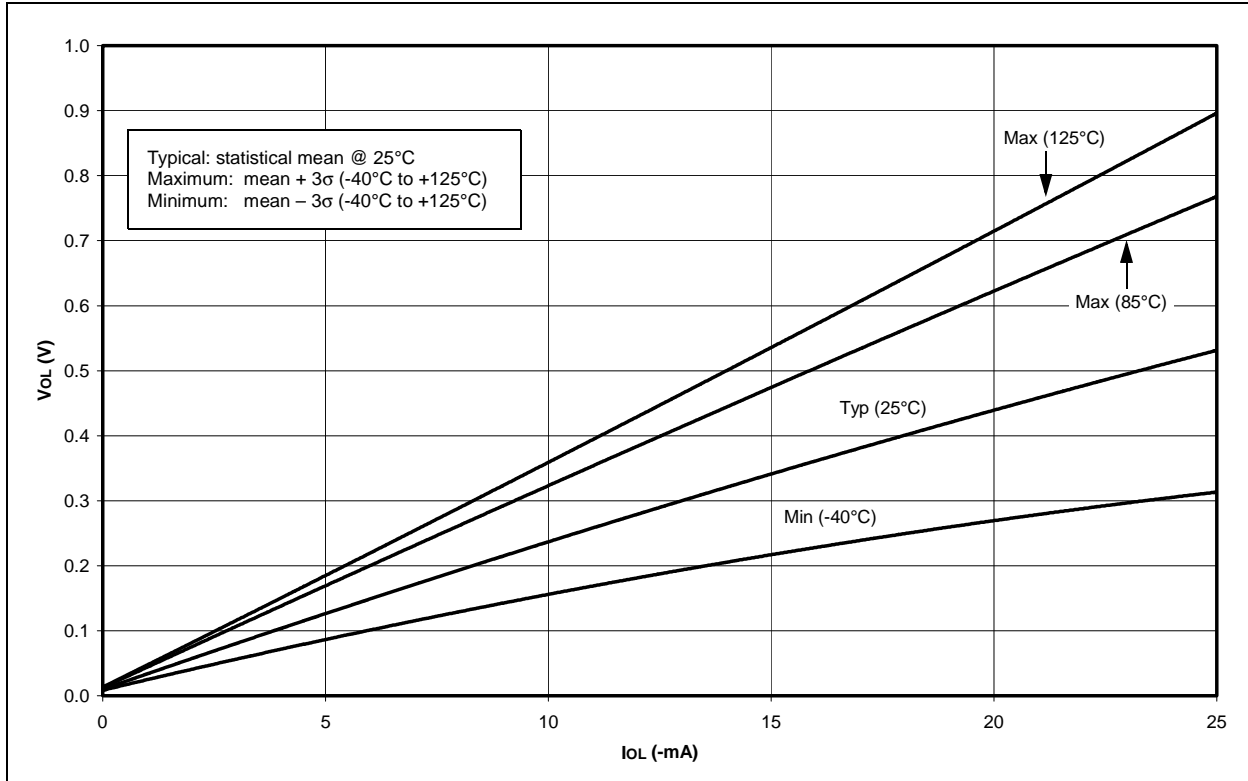


# PIC16F87/88

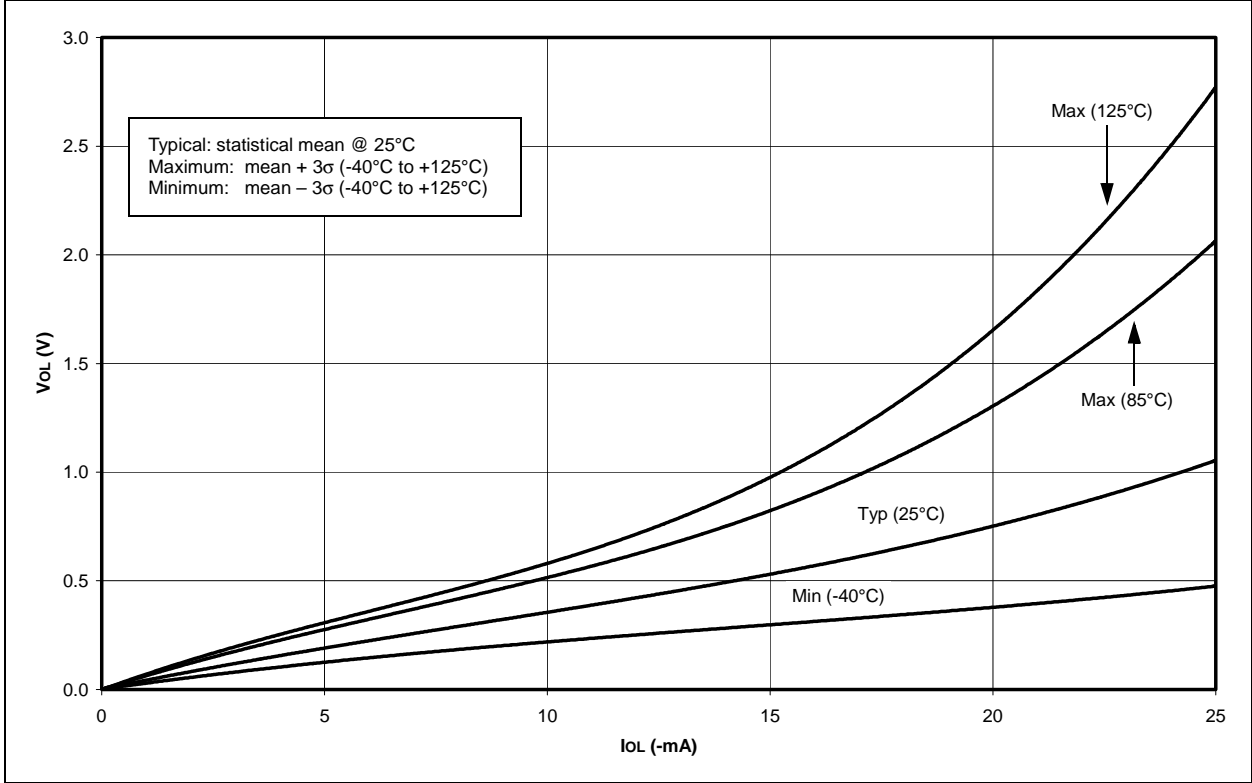
**FIGURE 19-19: TYPICAL, MINIMUM AND MAXIMUM  $V_{OH}$  vs.  $I_{OH}$  ( $V_{DD} = 3V$ ,  $-40^{\circ}C$  TO  $+125^{\circ}C$ )**



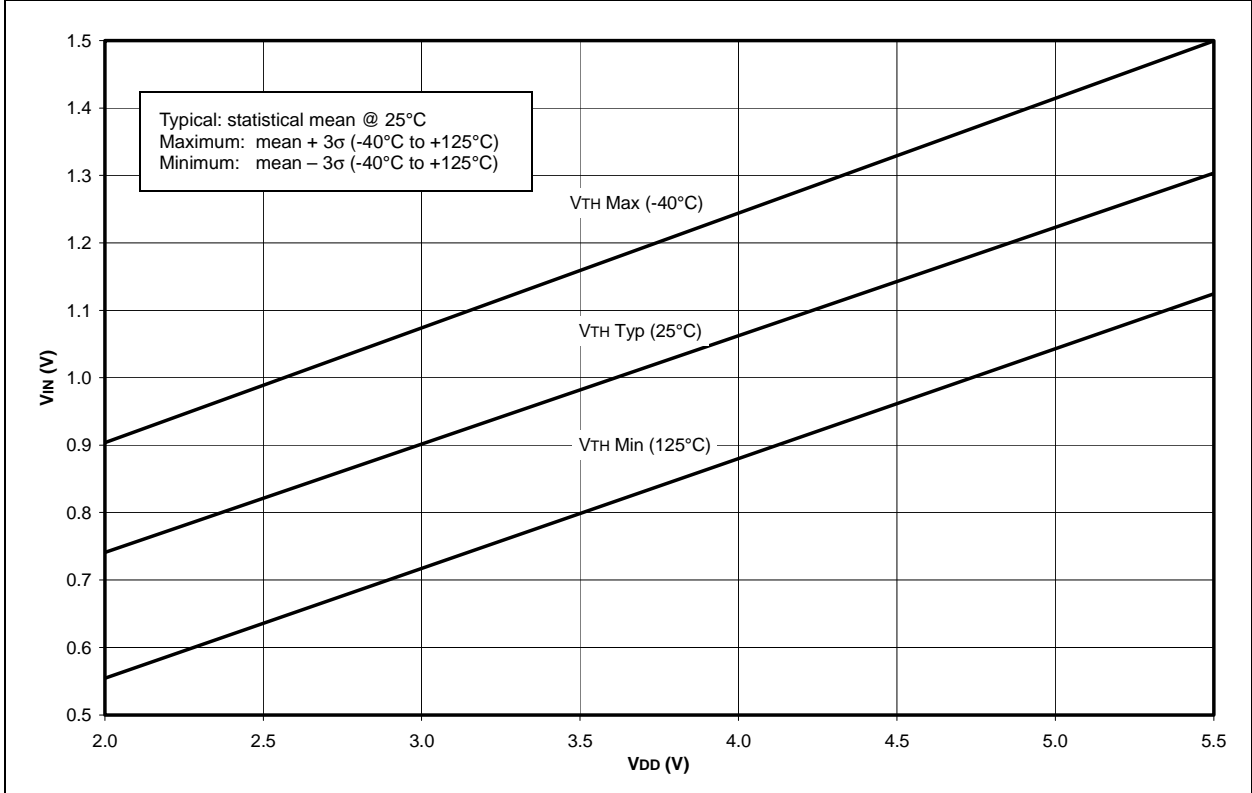
**FIGURE 19-20: TYPICAL, MINIMUM AND MAXIMUM  $V_{OL}$  vs.  $I_{OL}$  ( $V_{DD} = 5V$ ,  $-40^{\circ}C$  TO  $+125^{\circ}C$ )**



**FIGURE 19-21: TYPICAL, MINIMUM AND MAXIMUM  $V_{OL}$  vs.  $I_{OL}$  ( $V_{DD} = 3V$ ,  $-40^{\circ}C$  TO  $+125^{\circ}C$ )**

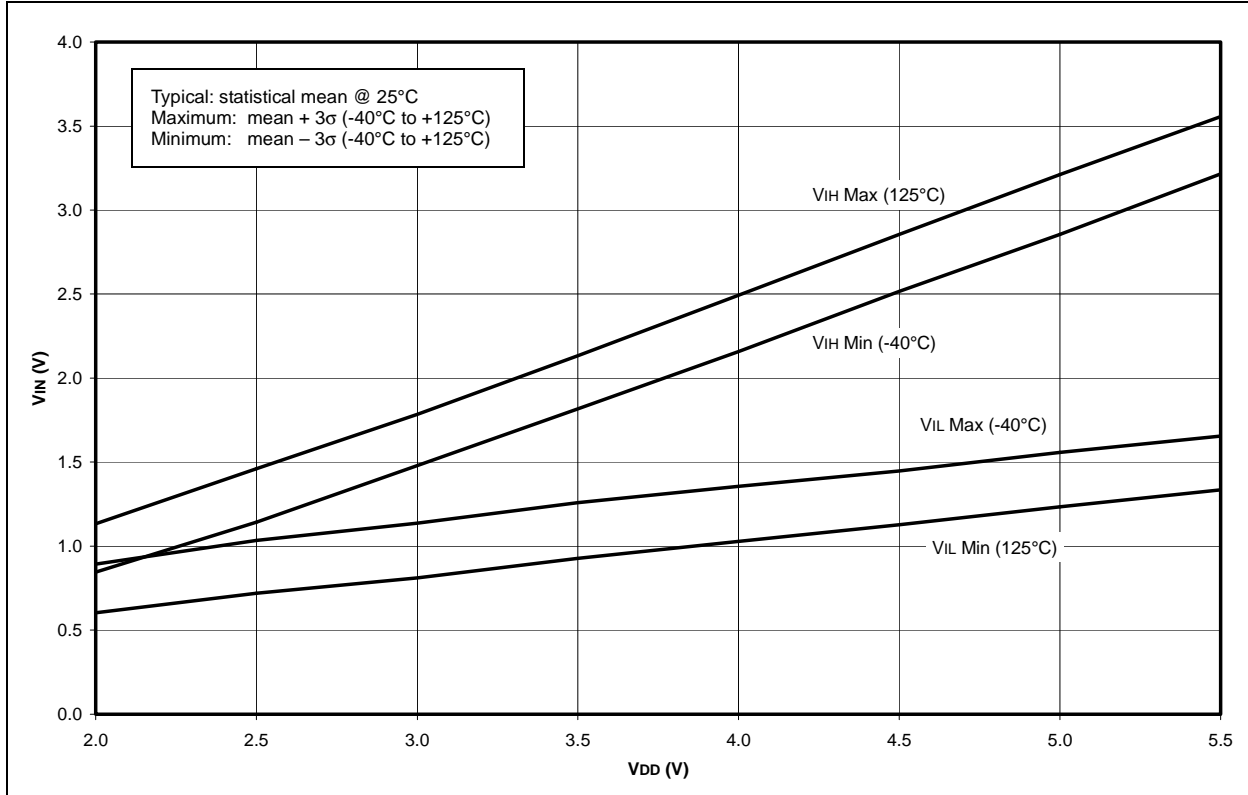


**FIGURE 19-22: MINIMUM AND MAXIMUM  $V_{IN}$  vs.  $V_{DD}$  (TTL INPUT,  $-40^{\circ}C$  TO  $+125^{\circ}C$ )**



# PIC16F87/88

**FIGURE 19-23: MINIMUM AND MAXIMUM  $V_{IN}$  vs.  $V_{DD}$  (ST INPUT,  $-40^{\circ}\text{C}$  TO  $+125^{\circ}\text{C}$ )**



**FIGURE 19-24: MINIMUM AND MAXIMUM  $V_{IN}$  vs.  $V_{DD}$  ( $I^2C^{\text{TM}}$  INPUT,  $-40^{\circ}\text{C}$  TO  $+125^{\circ}\text{C}$ )**

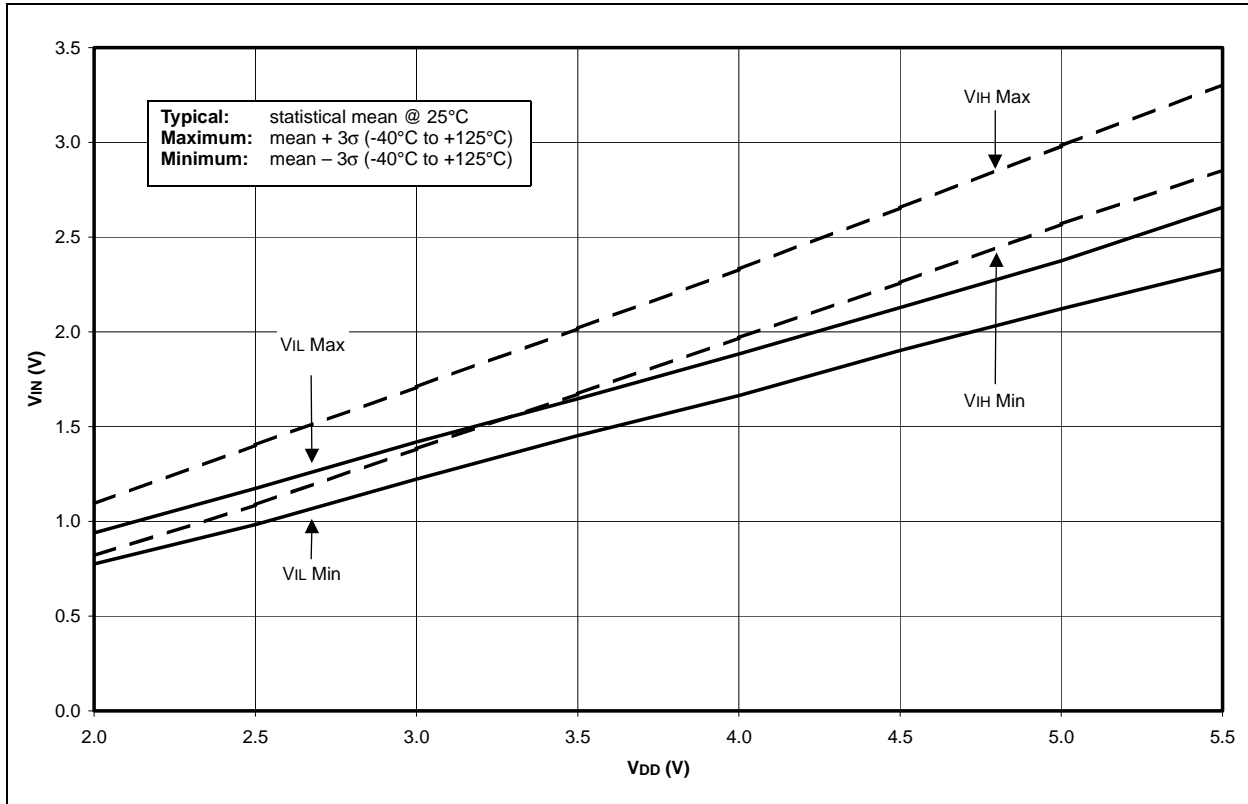


FIGURE 19-25: A/D NONLINEARITY vs. VREFH (VDD = VREFH, -40°C TO +125°C)

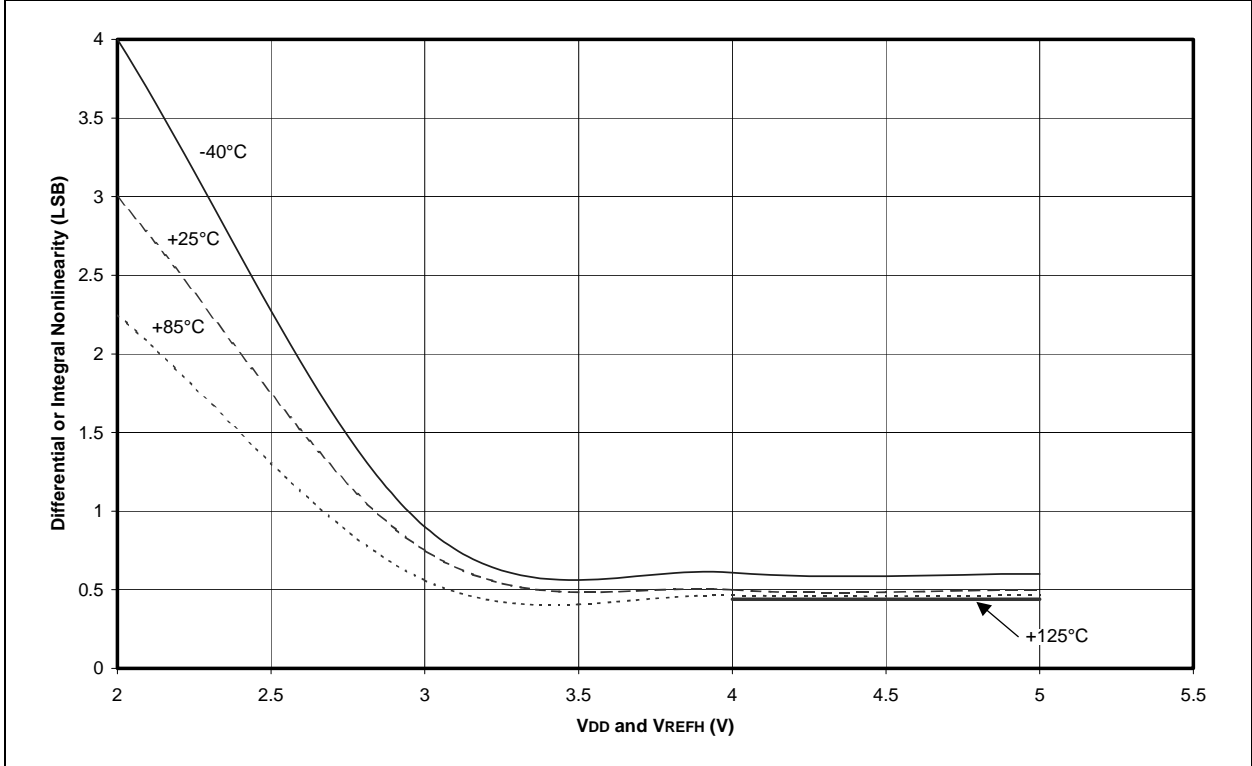
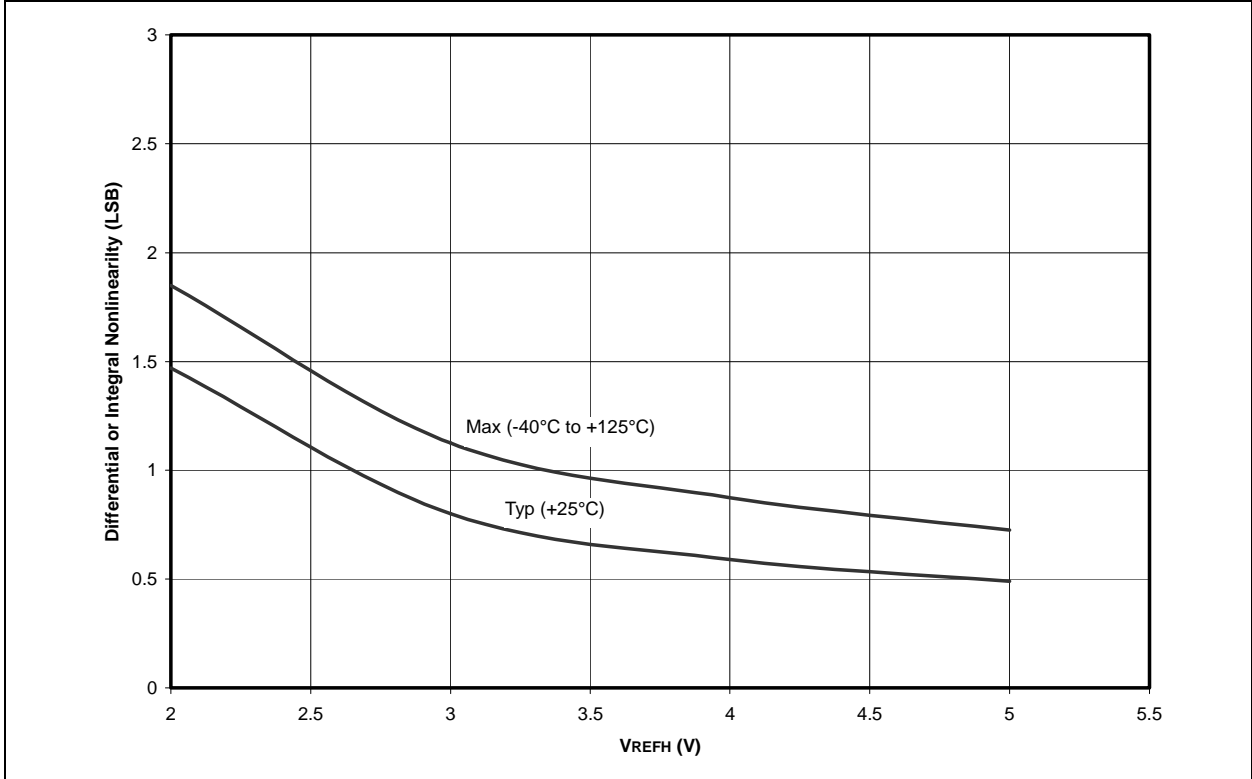


FIGURE 19-26: A/D NONLINEARITY vs. VREFH (VDD = 5V, -40°C TO +125°C)



# PIC16F87/88

---

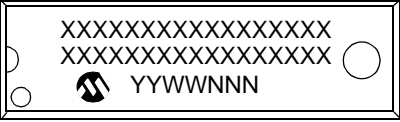
NOTES:



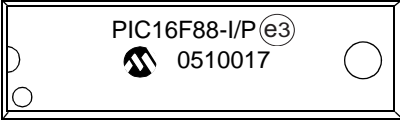
20.0 PACKAGING INFORMATION

20.1 Package Marking Information

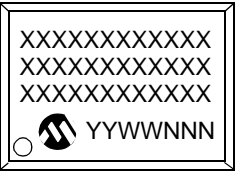
18-Lead PDIP



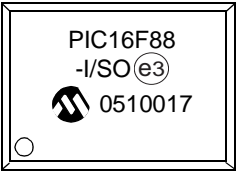
Example



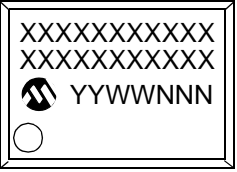
18-Lead SOIC



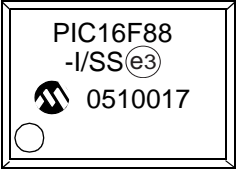
Example



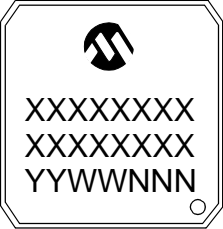
20-Lead SSOP



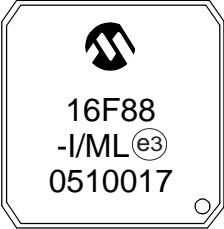
Example



28-Lead QFN



Example

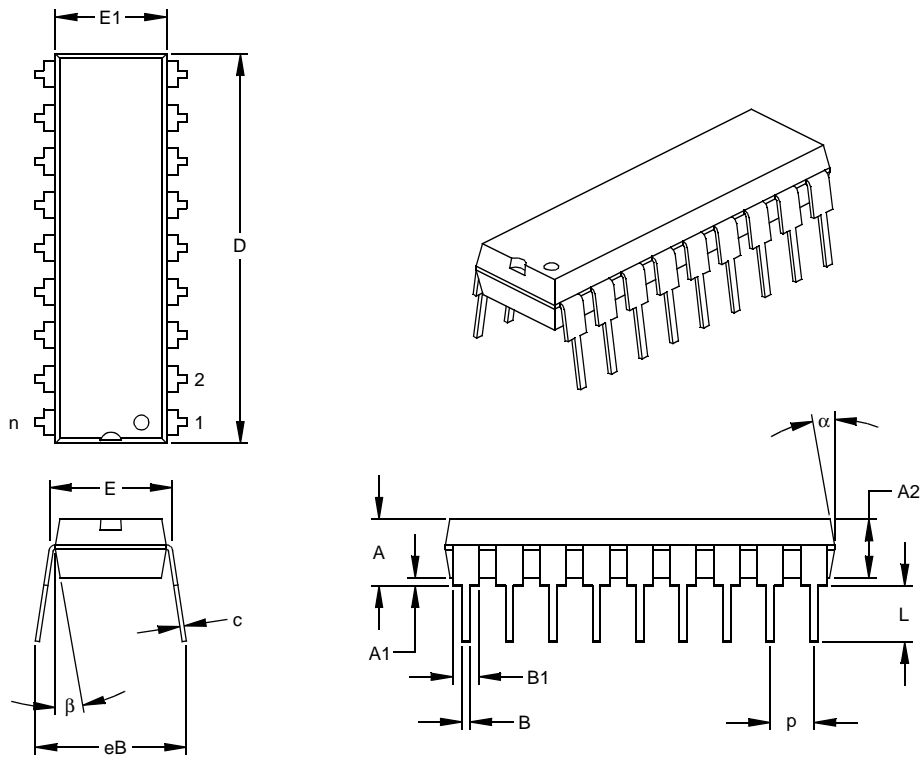


**Legend:** XX...X Customer-specific information  
 Y Year code (last digit of calendar year)  
 YY Year code (last 2 digits of calendar year)  
 WW Week code (week of January 1 is week '01')  
 NNN Alphanumeric traceability code  
 (e3) Pb-free JEDEC designator for Matte Tin (Sn)  
 \* This package is Pb-free. The Pb-free JEDEC designator (e3) can be found on the outer packaging for this package.

**Note:** In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line, thus limiting the number of available characters for customer-specific information.

# PIC16F87/88

## 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

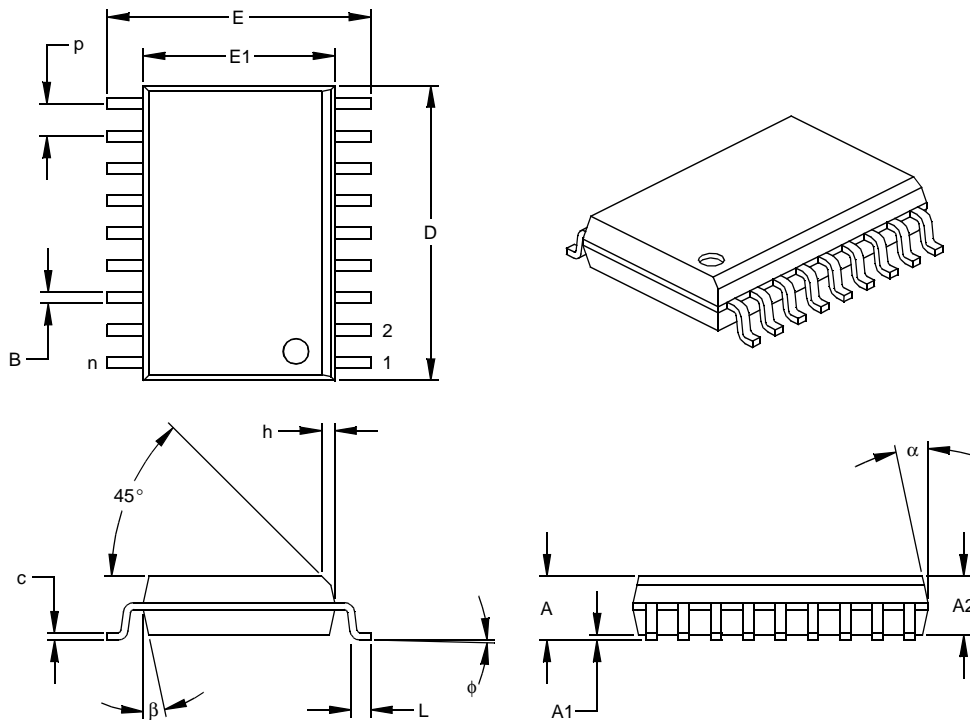
Notes:

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

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



Dimension Limits	Units	INCHES*			MILLIMETERS		
		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:

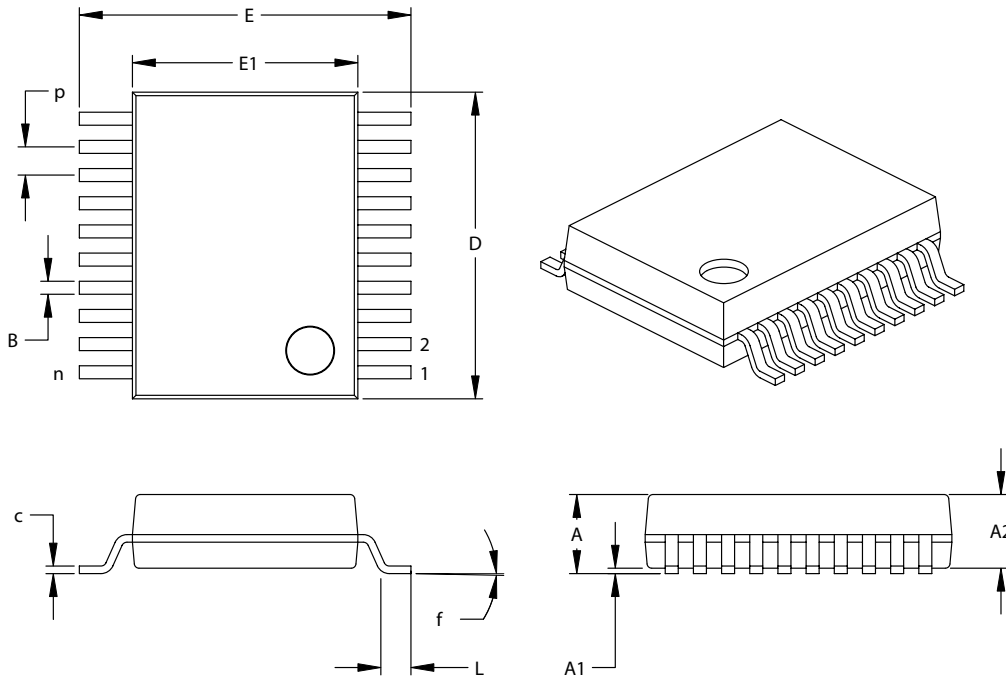
Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" (0.254mm) per side.

JEDEC Equivalent: MS-013

Drawing No. C04-051

# PIC16F87/88

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



Dimension Limits		INCHES			MILLIMETERS*		
		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:

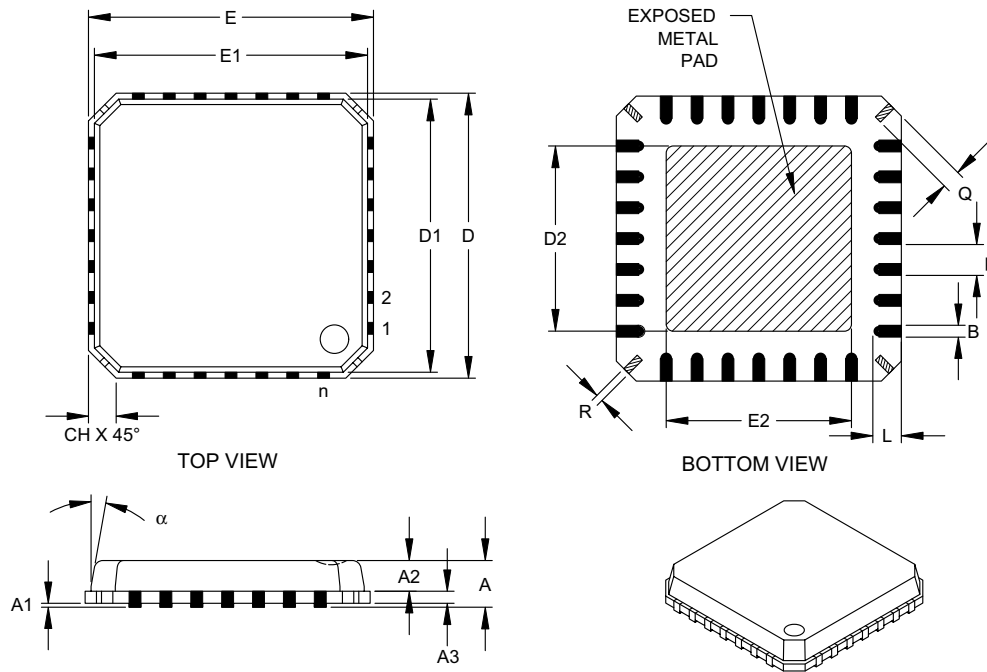
Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" (0.254mm) per side.

JEDEC Equivalent: MO-150

Drawing No. C04-072

Revised 11/03/03

## 28-Lead Plastic Quad Flat No Lead Package (ML) 6x6 mm Body, Punch Singulated (QFN)



Dimension	Units	INCHES			MILLIMETERS*		
		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n	28			28		
Pitch	p	.026 BSC			0.65 BSC		
Overall Height	A		.033	.039		0.85	1.00
Molded Package Thickness	A2		.026	.031		0.65	0.80
Standoff	A1	.000	.0004	.002	0.00	0.01	0.05
Base Thickness	A3		.008 REF		0.20 REF		
Overall Width	E	.236 BSC			6.00 BSC		
Molded Package Width	E1	.226 BSC			5.75 BSC		
Exposed Pad Width	E2	.140	.146	.152	3.55	3.70	3.85
Overall Length	D	.236 BSC			6.00 BSC		
Molded Package Length	D1	.226 BSC			5.75 BSC		
Exposed Pad Length	D2	.140	.146	.152	3.55	3.70	3.85
Lead Width	B	.009	.011	.014	0.23	0.28	0.35
Lead Length	L	.020	.024	.030	0.50	0.60	0.75
Tie Bar Width	R	.005	.007	.010	0.13	0.17	0.23
Tie Bar Length	Q	.012	.016	.026	0.30	0.40	0.65
Chamfer	CH	.009	.017	.024	0.24	0.42	0.60
Mold Draft Angle Top	$\alpha$			12°			12°

\*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-220

Drawing No. C04-114

# PIC16F87/88

---

NOTES:

## APPENDIX A: REVISION HISTORY

### Revision A (November 2003)

Original data sheet for PIC16F87/88 devices.

### Revision B (August 2003)

The specifications in **Section 18.0 “Electrical Characteristics”** have been updated to include the addition of maximum specifications to the DC Characteristics tables, text clarification has been made to **Section 4.6.2 “Clock Switching”** and there have been minor updates to the data sheet text.

### Revision C (January 2005)

This revision includes the DC and AC Characteristics Graphs and Tables. The Electrical Specifications in **Section 18.0 “Electrical Characteristics”** 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 PIC16F87 AND PIC16F88**

Features	PIC16F87	PIC16F88
Analog-to-Digital Converter	N/A	10-bit, 7-channel

# PIC16F87/88

---

NOTES:



## INDEX

### A

#### A/D

Acquisition Requirements .....	117
ADIF Bit .....	116
Analog-to-Digital Converter .....	113
Associated Registers .....	120
Calculating Acquisition Time .....	117
Configuring Analog Port Pins .....	119
Configuring the Interrupt .....	116
Configuring the Module .....	116
Conversion Clock .....	118
Conversions .....	119
Converter Characteristics .....	190
Delays .....	117
Effects of a Reset .....	120
GO/DONE Bit .....	116
Internal Sampling Switch (Rss) Impedance .....	117
Operation During Sleep .....	120
Operation in Power-Managed Modes .....	118
Result Registers .....	119
Source Impedance .....	117
Time Delays .....	117
Using the CCP Trigger .....	120
Absolute Maximum Ratings .....	163
ACK .....	93
ADCON0 Register .....	14, 113
ADCON1 Register .....	15, 113
Addressable Universal Synchronous Asynchronous Receiver Transmitter. See AUSART.	
ADRESH Register .....	14, 113
ADRESH, ADRESL Register Pair .....	116
ADRESL Register .....	15, 113
ANSEL Register .....	15, 52, 58, 113
Application Notes	
AN556 (Implementing a Table Read) .....	25
AN578 (Use of the SSP Module in the I <sup>2</sup> C Multi-Master Environment) .....	87
AN607 (Power-up Trouble Shooting) .....	133
Assembler	
MPASM Assembler .....	157
Asynchronous Reception	
Associated Registers .....	105, 107
Asynchronous Transmission	
Associated Registers .....	103
AUSART .....	97
Address Detect Enable (ADDEN Bit) .....	98
Asynchronous Mode .....	102
Asynchronous Receive (9-bit Mode) .....	106
Asynchronous Receive with Address Detect. See Asynchronous Receive (9-Bit Mode).	
Asynchronous Receiver .....	104
Asynchronous Reception .....	105
Asynchronous Transmitter .....	102
Baud Rate Generator (BRG) .....	99
Baud Rate Formula .....	99
Baud Rates, Asynchronous Mode (BRGH = 0) .....	100
Baud Rates, Asynchronous Mode (BRGH = 1) .....	100
High Baud Rate Select (BRGH Bit) .....	97

INTRC Baud Rates, Asynchronous Mode (BRGH = 0) .....	101
INTRC Baud Rates, Asynchronous Mode (BRGH = 1) .....	101
INTRC Operation .....	99
Low-Power Mode Operation .....	99
Sampling .....	99
Clock Source Select (CSRC Bit) .....	97
Continuous Receive Enable (CREN Bit) .....	98
Framing Error (FERR Bit) .....	98
Mode Select (SYNC Bit) .....	97
Receive Data, 9th bit (RX9D Bit) .....	98
Receive Enable, 9-bit (RX9 Bit) .....	98
Serial Port Enable (SPEN Bit) .....	97, 98
Single Receive Enable (SREN Bit) .....	98
Synchronous Master Mode .....	108
Synchronous Master Reception .....	110
Synchronous Master Transmission .....	108
Synchronous Slave Mode .....	111
Synchronous Slave Reception .....	112
Synchronous Slave Transmit .....	111
Transmit Data, 9th Bit (TX9D) .....	97
Transmit Enable (TXEN Bit) .....	97
Transmit Enable, Nine-bit (TX9 Bit) .....	97
Transmit Shift Register Status (TRMT Bit) .....	97

### B

Baud Rate Generator	
Associated Registers .....	99
BF Bit .....	93
Block Diagrams	
A/D .....	116
Analog Input Model .....	117, 125
AUSART Receive .....	104, 106
AUSART Transmit .....	102
Capture Mode Operation .....	82
Comparator I/O Operating Modes .....	122
Comparator Output .....	124
Comparator Voltage Reference .....	128
Compare Mode Operation .....	83
Fail-Safe Clock Monitor .....	144
In-Circuit Serial Programming	
Connections .....	147
Interrupt Logic .....	139
On-Chip Reset Circuit .....	132
PIC16F87 .....	6
PIC16F88 .....	7
RA0/AN0:RA1/AN1 Pins .....	52
RA2/AN2/CVREF/VREF- Pin .....	53
RA3/AN3/VREF+/C1OUT Pin .....	53
RA4/AN4/T0CKI/C2OUT Pin .....	54
RA5/MCLR/VPP Pin .....	54
RA6/OSC2/CLKO Pin .....	55
RA7/OSC1/CLKI Pin .....	56
RB0/INT/CCP1 Pin .....	59
RB1/SDI/SDA Pin .....	60
RB2/SDO/RX/DT Pin .....	61
RB3/PGM/CCP1 Pin .....	62
RB4/SCK/SCL Pin .....	63
RB5/SS/TX/CK Pin .....	64

# PIC16F87/88

RB6/AN5/PGC/T1OSO/T1CKI Pin .....	65	Initializing PORTA .....	51
RB7/AN6/PGD/T1OSI Pin .....	66	Reading a 16-Bit Free Running Timer .....	74
Simplified PWM .....	84	Reading Data EEPROM .....	29
SSP in I <sup>2</sup> C Mode .....	92	Reading Flash Program Memory .....	30
SSP in SPI Mode .....	90	Saving STATUS, W and PCLATH Registers in RAM .....	140
System Clock .....	41	Writing a 16-Bit Free Running Timer .....	74
Timer0/WDT Prescaler .....	67	Writing to Data EEPROM .....	29
Timer1 .....	73	Writing to Flash Program Memory .....	33
Timer2 .....	79	Code Protection .....	129, 147
Watchdog Timer (WDT) .....	141	Comparator Module .....	121
BOR. See Brown-out Reset.		Analog Input Connection Considerations .....	125
BRGH Bit .....	99	Associated Registers .....	126
Brown-out Reset (BOR) .....	129, 132, 133, 135	Configuration .....	122
BOR Status ( $\overline{\text{BOR}}$ Bit) .....	24	Effects of a Reset .....	125
<b>C</b>		External Reference Signal .....	123
C Compilers		Internal Reference Signal .....	123
MPLAB C17 .....	158	Interrupts .....	124
MPLAB C18 .....	158	Operation .....	123
MPLAB C30 .....	158	Operation During Sleep .....	125
Capture/Compare/PWM (CCP) .....	81	Outputs .....	123
Capture Mode .....	82	Reference .....	123
CCP Pin Configuration .....	82	Response Time .....	123
Software Interrupt .....	82	Comparator Specifications .....	177
Timer1 Mode Selection .....	82	Comparator Voltage Reference .....	127
Capture, Compare and Timer1		Associated Registers .....	128
Associated Registers .....	83	Computed GOTO .....	25
CCP Prescaler .....	82	Configuration Bits .....	129
CCP Timer Resources .....	81	Crystal and Ceramic Resonators .....	35
CCP1IF .....	82	Customer Change Notification Service .....	223
CCPR1 .....	82	Customer Notification Service .....	223
CCPR1H:CCPR1L .....	82	Customer Support .....	223
Compare Mode .....	83	CVRCON Register .....	15
CCP Pin Configuration .....	83	<b>D</b>	
Software Interrupt Mode .....	83	Data EEPROM Memory .....	27
Special Event Trigger .....	83	Associated Registers .....	34
Special Event Trigger Output of CCP1 .....	83	EEADR Register .....	27
Timer1 Mode Selection .....	83	EEADRH Register .....	27
PWM and Timer2 Associated Registers .....	85	EECON1 Register .....	27
PWM Mode .....	84	EECON2 Register .....	27
Example Frequencies/Resolutions .....	85	EEDATA Register .....	27
Operation Setup .....	85	EEDATH Register .....	27
Operation Setup .....	85	Operation During Code-Protect .....	34
CCP1CON Register .....	14	Protection Against Spurious Writes .....	34
CCP1M0 Bit .....	81	Reading .....	29
CCP1M1 Bit .....	81	Write Complete Flag (EEIF Bit) .....	27
CCP1M2 Bit .....	81	Writing .....	29
CCP1M3 Bit .....	81	Data Memory	
CCP1X Bit .....	81	Special Function Registers .....	14
CCP1Y Bit .....	81	DC and AC Characteristics	
CCPR1H Register .....	14, 81	Graphs and Tables .....	193
CCPR1L Register .....	14, 81	DC Characteristics	
Clock Sources .....	39	Internal RC Accuracy .....	174
Selection Using OSCCON Register .....	39	PIC16F87/88, PIC16LF87/88 .....	175
Clock Switching .....	39	Power-Down and Supply Current .....	166
Transition and the Watchdog Timer .....	40	Supply Voltage .....	165
Transition Sequence .....	41	Demonstration Boards	
CMCON Register .....	15	PICDEM 1 .....	160
Code Examples		PICDEM 17 .....	161
Call of a Subroutine in Page 1 from Page 0 .....	25	PICDEM 18R .....	161
Changing Between Capture Prescalers .....	82	PICDEM 2 Plus .....	160
Changing Prescaler Assignment from		PICDEM 3 .....	160
WDT to Timer0 .....	69	PICDEM 4 .....	160
Erasing a Flash Program Memory Row .....	31	PICDEM LIN .....	161
Implementing a Real-Time Clock Using		PICDEM USB .....	161
a Timer1 Interrupt Service .....	77	PICDEM.net Internet/Ethernet .....	160
Indirect Addressing .....	26		

Development Support .....	157	Instruction Set .....	149
Device Differences .....	213	ADDLW .....	151
Device Overview .....	5	ADDWF .....	151
Direct Addressing .....	26	ANDLW .....	151
<b>E</b>		ANDWF .....	151
EEADR Register .....	16, 27	BCF .....	151
EEADRH Register .....	16, 27	BSF .....	151
EECON1 Register .....	16, 27	BTFSC .....	152
EECON2 Register .....	16, 27	BTFSS .....	152
EEDATA Register .....	16, 27	CALL .....	152
EEDATH Register .....	16, 27	CLRF .....	152
Electrical Characteristics .....	163	CLRW .....	152
Errata .....	4	CLRWDT .....	152
Evaluation and Programming Tools .....	161	COMF .....	153
Exiting Sleep with an Interrupt .....	50	DECF .....	153
External Clock Input .....	36	DECFSZ .....	153
External Clock Input (RA4/T0CKI). See Timer0.		Descriptions .....	151
External Interrupt Input (RB0/INT). See Interrupt Sources.		General Format .....	149
<b>F</b>		GOTO .....	153
Fail-Safe Clock Monitor .....	129, 144	INCF .....	153
Flash Program Memory .....	27	INCFSZ .....	153
Associated Registers .....	34	IORLW .....	154
EEADR Register .....	27	IORWF .....	154
EEADRH Register .....	27	MOVF .....	154
EECON1 Register .....	27	MOVLW .....	154
EECON2 Register .....	27	MOVWF .....	154
EEDATA Register .....	27	NOP .....	154
EEDATH Register .....	27	Read-Modify-Write Operations .....	149
Erasing .....	30	RETFIE .....	155
Reading .....	30	RETLW .....	155
Writing .....	32	RETURN .....	155
FSR Register .....	14, 15, 26	RLF .....	155
<b>G</b>		RRF .....	155
General Purpose Register File .....	12	SLEEP .....	155
<b>I</b>		SUBLW .....	156
I/O Ports .....	51	SUBWF .....	156
PORTA .....	51	Summary Table .....	150
PORTB .....	57	SWAPF .....	156
TRISB Register .....	57	XORLW .....	156
I <sup>2</sup> C		XORWF .....	156
Addressing .....	93	INT Interrupt (RB0/INT). See Interrupt Sources.	
Associated Registers .....	95	INTCON Register	
Master Mode .....	95	GIE Bit .....	19
Mode .....	92	INT0IE Bit .....	19
Mode Selection .....	92	INT0IF Bit .....	19
Multi-Master Mode .....	95	PEIE Bit .....	19
Reception .....	93	RBIE Bit .....	19
SCL and SDA Pins .....	93	RBIF Bit .....	19
Slave Mode .....	93	TMR0IE Bit .....	19
Transmission .....	93	Internal Oscillator Block .....	37
ID Locations .....	129, 147	INTRC Modes .....	38
In-Circuit Debugger .....	147	Internet Address .....	223
In-Circuit Serial Programming .....	129	Interrupt Sources .....	129, 138
In-Circuit Serial Programming (ICSP) .....	147	AUSART Receive/Transmit Complete .....	97
INDF Register .....	14, 15, 26	RB0/INT Pin, External .....	140
Indirect Addressing .....	26	TMR0 Overflow .....	140
		Interrupts	
		RB7:RB4 Port Change .....	57
		Interrupts, Context Saving During .....	140

# PIC16F87/88

Interrupts, Enable Bits	
A/D Converter Interrupt Enable (ADIE Bit) .....	20
AUSART Receive Interrupt Enable (RCIE Bit) .....	20
AUSART Transmit Interrupt Enable (TXIE Bit) .....	20
CCP1 Interrupt Enable (CCP1IE Bit) .....	20
Comparator Interrupt Enable (CMIE Bit) .....	22
EEPROM Write Operation Interrupt	
Enable (EEIE Bit) .....	22
Global Interrupt Enable (GIE Bit) .....	19, 138
Interrupt-on-Change (RB7:RB4)	
Enable (RBIE Bit) .....	140
Oscillator Fail Interrupt Enable (OSFIE Bit) .....	22
Peripheral Interrupt Enable (PEIE Bit) .....	19
Port Change Interrupt Enable (RBIE Bit) .....	19
RB0/INT Enable (INT0IE Bit) .....	19
Synchronous Serial Port (SSP)	
Interrupt Enable (SSPIE Bit) .....	20
TMR0 Overflow Enable (TMR0IE Bit) .....	19
TMR1 Overflow Interrupt Enable	
(TMR1IE Bit) .....	20
TMR2 to PR2 Match Interrupt Enable	
(TMR2IE Bit) .....	20
Interrupts, Flag Bits	
A/D Converter Interrupt Flag (ADIF Bit) .....	21
AUSART Receive Interrupt Flag (RCIF Bit) .....	21
AUSART Transmit Interrupt Flag (TXIF Bit) .....	21
CCP1 Interrupt Flag (CCP1IF Bit) .....	21
Comparator Interrupt Flag (CMIF Bit) .....	23
EEPROM Write Operation Interrupt	
Flag (EEIF Bit) .....	23
Interrupt-on-Change (RB7:RB4)	
Flag (RBIF Bit) .....	19, 140
Oscillator Fail Interrupt Flag (OSFIF Bit) .....	23
RB0/INT Flag (INT0IF Bit) .....	19
Synchronous Serial Port (SSP) Interrupt	
Flag (SSPIF Bit) .....	21
TMR0 Overflow Flag (TMR0IF Bit) .....	140
TMR1 Overflow Interrupt Flag (TMR1IF Bit) .....	21
TMR2 to PR2 Interrupt Flag (TMR2IF Bit) .....	21
INTRC Modes	
Adjustment .....	38
<b>L</b>	
Loading of PC .....	25
Low-Voltage ICSP Programming .....	148
<b>M</b>	
Master Clear ( $\overline{\text{MCLR}}$ )	
MCLR Reset, Normal Operation .....	132, 135
MCLR Reset, Sleep .....	132, 135
Operation and ESD Protection .....	133
Memory Organization .....	11
Data Memory .....	11
Program Memory .....	11
Microchip Internet Web Site .....	223
MPLAB ASM30 Assembler, Linker, Librarian .....	158
MPLAB ICD 2 In-Circuit Debugger .....	159
MPLAB ICE 2000 High-Performance	
Universal In-Circuit Emulator .....	159
MPLAB ICE 4000 High-Performance	
Universal In-Circuit Emulator .....	159
MPLAB Integrated Development	
Environment Software .....	157
MPLAB PM3 Device Programmer .....	159
MPLINK Object Linker/MPLIB Object Librarian .....	158
<b>O</b>	
Opcode Field Descriptions .....	149
OPTION_REG Register	
INTEDG Bit .....	18
PS2:PS0 Bits .....	18
PSA Bit .....	18
RBPU Bit .....	18
T0CS Bit .....	18
T0SE Bit .....	18
OSCCON Register .....	15
Oscillator Configuration .....	35
ECIO .....	35
EXTRC .....	134
HS .....	35, 134
INTIO1 .....	35
INTIO2 .....	35
INTRC .....	134
LP .....	35, 134
RC .....	35, 37
RCIO .....	35
XT .....	35, 134
Oscillator Control Register	
Modifying IRCF Bits .....	41
Oscillator Delay upon Power-up, Wake-up	
and Clock Switching .....	42
Oscillator Start-up Timer (OST) .....	129, 133
Oscillator Switching .....	39
OSCTUNE Register .....	15
<b>P</b>	
Packaging Information .....	207
Marking .....	207
Paging, Program Memory .....	25
PCL Register .....	14, 15, 25
PCLATH Register .....	14, 15, 25
PCON Register .....	15, 134
BOR Bit .....	24
POR Bit .....	24
PICkit 1 Flash Starter Kit .....	161
PICSTART Plus Development Programmer .....	160
PIE1 Register .....	15
ADIE Bit .....	20
CCP1IE Bit .....	20
RCIE Bit .....	20
SSPIE Bit .....	20
TMR1IE Bit .....	20
TMR2IE Bit .....	20
TXIE Bit .....	20
PIE2 Register .....	15
CMIE Bit .....	22
EEIE Bit .....	22
OSFIE Bit .....	22
Pinout Descriptions	
PIC16F87/88 .....	8
PIR1 Register .....	14
ADIF Bit .....	21
CCP1IF Bit .....	21
RCIF Bit .....	21
SSPIF Bit .....	21
TMR1IF Bit .....	21
TMR2IF Bit .....	21
TXIF Bit .....	21

PIR2 Register .....	14	RB3/PGM/CCP1 Pin .....	9
CMIF Bit .....	23	RB4/SCK/SCL Pin .....	9
EEIF Bit .....	23	RB5/SS/TX/CK Pin .....	9
OSFIF Bit .....	23	RB6/AN5/PGC/T1OSO/T1CKI Pin .....	9
POP .....	25	RB7/AN6/PGD/T1OSI Pin .....	9
POR. See Power-on Reset.		RBIF Bit .....	57
PORTA .....	8	RCIO Oscillator .....	37
Associated Register Summary .....	52	RCREG Register .....	14
PORTA Register .....	14	RCSTA Register .....	14
PORTB .....	9	ADDEN Bit .....	98
Associated Register Summary .....	58	CREN Bit .....	98
PORTB Register .....	14, 16	FERR Bit .....	98
Pull-up Enable (RBPU Bit) .....	18	RX9 Bit .....	98
RB0/INT Edge Select (INTEDG Bit) .....	18	RX9D Bit .....	98
RB0/INT Pin, External .....	140	SPEN Bit .....	97, 98
RB2/SDO/RX/DT Pin .....	98, 99	SREN Bit .....	98
RB5/SS/TX/CK Pin .....	98	Reader Response .....	224
RB7:RB4 Interrupt-on-Change .....	140	Receive Overflow Indicator Bit, SSPOV .....	89
RB7:RB4 Interrupt-on-Change		Register File Map	
Enable (RBIE Bit) .....	140	PIC16F87 .....	12
RB7:RB4 Interrupt-on-Change		PIC16F88 .....	13
Flag (RBIF Bit) .....	19, 140	Registers	
TRISB Register .....	97	ADCON0 (A/D Control 0) .....	114
Postscaler, WDT		ADCON1 (A/D Control 1) .....	115
Assignment (PSA Bit) .....	18	ANSEL (Analog Select) .....	113
Rate Select (PS2:PS0 Bits) .....	18	CCP1CON (Capture/Compare/PWM	
Power-Down Mode. See Sleep.		Control 1) .....	81
Power-Managed Modes .....	43	CMCON (Comparator Control) .....	121
RC_RUN .....	43	CONFIG1 (Configuration Word 1) .....	130
SEC_RUN .....	44	CONFIG2 (Configuration Word 2) .....	131
SEC_RUN/RC_RUN to Primary Clock Source .....	45	CVRCON (Comparator Voltage	
Power-on Reset (POR) .....	129, 132, 133, 135	Reference Control) .....	127
POR Status (POR Bit) .....	24	EECON1 (Data EEPROM Access Control 1) .....	28
Power Control (PCON) Register .....	134	FSR .....	26
Power-Down (PD Bit) .....	132	Initialization Conditions (table) .....	135–136
Time-out (TO Bit) .....	17, 132	INTCON (Interrupt Control) .....	19
Power-up Timer (PWRT) .....	129, 133	OPTION_REG (Option Control) .....	18, 68
PR2 Register .....	15, 79	OSCCON (Oscillator Control) .....	40
Prescaler, Timer0		OSTUNE (Oscillator Tuning) .....	38
Assignment (PSA Bit) .....	18	PCON (Power Control) .....	24
Rate Select (PS2:PS0 Bits) .....	18	PIE1 (Peripheral Interrupt Enable 1) .....	20
PRO MATE II Universal Device Programmer .....	159	PIE2 (Peripheral Interrupt Enable 2) .....	22
Program Counter		PIR1 (Peripheral Interrupt Request (Flag) 1) .....	21
Reset Conditions .....	135	PIR2 (Peripheral Interrupt Request (Flag) 2) .....	23
Program Memory		RCSTA (Receive Status and Control) .....	98
Interrupt Vector .....	11	Special Function, Summary .....	14
Map and Stack		SSPCON (Synchronous Serial Port Control) .....	89
PIC16F87/88 .....	11	SSPSTAT (Synchronous Serial Port Status) .....	88
Paging .....	25	STATUS (Arithmetic Status) .....	17
Reset Vector .....	11	T1CON (Timer1 Control) .....	72
Program Verification .....	147	T2CON (Timer2 Control) .....	80
PUSH .....	25	TXSTA (Transmit Status and Control) .....	97
<b>R</b>		WDTCON (Watchdog Timer Control) .....	142
R/W Bit .....	93	Reset .....	129, 132
RA0/AN0 Pin .....	8	Brown-out Reset (BOR). See Brown-out Reset (BOR).	
RA1/AN1 Pin .....	8	MCLR Reset. See MCLR.	
RA2/AN2/CVREF/VREF- Pin .....	8	Power-on Reset (POR). See Power-on Reset (POR).	
RA3/AN3/VREF+/C1OUT Pin .....	8	Reset Conditions for All Registers .....	135
RA4/AN4/T0CKI/C2OUT Pin .....	8	Reset Conditions for PCON Register .....	135
RA5/MCLR/VPP Pin .....	8	Reset Conditions for Program Counter .....	135
RA6/OSC2/CLKO Pin .....	8	Reset Conditions for STATUS Register .....	135
RA7/OSC1/CLKI Pin .....	8	WDT Reset. See Watchdog Timer (WDT).	
RB0/INT/CCP1 Pin .....	9	Revision History .....	213
RB1/SDI/SDA Pin .....	9	RP0 Bit .....	11
RB2/SDO/RX/DT Pin .....	9	RP1 Bit .....	11

# PIC16F87/88

## S

SCI. See AUSART.	
SCL .....	93
Serial Communication Interface. See AUSART.	
Slave Mode	
SCL .....	93
SDA .....	93
Sleep .....	129, 132, 145
Software Simulator (MPLAB SIM) .....	158
Software Simulator (MPLAB SIM30) .....	158
SPBRG Register .....	15
Special Event Trigger .....	120
Special Features of the CPU .....	129
Special Function Registers .....	14
Special Function Registers (SFRs) .....	14
SPI	
Associated Registers .....	90
Serial Clock .....	87
Serial Data In .....	87
Serial Data Out .....	87
Slave Select .....	87
SSP	
ACK .....	93
I <sup>2</sup> C	
I <sup>2</sup> C Operation .....	92
SSPADD Register .....	15
SSPBUF Register .....	14
SSPCON Register .....	14
SSPOV .....	89
SSPOV Bit .....	93
SSPSTAT Register .....	15
Stack .....	25
Overflows .....	25
Underflow .....	25
STATUS Register	
C Bit .....	17
DC Bit .....	17
IRP Bit .....	17
PD Bit .....	17, 132
RP Bits .....	17
TO Bit .....	17, 132
Z Bit .....	17
Synchronous Master Reception	
Associated Registers .....	110
Synchronous Master Transmission	
Associated Registers .....	109
Synchronous Serial Port (SSP) .....	87
Overview .....	87
SPI Mode .....	87
Synchronous Slave Reception	
Associated Registers .....	112
Synchronous Slave Transmission	
Associated Registers .....	111
<b>T</b>	
T1CKPS0 Bit .....	72
T1CKPS1 Bit .....	72
T1CON Register .....	14
T1OSCN Bit .....	72
T1SYN Bit .....	72
T2CKPS0 Bit .....	80
T2CKPS1 Bit .....	80
T2CON Register .....	14
TAD .....	118
Time-out Sequence .....	134

Timer0 .....	67
Associated Registers .....	69
Clock Source Edge Select (T0SE Bit) .....	18
Clock Source Select (T0CS Bit) .....	18
External Clock .....	68
Interrupt .....	67
Operation .....	67
Overflow Enable (TMR0IE Bit) .....	19
Overflow Flag (TMR0IF Bit) .....	140
Overflow Interrupt .....	140
Prescaler .....	68
T0CKI .....	68
Timer1 .....	71
Associated Registers .....	77
Capacitor Selection .....	75
Counter Operation .....	73
Operation .....	71
Operation in Asynchronous Counter Mode .....	74
Reading and Writing .....	74
Operation in Synchronized Counter Mode .....	73
Operation in Timer Mode .....	73
Oscillator .....	75
Oscillator Layout Considerations .....	75
Prescaler .....	76
Resetting Timer1 Register Pair .....	76
Resetting Timer1 Using a CCP Trigger Output .....	76
Use as a Real-Time Clock .....	76
Timer2 .....	79
Associated Registers .....	80
Output .....	79
Postscaler .....	79
Prescaler .....	79
Prescaler and Postscaler .....	79
Timing Diagrams	
A/D Conversion .....	191
Asynchronous Master Transmission .....	103
Asynchronous Master Transmission	
(Back to Back) .....	103
Asynchronous Reception .....	104
Asynchronous Reception with	
Address Byte First .....	107
Asynchronous Reception with	
Address Detect .....	107
AUSART Synchronous Receive	
(Master/Slave) .....	189
AUSART Synchronous Transmission	
(Master/Slave) .....	189
Brown-out Reset .....	181
Capture/Compare/PWM (CCP1) .....	183
CLKO and I/O .....	180
External Clock .....	179
Fail-Safe Clock Monitor .....	144
I <sup>2</sup> C Bus Data .....	187
I <sup>2</sup> C Bus Start/Stop Bits .....	186
I <sup>2</sup> C Reception (7-Bit Address) .....	94
I <sup>2</sup> C Transmission (7-Bit Address) .....	94
Primary System Clock After Reset	
(EC, RC, INTRC) .....	48
Primary System Clock After Reset	
(HS, XT, LP) .....	47
PWM Output .....	84
Reset, Watchdog Timer, Oscillator Start-up	
Timer and Power-up Timer .....	181
Slow Rise Time (MCLR Tied to VDD	
Through RC Network) .....	138

SPI Master Mode .....	91	TMR1ON Bit .....	72
SPI Master Mode (CKE = 0, SMP = 0) .....	184	TMR2 Register .....	14
SPI Master Mode (CKE = 1, SMP = 1) .....	184	TMR2ON Bit .....	80
SPI Slave Mode (CKE = 0) .....	91, 185	TOUTPS0 Bit .....	80
SPI Slave Mode (CKE = 1) .....	91, 185	TOUTPS1 Bit .....	80
Switching to SEC_RUN Mode .....	44	TOUTPS2 Bit .....	80
Synchronous Reception		TOUTPS3 Bit .....	80
(Master Mode, SREN) .....	111	TRISA Register .....	15, 51
Synchronous Transmission .....	109	TRISB Register .....	15, 16
Synchronous Transmission (Through TXEN) .....	109	Two-Speed Clock Start-up Mode .....	143
Time-out Sequence on Power-up ( $\overline{\text{MCLR}}$ )		Two-Speed Start-up .....	129
Tied to VDD Through Pull-up Resistor .....	137	TXREG Register .....	14
Time-out Sequence on Power-up ( $\overline{\text{MCLR}}$ )		TXSTA Register .....	15
Tied to VDD Through RC Network): Case 1 .....	137	BRGH Bit .....	97
Time-out Sequence on Power-up ( $\overline{\text{MCLR}}$ )		CSRC Bit .....	97
Tied to VDD Through RC Network): Case 2 .....	137	SYNC Bit .....	97
Timer0 and Timer1 External Clock .....	182	TRMT Bit .....	97
Timer1 Incrementing Edge .....	73	TX9 Bit .....	97
Transition Between SEC_RUN/RC_RUN		TX9D Bit .....	97
and Primary Clock .....	46	TXEN Bit .....	97
Two-Speed Start-up Mode .....	143		
Wake-up from Sleep via Interrupt .....	146	<b>V</b>	
XT, HS, LP, EC and EXTRC to		VDD Pin .....	9
RC_RUN Mode .....	43	Voltage Reference Specifications .....	177
Timing Parameter Symbology .....	178	VSS Pin .....	9
Timing Requirements		<b>W</b>	
A/D Conversion .....	191	Wake-up from Sleep .....	129, 146
AUSART Synchronous Receive .....	189	Interrupts .....	135
AUSART Synchronous Transmission .....	189	$\overline{\text{MCLR}}$ Reset .....	135
Capture/Compare/PWM (CCP1) .....	183	WDT Reset .....	135
CLKO and I/O .....	180	Wake-up Using Interrupts .....	146
External Clock .....	179	Watchdog Timer (WDT) .....	129, 141
I <sup>2</sup> C Bus Data .....	188	Associated Registers .....	142
I <sup>2</sup> C Bus Start/Stop Bits .....	187	WDT Reset, Normal Operation .....	132, 135
Reset, Watchdog Timer,		WDT Reset, Sleep .....	132, 135
Oscillator Start-up Timer,		WCOL .....	89
Power-up Timer and Brown-out Reset .....	181	WDTCON Register .....	16
SPI Mode .....	186	Write Collision Detect Bit, WCOL .....	89
Timer0 and Timer1 External Clock .....	182	WWW Address .....	223
TMR0 Register .....	14, 16	WWW, On-Line Support .....	4
TMR1CS Bit .....	72		
TMR1H Register .....	14		
TMR1L Register .....	14		

# PIC16F87/88

---

NOTES:



## THE MICROCHIP WEB SITE

Microchip provides online support via our WWW site at [www.microchip.com](http://www.microchip.com). This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

## CUSTOMER CHANGE NOTIFICATION SERVICE

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip web site at [www.microchip.com](http://www.microchip.com), click on Customer Change Notification and follow the registration instructions.

## CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support
- Development Systems Information Line

Customers should contact their distributor, representative or field application engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

**Technical support is available through the web site at: <http://support.microchip.com>**

In addition, there is a Development Systems Information Line which lists the latest versions of Microchip's development systems software products. This line also provides information on how customers can receive currently available upgrade kits.

**The Development Systems Information Line numbers are:**

1-800-755-2345 – United States and most of Canada

1-480-792-7302 – Other International Locations

# PIC16F87/88

---

---

## READER RESPONSE

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

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

To: Technical Publications Manager  
RE: Reader Response  
Total Pages Sent \_\_\_\_\_

From: Name \_\_\_\_\_  
Company \_\_\_\_\_  
Address \_\_\_\_\_  
City / State / ZIP / Country \_\_\_\_\_  
Telephone: (\_\_\_\_\_) \_\_\_\_\_ - \_\_\_\_\_ FAX: (\_\_\_\_\_) \_\_\_\_\_ - \_\_\_\_\_

Application (optional):

Would you like a reply? \_\_\_Y \_\_\_N

Device: PIC16F87/88

Literature Number: DS30487C

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?

---

---

## PIC16F87/88 PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

<u>PART NO.</u>	<u>X</u>	<u>/XX</u>	<u>XXX</u>
Device	Temperature Range	Package	Pattern
Device	PIC16F87: Standard VDD range PIC16F87T: (Tape and Reel) PIC16LF87: Extended VDD range		
Temperature Range	- = 0°C to +70°C I = -40°C to +85°C (Industrial) E = -40°C to +125°C (Extended)		
Package	P = PDIP SO = SOIC SS = SSOP ML = QFN		
Pattern	QTP, SQTP, ROM Code (factory specified) or Special Requirements. Blank for OTP and Windowed devices.		

**Examples:**

- a) PIC16F87-I/P = Industrial temp., PDIP package, Extended VDD limits.
- b) PIC16F87-I/SO = Industrial temp., SOIC package, normal VDD limits.

**Note 1:** F = CMOS Flash  
LF = Low-power CMOS Flash

**Note 2:** T = in tape and reel – SOIC, SSOP packages only.



---

---

## 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](http://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: 86-591-8750-3506  
Fax: 86-591-8750-3521

**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: 91-11-5160-8631  
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: 45-4450-2828  
Fax: 45-4485-2829

**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