

# **PIC16F7X7**

## **Data Sheet**

28/40/44-Pin, 8-Bit CMOS Flash  
Microcontrollers with 10-Bit A/D  
and 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in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates.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, rPIC, and SmartShunt are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.


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

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

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

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

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

 Printed on recycled paper.

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

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

## 28/40/44-Pin, 8-Bit CMOS Flash Microcontrollers with 10-Bit A/D and nanoWatt Technology

### Low-Power Features:

- Power-Managed modes:
  - Primary Run (XT, 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 (0.7  $\mu$ 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)

### Analog Features:

- 10-bit, up to 14-channel Analog-to-Digital Converter:
  - Programmable Acquisition Time
  - Conversion available during Sleep mode
- Dual Analog Comparators
- Programmable Low-Current Brown-out Reset (BOR) Circuitry and Programmable Low-Voltage Detect (LVD)

### Peripheral Features:

- High Sink/Source Current: 25 mA
- Two 8-bit Timers with Prescaler
- Timer1/RTC module:
  - 16-bit timer/counter with prescaler
  - Can be incremented during Sleep via external 32 kHz watch crystal
- Master Synchronous Serial Port (MSSP) with 3-wire SPI™ and I<sup>2</sup>C™ (Master and Slave) modes
- Addressable Universal Synchronous Asynchronous Receiver Transmitter (AUSART)
- Three Capture, Compare, PWM modules:
  - Capture is 16-bit, max. resolution is 12.5 ns
  - Compare is 16-bit, max. resolution is 200 ns
  - PWM max. resolution is 10 bits
- Parallel Slave Port (PSP) – 40/44-pin devices only

### Special Microcontroller Features:

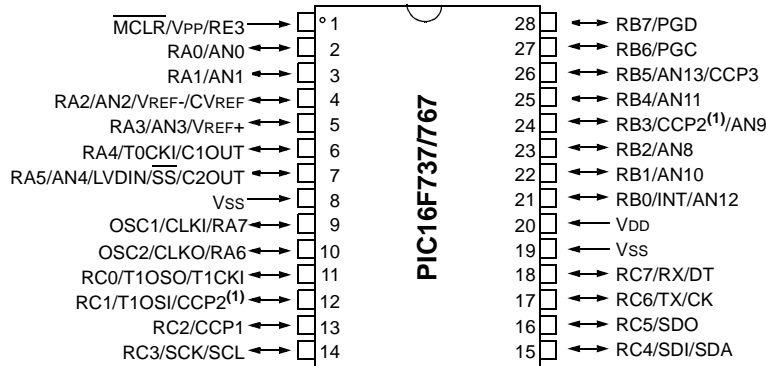
- Fail-Safe Clock Monitor for protecting critical applications against crystal failure
- Two-Speed Start-up mode for immediate code execution
- Power-on Reset (POR), Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Programmable Code Protection
- Processor Read Access to Program Memory
- Power-Saving Sleep mode
- In-Circuit Serial Programming™ (ICSP™) via two pins
- MPLAB® In-Circuit Debug (ICD) via two pins
- MCLR pin function replaceable with input only pin

Device	Program Memory (# Single-Word Instructions)	Data SRAM (Bytes)	I/O	Interrupts	10-bit A/D (ch)	Comparators	CCP (PWM)	MSSP		AUSART	Timers 8/16-bit
								SPI™	I <sup>2</sup> C™ (Master)		
PIC16F737	4096	368	25	16	11	2	3	Yes	Yes	Yes	2/1
PIC16F747	4096	368	36	17	14	2	3	Yes	Yes	Yes	2/1
PIC16F767	8192	368	25	16	11	2	3	Yes	Yes	Yes	2/1
PIC16F777	8192	368	36	17	14	2	3	Yes	Yes	Yes	2/1

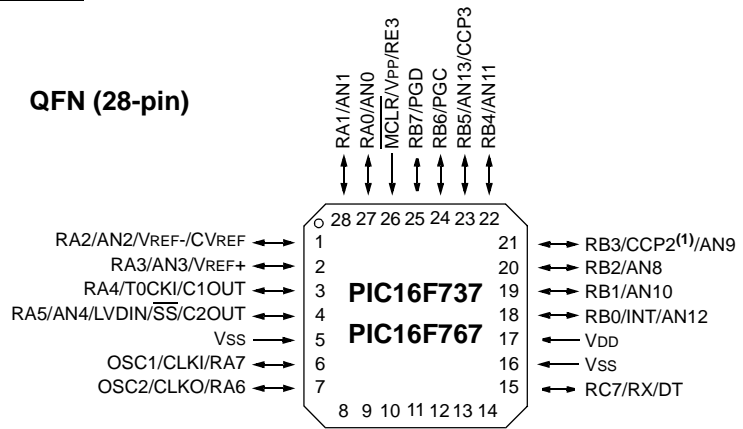
# PIC16F7X7

## Pin Diagrams

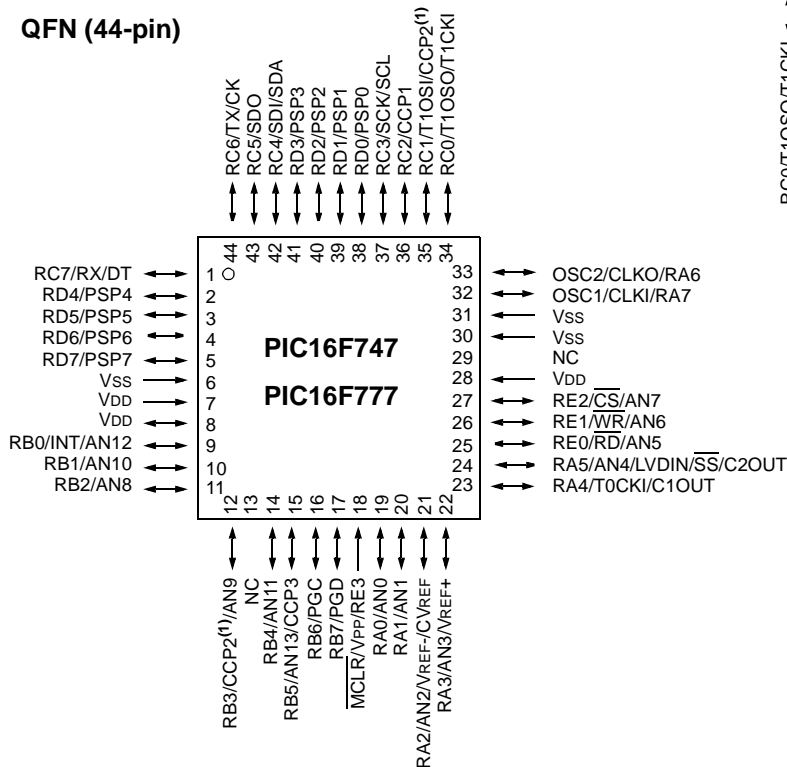
### PDIP, SOIC, SSOP (28-pin)



### QFN (28-pin)



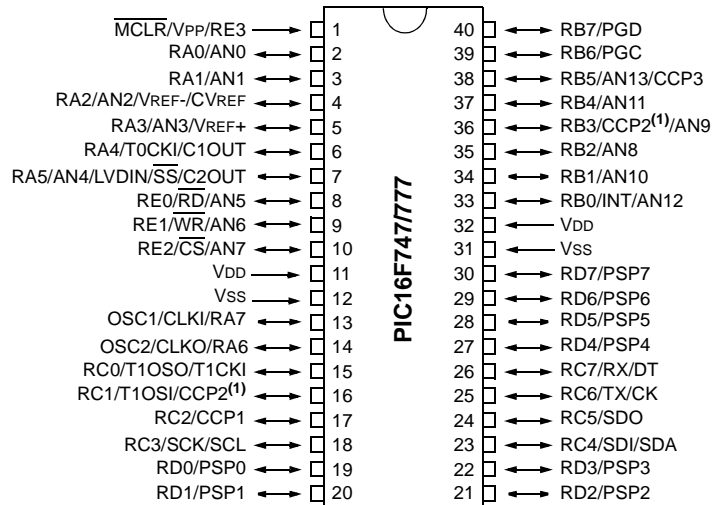
### QFN (44-pin)



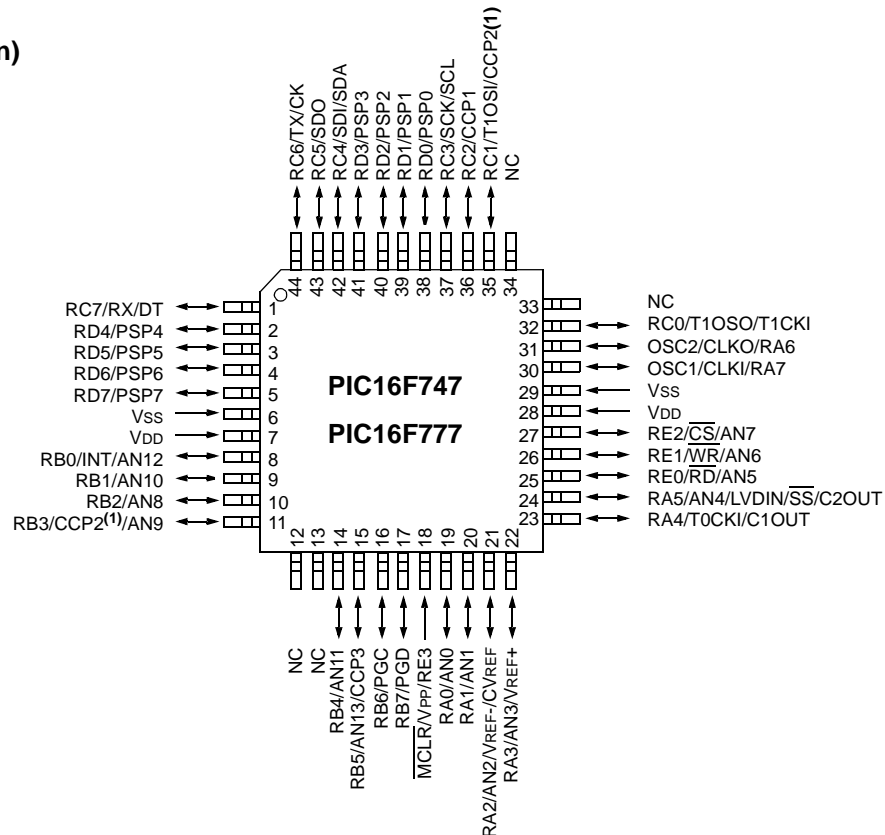
**Note 1:** Pin location of CCP2 is determined by the CCPMX bit in Configuration Word Register 1.

## Pin Diagrams (Continued)

### PDIP (40-pin)



### TQFP (44-pin)



**Note 1:** Pin location of CCP2 is determined by the CCPMX bit in Configuration Word Register 1.

# PIC16F7X7

## Table of Contents

1.0	Device Overview .....	5
2.0	Memory Organization .....	15
3.0	Reading Program Memory .....	31
4.0	Oscillator Configurations .....	33
5.0	I/O Ports .....	49
6.0	Timer0 Module .....	73
7.0	Timer1 Module .....	77
8.0	Timer2 Module .....	85
9.0	Capture/Compare/PWM Modules .....	87
10.0	Master Synchronous Serial Port (MSSP) Module .....	93
11.0	Addressable Universal Synchronous Asynchronous Receiver Transmitter (AUSART) .....	133
12.0	Analog-to-Digital Converter (A/D) Module .....	151
13.0	Comparator Module .....	161
14.0	Comparator Voltage Reference Module .....	167
15.0	Special Features of the CPU .....	169
16.0	Instruction Set Summary .....	193
17.0	Development Support .....	201
18.0	Electrical Characteristics .....	207
19.0	DC and AC Characteristics Graphs and Tables .....	237
20.0	Packaging Information .....	251
	Appendix A: Revision History .....	261
	Appendix B: Device Differences .....	261
	Appendix C: Conversion Considerations .....	262
	Index .....	263
	On-Line Support .....	271
	Systems Information and Upgrade Hot Line .....	271
	Reader Response .....	272
	PIC16F7X7 Product Identification System .....	273

## 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 about the following devices:

- PIC16F737
- PIC16F767
- PIC16F747
- PIC16F777

PIC16F737/767 devices are available only in 28-pin packages, while PIC16F747/777 devices are available in 40-pin and 44-pin packages. All devices in the PIC16F7X7 family share common architecture with the following differences:

- The PIC16F737 and PIC16F767 have one-half of the total on-chip memory of the PIC16F747 and PIC16F777.
- The 28-pin devices have 3 I/O ports, while the 40/44-pin devices have 5.
- The 28-pin devices have 16 interrupts, while the 40/44-pin devices have 17.
- The 28-pin devices have 11 A/D input channels, while the 40/44-pin devices have 14.
- The Parallel Slave Port is implemented only on the 40/44-pin devices.
- 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  $\mu$ A (previous PIC16 devices) to 1.8  $\mu$ A 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 15.17 “Watchdog Timer (WDT)”** for further details.
- Two-Speed Start-up: When the oscillator is configured for LP, XT or HS, 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 15.17.3 “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 available features are summarized in Table 1-1. Block diagrams of the PIC16F737/767 and PIC16F747/777 devices are provided in Figure 1-1 and Figure 1-2, respectively. The pinouts for these device families are listed in Table 1-2 and Table 1-3.

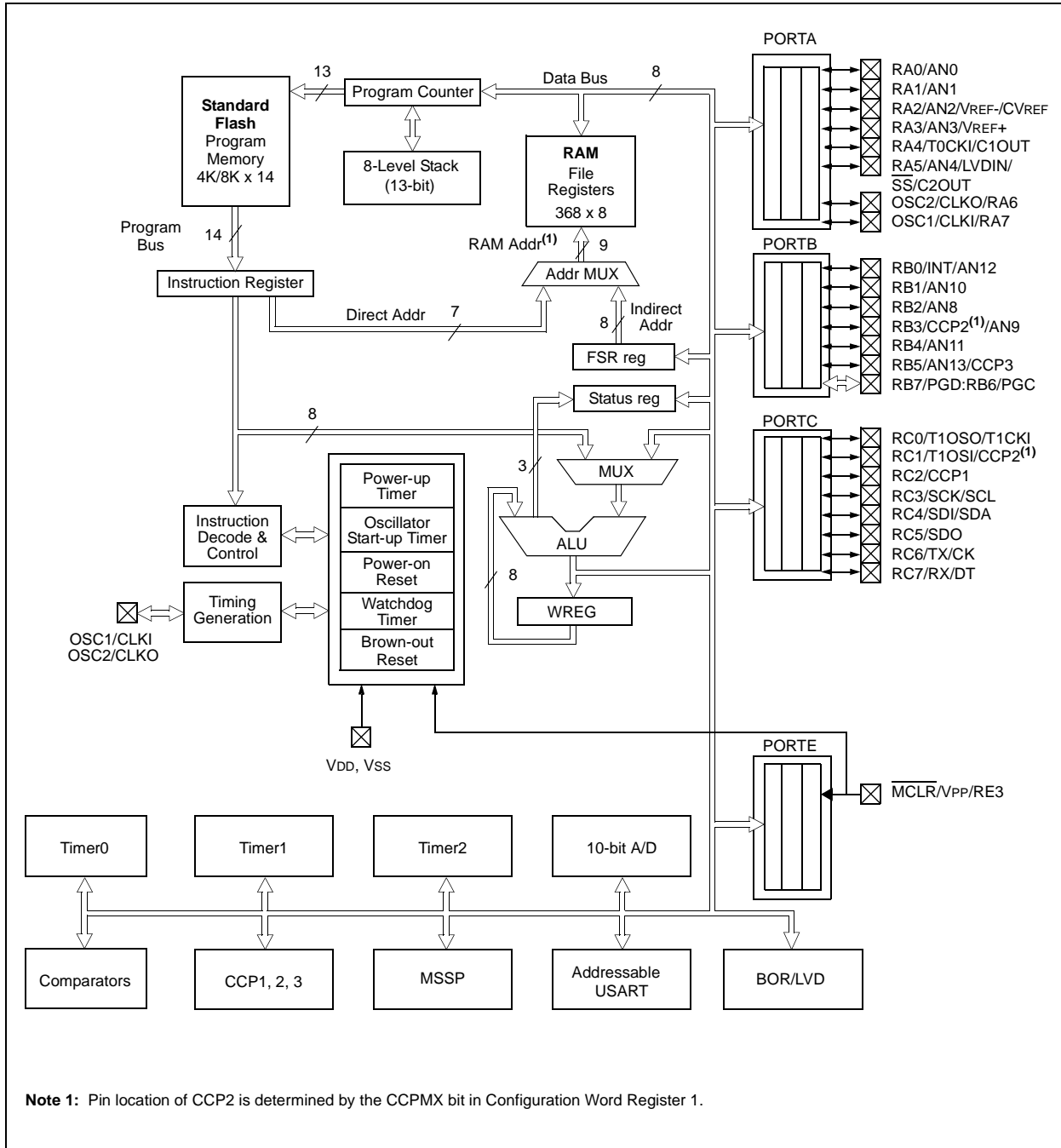
Additional information may be found in the “PICmicro® Mid-Range MCU Family Reference Manual” (DS33023) which may be obtained from your local Microchip Sales Representative or downloaded from the Microchip web site. The Reference Manual should be considered a complementary document to this data sheet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

**TABLE 1-1: PIC16F7X7 DEVICE FEATURES**

Key Features	PIC16F737	PIC16F747	PIC16F767	PIC16F777
Operating Frequency	DC – 20 MHz	DC – 20 MHz	DC – 20 MHz	DC – 20 MHz
Resets (and Delays)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)
Flash Program Memory (14-bit words)	4K	4K	8K	8K
Data Memory (bytes)	368	368	368	368
Interrupts	16	17	16	17
I/O Ports	Ports A, B, C	Ports A, B, C, D, E	Ports A, B, C	Ports A, B, C, D, E
Timers	3	3	3	3
Capture/Compare/PWM Modules	3	3	3	3
Master Serial Communications	MSSP, AUSART	MSSP, AUSART	MSSP, AUSART	MSSP, AUSART
Parallel Communications	—	PSP	—	PSP
10-bit Analog-to-Digital Module	11 Input Channels	14 Input Channels	11 Input Channels	14 Input Channels
Instruction Set	35 Instructions	35 Instructions	35 Instructions	35 Instructions
Packaging	28-pin PDIP 28-pin SOIC 28-pin SSOP 28-pin QFN	40-pin PDIP 44-pin QFN 44-pin TQFP	28-pin PDIP 28-pin SOIC 28-pin SSOP 28-pin QFN	40-pin PDIP 44-pin QFN 44-pin TQFP

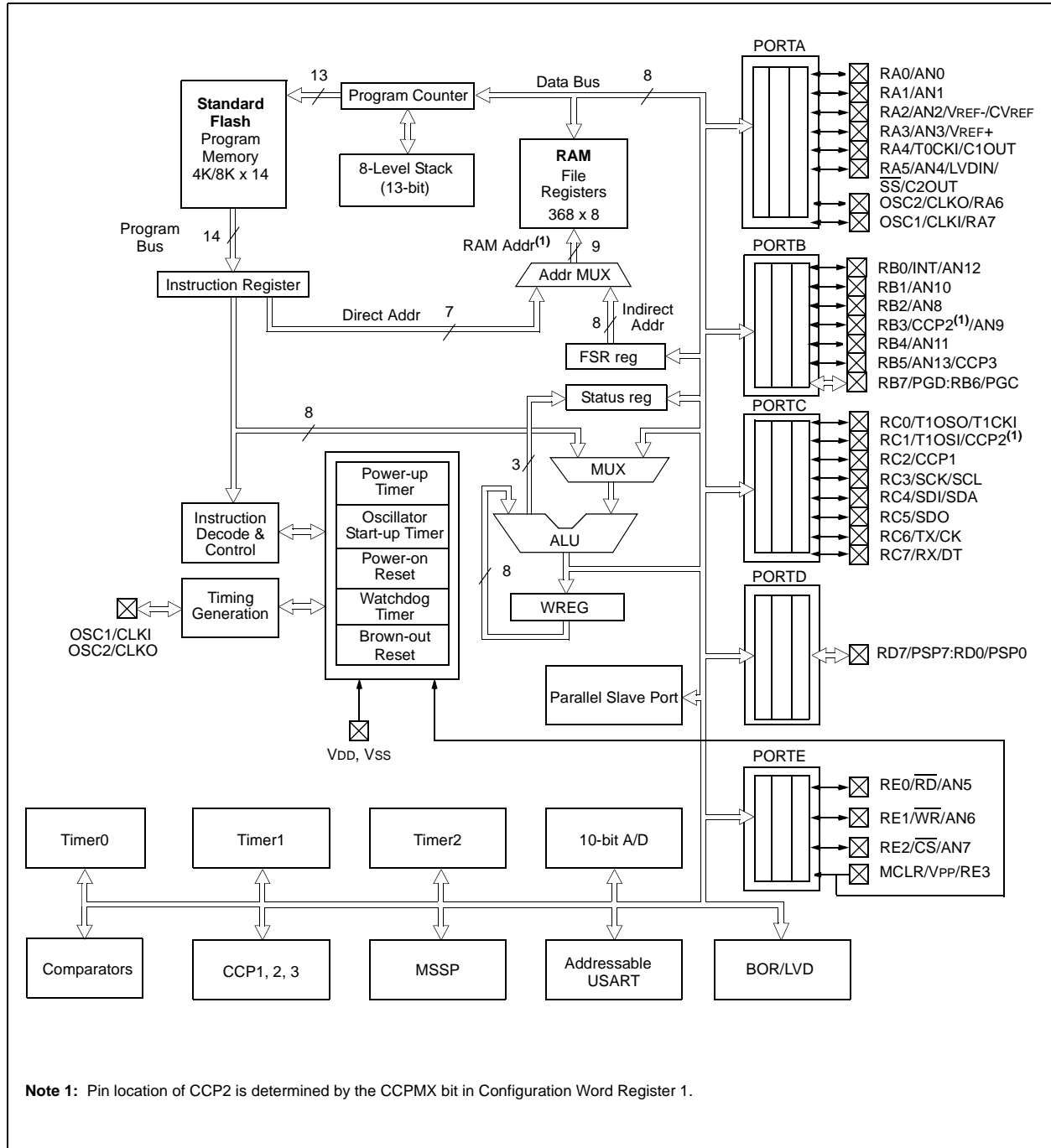
# PIC16F7X7

FIGURE 1-1: PIC16F737 AND PIC16F767 BLOCK DIAGRAM





**FIGURE 1-2: PIC16F747 AND PIC16F777 BLOCK DIAGRAM**



# PIC16F7X7

**TABLE 1-2: PIC16F737 AND PIC16F767 PINOUT DESCRIPTION**

Pin Name	PDIP SOIC SSOP Pin #	QFN Pin #	I/O/P Type	Buffer Type	Description
OSC1/CLKI/RA7 OSC1  CLKI  RA7	9	6	I  I  I/O	ST/CMOS <sup>(3)</sup>   ST	Oscillator crystal or external clock input. Oscillator crystal input or external clock source input. ST buffer when configured in RC mode; otherwise CMOS. External clock source input. Always associated with pin function OSC1 (see OSC1/CLKI, OSC2/CLKO pins). Digital I/O.
OSC2/CLKO/RA6 OSC2  CLKO  RA6	10	7	O  O  I/O	—   ST	Oscillator crystal or clock output. Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode. In RC mode, OSC2 pin outputs CLKO which has 1/4 the frequency of OSC1 and denotes the instruction cycle rate. Digital I/O.
MCLR/VPP/RE3 MCLR  VPP RE3	1	26	I  P I	ST   ST	Master Clear (input) or programming voltage (output). Master Clear (Reset) input. This pin is an active-low Reset to the device. Programming voltage input. Digital input only pin.
RA0/AN0 RA0 AN0  RA1/AN1 RA1 AN1  RA2/AN2/VREF-/CVREF RA2 AN2 VREF- CVREF  RA3/AN3/VREF+ RA3 AN3 VREF+  RA4/T0CKI/C1OUT RA4 T0CKI C1OUT  RA5/AN4/LVDIN/SS/C2OUT RA5 AN4 LVDIN SS C2OUT	2   3  4  5  6  7	27   28  1  2  3  4	I/O I  I/O I  I/O I I O  I/O I I/O I O	TTL   TTL  TTL  TTL  ST  TTL	PORTA is a bidirectional I/O port.  Digital I/O. Analog input 0.  Digital I/O. Analog input 1.  Digital I/O. Analog input 2. A/D reference voltage input (low). Comparator voltage reference output.  Digital I/O. Analog input 3. A/D reference voltage input (high).  Digital I/O – Open-drain when configured as output. Timer0 external clock input. Comparator 1 output bit.  Digital I/O. Analog input 4. Low-Voltage Detect input. SPI™ slave select input. Comparator 2 output bit.

**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: Pin location of CCP2 is determined by the CCPMX bit in Configuration Word Register 1.

**TABLE 1-2: PIC16F737 AND PIC16F767 PINOUT DESCRIPTION (CONTINUED)**

Pin Name	PDIP SOIC SSOP Pin #	QFN Pin #	I/O/P Type	Buffer Type	Description
RB0/INT/AN12 RB0 INT AN12	21	18	I/O I I	TTL/ST <sup>(1)</sup>	PORTB is a bidirectional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs.  Digital I/O. External interrupt. Analog input channel 12.
RB1/AN10 RB1 AN10	22	19	I/O I	TTL	Digital I/O. Analog input channel 10.
RB2/AN8 RB2 AN8	23	20	I/O I	TTL	Digital I/O. Analog input channel 8.
RB3/CCP2/AN9 RB3 CCP2 <sup>(4)</sup> AN9	24	21	I/O I/O I	TTL	Digital I/O. CCP2 capture input, compare output, PWM output. Analog input channel 9.
RB4/AN11 RB4 AN11	25	22	I/O I	TTL	Digital I/O. Analog input channel 11.
RB5/AN13/CCP3 RB5 AN13 CCP3	26	23	I/O I I/O	TTL	Digital I/O. Analog input channel 13. CCP3 capture input, compare output, PWM output.
RB6/PGC RB6 PGC	27	24	I/O I/O	TTL/ST <sup>(2)</sup>	Digital I/O. In-Circuit Debugger and ICSP™ programming clock.
RB7/PGD RB7 PGD	28	25	I/O I/O	TTL/ST <sup>(2)</sup>	Digital I/O. In-Circuit Debugger and ICSP programming data.

**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: Pin location of CCP2 is determined by the CCPMX bit in Configuration Word Register 1.

# PIC16F7X7

**TABLE 1-2: PIC16F737 AND PIC16F767 PINOUT DESCRIPTION (CONTINUED)**

Pin Name	PDIP SOIC SSOP Pin #	QFN Pin #	I/O/P Type	Buffer Type	Description
RC0/T1OSO/T1CKI RC0 T1OSO T1CKI	11	8	I/O O I	ST	PORTC is a bidirectional I/O port.  Digital I/O. Timer1 oscillator output. Timer1 external clock input.
RC1/T1OSI/CCP2 RC1 T1OSI CCP2 <sup>(4)</sup>	12	9	I/O I I/O	ST	Digital I/O. Timer1 oscillator input. Capture2 input, Compare2 output, PWM2 output.
RC2/CCP1 RC2 CCP1	13	10	I/O I/O	ST	Digital I/O. Capture1 input, Compare1 output, PWM1 output.
RC3/SCK/SCL RC3 SCK SCL	14	11	I/O I/O I/O	ST	Digital I/O. Synchronous serial clock input/output for SPI™ mode. Synchronous serial clock input/output for I <sup>2</sup> C™ mode.
RC4/SDI/SDA RC4 SDI SDA	15	12	I/O I I/O	ST	Digital I/O. SPI data in. I <sup>2</sup> C data I/O.
RC5/SDO RC5 SDO	16	13	I/O O	ST	Digital I/O. SPI data out.
RC6/TX/CK RC6 TX CK	17	14	I/O O I/O	ST	Digital I/O. AUSART asynchronous transmit. AUSART synchronous clock.
RC7/RX/DT RC7 RX DT	18	15	I/O I I/O	ST	Digital I/O. AUSART asynchronous receive. AUSART synchronous data.
V <sub>SS</sub>	8, 19	5, 16	P	—	Ground reference for logic and I/O pins.
V <sub>DD</sub>	20	17	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: Pin location of CCP2 is determined by the CCPMX bit in Configuration Word Register 1.

**TABLE 1-3: PIC16F747 AND PIC16F777 PINOUT DESCRIPTION**

Pin Name	PDIP Pin #	QFN Pin #	TQFP Pin #	I/O/P Type	Buffer Type	Description
OSC1/CLKI/RA7 OSC1  CLKI  RA7	13	32	30	I  I  I/O	ST/CMOS <sup>(4)</sup>   ST	Oscillator crystal or external clock input. Oscillator crystal input or external clock source input. ST buffer when configured in RC mode; otherwise CMOS. External clock source input. Always associated with pin function OSC1 (see OSC1/CLKI, OSC2/CLKO pins). Bidirectional I/O pin.
OSC2/CLKO/RA6 OSC2  CLKO  RA6	14	33	31	O  O  I/O	—   ST	Oscillator crystal or clock output. Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode. In RC mode, OSC2 pin outputs CLKO which has 1/4 the frequency of OSC1 and denotes the instruction cycle rate. Bidirectional I/O pin.
MCLR/VPP/RE3 MCLR  VPP RE3	1	18	18	I  P I	ST   ST	Master Clear (input) or programming voltage (output). Master Clear (Reset) input. This pin is an active-low Reset to the device. Programming voltage input. Digital input only pin.
RA0/AN0 RA0 AN0 RA1/AN1 RA1 AN1 RA2/AN2/VREF-/CVREF RA2 AN2 VREF- CVREF RA3/AN3/VREF+ RA3 AN3 VREF+ RA4/T0CKI/C1OUT RA4 T0CKI C1OUT RA5/AN4/LVDIN/SS/C2OUT RA5 AN4 LVDIN SS C2OUT	2   3   4   5   6   7	19   20   21   22   23   24	19   20   21   22   23   24	I/O I  I/O I I I  I/O I O  I/O I I I	TTL   TTL   TTL   TTL   ST   TTL	PORTA is a bidirectional I/O port.  Digital I/O. Analog input 0.  Digital I/O. Analog input 1.  Digital I/O. Analog input 2. A/D reference voltage input (low). Comparator voltage reference output.  Digital I/O. Analog input 3. A/D reference voltage input (high).  Digital I/O – Open-drain when configured as output. Timer0 external clock input. Comparator 1 output.  Digital I/O. Analog input 4. Low-Voltage Detect input. SPI™ slave select input. Comparator 2 output.

**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 an external interrupt.  
**Note 2:** This buffer is a Schmitt Trigger input when used in Serial Programming mode.  
**Note 3:** This buffer is a Schmitt Trigger input when configured as a general purpose I/O and a TTL input when used in the Parallel Slave Port mode (for interfacing to a microprocessor bus).  
**Note 4:** This buffer is a Schmitt Trigger input when configured in RC Oscillator mode and a CMOS input otherwise.  
**Note 5:** Pin location of CCP2 is determined by the CCPMX bit in Configuration Word Register 1.

# PIC16F7X7

**TABLE 1-3: PIC16F747 AND PIC16F777 PINOUT DESCRIPTION (CONTINUED)**

Pin Name	PDIP Pin #	QFN Pin #	TQFP Pin #	I/O/P Type	Buffer Type	Description
RB0/INT/AN12 RB0 INT AN12	33	9	8	I/O I I	TTL/ST <sup>(1)</sup>	PORTB is a bidirectional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs.  Digital I/O. External interrupt. Analog input channel 12.
RB1/AN10 RB1 AN10	34	10	9	I/O I	TTL	Digital I/O. Analog input channel 10.
RB2/AN8 RB2 AN8	35	11	10	I/O I	TTL	Digital I/O. Analog input channel 8.
RB3/CCP2/AN9 RB3 CCP2 <sup>(5)</sup> AN9	36	12	11	I/O I/O I	TTL	Digital I/O. CCP2 capture input, compare output, PWM output. Analog input channel 9.
RB4/AN11 RB4 AN11	37	14	14	I/O I	TTL	Digital I/O. Analog input channel 11
RB5/AN13/CCP3 RB5 AN13 CCP3	38	15	15	I/O I I	TTL	Digital I/O. Analog input channel 13. CCP3 capture input, compare output, PWM output.
RB6/PGC RB6 PGC	39	16	16	I/O I/O	TTL/ST <sup>(2)</sup>	Digital I/O. In-Circuit Debugger and ICSP™ programming clock.
RB7/PGD RB7 PGD	40	17	17	I/O I/O	TTL/ST <sup>(2)</sup>	Digital I/O. In-Circuit Debugger and ICSP programming data.

**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 an 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 as a general purpose I/O and a TTL input when used in the Parallel Slave Port mode (for interfacing to a microprocessor bus).  
 4: This buffer is a Schmitt Trigger input when configured in RC Oscillator mode and a CMOS input otherwise.  
 5: Pin location of CCP2 is determined by the CCPMX bit in Configuration Word Register 1.

**TABLE 1-3: PIC16F747 AND PIC16F777 PINOUT DESCRIPTION (CONTINUED)**

Pin Name	PDIP Pin #	QFN Pin #	TQFP Pin #	I/O/P Type	Buffer Type	Description
RC0/T1OSO/T1CKI RC0 T1OSO T1CKI	15	34	32	I/O O I	ST	PORTC is a bidirectional I/O port.  Digital I/O. Timer1 oscillator output. Timer1 external clock input.
RC1/T1OSI/CCP2 RC1 T1OSI CCP2 <sup>(5)</sup>	16	35	35	I/O I I/O	ST	Digital I/O. Timer1 oscillator input. Capture 2 input, Compare 2 output, PWM 2 output.
RC2/CCP1 RC2 CCP1	17	36	36	I/O I/O	ST	Digital I/O. Capture 1 input, Compare 1 output, PWM 1 output.
RC3/SCK/SCL RC3 SCK  SCL	18	37	37	I/O I/O  I/O	ST	Digital I/O. Synchronous serial clock input/output for SPI™ mode. Synchronous serial clock input/output for I <sup>2</sup> C™ mode.
RC4/SDI/SDA RC4 SDI SDA	23	42	42	I/O I I/O	ST	Digital I/O. SPI data in. I <sup>2</sup> C data I/O.
RC5/SDO RC5 SDO	24	43	43	I/O O	ST	Digital I/O. SPI data out.
RC6/TX/CK RC6 TX CK	25	44	44	I/O O I/O	ST	Digital I/O. AUSART asynchronous transmit. AUSART synchronous clock.
RC7/RX/DT RC7 RX DT	26	1	1	I/O I I/O	ST	Digital I/O. AUSART asynchronous receive. AUSART synchronous data.

**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 an 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 as a general purpose I/O and a TTL input when used in the Parallel Slave Port mode (for interfacing to a microprocessor bus).
  - 4: This buffer is a Schmitt Trigger input when configured in RC Oscillator mode and a CMOS input otherwise.
  - 5: Pin location of CCP2 is determined by the CCPMX bit in Configuration Word Register 1.

# PIC16F7X7

**TABLE 1-3: PIC16F747 AND PIC16F777 PINOUT DESCRIPTION (CONTINUED)**

Pin Name	PDIP Pin #	QFN Pin #	TQFP Pin #	I/O/P Type	Buffer Type	Description
RD0/PSP0 RD0 PSP0	19	38	38	I/O I/O	ST/TTL <sup>(3)</sup>	PORTD is a bidirectional I/O port or Parallel Slave Port when interfacing to a microprocessor bus.  Digital I/O. Parallel Slave Port data.
RD1/PSP1 RD1 PSP1	20	39	39	I/O I/O	ST/TTL <sup>(3)</sup>	Digital I/O. Parallel Slave Port data.
RD2/PSP2 RD2 PSP2	21	40	40	I/O I/O	ST/TTL <sup>(3)</sup>	Digital I/O. Parallel Slave Port data.
RD3/PSP3 RD3 PSP3	22	41	41	I/O I/O	ST/TTL <sup>(3)</sup>	Digital I/O. Parallel Slave Port data.
RD4/PSP4 RD4 PSP4	27	2	2	I/O I/O	ST/TTL <sup>(3)</sup>	Digital I/O. Parallel Slave Port data.
RD5/PSP5 RD5 PSP5	28	3	3	I/O I/O	ST/TTL <sup>(3)</sup>	Digital I/O. Parallel Slave Port data.
RD6/PSP6 RD6 PSP6	29	4	4	I/O I/O	ST/TTL <sup>(3)</sup>	Digital I/O. Parallel Slave Port data.
RD7/PSP7 RD7 PSP7	30	5	5	I/O I/O	ST/TTL <sup>(3)</sup>	Digital I/O. Parallel Slave Port data.
RE0/RD/AN5 RE0 RD AN5	8	25	25	I/O I I	ST/TTL <sup>(3)</sup>	PORTE is a bidirectional I/O port.  Digital I/O. Read control for Parallel Slave Port. Analog input 5.
RE1/WR/AN6 RE1 WR AN6	9	26	26	I/O I I	ST/TTL <sup>(3)</sup>	Digital I/O. Write control for Parallel Slave Port. Analog input 6.
RE2/CS/AN7 RE2 CS AN7	10	27	27	I/O I I	ST/TTL <sup>(3)</sup>	Digital I/O. Chip select control for Parallel Slave Port. Analog input 7.
Vss	—	31	—	P	—	Analog ground reference.
Vss	12, 31	6, 30	6, 29	P	—	Ground reference for logic and I/O pins.
VDD	—	8	—	P	—	Analog positive supply.
VDD	11, 32	7, 28	7, 28	P	—	Positive supply for logic and I/O pins.
NC	—	13, 29	12, 13, 33, 34	—	—	These pins are not internally connected. These pins should be left unconnected.

**Legend:** I = input                      O = output                      I/O = input/output                      P = power  
 — = Not used                      TTL = TTL input                      ST = Schmitt Trigger input

- Note**
- This buffer is a Schmitt Trigger input when configured as an external interrupt.
  - This buffer is a Schmitt Trigger input when used in Serial Programming mode.
  - This buffer is a Schmitt Trigger input when configured as a general purpose I/O and a TTL input when used in the Parallel Slave Port mode (for interfacing to a microprocessor bus).
  - This buffer is a Schmitt Trigger input when configured in RC Oscillator mode and a CMOS input otherwise.
  - Pin location of CCP2 is determined by the CCPMX bit in Configuration Word Register 1.



## 2.0 MEMORY ORGANIZATION

There are two memory blocks in each of these PICmicro® MCUs. The program memory and data memory have separate buses so that concurrent access can occur and is detailed in this section. The program memory can be read internally by user code (see **Section 3.0 “Reading 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 PIC16F7X7 devices have a 13-bit program counter capable of addressing an 8K word x 14-bit program memory space. The PIC16F767/777 devices have 8K words of Flash program memory and the PIC16F737/747 devices have 4K words. The program memory maps for PIC16F7X7 devices are shown in Figure 2-1. Accessing a location above the physically implemented address will cause a wraparound.

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

## 2.2 Data Memory Organization

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

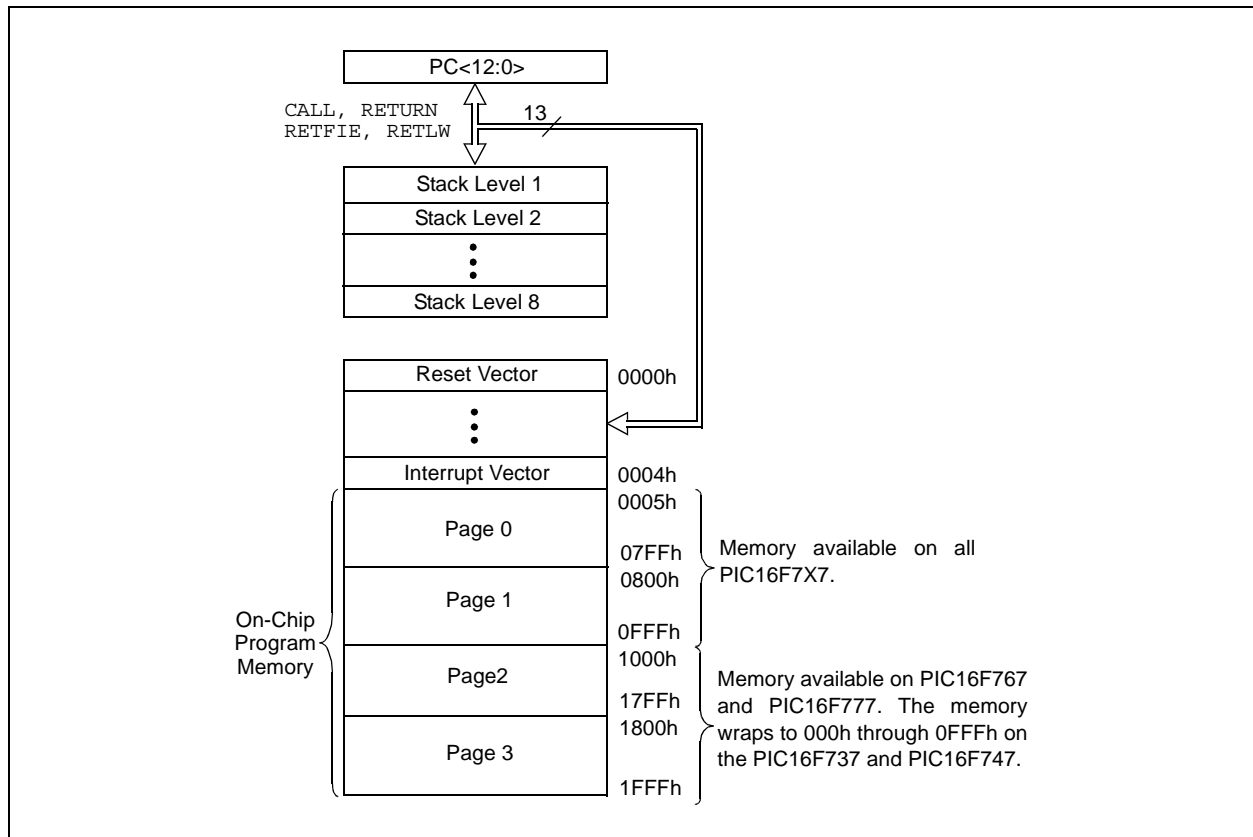
RP1:RP0	Bank
00	0
01	1
10	2
11	3

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 Special Function Registers. Some frequently used Special Function Registers from one bank may be mirrored in another bank for code reduction and quicker access.

### 2.2.1 GENERAL PURPOSE REGISTER FILE

The register file (shown in Figure 2-2 and Figure 2-3) can be accessed either directly, or indirectly, through the File Select Register (FSR).

**FIGURE 2-1: PROGRAM MEMORY MAPS AND STACKS FOR PIC16F7X7 DEVICES**



# PIC16F7X7

**FIGURE 2-2: DATA MEMORY MAP FOR PIC16F737 AND THE PIC16F767**

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
PORTC	07h	TRISC	87h		107h		187h
	08h		88h		108h		188h
PORTE	09h	TRISE	89h	LVDCON	109h		189h
PCLATH	0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah
INTCON	0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh
PIR1	0Ch	PIE1	8Ch	PMDATA	10Ch	PMCON1	18Ch
PIR2	0Dh	PIE2	8Dh	PMADR	10Dh		18Dh
TMR1L	0Eh	PCON	8Eh	PMDATH	10Eh		18Eh
TMR1H	0Fh	OSCCON	8Fh	PMADRH	10Fh		18Fh
T1CON	10h	OSCTUNE	90h	General Purpose Register 16 Bytes	110h	General Purpose Register 16 Bytes	190h
TMR2	11h	SSPCON2	91h				
T2CON	12h	PR2	92h				
SSPBUF	13h	SSPADD	93h				
SSPCON	14h	SSPSTAT	94h				
CCPR1L	15h	CCPR3L	95h				
CCPR1H	16h	CCPR3H	96h				
CCP1CON	17h	CCP3CON	97h				
RCSTA	18h	TXSTA	98h				
TXREG	19h	SPBRG	99h				
RCREG	1Ah		9Ah				
CCPR2L	1Bh	ADCON2	9Bh				
CCPR2H	1Ch	CMCON	9Ch				
CCP2CON	1Dh	CVRCON	9Dh				
ADRESH	1Eh	ADRESL	9Eh				
ADCON0	1Fh	ADCON1	9Fh				
General Purpose Register 96 Bytes	20h	General Purpose Register 80 Bytes	A0h	General Purpose Register 80 Bytes	11Fh	General Purpose Register 80 Bytes	19Fh
			EFh		120h		1A0h
			F0h		16Fh		1EFh
		Accesses 70h-7Fh		Accesses 70h-7Fh	170h	Accesses 70h-7Fh	1F0h
			FFh		17Fh		1FFh
Bank 0	7Fh	Bank 1	FFh	Bank 2	17Fh	Bank 3	1FFh

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

**FIGURE 2-3: DATA MEMORY MAP FOR PIC16F747 AND THE PIC16F777**

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
PORTC	07h	TRISC	87h		107h		187h
PORTD	08h	TRISD	88h		108h		188h
PORTE	09h	TRISE	89h	LVDCON	109h		189h
PCLATH	0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah
INTCON	0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh
PIR1	0Ch	PIE1	8Ch	PMDATA	10Ch	PMCON1	18Ch
PIR2	0Dh	PIE2	8Dh	PMADR	10Dh		18Dh
TMR1L	0Eh	PCON	8Eh	PMDATH	10Eh		18Eh
TMR1H	0Fh	OSCCON	8Fh	PMADRH	10Fh		18Fh
T1CON	10h	OSCTUNE	90h		110h		190h
TMR2	11h	SSPCON2	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	CCPR3L	95h				
CCPR1H	16h	CCPR3H	96h				
CCP1CON	17h	CCP3CON	97h				
RCSTA	18h	TXSTA	98h				
TXREG	19h	SPBRG	99h				
RCREG	1Ah		9Ah				
CCPR2L	1Bh	ADCON2	9Bh				
CCPR2H	1Ch	CMCON	9Ch				
CCP2CON	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	
			EFh		16Fh		1EFh
		Accesses 70h-7Fh	F0h	Accesses 70h-7Fh	170h	Accesses 70h-7Fh	1F0h
			FFh		17Fh		1FFh
Bank 0	7Fh	Bank 1	FFh	Bank 2	17Fh	Bank 3	1FFh

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

# PIC16F7X7

## 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>(4)</sup>	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)									0000 0000	30, 180
01h	TMR0	Timer0 Module Register									xxxx xxxx	76, 180
02h <sup>(4)</sup>	PCL	Program Counter (PC) Least Significant Byte									0000 0000	29, 180
03h <sup>(4)</sup>	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	21, 180	
04h <sup>(4)</sup>	FSR	Indirect Data Memory Address Pointer									xxxx xxxx	30, 180
05h	PORTA	PORTA Data Latch when written: PORTA pins when read									xx0x 0000	55, 180
06h	PORTB	PORTB Data Latch when written: PORTB pins when read									xx00 0000	64, 180
07h	PORTC	PORTC Data Latch when written: PORTC pins when read									xxxx xxxx	66, 180
08h <sup>(5)</sup>	PORTD	PORTD Data Latch when written: PORTD pins when read									xxxx xxxx	67, 180
09h <sup>(5)</sup>	PORTE	—	—	—	—	RE3	RE2	RE1	RE0	---- x000	68, 180	
0Ah <sup>(1,4)</sup>	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	29, 180	
0Bh <sup>(4)</sup>	INTCON	GIE	PEIE	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	23, 180	
0Ch	PIR1	PSPIF <sup>(3)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	25, 180	
0Dh	PIR2	OSFIF	CMIF	LVDIF	—	BCLIF	—	CCP3IF	CCP2IF	000- 0-00	27, 180	
0Eh	TMR1L	Holding Register for the Least Significant Byte of the 16-bit TMR1 Register									xxxx xxxx	83, 180
0Fh	TMR1H	Holding Register for the Most Significant Byte of the 16-bit TMR1 Register									xxxx xxxx	83, 180
10h	T1CON	—	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{T1SYNC}$	TMR1CS	TMR1ON	-00 0000	83, 180	
11h	TMR2	Timer2 Module Register									0000 0000	86, 180
12h	T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-00 0000	86, 180	
13h	SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register									xxxx xxxx	101, 180
14h	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	101, 180	
15h	CCPR1L	Capture/Compare/PWM Register 1 (LSB)									xxxx xxxx	90, 180
16h	CCPR1H	Capture/Compare/PWM Register 1 (MSB)									xxxx xxxx	90, 180
17h	CCP1CON	—	—	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	88, 180	
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	134, 180	
19h	TXREG	AUSART Transmit Data Register									0000 0000	139, 180
1Ah	RCREG	AUSART Receive Data Register									0000 0000	141, 180
1Bh	CCPR2L	Capture/Compare/PWM Register 2 (LSB)									xxxx xxxx	92, 180
1Ch	CCPR2H	Capture/Compare/PWM Register 2 (MSB)									xxxx xxxx	92, 180
1Dh	CCP2CON	—	—	CCP2X	CCP2Y	CCP2M3	CCP2M2	CCP2M1	CCP2M0	--00 0000	88, 180	
1Eh	ADRESH	A/D Result Register High Byte									xxxx xxxx	160, 180
1Fh	ADCON0	ADCS1	ADCS0	CHS2	CHS1	CHS0	$\overline{GO/DONE}$	CHS3	ADON	0000 0000	152, 180	

**Legend:** x = unknown, u = unchanged, c = 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 the PC<12:8> bits, whose contents are transferred to the upper byte of the program counter during branches (CALL or GOTO).
- 2:** Other (non Power-up) Resets include external Reset through MCLR and Watchdog Timer Reset.
- 3:** Bits PSPIE and PSPIF are reserved on the 28-pin devices; always maintain these bits clear.
- 4:** These registers can be addressed from any bank.
- 5:** PORTD, PORTE, TRISD and TRISE are not physically implemented on the 28-pin devices (except for RE3), read as '0'.
- 6:** This bit always reads as a '1'.
- 7:** OSCCON<OSTS> bit resets to '0' with dual-speed start-up and LP, HS or HS-PLL selected as the oscillator.
- 8:** RE3 is an input only. The state of the TRISE3 bit has no effect and will always read '1'.

**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>(4)</sup>	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)									0000 0000	30, 180
81h	OPTION_REG	RBP $\bar{U}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	22, 180	
82h <sup>(4)</sup>	PCL	Program Counter's (PC) Least Significant Byte									0000 0000	29, 180
83h <sup>(4)</sup>	STATUS	IRP	RP1	RP0	$\bar{T}O$	$\bar{P}D$	Z	DC	C	0001 1xxx	21, 180	
84h <sup>(4)</sup>	FSR	Indirect Data Memory Address Pointer									xxxx xxxx	30, 180
85h	TRISA	PORTA Data Direction Register									1111 1111	55, 181
86h	TRISB	PORTB Data Direction Register									1111 1111	64, 181
87h	TRISC	PORTC Data Direction Register									1111 1111	66, 181
88h <sup>(5)</sup>	TRISD	PORTD Data Direction Register									1111 1111	67, 181
89h <sup>(5)</sup>	TRISE	IBF <sup>(5)</sup>	OBF <sup>(5)</sup>	IBOV <sup>(5)</sup>	PSPMODE <sup>(5)</sup>	— <sup>(8)</sup>	PORTE Data Direction bits				0000 1111	69, 181
8Ah <sup>(1,4)</sup>	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	23, 180	
8Bh <sup>(4)</sup>	INTCON	GIE	PEIE	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	25, 180	
8Ch	PIE1	PSPIE <sup>(3)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	24, 181	
8Dh	PIE2	OSFIE	CMIE	LVDIE	—	BCLIE	—	CCP3IE	CCP2IE	000- 0-00	26, 181	
8Eh	PCON	—	—	—	—	—	SBOREN	$\bar{P}OR$	$\bar{B}OR$	---- -1qq	28, 181	
8Fh	OSCCON	—	IRCF2	IRCF1	IRCF0	OSTS <sup>(7)</sup>	IOFS	SCS1	SCS0	-000 1000	38, 181	
90h	OSCTUNE	—	—	TUN5	TUN4	TUN3	TUN2	TUN1	TUN0	--00 0000	36, 181	
91h	SSPCON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	0000 0000	105	
92h	PR2	Timer2 Period Register									1111 1111	86, 181
93h	SSPADD	Synchronous Serial Port (I <sup>2</sup> C™ mode) Address Register									0000 0000	101, 181
94h	SSPSTAT	SMP	CKE	D/ $\bar{A}$	P	S	R/ $\bar{W}$	UA	BF	0000 0000	101, 181	
95h	CCPR3L	Capture/Compare/PWM Register 3 (LSB)									xxxx xxxx	92
96h	CCPR3H	Capture/Compare/PWM Register 3 (MSB)									xxxx xxxx	92
97h	CCP3CON	—	—	CCP3X	CCP3Y	CCP3M3	CCP3M2	CCP3M1	CCP3M0	--00 0000	92	
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	145, 181	
99h	SPBRG	Baud Rate Generator Register									0000 0000	145, 181
9Ah	—	Unimplemented									—	—
9Bh	ADCON2	—	—	ACQT2	ACQT1	ACQT0	—	—	—	--00 0---	154	
9Ch	CMCON	C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0	0000 0111	55, 161	
9Dh	CVRCON	CVREN	CVROE	CVRR	—	CVR3	CVR2	CVR1	CVR0	000- 0000	55, 167	
9Eh	ADRESL	A/D Result Register Low Byte									xxxx xxxx	180
9Fh	ADCON1	ADFM	ADCS2	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0	0000 0000	153, 181	

**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 the PC<12:8> bits, whose contents are transferred to the upper byte of the program counter during branches (CALL or GOTO).
- Other (non Power-up) Resets include external Reset through MCLR and Watchdog Timer Reset.
  - Bits PSPIE and PSPIF are reserved on the 28-pin devices; always maintain these bits clear.
  - These registers can be addressed from any bank.
  - PORTD, PORTE, TRISD and TRISE are not physically implemented on the 28-pin devices (except for RE3), read as '0'.
  - This bit always reads as a '1'.
  - OSCCON<OSTS> bit resets to '0' with dual-speed start-up and LP, HS or HS-PLL selected as the oscillator.
  - RE3 is an input only. The state of the TRISE3 bit has no effect and will always read '1'.

# PIC16F7X7

**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>(4)</sup>	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)									0000 0000	30, 180
101h	TMR0	Timer0 Module Register									xxxx xxxx	76, 180
102h <sup>(4)</sup>	PCL	Program Counter (PC) Least Significant Byte									0000 0000	29, 180
103h <sup>(4)</sup>	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxxx	21, 180	
104h <sup>(4)</sup>	FSR	Indirect Data Memory Address Pointer									xxxx xxxx	30, 180
105h	WDTCON	—	—	—	WDTPS3	WDTPS2	WDTPS1	WDTPS0	SWDTEN	---0 1000	187	
106h	PORTB	PORTB Data Latch when written: PORTB pins when read									xxxx xxxx	64, 180
107h	—	Unimplemented									—	—
108h	—	Unimplemented									—	—
109h	LVDCON	—	—	IRVST	LVDEN	LVDL3	LVDL2	LVDL1	LVDL0	--00 0101	176	
10Ah <sup>(1,4)</sup>	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	23, 180	
10Bh <sup>(4)</sup>	INTCON	GIE	PEIE	TMROIE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	25, 180	
10Ch	PMDATA	EEPROM Data Register Low Byte									xxxx xxxx	32, 181
10Dh	PMADR	EEPROM Address Register Low Byte									xxxx xxxx	32, 181
10Eh	PMDATH	—	—	EEPROM Data Register High Byte					---x xxxx	32, 181		
10Fh	PMADRH	—	—	—	—	EEPROM Address Register High Byte				---x xxxx	32, 181	
<b>Bank 3</b>												
180h <sup>(4)</sup>	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)									0000 0000	30, 180
181h	OPTION_REG	$\overline{RBPU}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	22, 180	
182h <sup>(4)</sup>	PCL	Program Counter (PC) Least Significant Byte									0000 0000	29, 180
183h <sup>(4)</sup>	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxxx	21, 180	
184h <sup>(4)</sup>	FSR	Indirect Data Memory Address Pointer									xxxx xxxx	30, 180
185h	—	Unimplemented									—	—
186h	TRISB	PORTB Data Direction Register									1111 1111	64, 181
187h	—	Unimplemented									—	—
188h	—	Unimplemented									—	—
189h	—	Unimplemented									—	—
18Ah <sup>(1,4)</sup>	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	23, 180	
18Bh <sup>(4)</sup>	INTCON	GIE	PEIE	TMROIE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	25, 180	
18Ch	PMCON1	r <sup>(6)</sup>	—	—	—	—	—	—	RD	1--- ---0	32, 181	
18Dh	—	Reserved, maintain clear									—	—
18Eh	—	Reserved, maintain clear									—	—
18Fh	—	Reserved, maintain clear									—	—

**Legend:** x = unknown, u = unchanged, c = 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 the PC<12:8> bits, whose contents are transferred to the upper byte of the program counter during branches (CALL or GOTO).
- 2:** Other (non Power-up) Resets include external Reset through MCLR and Watchdog Timer Reset.
- 3:** Bits PSPIE and PSPIF are reserved on the 28-pin devices; always maintain these bits clear.
- 4:** These registers can be addressed from any bank.
- 5:** PORTD, PORTE, TRISD and TRISE are not physically implemented on the 28-pin devices (except for RE3), read as '0'.
- 6:** This bit always reads as a '1'.
- 7:** OSCCON<OSTS> bit resets to '0' with dual-speed start-up and LP, HS or HS-PLL selected as the oscillator.
- 8:** RE3 is an input only. The state of the TRISE3 bit has no effect and will always read '1'.

## 2.2.2.1 Status Register

The Status register 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 1:** 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 **RP1:RP0:** 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`, `SUBWF` instructions)  
 1 = A carry-out from the 4th low-order bit of the result occurred  
 0 = No carry-out from the 4th low-order bit of the result
  - bit 0 **C:** Carry/borrow bit (`ADDWF`, `ADDLW`, `SUBLW`, `SUBWF` instructions)  
 1 = A carry-out from the Most Significant bit of the result occurred  
 0 = No carry-out from the Most Significant bit of the result occurred
- Note:** For borrow, the polarity is reversed. A subtraction is executed by adding the two's complement of the second operand. For rotate (`RRF`, `RLF`) instructions, this bit is loaded with either the high or low-order bit of the source register.

#### Legend:

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

# PIC16F7X7

## 2.2.2.2 OPTION\_REG Register

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

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

### REGISTER 2-2: OPTION\_REG: OPTION 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
	$\overline{\text{RBP}}\text{U}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
bit 7								bit 0

- bit 7  **$\overline{\text{RBP}}\text{U}$** : 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 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 pin  
 0 = Increment on low-to-high transition on RA4/T0CKI pin
- bit 3 **PSA**: Prescaler Assignment bit  
 1 = Prescaler is assigned to the WDT  
 0 = Prescaler is assigned to the Timer0 module
- bit 2-0 **PS2:PS0**: Prescaler Rate Select bits

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

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



## 2.2.2.3 INTCON Register

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

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

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

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE	PEIE	TMR0IE	INTOIE	RBIE	TMR0IF	INTOIF	RBIF
							bit 0
							bit 7

- 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 **INTOIE:** 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 **INTOIF:** 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

# PIC16F7X7

## 2.2.2.4 PIE1 Register

The PIE1 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)

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

bit 7 **PSPIE:** Parallel Slave Port Read/Write Interrupt Enable bit<sup>(1)</sup>

1 = Enables the PSP read/write interrupt  
0 = Disables the PSP read/write interrupt

**Note 1:** PSPIE is reserved on 28-pin devices; always maintain this bit clear.

bit 6 **ADIE:** A/D Converter Interrupt Enable bit

1 = Enables the A/D converter interrupt  
0 = Disables the A/D converter interrupt

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

1 = Enables the AUSART receive interrupt  
0 = Disables the AUSART receive interrupt

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

1 = Enables the AUSART transmit interrupt  
0 = Disables the AUSART transmit interrupt

bit 3 **SSPIE:** Synchronous Serial Port Interrupt Enable bit

1 = Enables the SSP interrupt  
0 = Disables the SSP interrupt

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

1 = Enables the CCP1 interrupt  
0 = Disables the CCP1 interrupt

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

1 = Enables the TMR2 to PR2 match interrupt  
0 = Disables the TMR2 to PR2 match interrupt

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

1 = Enables the TMR1 overflow interrupt  
0 = Disables the TMR1 overflow interrupt

#### Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

## 2.2.2.5 PIR1 Register

The PIR1 register contains the individual flag bits for the peripheral interrupts.

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

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

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

- bit 7 **PSPIF:** Parallel Slave Port Read/Write Interrupt Flag bit<sup>(1)</sup>  
 1 = A read or a write operation has taken place (must be cleared in software)  
 0 = No read or write has occurred  
**Note:** PSPIF is reserved on 28-pin devices; always maintain this bit clear.
- bit 6 **ADIF:** A/D Converter Interrupt Flag bit  
 1 = An A/D conversion is completed (must be cleared in software)  
 0 = The A/D conversion is not complete
- bit 5 **RCIF:** AUSART Receive Interrupt Flag bit  
 1 = The AUSART receive buffer is full  
 0 = The AUSART receive buffer is empty
- bit 4 **TXIF:** AUSART Transmit Interrupt Flag bit  
 1 = The AUSART transmit buffer is empty  
 0 = The AUSART transmit buffer is full
- bit 3 **SSPIF:** Synchronous Serial Port (SSP) Interrupt Flag bit  
 1 = The SSP interrupt condition has occurred and must be cleared in software before returning from the Interrupt Service Routine. The conditions that will set this bit are:  
SPI:  
 A transmission/reception has taken place.  
I<sup>2</sup>C Slave:  
 A transmission/reception has taken place.  
I<sup>2</sup>C Master:  
 A transmission/reception has taken place. The initiated Start condition was completed by the SSP module. The initiated Stop condition was completed by the SSP module. The initiated Restart condition was completed by the SSP module. The initiated Acknowledge condition was completed by the SSP module. A Start condition occurred while the SSP module was Idle (multi-master system). A Stop condition occurred while the SSP module was Idle (multi-master system).  
 0 = No SSP interrupt condition has occurred
- bit 2 **CCP1IF:** CCP1 Interrupt Flag bit  
Capture mode:  
 1 = A TMR1 register capture occurred (must be cleared in software)  
 0 = No TMR1 register capture occurred  
Compare mode:  
 1 = A TMR1 register compare match occurred (must be cleared in software)  
 0 = No TMR1 register compare match occurred  
PWM mode:  
 Unused in this mode.
- bit 1 **TMR2IF:** TMR2 to PR2 Match Interrupt Flag bit  
 1 = TMR2 to PR2 match occurred (must be cleared in software)  
 0 = No TMR2 to PR2 match occurred
- bit 0 **TMR1IF:** TMR1 Overflow Interrupt Flag bit  
 1 = TMR1 register overflowed (must be cleared in software)  
 0 = TMR1 register did not overflow

**Legend:**

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

# PIC16F7X7

## 2.2.2.6 PIE2 Register

The PIE2 register contains the individual enable bits for the CCP2 and CCP3 peripheral interrupts.

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

R/W-0	R/W-0	R/W-0	U-0	R/W-0	U-0	R/W-0	R/W-0
OSFIE	CMIE	LVDIE	—	BCLIE	—	CCP3IE	CCP2IE
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 **LVDIE:** Low-Voltage Detect Interrupt Enable bit  
1 = LVD interrupt is enabled  
0 = LVD interrupt is disabled
- bit 4 **Unimplemented:** Read as '0'
- bit 3 **BCLIE:** Bus Collision Interrupt Enable bit  
1 = Enable bus collision interrupt in the SSP when configured for I<sup>2</sup>C Master mode  
0 = Disable bus collision interrupt in the SSP when configured for I<sup>2</sup>C Master mode
- bit 2 **Unimplemented:** Read as '0'
- bit 1 **CCP3IE:** CCP3 Interrupt Enable bit  
1 = Enables the CCP3 interrupt  
0 = Disables the CCP3 interrupt
- bit 0 **CCP2IE:** CCP2 Interrupt Enable bit  
1 = Enables the CCP2 interrupt  
0 = Disables the CCP2 interrupt

#### Legend:

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

## 2.2.2.7 PIR2 Register

The PIR2 register contains the flag bits for the CCP2 interrupt.

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

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

R/W-0	R/W-0	R/W-0	U-0	R/W-0	U-0	R/W-0	R/W-0
OSFIF	CMIF	LVDIF	—	BCLIF	—	CCP3IF	CCP2IF
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 **LVDIF:** Low-Voltage Detect Interrupt Flag bit  
1 = The supply voltage has fallen below the specified LVD voltage (must be cleared in software)  
0 = The supply voltage is greater than the specified LVD voltage
- bit 4 **Unimplemented:** Read as '0'
- bit 3 **BCLIF:** Bus Collision Interrupt Flag bit  
1 = A bus collision has occurred in the SSP when configured for I<sup>2</sup>C Master mode  
0 = No bus collision has occurred
- bit 2 **Unimplemented:** Read as '0'
- bit 1 **CCP3IF:** CCP3 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 0 **CCP2IF:** CCP2 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.

**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

# PIC16F7X7

## 2.2.2.8 PCON Register

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

**Note:**  $\overline{\text{BOR}}$  is unknown on POR. It must be set by the user and checked on subsequent Resets to see if BOR is clear, indicating a brown-out has occurred. The BOR status bit is not predictable if the brown-out circuit is disabled (by clearing the BOREN bit in the Configuration Word register).

### REGISTER 2-8: PCON: POWER CONTROL/STATUS REGISTER (ADDRESS 8Eh)

	U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-1
	—	—	—	—	—	SBOREN	$\overline{\text{POR}}$	$\overline{\text{BOR}}$
bit 7								bit 0

bit 7-3 **Unimplemented:** Read as '0'

bit 2 **SBOREN:** Software Brown-out Reset Enable bit

If BORSEN in Configuration Word 2 is a '1' and BOREN in Configuration Word 1 is '0':

1 = BOR enabled

0 = BOR disabled

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

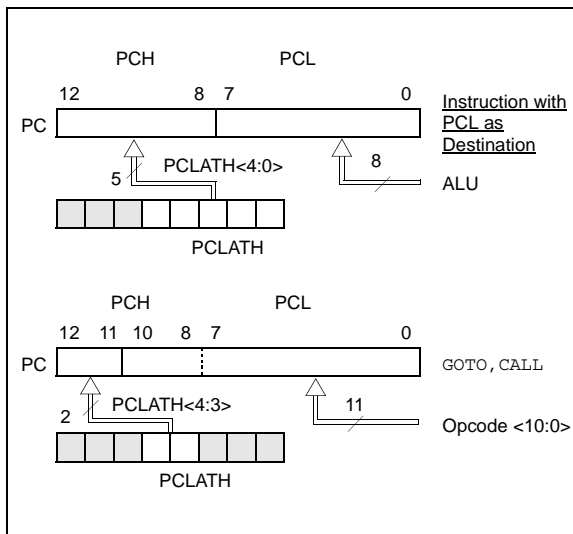
'0' = Bit is cleared

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" (DS00556).

### 2.3.2 STACK

The PIC16F7X7 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

PIC16F7X7 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 POPed 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 are unchanged after a RETURN or RETFIE instruction is executed. The user must set up the PCLATH for any subsequent CALLS or GOTOS.

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)
    
```

# PIC16F7X7

## 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.

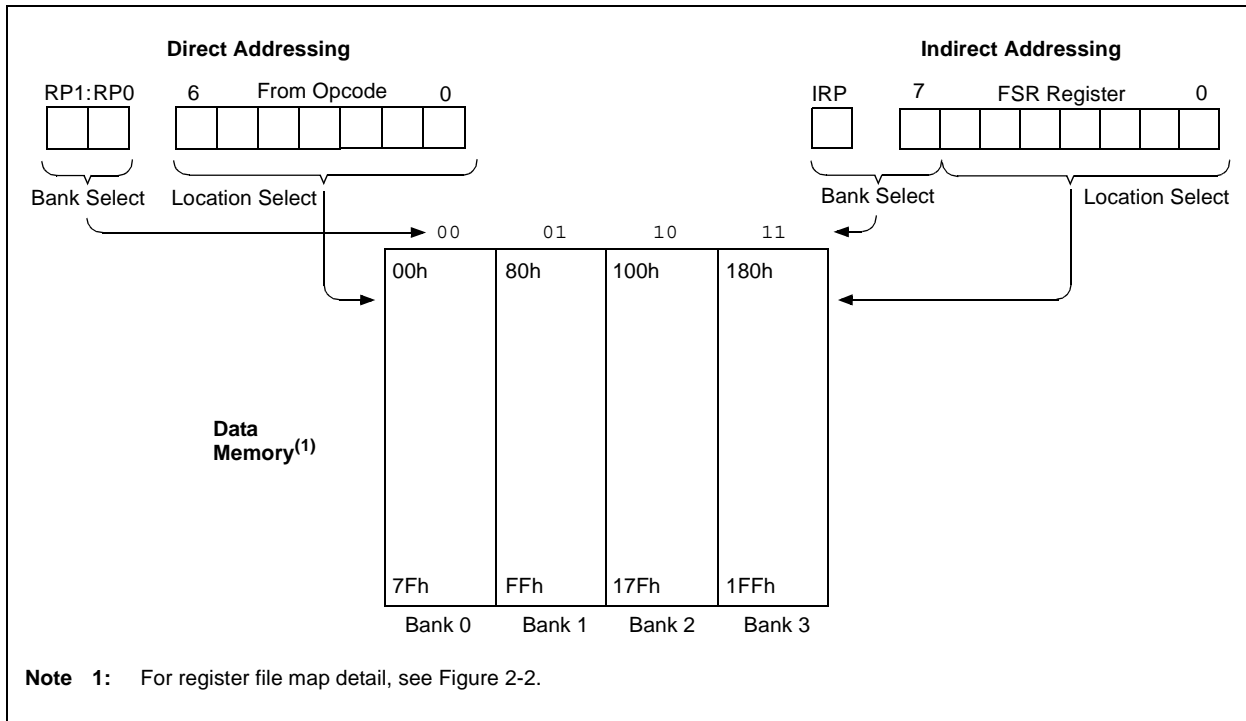
A simple program to clear RAM locations 20h-2Fh using indirect addressing is shown in Example 2-2.

### EXAMPLE 2-2: INDIRECT ADDRESSING

```

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

FIGURE 2-5: DIRECT/INDIRECT ADDRESSING





## 3.0 READING PROGRAM MEMORY

The Flash program memory is readable during normal operation over the entire VDD range. It is indirectly addressed through Special Function Registers (SFR). Up to 14-bit numbers can be stored in memory for use as calibration parameters, serial numbers, packed 7-bit ASCII, etc. Executing a program memory location containing data that forms an invalid instruction results in a NOP.

There are five SFRs used to read the program and memory. These registers are:

- PMCON1
- PMDATA
- PMDATH
- PMADR
- PMADRH

The program memory allows word reads. Program memory access allows for checksum calculation and reading calibration tables.

When interfacing to the program memory block, the PMDATH:PMDATA registers form a two-byte word which holds the 14-bit data for reads. The PMADRH:PMADR registers form a two-byte word which holds the 13-bit address of the Flash location being accessed. These devices can have up to 8K words of program Flash, with an address range from 0h to 3FFFh. The unused upper bits in both the PMDATH and PMADRH registers are not implemented and read as '0's.

### 3.1 PMADR

The address registers can address up to a maximum of 8K words of program Flash.

When selecting a program address value, the MSB of the address is written to the PMADRH register and the LSB is written to the PMADR register. The upper Most Significant bits of PMADRH must always be clear.

### 3.2 PMCON1 Register

PMCON1 is the control register for memory accesses.

The control bit, RD, initiates read operations. This bit cannot be cleared, only set, in software. It is cleared in hardware at the completion of the read operation.

#### REGISTER 3-1: PMCON1: PROGRAM MEMORY CONTROL REGISTER 1 (ADDRESS 18Ch)

R-1	U-0	U-0	U-0	U-x	U-0	U-0	R/S-0
reserved	—	—	—	—	—	—	RD
bit 7							bit 0

bit 7 **Reserved:** Read as '1'

bit 6-1 **Unimplemented:** Read as '0'

bit 0 **RD:** Read Control bit

1 = Initiates a Flash read, RD is cleared in hardware. The RD bit can only be set (not cleared) in software.

0 = Flash read completed

#### 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

# PIC16F7X7

## 3.3 Reading the Flash Program Memory

A program memory location may be read by writing two bytes of the address to the PMADR and PMADRH registers and then setting control bit, RD (PMCON1<0>). Once the read control bit is set, the microcontroller will use the next two instruction cycles to read the data. The data is available in the PMDATA and PMDATH registers after the second NOP instruction; therefore, it can be read as two bytes in the following instructions. The PMDATA and PMDATH registers will hold this value until the next read operation.

## 3.4 Operation During Code-Protect

Flash program memory has its own code-protect mechanism. External read and write operations by programmers are disabled if this mechanism is enabled.

The microcontroller can read and execute instructions out of the internal Flash program memory, regardless of the state of the code-protect configuration bits.

### EXAMPLE 3-1: FLASH PROGRAM READ

	BSF	STATUS, RP1	;
	BCF	STATUS, RP0	; Bank 2
	MOVF	ADDRH, W	;
	MOVWF	PMADRH	; MSByte of Program Address to read
	MOVF	ADDRL, W	;
	MOVWF	PMADR	; LSByte of Program Address to read
	BSF	STATUS, RP0	; Bank 3 Required
<b>Required Sequence</b>	BSF	PMCON1, RD	; EEPROM Read Sequence
	NOP		; memory is read in the next two cycles after BSF PMCON1,RD
	NOP		;
	BCF	STATUS, RP0	; Bank 2
	MOVF	PMDATA, W	; W = LSByte of Program PMDATA
	MOVF	PMDATH, W	; W = MSByte of Program PMDATH

TABLE 3-1: REGISTERS ASSOCIATED WITH PROGRAM FLASH

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
10Dh	PMADR	EEPROM Address Register Low Byte								xxxx xxxx	uuuu uuuu
10Fh	PMADRH	—	—	—	—	EEPROM Address Register High Byte				---- xxxx	---u uuuu
10Ch	PMDATA	EEPROM Data Register Low Byte								xxxx xxxx	uuuu uuuu
10Eh	PMDATH	—	—	EEPROM Data Register High Byte						--xx xxxx	--uu uuuu
18Ch	PMCON1	reserved <sup>(1)</sup>	—	—	—	—	—	—	RD	1--- ---0	1--- ---0

**Legend:** x = unknown, u = unchanged, — = unimplemented, read as '0'. Shaded cells are not used during Flash access.

**Note 1:** This bit always reads as a '1'.

## 4.0 OSCILLATOR CONFIGURATIONS

### 4.1 Oscillator Types

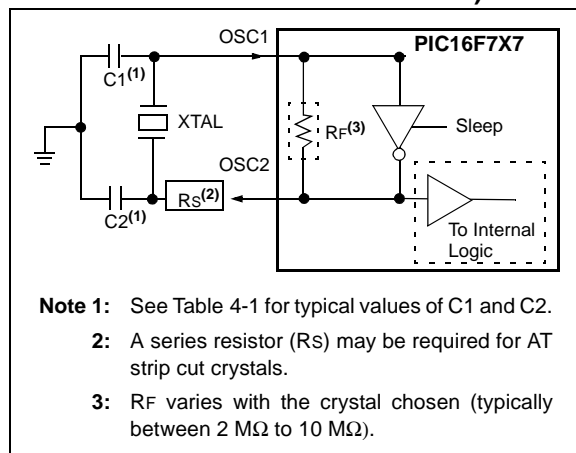
The PIC16F7X7 can be operated in eight different oscillator modes. The user can program three configuration bits (FOSC2:FOSC0) to select one of these eight modes (modes 5-8 are new PIC16 oscillator configurations):

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 PIC16F7X7 oscillator design requires the use of a parallel cut crystal. Use of a series cut crystal may give a frequency out of the crystal manufacturer's specifications.

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



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

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

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

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

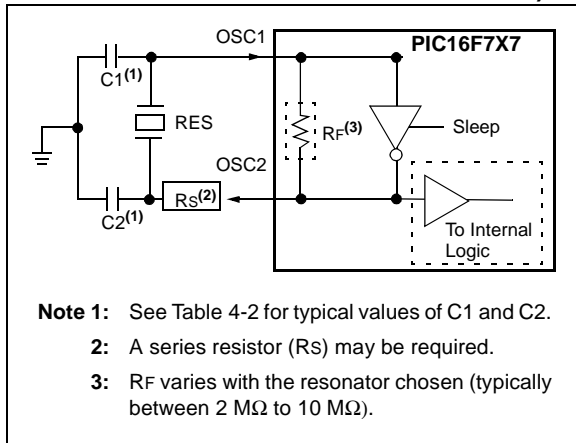
Different capacitor values may be required to produce acceptable oscillator operation. The user should test the performance of the oscillator over the expected VDD and temperature range for the application.

See the notes following this table for additional information.

- Note 1:** Higher capacitance increases the stability of oscillator but also increases the start-up time.
- 2:** Since each crystal has its own characteristics, the user should consult the crystal manufacturer for appropriate values of external components.
- 3:** Rs may be required in HS mode, as well as XT mode, to avoid overdriving crystals with low drive level specification.
- 4:** Always verify oscillator performance over the VDD and temperature range that is expected for the application.

# PIC16F7X7

**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.

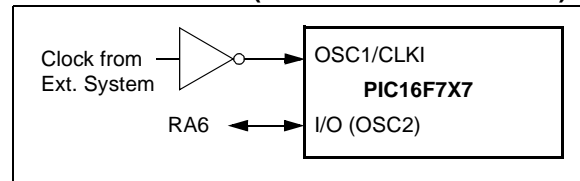
**Note:** When using resonators with frequencies above 3.5 MHz, the use of HS mode rather than XT mode is recommended. HS mode may be used at any VDD for which the controller is rated. If HS is selected, it is possible that the gain of the oscillator will overdrive the resonator. Therefore, a series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

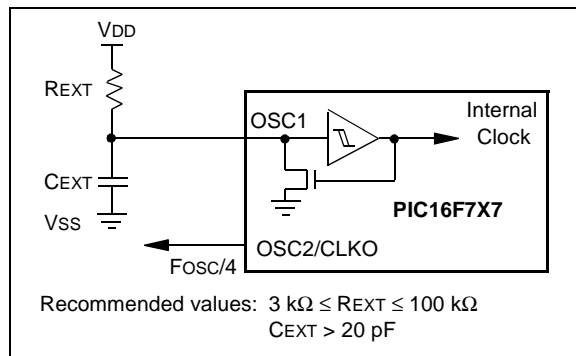


## 4.4 RC Oscillator

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

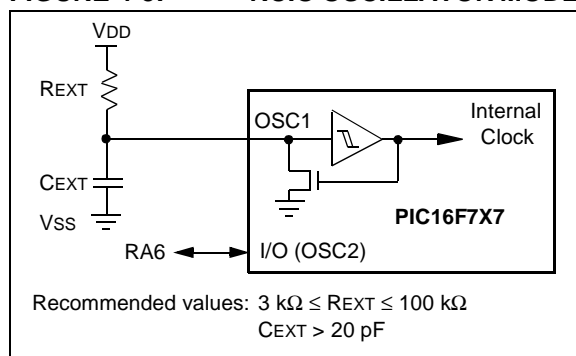
In the RC Oscillator mode, the oscillator frequency divided by 4 is available on the OSC2 pin. This signal may be used for test purposes or to synchronize other logic.

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



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

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



## 4.5 Internal Oscillator Block

The PIC16F7X7 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  $\mu$ s 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 38).

**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).

# PIC16F7X7

## 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 PIC16F7X7 devices include a feature that allows the system clock source to be switched from the main oscillator to an alternate low-frequency clock source. PIC16F7X7 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.

PIC16F7X7 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 RC0/T1OSO/T1CKI and RC1/T1OSI/CCP2 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 PIC16F7X7 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 Register 1. 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'.

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 Timer1 oscillator (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 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 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).

# PIC16F7X7

## 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 (OST).

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-0	R/W-0	R/W-0
—	IRCF2	IRCF1	IRCF0	OSTS <sup>(1)</sup>	IOFS	SCS1	SCS0
bit 7							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 the Timer1 oscillator (T1OSC) or INTRC as a secondary system clock

**Note 1:** Bit resets to '0' with Two-Speed Start-up 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

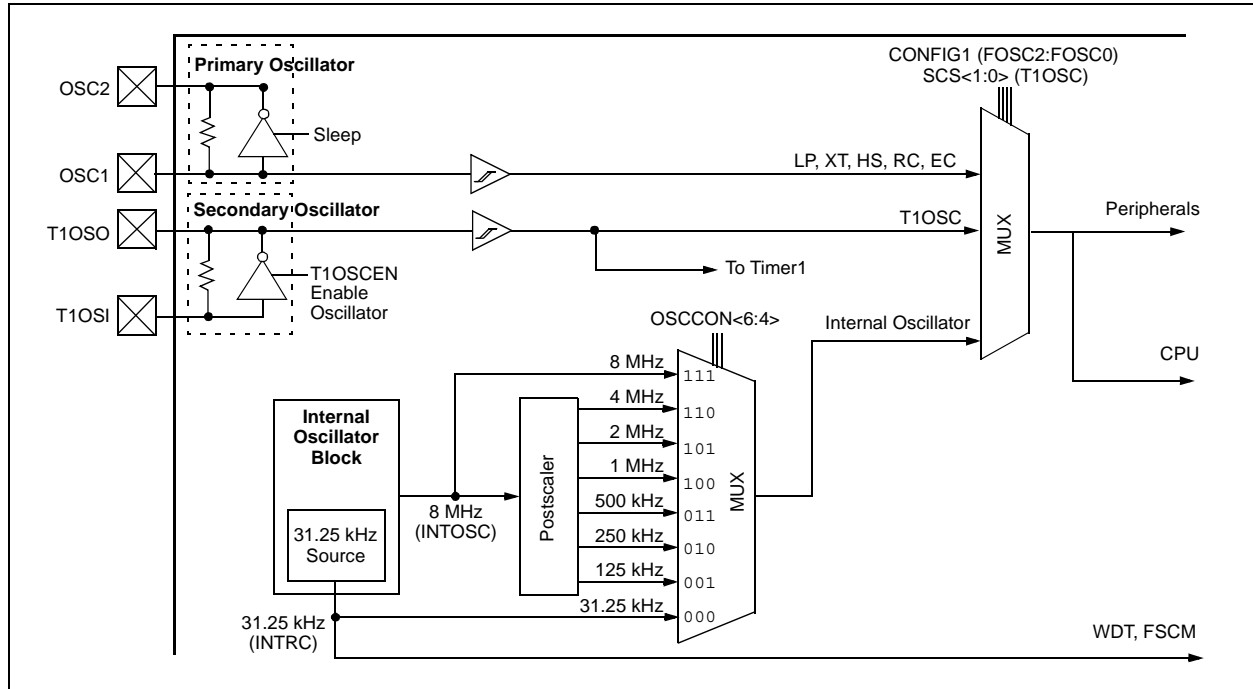
- 00 = Oscillator mode defined by FOSC<2:0>
- 01 = T1OSC is used for system clock
- 10 = Internal RC is used for system clock
- 11 = Reserved

#### 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: PIC16F7X7 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).

# PIC16F7X7

## 4.6.5 CLOCK TRANSITION SEQUENCE

The following are three different sequences for switching the internal RC oscillator frequency:

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

- Clock before switch: One of INTOSC/INTOSC postscaler (IRCF<2:0> ≠ 000)
  1. IRCF bits are modified to a different INTOSC/INTOSC postscaler frequency.
  2. The clock switching circuitry waits for a falling edge of the current clock, at which point CLKO is held low.
  3. The clock switching circuitry then waits for **eight** falling edges of requested clock, after which it switches CLKO to this new clock source.
  4. The IOFS bit is set.
  5. Oscillator switchover is complete.

## 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	Following a change from INTRC, the OST count 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 μs-10 μs start-up delay is based on a 1 MHz system clock.

## 4.7 Power-Managed Modes

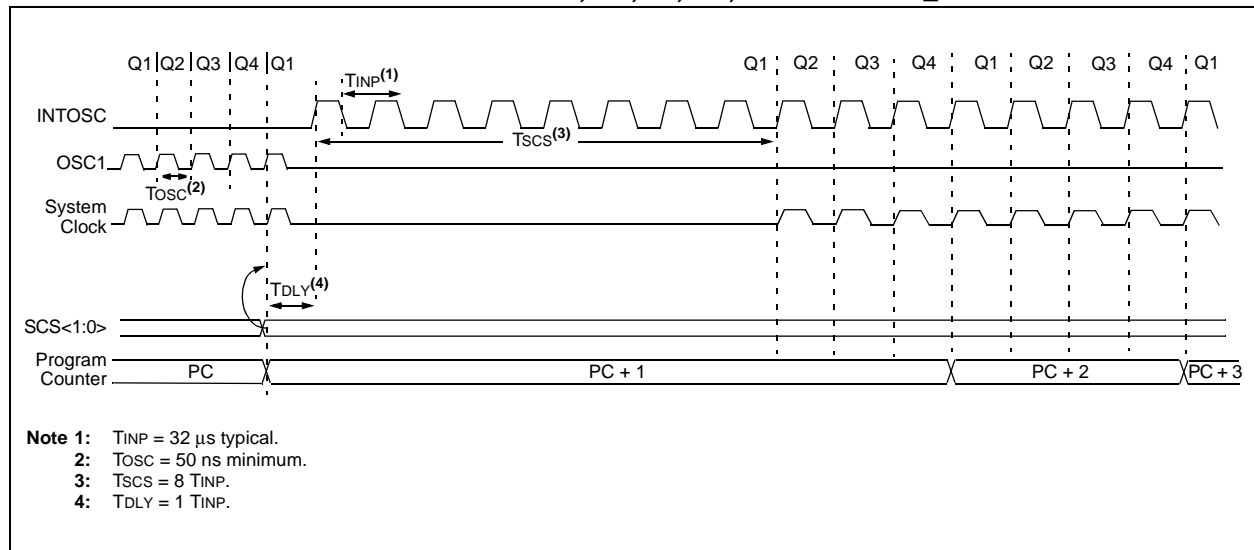
### 4.7.1 RC\_RUN MODE

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 and 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, EXTRC TO RC\_RUN MODE**



# PIC16F7X7

## 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.

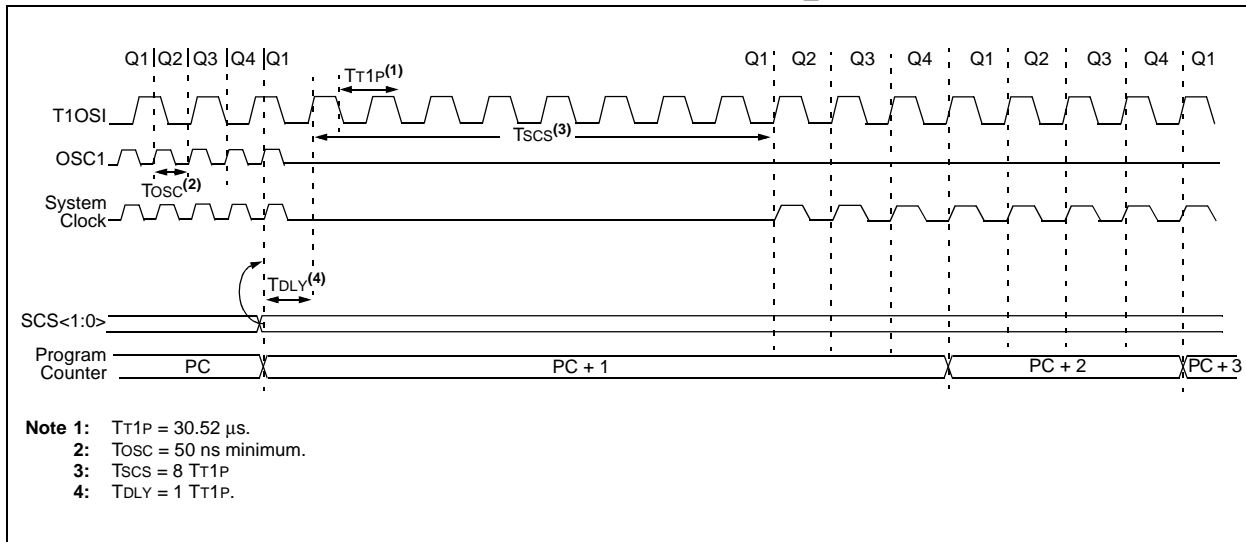
**Note 1:** The T1OSCN bit must be enabled and it is the user’s responsibility to ensure T1OSC is stable before clock switching to the T1OSC input clock can occur.

**2:** When T1OSCN = 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 take 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 counts have 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 Timer1 oscillator will be shut-down.

<b>Note:</b> 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

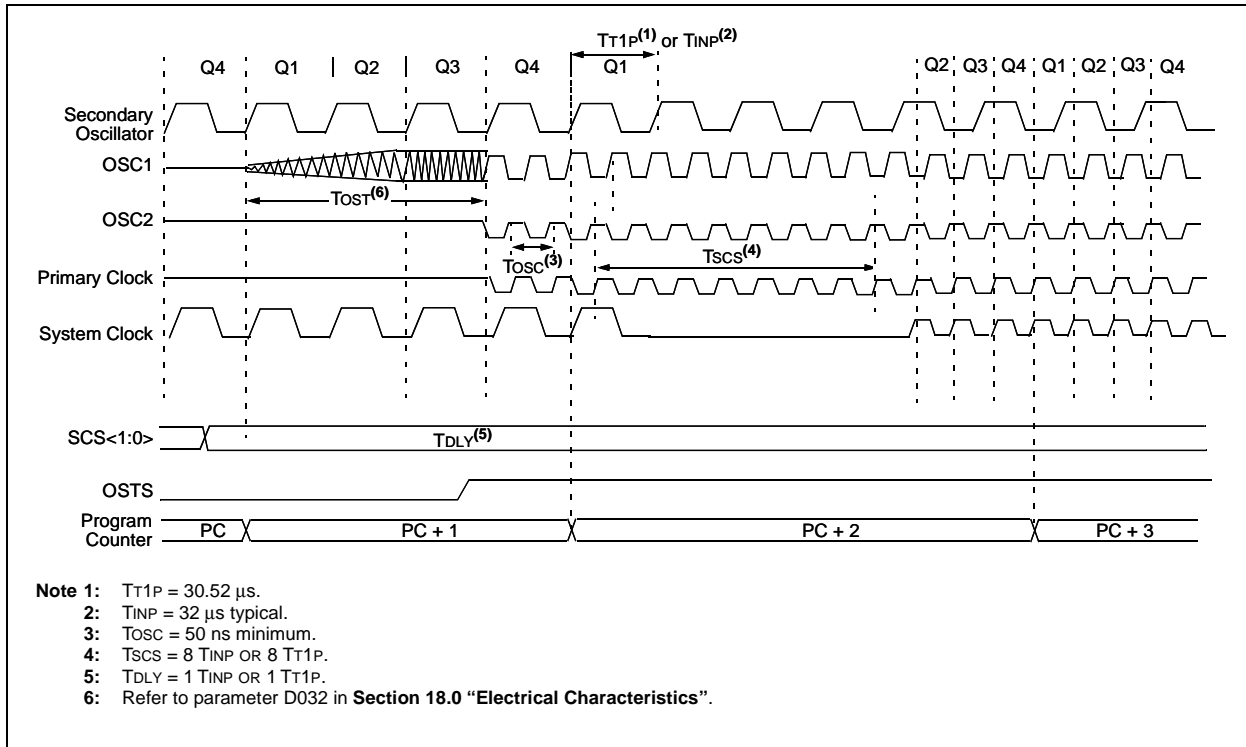
Changing back to the primary oscillator from SEC\_RUN or RC\_RUN can be accomplished by either changing SCS<1:0> to '00' or clearing the T1OSCEN bit in the T1CON register (if T1OSC was the secondary clock).

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 clock 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 Timer1 oscillator will be shut-down.

# PIC16F7X7

**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 take 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 had 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 Oscillator Start-up Timer has timed out.

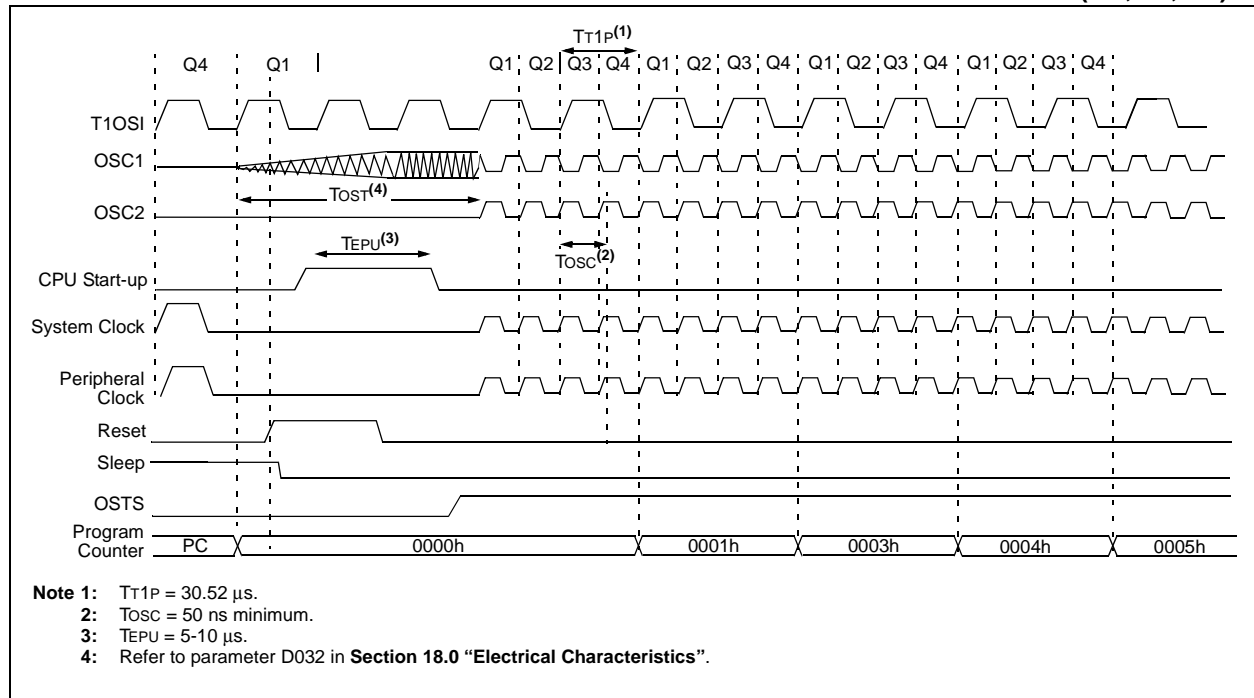
If the primary system clock is either RC, EC or INTRC, the CPU will begin operating on the first Q1 cycle following the wake-up event. This means that there is

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:

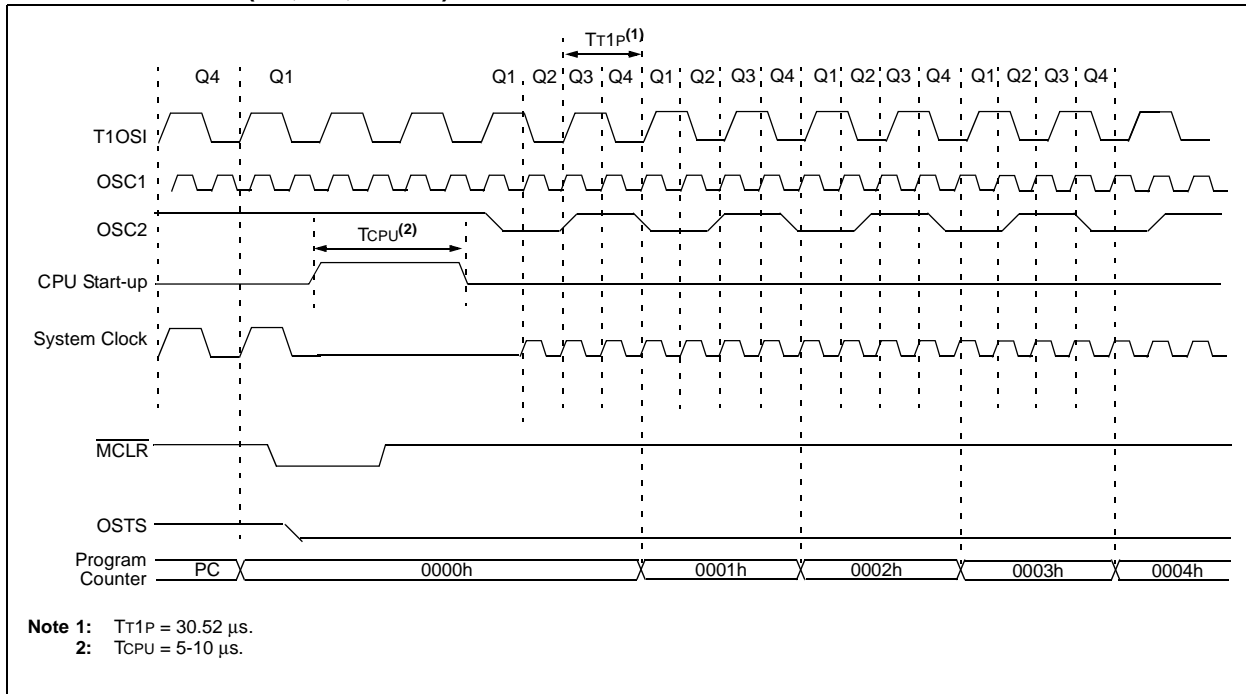
1. A device Reset is asserted from one of many sources (WDT, BOR, MCLR, etc.).
2. The device resets and the CPU start-up timer is enabled if in Sleep mode. The device is held in Reset until the CPU start-up time-out is complete.
3. If the primary system clock is configured as an external oscillator (HS, 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 timer and the Oscillator Start-up Timer have timed out, the device will wait for one additional clock cycle and instruction execution will begin.

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



# PIC16F7X7

**FIGURE 4-11: TIMING LP CLOCK TO 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 + 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	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 Oscillator Start-up 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.

# PIC16F7X7

---

## 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 15.17.3 “Two-Speed Clock Start-up Mode”** for details on Two-Speed Start-up.
3. After both the CPU start-up timer and the Oscillator Start-up Timer 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`, `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 a 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).

Reading the PORTA register reads the status of the pins, whereas writing to it, will write to the port latch.

The RA4 pin is multiplexed with the Timer0 module clock input and one of the comparator outputs to become the RA4/T0CKI/C1OUT pin. Pins RA6 and RA7 are multiplexed with the main oscillator pins; they are enabled as oscillator or I/O pins by the selection of the main oscillator in Configuration Register 1H (see **Section 15.1 "Configuration Bits"** for details). When they are not used as port pins, RA6 and RA7 and their associated TRIS and LAT bits are read as '0'.

The other PORTA pins are multiplexed with analog inputs, the analog VREF+ and VREF- inputs and the comparator voltage reference output. The operation of pins RA3:RA0 and RA5 as A/D converter inputs is selected by clearing/setting the control bits in the ADCON1 register (A/D Control Register 1). Pins RA0 through RA5 may also be used as comparator inputs or outputs by setting the appropriate bits in the CMCON register.

**Note:** On a Power-on Reset, RA5 and RA3:RA0 are configured as analog inputs and read as '0'. RA4 is configured as a digital input.

The RA4/T0CKI/C1OUT pin is a Schmitt Trigger input and an open-drain output. All other PORTA pins have TTL input levels and full CMOS output drivers.

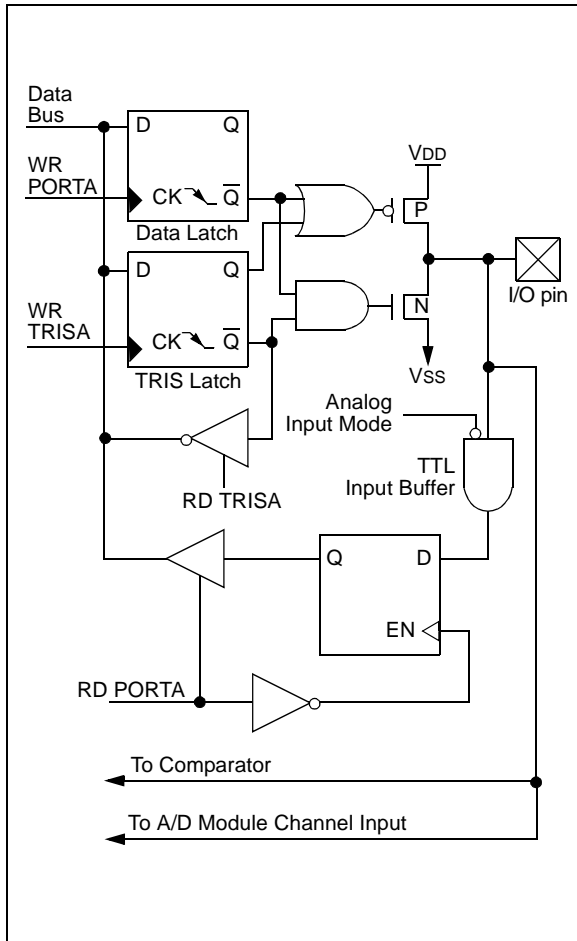
The TRISA register controls the direction of the RA pins even when they are being used as analog inputs. The user must ensure the bits in the TRISA register are maintained set when using them as analog inputs.

#### EXAMPLE 5-1: INITIALIZING PORTA

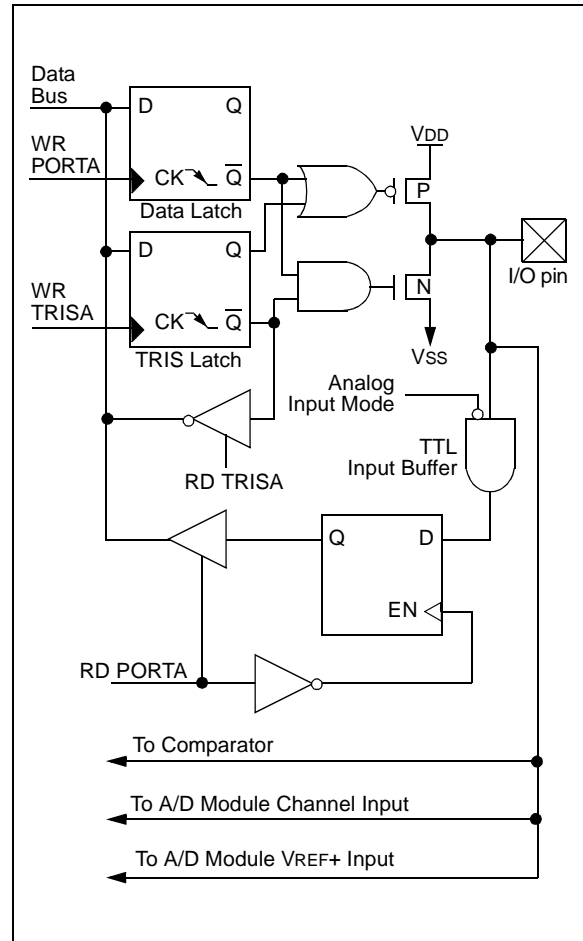
```
BCF     STATUS, RP0    ;
BCF     STATUS, RP1    ; Bank0
CLRF    PORTA         ; Initialize PORTA by
                        ; clearing output
                        ; data latches
BSF     STATUS, RP0    ; Select Bank 1
MOVLW   0x0F          ; Configure all pins
MOVWF   ADCON1        ; as digital inputs
MOVLW   0xCF          ; Value used to
                        ; initialize data
                        ; direction
MOVWF   TRISA         ; Set RA<3:0> as inputs
                        ; RA<5:4> as outputs
                        ; TRISA<7:6>are always
                        ; read as '0'.
```

# PIC16F7X7

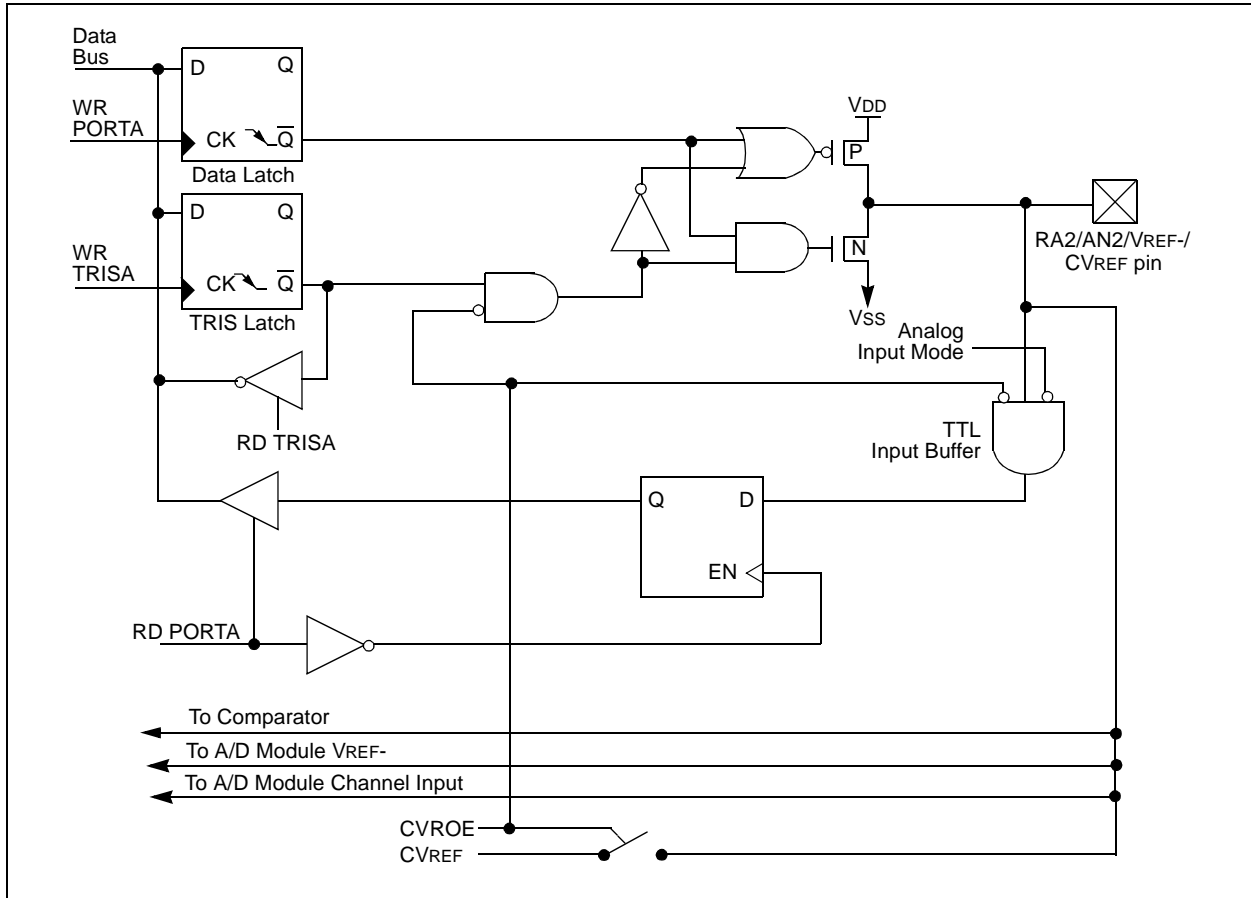
**FIGURE 5-1: BLOCK DIAGRAM OF RA0/AN0:RA1/AN1 PINS**



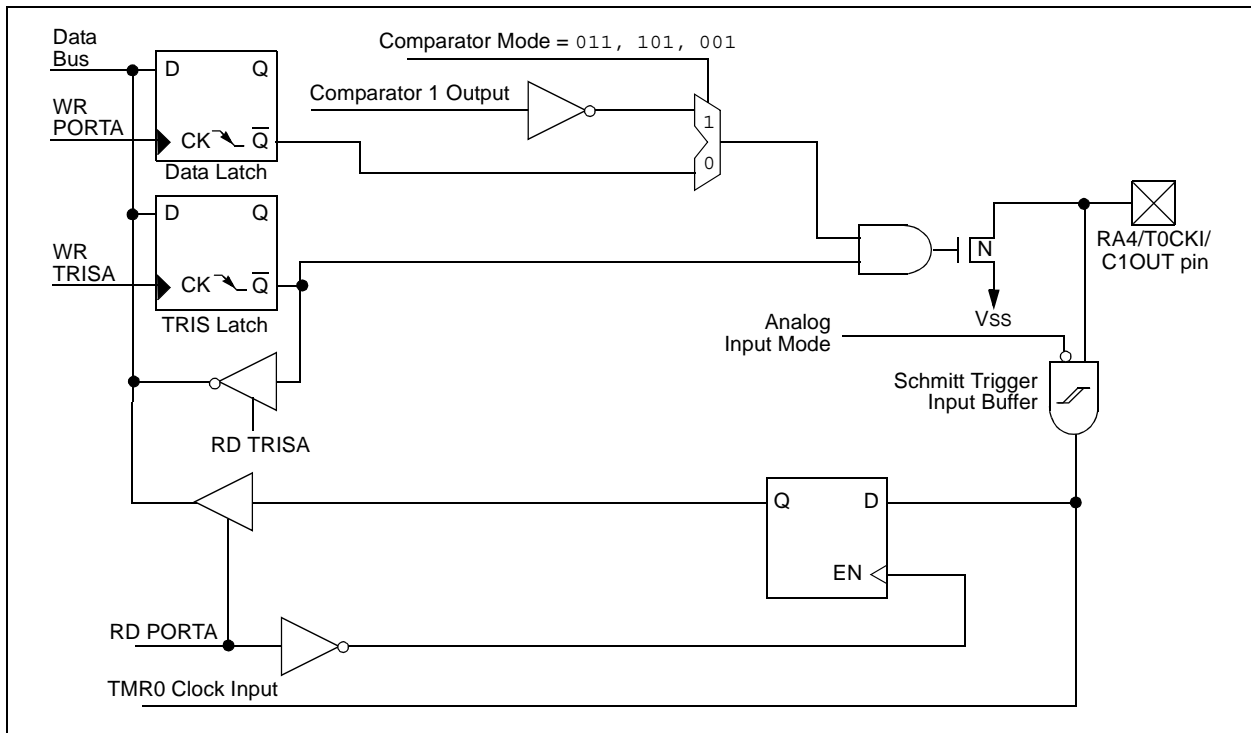
**FIGURE 5-2: BLOCK DIAGRAM OF RA3/AN3/VREF+ PIN**



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

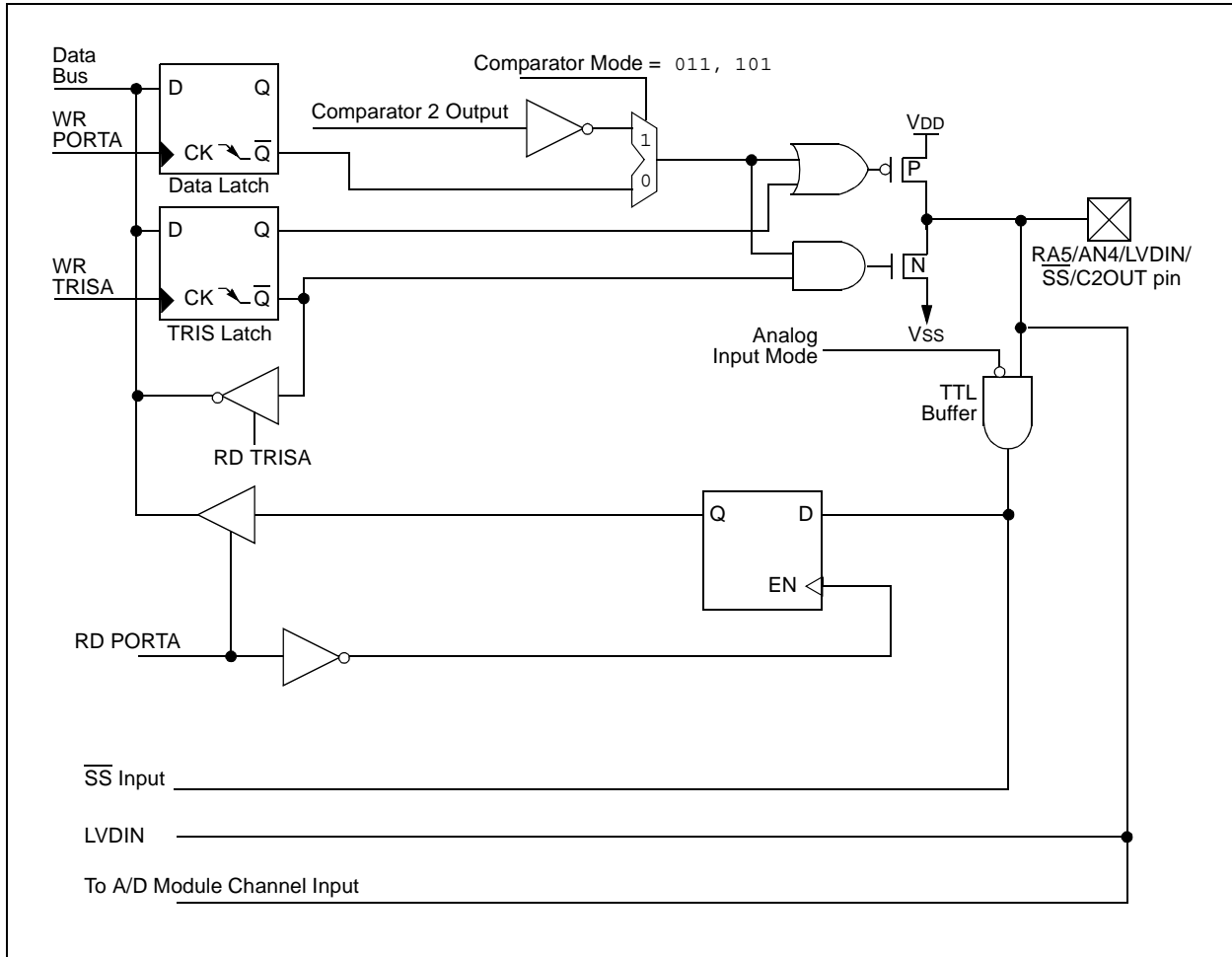


**FIGURE 5-4: BLOCK DIAGRAM OF RA4/T0CKI/C1OUT PIN**

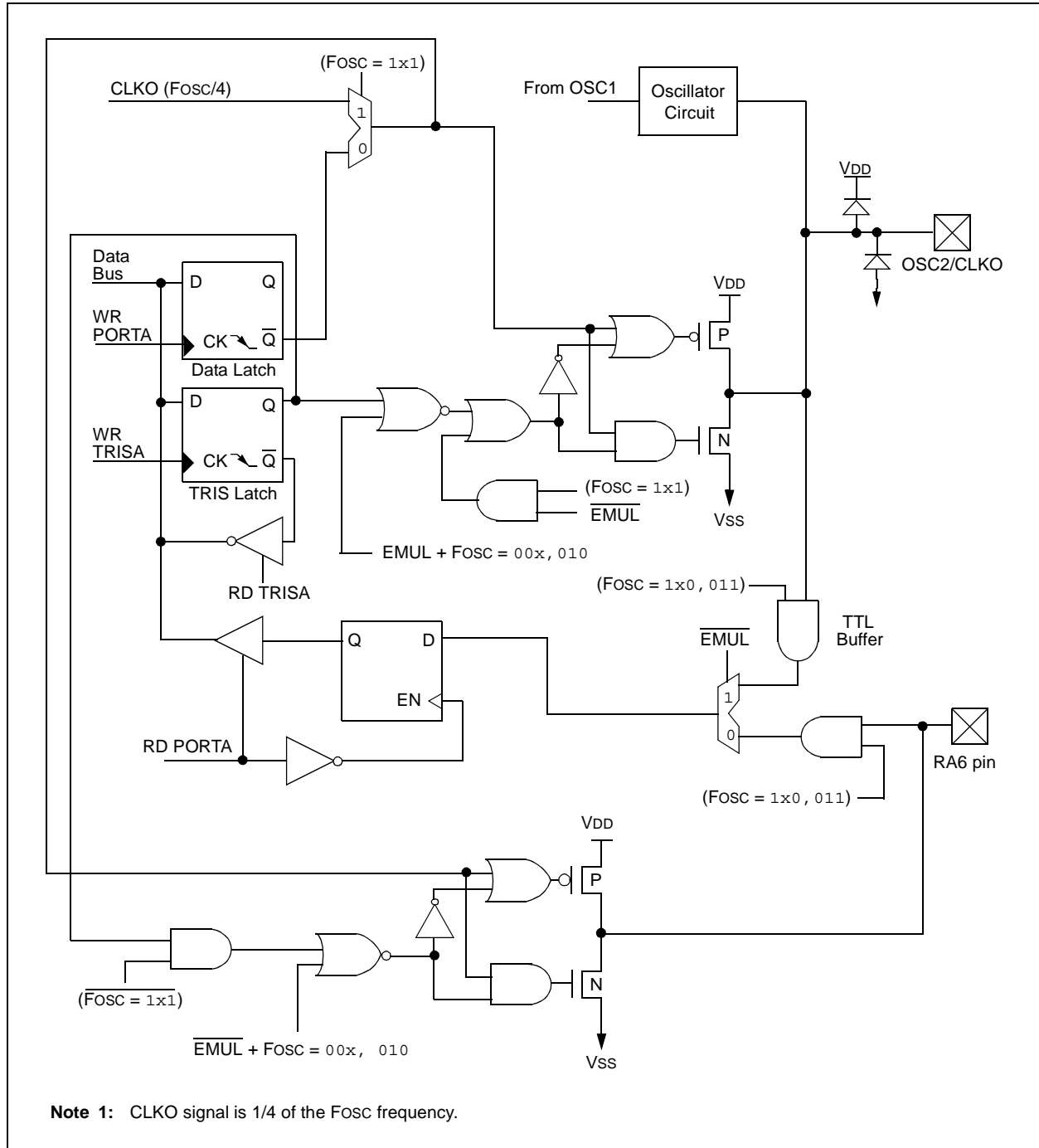


# PIC16F7X7

**FIGURE 5-5: BLOCK DIAGRAM OF RA5/AN4/LVDIN/SS/C2OUT PIN**

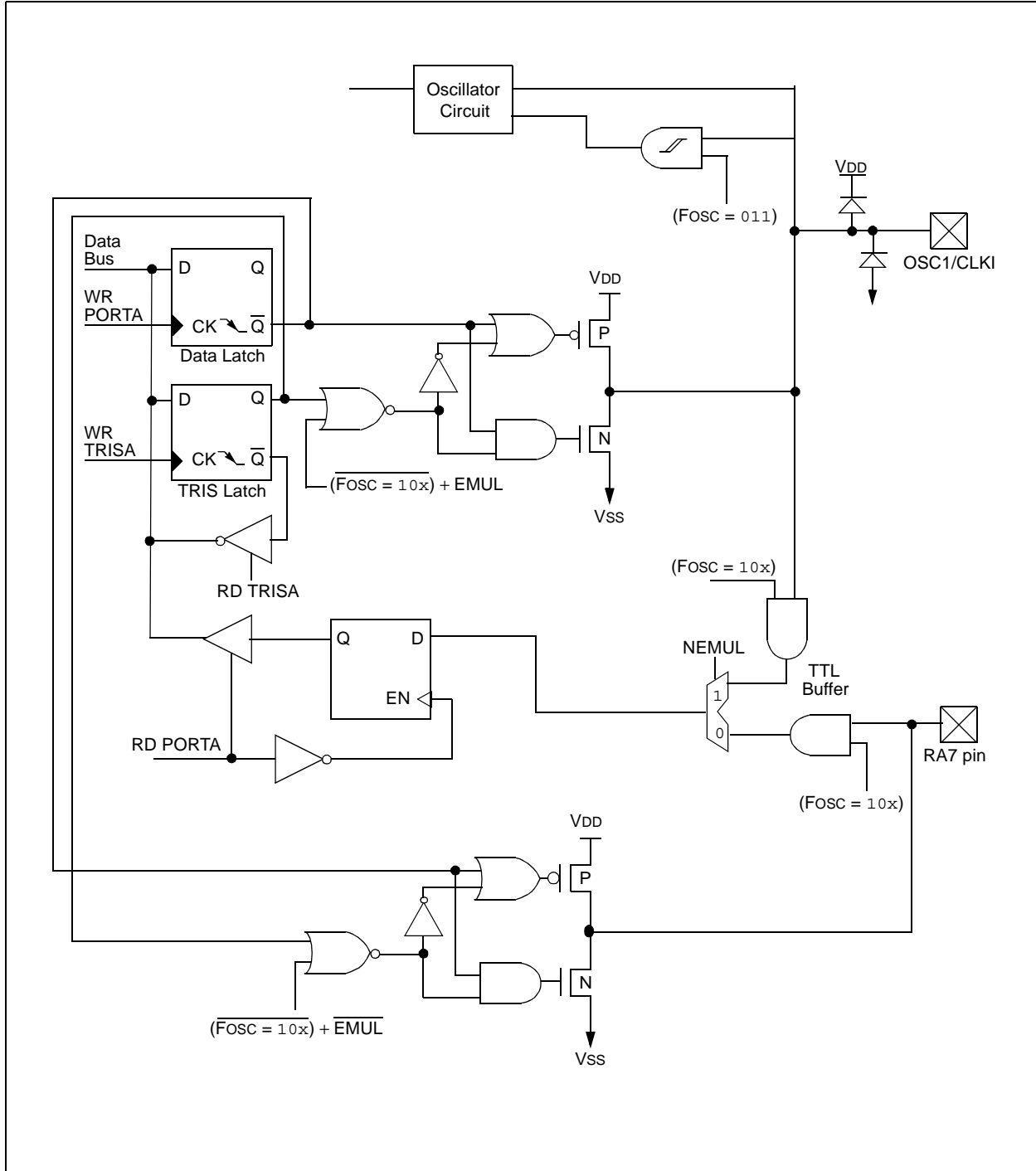


**FIGURE 5-6: BLOCK DIAGRAM OF OSC2/CLKO/RA6 PIN**



# PIC16F7X7

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





**TABLE 5-1: PORTA FUNCTIONS**

Name	Bit#	Buffer	Function
RA0/AN0	bit 0	TTL	Input/output or analog input.
RA1/AN1	bit 1	TTL	Input/output or analog input.
RA2/AN2/VREF-/CVREF	bit 2	TTL	Input/output or analog input or VREF-.
RA3/AN3/VREF+	bit 3	TTL	Input/output or analog input or VREF+.
RA4/T0CKI/C1OUT	bit 4	ST	Input/output or external clock input for Timer0. Output is open-drain type.
RA5/AN4/LVDIN/ $\overline{SS}$ /C2OUT	bit 5	TTL	Input/output or slave select input for synchronous serial port or analog input.
OSC2/CLKO/RA6	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.
OSC1/CLKI/RA7	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.

**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	xx0x 0000	uu0u 0000
85h	TRISA	PORTA Data Direction Register								1111 1111	1111 1111
9Fh	ADCON1	ADFM	ADCS2	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0	0000 0000	0000 0000
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

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

**Note:** When using the SSP module in SPI Slave mode and  $\overline{SS}$  enabled, the A/D converter must be set to one of the following modes, where PCFG2:PCFG0 = 100, 101, 11x.

# PIC16F7X7

## 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.

PORTB pins are multiplexed with analog inputs. The operation of each pin is selected by clearing/setting the appropriate control bits in the ADCON1 register.

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

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

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

- a) Any read or write of PORTB. This will end the mismatch condition.
- b) Clear flag bit RBIF.

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

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

This interrupt on mismatch feature, together with software configurable pull-ups on these four pins, allow easy interface to a keypad and make it possible for wake-up on key depression. Refer to the Application Note AN552 "Implementing Wake-up on Key Stroke" (DS00552).

RB0/INT is an external interrupt input pin and is configured using the INTEDG bit (OPTION\_REG<6>). RB0/INT is discussed in detail in **Section 15.15.1 "INT Interrupt"**.

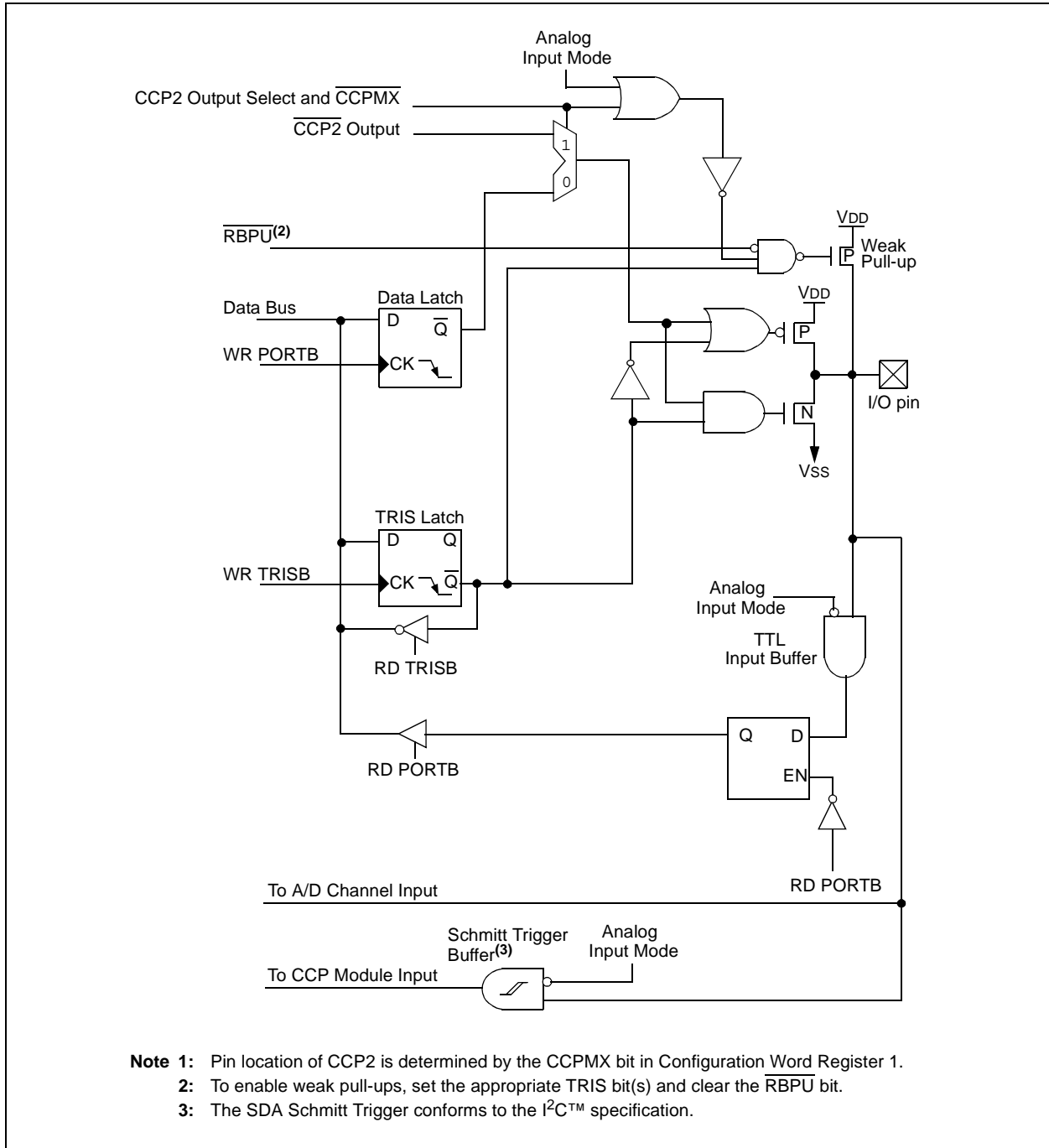
PORTB is multiplexed with several peripheral functions (see Table 5-3). PORTB pins have Schmitt Trigger input buffers.

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 destination should be avoided. The user should refer to the corresponding peripheral section for the correct TRIS bit settings.



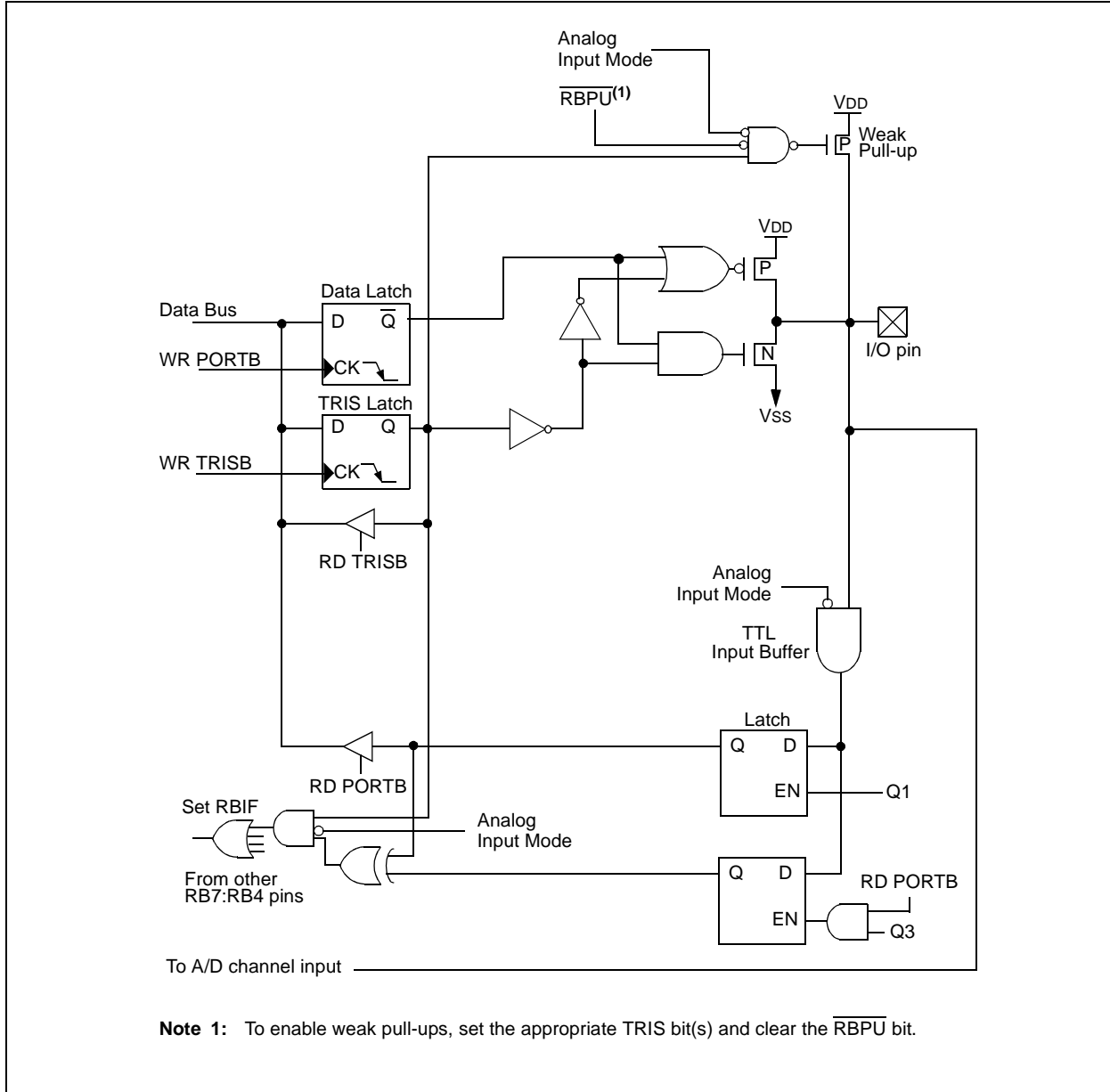


**FIGURE 5-11: BLOCK DIAGRAM OF RB3/CCP2<sup>(1)</sup>/AN9 PIN**

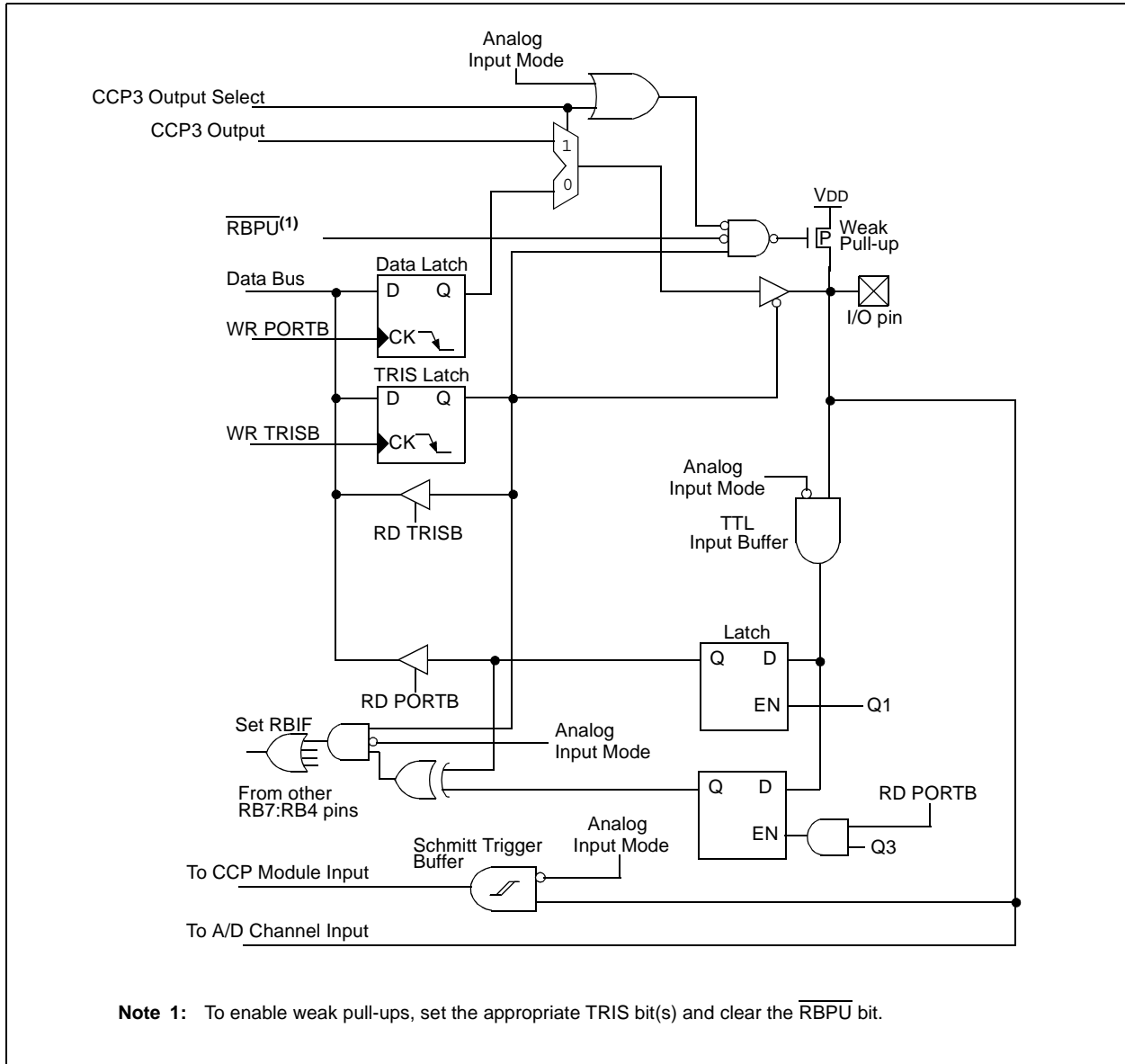


# PIC16F7X7

FIGURE 5-12: BLOCK DIAGRAM OF RB4/AN11 PIN

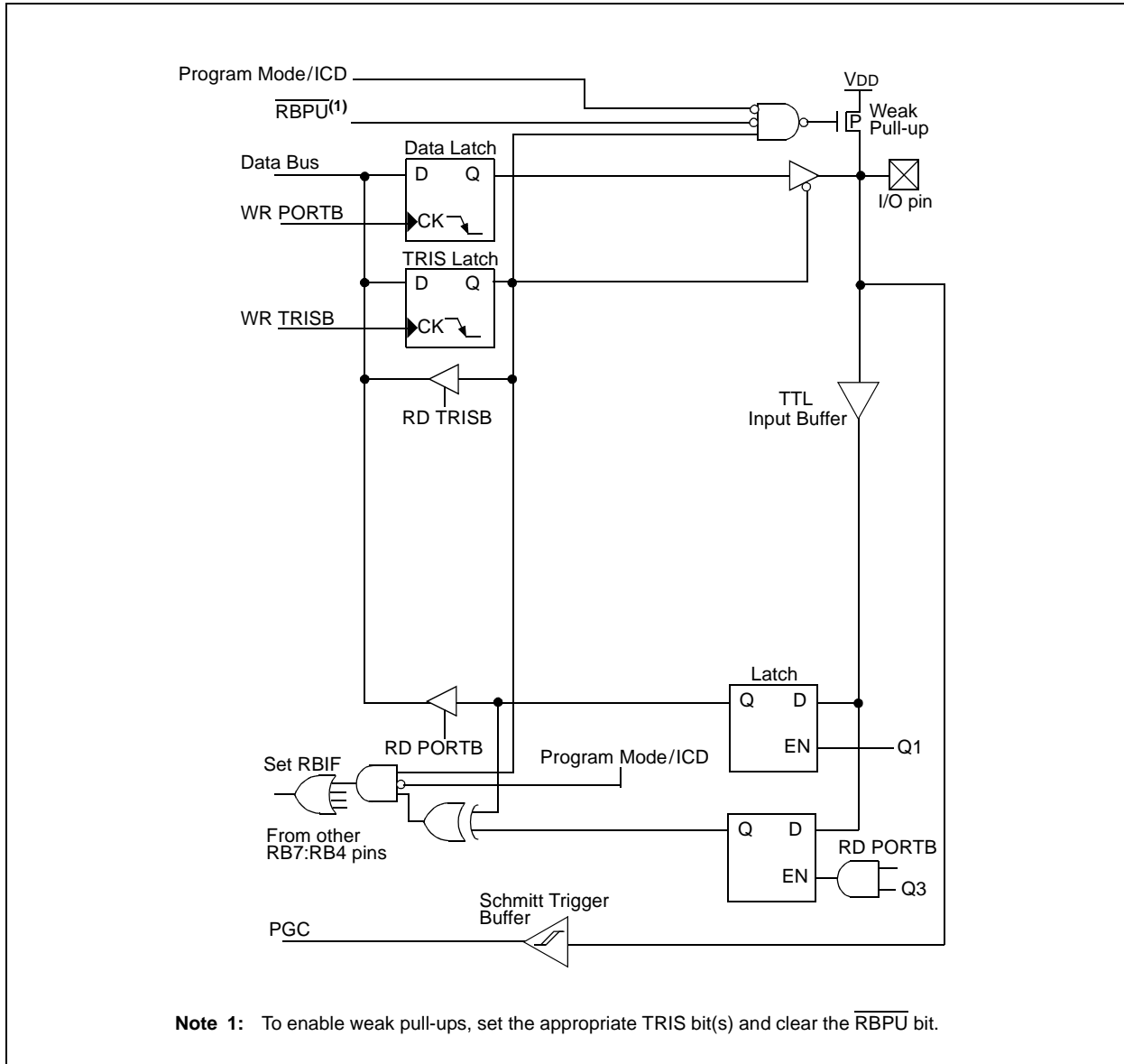


**FIGURE 5-13: BLOCK DIAGRAM OF RB5/AN13/CCP3 PIN**



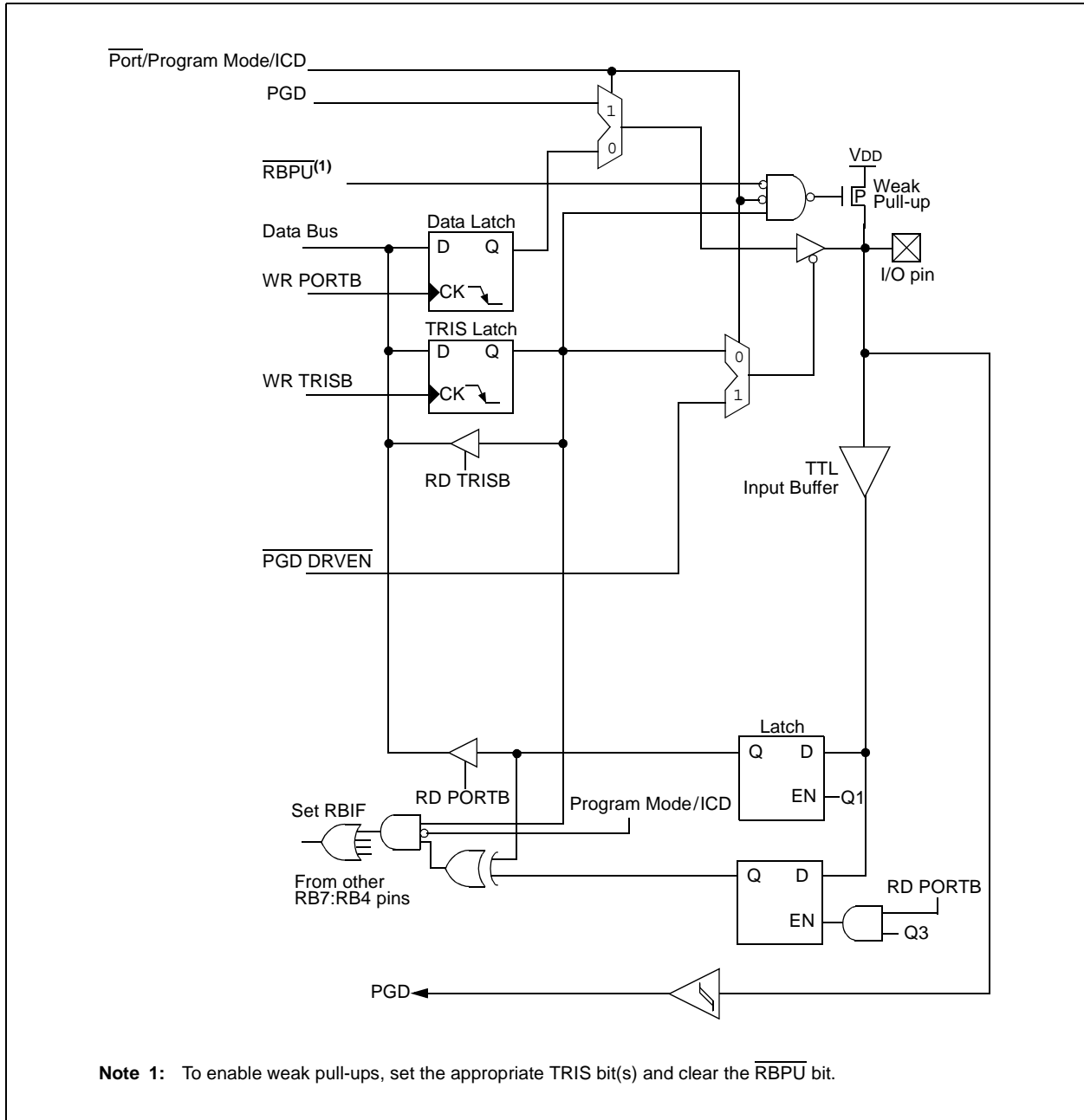
# PIC16F7X7

FIGURE 5-14: BLOCK DIAGRAM OF RB6/PGC PIN





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



# PIC16F7X7

**TABLE 5-3: PORTB FUNCTIONS**

Name	Bit#	Buffer	Function
RB0/INT/AN12	bit 0	TTL/ST <sup>(1)</sup>	Input/output pin or external interrupt input. Internal software programmable weak pull-up or analog input.
RB1/AN10	bit 1	TTL	Input/output pin. Internal software programmable weak pull-up or analog input.
RB2/AN8	bit 2	TTL	Input/output pin. Internal software programmable weak pull-up or analog input.
RB3/CCP2/AN9	bit 3	TTL	Input/output pin or Capture 2 input/Compare 2 output/PWM 2 output. Internal software programmable weak pull-up or analog input.
RB4/AN11	bit 4	TTL	Input/output pin (with interrupt-on-change). Internal software programmable weak pull-up or analog input.
RB5/AN13/CCP3	bit 5	TTL	Input/output pin (with interrupt-on-change). Internal software programmable weak pull-up or analog input or Capture 2 input/Compare 2 output/PWM 2 output.
RB6/PGC	bit 6	TTL/ST <sup>(2)</sup>	Input/output pin (with interrupt-on-change). Internal software programmable weak pull-up. Serial programming clock.
RB7/PGD	bit 7	TTL/ST <sup>(2)</sup>	Input/output pin (with interrupt-on-change). Internal software programmable weak pull-up. Serial programming data.

**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.

**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	xx00 0000	uu00 0000
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
9Fh	ADCON1	ADFM	ADCS2	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0	0000 0000	0000 0000

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

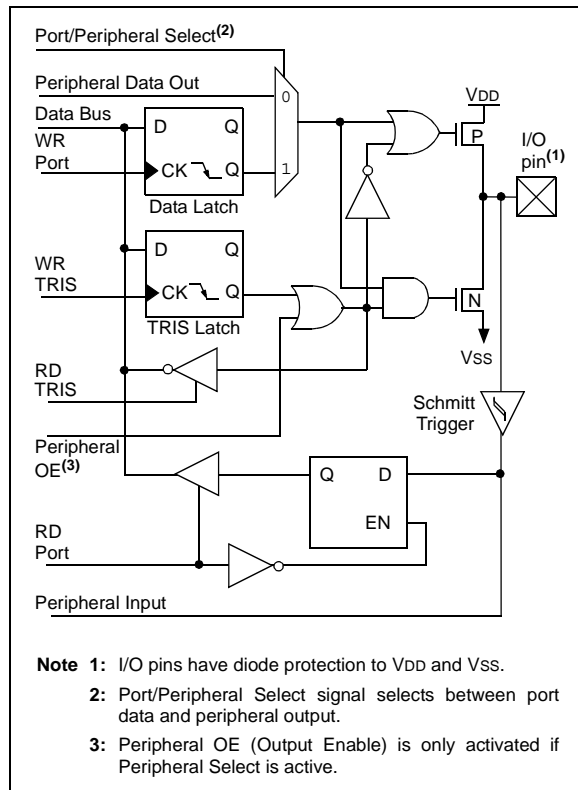
## 5.3 PORTC and the TRISC Register

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

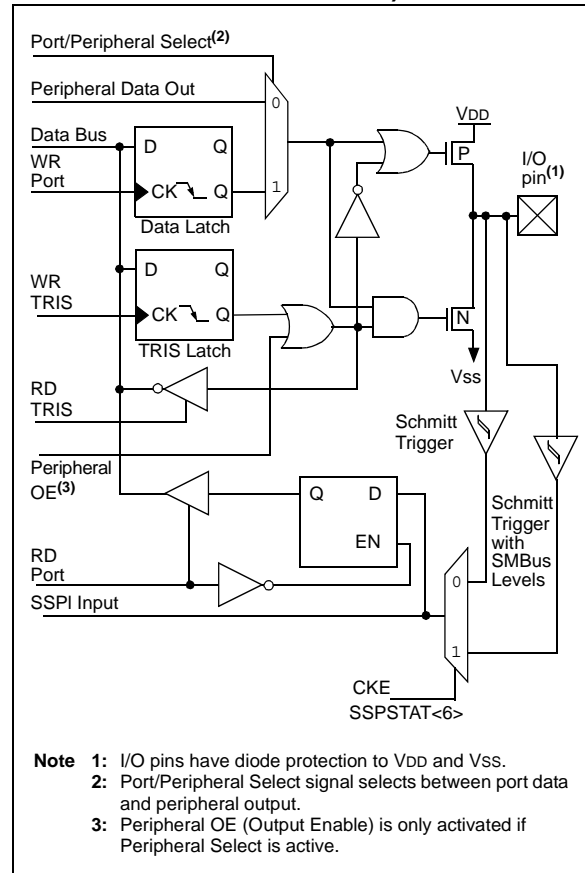
PORTC is multiplexed with several peripheral functions (Table 5-5). PORTC pins have Schmitt Trigger input buffers.

When enabling peripheral functions, care should be taken in defining TRIS bits for each PORTC 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 TRISC as destination should be avoided. The user should refer to the corresponding peripheral section for the correct TRIS bit settings and to **Section 16.1 “Read-Modify-Write Operations”** for additional information on read-modify-write operations.

**FIGURE 5-16: PORTC BLOCK DIAGRAM (PERIPHERAL OUTPUT OVERRIDE) RC<2:0>, RC<7:5> PINS**



**FIGURE 5-17: PORTC BLOCK DIAGRAM (PERIPHERAL OUTPUT OVERRIDE) RC<4:3> PINS**



# PIC16F7X7

**TABLE 5-5: PORTC FUNCTIONS**

Name	Bit#	Buffer Type	Function
RC0/T1OSO/T1CKI	bit 0	ST	Input/output port pin or Timer1 oscillator output/Timer1 clock input.
RC1/T1OSI/CCP2	bit 1	ST	Input/output port pin or Timer1 oscillator input or Capture 2 input/Compare 2 output/PWM 2 output.
RC2/CCP1	bit 2	ST	Input/output port pin or Capture 1 input/Compare 1 output/PWM 1 output.
RC3/SCK/SCL	bit 3	ST	RC3 can also be the synchronous serial clock for both SPI™ and I <sup>2</sup> C™ modes.
RC4/SDI/SDA	bit 4	ST	RC4 can also be the SPI data in (SPI mode) or data I/O (I <sup>2</sup> C mode).
RC5/SDO	bit 5	ST	Input/output port pin or Synchronous Serial Port data output.
RC6/TX/CK	bit 6	ST	Input/output port pin or AUSART asynchronous transmit or synchronous clock.
RC7/RX/DT	bit 7	ST	Input/output port pin or AUSART asynchronous receive or synchronous data.

**Legend:** ST = Schmitt Trigger input

**TABLE 5-6: SUMMARY OF REGISTERS ASSOCIATED WITH PORTC**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other Resets
07h	PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	xxxx xxxx	uuuu uuuu
87h	TRISC	PORTC Data Direction Register								1111 1111	1111 1111

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

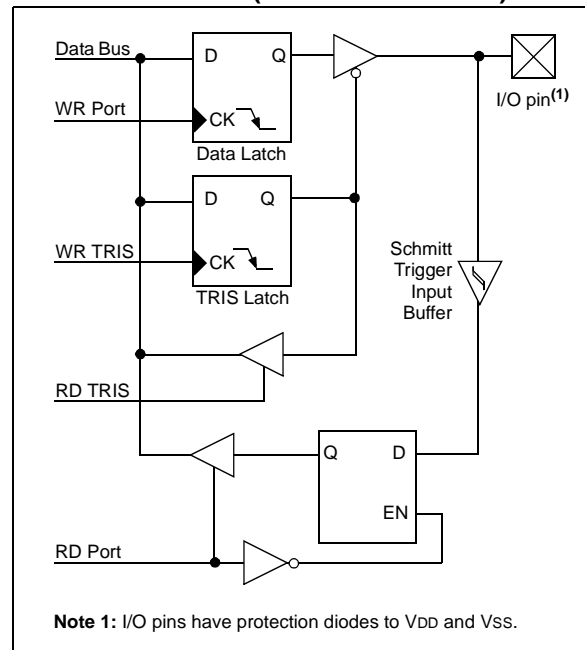
## 5.4 PORTD and TRISD Registers

This section is not applicable to the PIC16F737 or PIC16F767.

PORTD is an 8-bit port with Schmitt Trigger input buffers. Each pin is individually configurable as an input or output.

PORTD can be configured as an 8-bit wide micro-processor port (Parallel Slave Port) by setting control bit, PSPMODE (TRISE<4>). In this mode, the input buffers are TTL.

**FIGURE 5-18: PORTD BLOCK DIAGRAM (IN I/O PORT MODE)**



**TABLE 5-7: PORTD FUNCTIONS**

Name	Bit#	Buffer Type	Function
RD0/PSP0	bit 0	ST/TTL <sup>(1)</sup>	Input/output port pin or Parallel Slave Port bit 0.
RD1/PSP1	bit 1	ST/TTL <sup>(1)</sup>	Input/output port pin or Parallel Slave Port bit 1.
RD2/PSP2	bit 2	ST/TTL <sup>(1)</sup>	Input/output port pin or Parallel Slave Port bit 2.
RD3/PSP3	bit 3	ST/TTL <sup>(1)</sup>	Input/output port pin or Parallel Slave Port bit 3.
RD4/PSP4	bit 4	ST/TTL <sup>(1)</sup>	Input/output port pin or Parallel Slave Port bit 4.
RD5/PSP5	bit 5	ST/TTL <sup>(1)</sup>	Input/output port pin or Parallel Slave Port bit 5.
RD6/PSP6	bit 6	ST/TTL <sup>(1)</sup>	Input/output port pin or Parallel Slave Port bit 6.
RD7/PSP7	bit 7	ST/TTL <sup>(1)</sup>	Input/output port pin or Parallel Slave Port bit 7.

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

**Note 1:** Input buffers are Schmitt Triggers when in I/O mode and TTL buffers when in Parallel Slave Port mode.

**TABLE 5-8: SUMMARY OF REGISTERS ASSOCIATED WITH PORTD**

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
08h	PORTD	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	xxxx xxxx	uuuu uuuu
88h	TRISD	PORTD Data Direction Register								1111 1111	1111 1111
89h	TRISE	IBF	OBF	IBOV	PSPMODE	— <sup>(1)</sup>	PORTE Data Direction bits			0000 1111	0000 1111

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

**Note 1:** RE3 is an input only. The state of the TRISE3 bit has no effect and will always read '1'.

# PIC16F7X7

## 5.5 PORTE and TRISE Register

This section is not applicable to the PIC16F737 or PIC16F767.

PORTE has four pins, RE0/ $\overline{RD}$ /AN5, RE1/ $\overline{WR}$ /AN6, RE2/ $\overline{CS}$ /AN7 and MCLR/ $V_{PP}$ /RE3, which are individually configurable as inputs or outputs. These pins have Schmitt Trigger input buffers. RE3 is only available as an input if MCLRE is '0' in Configuration Word 1.

I/O PORTE becomes control inputs for the micro-processor port when bit, PSPMODE (TRISE<4>), is set. In this mode, the user must make sure that the TRISE<2:0> bits are set (pins are configured as digital inputs). Ensure ADCON1 is configured for digital I/O. In this mode, the input buffers are TTL.

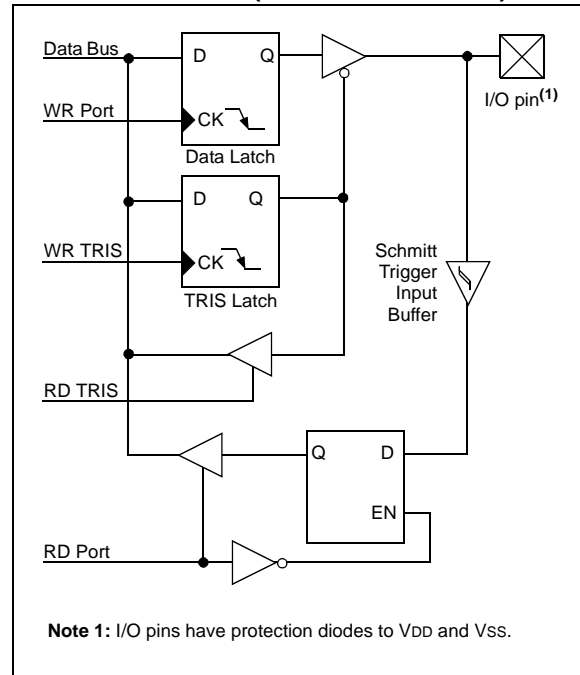
Register 5-1 shows the TRISE register which also controls the Parallel Slave Port operation.

PORTE pins are multiplexed with analog inputs. When selected as an analog input, these pins will read as '0's.

TRISE controls the direction of the RE pins, even when they are being used as analog inputs. The user must make sure to keep the pins configured as inputs when using them as analog inputs.

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

**FIGURE 5-19: PORTE BLOCK DIAGRAM (IN I/O PORT MODE)**



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

**TABLE 5-9: PORTE FUNCTIONS**

Name	Bit#	Buffer Type	Function
RE0/ $\overline{RD}$ /AN5	bit 0	ST/TTL <sup>(1)</sup>	Input/output port pin or read control input in Parallel Slave Port mode or analog input. For $\overline{RD}$ (PSP mode): 1 = Idle 0 = Read operation. Contents of PORTD register output to PORTD I/O pins (if chip selected).
RE1/ $\overline{WR}$ /AN6	bit 1	ST/TTL <sup>(1)</sup>	Input/output port pin or write control input in Parallel Slave Port mode or analog input. For $\overline{WR}$ (PSP mode): 1 = Idle 0 = Write operation. Value of PORTD I/O pins latched into PORTD register (if chip selected).
RE2/ $\overline{CS}$ /AN7	bit 2	ST/TTL <sup>(1)</sup>	Input/output port pin or chip select control input in Parallel Slave Port mode or analog input. For $\overline{CS}$ (PSP mode): 1 = Device is not selected 0 = Device is selected
MCLR/ $V_{PP}$ /RE3	bit 3	ST	Input, Master Clear (Reset) or programming input voltage.

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

**Note 1:** Input buffers are Schmitt Triggers when in I/O mode and TTL buffers when in Parallel Slave Port mode.

**TABLE 5-10: SUMMARY OF REGISTERS ASSOCIATED WITH PORTE**

Addr	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
09h	PORTE	—	—	—	—	RE3	RE2	RE1	RE0	---- x000	---- x000
89h	TRISE	IBF	OBF	IBOV	PSPMODE	— <sup>(1)</sup>	PORTE Data Direction bits			0000 1111	0000 1111
9Fh	ADCON1	ADFM	ADCS2	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0	0000 0000	0000 0000

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

**Note 1:** RE3 is an input only. The state of the TRISE3 bit has no effect and will always read '1'.

## REGISTER 5-1: TRISE REGISTER (ADDRESS 89h)

R-0	R-0	R/W-0	R/W-0	U-0	R/W-1	R/W-1	R/W-1
IBF	OBF	IBOV	PSPMODE	— <sup>(1)</sup>	TRISE2	TRISE1	TRISE0
bit 7					bit 0		

bit 7 **Parallel Slave Port Status/Control bits:**

**IBF:** Input Buffer Full Status bit

1 = A word has been received and is waiting to be read by the CPU  
0 = No word has been received

bit 6 **OBF:** Output Buffer Full Status bit

1 = The output buffer still holds a previously written word  
0 = The output buffer has been read

bit 5 **IBOV:** Input Buffer Overflow Detect bit (in Microprocessor mode)

1 = A write occurred when a previously input word has not been read (must be cleared in software)  
0 = No overflow occurred

bit 4 **PSPMODE:** Parallel Slave Port Mode Select bit

1 = Parallel Slave Port mode  
0 = General Purpose I/O mode

bit 3 **Unimplemented:** Read as '1'<sup>(1)</sup>

**Note 1:** RE3 is an input only. The state of the TRISE3 bit has no effect and will always read '1'.

bit 2 **PORTE Data Direction bits:**

**TRISE2:** Direction Control bit for pin RE2/ $\overline{\text{CS}}$ /AN7

1 = Input  
0 = Output

bit 1 **TRISE1:** Direction Control bit for pin RE1/ $\overline{\text{WR}}$ /AN6

1 = Input  
0 = Output

bit 0 **TRISE0:** Direction Control bit for pin RE0/ $\overline{\text{RD}}$ /AN5

1 = Input  
0 = Output

**Legend:**

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

# PIC16F7X7

## 5.6 Parallel Slave Port

The Parallel Slave Port (PSP) is not implemented on the PIC16F737 or PIC16F767.

PORTD operates as an 8-bit wide Parallel Slave Port or microprocessor port when control bit, PSPMODE (TRISE<4>), is set. In Slave mode, it is asynchronously readable and writable by an external system using the read control input pin RE0/ $\overline{RD}$ /AN5, the write control input pin RE1/ $\overline{WR}$ /AN6 and the chip select control input pin RE2/ $\overline{CS}$ /AN7.

The PSP can directly interface to an 8-bit microprocessor data bus. The external microprocessor can read or write the PORTD latch as an 8-bit latch. Setting bit PSPMODE enables port pin RE0/ $\overline{RD}$ /AN5 to be the  $\overline{RD}$  input, RE1/ $\overline{WR}$ /AN6 to be the  $\overline{WR}$  input and RE2/ $\overline{CS}$ /AN7 to be the  $\overline{CS}$  (Chip Select) input. For this functionality, the corresponding data direction bits of the TRISE register (TRISE<2:0>) must be configured as inputs (i.e., set). The A/D port configuration bits, PCFG3:PCFG0 (ADCON1<3:0>), must be set to configure pins RE2:RE0 as digital I/O.

There are actually two 8-bit latches, one for data output (external reads) and one for data input (external writes). The firmware writes 8-bit data to the PORTD output data latch and reads data from the PORTD input data latch (note that they have the same address). In this mode, the TRISD register is ignored since the external device is controlling the direction of data flow.

An external write to the PSP occurs when the  $\overline{CS}$  and  $\overline{WR}$  lines are both detected low. Firmware can read the actual data on the PORTD pins during this time. When either the  $\overline{CS}$  or  $\overline{WR}$  lines become high (level triggered), the data on the PORTD pins is latched and the Input Buffer Full (IBF) status flag bit (TRISE<7>) and interrupt flag bit, PSPIF (PIR1<7>), are set on the Q4 clock cycle following the next Q2 cycle to signal the write is complete (Figure 5-21). Firmware clears the IBF flag by reading the latched PORTD data and clears the PSPIF bit.

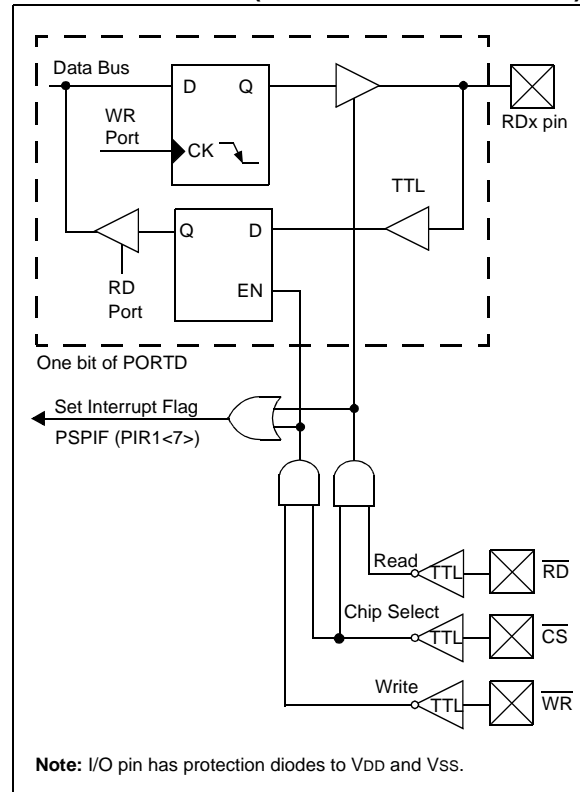
The Input Buffer Overflow (IBOV) status flag bit (TRISE<5>) is set if an external write to the PSP occurs while the IBF flag is set from a previous external write. The previous PORTD data is overwritten with the new data. IBOV is cleared by reading PORTD and clearing IBOV.

A read from the PSP occurs when both the  $\overline{CS}$  and  $\overline{RD}$  lines are detected low. The data in the PORTD output latch is output to the PORTD pins. The Output Buffer Full (OBF) status flag bit (TRISE<6>) is cleared immediately (Figure 5-22), indicating that the PORTD latch is being read or has been read by the external bus. If firmware writes new data to the output latch during this time, it is immediately output to the PORTD pins but OBF will remain cleared.

When either the  $\overline{CS}$  or  $\overline{RD}$  pins are detected high, the PORTD outputs are disabled and the interrupt flag bit PSPIF is set on the Q4 clock cycle following the next Q2 cycle, indicating that the read is complete. OBF remains low until firmware writes new data to PORTD.

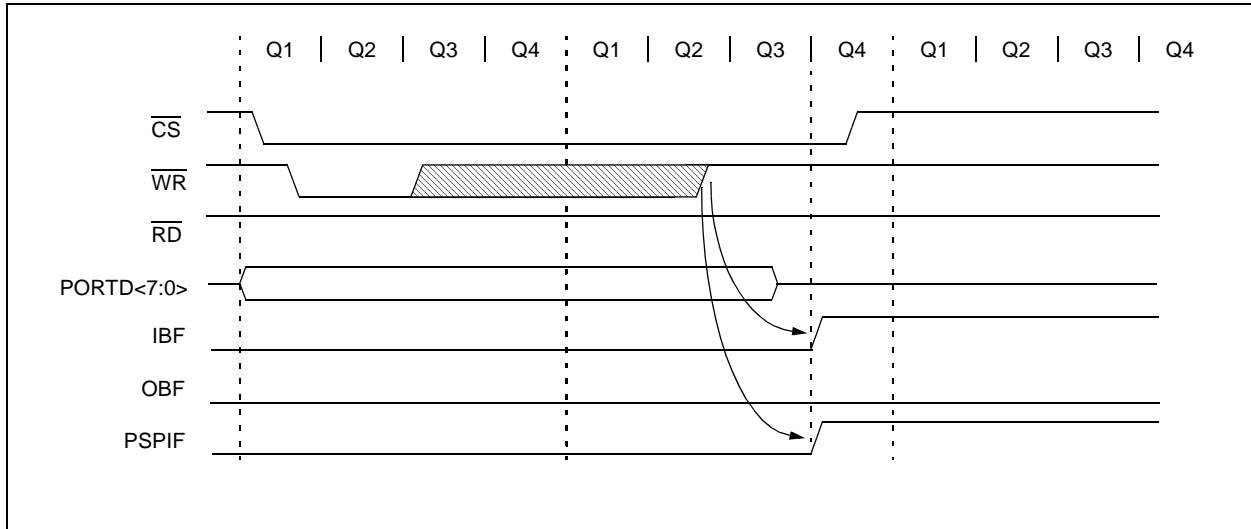
When not in PSP mode, the IBF and OBF bits are held clear. Flag bit IBOV remains unchanged. The PSPIF bit must be cleared by the user in firmware; the interrupt can be disabled by clearing the interrupt enable bit, PSPIE (PIE1<7>).

**FIGURE 5-20: PORTD AND PORTE BLOCK DIAGRAM (PARALLEL SLAVE PORT)**

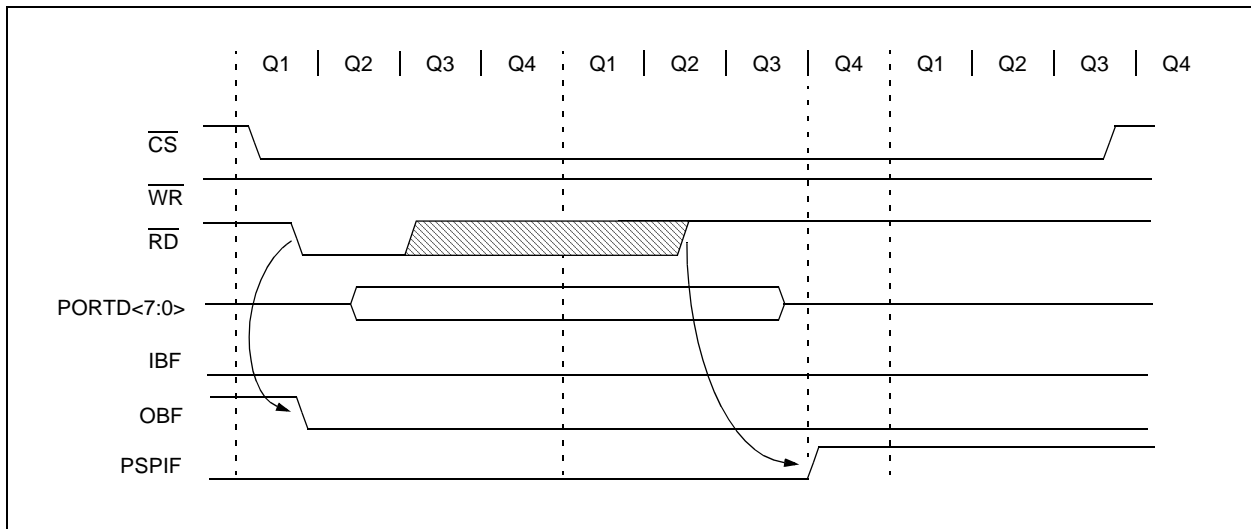




**FIGURE 5-21: PARALLEL SLAVE PORT WRITE WAVEFORMS**



**FIGURE 5-22: PARALLEL SLAVE PORT READ WAVEFORMS**



**TABLE 5-11: REGISTERS ASSOCIATED WITH PARALLEL SLAVE PORT**

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
08h	PORTD	Port Data Latch when written: Port pins when read								xxxx xxxx	uuuu uuuu
09h	PORTE	—	—	—	—	RE3	RE2	RE1	RE0	---- x000	---- x000
89h	TRISE	IBF	OBF	IBOV	PSPMODE	— <sup>(2)</sup>	PORTE Data Direction bits			0000 1111	0000 1111
0Ch	PIR1	PSPIF <sup>(1)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
8Ch	PIE1	PSPIE <sup>(1)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
9Fh	ADCON1	ADFM	ADCS2	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0	0000 0000	0000 0000

**Legend:** x = unknown, u = unchanged, — = unimplemented, read as '0'. Shaded cells are not used by the Parallel Slave Port.

**Note 1:** Bits PSPIE and PSPIF are reserved on the PIC16F737/767; always maintain these bits clear.

**Note 2:** RE3 is an input only. The state of the TRISE3 bit has no effect and will always read '1'.

# PIC16F7X7

---

NOTES:

## 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.

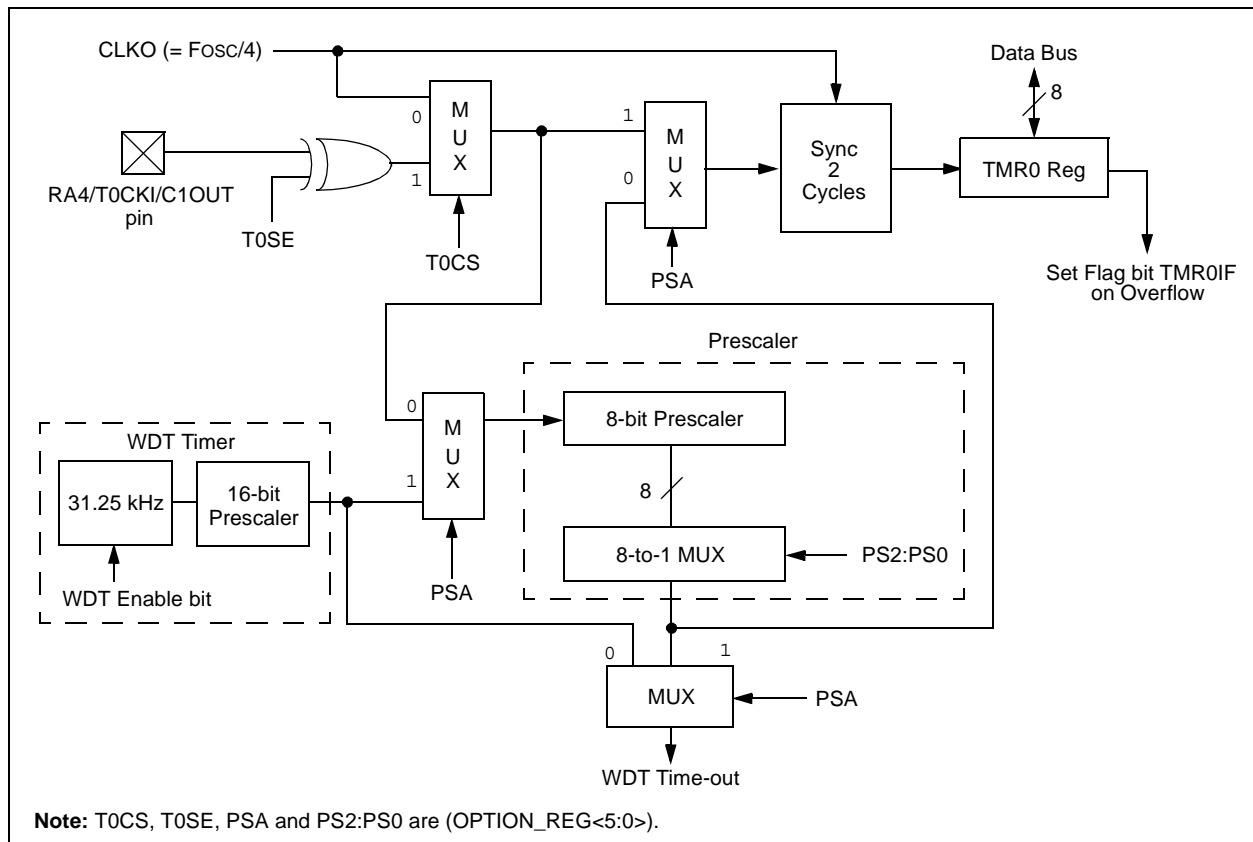
Counter mode is selected by setting bit, T0CS (OPTION\_REG<5>). In Counter mode, Timer0 will increment, either on every rising or falling edge of pin RA4/T0CKI/C1OUT. The incrementing edge is determined by the Timer0 Source Edge Select bit, T0SE (OPTION\_REG<4>). Clearing bit T0SE selects the rising 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**



# PIC16F7X7

---

## 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 T<sub>OSC</sub> (and a small RC delay of 20 ns) and low for at least 2 T<sub>OSC</sub> (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 15.17 “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., CLRF 1, MOVWF 1, BSF 1, x,...etc.) will clear the prescaler. When assigned to WDT, a CLRWDT instruction will clear the prescaler along with the Watchdog Timer. The prescaler is not readable or writable.

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

## REGISTER 6-1: OPTION\_REG: OPTION CONTROL REGISTER (ADDRESS 181h)

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
$\overline{\text{RBP}}\text{U}$	INTEDG	T0CS	T0SE	PSA <sup>(1)</sup>	PS2	PS1	PS0
bit 7						bit 0	

bit 7 **RBP**U: PORTB Pull-up Enable bit

- 1 = PORTB pull-ups are disabled
- 0 = PORTB pull-ups are enabled

bit 6 **INTEDG**: Interrupt Edge Select bit

- 1 = Interrupt on rising edge of RB0/INT pin
- 0 = Interrupt on falling edge of RB0/INT pin

bit 5 **T0CS**: TMR0 Clock Source Select bit

- 1 = Transition on T0CKI pin
- 0 = Internal instruction cycle clock (CLKO)

bit 4 **T0SE**: TMR0 Source Edge Select bit

- 1 = Increment on high-to-low transition on T0CKI pin
- 0 = Increment on low-to-high transition on T0CKI pin

bit 3 **PSA**: Prescaler Assignment bit<sup>(1)</sup>

- 1 = Prescaler is assigned to the WDT
- 0 = Prescaler is assigned to the Timer0 module

**Note 1:** To avoid an unintended device Reset, the instruction sequence shown in the "PICmicro<sup>®</sup> 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.

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

# PIC16F7X7

## 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, - = unimplemented locations read as '0'. Shaded cells are not used by Timer0.

## 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 9.4 "Capture Mode"**). Register 7-1 shows the Timer1 Control register.

When the Timer1 oscillator is enabled (T1OSCEN is set), the RC0/T1OSO/T1CKI and RC1/T1OSI/CCP2 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).

# PIC16F7X7

## 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 RC0/T1OSO/T1CKI (on the rising edge)  
 0 = Internal clock (Fosc/4)
- bit 0 **TMR1ON:** Timer1 On bit  
 1 = Enables Timer1  
 0 = Stops Timer1

### Legend:

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



## 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, 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.

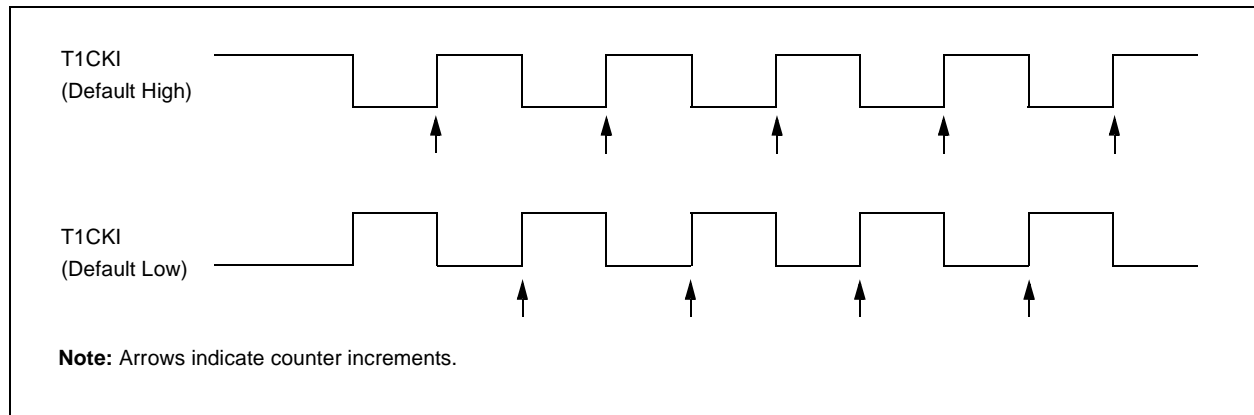
## 7.4 Timer1 Operation in Synchronized Counter Mode

Counter mode is selected by setting bit TMR1CS. In this mode, the timer increments on every rising edge of clock input on pin RC1/T1OSI/CCP2 when bit T1OSCEEN is set, or on pin RC0/T1OSO/T1CKI when bit T1OSCEEN is cleared.

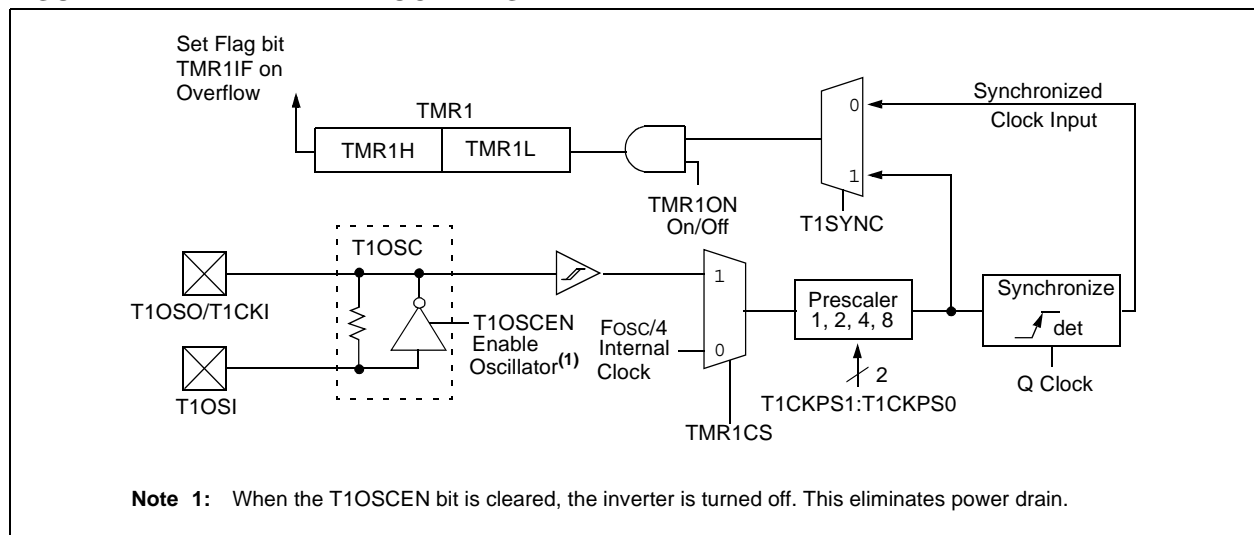
If T1SYNC is cleared, then the external clock input is synchronized with internal phase clocks. The synchronization is done after the prescaler stage. The prescaler stage is an asynchronous ripple counter.

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**



# PIC16F7X7

## 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 (**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   TMR1H, F    ; Write Low byte
; Re-enable the Interrupt (if required)
CONTINUE ; Continue with your code
```

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

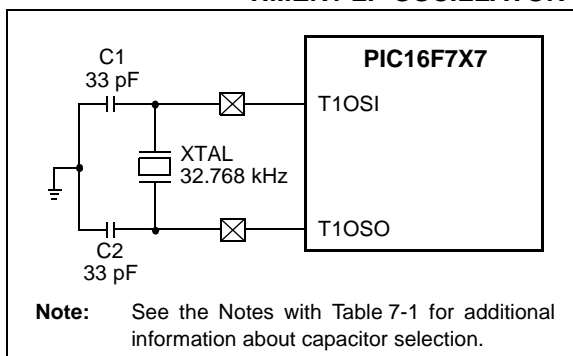
```
; All interrupts are disabled
MOVF    TMR1H, W    ; Read high byte
MOVWF   TMPH
MOVF    TMR1L, W    ; Read low byte
MOVWF   TMPL
MOVF    TMR1H, W    ; Read high byte
SUBWF   TMPH, W    ; Sub 1st read with 2nd read
BTFS    STATUS, Z   ; Is result = 0
GOTO    CONTINUE    ; Good 16-bit read
; TMR1L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

## 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.

**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.

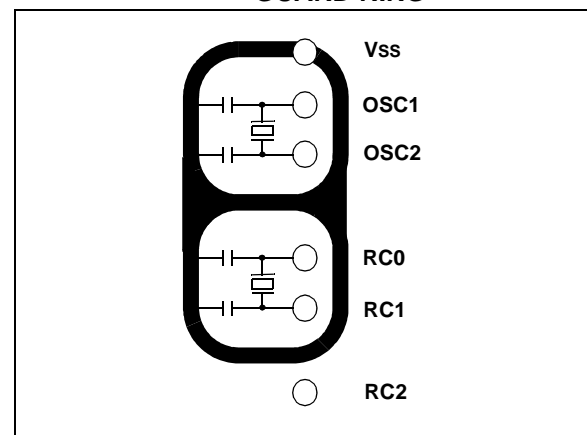
## 7.7 Timer1 Oscillator Layout Considerations

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**



## 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.

# PIC16F7X7

## 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 prescaler. In all other Resets, the register is unaffected.

## 7.10 Timer1 Prescaler

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

## 7.11 Using Timer1 as a Real-Time Clock

Adding an external LP oscillator to Timer1 (such as the one described in **Section 7.6 “Timer1 Oscillator”**) gives users the option to include RTC functionality in their applications. This is accomplished with an inexpensive watch crystal to provide an accurate time base and several lines of application code to calculate the time. When operating in Sleep mode and using a

battery or supercapacitor as a power source, it can completely eliminate the need for a separate RTC device and battery backup.

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

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

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

### 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	PSPIF <sup>(1)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
8Ch	PIE1	PSPIE <sup>(1)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 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:** Bits PSPIE and PSPIF are reserved on the PIC16F737/767 devices; always maintain these bits clear.

# PIC16F7X7

---

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 CCP module(s). 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:

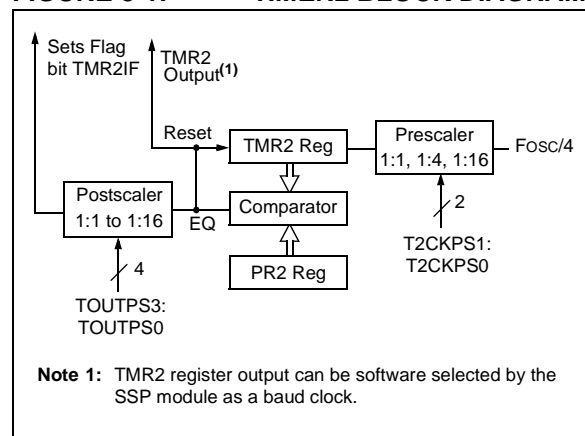
- a write to the TMR2 register
- a write to the T2CON register
- any device Reset (POR, MCLR Reset, WDT Reset or BOR)

TMR2 is not cleared when T2CON is written.

## 8.2 Output of TMR2

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

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



# PIC16F7X7

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

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

- bit 7 **Unimplemented:** Read as '0'
- bit 6-3 **TOUTPS3:TOUTPS0:** Timer2 Output Postscale Select bits  
 0000 = 1:1 Postscale  
 0001 = 1:2 Postscale  
 0010 = 1:3 Postscale  
 •  
 •  
 •  
 1111 = 1:16 Postscale
- bit 2 **TMR2ON:** Timer2 On bit  
 1 = Timer2 is on  
 0 = Timer2 is off
- bit 1-0 **T2CKPS1:T2CKPS0:** Timer2 Clock Prescale Select bits  
 00 = Prescaler is 1  
 01 = Prescaler is 4  
 1x = Prescaler is 16

### Legend:

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

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

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other Resets
0Bh,8Bh, 10Bh, 18Bh	INTCON	GIE	PEIE	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF <sup>(1)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
8Ch	PIE1	PSPIE <sup>(1)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 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:** Bits PSPIE and PSPIF are reserved on the PIC16F737/767 devices; always maintain these bits clear.



## 9.0 CAPTURE/COMPARE/PWM MODULES

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

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

The CCP1, CCP2 and CCP3 modules are identical in operation, with the exception being the operation of the special event trigger. Table 9-1 and Table 9-2 show the resources and interactions of the CCP module(s). In the following sections, the operation of a CCP module is described with respect to CCP1. CCP2 and CCP3 operate the same as CCP1, except where noted.

### 9.1 CCP1 Module

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 and will clear both TMR1H and TMR1L registers.

### 9.2 CCP2 Module

Capture/Compare/PWM Register 2 (CCPR2) is comprised of two 8-bit registers: CCPR2L (low byte) and CCPR2H (high byte). The CCP2CON register controls the operation of CCP2. The special event trigger is generated by a compare match; it will clear both TMR1H and TMR1L registers and start an A/D conversion (if the A/D module is enabled).

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

### 9.3 CCP3 Module

Capture/Compare/PWM Register 3 (CCPR3) is comprised of two 8-bit registers: CCPR3L (low byte) and CCPR3H (high byte). The CCP3CON register controls the operation of CCP3.

**TABLE 9-1: CCP MODE – TIMER RESOURCES REQUIRED**

CCP Mode	Timer Resource
Capture	Timer1
Compare	Timer1
PWM	Timer2

**TABLE 9-2: INTERACTION OF TWO CCP MODULES**

CCPx Mode	CCPy Mode	Interaction
Capture	Capture	Same TMR1 time base.
Capture	Compare	Same TMR1 time base.
Compare	Compare	Same TMR1 time base.
PWM	PWM	The PWMs will have the same frequency and update rate (TMR2 interrupt). The rising edges are aligned.
PWM	Capture	None.
PWM	Compare	None.

# PIC16F7X7

## REGISTER 9-1: CCPxCON: CCPx CONTROL REGISTER (ADDRESS 17h, 1Dh, 97h)

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	
—	—	CCPxX	CCPxY	CCPxM3	CCPxM2	CCPxM1	CCPxM0	
bit 7								bit 0

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

bit 5-4 **CCPxX:CCPxY:** 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 CCPxL.

bit 3-0 **CCPxM3:CCPxM0:** CCPx Mode Select bits

0000 = Capture/Compare/PWM disabled (resets CCPx 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 (CCPxIF bit is set)

1001 = Compare mode, clear output on match (CCPxIF bit is set)

1010 = Compare mode, generate software interrupt on match (CCPxIF bit is set, CCPx pin is unaffected)

1011 = Compare mode, trigger special event (CCPxIF bit is set, CCPx pin is unaffected);

CCP1 clears Timer1; CCP2 clears Timer1 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

## 9.4 Capture Mode

In Capture mode, CCPR1H:CCPR1L captures the 16-bit value of the TMR1 register when an event occurs on pin RC2/CCP1. An event is defined as one of the following and is configured by CCPxCON<3:0>:

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

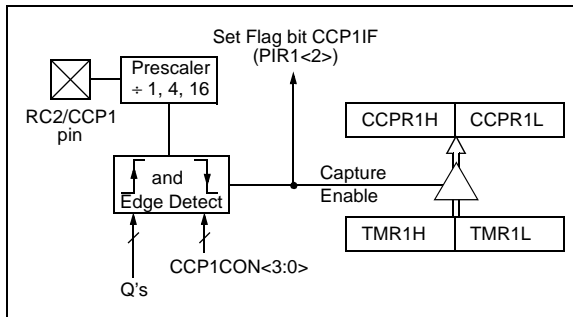
An event is selected by control bits, CCP1M3:CCP1M0 (CCP1CON<3:0>). When a capture is made, the interrupt request flag bit, CCP1IF (PIR1<2>), is set. The interrupt flag must be cleared in software. If another capture occurs before the value in register CCPR1 is read, the old captured value is overwritten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

### 9.4.1 CCP PIN CONFIGURATION

In Capture mode, the RC2/CCP1 pin should be configured as an input by setting the TRISC<2> bit.

**Note:** If the RC2/CCP1 pin is configured as an output, a write to the port can cause a capture condition.

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



### 9.4.2 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 will not work.

### 9.4.3 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.

### 9.4.4 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. Any Reset will clear the prescaler counter.

Switching from one capture prescaler to another may generate an interrupt. Also, the prescaler counter will not be cleared, therefore, the first capture may be from a non-zero prescaler. Example 9-1 shows the recommended method for switching between capture prescalers. This example also clears the prescaler counter and will not generate the “false” interrupt.

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

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

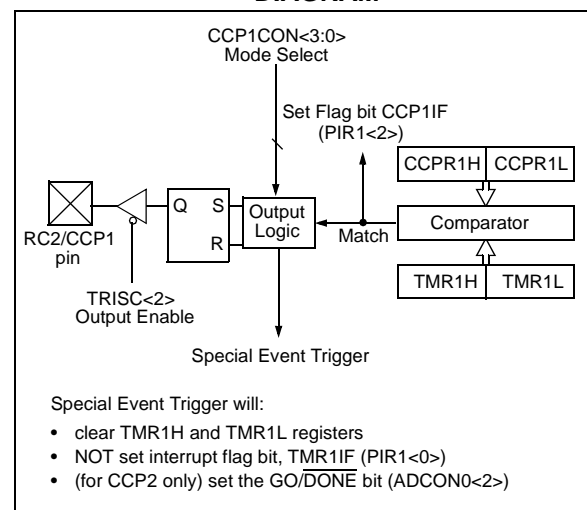
## 9.5 Compare Mode

In Compare mode, the 16-bit CCPR1 register value is constantly compared against the TMR1 register pair value. When a match occurs, the RC2/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**



# PIC16F7X7

## 9.5.1 CCP PIN CONFIGURATION

The user must configure the RC2/CCP1 pin as an output by clearing the TRISC<2> bit.

**Note:** Clearing the CCP1CON register will force the RC2/CCP1 compare output latch to the default low level. This is not the PORTC I/O data latch.

## 9.5.2 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.

## 9.5.3 SOFTWARE INTERRUPT MODE

When Generate Software Interrupt mode is chosen, the CCP1 pin is not affected. The CCP1IF or CCP2IF bit is set, causing a CCP interrupt (if enabled).

## 9.5.4 SPECIAL EVENT TRIGGER

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

The special event trigger output of CCP1 resets the TMR1 register pair. This allows the CCPR1 register to effectively be a 16-bit programmable period register for Timer1.

The special event trigger output of CCP2 resets the TMR1 register pair and starts an A/D conversion (if the A/D module is enabled).

**Note:** The special event trigger from the CCP1 and CCP2 modules will not set interrupt flag bit, TMR1IF (PIR1<0>).

**TABLE 9-3: 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	PSPIF <sup>(1)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
0Dh	PIR2	OSFIF	CMIF	LVDIF	—	BCLIF	—	CCP3IF	CCP2IF	000- 0-00	000- 0-00
8Ch	PIE1	PSPIE <sup>(1)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
8Dh	PIE2	OSFIE	CMIE	LVDIE	—	BCLIE	—	CCP3IE	CCP2IE	000- 0-00	000- 0-00
87h	TRISC	PORTC 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
1Bh	CCPR2L	Capture/Compare/PWM Register 2 (LSB)								xxxx xxxx	uuuu uuuu
1Ch	CCPR2H	Capture/Compare/PWM Register 2 (MSB)								xxxx xxxx	uuuu uuuu
1Dh	CCP2CON	—	—	CCP2X	CCP2Y	CCP2M3	CCP2M2	CCP2M1	CCP2M0	--00 0000	--00 0000
95h	CCPR3L	Capture/Compare/PWM Register 3 (LSB)								xxxx xxxx	uuuu uuuu
96h	CCPR3H	Capture/Compare/PWM Register 3 (MSB)								xxxx xxxx	uuuu uuuu
97h	CCP3CON	—	—	CCP3X	CCP3Y	CCP3M3	CCP3M2	CCP3M1	CCP3M0	--00 0000	--00 0000

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

**Note 1:** The PSP is not implemented on the PIC16F737/767 devices; always maintain these bits clear.

## 9.6 PWM Mode (PWM)

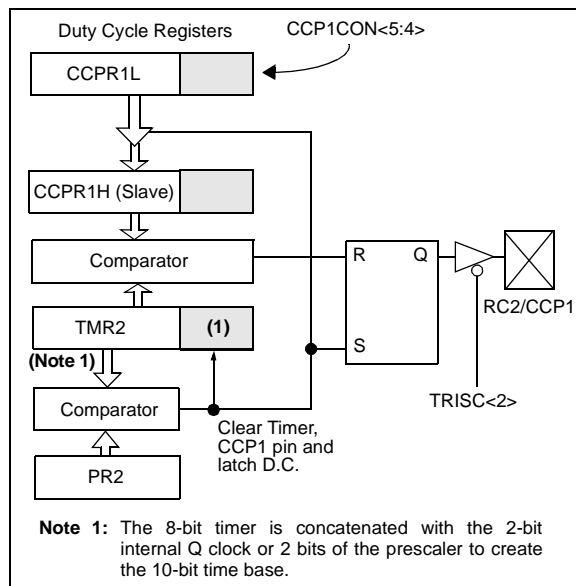
In Pulse-Width Modulation mode, the CCPx pin produces up to a 10-bit resolution PWM output. Since the CCP1 pin is multiplexed with the PORTC data latch, the TRISC<2> bit must be cleared to make the CCP1 pin an output.

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

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

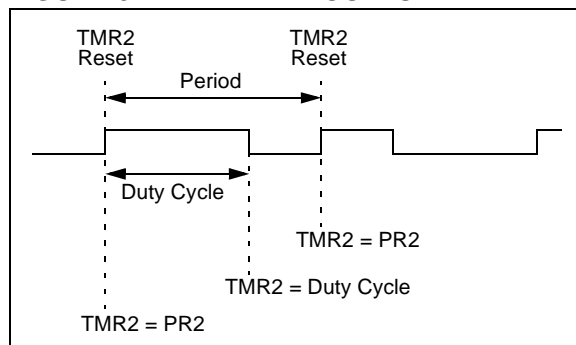
For a step-by-step procedure on how to set up the CCP module for PWM operation, see **Section 9.6.3 “Setup for PWM Operation”**.

**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**



### 9.6.1 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 9.4 “Capture Mode”**) 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.

### 9.6.2 PWM DUTY CYCLE

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

**EQUATION 9-2:**

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

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

# PIC16F7X7

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 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.

### 9.6.3 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 TRISC<2> bit.
4. Set the TMR2 prescale value and enable Timer2 by writing to T2CON.
5. Configure the CCP1 module for PWM operation.

**TABLE 9-4: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS (Fosc = 20 MHz)**

PWM Frequency	1.22 kHz	4.88 kHz	19.53 kHz	78.12 kHz	156.3 kHz	208.3 kHz
Timer Prescale (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 9-5: 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	PSPIF <sup>(1)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
0Dh	PIR2	OSFIF	CMIF	LVDIF	—	BCLIF	—	CCP3IF	CCP2IF	000- 0-00	000- 0-00
8Ch	PIE1	PSPIE <sup>(1)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
8Dh	PIE2	OSFIE	CMIE	LVDIE	—	BCLIE	—	CCP3IE	CCP2IE	000- 0-00	000- 0-00
87h	TRISC	PORTC 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
1Bh	CCPR2L	Capture/Compare/PWM Register 2 (LSB)								xxxx xxxx	uuuu uuuu
1Ch	CCPR2H	Capture/Compare/PWM Register 2 (MSB)								xxxx xxxx	uuuu uuuu
1Dh	CCP2CON	—	—	CCP2X	CCP2Y	CCP2M3	CCP2M2	CCP2M1	CCP2M0	--00 0000	--00 0000
95h	CCPR3L	Capture/Compare/PWM Register 3 (LSB)								xxxx xxxx	uuuu uuuu
96h	CCPR3H	Capture/Compare/PWM Register 3 (MSB)								xxxx xxxx	uuuu uuuu
97h	CCP3CON	—	—	CCP3X	CCP3Y	CCP3M3	CCP3M2	CCP3M1	CCP3M0	--00 0000	--00 0000

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

**Note 1:** Bits PSPIE and PSPIF are reserved on the PIC16F737/767 devices; always maintain these bits clear.

## 10.0 MASTER SYNCHRONOUS SERIAL PORT (MSSP) MODULE

### 10.1 Master SSP (MSSP) Module Overview

The Master Synchronous Serial Port (MSSP) 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 MSSP module can operate in one of two modes:

- Serial Peripheral Interface (SPI™)
- Inter-Integrated Circuit (I<sup>2</sup>C™)
  - Full Master mode
  - Slave mode (with general address call)

The I<sup>2</sup>C interface supports the following modes in hardware:

- Master mode
- Multi-Master mode
- Slave mode

### 10.2 Control Registers

The MSSP module has three associated registers. These include a status register (SSPSTAT) and two control registers (SSPCON and SSPCON2). The use of these registers and their individual configuration bits differ significantly, depending on whether the MSSP module is operated in SPI or I<sup>2</sup>C mode.

Additional details are provided under the individual sections.

### 10.3 SPI Mode

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

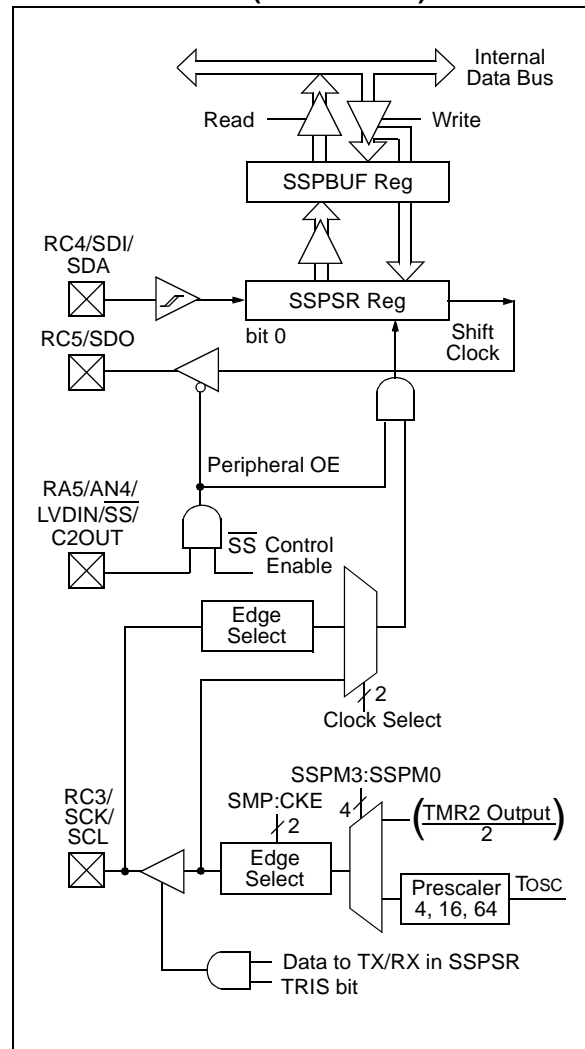
- Serial Data Out (SDO) – RC5/SDO
- Serial Data In (SDI) – RC4/SDI/SDA
- Serial Clock (SCK) – RC3/SCK/SCL

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

- Slave Select ( $\overline{SS}$ ) – RA5/AN4/LVDIN/ $\overline{SS}$ /C2OUT

Figure 10-1 shows the block diagram of the MSSP module when operating in SPI mode.

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



# PIC16F7X7

## 10.3.1 REGISTERS

The MSSP module has four registers for SPI mode operation. These are:

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

SSPCON and SSPSTAT are the control and status registers in SPI mode operation. The SSPCON register is readable and writable. The lower 6 bits of the SSPSTAT are read-only. The upper two bits of the SSPSTAT are read/write.

SSPSR is the shift register used for shifting data in or out. SSPBUF is the buffer register to which data bytes are written to or read from.

In receive operations, SSPSR and SSPBUF together create a double-buffered receiver. When SSPSR receives a complete byte, it is transferred to SSPBUF and the SSPIF interrupt is set.

During transmission, the SSPBUF is not double-buffered. A write to SSPBUF will write to both SSPBUF and SSPSR.

### REGISTER 10-1: SSPSTAT: MSSP STATUS (SPI MODE) 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	S	R $\bar{W}$	UA	BF
bit 7						bit 0	

- bit 7     **SMP:** Sample bit  
SPI Master mode:  
 1 = Input data sampled at end of data output time  
 0 = Input data sampled at middle of data output time  
SPI Slave mode:  
 SMP must be cleared when SPI is used in Slave mode.
- 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 (SSPCON1<4>).
- bit 5     **D $\bar{A}$ :** Data/Address bit  
 Used in I<sup>2</sup>C mode only.
- bit 4     **P:** Stop bit  
 Used in I<sup>2</sup>C mode only. This bit is cleared when the MSSP module is disabled, SSPEN is cleared.
- bit 3     **S:** Start bit  
 Used in I<sup>2</sup>C mode only.
- bit 2     **R $\bar{W}$ :** Read/Write bit Information  
 Used in I<sup>2</sup>C mode only.
- bit 1     **UA:** Update Address bit  
 Used in I<sup>2</sup>C mode only.
- bit 0     **BF:** Buffer Full Status bit (Receive mode only)  
 1 = Receive complete, SSPBUF is full  
 0 = Receive not complete, SSPBUF is empty

<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



## REGISTER 10-2: SSPCON: MSSP CONTROL (SPI MODE) REGISTER 1 (ADDRESS 14h)

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

bit 7 **WCOL:** Write Collision Detect bit (Transmit mode only)

- 1 = The SSPBUF register is written while it is still transmitting the previous word. (Must be cleared in software.)
- 0 = No collision

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

SPI Slave 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. (Must be cleared in software.)
- 0 = No overflow

**Note:** In Master mode, the overflow bit is not set since each new reception (and transmission) is initiated by writing to the SSPBUF register.

bit 5 **SSPEN:** Synchronous Serial Port Enable bit

- 1 = Enables serial port and configures SCK, SDO, SDI and  $\overline{SS}$  as serial port pins
- 0 = Disables serial port and configures these pins as I/O port pins

**Note:** When enabled, these pins must be properly configured as input or output.

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

- 1 = Idle state for clock is a high level
- 0 = Idle state for clock is a low level

bit 3-0 **SSPM3:SSPM0:** Synchronous Serial Port Mode Select bits

- 0101 = SPI Slave mode, clock = SCK pin.  $\overline{SS}$  pin control disabled.  $\overline{SS}$  can be used as I/O pin.
- 0100 = SPI Slave mode, clock = SCK pin.  $\overline{SS}$  pin control enabled.
- 0011 = SPI Master mode, clock = TMR2 output/2
- 0010 = SPI Master mode, clock = FOSC/64
- 0001 = SPI Master mode, clock = FOSC/16
- 0000 = SPI Master mode, clock = FOSC/4

**Note:** Bit combinations not specifically listed here are either reserved or implemented in I<sup>2</sup>C mode 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

# PIC16F7X7

## 10.3.2 OPERATION

When initializing the SPI, several options need to be specified. This is done by programming the appropriate control bits (SSPCON<5:0> and 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)
- Data Input Sample Phase (middle or end of data output time)
- Clock Edge (output data on rising/falling edge of SCK)
- Clock Rate (Master mode only)
- Slave Select mode (Slave mode only)

The MSSP consists of a Transmit/Receive Shift register (SSPSR) and a Buffer register (SSPBUF). The SSPSR shifts the data in and out of the device, MSb first. The SSPBUF holds the data that was written to the SSPSR until the received data is ready. Once the 8 bits of data have been received, that byte is moved to the SSPBUF register. Then, the Buffer Full detect bit, BF (SSPSTAT<0>) and the interrupt flag bit, SSPIF, are set. This double-buffering of the received data (SSPBUF) allows the next byte to start reception before reading the

data that was just received. Any write to the SSPBUF register during transmission/reception of data will be ignored and the Write Collision detect bit, WCOL (SSPCON<7>), will be set. User software must clear the WCOL bit so that it can be determined if the following write(s) to the SSPBUF register completed successfully.

When the application software is expecting to receive valid data, the SSPBUF should be read before the next byte of data to transfer is written to the SSPBUF. Buffer Full bit, BF (SSPSTAT<0>), indicates when SSPBUF has been loaded with the received data (transmission is complete). When the SSPBUF is read, the BF bit is cleared. This data may be irrelevant if the SPI is only a transmitter. Generally, the MSSP interrupt is used to determine when the transmission/reception has completed. The SSPBUF must be read and/or written. If the interrupt method is not going to be used, then software polling can be done to ensure that a write collision does not occur. Example 10-1 shows the loading of the SSPBUF (SSPSR) for data transmission.

The SSPSR is not directly readable or writable and can only be accessed by addressing the SSPBUF register. Additionally, the MSSP Status register (SSPSTAT) indicates the various status conditions.

### EXAMPLE 10-1: LOADING THE SSPBUF (SSPSR) REGISTER

LOOP	BTFSS	SSPSTAT, BF	;Has data been received (transmit complete)?
	BRA	LOOP	;No
	MOVF	SSPBUF, W	;WREG reg = contents of SSPBUF
	MOVWF	RXDATA	;Save in user RAM, if data is meaningful
	MOVF	TXDATA, W	;W reg = contents of TXDATA
	MOVWF	SSPBUF	;New data to xmit

### 10.3.3 ENABLING SPI I/O

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

- SDI is automatically controlled by the SPI module
- SDO must have TRISC<5> bit cleared
- SCK (Master mode) must have TRISC<3> bit cleared
- SCK (Slave mode) must have TRISC<3> bit set
- $\overline{\text{SS}}$  must have TRISA<5> bit set

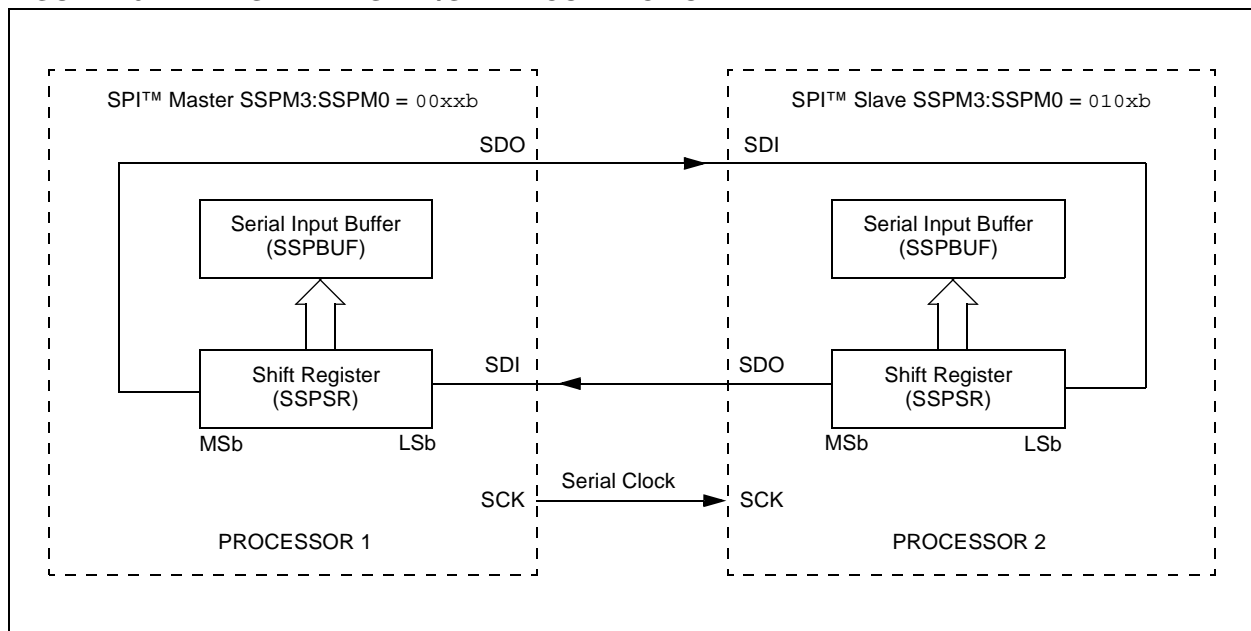
Any serial port function that is not desired may be overridden by programming the corresponding data direction (TRIS) register to the opposite value.

### 10.3.4 TYPICAL CONNECTION

Figure 10-2 shows a typical connection between two microcontrollers. The master controller (Processor 1) initiates the data transfer by sending the SCK signal. Data is shifted out of both shift registers on their programmed clock edge and latched on the opposite edge of the clock. Both processors should be programmed to the same Clock Polarity (CKP), then both controllers would send and receive data at the same time. Whether the data is meaningful (or dummy data) depends on the application software. This leads to three scenarios for data transmission:

- Master sends data – Slave sends dummy data
- Master sends data – Slave sends data
- Master sends dummy data – Slave sends data

**FIGURE 10-2: SPI™ MASTER/SLAVE CONNECTION**



# PIC16F7X7

## 10.3.5 MASTER MODE

The master can initiate the data transfer at any time because it controls the SCK. The master determines when the slave (Processor 2, Figure 10-2) is to broadcast data by the software protocol.

In Master mode, the data is transmitted/received as soon as the SSPBUF register is written to. If the SPI is only going to receive, the SDO output could be disabled (programmed as an input). The SSPSR register will continue to shift in the signal present on the SDI pin at the programmed clock rate. As each byte is received, it will be loaded into the SSPBUF register as if it is a normal received byte (interrupts and status bits appropriately set). This could be useful in receiver applications, such as a "Line Activity Monitor" mode.

The clock polarity is selected by appropriately programming the CKP bit (SSPCON<4>). This then, would give waveforms for SPI communication as shown in

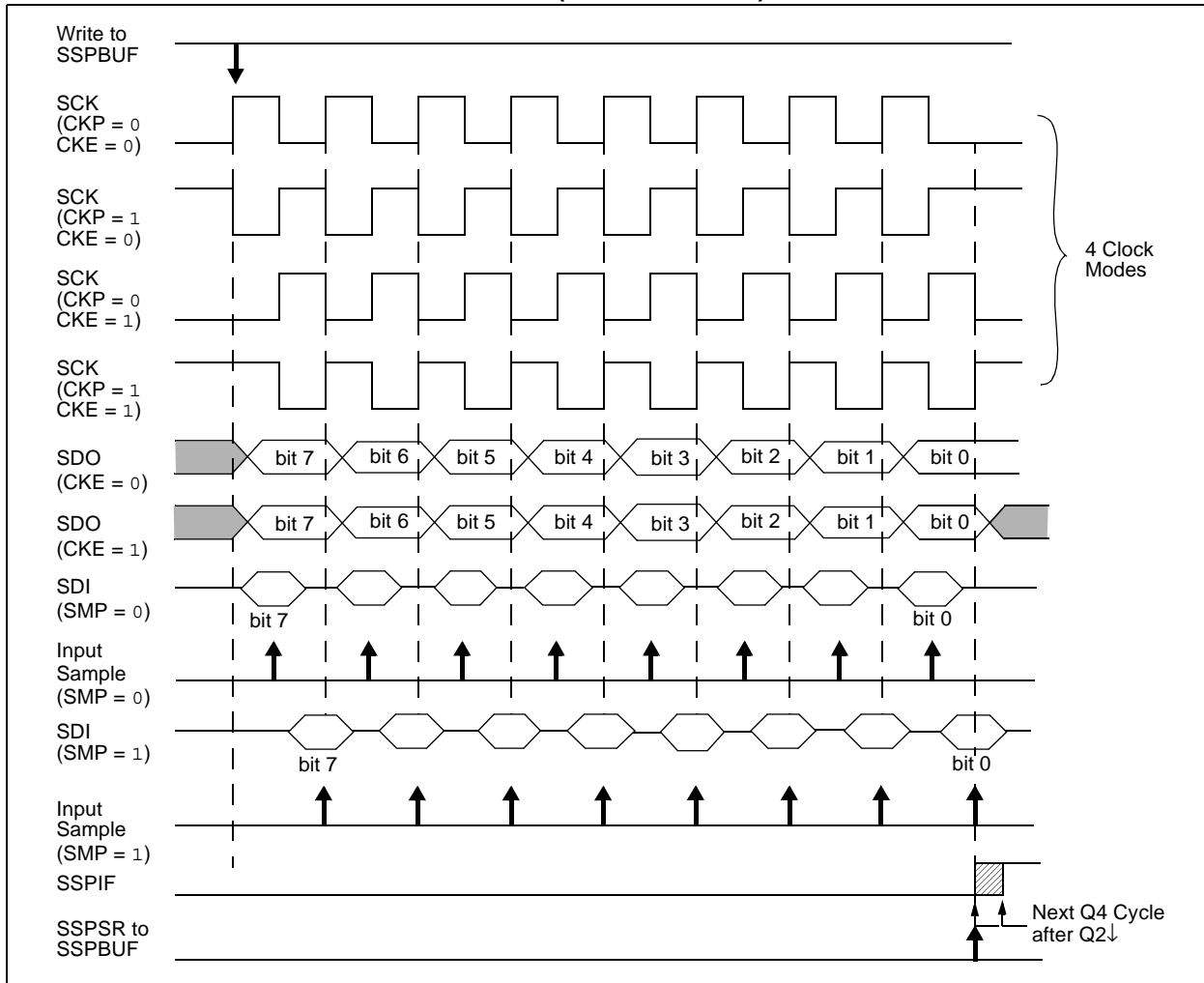
Figure 10-3, Figure 10-5 and Figure 10-6, where the MSB is transmitted first. In Master mode, the SPI clock rate (bit rate) is user programmable to be one of the following:

- $F_{OSC}/4$  (or  $T_{CY}$ )
- $F_{OSC}/16$  (or  $4 \cdot T_{CY}$ )
- $F_{OSC}/64$  (or  $16 \cdot T_{CY}$ )
- $\text{Timer2 output}/2$

This allows a maximum data rate (at 40 MHz) of 10.00 Mbps.

Figure 10-3 shows the waveforms for Master mode. When the CKE bit is set, the SDO data is valid before there is a clock edge on SCK. The change of the input sample is shown based on the state of the SMP bit. The time when the SSPBUF is loaded with the received data is shown.

**FIGURE 10-3: SPI™ MODE WAVEFORM (MASTER MODE)**



## 10.3.6 SLAVE MODE

In Slave mode, the data is transmitted and received as the external clock pulses appear on SCK. When the last bit is latched, the SSPIF interrupt flag bit is set.

While in Slave mode, the external clock is supplied by the external clock source on the SCK pin. This external clock must meet the minimum high and low times, as specified in the electrical specifications.

While in Sleep mode, the slave can transmit/receive data. When a byte is received, the device will wake-up from Sleep.

Before enabling the module in SPI Slave mode, the clock line must match the proper Idle state. The clock line can be observed by reading the SCK pin. The Idle state is determined by the CKP bit (SSPCON1<4>).

## 10.3.7 SLAVE SELECT SYNCHRONIZATION

The  $\overline{SS}$  pin allows a Synchronous Slave mode. The SPI must be in Slave mode with  $\overline{SS}$  pin control enabled (SSPCON<3:0> = 4h). The pin must not be driven low for the  $\overline{SS}$  pin to function as an input. The data latch

must be high. When the  $\overline{SS}$  pin is low, transmission and reception are enabled and the SDO pin is driven. When the  $\overline{SS}$  pin goes high, the SDO pin is no longer driven, even if in the middle of a transmitted byte and becomes a floating output. External pull-up/pull-down resistors may be desirable, depending on the application.

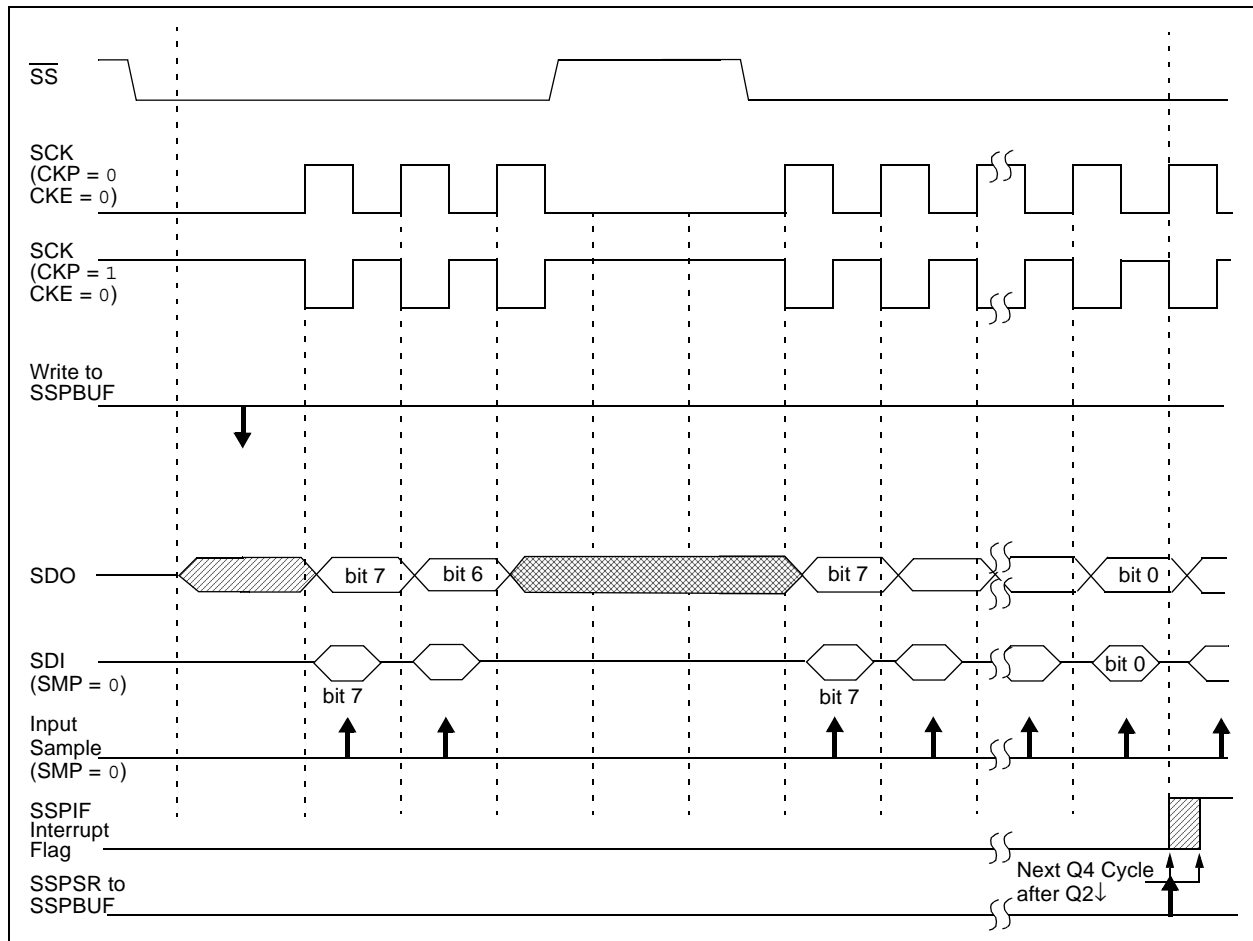
**Note 1:** When the SPI is in Slave mode with  $\overline{SS}$  pin control enabled (SSPCON<3:0> = 0100), the SPI module will reset if the  $\overline{SS}$  pin is set to VDD.

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

When the SPI module resets, the bit counter is forced to '0'. This can be done by either forcing the  $\overline{SS}$  pin to a high level or clearing the SSPEN bit.

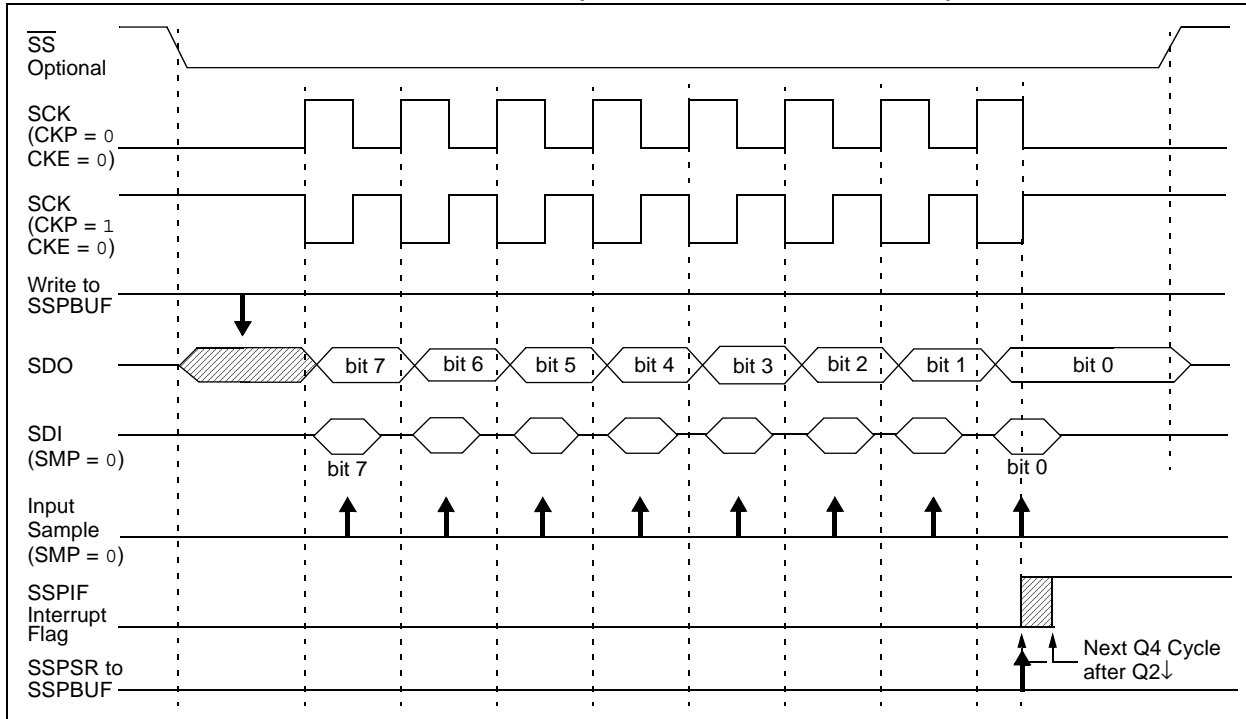
To emulate two-wire communication, the SDO pin can be connected to the SDI pin. When the SPI needs to operate as a receiver, the SDO pin can be configured as an input. This disables transmissions from the SDO. The SDI can always be left as an input (SDI function) since it cannot create a bus conflict.

**FIGURE 10-4: SLAVE SYNCHRONIZATION WAVEFORM**

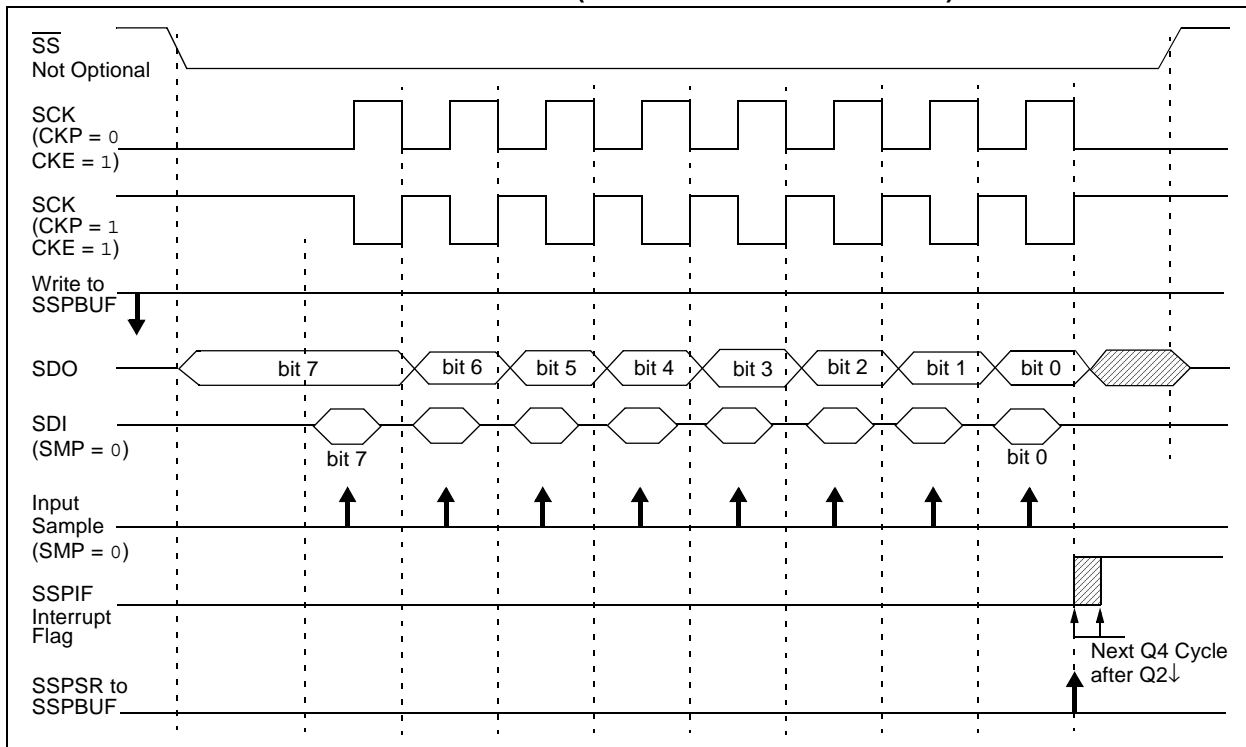


# PIC16F7X7

**FIGURE 10-5: SPI™ MODE WAVEFORM (SLAVE MODE WITH CKE = 0)**



**FIGURE 10-6: SPI™ MODE WAVEFORM (SLAVE MODE WITH CKE = 1)**



## 10.3.8 SLEEP OPERATION

In Master mode, all module clocks are halted and the transmission/reception will remain in that state until the device wakes from Sleep. After the device returns to normal mode, the module will continue to transmit/receive data.

In Slave mode, the SPI Transmit/Receive Shift register operates asynchronously to the device. This allows the device to be placed in Sleep mode and data to be shifted into the SPI Transmit/Receive Shift register. When all 8 bits have been received, the MSSP interrupt flag bit will be set and if enabled, will wake the device from Sleep.

## 10.3.9 EFFECTS OF A RESET

A Reset disables the MSSP module and terminates the current transfer.

## 10.3.10 BUS MODE COMPATIBILITY

Table 10-1 shows the compatibility between the standard SPI modes and the states of the CKP and CKE control bits.

**TABLE 10-1: SPI™ BUS MODES**

Standard SPI™ Mode Terminology	Control Bits State	
	CKP	CKE
0, 0	0	1
0, 1	0	0
1, 0	1	1
1, 1	1	0

There is also an SMP bit which controls when the data is sampled.

**TABLE 10-2: REGISTERS ASSOCIATED WITH SPI™ OPERATION**

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
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u
PIR1	PSPIF <sup>(1)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
PIE1	PSPIE <sup>(1)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
TRISC	PORTC Data Direction Register								1111 1111	1111 1111
SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register								xxxx xxxx	uuuu uuuu
SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	0000 0000
TRISA	PORTA Data Direction Register								1111 1111	1111 1111
SSPSTAT	SMP	CKE	D $\bar{A}$	P	S	R $\bar{W}$	UA	BF	0000 0000	0000 0000

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

**Note 1:** The PSPIF and PSPIE bits are reserved on 28-pin devices; always maintain these bits clear.

# PIC16F7X7

## 10.4 I<sup>2</sup>C Mode

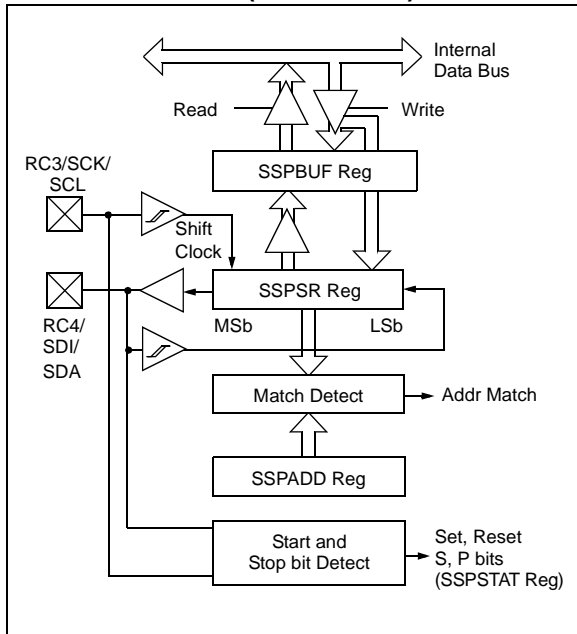
The MSSP module in I<sup>2</sup>C mode fully implements all master and slave functions (including general call support) and provides interrupts on Start and Stop bits in hardware to determine a free bus (multi-master function). The MSSP module implements the standard mode specifications, as well as 7-bit and 10-bit addressing.

Two pins are used for data transfer:

- Serial clock (SCL) – RC3/SCK/SCL
- Serial data (SDA) – RC4/SDI/SDA

The user must configure these pins as inputs or outputs through the TRISC<4:3> bits.

**FIGURE 10-7: MSSP BLOCK DIAGRAM (I<sup>2</sup>C™ MODE)**



### 10.4.1 REGISTERS

The MSSP module has six registers for I<sup>2</sup>C operation. These are:

- MSSP Control Register (SSPCON)
- MSSP Control Register 2 (SSPCON2)
- MSSP Status Register (SSPSTAT)
- Serial Receive/Transmit Buffer (SSPBUF)
- MSSP Shift Register (SSPSR) – Not directly accessible
- MSSP Address Register (SSPADD)

SSPCON, SSPCON2 and SSPSTAT are the control and status registers in I<sup>2</sup>C mode operation. The SSPCON and SSPCON2 registers are readable and writable. The lower 6 bits of the SSPSTAT are read-only. The upper two bits of the SSPSTAT are read/write.

SSPSR is the shift register used for shifting data in or out. SSPBUF is the buffer register to which data bytes are written to or read from.

SSPADD register holds the slave device address when the SSP is configured in I<sup>2</sup>C Slave mode. When the SSP is configured in Master mode, the lower seven bits of SSPADD act as the Baud Rate Generator reload value.

In receive operations, SSPSR and SSPBUF together create a double-buffered receiver. When SSPSR receives a complete byte, it is transferred to SSPBUF and the SSPIF interrupt is set.

During transmission, the SSPBUF is not double-buffered. A write to SSPBUF will write to both SSPBUF and SSPSR.



## REGISTER 10-3: SSPSTAT: MSSP STATUS (I<sup>2</sup>C MODE) 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	S	R $\bar{W}$	UA	BF
bit 7							bit 0

- bit 7 **SMP:** Slew Rate Control bit  
In Master or Slave mode:  
 1 = Slew rate control disabled for Standard Speed mode (100 kHz and 1 MHz)  
 0 = Slew rate control enabled for High-Speed mode (400 kHz)
- bit 6 **CKE:** SMBus Select bit  
In Master or Slave mode:  
 1 = Enable SMBus specific inputs  
 0 = Disable SMBus specific inputs
- bit 5 **D $\bar{A}$ :** Data/Address bit  
In Master mode:  
 Reserved.  
In Slave mode:  
 1 = Indicates that the last byte received or transmitted was data  
 0 = Indicates that the last byte received or transmitted was address
- bit 4 **P:** Stop bit  
 1 = Indicates that a Stop bit has been detected last  
 0 = Stop bit was not detected last  
**Note:** This bit is cleared on Reset and when SSPEN is cleared.
- bit 3 **S:** Start bit  
 1 = Indicates that a Start bit has been detected last  
 0 = Start bit was not detected last  
**Note:** This bit is cleared on Reset and when SSPEN is cleared.
- bit 2 **R $\bar{W}$ :** Read/Write bit Information bit (I<sup>2</sup>C mode only)  
In Slave mode:  
 1 = Read  
 0 = Write  
**Note:** This bit holds the R $\bar{W}$  bit information following the last address match. This bit is only valid from the address match to the next Start bit, Stop bit or not ACK bit.  
In Master mode:  
 1 = Transmit is in progress  
 0 = Transmit is not in progress  
**Note:** ORing this bit with SEN, RSEN, PEN, RCEN or ACKEN will indicate if the MSSP is in Idle mode.
- bit 1 **UA:** Update Address bit (10-bit Slave 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  
In Transmit mode:  
 1 = Receive complete, SSPBUF is full  
 0 = Receive not complete, SSPBUF is empty  
In Receive mode:  
 1 = Data transmit in progress (does not include the  $\bar{ACK}$  and Stop bits), SSPBUF is full  
 0 = Data transmit complete (does not include the  $\bar{ACK}$  and Stop bits), SSPBUF is empty

### 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

# PIC16F7X7

## REGISTER 10-4: SSPCON: MSSP CONTROL (I<sup>2</sup>C MODE) REGISTER 1 (ADDRESS 14h)

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

- bit 7 **WCOL:** Write Collision Detect bit  
In Master Transmit mode:  
 1 = A write to the SSPBUF register was attempted while the I<sup>2</sup>C conditions were not valid for a transmission to be started (must be cleared in software)  
 0 = No collision  
In Slave Transmit mode:  
 1 = The SSPBUF register is written while it is still transmitting the previous word (must be cleared in software)  
 0 = No collision  
In Receive mode (Master or Slave modes):  
 This is a “don’t care” bit.
- bit 6 **SSPOV:** Receive Overflow Indicator bit  
In Receive mode:  
 1 = A byte is received while the SSPBUF register is still holding the previous byte (must be cleared in software)  
 0 = No overflow  
In Transmit mode:  
 This is a “don’t care” bit in Transmit mode.
- bit 5 **SSPEN:** Synchronous Serial Port Enable bit  
 1 = Enables the serial port and configures the SDA and SCL pins as the serial port pins  
 0 = Disables serial port and configures these pins as I/O port pins  
**Note:** When enabled, the SDA and SCL pins must be properly configured as input or output.
- bit 4 **CKP:** SCK Release Control bit  
In Slave mode:  
 1 = Release clock  
 0 = Holds clock low (clock stretch). (Used to ensure data setup time.)  
In Master mode:  
 Unused in this mode.
- bit 3-0 **SSPM3:SSPM0:** Synchronous Serial Port Mode Select bits  
 1111 = I<sup>2</sup>C Slave mode, 10-bit address with Start and Stop bit interrupts enabled  
 1110 = I<sup>2</sup>C Slave mode, 7-bit address with Start and Stop bit interrupts enabled  
 1011 = I<sup>2</sup>C Firmware Controlled Master mode (slave Idle)  
 1000 = I<sup>2</sup>C Master mode, clock = FOSC/(4 \* (SSPADD + 1))  
 0111 = I<sup>2</sup>C Slave mode, 10-bit address  
 0110 = I<sup>2</sup>C Slave mode, 7-bit address  
**Note:** Bit combinations not specifically listed here are either reserved or implemented in SPI mode 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

## REGISTER 10-5: SSPCON2: MSSP CONTROL (I<sup>2</sup>C MODE) REGISTER 2 (ADDRESS 91h)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN
bit 7							bit 0

- bit 7 **GCEN:** General Call Enable bit (Slave mode only)  
 1 = Enable interrupt when a general call address (0000h) is received in the SSPSR  
 0 = General call address disabled
- bit 6 **ACKSTAT:** Acknowledge Status bit (Master Transmit mode only)  
 1 = Acknowledge was not received from slave  
 0 = Acknowledge was received from slave
- bit 5 **ACKDT:** Acknowledge Data bit (Master Receive mode only)  
 1 = Not Acknowledge  
 0 = Acknowledge  
**Note:** Value that will be transmitted when the user initiates an Acknowledge sequence at the end of a receive.
- bit 4 **ACKEN:** Acknowledge Sequence Enable bit (Master Receive mode only)  
 1 = Initiate Acknowledge sequence on SDA and SCL pins and transmit ACKDT data bit. Automatically cleared by hardware.  
 0 = Acknowledge sequence Idle
- bit 3 **RCEN:** Receive Enable bit (Master mode only)  
 1 = Enables Receive mode for I<sup>2</sup>C  
 0 = Receive Idle
- bit 2 **PEN:** Stop Condition Enable bit (Master mode only)  
 1 = Initiate Stop condition on SDA and SCL pins. Automatically cleared by hardware.  
 0 = Stop condition Idle
- bit 1 **RSEN:** Repeated Start Condition Enable bit (Master mode only)  
 1 = Initiate Repeated Start condition on SDA and SCL pins. Automatically cleared by hardware.  
 0 = Repeated Start condition Idle
- bit 0 **SEN:** Start Condition Enable/Stretch Enable bit  
In Master mode:  
 1 = Initiate Start condition on SDA and SCL pins. Automatically cleared by hardware.  
 0 = Start condition Idle  
In Slave mode:  
 1 = Clock stretching is enabled for both slave transmit and slave receive (stretch enabled)  
 0 = Clock stretching is enabled for slave transmit only (PIC16F87X compatibility)  
**Note:** For bits ACKEN, RCEN, PEN, RSEN, SEN: If the I<sup>2</sup>C module is not in the Idle mode, this bit may not be set (no spooling) and the SSPBUF may not be written (or writes to the SSPBUF are 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

# PIC16F7X7

## 10.4.2 OPERATION

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

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 Master mode, clock = Oscillator/4 (SSPADD + 1)
- 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
- I<sup>2</sup>C Slave mode (10-bit address), with Start and Stop bit interrupts enabled
- I<sup>2</sup>C Firmware Controlled Master mode, 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 TRISC bits. To ensure proper operation of the module, pull-up resistors must be provided externally to the SCL and SDA pins.

```
MOVWF  TRISC, W      ; Example for a 40-pin part such as the PIC16F877A
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 I<sup>2</sup>C Slave mode hardware will always generate an interrupt on an address match. Through the mode select bits, the user can also choose to interrupt on Start and Stop bits.

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

Any combination of the following conditions will cause the MSSP module not to give this  $\overline{ACK}$  pulse:

- The Buffer Full bit, BF (SSPSTAT<0>), was set before the transfer was received.
- 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. The BF bit 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 MSSP module, are shown in timing parameter #100 and parameter #101.

## 10.4.3 SLAVE MODE

In Slave mode, the SCL and SDA pins must be configured as inputs (TRISC<4:3> set). The MSSP module will override the input state with the output data when required (slave-transmitter).

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:

### 10.4.3.1 Addressing

Once the MSSP module has been enabled, it waits for a Start condition to occur. Following the Start condition, the 8 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:

1. The SSPSR register value is loaded into the SSPBUF register.
2. The Buffer Full bit, BF, is set.
3. An  $\overline{ACK}$  pulse is generated.
4. MSSP 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. The five Most Significant bits (MSBs) of the first address byte specify if this is a 10-bit address. Bit  $\overline{R/W}$  (SSPSTAT<2>) must specify a write so the slave device will receive the second address byte. For a 10-bit address, the first byte would equal '11110 A9 A8 0', where 'A9' and 'A8' are the two MSBs of the address. The sequence of events for 10-bit address is as follows, with steps 7 through 9 for the 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.

### 10.4.3.2 Reception

When the  $\overline{R/W}$  bit of the address byte is clear and an address match occurs, the  $\overline{R/W}$  bit of the SSPSTAT register is cleared. The received address is loaded into the SSPBUF register and the SDA line is held low ( $\overline{ACK}$ ).

When the address byte overflow condition exists, then the no Acknowledge ( $\overline{ACK}$ ) pulse is given. An overflow condition is defined as either bit BF (SSPSTAT<0>) is set or bit SSPOV (SSPCON<6>) is set.

An MSSP 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.

If SEN is enabled (SSPCON<0> = 1), RC3/SCK/SCL will be held low (clock stretch) following each data transfer. The clock must be released by setting bit, CKP (SSPCON<4>). See **Section 10.4.4 "Clock Stretching"** for more detail.

### 10.4.3.3 Transmission

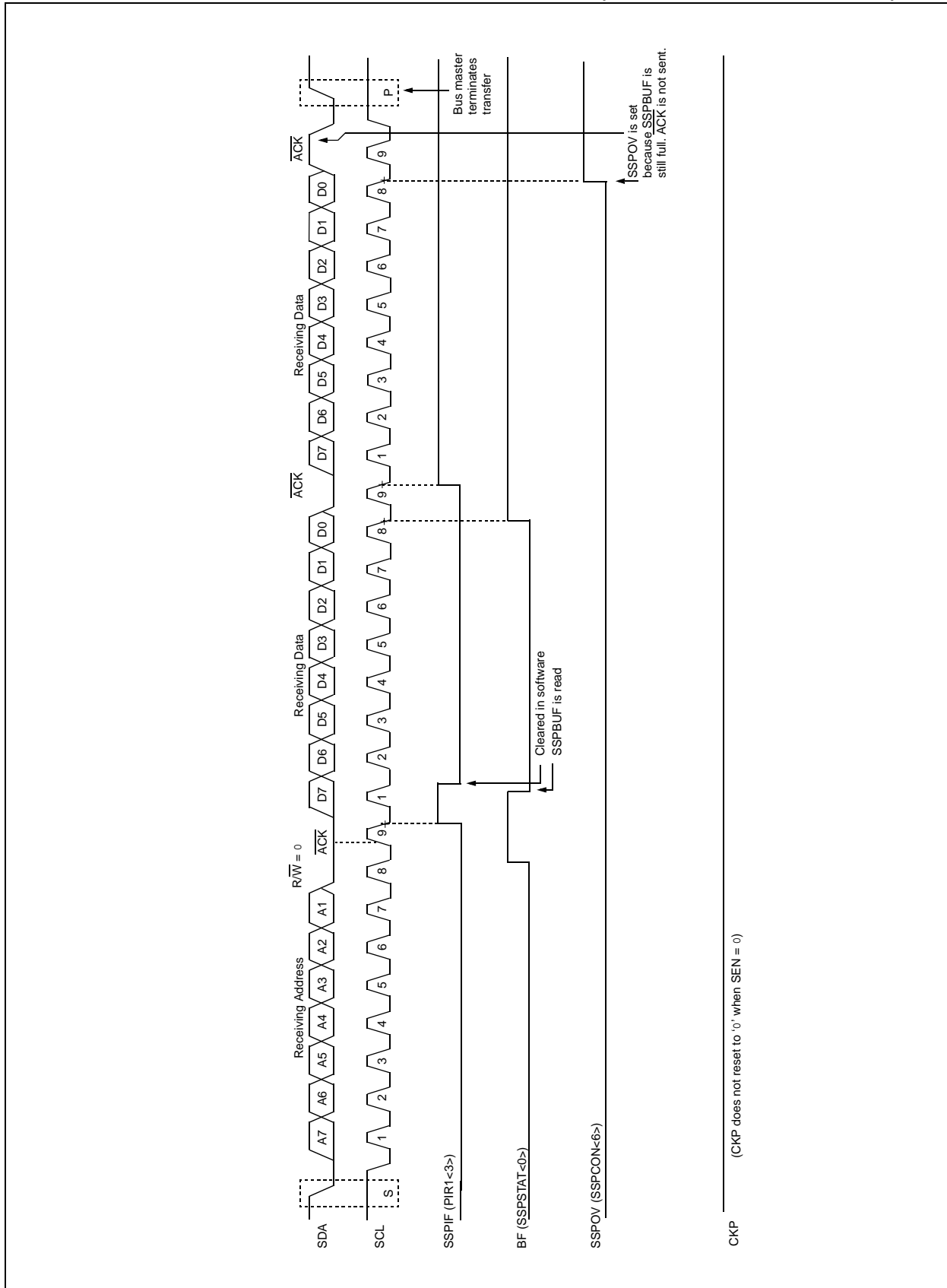
When the  $\overline{R/W}$  bit of the incoming address byte is set and an address match occurs, the  $\overline{R/W}$  bit of the SSPSTAT register is set. The received address is loaded into the SSPBUF register. The  $\overline{ACK}$  pulse will be sent on the ninth bit and pin RC3/SCK/SCL is held low regardless of SEN (see **Section 10.4.4 "Clock Stretching"** for more detail). By stretching the clock, the master will be unable to assert another clock pulse until the slave is done preparing the transmit data. The transmit data must be loaded into the SSPBUF register, which also loads the SSPSR register. Then pin RC3/SCK/SCL should be enabled by setting bit CKP (SSPCON<4>). 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 10-9).

The  $\overline{ACK}$  pulse from the master-receiver is latched on the rising edge of the ninth SCL input pulse. If the SDA line is high (not  $\overline{ACK}$ ), then the data transfer is complete. In this case, when the  $\overline{ACK}$  is latched by the slave, the slave logic is reset (resets SSPSTAT register) and the slave monitors for another occurrence of the Start bit. If the SDA line was low ( $\overline{ACK}$ ), the next transmit data must be loaded into the SSPBUF register. Again, pin RC3/SCK/SCL must be enabled by setting bit CKP.

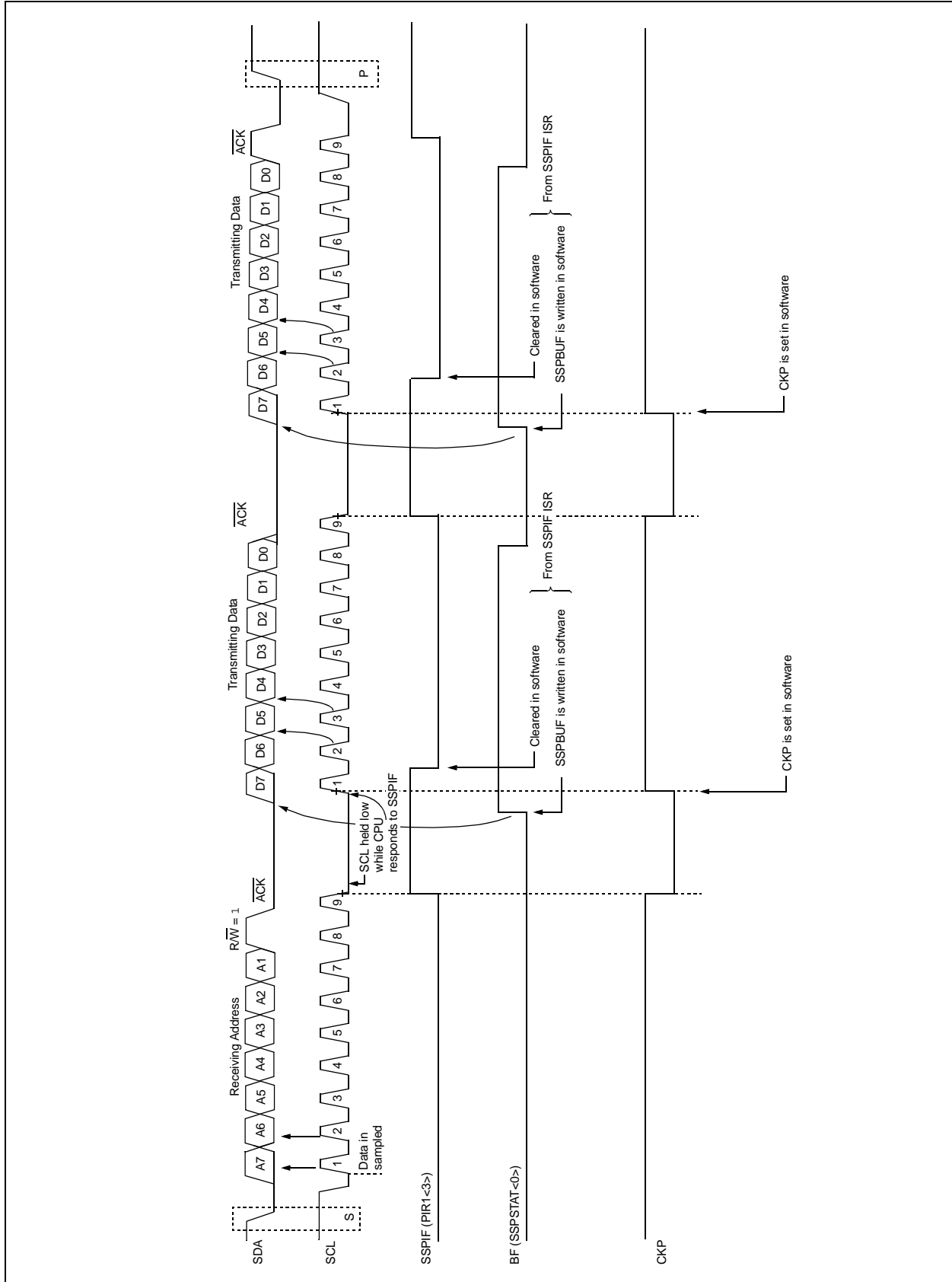
An MSSP interrupt is generated for each data transfer byte. The SSPIF bit must be cleared in software and the SSPSTAT register is used to determine the status of the byte. The SSPIF bit is set on the falling edge of the ninth clock pulse.

# PIC16F7X7

FIGURE 10-8: I<sup>2</sup>C™ SLAVE MODE TIMING WITH SEN = 0 (RECEPTION, 7-BIT ADDRESS)

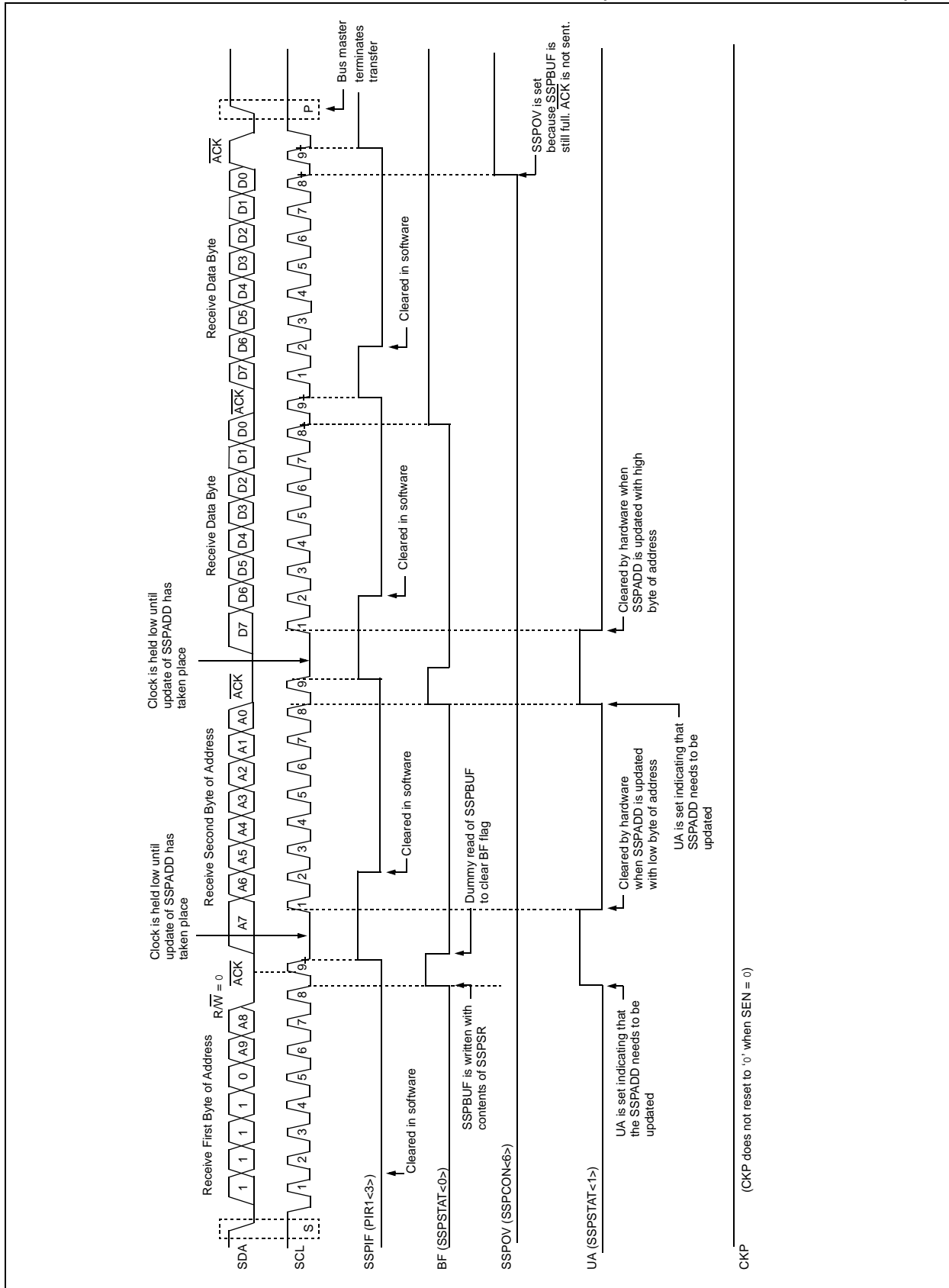


**FIGURE 10-9: I<sup>2</sup>C™ SLAVE MODE TIMING (TRANSMISSION, 7-BIT ADDRESS)**



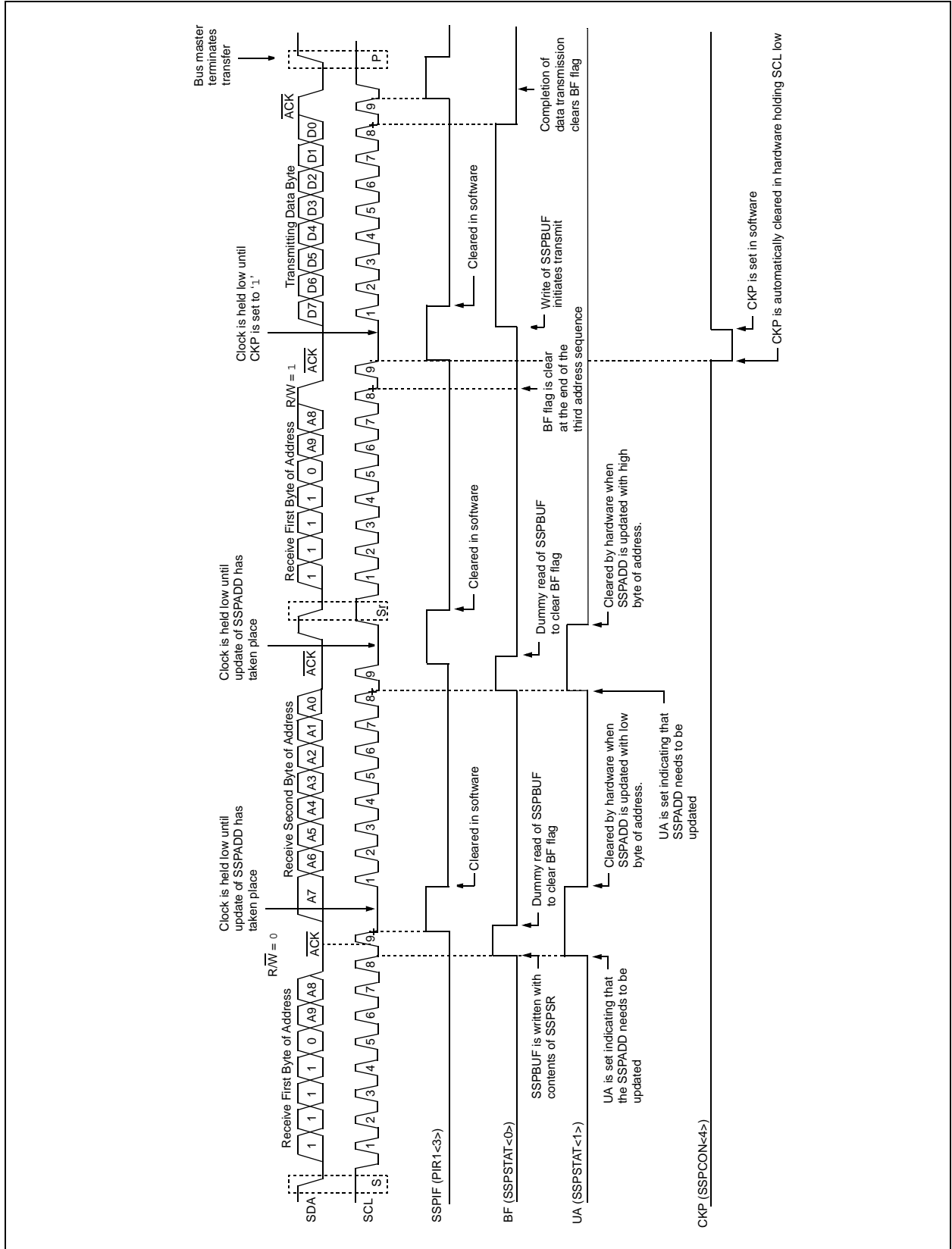
# PIC16F7X7

FIGURE 10-10: I<sup>2</sup>C™ SLAVE MODE TIMING WITH SEN = 0 (RECEPTION, 10-BIT ADDRESS)





**FIGURE 10-11: I<sup>2</sup>C™ SLAVE MODE TIMING (TRANSMISSION, 10-BIT ADDRESS)**



# PIC16F7X7

## 10.4.4 CLOCK STRETCHING

Both 7-bit and 10-bit Slave modes implement automatic clock stretching during a transmit sequence.

The SEN bit (SSPCON2<0>) allows clock stretching to be enabled during receives. Setting SEN will cause the SCL pin to be held low at the end of each data receive sequence.

### 10.4.4.1 Clock Stretching for 7-bit Slave Receive Mode (SEN = 1)

In 7-bit Slave Receive mode, on the falling edge of the ninth clock, at the end of the ACK sequence if the BF bit is set, the CKP bit in the SSPCON register is automatically cleared, forcing the SCL output to be held low. The CKP being cleared to '0' will assert the SCL line low. The CKP bit must be set in the user's ISR before reception is allowed to continue. By holding the SCL line low, the user has time to service the ISR and read the contents of the SSPBUF before the master device can initiate another receive sequence. This will prevent buffer overruns from occurring (see Figure 10-13).

**Note 1:** If the user reads the contents of the SSPBUF before the falling edge of the ninth clock, thus clearing the BF bit, the CKP bit will not be cleared and clock stretching will not occur.

**2:** The CKP bit can be set in software regardless of the state of the BF bit. The user should be careful to clear the BF bit in the ISR before the next receive sequence in order to prevent an overflow condition.

### 10.4.4.2 Clock Stretching for 10-bit Slave Receive Mode (SEN = 1)

In 10-bit Slave Receive mode during the address sequence, clock stretching automatically takes place but CKP is not cleared. During this time, if the UA bit is set after the ninth clock, clock stretching is initiated. The UA bit is set after receiving the upper byte of the 10-bit address and following the receive of the second byte of the 10-bit address, with the  $R/\overline{W}$  bit cleared to '0'. The release of the clock line occurs upon updating SSPADD. Clock stretching will occur on each data receive sequence as described in 7-bit mode.

**Note:** If the user polls the UA bit and clears it by updating the SSPADD register before the falling edge of the ninth clock occurs and if the user hasn't cleared the BF bit by reading the SSPBUF register before that time, then the CKP bit will still NOT be asserted low. Clock stretching on the basis of the state of the BF bit only occurs during a data sequence, not an address sequence.

### 10.4.4.3 Clock Stretching for 7-bit Slave Transmit Mode

7-bit Slave Transmit mode implements clock stretching by clearing the CKP bit after the falling edge of the ninth clock, if the BF bit is clear. This occurs regardless of the state of the SEN bit.

The user's ISR must set the CKP bit before transmission is allowed to continue. By holding the SCL line low, the user has time to service the ISR and load the contents of the SSPBUF before the master device can initiate another transmit sequence (see Figure 10-9).

**Note 1:** If the user loads the contents of SSPBUF, setting the BF bit before the falling edge of the ninth clock, the CKP bit will not be cleared and clock stretching will not occur.

**2:** The CKP bit can be set in software regardless of the state of the BF bit.

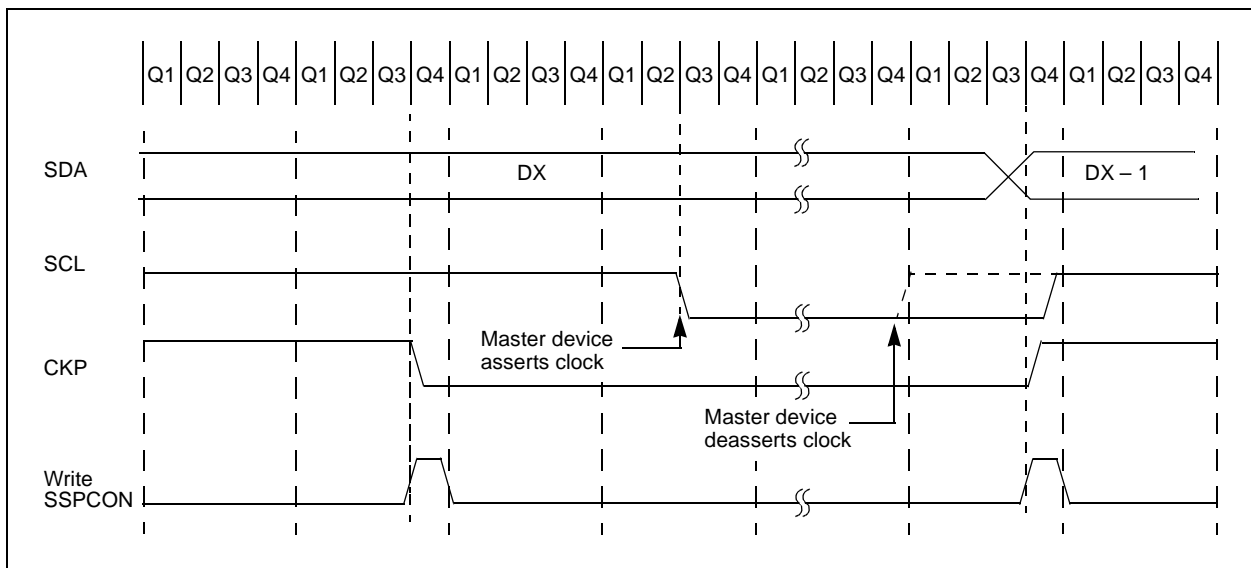
### 10.4.4.4 Clock Stretching for 10-bit Slave Transmit Mode

In 10-bit Slave Transmit mode, clock stretching is controlled during the first two address sequences by the state of the UA bit, just as it is in 10-bit Slave Receive mode. The first two addresses are followed by a third address sequence, which contains the high-order bits of the 10-bit address and the  $R/\overline{W}$  bit set to '1'. After the third address sequence is performed, the UA bit is not set, the module is now configured in Transmit mode and clock stretching is controlled by the BF flag as in 7-bit Slave Transmit mode (see Figure 10-11).

## 10.4.4.5 Clock Synchronization and the CKP Bit

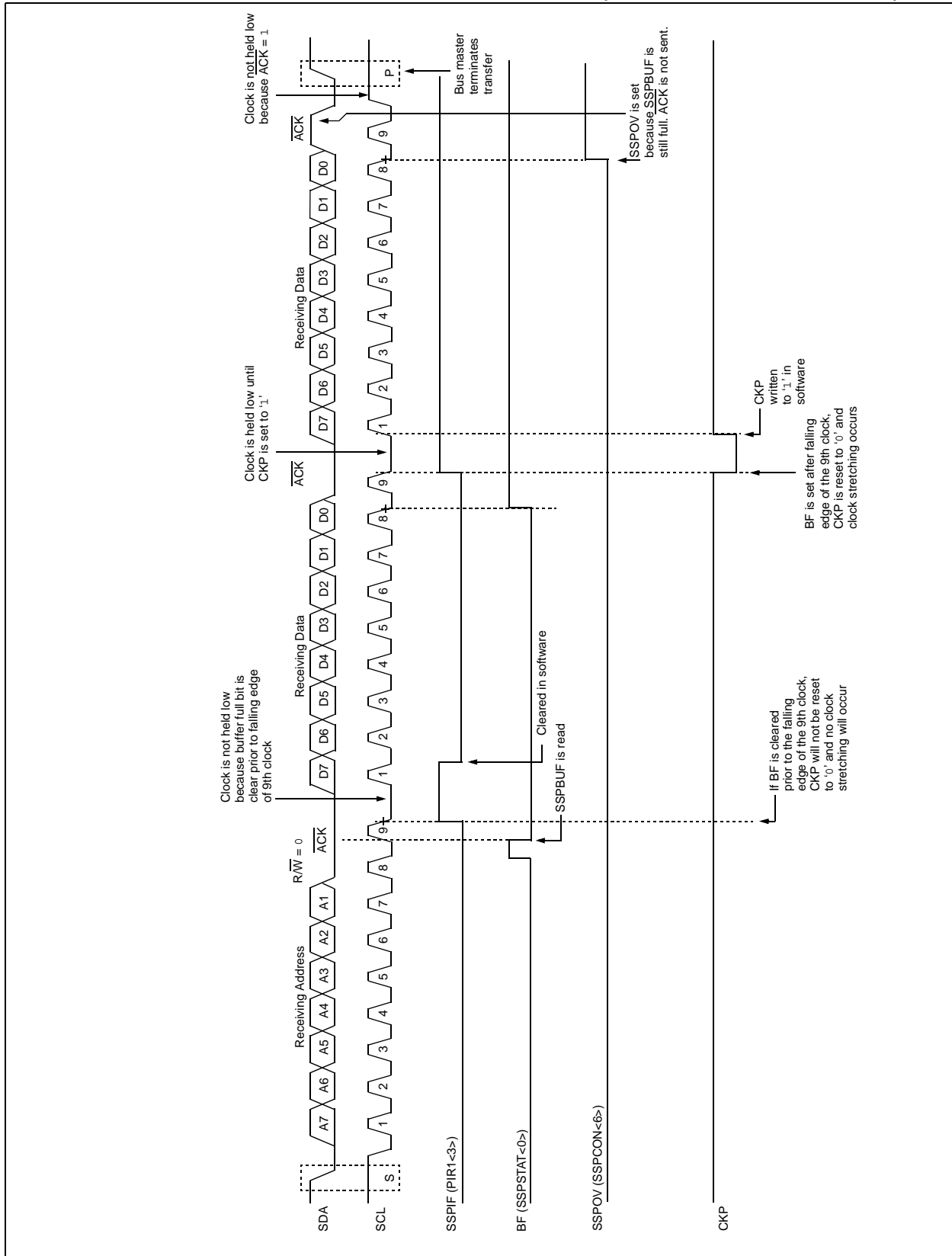
When the CKP bit is cleared, the SCL output is forced to '0'; however, setting the CKP bit will not assert the SCL output low until the SCL output is already sampled low. Therefore, the CKP bit will not assert the SCL line until an external I<sup>2</sup>C master device has already asserted the SCL line. The SCL output will remain low until the CKP bit is set and all other devices on the I<sup>2</sup>C bus have deasserted SCL. This ensures that a write to the CKP bit will not violate the minimum high time requirement for SCL (see Figure 10-12).

**FIGURE 10-12: CLOCK SYNCHRONIZATION TIMING**

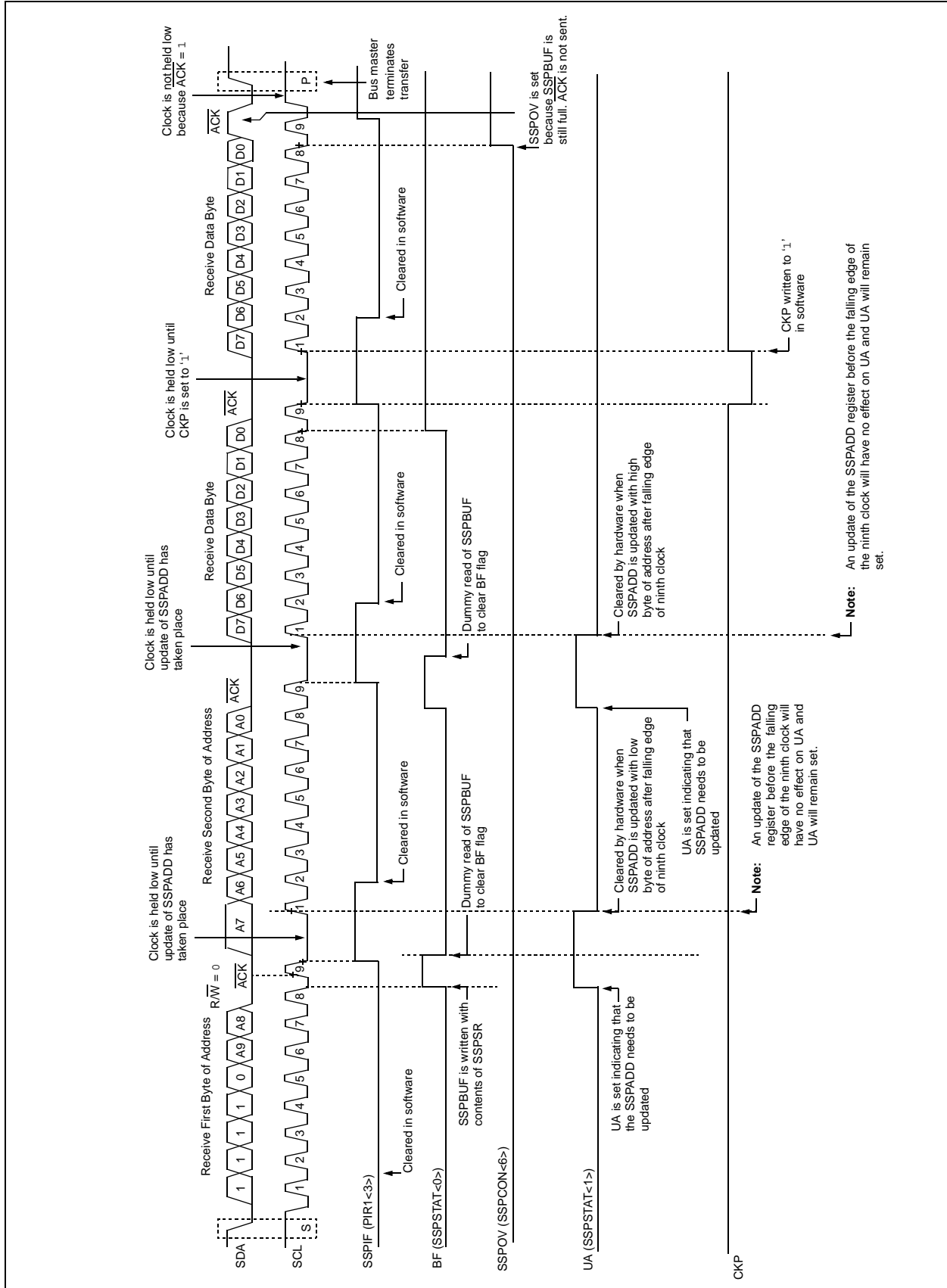


# PIC16F7X7

FIGURE 10-13: I<sup>2</sup>C™ SLAVE MODE TIMING WITH SEN = 1 (RECEPTION, 7-BIT ADDRESS)



**FIGURE 10-14: I<sup>2</sup>C™ SLAVE MODE TIMING WITH SEN = 1 (RECEPTION, 10-BIT ADDRESS)**



# PIC16F7X7

## 10.4.5 GENERAL CALL ADDRESS SUPPORT

The addressing procedure for the I<sup>2</sup>C bus is such that the first byte after the Start condition usually determines which device will be the slave addressed by the master. The exception is the general call address which can address all devices. When this address is used, all devices should, in theory, respond with an Acknowledge.

The general call address is one of eight addresses reserved for specific purposes by the I<sup>2</sup>C protocol. It consists of all '0's with R/W = 0.

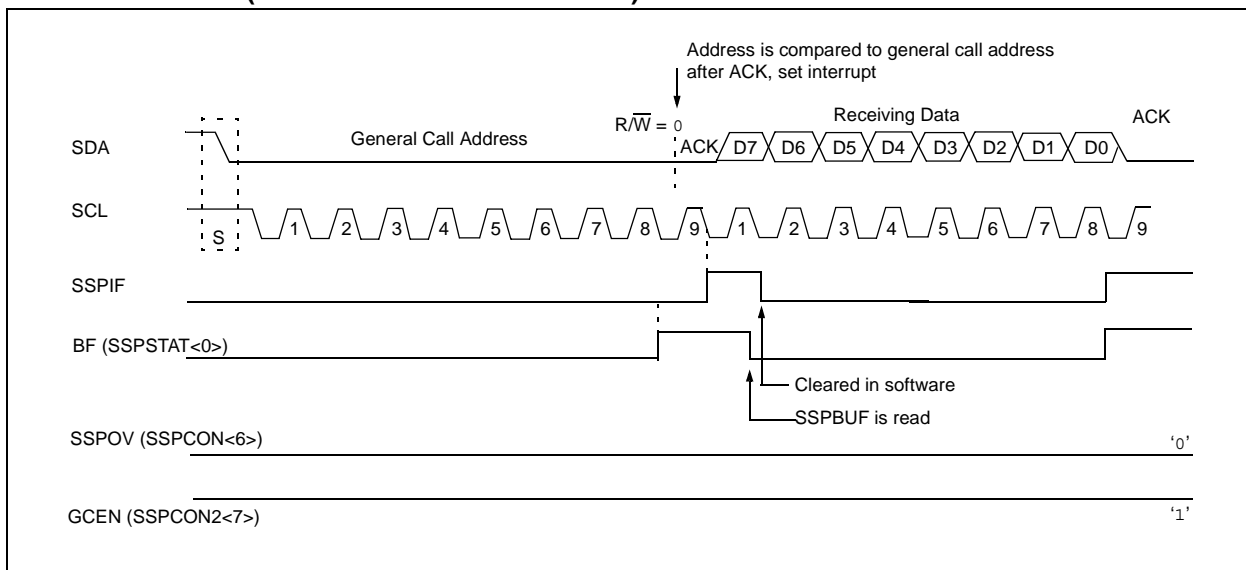
The general call address is recognized when the General Call Enable bit (GCEN) is enabled (SSPCON2<7> set). Following a Start bit detect, 8 bits are shifted into the SSPSR and the address is compared against the SSPADD. It is also compared to the general call address and fixed in hardware.

If the general call address matches, the SSPSR is transferred to the SSPBUF, the BF flag bit is set (eighth bit) and on the falling edge of the ninth bit ( $\overline{\text{ACK}}$  bit), the SSPIF interrupt flag bit is set.

When the interrupt is serviced, the source for the interrupt can be checked by reading the contents of the SSPBUF. The value can be used to determine if the address was device specific or a general call address.

In 10-bit mode, the SSPADD is required to be updated for the second half of the address to match and the UA bit is set (SSPSTAT<1>). If the general call address is sampled when the GCEN bit is set and while the slave is configured in 10-bit Address mode, then the second half of the address is not necessary, the UA bit will not be set and the slave will begin receiving data after the Acknowledge (Figure 10-15).

**FIGURE 10-15: SLAVE MODE GENERAL CALL ADDRESS SEQUENCE (7 OR 10-BIT ADDRESS MODE)**



## 10.4.6 MASTER MODE

Master mode is enabled by setting and clearing the appropriate SSPM bits in SSPCON and by setting the SSPEN bit. In Master mode, the SCL and SDA lines are manipulated by the MSSP hardware.

Master mode of operation is supported by 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 MSSP module is disabled. Control of the I<sup>2</sup>C bus may be taken when the P bit is set or the bus is Idle, with both the S and P bits clear.

In Firmware Controlled Master mode, user code conducts all I<sup>2</sup>C bus operations based on Start and Stop bit conditions.

Once Master mode is enabled, the user has six options:

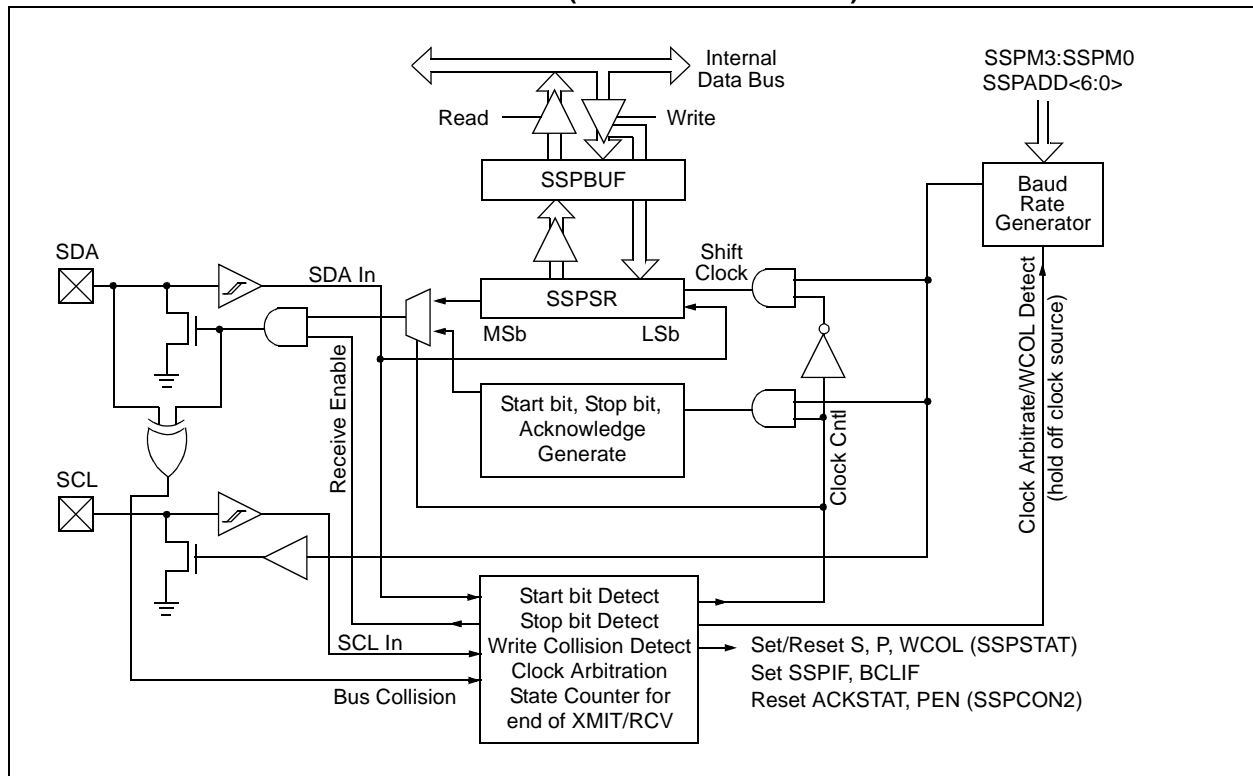
1. Assert a Start condition on SDA and SCL.
2. Assert a Repeated Start condition on SDA and SCL.
3. Write to the SSPBUF register, initiating transmission of data/address.
4. Configure the I<sup>2</sup>C port to receive data.
5. Generate an Acknowledge condition at the end of a received byte of data.
6. Generate a Stop condition on SDA and SCL.

**Note:** The MSSP module, when configured in I<sup>2</sup>C Master mode, does not allow queuing of events. For instance, the user is not allowed to initiate a Start condition and immediately write the SSPBUF register to initiate transmission before the Start condition is complete. In this case, the SSPBUF will not be written to and the WCOL bit will be set, indicating that a write to the SSPBUF did not occur.

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

- Start condition
- Stop condition
- Data transfer byte transmitted/received
- Acknowledge Transmit
- Repeated Start

**FIGURE 10-16: MSSP BLOCK DIAGRAM (I<sup>2</sup>C™ MASTER MODE)**



# PIC16F7X7

---

## 10.4.6.1 I<sup>2</sup>C Master Mode Operation

The master device generates all of the serial clock pulses and the Start and Stop conditions. A transfer is ended with a Stop condition or with a Repeated Start condition. Since the Repeated Start condition is also the beginning of the next serial transfer, the I<sup>2</sup>C bus will not be released.

In Master Transmitter mode, serial data is output through SDA while SCL outputs the serial clock. The first byte transmitted contains the slave address of the receiving device (7 bits) and the Read/Write (R/W) bit. In this case, the R/W bit will be logic '0'. Serial data is transmitted 8 bits at a time. After each byte is transmitted, an Acknowledge bit is received. Start and Stop conditions are output to indicate the beginning and the end of a serial transfer.

In Master Receive mode, the first byte transmitted contains the slave address of the transmitting device (7 bits) and the R/W bit. In this case, the R/W bit will be logic '1'. Thus, the first byte transmitted is a 7-bit slave address followed by a '1' to indicate a receive bit. Serial data is received via SDA, while SCL outputs the serial clock. Serial data is received 8 bits at a time. After each byte is received, an Acknowledge bit is transmitted. Start and Stop conditions indicate the beginning and end of transmission.

The Baud Rate Generator used for the SPI mode operation is used to set the SCL clock frequency for either 100 kHz, 400 kHz or 1 MHz I<sup>2</sup>C operation. See **Section 10.4.7 "Baud Rate Generator"** for more detail.

A typical transmit sequence would go as follows:

1. The user generates a Start condition by setting the Start enable bit, SEN (SSPCON2<0>).
2. SSPIF is set. The MSSP module will wait the required Start time before any other operation takes place.
3. The user loads the SSPBUF with the slave address to transmit.
4. Address is shifted out the SDA pin until all 8 bits are transmitted.
5. The MSSP module shifts in the ACK bit from the slave device and writes its value into the SSPCON2 register (SSPCON2<6>).
6. The MSSP module generates an interrupt at the end of the ninth clock cycle by setting the SSPIF bit.
7. The user loads the SSPBUF with eight bits of data.
8. Data is shifted out the SDA pin until all 8 bits are transmitted.
9. The MSSP module shifts in the ACK bit from the slave device and writes its value into the SSPCON2 register (SSPCON2<6>).
10. The MSSP module generates an interrupt at the end of the ninth clock cycle by setting the SSPIF bit.
11. The user generates a Stop condition by setting the Stop enable bit, PEN (SSPCON2<2>).
12. Interrupt is generated once the Stop condition is complete.



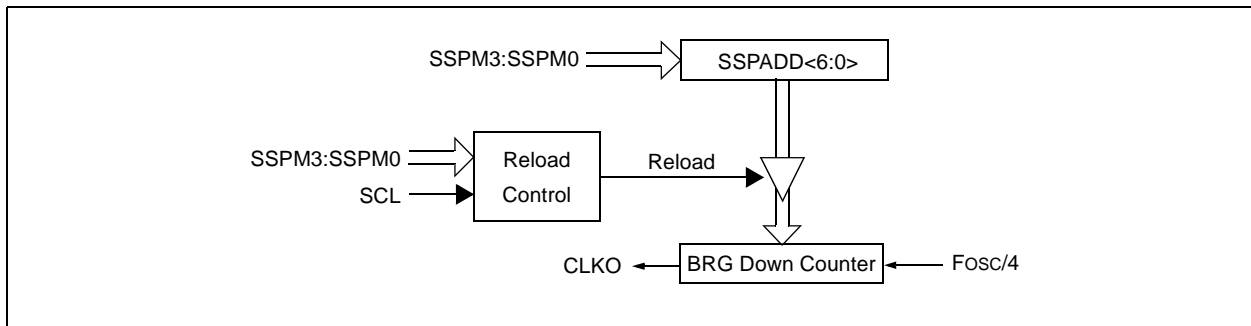
## 10.4.7 BAUD RATE GENERATOR

In I<sup>2</sup>C Master mode, the Baud Rate Generator (BRG) reload value is placed in the lower 7 bits of the SSPADD register (Figure 10-17). When a write occurs to SSPBUF, the Baud Rate Generator will automatically begin counting. The BRG counts down to 0 and stops until another reload has taken place. The BRG count is decremented twice per instruction cycle (T<sub>CY</sub>) on the Q2 and Q4 clocks. In I<sup>2</sup>C Master mode, the BRG is reloaded automatically.

Once the given operation is complete (i.e., transmission of the last data bit is followed by  $\overline{\text{ACK}}$ ), the internal clock will automatically stop counting and the SCL pin will remain in its last state.

Table 10-3 demonstrates clock rates based on instruction cycles and the BRG value loaded into SSPADD.

**FIGURE 10-17: BAUD RATE GENERATOR BLOCK DIAGRAM**



**TABLE 10-3: I<sup>2</sup>C™ CLOCK RATE w/BRG**

Fosc	Fcy	Fcy*2	BRG Value	Fscl (2 Rollovers of BRG)
40 MHz	10 MHz	20 MHz	18h	400 kHz <sup>(1)</sup>
40 MHz	10 MHz	20 MHz	1Fh	312.5 kHz
40 MHz	10 MHz	20 MHz	63h	100 kHz
16 MHz	4 MHz	8 MHz	09h	400 kHz <sup>(1)</sup>
16 MHz	4 MHz	8 MHz	0Ch	308 kHz
16 MHz	4 MHz	8 MHz	27h	100 kHz
4 MHz	1 MHz	2 MHz	02h	333 kHz <sup>(1)</sup>
4 MHz	1 MHz	2 MHz	09h	100 kHz
4 MHz	1 MHz	2 MHz	00h	1 MHz <sup>(1)</sup>

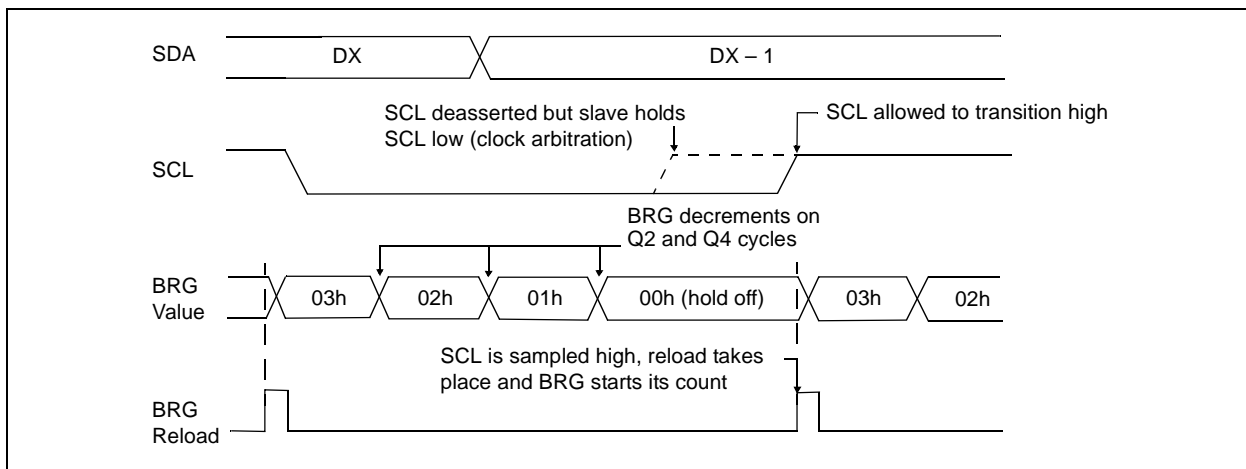
**Note 1:** The I<sup>2</sup>C™ interface does not conform to the 400 kHz I<sup>2</sup>C specification (which applies to rates greater than 100 kHz) in all details, but may be used with care where higher rates are required by the application.

# PIC16F7X7

## 10.4.7.1 Clock Arbitration

Clock arbitration occurs when the master, during any receive, transmit or Repeated Start/Stop condition, deasserts the SCL pin (SCL allowed to float high). When the SCL pin is allowed to float high, the Baud Rate Generator (BRG) is suspended from counting until the SCL pin is actually sampled high. When the SCL pin is sampled high, the Baud Rate Generator is reloaded with the contents of SSPADD<6:0> and begins counting. This ensures that the SCL high time will always be at least one BRG rollover count in the event that the clock is held low by an external device (Figure 10-18).

**FIGURE 10-18: BAUD RATE GENERATOR TIMING WITH CLOCK ARBITRATION**



## 10.4.8 I<sup>2</sup>C MASTER MODE START CONDITION TIMING

To initiate a Start condition, the user sets the Start Condition Enable bit, SEN (SSPCON2<0>). If the SDA and SCL pins are sampled high, the Baud Rate Generator is reloaded with the contents of SSPADD<6:0> and starts its count. If SCL and SDA are both sampled high when the Baud Rate Generator times out (TBRG), the SDA pin is driven low. The action of the SDA being driven low while SCL is high is the Start condition and causes the S bit (SSPSTAT<3>) to be set. Following this, the Baud Rate Generator is reloaded with the contents of SSPADD<6:0> and resumes its count. When the Baud Rate Generator times out (TBRG), the SEN bit (SSPCON2<0>) will be automatically cleared by hardware, the Baud Rate Generator is suspended, leaving the SDA line held low and the Start condition is complete.

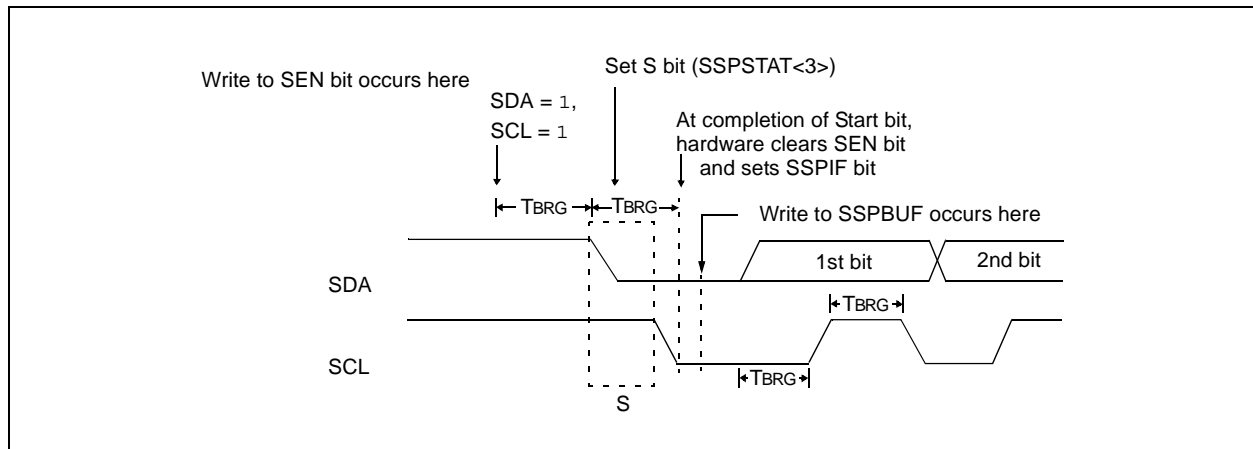
**Note:** If at the beginning of the Start condition, the SDA and SCL pins are already sampled low, or if during the Start condition, the SCL line is sampled low before the SDA line is driven low, a bus collision occurs, the Bus Collision Interrupt Flag, BCLIF, is set, the Start condition is aborted and the I<sup>2</sup>C module is reset into its Idle state.

### 10.4.8.1 WCOL Status Flag

If the user writes the SSPBUF when a Start sequence is in progress, the WCOL is set and the contents of the buffer are unchanged (the write doesn't occur).

**Note:** Because queueing of events is not allowed, writing to the lower 5 bits of SSPCON2 is disabled until the Start condition is complete.

**FIGURE 10-19: FIRST START BIT TIMING**



# PIC16F7X7

## 10.4.9 I<sup>2</sup>C MASTER MODE REPEATED START CONDITION TIMING

A Repeated Start condition occurs when the RSEN bit (SSPCON2<1>) is programmed high and the I<sup>2</sup>C logic module is in the Idle state. When the RSEN bit is set, the SCL pin is asserted low. When the SCL pin is sampled low, the Baud Rate Generator is loaded with the contents of SSPADD<5:0> and begins counting. The SDA pin is released (brought high) for one Baud Rate Generator count (TBRG). When the Baud Rate Generator times out, if SDA is sampled high, the SCL pin will be deasserted (brought high). When SCL is sampled high, the Baud Rate Generator is reloaded with the contents of SSPADD<6:0> and begins counting. SDA and SCL must be sampled high for one TBRG. This action is then followed by assertion of the SDA pin (SDA = 0) for one TBRG while SCL is high. Following this, the RSEN bit (SSPCON2<1>) will be automatically cleared and the Baud Rate Generator will not be reloaded, leaving the SDA pin held low. As soon as a Start condition is detected on the SDA and SCL pins, the S bit (SSPSTAT<3>) will be set. The SSPIF bit will not be set until the Baud Rate Generator has timed out.

**Note 1:** If RSEN is programmed while any other event is in progress, it will not take effect.

**2:** A bus collision during the Repeated Start condition occurs if:

- SDA is sampled low when SCL goes from low-to-high.
- SCL goes low before SDA is asserted low. This may indicate that another master is attempting to transmit a data '1'.

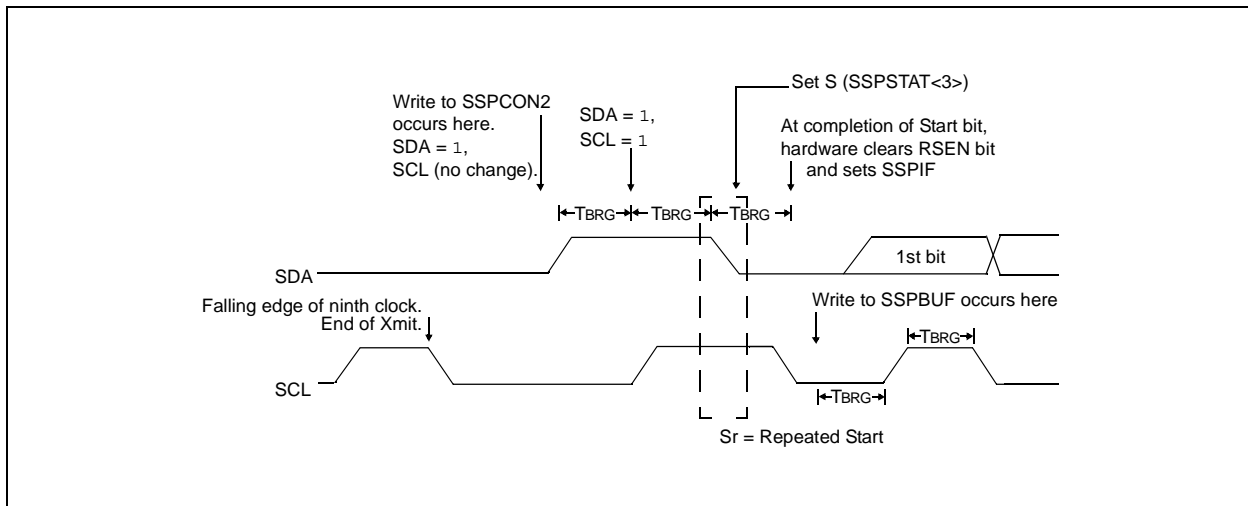
Immediately following the SSPIF bit getting set, the user may write the SSPBUF with the 7-bit address in 7-bit mode or the default first address in 10-bit mode. After the first eight bits are transmitted and an ACK is received, the user may then transmit an additional eight bits of address (10-bit mode) or eight bits of data (7-bit mode).

### 10.4.9.1 WCOL Status Flag

If the user writes the SSPBUF when a Repeated Start sequence is in progress, the WCOL is set and the contents of the buffer are unchanged (the write doesn't occur).

**Note:** Because queueing of events is not allowed, writing of the lower 5 bits of SSPCON2 is disabled until the Repeated Start condition is complete.

**FIGURE 10-20: REPEATED START CONDITION WAVEFORM**



## 10.4.10 I<sup>2</sup>C MASTER MODE TRANSMISSION

Transmission of a data byte, a 7-bit address or the other half of a 10-bit address, is accomplished by simply writing a value to the SSPBUF register. This action will set the Buffer Full flag bit, BF and allow the Baud Rate Generator to begin counting and start the next transmission. Each bit of address/data will be shifted out onto the SDA pin after the falling edge of SCL is asserted (see data hold time specification parameter #106). SCL is held low for one Baud Rate Generator rollover count (TBRG). Data should be valid before SCL is released high (see data setup time specification parameter #107). When the SCL pin is released high, it is held that way for TBRG. The data on the SDA pin must remain stable for that duration and some hold time after the next falling edge of SCL. After the eighth bit is shifted out (the falling edge of the eighth clock), the BF flag is cleared and the master releases SDA. This allows the slave device being addressed to respond with an  $\overline{\text{ACK}}$  bit, during the ninth bit time, if an address match occurred or if data was received properly. The status of ACK is written into the ACKDT bit on the falling edge of the ninth clock. If the master receives an Acknowledge, the Acknowledge Status bit, ACKSTAT, is cleared. If not, the bit is set. After the ninth clock, the SSPIF bit is set and the master clock (Baud Rate Generator) is suspended until the next data byte is loaded into the SSPBUF, leaving SCL low and SDA unchanged (Figure 10-21).

After the write to the SSPBUF, each bit of address will be shifted out on the falling edge of SCL until all seven address bits and the R/W bit are completed. On the falling edge of the eighth clock, the master will deassert the SDA pin, allowing the slave to respond with an Acknowledge. On the falling edge of the ninth clock, the master will sample the SDA pin to see if the address was recognized by a slave. The status of the ACK bit is loaded into the ACKSTAT status bit (SSPCON2<6>). Following the falling edge of the ninth clock transmission of the address, the SSPIF is set, The BF flag is cleared and the Baud Rate Generator is turned off until another write to the SSPBUF takes place, holding SCL low and allowing SDA to float.

### 10.4.10.1 BF Status Flag

In Transmit mode, the BF bit (SSPSTAT<0>) is set when the CPU writes to SSPBUF and is cleared when all 8 bits are shifted out.

### 10.4.10.2 WCOL Status Flag

If the user writes the SSPBUF when a transmit is already in progress (i.e., SSPSR is still shifting out a data byte), the WCOL is set and the contents of the buffer are unchanged (the write doesn't occur).

WCOL must be cleared in software.

### 10.4.10.3 ACKSTAT Status Flag

In Transmit mode, the ACKSTAT bit (SSPCON2<6>) is cleared when the slave has sent an Acknowledge ( $\text{ACK} = 0$ ) and is set when the slave does not Acknowledge ( $\text{ACK} = 1$ ). A slave sends an Acknowledge when it has recognized its address (including a general call) or when the slave has properly received its data.

## 10.4.11 I<sup>2</sup>C MASTER MODE RECEPTION

Master mode reception is enabled by programming the Receive Enable bit, RCEN (SSPCON2<3>).

**Note:** The MSSP module must be in an Idle state before the RCEN bit is set or the RCEN bit will be disregarded.

The Baud Rate Generator begins counting and on each rollover, the state of the SCL pin changes (high-to-low/low-to-high) and data is shifted into the SSPSR. After the falling edge of the eighth clock, the receive enable flag is automatically cleared, the contents of the SSPSR are loaded into the SSPBUF, the BF flag bit is set, the SSPIF flag bit is set and the Baud Rate Generator is suspended from counting, holding SCL low. The MSSP is now in Idle state, awaiting the next command. When the buffer is read by the CPU, the BF flag bit is automatically cleared. The user can then send an Acknowledge bit at the end of reception by setting the Acknowledge Sequence Enable bit, ACKEN (SSPCON2<4>).

### 10.4.11.1 BF Status Flag

In receive operation, the BF bit is set when an address or data byte is loaded into SSPBUF from SSPSR. It is cleared when the SSPBUF register is read.

### 10.4.11.2 SSPOV Status Flag

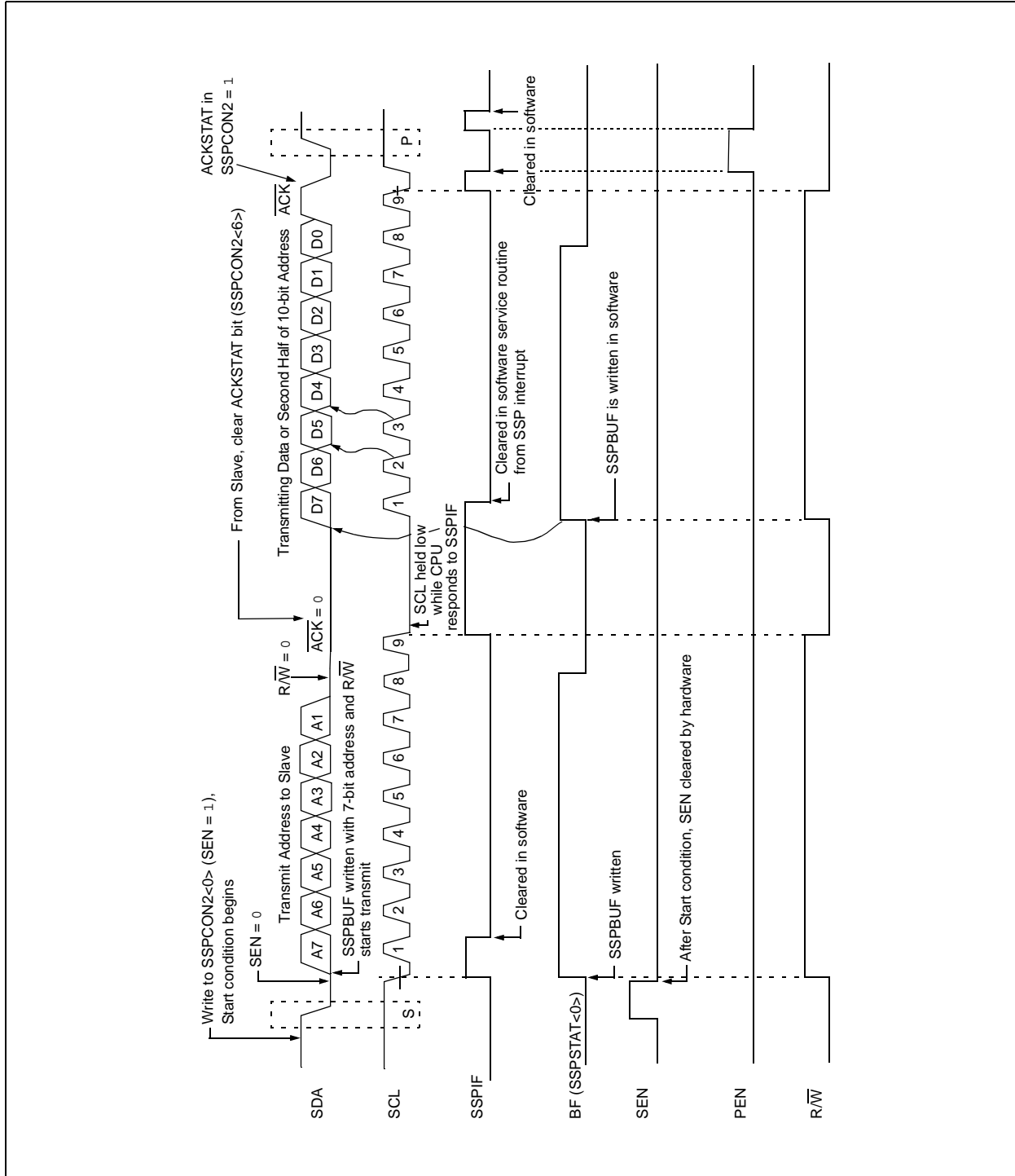
In receive operation, the SSPOV bit is set when 8 bits are received into the SSPSR and the BF flag bit is already set from a previous reception.

### 10.4.11.3 WCOL Status Flag

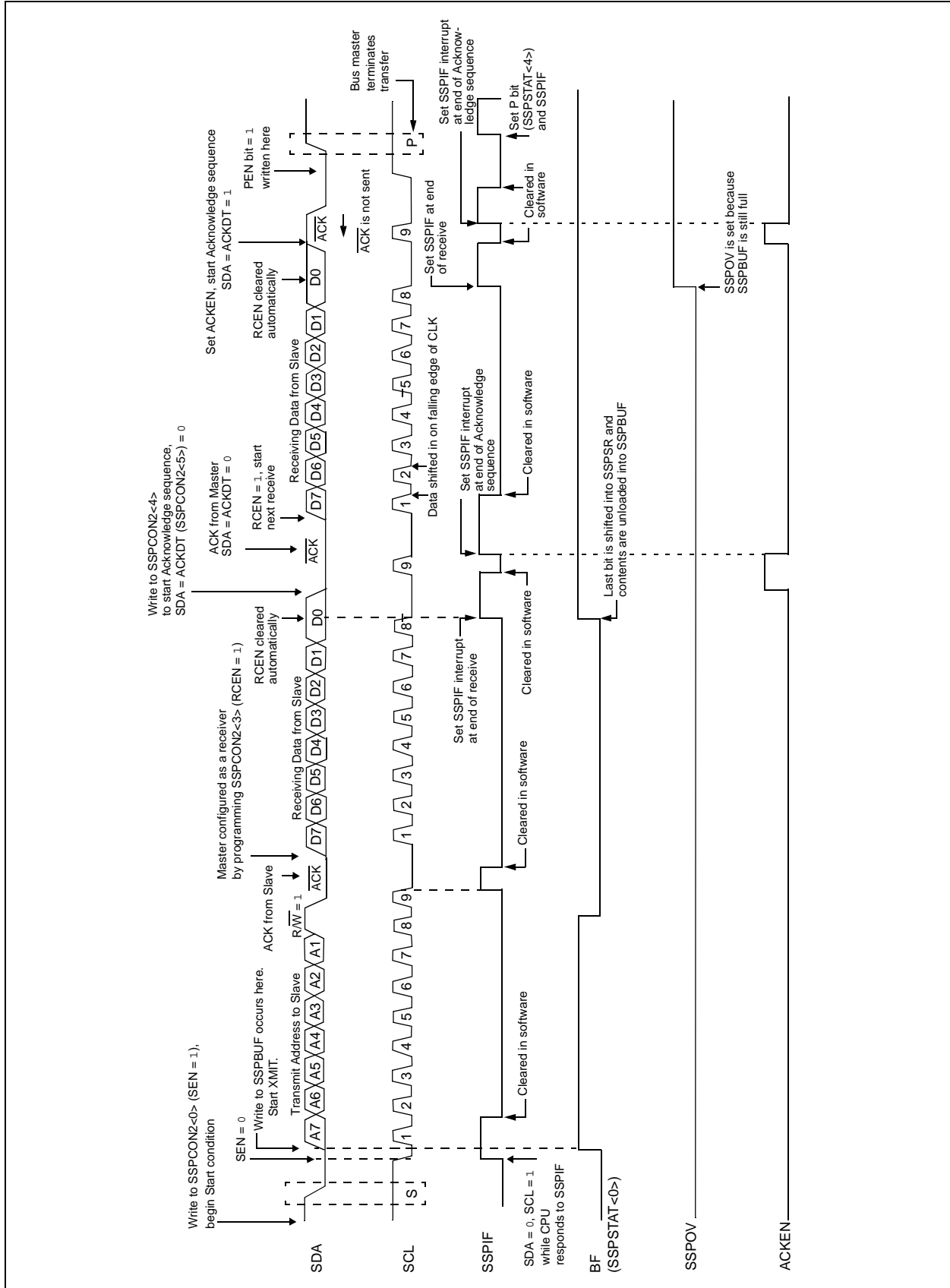
If the user writes the SSPBUF when a receive is already in progress (i.e., SSPSR is still shifting in a data byte), the WCOL bit is set and the contents of the buffer are unchanged (the write doesn't occur).

# PIC16F7X7

FIGURE 10-21: I<sup>2</sup>C™ MASTER MODE WAVEFORM (TRANSMISSION, 7 OR 10-BIT ADDRESS)



**FIGURE 10-22: I<sup>2</sup>C™ MASTER MODE WAVEFORM (RECEPTION, 7-BIT ADDRESS)**



# PIC16F7X7

## 10.4.12 ACKNOWLEDGE SEQUENCE TIMING

An Acknowledge sequence is enabled by setting the Acknowledge Sequence Enable bit, ACKEN (SSPCON2<4>). When this bit is set, the SCL pin is pulled low and the contents of the Acknowledge data bit are presented on the SDA pin. If the user wishes to generate an Acknowledge, then the ACKDT bit should be cleared. If not, the user should set the ACKDT bit before starting an Acknowledge sequence. The Baud Rate Generator then counts for one rollover period (TBRG) and the SCL pin is deasserted (pulled high). When the SCL pin is sampled high (clock arbitration), the Baud Rate Generator counts for TBRG. The SCL pin is then pulled low. Following this, the ACKEN bit is automatically cleared, the Baud Rate Generator is turned off and the MSSP module then goes into Idle mode (Figure 10-23).

### 10.4.12.1 WCOL Status Flag

If the user writes the SSPBUF when an Acknowledge sequence is in progress, then WCOL is set and the contents of the buffer are unchanged (the write doesn't occur).

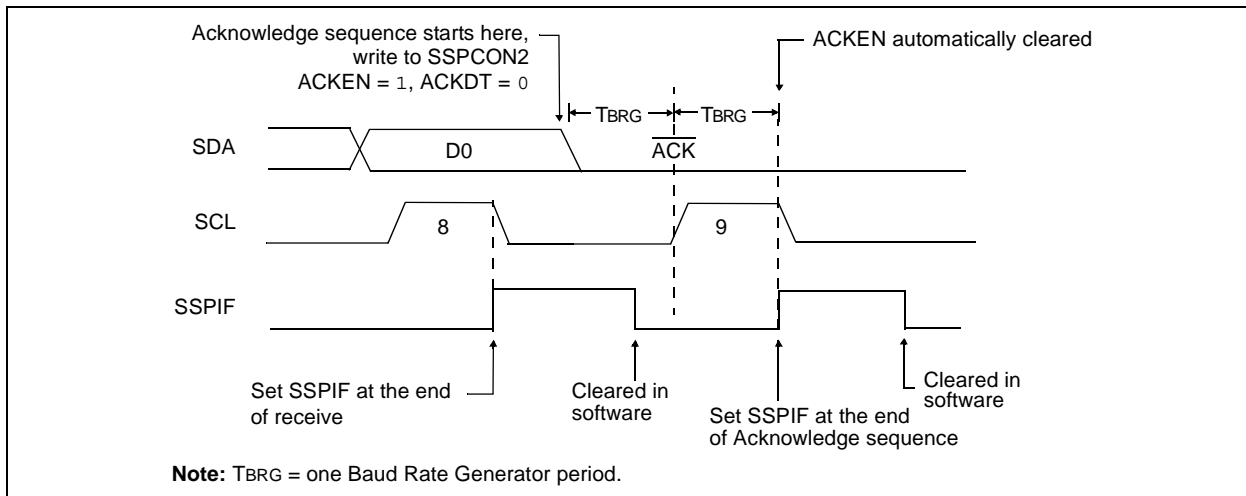
## 10.4.13 STOP CONDITION TIMING

A Stop bit is asserted on the SDA pin at the end of a receive/transmit by setting the Stop Sequence Enable bit, PEN (SSPCON2<2>). At the end of a receive/transmit, the SCL line is held low after the falling edge of the ninth clock. When the PEN bit is set, the master will assert the SDA line low. When the SDA line is sampled low, the Baud Rate Generator is reloaded and counts down to '0'. When the Baud Rate Generator times out, the SCL pin will be brought high and one TBRG (Baud Rate Generator rollover count) later, the SDA pin will be deasserted. When the SDA pin is sampled high while SCL is high, the P bit (SSPSTAT<4>) is set. A TBRG later, the PEN bit is cleared and the SSPIF bit is set (Figure 10-24).

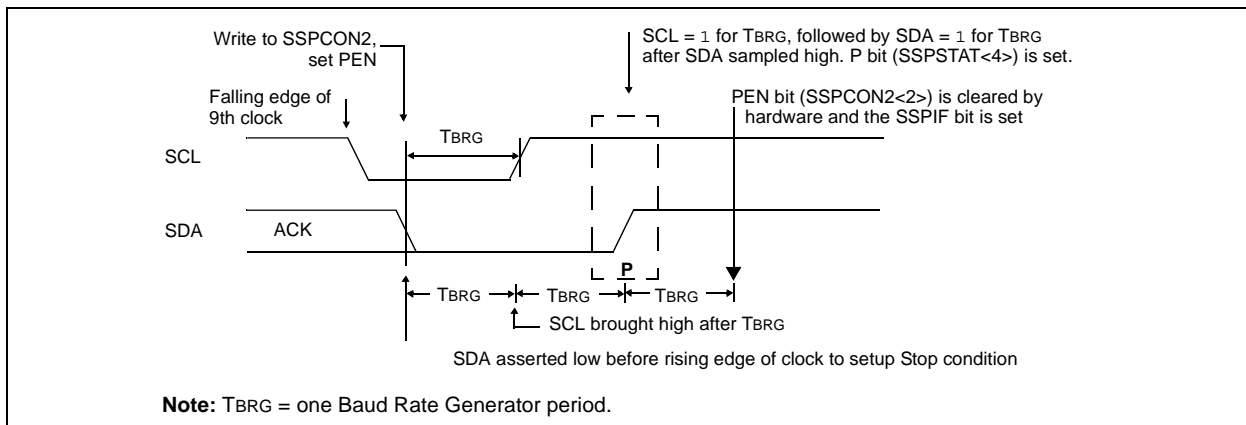
### 10.4.13.1 WCOL Status Flag

If the user writes the SSPBUF when a Stop sequence is in progress, then the WCOL bit is set and the contents of the buffer are unchanged (the write doesn't occur).

**FIGURE 10-23: ACKNOWLEDGE SEQUENCE WAVEFORM**



**FIGURE 10-24: STOP CONDITION RECEIVE OR TRANSMIT MODE**





## 10.4.14 SLEEP OPERATION

While in Sleep mode, the I<sup>2</sup>C module can receive addresses or data and when an address match or complete byte transfer occurs, wake the processor from Sleep (if the MSSP interrupt is enabled).

## 10.4.15 EFFECT OF A RESET

A Reset disables the MSSP module and terminates the current transfer.

## 10.4.16 MULTI-MASTER MODE

In Multi-Master mode, 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 MSSP module is disabled. Control of the I<sup>2</sup>C bus may be taken when the P bit (SSPSTAT<4>) is set or the bus is Idle, with 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 operation, the SDA line must be monitored for arbitration to see if the signal level is at the expected output level. This check is performed in hardware with the result placed in the BCLIF bit.

The states where arbitration can be lost are:

- Address Transfer
- Data Transfer
- A Start Condition
- A Repeated Start Condition
- An Acknowledge Condition

## 10.4.17 MULTI-MASTER COMMUNICATION, BUS COLLISION AND BUS ARBITRATION

Multi-Master mode support is achieved by bus arbitration. When the master outputs address/data bits onto the SDA pin, arbitration takes place when the master outputs a '1' on SDA by letting SDA float high and another master asserts a '0'. When the SCL pin floats high, data should be stable. If the expected data on SDA is a '1' and the data sampled on the SDA pin = 0, then a bus collision has taken place. The master will set the Bus Collision Interrupt Flag, BCLIF and reset the I<sup>2</sup>C port to its Idle state (Figure 10-25).

If a transmit was in progress when the bus collision occurred, the transmission is halted, the BF flag is cleared, the SDA and SCL lines are deasserted and the SSPBUF can be written to. When the user services the bus collision Interrupt Service Routine and if the I<sup>2</sup>C bus is free, the user can resume communication by asserting a Start condition.

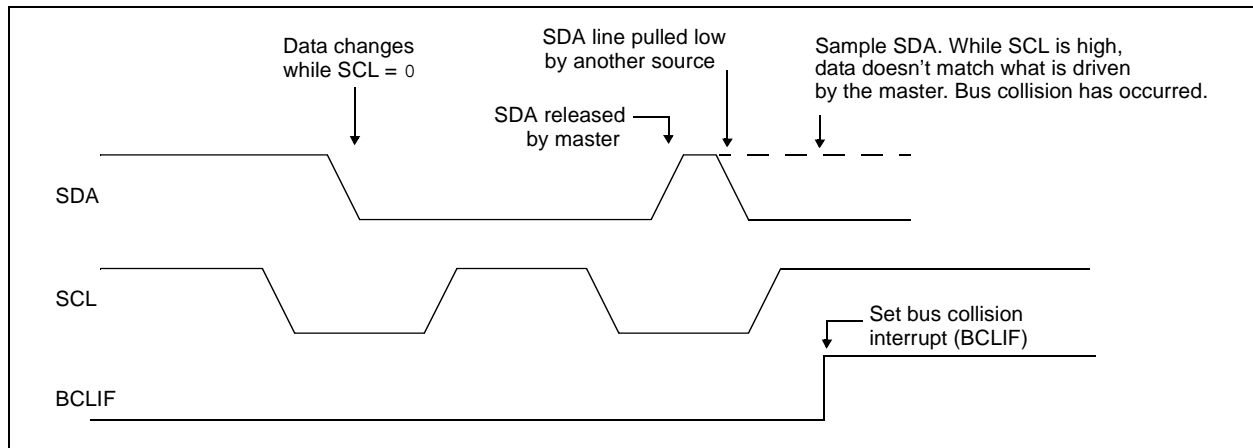
If a Start, Repeated Start, Stop or Acknowledge condition was in progress when the bus collision occurred, the condition is aborted, the SDA and SCL lines are deasserted and the respective control bits in the SSPCON2 register are cleared. When the user services the bus collision Interrupt Service Routine and if the I<sup>2</sup>C bus is free, the user can resume communication by asserting a Start condition.

The master will continue to monitor the SDA and SCL pins. If a Stop condition occurs, the SSPIF bit will be set.

A write to the SSPBUF will start the transmission of data at the first data bit, regardless of where the transmitter left off when the bus collision occurred.

In Multi-Master mode, the interrupt generation on the detection of Start and Stop conditions allows the determination of when the bus is free. Control of the I<sup>2</sup>C bus can be taken when the P bit is set in the SSPSTAT register or the bus is Idle and the S and P bits are cleared.

**FIGURE 10-25: BUS COLLISION TIMING FOR TRANSMIT AND ACKNOWLEDGE**



# PIC16F7X7

## 10.4.17.1 Bus Collision During a Start Condition

During a Start condition, a bus collision occurs if:

- SDA or SCL are sampled low at the beginning of the Start condition (Figure 10-26).
- SCL is sampled low before SDA is asserted low (Figure 10-27).

During a Start condition, both the SDA and the SCL pins are monitored.

If the SDA pin is already low, or the SCL pin is already low, then all of the following occur:

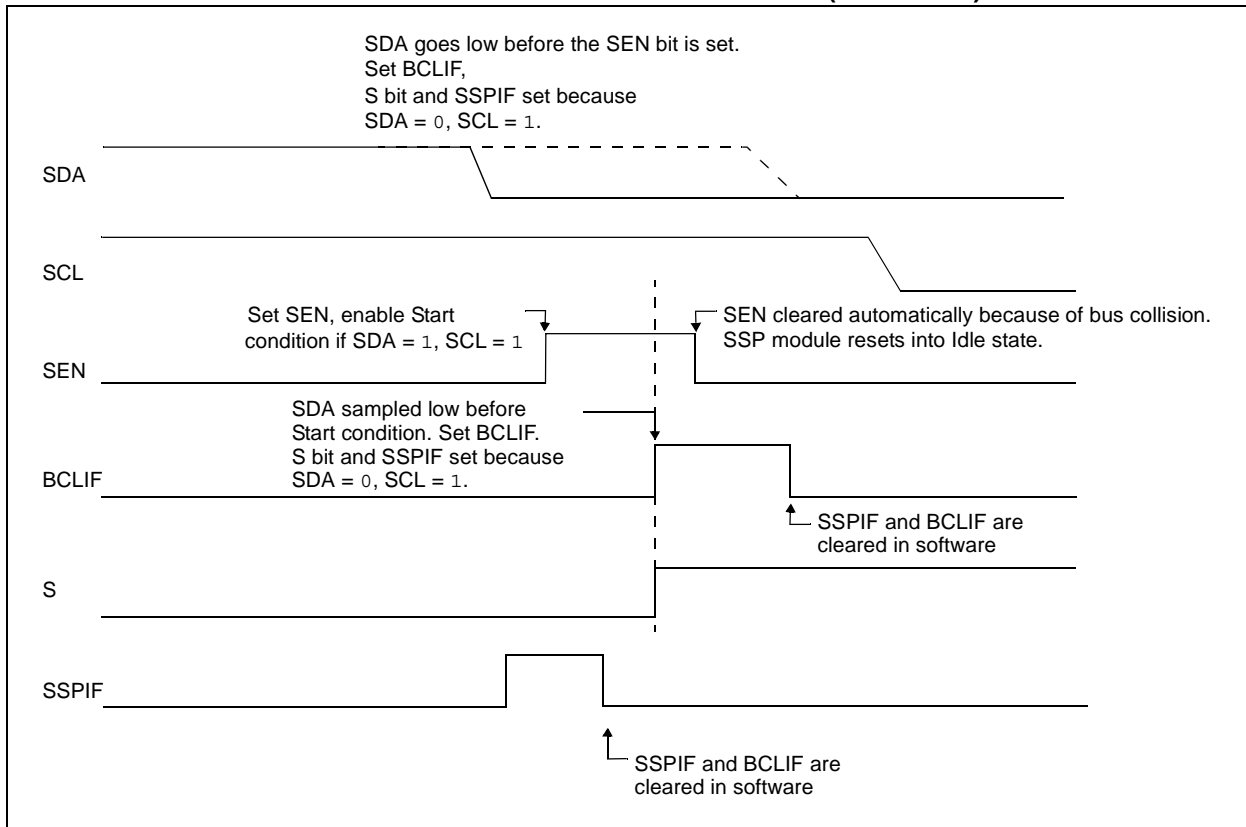
- the Start condition is aborted,
- the BCLIF flag is set and
- the MSSP module is reset to its Idle state (Figure 10-26).

The Start condition begins with the SDA and SCL pins deasserted. When the SDA pin is sampled high, the Baud Rate Generator is loaded from  $SSPADD<6:0>$  and counts down to 0. If the SCL pin is sampled low while SDA is high, a bus collision occurs because it is assumed that another master is attempting to drive a data '1' during the Start condition.

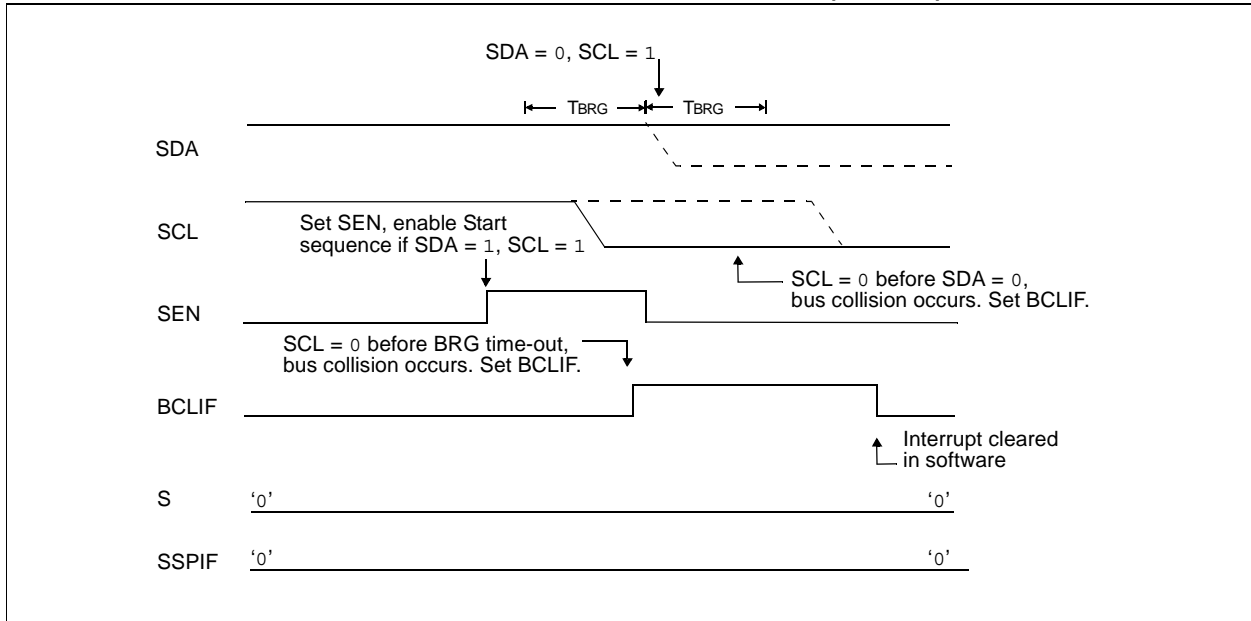
If the SDA pin is sampled low during this count, the BRG is reset and the SDA line is asserted early (Figure 10-28). If, however, a '1' is sampled on the SDA pin, the SDA pin is asserted low at the end of the BRG count. The Baud Rate Generator is then reloaded and counts down to 0 and during this time, if the SCL pin is sampled as '0', a bus collision does not occur. At the end of the BRG count, the SCL pin is asserted low.

**Note:** The reason that bus collision is not a factor during a Start condition is that no two bus masters can assert a Start condition at the exact same time. Therefore, one master will always assert SDA before the other. This condition does not cause a bus collision because the two masters must be allowed to arbitrate the first address following the Start condition. If the address is the same, arbitration must be allowed to continue into the data portion, Repeated Start or Stop conditions.

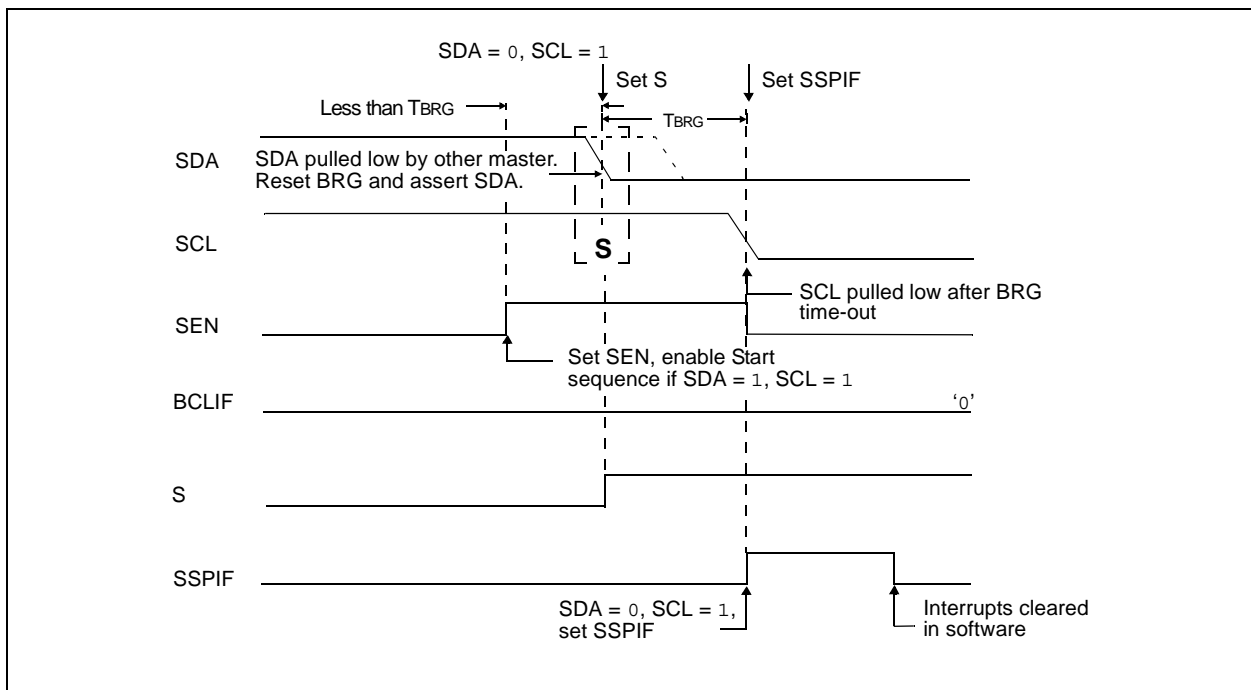
**FIGURE 10-26: BUS COLLISION DURING START CONDITION (SDA ONLY)**



**FIGURE 10-27: BUS COLLISION DURING START CONDITION (SCL = 0)**



**FIGURE 10-28: BRG RESET DUE TO SDA ARBITRATION DURING START CONDITION**



# PIC16F7X7

## 10.4.17.2 Bus Collision During a Repeated Start Condition

During a Repeated Start condition, a bus collision occurs if:

- A low level is sampled on SDA when SCL goes from low level to high level.
- SCL goes low before SDA is asserted low, indicating that another master is attempting to transmit a data '1'.

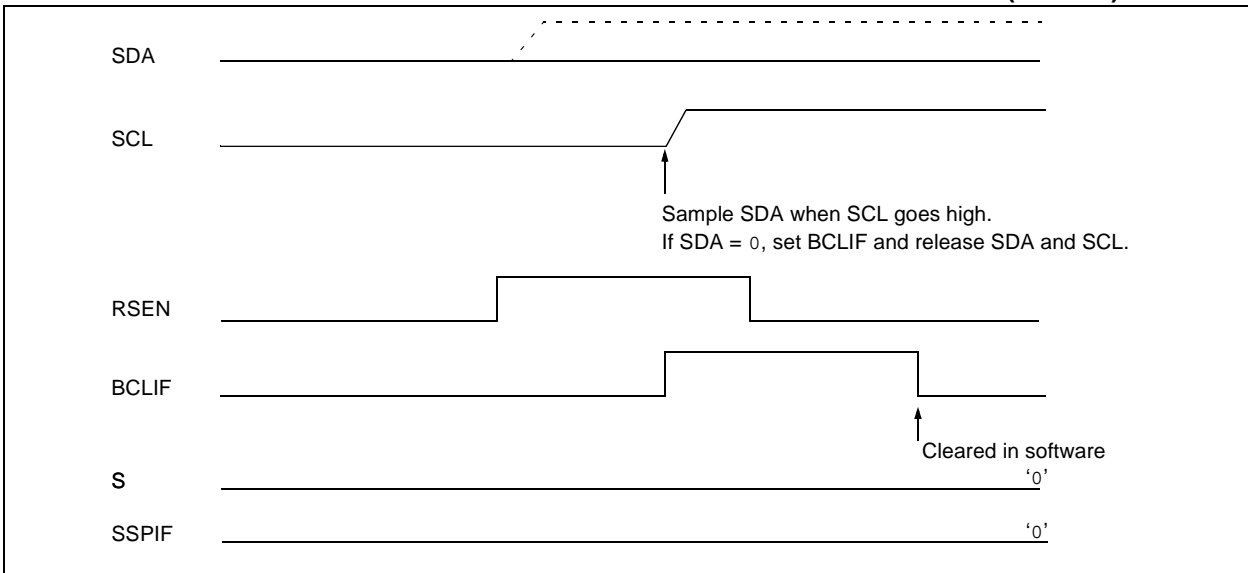
When the user deasserts SDA and the pin is allowed to float high, the BRG is loaded with SSPADD<6:0> and counts down to 0. The SCL pin is then deasserted and when sampled high, the SDA pin is sampled.

If SDA is low, a bus collision has occurred (i.e., another master is attempting to transmit a data '0', see Figure 10-29). If SDA is sampled high, the BRG is reloaded and begins counting. If SDA goes from high-to-low before the BRG times out, no bus collision occurs because no two masters can assert SDA at exactly the same time.

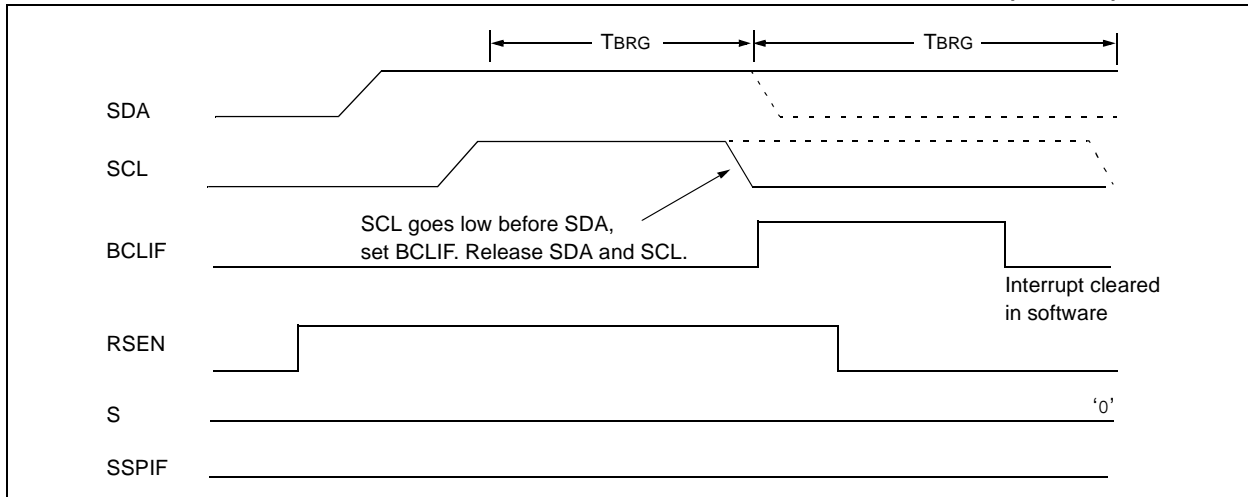
If SCL goes from high-to-low before the BRG times out and SDA has not already been asserted, a bus collision occurs. In this case, another master is attempting to transmit a data '1' during the Repeated Start condition (Figure 10-30).

If at the end of the BRG time-out, both SCL and SDA are still high, the SDA pin is driven low and the BRG is reloaded and begins counting. At the end of the count, regardless of the status of the SCL pin, the SCL pin is driven low and the Repeated Start condition is complete.

**FIGURE 10-29: BUS COLLISION DURING A REPEATED START CONDITION (CASE 1)**



**FIGURE 10-30: BUS COLLISION DURING A REPEATED START CONDITION (CASE 2)**



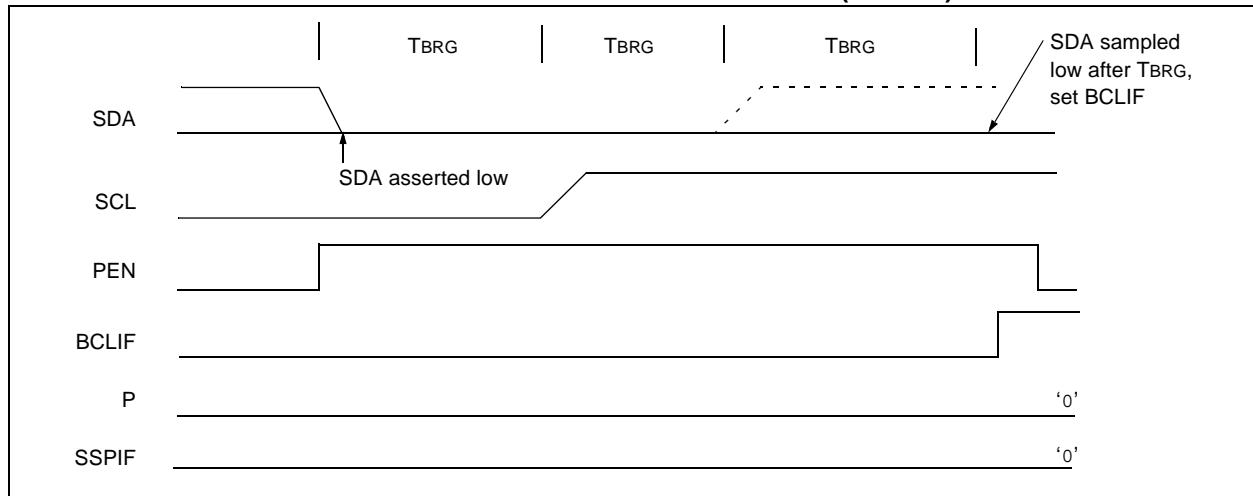
## 10.4.17.3 Bus Collision During a Stop Condition

Bus collision occurs during a Stop condition if:

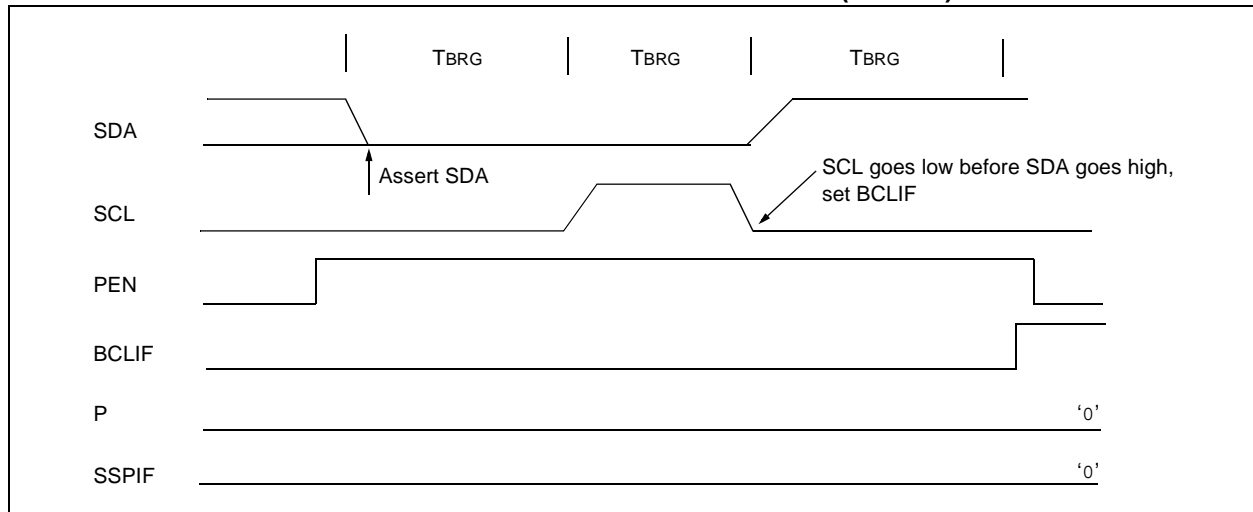
- After the SDA pin has been deasserted and allowed to float high, SDA is sampled low after the BRG has timed out.
- After the SCL pin is deasserted, SCL is sampled low before SDA goes high.

The Stop condition begins with SDA asserted low. When SDA is sampled low, the SCL pin is allowed to float. When the pin is sampled high (clock arbitration), the Baud Rate Generator is loaded with SSPADD<6:0> and counts down to 0. After the BRG times out, SDA is sampled. If SDA is sampled low, a bus collision has occurred. This is due to another master attempting to drive a data '0' (Figure 10-31). If the SCL pin is sampled low before SDA is allowed to float high, a bus collision occurs. This is another case of another master attempting to drive a data '0' (Figure 10-32).

**FIGURE 10-31: BUS COLLISION DURING A STOP CONDITION (CASE 1)**



**FIGURE 10-32: BUS COLLISION DURING A STOP CONDITION (CASE 2)**



# PIC16F7X7

---

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 TRISC<7:6> have to be set in order to configure pins RC6/TX/CK and RC7/RX/DT as the 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

# PIC16F7X7

## 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 RC7/RX/DT and RC6/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 receiving 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  $F_{osc}/(16(X + 1))$  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 SAMPLING

The data on the RC7/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 = $F_{osc}/(64(X + 1))$	Baud Rate = $F_{osc}/(16(X + 1))$
1	(Synchronous) Baud Rate = $F_{osc}/(4(X + 1))$	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.

# PIC16F7X7

**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

# PIC16F7X7

## 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.

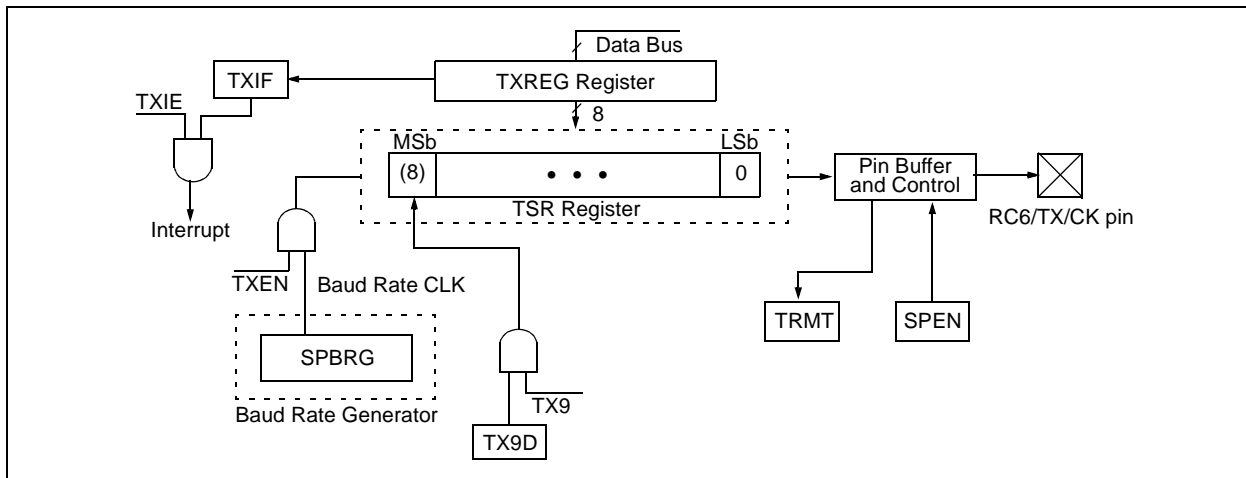
**Note 1:** The TSR register is not mapped in data memory so it is not available to the user.

**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 RC6/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.

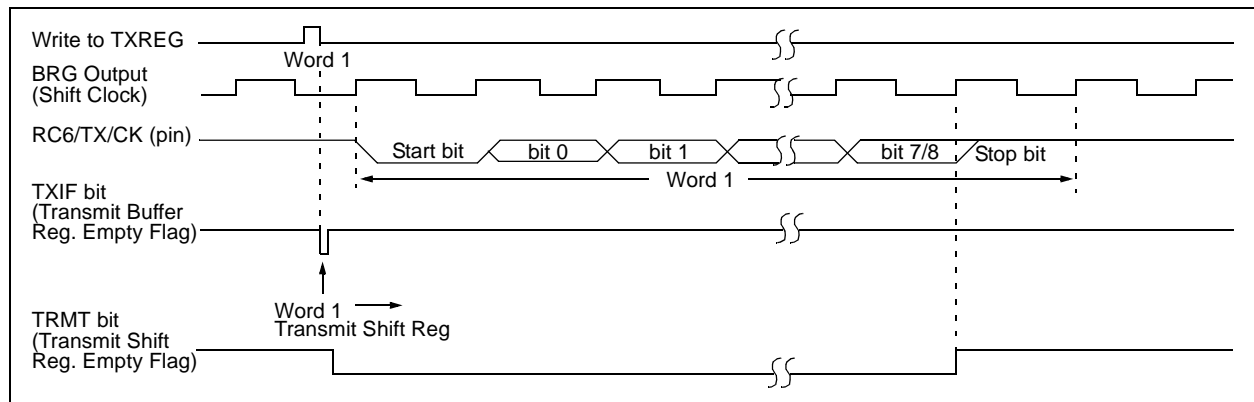
FIGURE 11-1: AUSART TRANSMIT BLOCK DIAGRAM



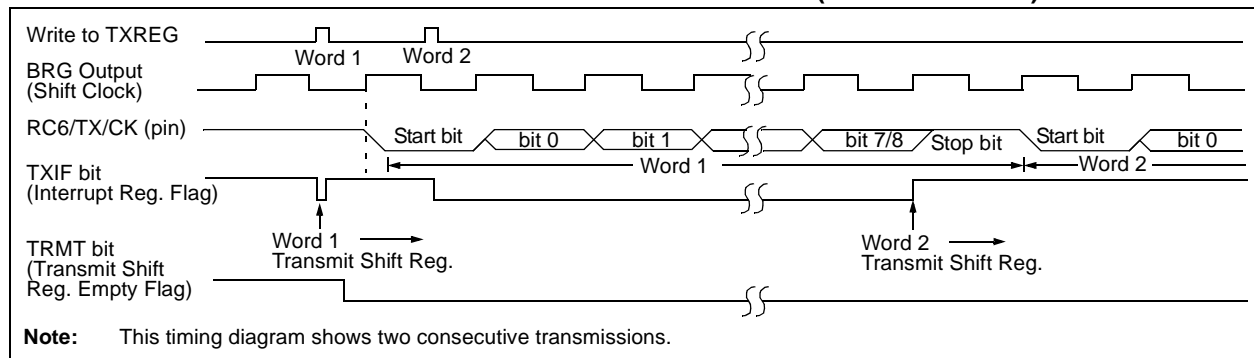
When setting up an Asynchronous Transmission, follow these steps:

1. Initialize the SPBRG register for the appropriate baud rate. If a high-speed baud rate is desired, set bit BRGH (see **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	PSPIF <sup>(1)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 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	PSPIE <sup>(1)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 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:** Bits PSPIE and PSPIF are reserved on 28-pin devices; always maintain these bits clear.

# PIC16F7X7

## 11.2.2 AUSART ASYNCHRONOUS RECEIVER

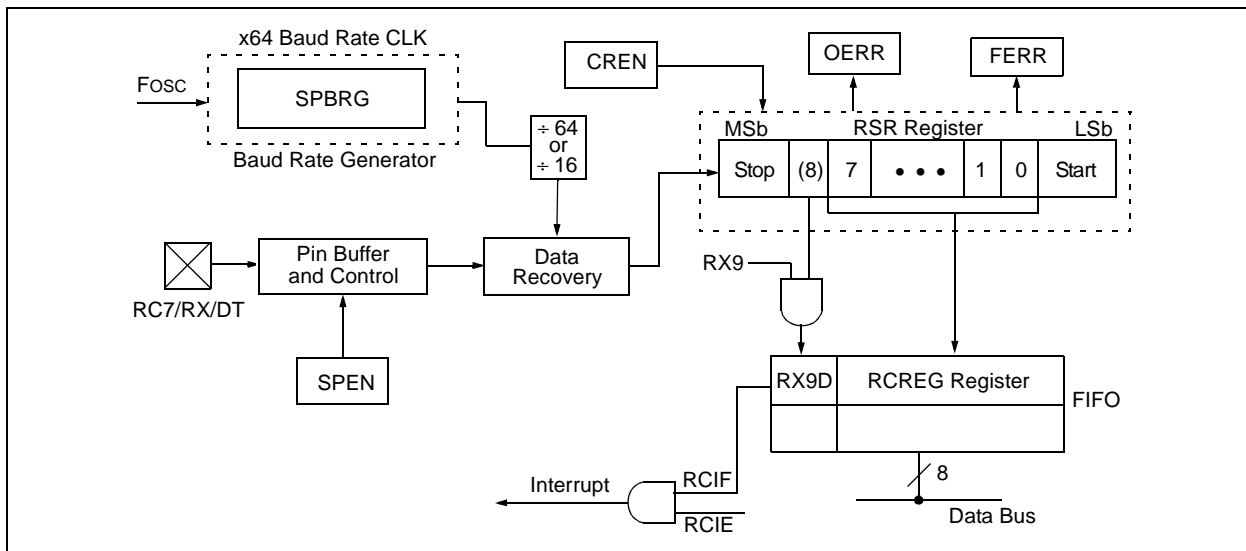
The receiver block diagram is shown in Figure 11-4. The data is received on the RC7/RX/DT pin and drives the data recovery block. The data recovery block is actually a high-speed shifter, operating at x16 times the baud rate; whereas, the main receive serial shifter operates at the bit rate or at Fosc.

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

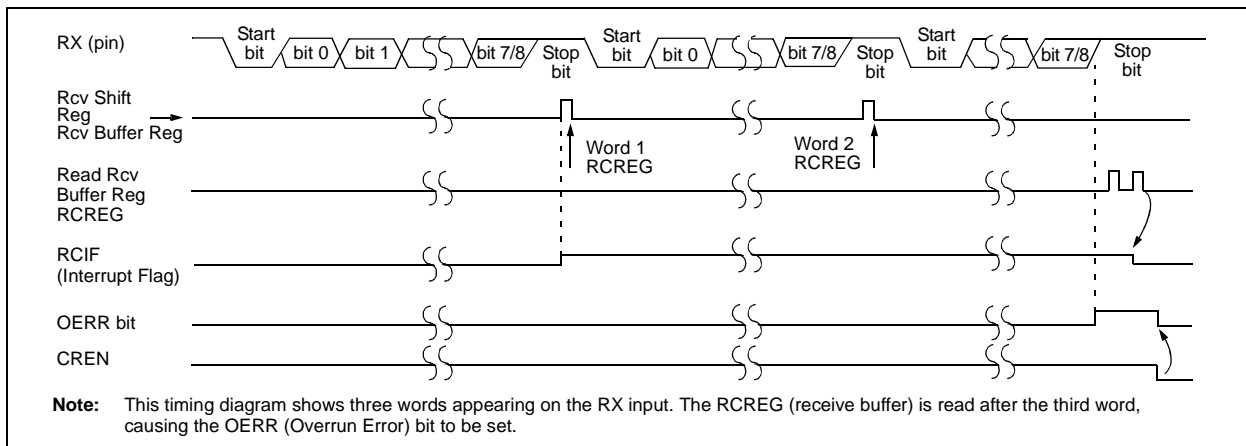
The heart of the receiver is the Receive (Serial) Shift Register (RSR). After sampling the Stop bit, the received data in the RSR is transferred to the RCREG register (if it is empty). If the transfer is complete, flag bit, RCIF (PIR1<5>), is set. The actual interrupt can be enabled/disabled by setting/clearing 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 (see **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	PSPIF <sup>(1)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 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	PSPIE <sup>(1)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 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:** Bits PSPIE and PSPIF are reserved on 28-pin devices; always maintain these bits clear.

# PIC16F7X7

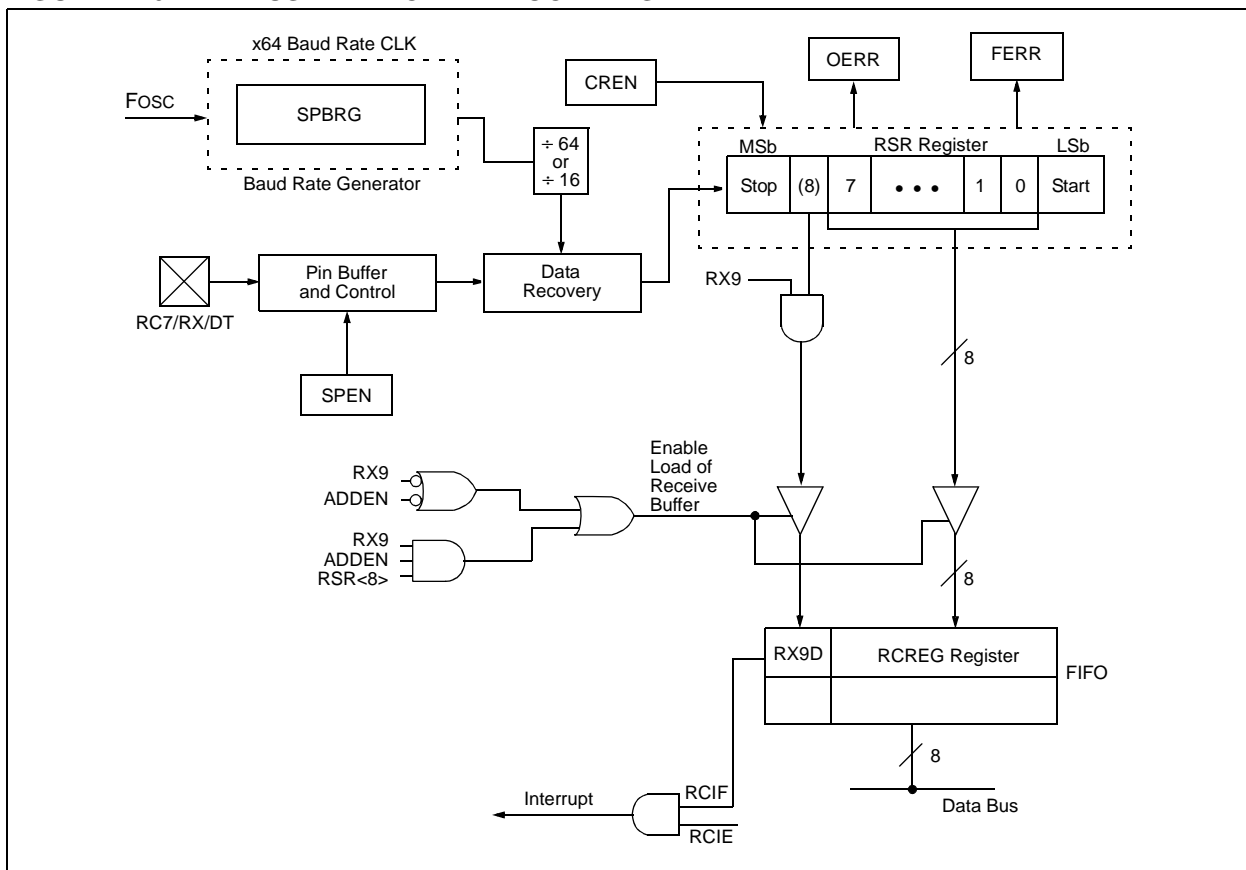
## 11.2.3 SETTING UP 9-BIT MODE WITH ADDRESS DETECT

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.

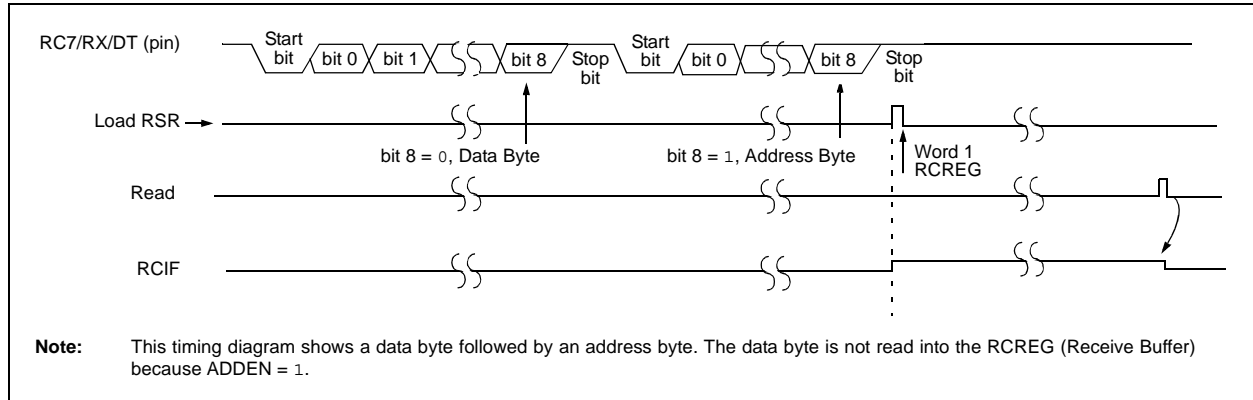
- Flag bit RCIF will be set when reception is complete and an interrupt will be generated if enable bit RCIE was set.
- Read the RCSTA register to get the ninth bit and determine if any error occurred during reception.
- Read the 8-bit received data by reading the RCREG register to determine if the device 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**

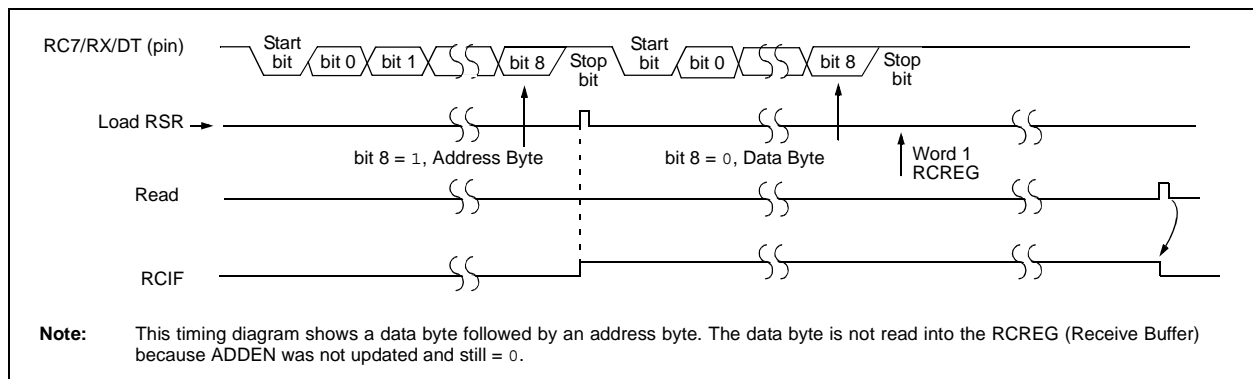




**FIGURE 11-7: ASYNCHRONOUS RECEPTION WITH ADDRESS DETECT**



**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	PSPIF <sup>(1)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
1Ah	RCREG	AUSART Receive Register								0000 0000	0000 0000
8Ch	PIE1	PSPIE <sup>(1)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 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:** Bits PSPIE and PSPIF are reserved on 28-pin devices; always maintain these bits clear.

## 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 RC6/TX/CK and RC7/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) and 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” value to 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 (see **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.

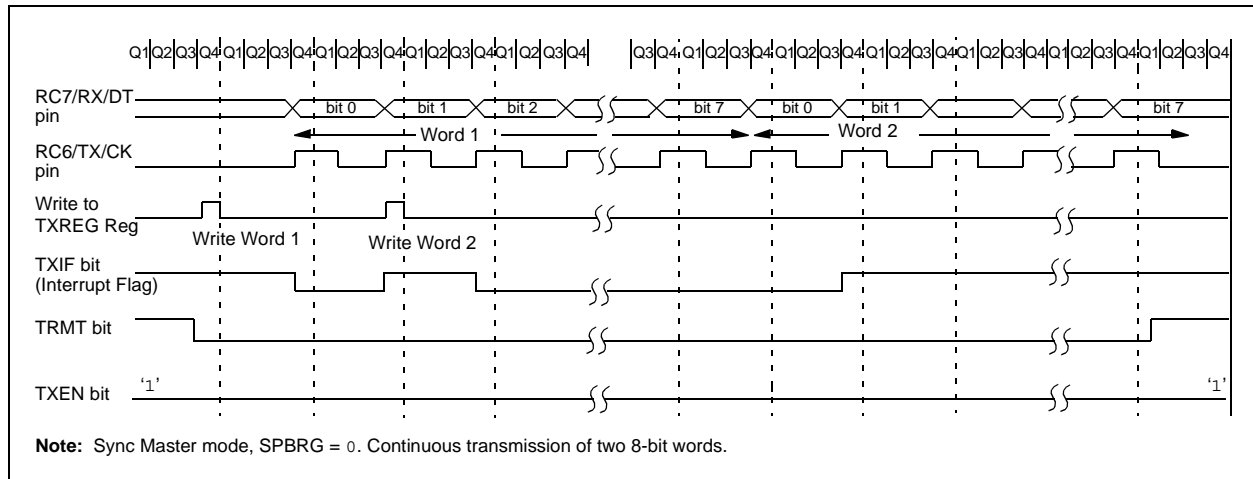
**TABLE 11-10: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER TRANSMISSION**

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

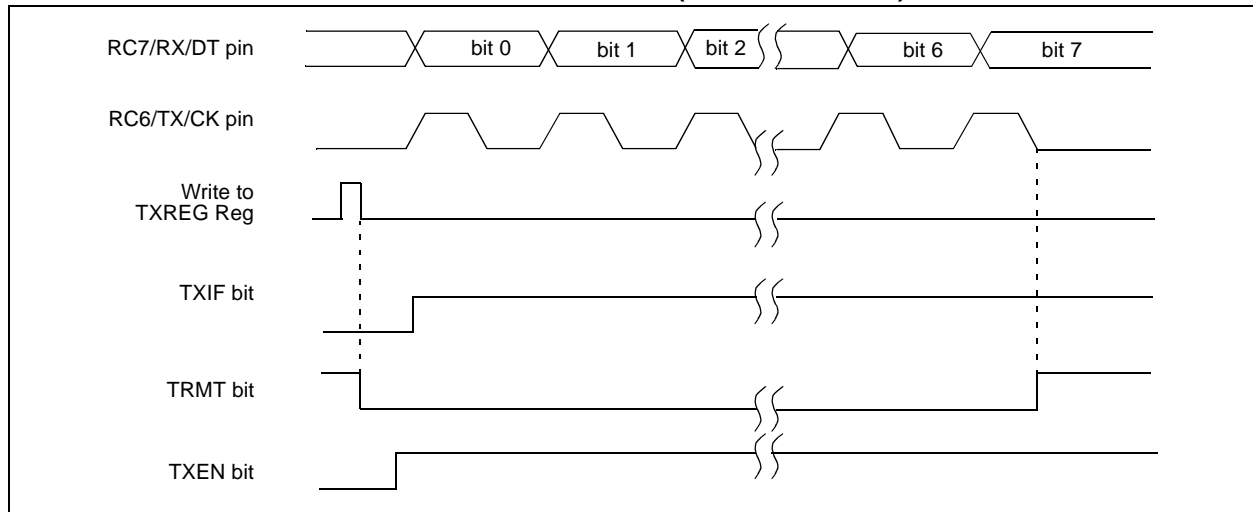
**Legend:** x = unknown, — = unimplemented, read as '0'. Shaded cells are not used for synchronous master transmission.

**Note 1:** Bits PSPIE and PSPIF are reserved on 28-pin devices; always maintain these bits clear.

**FIGURE 11-9: SYNCHRONOUS TRANSMISSION**



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



# PIC16F7X7

## 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 RC7/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 (see **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.

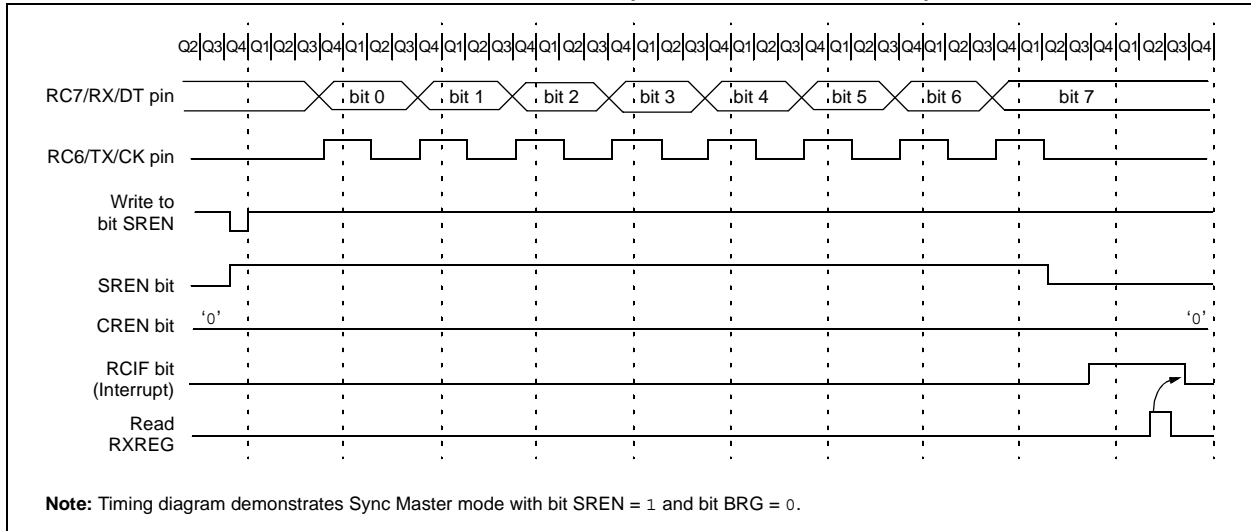
**TABLE 11-11: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER RECEPTION**

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	PSPIF <sup>(1)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
1Ah	RCREG	AUSART Receive Register								0000 0000	0000 0000
8Ch	PIE1	PSPIE <sup>(1)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 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:** Bits PSPIE and PSPIF are reserved on 28-pin devices; always maintain these bits clear.

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



# PIC16F7X7

## 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 RC6/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 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	PSPIF <sup>(1)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 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	PSPIE <sup>(1)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 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:** Bits PSPIE and PSPIF are reserved on 28-pin devices; always maintain these bits clear.

## 11.4.2 AUSART SYNCHRONOUS SLAVE RECEPTION

The operation of the Synchronous Master and Slave modes is identical, except in the case of 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	PSPIF <sup>(1)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 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	PSPIE <sup>(1)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 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:** Bits PSPIE and PSPIF are reserved on 28-pin devices, always maintain these bits clear.

# PIC16F7X7

---

NOTES:



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

The Analog-to-Digital (A/D) Converter module has 11 inputs for the PIC16F737 and PIC16F767 devices and 14 for the PIC16F747 AND PIC16F777 devices.

The A/D converter allows conversion of an analog input signal to a corresponding 10-bit digital number.

A new feature for the A/D converter is the addition of programmable acquisition time. This feature allows the user to select a new channel for conversion and to set the  $\overline{\text{GO/DONE}}$  bit immediately. When the  $\overline{\text{GO/DONE}}$  bit is set, the selected channel is sampled for the programmed acquisition time before a conversion is actually started. This removes the firmware overhead required to allow for an acquisition (sampling) period (see Register 12-3 and **Section 12.2 “Selecting and Configuring Automatic Acquisition Time”**).

The 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)
- A/D Control Register 2 (ADCON2)

The ADCON0 register, shown in Register 12-1, controls the operation of the A/D module and clock source. The ADCON1 register, shown in Register 12-2, configures the functions of the port pins, justification and voltage reference sources. The ADCON2, shown in Register 12-3, configures the programmed acquisition time.

Additional information on using the A/D module can be found in the *“PICmicro® Mid-Range MCU Family Reference Manual”* (DS33023) and in Application Note AN546 *“Using the Analog-to-Digital (A/D) Converter”* (DS00546).

# PIC16F7X7

## REGISTER 12-1: ADCON0: A/D CONTROL REGISTER 0 (ADDRESS 1Fh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	CHS3	ADON
bit 7						bit 0	

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

If ADCS2 = 0:

000 = FOSC/2

001 = FOSC/8

010 = FOSC/32

011 = FRC (clock derived from an RC oscillation)

If ADCS2 = 1:

00 = FOSC/4

01 = FOSC/16

10 = FOSC/64

11 = FRC (clock derived from an RC oscillation)

bit 5-3 **CHS<2:0>:** Analog Channel Select bits

0000 = Channel 00 (AN0)

0001 = Channel 01 (AN1)

0010 = Channel 02 (AN2)

0011 = Channel 03 (AN3)

0100 = Channel 04 (AN4)

0101 = Channel 05 (AN5)<sup>(1)</sup>

0110 = Channel 06 (AN6)<sup>(1)</sup>

0111 = Channel 07 (AN7)<sup>(1)</sup>

1000 = Channel 08 (AN8)

1001 = Channel 09 (AN9)

1010 = Channel 10 (AN10)

1011 = Channel 11 (AN11)

1100 = Channel 12 (AN12)

1101 = Channel 13 (AN13)

111x = Unused

**Note 1:** Selecting AN5 through AN7 on the 28-pin product variant (PIC16F737 and PIC16F767) will result in a full-scale conversion as unimplemented channels are connected to VDD.

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

1 = A/D conversion cycle in progress. Setting this bit starts an A/D conversion cycle. This bit is automatically cleared by hardware when the A/D conversion has completed.

0 = A/D conversion completed/not in progress

bit 1 **CHS<3>:** Analog Channel Select bit (see bit 5-3 for bit settings)

bit 0 **ADON:** A/D Conversion Status bit

1 = A/D converter module is operating

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

### Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

## REGISTER 12-2: ADCON1: A/D CONTROL REGISTER 1 (ADDRESS 9Fh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	ADCS2	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0
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 two when system clock is used  
 0 = Disabled
- bit 5 **VCFG1:** Voltage Reference Configuration bit 1  
 0 = VREF- is connected to VSS  
 1 = VREF- is connected to external VREF- (RA2)
- bit 4 **VCFG0:** Voltage Reference Configuration bit 0  
 0 = VREF+ is connected to VDD  
 1 = VREF+ is connected to external VREF+ (RA3)
- bit 3-0 **PCFG<3:0>:** A/D Port Configuration bits

	AN13	AN12	AN11	AN10	AN9	AN8	AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0
0000	A	A	A	A	A	A	A	A	A	A	A	A	A	A
0001	A	A	A	A	A	A	A	A	A	A	A	A	A	A
0010	D	A	A	A	A	A	A	A	A	A	A	A	A	A
0011	D	D	A	A	A	A	A	A	A	A	A	A	A	A
0100	D	D	D	A	A	A	A	A	A	A	A	A	A	A
0101	D	D	D	D	A	A	A	A	A	A	A	A	A	A
0110	D	D	D	D	D	A	A	A	A	A	A	A	A	A
0111	D	D	D	D	D	D	A	A	A	A	A	A	A	A
1000	D	D	D	D	D	D	D	A	A	A	A	A	A	A
1001	D	D	D	D	D	D	D	D	A	A	A	A	A	A
1010	D	D	D	D	D	D	D	D	D	A	A	A	A	A
1011	D	D	D	D	D	D	D	D	D	D	A	A	A	A
1100	D	D	D	D	D	D	D	D	D	D	D	A	A	A
1101	D	D	D	D	D	D	D	D	D	D	D	D	A	A
1110	D	D	D	D	D	D	D	D	D	D	D	D	D	A
1111	D	D	D	D	D	D	D	D	D	D	D	D	D	D

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

**Note:** AN5 through AN7 are only available on the 40-pin product variant (PIC16F747 and PIC16F777).

**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

# PIC16F7X7

## REGISTER 12-3: ADCON2: A/D CONTROL REGISTER 2 (ADDRESS 9Bh)

U-0	U-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0
—	—	ACQT2	ACQT1	ACQT0	—	—	—
bit 7						bit 0	

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

bit 5-3 **ACQT<2:0>:** A/D Acquisition Time Select bits

000 = 0<sup>(1)</sup>  
 001 = 2 TAD  
 010 = 4 TAD  
 011 = 6 TAD  
 100 = 8 TAD  
 101 = 12TAD  
 110 = 16 TAD  
 111 = 20 TAD

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

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

<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

The analog reference voltage is software selectable to either the device's positive and negative supply voltage (V<sub>DD</sub> and V<sub>SS</sub>) or the voltage level on the RA3/AN3/VREF+ and RA2/AN2/VREF-/CVREF pins.

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 output of the sample and hold is the input into the converter which generates the result via successive approximation.

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

Each port pin associated with the A/D converter can be configured as an analog input or as a digital I/O. The ADRESH and ADRESL registers contain the result of the A/D conversion. When the A/D conversion is complete, the result is loaded into the ADRESH/ADRESL registers, the GO/DONE bit (ADCON0 register) 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.

The value 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.

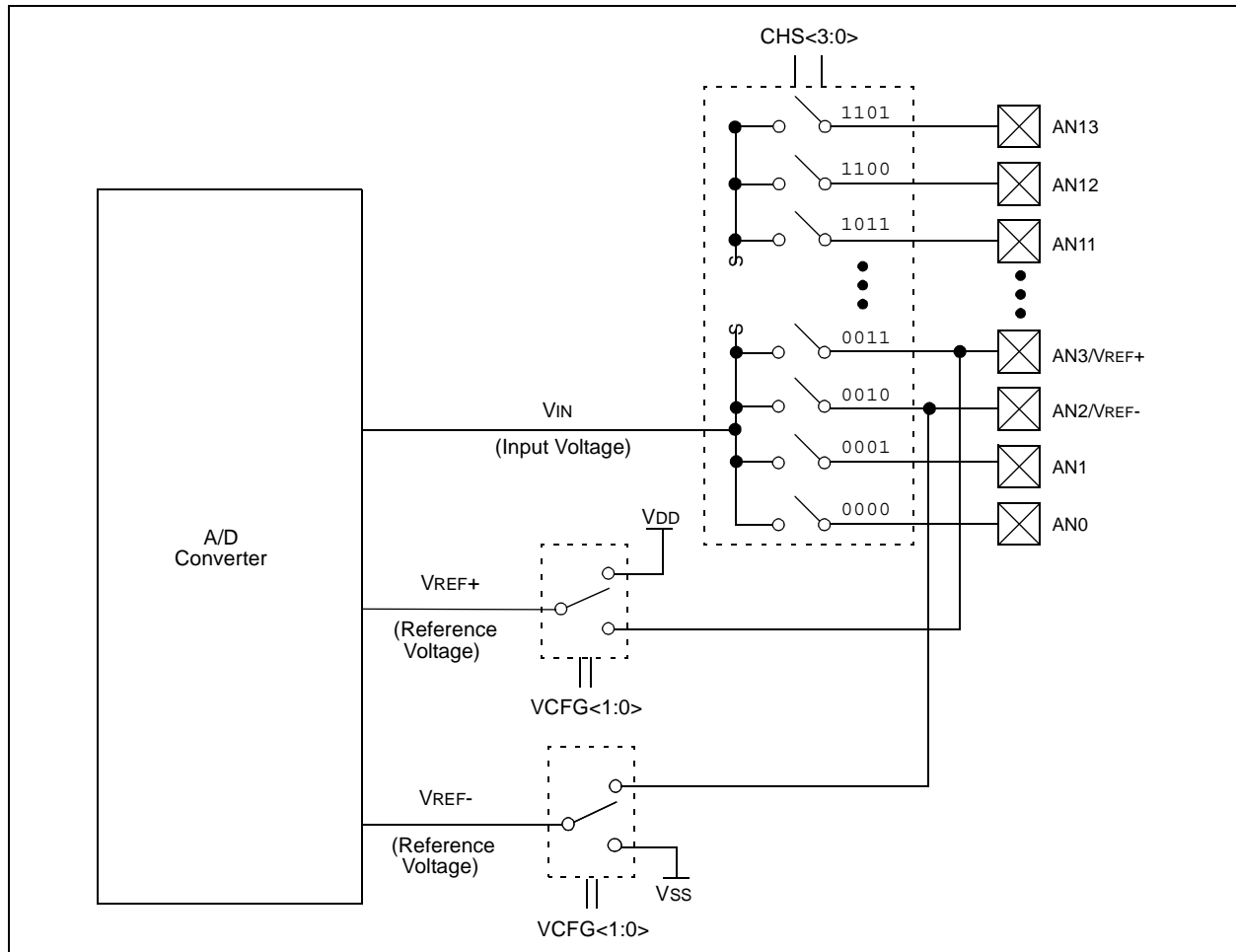
After the A/D module has been configured as desired, the selected channel must be acquired before the conversion is started. The analog input channels must have their corresponding TRIS bits selected as an input. To determine acquisition time, see **Section 12.1 “A/D Acquisition Requirements”**. After this acquisition time has elapsed, the A/D conversion can be started. An acquisition time can be programmed to occur between setting the GO/DONE bit and the actual start of the conversion.

The following steps should be followed to do an A/D conversion:

1. Configure the A/D module:
  - Configure analog pins, voltage reference and digital I/O (ADCON1)
  - Select A/D input channel (ADCON0)
  - Select A/D acquisition time (ADCON2)
  - 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 (if required).
4. Start conversion:
  - Set GO/DONE bit (ADCON0 register)
5. Wait for A/D conversion to complete, by either:
  - Polling for the GO/DONE bit to be cleared
  - OR
  - Waiting for the A/D interrupt
6. Read A/D Result registers (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 T<sub>AD</sub>. A minimum wait of 2 T<sub>AD</sub> is required before the next acquisition starts.

**FIGURE 12-1: A/D BLOCK DIAGRAM**



# PIC16F7X7

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

To calculate the minimum acquisition time, Equation 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.

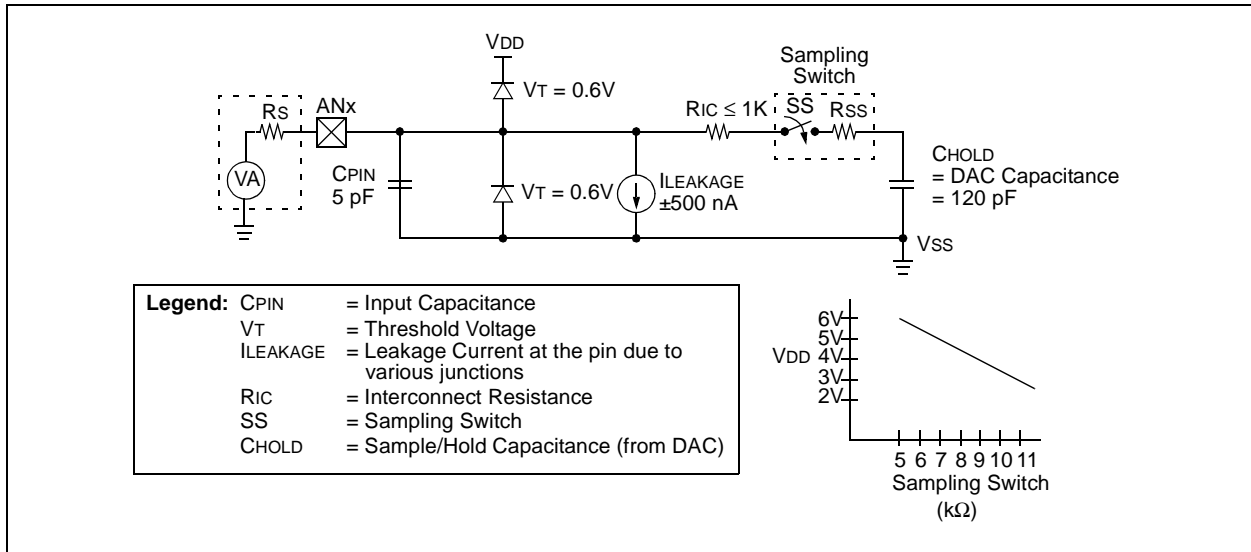
To calculate the minimum acquisition time, TACQ, see the "PICmicro® Mid-Range MCU Family Reference Manual" (DS33023).

### EQUATION 12-1: ACQUISITION TIME

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

- 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



## 12.2 Selecting and Configuring Automatic Acquisition Time

The ADCON2 register allows the user to select an acquisition time that occurs each time the GO/DONE bit is set.

When the GO/DONE bit is set, sampling is stopped and a conversion begins. The user is responsible for ensuring the required acquisition time has passed between selecting the desired input channel and setting the GO/DONE bit. This occurs when the ACQT2:ACQT0 bits (ADCON2<5:3>) remain in their Reset state ('000') and is compatible with devices that do not offer programmable acquisition times.

If desired, the ACQT bits can be set to select a programmable acquisition time for the A/D module. When the GO/DONE bit is set, the A/D module continues to sample the input for the selected acquisition time, then automatically begins a conversion. Since the acquisition time is programmed, there may be no need to wait for an acquisition time between selecting a channel and setting the GO/DONE bit.

In either case, when the conversion is completed, the GO/DONE bit is cleared, the ADIF flag is set and the A/D begins sampling the currently selected channel again. If an acquisition time is programmed, there is nothing to indicate if the acquisition time has ended or if the conversion has begun.

## 12.3 Selecting the A/D Conversion Clock

The A/D conversion time per bit is defined as TAD. The A/D conversion requires a minimum 12 TAD per 10-bit conversion. The source of the A/D conversion clock is software selected. 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 of 1.6  $\mu$ s.

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

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

AD Clock Source (TAD)		Maximum Device Frequency
Operation	ADCS2:ADCS1:ADCS0	
2 TOSC	000	1.25 MHz
4 TOSC	100	2.5 MHz
8 TOSC	001	5 MHz
16 TOSC	101	10 MHz
32 TOSC	010	20 MHz
64 TOSC	110	20 MHz
RC <sup>(1,2,3)</sup>	x11	<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 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 ACQT2:ACQT0 (ADCON2<5:3>) and ADCS2:ADCS0 (ADCON1<6>, ADCON0<7:6>) bits 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.

Operation in Sleep mode requires the A/D RC clock to be selected. If bits ACQT2:ACQT0 are set to '000' and a conversion is started, the conversion will be delayed one instruction cycle to allow execution of the SLEEP instruction and entry to Sleep mode.

## 12.5 Configuring Analog Port Pins

The ADCON1, TRISA, TRISB and TRISE 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 CHS2:CHS0 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, but not as an analog input, may cause the digital input buffer to consume current that is out of the device's specification.



## 12.6 A/D Conversions

Figure 12-3 shows the operation of the A/D converter after the  $\overline{\text{GO/DONE}}$  bit has been set and the ACQT2:ACQT0 bits are cleared. A conversion is started after the following instruction to allow entry into Sleep mode before the conversion begins.

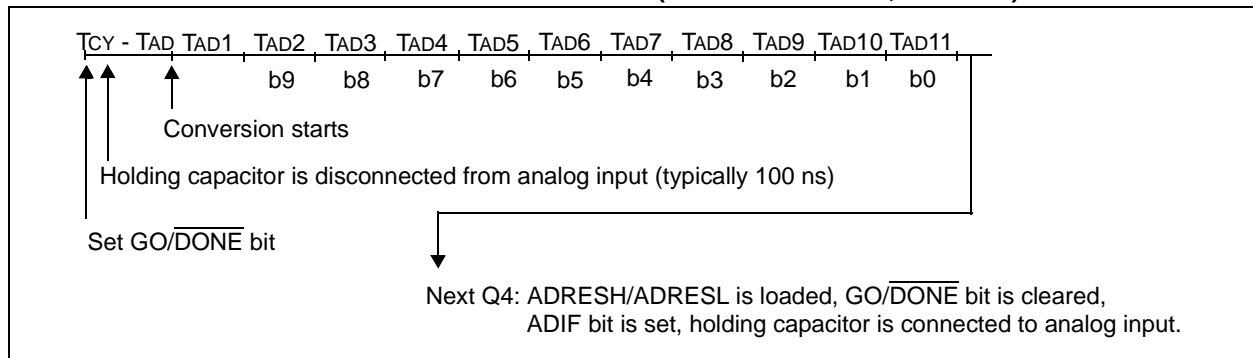
Figure 12-4 shows the operation of the A/D converter after the  $\overline{\text{GO/DONE}}$  bit has been set, the ACQT2:ACQT0 bits are set to '010' and a 4 TAD acquisition time is selected before the conversion starts.

Clearing the  $\overline{\text{GO/DONE}}$  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. This means 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).

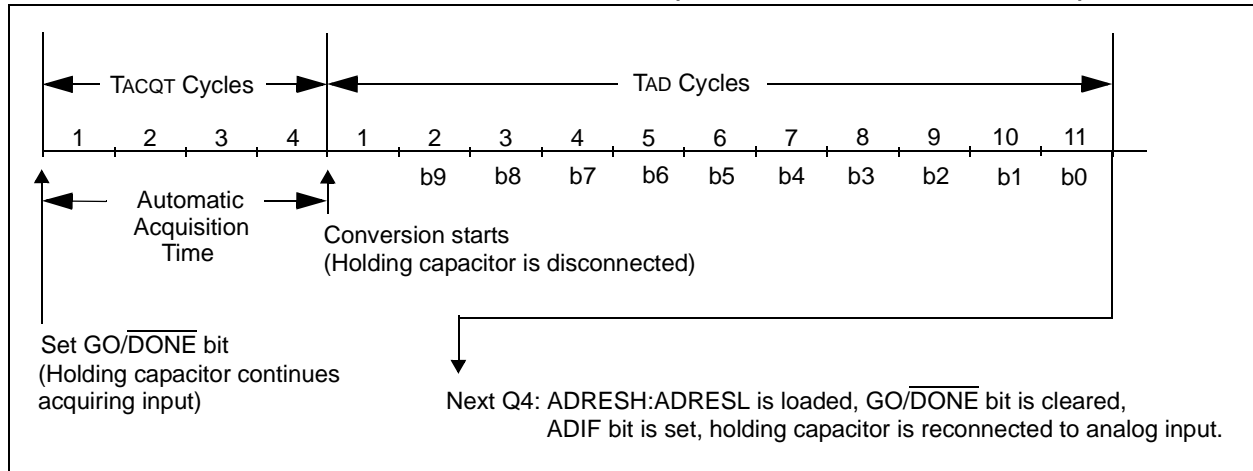
After the A/D conversion is completed or aborted, a 2 TAD wait is required before the next acquisition can be started. After this wait, acquisition on the selected channel is automatically started.

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

**FIGURE 12-3: A/D CONVERSION TAD CYCLES (ACQT<2:0> = 000, TACQ = 0)**



**FIGURE 12-4: A/D CONVERSION TAD CYCLES (ACQT<2:0> = 010, TACQ = 4 TAD)**



# PIC16F7X7

## 12.7 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 ADRESH register. If the A/D interrupt is enabled, the device will wake-up from Sleep. If the A/D interrupt is not enabled, the A/D module will then be turned off, although the ADON bit will remain set.

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

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

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

## 12.8 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 ADRESH register will contain unknown data after a Power-on Reset.

## 12.9 Use of the CCP Trigger

An A/D conversion can be started by the “special event trigger” of the CCP2 module. This requires that the CCP2M3:CCP2M0 bits (CCP2CON<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 to the desired location). The appropriate analog input channel must be selected and an appropriate acquisition time should pass 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: SUMMARY OF A/D REGISTERS**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, 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	PSPIF <sup>(1)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
0Dh	PIR2	OSFIF	CMIF	LVDIF	—	BCLIF	—	CCP3IF	CCP2IF	000- 0-00	000- 0-00
8Ch	PIE1	PSPIE <sup>(1)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
8Dh	PIE2	OSFIE	CMIE	LVDIE	—	BCLIE	—	CCP3IE	CCP2IE	000- 0-00	000- 0-00
1Eh	ADRESH	A/D Result Register High Byte								xxxx xxxx	uuuu uuuu
1Fh	ADCON0	ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	CHS3	ADON	0000 0000	0000 0000
9Fh	ADCON1	ADFM	ADCS2	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0	0000 000	0000 0000
05h	PORTA	RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0	xx0x 0000	uu0u 0000
85h	TRISA	PORTA Data Direction Register								1111 1111	1111 1111
09h	PORTE <sup>(2)</sup>	—	—	—	—	RE3 <sup>(3)</sup>	RE2	RE1	RE0	---- x000	---- x000
89h	TRISE <sup>(2)</sup>	IBF	OBF	IBOV	PSPMODE	— <sup>(3)</sup>	PORTE Data Direction bits			0000 1111	0000 1111

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

**Note 1:** Bits PSPIE and PSPIF are reserved on the PIC16F737/767 devices; always maintain these bits clear.

**2:** These registers are reserved on the PIC16F737/767 devices.

**3:** RE3 is an input only. The state of the TRISE3 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 RA4 and RA5. 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 multiplexers. 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 = 110:

1 = C1 VIN- connects to RA3/AN3

C2 VIN- connects to RA2/AN2

0 = C1 VIN- connects to RA0/AN0

C2 VIN- connects to RA1/AN1

bit 2-0 **CM2:CM0**: Comparator Mode bits

Figure 13-1 shows the Comparator modes and CM2:CM0 bit settings.

**Legend:**

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

# PIC16F7X7

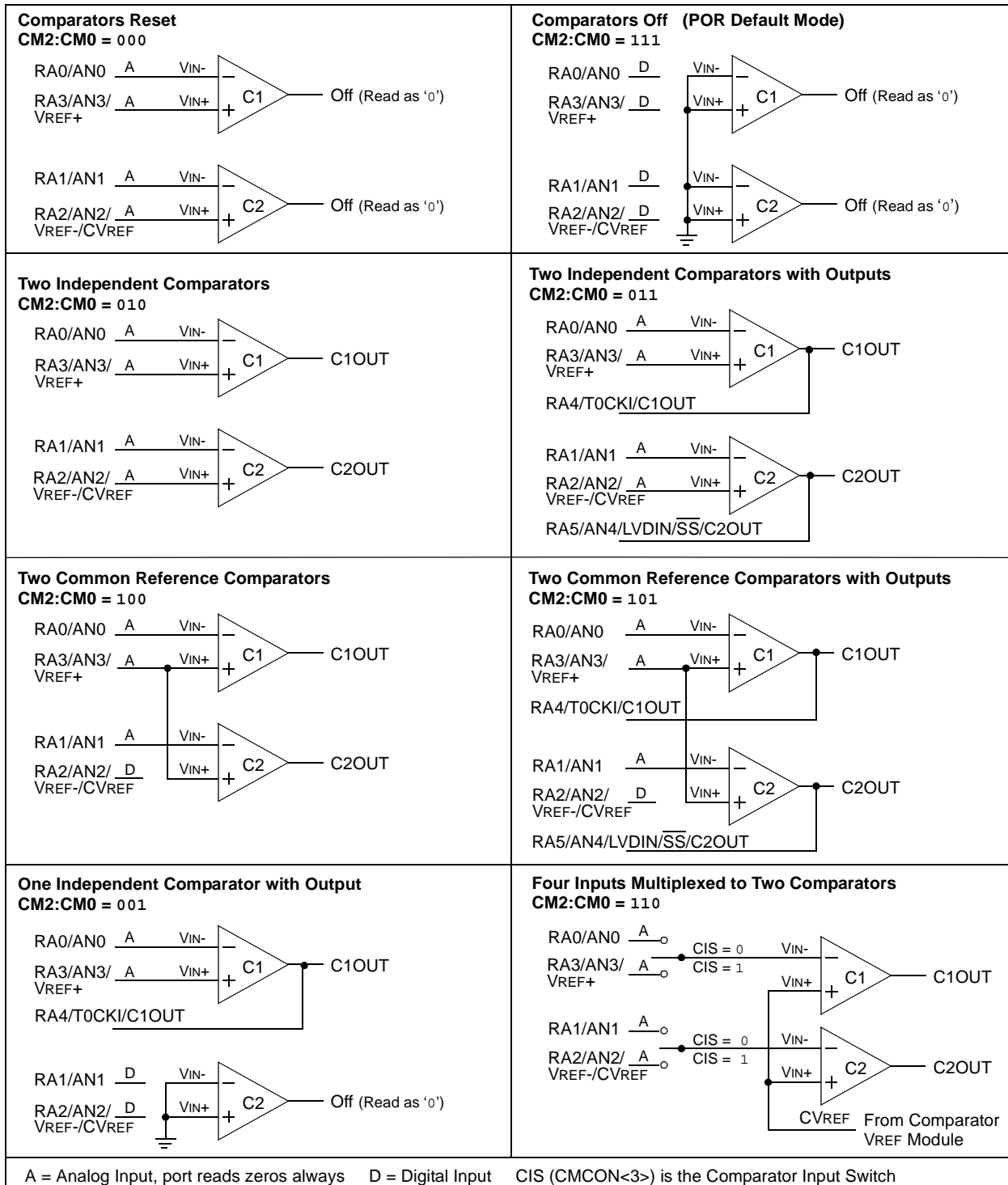
## 13.1 Comparator Configuration

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 the electrical specifications (**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**



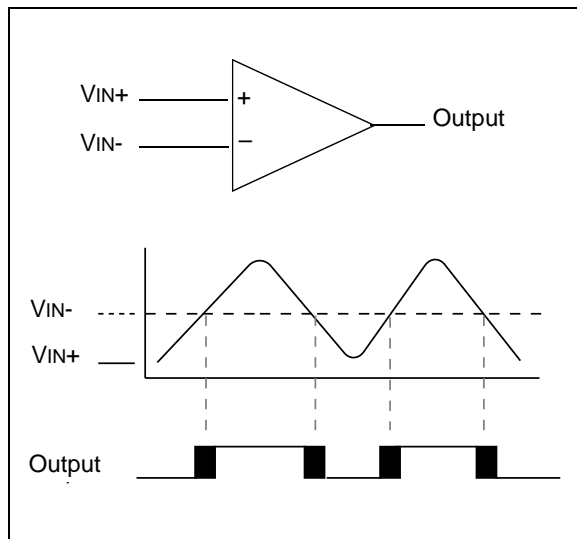
## 13.2 Comparator Operation

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  $V_{IN+}$  is less than the analog input  $V_{IN-}$ , the output of the comparator is a digital low level. When the analog input at  $V_{IN+}$  is greater than the analog input  $V_{IN-}$ , 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  $V_{IN-}$  is compared to the signal at  $V_{IN+}$  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  $V_{SS}$  and  $V_{DD}$  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  $CM\langle 2:0 \rangle = 110$  (Figure 13-1). In this mode, the internal voltage reference is applied to the  $V_{IN+}$  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  $RA4$  and  $RA5$  I/O pins. When enabled, multiplexors in the output path of the  $RA4$  and  $RA5$  pins 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.

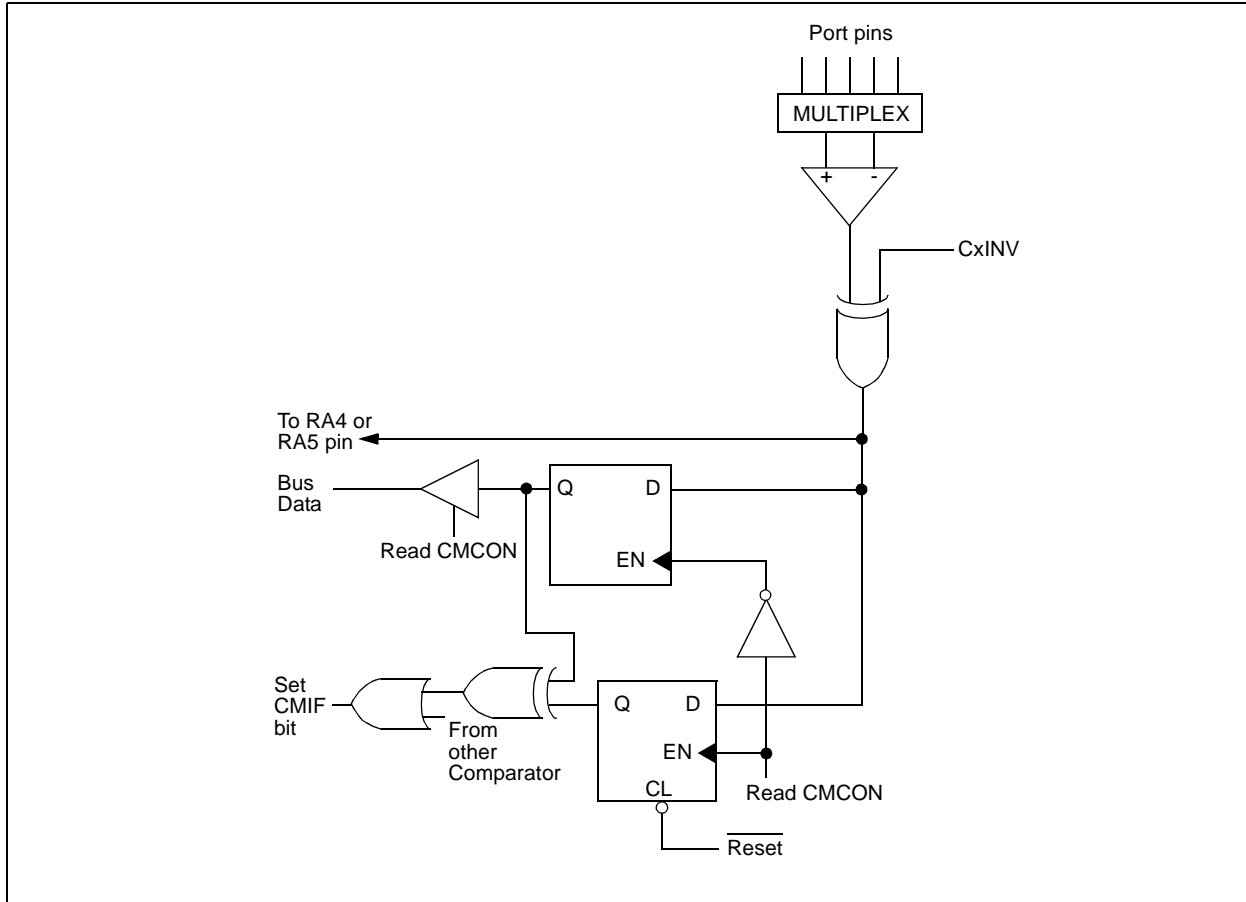
The  $TRISA$  bits will still function as an output enable/disable for the  $RA4$  and  $RA5$  pins while in this mode.

The polarity of the comparator outputs can be changed using the  $C2INV$  and  $C1INV$  bits ( $CMCON\langle 5:4 \rangle$ ).

- Note 1:** When reading the Port register, all pins configured as analog inputs will read as a ‘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.
- 3:**  $RA4$  is an open collector I/O pin. When used as an output, a pull-up resistor is required.

# PIC16F7X7

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<7:6>, 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:

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

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

## 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.

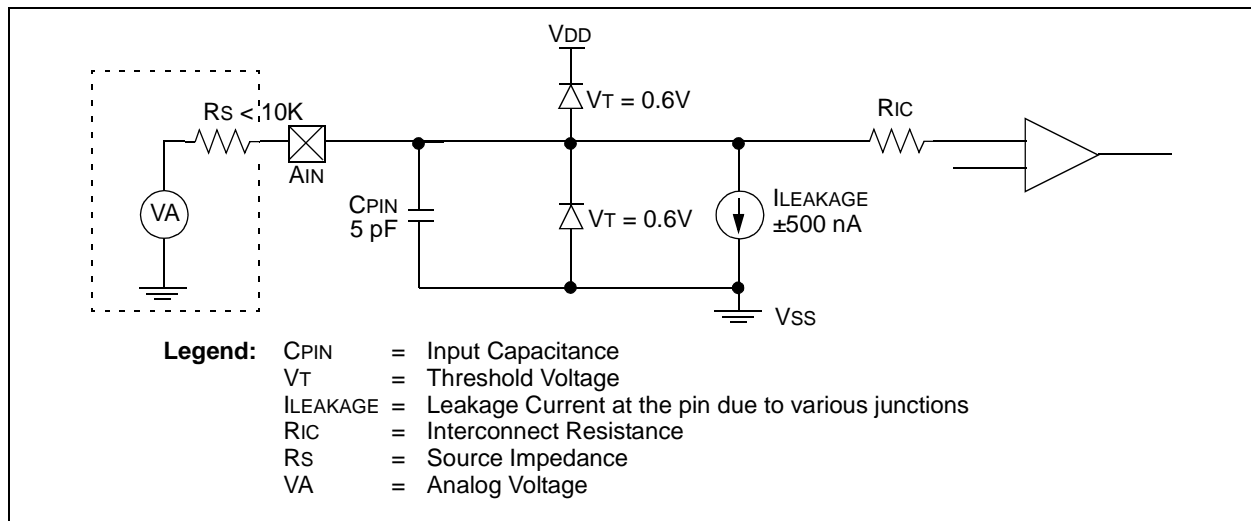
## 13.8 Effects of a Reset

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. This ensures compatibility to the PIC16F87X devices.

## 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 kΩ is recommended for the analog sources. Any external component connected to an analog input pin, such as a capacitor or a Zener diode, should have very little leakage current.

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



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

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR	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	LVDIF	—	BCLIF	—	CCP3IF	CCP2IF	000- 0-00	000- 0-00
8Dh	PIE2	OSFIE	CMIE	LVDIE	—	BCLIE	—	CCP3IE	CCP2IE	000- 0-00	000- 0-00
05h	PORTA	RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0	xx0x 0000	uu0u 0000
85h	TRISA	PORTA Data Direction Register								1111 1111	1111 1111

**Legend:** x = unknown, u = unchanged, — = unimplemented, read as '0'. Shaded cells are unused by the comparator module.

# PIC16F7X7

---

NOTES:



## 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 '110'. 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/VREF-/CVREF pin. 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

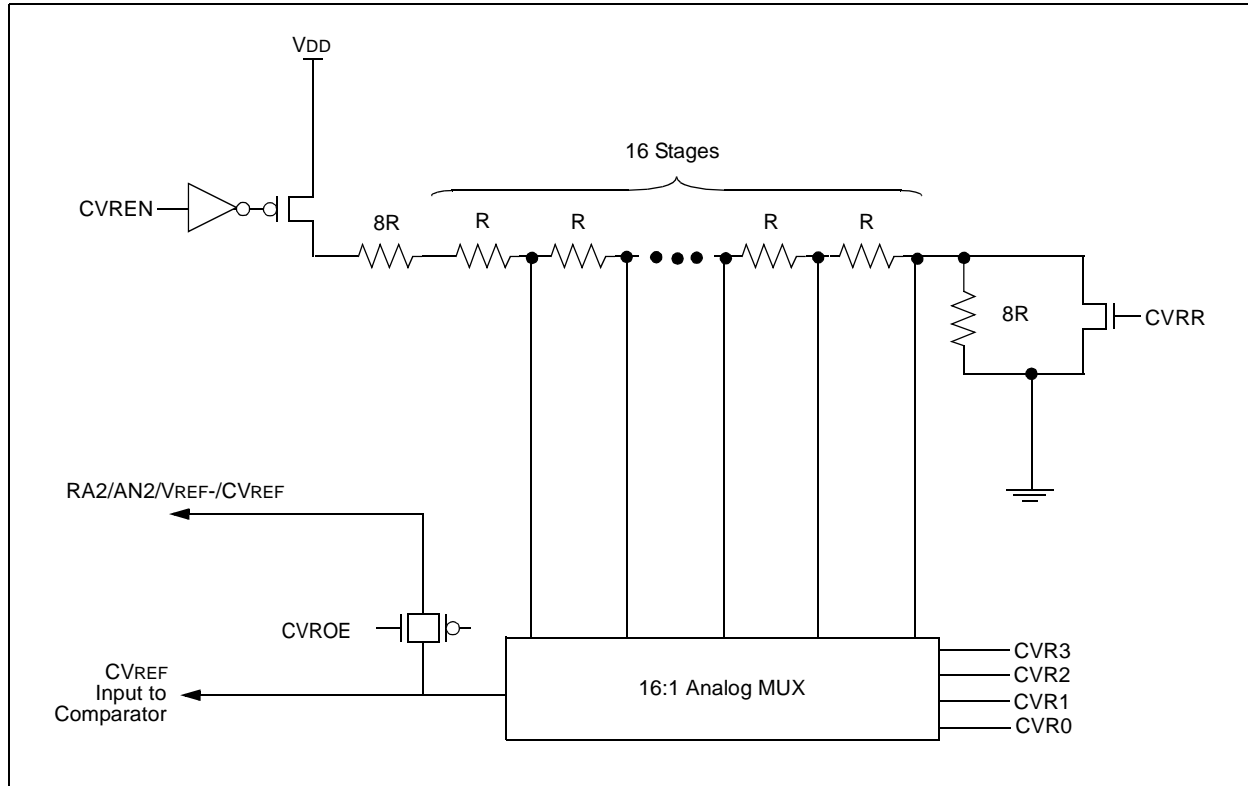
- bit 7 **CVREN:** Comparator Voltage Reference Enable bit  
 1 = CVREF circuit powered on  
 0 = CVREF circuit powered down
- bit 6 **CVROE:** Comparator VREF Output Enable bit  
 1 = CVREF voltage level is output on RA2/AN2/VREF-/CVREF pin  
 0 = CVREF voltage level is disconnected from RA2/AN2/VREF-/CVREF pin
- bit 5 **CVRR:** Comparator VREF Range Selection bit  
 1 = 0 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 **CVR3:CVR0:** Comparator VREF Value Selection bits  $0 \leq CVR3:CVR0 \leq 15$   
When CVRR = 1:  
 $CVREF = (CVR<3:0>/24) \cdot (CVRSRC)$   
When CVRR = 0:  
 $CVREF = 1/4 \cdot (CVRSRC) + (CVR3:CVR0/32) \cdot (CVRSRC)$

#### 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

# PIC16F7X7

**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)
  - Low-Voltage Detect (LVD)
- Interrupts
- Watchdog Timer (WDT)
- Two-Speed Start-up
- Fail-Safe Clock Monitor
- Sleep
- Code Protection
- ID Locations
- In-Circuit Serial Programming

There are two timers that offer necessary delays on power-up. One is the Oscillator Start-up Timer (OST), intended to keep the chip in Reset until the crystal oscillator is stable. The other is the Power-up Timer (PWRT) which provides a fixed delay of 72 ms (nominal) on power-up only. It is designed to keep the part in Reset while the power supply stabilizes and is enabled or disabled using a configuration bit. With these two timers on-chip, most applications need no external Reset circuitry.

Sleep mode is designed to offer a very low-current power-down mode. The user can wake-up from Sleep through external Reset, Watchdog Timer wake-up or through an interrupt.

Several oscillator options are also made available to allow the part to fit the application. The RC oscillator option saves system cost while the LP crystal option saves power. Configuration bits are used to select the desired oscillator mode.

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

### 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.

# PIC16F7X7

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

R/P-1	R/P-1	R/P-1	U-1	U-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	—	—	BORV1	BORV0	BOREN	MCLR $\overline{\text{E}}$	FOSC2	$\overline{\text{PWRTEN}}$	WDTEN	FOSC1	FOSC0	
bit 13													bit 0	

- bit 13 **CP:** Flash Program Memory Code Protection bits  
 1 = Code protection off  
 0 = 0000h to 1FFFh code-protected for PIC16F767/777 and 0000h to 0FFFh for PIC16F737/747 (all protected)
- bit 12 **CCPMX:** CCP2 Multiplex bit  
 1 = CCP2 is on RC1  
 0 = CCP2 is 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 **Unimplemented:** Read as '1'
- bit 8-7 **BORV<1:0>:** Brown-out Reset Voltage bits  
 11 = VBOR set to 2.0V  
 10 = VBOR set to 2.7V  
 01 = VBOR set to 4.2V  
 00 = VBOR set to 4.5V
- bit 6 **BOREN:** Brown-out Reset Enable bit  
 BOREN combines with BORSEN to control when BOR is enabled and how it is controlled.  
BOREN:BORSEN:  
 11 = BOR enabled and always on  
 10 = BOR enabled during operation and disabled during Sleep by hardware  
 01 = BOR controlled by software bit SBOREN – refer to Register 2-8 (PCON<2>)  
 00 = BOR disabled
- bit 5 **MCLR $\overline{\text{E}}$ :** MCLR/VPP/RE3 Pin Function Select bit  
 1 = MCLR/VPP/RE3 pin function is MCLR  
 0 = MCLR/VPP/RE3 pin function is digital input only, MCLR gated to '1'
- bit 3  **$\overline{\text{PWRTEN}}$ :** Power-up Timer Enable bit  
 1 = PWRT disabled  
 0 = PWRT enabled
- bit 2 **WDTEN:** Watchdog Timer Enable bit  
 1 = WDT enabled  
 0 = WDT disabled
- bit 4, 1-0 **FOSC2:FOSC0:** Oscillator Selection bits  
 111 = EXTRC oscillator; CLKO function on OSC2/CLKO/RA6  
 110 = EXTRC oscillator; port I/O function on OSC2/CLKO/RA6  
 101 = INTRC oscillator; CLKO function on OSC2/CLKO/RA6 and port I/O function on OSC1/CLKI/RA7  
 100 = INTRC oscillator; port I/O function on OSC1/CLKI/RA7 and OSC2/CLKO/RA6  
 011 = EXTCLK; port I/O function on OSC2/CLKO/RA6  
 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: CONFIGURATION WORD REGISTER 2 (ADDRESS 2008h)

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

bit 13-7 **Unimplemented:** Read as '1'

bit 6 **BORSEN:** Brown-out Reset Software Enable bit  
Refer to Configuration Word Register 1, bit 6 for the function of this bit.

bit 5-2 **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

### 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

# PIC16F7X7

## 15.2 Reset

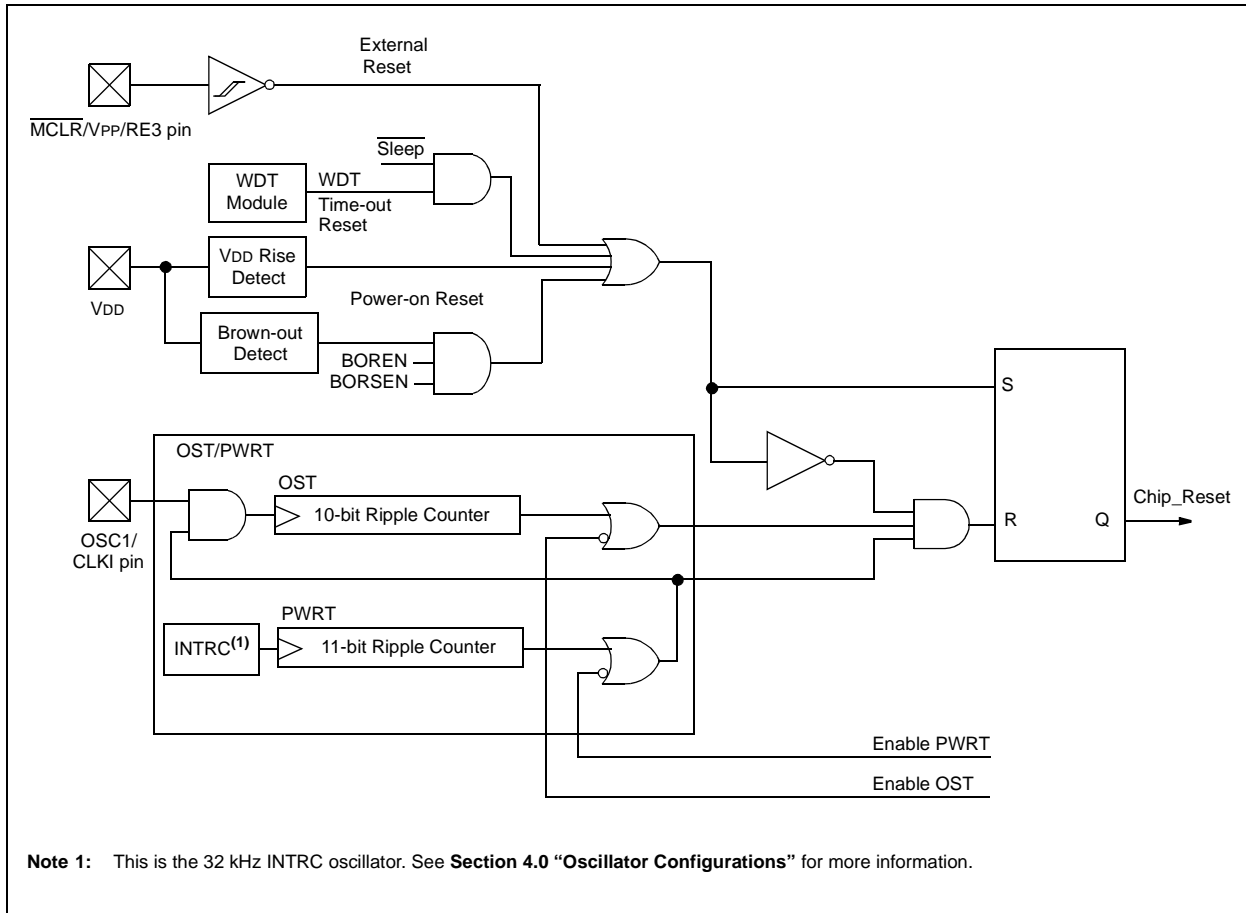
The PIC16F7X7 differentiates between various kinds of Reset:

- 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

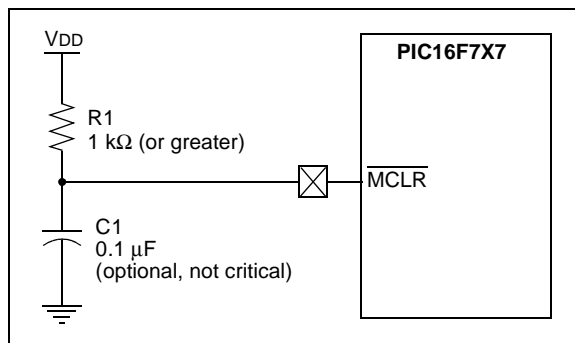
PIC16F7X7 devices have a noise filter in the MCLR Reset path. This filter will detect and ignore small pulses.

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

The behavior of the ESD protection on the MCLR pin has been altered from previous devices of this family. Voltages applied to the pin that exceed its specification can result in both MCLR and excessive current, beyond the device specification, during the ESD event. For this reason, Microchip recommends that the MCLR pin no longer be tied directly to VDD. The use of an RC network, as shown in Figure 15-2, is suggested.

The MCLR/VPP/RE3 pin can be configured for MCLR (default) or as an input pin (RE3). This is configured through the MCLRE bit in Configuration Word Register 1.

**FIGURE 15-2: RECOMMENDED MCLR CIRCUIT**



## 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 PIC16F7X7 is a counter that uses the INTRC oscillator as the clock input. This yields a count of 72 ms. While the PWRT is counting, the device is held in Reset.

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

The PWRT is enabled by clearing configuration bit PWRTEN.

## 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)

Three configuration bits (BOREN – Configuration Word Register 1, bit 6; BORSEN – Configuration Word Register 2, bit 6; SBORSEN – PCON register, bit 2) together disable or enable the Brown-out Reset circuit in one of its three operating modes.

If VDD falls below VBOR (defined by BORV<1:0> bits in Configuration Word Register 1) for longer than TBOR (parameter #35, about 100 μs), the brown-out situation will reset the device. If VDD falls below VBOR for less than TBOR, a Reset may not occur.

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

# PIC16F7X7

## 15.8 Low-Voltage Detect

In many applications, the ability to determine if the device voltage (VDD) is below a specified voltage level is a desirable feature. A window of operation for the application can be created where the application software can do “housekeeping tasks” before the device voltage exits the valid operating range. This can be done using the Low-Voltage Detect module.

This module is a software programmable circuitry where a device voltage trip point can be specified. When the voltage of the device becomes lower than the specified point, an interrupt flag is set. If the interrupt is enabled, the program execution will branch to the interrupt vector address and the software can then respond to that interrupt source.

The Low-Voltage Detect circuitry is completely under software control. This allows the circuitry to be turned off by the software which minimizes the current consumption for the device.

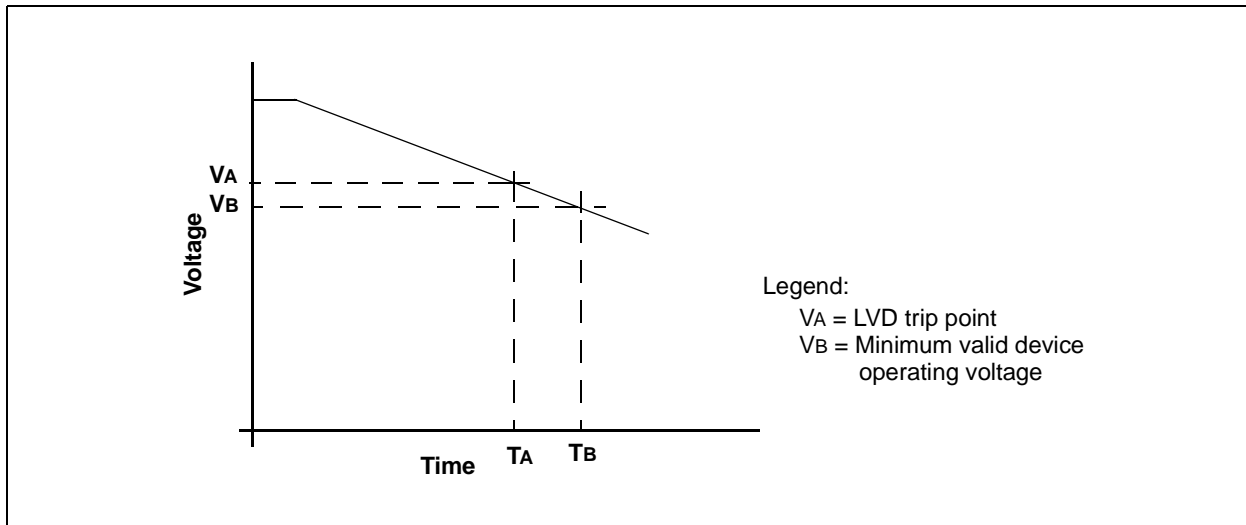
Figure 15-3 shows a possible application voltage curve (typically for batteries). Over time, the device voltage decreases. When the device voltage equals voltage VA, the LVD logic generates an interrupt. This occurs at

time TA. The application software then has the time, until the device voltage is no longer in valid operating range, to shut-down the system. Voltage point VB is the minimum valid operating voltage specification. This occurs at time TB. The difference, TB – TA, is the total time for shutdown.

The block diagram for the LVD module is shown in Figure 15-4. A comparator uses an internally generated reference voltage as the set point. When the selected tap output of the device voltage crosses the set point (is lower than), the LVDIF bit is set.

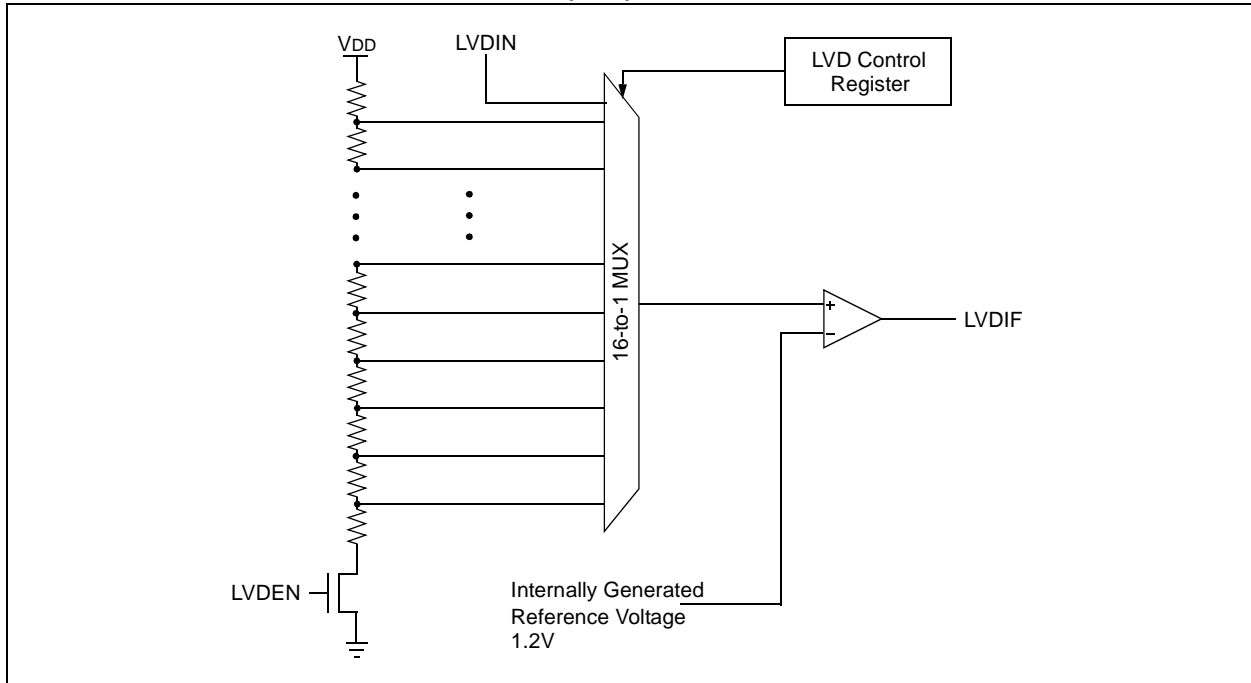
Each node in the resistor divider represents a “trip point” voltage. The “trip point” voltage is the minimum supply voltage level at which the device can operate before the LVD module asserts an interrupt. When the supply voltage is equal to the trip point, the voltage tapped off of the resistor array is equal to the 1.2V internal reference voltage generated by the voltage reference module. The comparator then generates an interrupt signal setting the LVDIF bit. This voltage is software programmable to any one of 16 values (see Figure 15-4). The trip point is selected by programming the LVDL3:LVDL0 bits (LVDCON<3:0>).

**FIGURE 15-3: TYPICAL LOW-VOLTAGE DETECT APPLICATION**





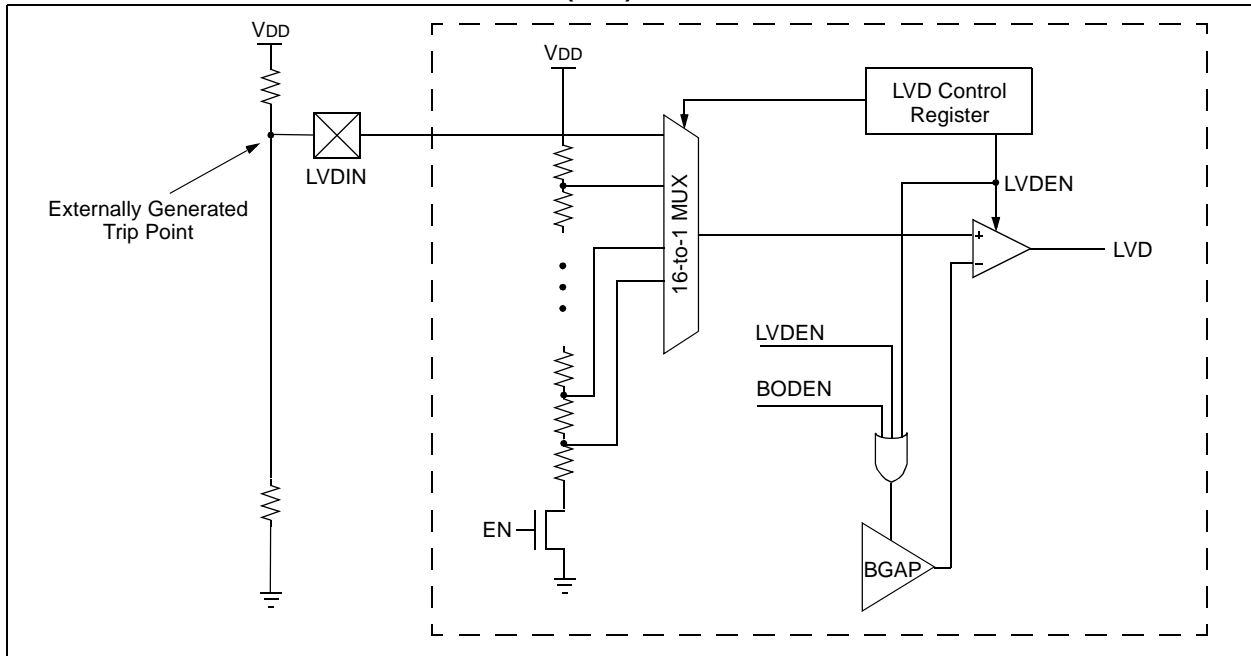
**FIGURE 15-4: LOW-VOLTAGE DETECT (LVD) BLOCK DIAGRAM**



The LVD module has an additional feature that allows the user to supply the sense voltage to the module from an external source. This mode is enabled when bits LVDL3:LVDL0 are set to '1111'. In this state, the comparator input is multiplexed from the external input

pin, LVDIN (Figure 15-5). This gives users flexibility because it allows them to configure the Low-Voltage Detect interrupt to occur at any voltage in the valid operating range.

**FIGURE 15-5: LOW-VOLTAGE DETECT (LVD) WITH EXTERNAL INPUT BLOCK DIAGRAM**



# PIC16F7X7

## 15.9 Control Register

The Low-Voltage Detect Control register controls the operation of the Low-Voltage Detect circuitry.

### REGISTER 15-3: LVDCON: LOW-VOLTAGE DETECT CONTROL REGISTER (ADDRESS 109h)

U-0	U-0	R-0	R/W-0	R/W-0	R/W-1	R/W-0	R/W-1	
—	—	IRVST	LV DEN	LV DL3	LV DL2	LV DL1	LV DL0	
bit 7								bit 0

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

bit 5 **IRVST:** Internal Reference Voltage Stable Flag bit

1 = Indicates that the Low-Voltage Detect logic will generate the interrupt flag at the specified voltage range

0 = Indicates that the Low-Voltage Detect logic will not generate the interrupt flag at the specified voltage range and the LVD interrupt should not be enabled

bit 4 **LV DEN:** Low-Voltage Detect Power Enable bit

1 = Enables LVD, powers up LVD circuit

0 = Disables LVD, powers down LVD circuit

bit 3-0 **LV DL3:LV DL0:** Voltage Detection Limit bits

1111 = External analog input is used (input comes from the LVDIN pin)

1110 = Maximum setting

.

.

.

0001 = Minimum setting

**Note:** See Table 18-3 in **Section 18.0 “Electrical Characteristics”** for the specifications.

#### 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

## 15.10 Operation

Depending on the power source for the device voltage, the voltage normally decreases relatively slowly. This means that the LVD module does not need to be constantly operating. To decrease the current requirements, the LVD circuitry only needs to be enabled for short periods where the voltage is checked. After doing the check, the LVD module may be disabled.

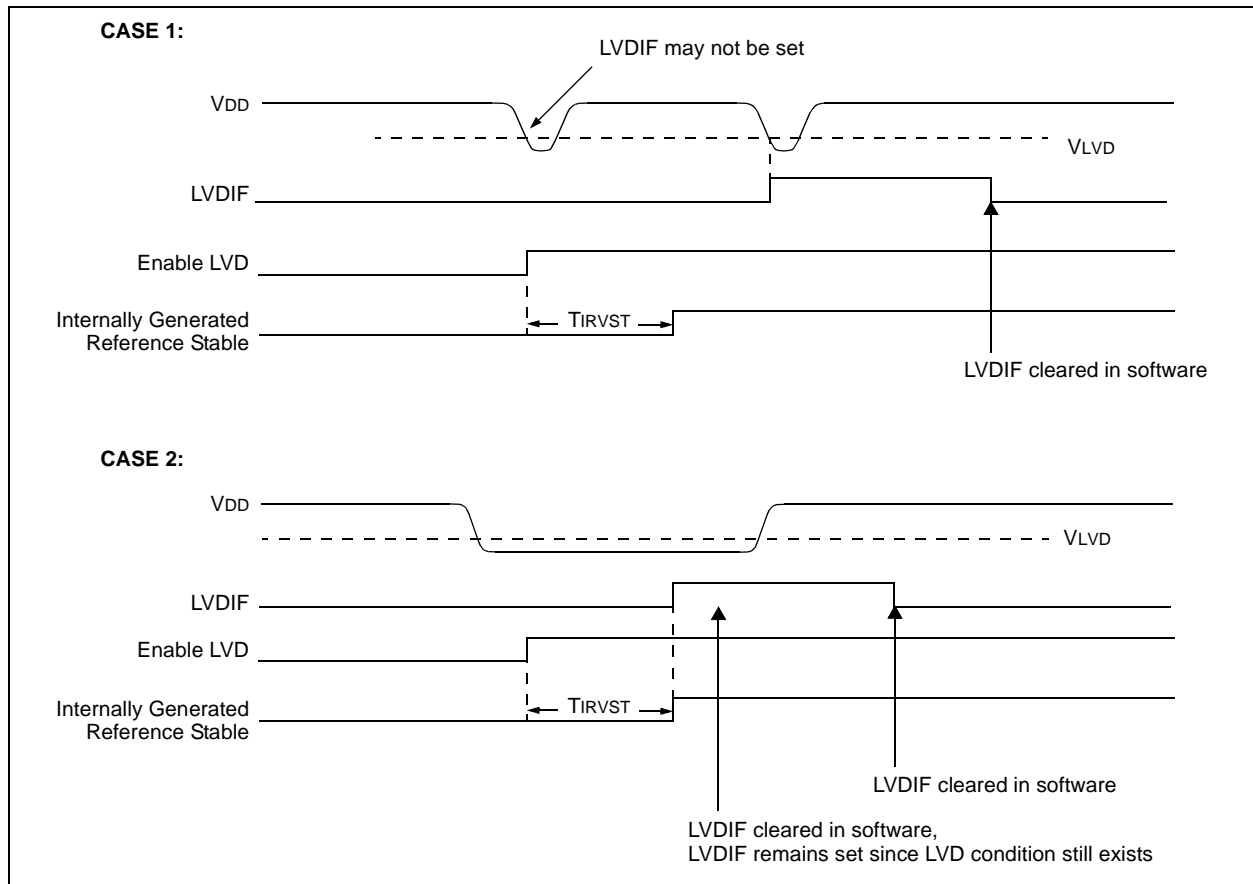
Each time that the LVD module is enabled, the circuitry requires some time to stabilize. After the circuitry has stabilized, all status flags may be cleared. The module will then indicate the proper state of the system.

The following steps are needed to set up the LVD module:

1. Write the value to the LVDL3:LVDL0 bits (LVDCON register) which selects the desired LVD trip point.
2. Ensure that LVD interrupts are disabled (the LVDIE bit is cleared or the GIE bit is cleared).
3. Enable the LVD module (set the LVDEN bit in the LVDCON register).
4. Wait for the LVD module to stabilize (the IRVST bit to become set).
5. Clear the LVD interrupt flag, which may have falsely become set, until the LVD module has stabilized (clear the LVDIF bit).
6. Enable the LVD interrupt (set the LVDIE and the GIE bits).

Figure 15-6 shows the typical waveforms that the LVD module may be used to detect.

**FIGURE 15-6: LOW-VOLTAGE DETECT WAVEFORMS**



# PIC16F7X7

---

## 15.10.1 REFERENCE VOLTAGE SET POINT

The internal reference voltage of the LVD module may be used by other internal circuitry (the Programmable Brown-out Reset). If these circuits are disabled (lower current consumption), the reference voltage circuit requires a time to become stable before a low-voltage condition can be reliably detected. This time is invariant of system clock speed. This start-up time is specified in electrical specification parameter #36. The low-voltage interrupt flag will not be enabled until a stable reference voltage is reached. Refer to the waveform in Figure 15-6.

## 15.10.2 CURRENT CONSUMPTION

When the module is enabled, the LVD comparator and voltage divider are enabled and will consume static current. The voltage divider can be tapped from multiple places in the resistor array. Total current consumption, when enabled, is specified in electrical specification parameter #D022B.

## 15.11 Operation During Sleep

When enabled, the LVD circuitry continues to operate during Sleep. If the device voltage crosses the trip point, the LVDIF bit will be set and the device will wake-up from Sleep. Device execution will continue from the interrupt vector address if interrupts have been globally enabled.

## 15.12 Effects of a Reset

A device Reset forces all registers to their Reset state. This forces the LVD module to be turned off.

**Note:** If the LVD is enabled and the BOR module is not enabled, the band gap will require a start-up time of no more than 50  $\mu$ s before the band gap reference is stable. Before enabling the LVD interrupt, the user should ensure that the band gap reference voltage is stable by monitoring the IRVST bit in the LVDCON register. The LVD could cause erroneous interrupts before the band gap is stable.

## 15.13 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 PIC16F7X7 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.

## 15.14 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 Power-on Reset Status bit,  $\overline{\text{POR}}$ . 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	$T_{\text{PWRT}} + 1024 \cdot T_{\text{OSC}}$	$1024 \cdot T_{\text{OSC}}$	$T_{\text{PWRT}} + 1024 \cdot T_{\text{OSC}}$	$1024 \cdot T_{\text{OSC}}$	$1024 \cdot T_{\text{OSC}}$
EXTRC, INTRC	$T_{\text{PWRT}}$	$5\text{-}10 \mu\text{s}^{(1)}$	$T_{\text{PWRT}}$	$5\text{-}10 \mu\text{s}^{(1)}$	$5\text{-}10 \mu\text{s}^{(1)}$
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  $5 \mu\text{s}$ - $10 \mu\text{s}$  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	---- -10x
$\overline{\text{MCLR}}$ Reset during normal operation	000h	000u uuuu	---- -uuu
$\overline{\text{MCLR}}$ Reset during Sleep	000h	0001 0uuu	---- -uuu
WDT Reset	000h	0000 1uuu	---- -uuu
WDT Wake-up	PC + 1	uuu0 0uuu	---- -uuu
Brown-out Reset	000h	0001 1xxx	---- -1u0
Interrupt Wake-up from Sleep	PC + 1 <sup>(1)</sup>	uuu1 0uuu	---- -uuu

**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).

# PIC16F7X7

**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	xx0x 0000	uu0u 0000	uuuu uuuu
PORTB	xx00 0000	uu00 0000	uuuu uuuu
PORTC	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTD	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTE (PIC16F737/767)	---- x---	---- u---	---- u---
PORTE (PIC16F747/777)	---- x000	---- u000	---- uuuu
PCLATH	---0 0000	---0 0000	---u uuuu
INTCON	0000 000x	0000 000u	uuuu uuuu <sup>(1)</sup>
PIR1	0000 0000	0000 0000	uuuu uuuu <sup>(1)</sup>
PIR2	000- 0-00	000- 0-00	uuu- u-uu
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
SSPCON2	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
CCP2CON	--00 0000	--00 0000	--uu uuuu
CCP3CON	--00 0000	--00 0000	uuuu uuuu
CCPR2L	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCPR2H	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCPR3L	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCPR3H	xxxx xxxx	uuuu uuuu	uuuu uuuu
RCSTA	0000 000x	0000 000x	uuuu uuuu
TXREG	0000 0000	0000 0000	uuuu uuuu
RCREG	0000 0000	0000 0000	uuuu uuuu
ADRESH	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADCON0	0000 0000	0000 0000	uuuu uuuu
OPTION_REG	1111 1111	1111 1111	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.

**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
TRISA	1111 1111	1111 1111	uuuu uuuu
TRISB	1111 1111	1111 1111	uuuu uuuu
TRISC	1111 1111	1111 1111	uuuu uuuu
TRISD	1111 1111	1111 1111	uuuu uuuu
TRISE (PIC16F737/767)	---- 1---	---- u---	---- 1---
TRISE (PIC16F747/777)	0000 1111	0000 1111	uuuu uuuu
PIE1	0000 0000	0000 0000	-uuu uuuu
PIE2	000- 0-00	000- 0-00	uuu- u-uu
PCON	---- -1qq	---- -uuu	---- -uuu
OSCCON	-000 1000	-000 1000	-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
CMCON	0000 0111	0000 0111	uuuu uuuu
CVRCON	000- 0000	000- 0000	uuu- uuuu
WDTCON	---0 1000	---0 1000	---u uuuu
ADRESL	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADCON1	0000 0000	0000 0000	uuuu uuuu
ADCON2	--00 0---	--00 0---	uuuu uuuu
PMDATA	xxxx xxxx	uuuu uuuu	uuuu uuuu
PMADR	xxxx xxxx	uuuu uuuu	uuuu uuuu
PMDATH	--xx xxxx	--uu uuuu	--uu uuuu
PMADRH	---- xxxx	---- uuuu	---- uuuu
PMCON1	1--- ---0	1--- ---u	1--- ---u
LVDCON	--00 0101	--00 0101	--uu uuuu

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

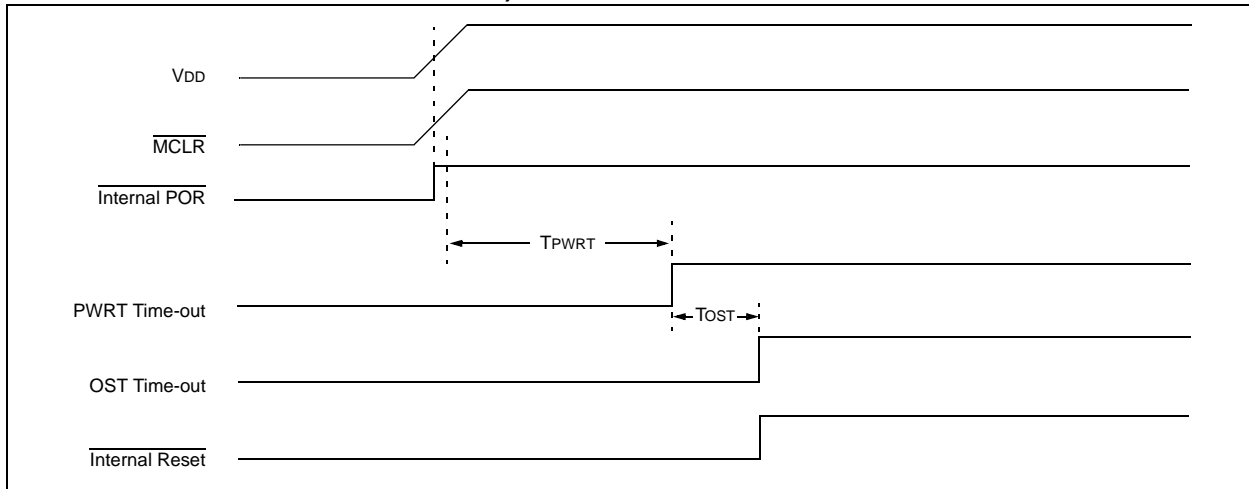
**Note 1:** One or more bits in INTCON, PIR1 and PR2 will be affected (to cause wake-up).

**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).

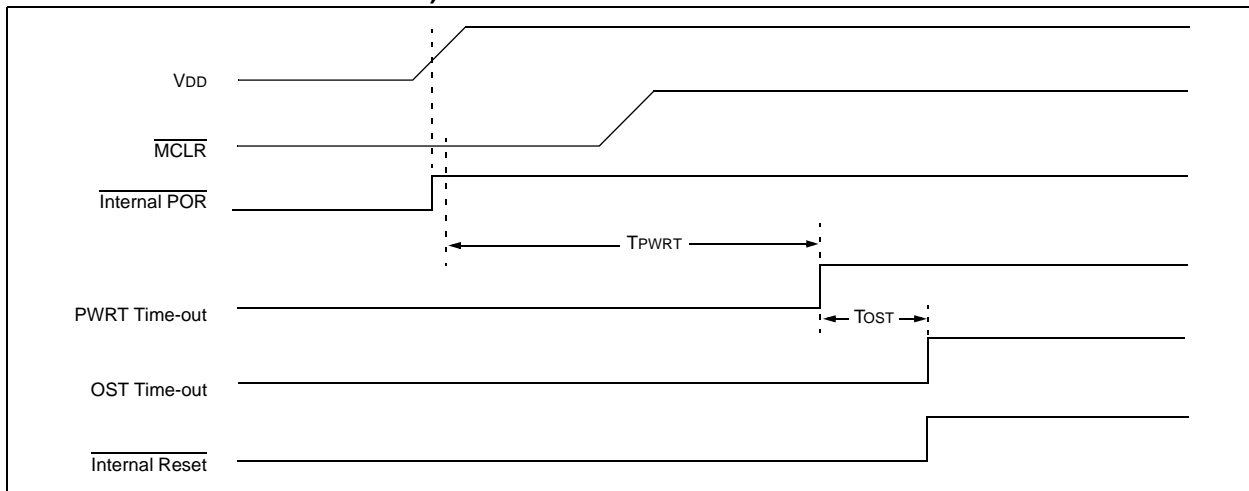
**3:** See Table 15-3 for Reset value for specific condition.

# PIC16F7X7

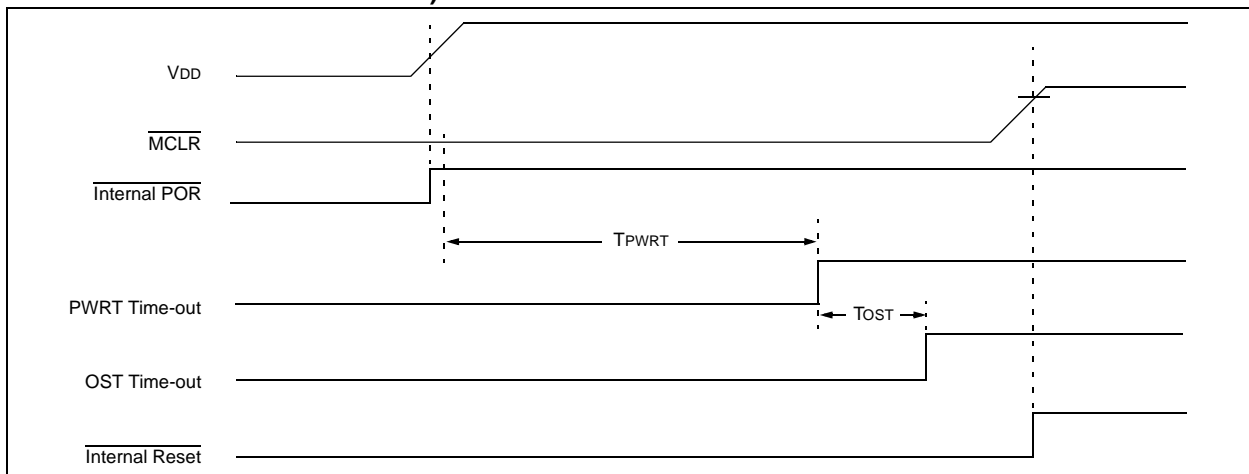
**FIGURE 15-7: TIME-OUT SEQUENCE ON POWER-UP ( $\overline{\text{MCLR}}$  TIED TO  $V_{\text{DD}}$  THROUGH PULL-UP RESISTOR)**



**FIGURE 15-8: TIME-OUT SEQUENCE ON POWER-UP ( $\overline{\text{MCLR}}$  TIED TO  $V_{\text{DD}}$  THROUGH RC NETWORK): CASE 1**

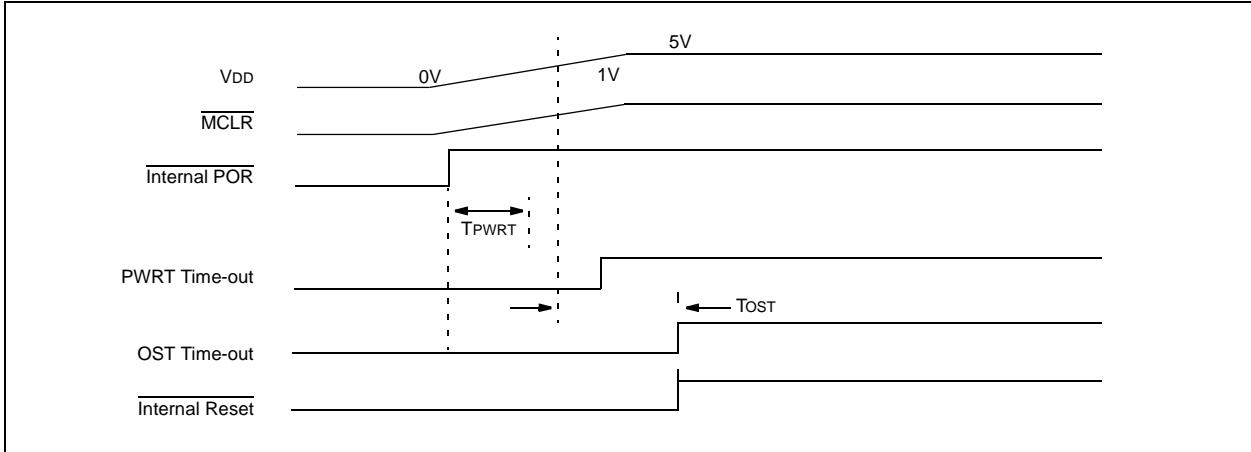


**FIGURE 15-9: TIME-OUT SEQUENCE ON POWER-UP ( $\overline{\text{MCLR}}$  TIED TO  $V_{\text{DD}}$  THROUGH RC NETWORK): CASE 2**





**FIGURE 15-10: SLOW RISE TIME ( $\overline{\text{MCLR}}$  TIED TO  $V_{\text{DD}}$  THROUGH RC NETWORK)**



# PIC16F7X7

## 15.15 Interrupts

The PIC16F7X7 has up to 17 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.

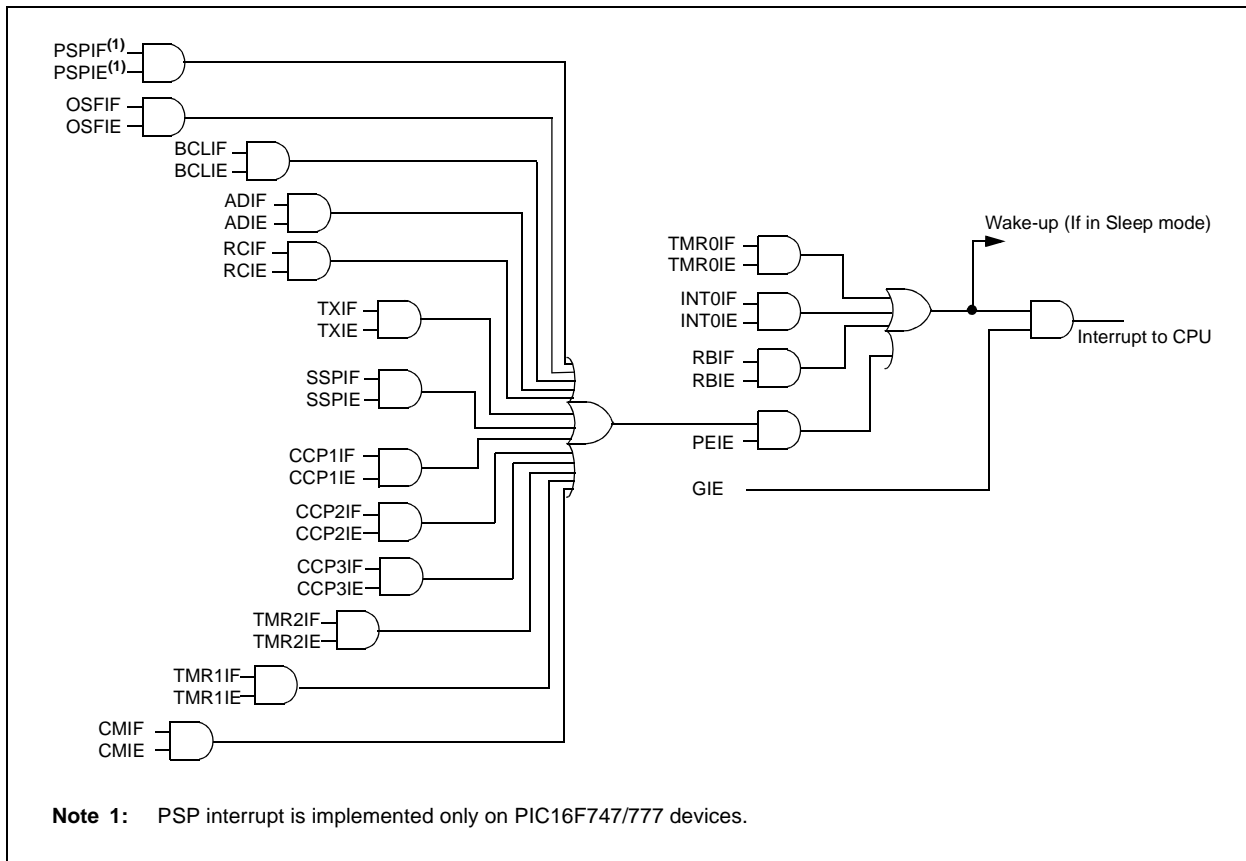
The RB0/INT pin interrupt, the RB port change interrupt and the TMR0 overflow interrupt flags are contained in the INTCON register.

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 15-11: INTERRUPT LOGIC



## 15.15.1 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 INTOIF (INTCON<1>) is set. This interrupt can be disabled by clearing enable bit, INTOIE (INTCON<4>). Flag bit INTOIF 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 INTOIE 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 15.18 “Power-Down Mode (Sleep)”** for details on Sleep mode.

## 15.15.2 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”**.

## 15.15.3 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 2.2 “Data Memory Organization”**.

## 15.16 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 PIC16F7X7 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 AND W REGISTERS IN RAM

```

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

# PIC16F7X7

## 15.17 Watchdog Timer (WDT)

For PIC16F7X7 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 the PSA bit is set to '1'.

### 15.17.1 WDT OSCILLATOR

The WDT derives its time base from the 31.25 kHz INTRC; therefore, the accuracy of the 31.25 kHz will be the same accuracy for the WDT time-out period.

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.

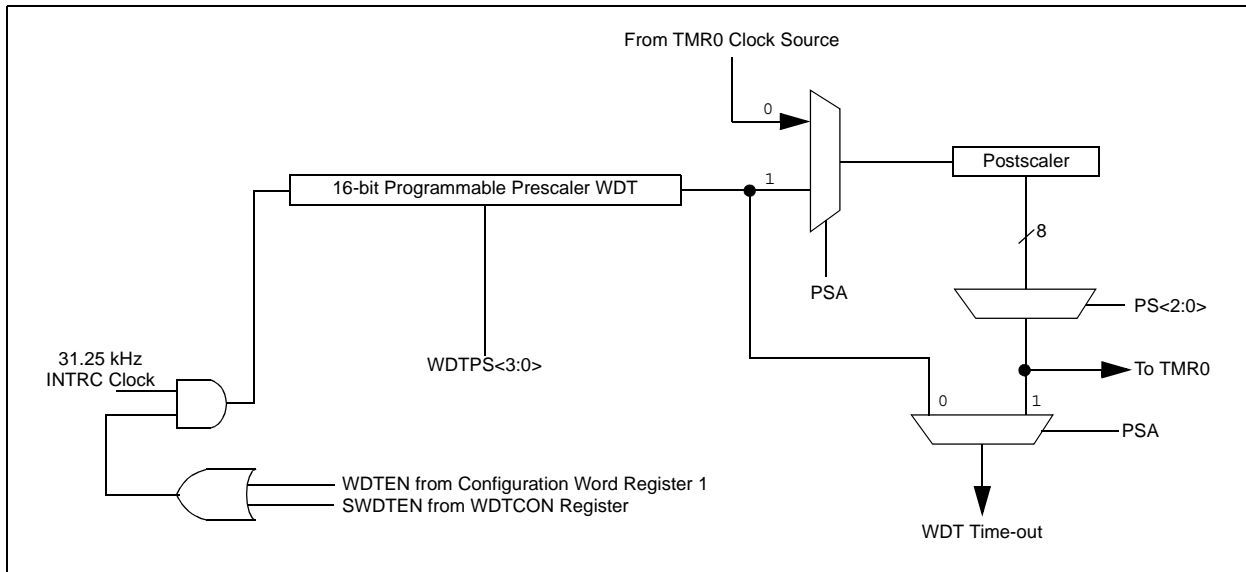
### 15.17.2 WDT CONTROL

The WDTEN bit is located in Configuration Word Register 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 Register 1 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) have the same function as in previous versions of the PIC16 family of microcontrollers.

**FIGURE 15-12: 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, EXTCLK		
Exit Sleep + System Clock = XT, HS, LP	Cleared at end of OST	Cleared at end of OST

## REGISTER 15-4: WDTCON: WATCHDOG TIMER 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	
bit 7								bit 0

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

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

0000 = 1:32 Prescale rate  
 0001 = 1:64 Prescale rate  
 0010 = 1:128 Prescale rate  
 0011 = 1:256 Prescale rate  
 0100 = 1:512 Prescale rate  
 0101 = 1:1024 Prescale rate  
 0110 = 1:2048 Prescale rate  
 0111 = 1:4096 Prescale rate  
 1000 = 1:8192 Prescale rate  
 1001 = 1:16394 Prescale rate  
 1010 = 1:32768 Prescale rate  
 1011 = 1:65536 Prescale rate  
 1100 = 1:1 Prescale rate

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 WDTEN configuration bit = 1, then WDT is always enabled irrespective of this control bit. If WDTEN 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	Value on POR, BOR	Value on all other Resets
81h, 181h	OPTION_REG	RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
2007h	Configuration bits <sup>(1)</sup>	BORV0	BOREN	MCLRE	FOSC2	PWRRTEN	WDTEN	FOSC1	FOSC0	1111 1111	1111 1111
105h	WDTCON	—	—	—	WDTPS3	WDTPS2	WDTPS1	WDTPS0	SWDTEN	---0 1000	---0 1000

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

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

# PIC16F7X7

## 15.17.3 TWO-SPEED CLOCK START-UP MODE

Two-Speed Start-up minimizes the latency between oscillator start-up and code execution that may be selected with the IESO (Internal/External Switchover) bit in Configuration Word Register 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 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 secondary 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 OSTs bit will confirm whether the primary clock configuration is engaged. If not, the OSTs 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.

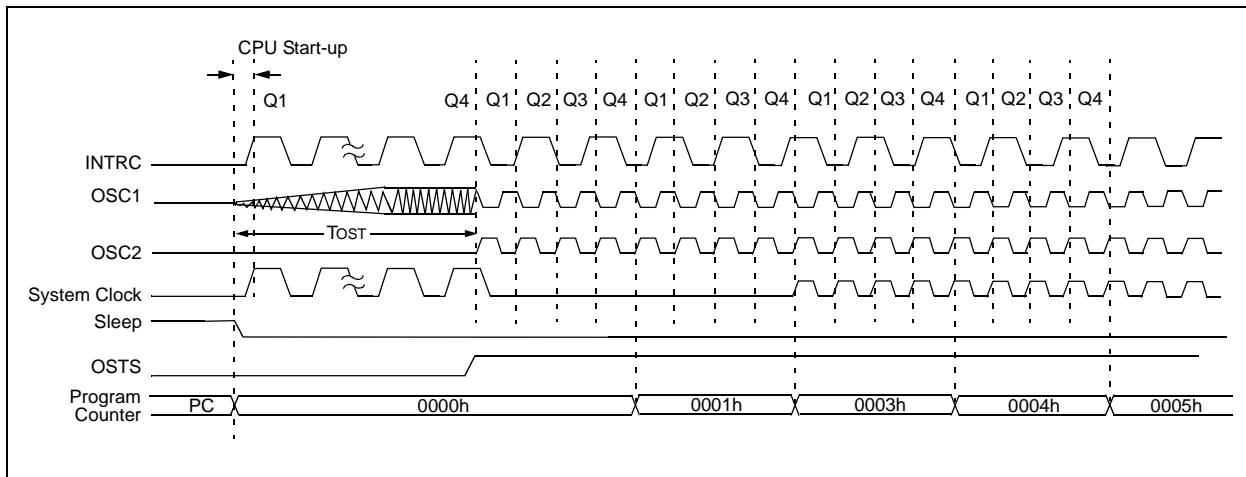
**Note:** Executing a SLEEP instruction will abort the oscillator start-up time and will cause the OSTs bit to remain clear.

### 15.17.3.1 Two-Speed Start-up Sequence

1. Wake-up from Sleep, Reset or POR.
2. OSCON 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. OSTs 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 OSTs bit to determine when the switchover takes place so that any software timing edges can be adjusted.

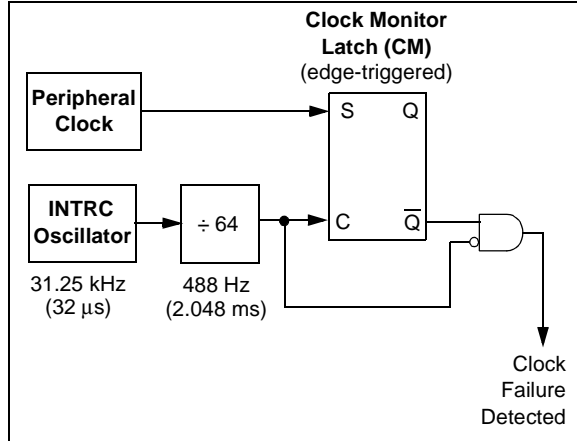
**FIGURE 15-13: TWO-SPEED START-UP**



## 15.17.4 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 15-14: FSCM BLOCK DIAGRAM**



The FSCM function is enabled by setting the FCMEN bit in Configuration Word Register 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 SCS bits of a different value.

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.

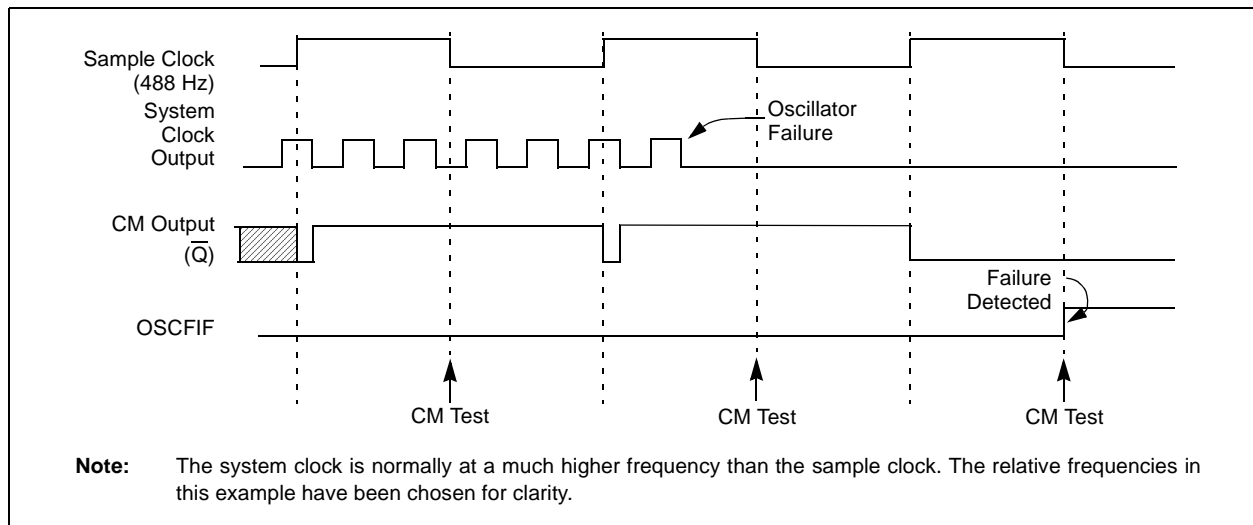
If Reset occurs while in Fail-Safe mode and the primary clock source is EC or RC, then the device will immediately switch back to EC or RC mode.

**Note:** Two-Speed Start-up is automatically enabled when the Fail-Safe option is enabled.

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

A change of SCS<1:0> or the 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 15-15: FSCM TIMING DIAGRAM**



# PIC16F7X7

## 15.17.4.2 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).

## 15.17.4.3 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 and PLL timers have 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 POR, 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.

## 15.18 Power-Down Mode (Sleep)

Power-Down mode is entered by executing a SLEEP instruction.

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

For lowest current consumption in this mode, place all I/O pins at either VDD or VSS, ensure no external circuitry is drawing current from the I/O pin, power-down the A/D and disable external clocks. Pull all I/O pins that are high-impedance inputs, high or low externally, to avoid switching currents caused by floating inputs. The T0CKI input should also be at VDD or VSS for lowest current consumption. The contribution from on-chip pull-ups on PORTB should also be considered.

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

## 15.18.1 WAKE-UP FROM SLEEP

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

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

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

The following peripheral interrupts can wake the device from Sleep:

1. TMR1 interrupt. Timer1 must be operating as an asynchronous counter.
2. CCP Capture mode interrupt.
3. Special event trigger (Timer1 in Asynchronous mode using an external clock).
4. SSP (Start/Stop) bit detect interrupt.
5. SSP transmit or receive in Slave mode (SPI/I<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.



## 15.18.2 WAKE-UP USING INTERRUPTS

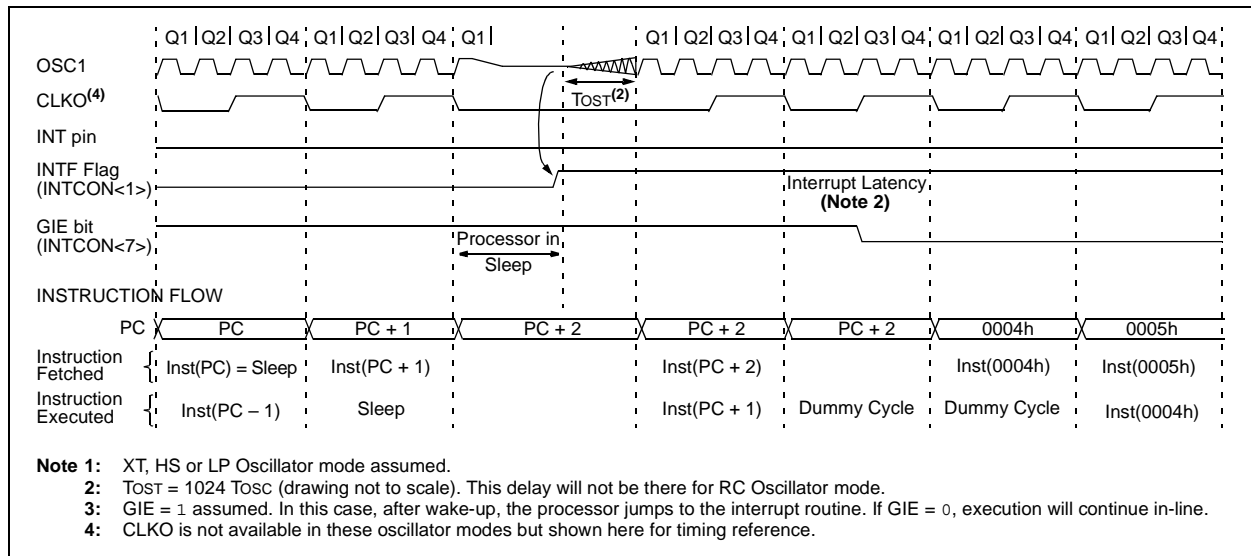
When global interrupts are disabled (GIE cleared) and any interrupt source has both its interrupt enable bit and interrupt flag bit set, one of the following will occur:

- If the interrupt occurs **before** the execution of a `SLEEP` instruction, the `SLEEP` instruction will complete as a `NOP`. Therefore, the `WDT` and `WDT` prescaler and `postscaler` (if enabled) will not be cleared, the `TO` bit will not be set and the `PD` bit will not be cleared.
- If the interrupt occurs **during or after** the execution of a `SLEEP` instruction, the device will immediately wake-up from Sleep. The `SLEEP` instruction will be completely executed before the wake-up. Therefore, the `WDT` and `WDT` prescaler and `postscaler` (if enabled) will be cleared, the `TO` bit will be set and the `PD` bit will be cleared.

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

To ensure that the `WDT` is cleared, a `CLRWDT` instruction should be executed before a `SLEEP` instruction.

**FIGURE 15-16: WAKE-UP FROM SLEEP THROUGH INTERRUPT**



# PIC16F7X7

## 15.19 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) 0x165-0x16F

To use the In-Circuit Debugger function of the microcontroller, the design must implement In-Circuit Serial Programming connections to MCLR/VPP, VDD, GND, RB7 and RB6. This will interface to the In-Circuit Debugger module available from Microchip or one of the third party development tool companies.

**Note:** In-Circuit Debugger operation must occur between the operating voltage range (VDD) of 4.75V-5.25V on PIC16F7X7 devices.

## 15.20 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.

## 15.21 ID Locations

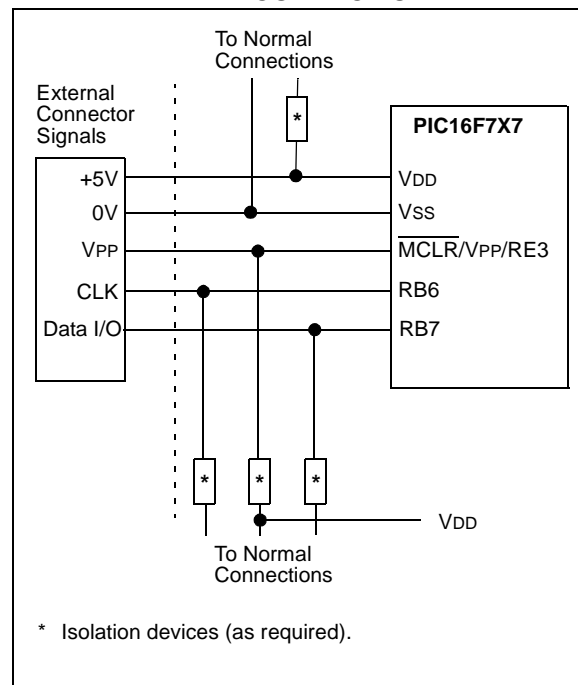
Four memory locations (2000h-2003h) are designated as ID locations where the user can store checksum or other code identification numbers. These locations are not accessible during normal execution but are readable and writable during program/verify. It is recommended that only the four Least Significant bits of the ID location are used.

## 15.22 In-Circuit Serial Programming

PIC16F7X7 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 15-17 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 general information of serial programming, please refer to the "In-Circuit Serial Programming™ (ICSP™) Guide" (DS30277).

**FIGURE 15-17: TYPICAL IN-CIRCUIT SERIAL PROGRAMMING™ CONNECTION**



## 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 13-2 lists the instructions recognized by the MPASM™ Assembler. A complete description of each instruction is also available in the “PICmicro® Mid-Range MCU Family Reference Manual” (DS33023).

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

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

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

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

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

**Note:** To maintain upward compatibility with future PIC16F7X7 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

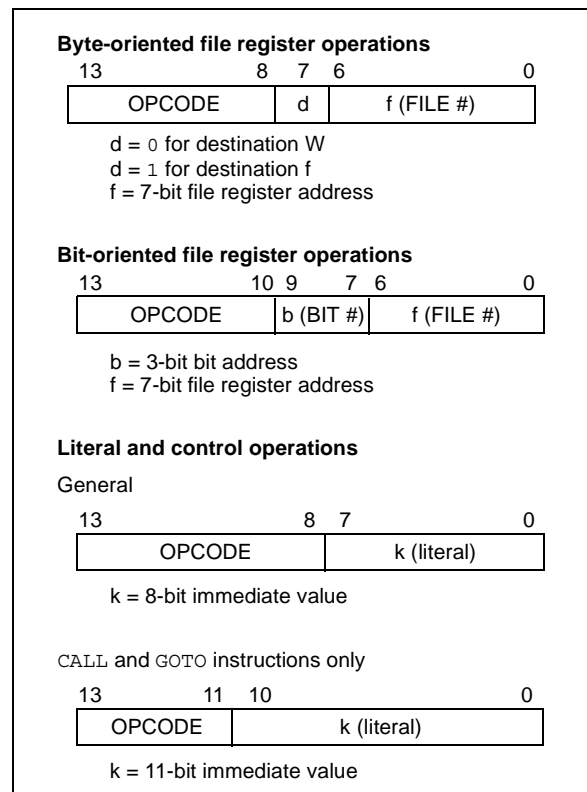
Any instruction that specifies a file register as part of the instruction performs a Read-Modify-Write (R-M-W) operation. The register is read, the data is modified and the result is stored according to either the instruction or the destination designator ‘d’. A read operation is performed on a register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

For example, a “CLRF PORTB” instruction will read PORTB, clear all the data bits, then write the result back to PORTB. This example would have the unintended result that the condition that sets the RBIF flag would be cleared for pins configured as inputs and using the PORTB interrupt-on-change feature.

**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**



# PIC16F7X7

**TABLE 16-2: PIC16F7X7 INSTRUCTION SET**

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

- Note 1:** When an I/O register is modified as a function of itself ( e.g., `MOVF PORTB, 1` ), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as input and is driven low by an external device, the data will be written back with a '0'.
- 2:** If this instruction is executed on the TMR0 register (and where applicable, d = 1), the prescaler will be cleared if assigned to the Timer0 module.
- 3:** If 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.

### **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.

### **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' is '0', the result is stored in the W register. If 'd' is '1', the result is stored back in register 'f'.

### **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.

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

**Syntax:**      [ *label* ] ANDLW    *k*

**Operands:**     $0 \leq k \leq 255$

**Operation:**     $(W) .\text{AND.} (k) \rightarrow (W)$

**Status Affected:**    Z

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

### **BTFSS**         **Bit Test f, Skip if Set**

**Syntax:**      [ *label* ] BTFSS    *f*,*b*

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

**Operation:**    skip if  $(f<b>) = 1$

**Status Affected:**    None

**Description:**    If bit 'b' in register 'f' is '0', the next instruction is executed. If bit 'b' is '1', then the next instruction is discarded and a NOP is executed instead, making this a 2 Tcy instruction.

### **ANDWF**        **AND W with f**

**Syntax:**      [ *label* ] ANDWF    *f*,*d*

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

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

**Status Affected:**    Z

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

### **BTFSC**         **Bit Test, Skip if Clear**

**Syntax:**      [ *label* ] BTFSC    *f*,*b*

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

**Operation:**    skip if  $(f<b>) = 0$

**Status Affected:**    None

**Description:**    If bit 'b' in register 'f' is '1', the next instruction is executed. If bit 'b' in register 'f' is '0', the next instruction is discarded and a NOP is executed instead, making this a 2 Tcy instruction.

# PIC16F7X7

---

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

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

---

## DECFSZ      Decrement f, Skip if 0

---

Syntax:            [ *label* ] DECFSZ f,d

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

Operation:         $(f) - 1 \rightarrow (\text{destination})$ ;  
 skip if result = 0

Status Affected: None

Description:     The contents of register 'f' are decremented. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'.  
 If the result is '1', the next instruction is executed. If the result is '0', then a NOP is executed instead, making it a 2 Tcy instruction.

---

## INCFSZ      Increment f, Skip if 0

---

Syntax:            [ *label* ] INCFSZ f,d

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

Operation:         $(f) + 1 \rightarrow (\text{destination})$ ,  
 skip if result = 0

Status Affected: None

Description:     The contents of register 'f' are incremented. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'.  
 If the result is '1', the next instruction is executed. If the result is '0', a NOP is executed instead, making it a 2 Tcy instruction.

---

## GOTO        Unconditional Branch

---

Syntax:            [ *label* ] 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.

---

## 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 ORed with the eight-bit literal 'k'. The result is placed in the W register.

---

## INCF        Increment f

---

Syntax:            [ *label* ] INCF f,d

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

Operation:         $(f) + 1 \rightarrow (\text{destination})$

Status Affected: Z

Description:     The contents of register 'f' are incremented. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'.

---

## IORWF      Inclusive OR W with f

---

Syntax:            [ *label* ] IORWF f,d

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

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

Status Affected: Z

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

# PIC16F7X7

---

## **MOVF**      **Move f**

---

Syntax:      [ *label* ] MOVF f,d  
Operands:     $0 \leq f \leq 127$   
               $d \in [0,1]$   
Operation:    (f) → (destination)  
Status Affected: Z  
Description:    The contents of register 'f' are moved to a destination dependant upon the status of 'd'. If  $d = 0$ , 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.

## **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.

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

---

Syntax:      [ *label* ] RETFIE  
Operands:    None  
Operation:    TOS → PC,  
              1 → GIE  
Status Affected: None

## **MOVWF**      **Move W to f**

---

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

## **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.



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

---

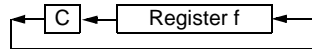
Syntax:                [ *label* ] RLF f,d

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' is '0', the result is placed in the W register. If 'd' is '1', the result is stored back in register 'f'.



**SLEEP**                    **Enter Sleep mode**

---

Syntax:                [ *label* ] SLEEP

Operands:            None

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

Status Affected:     $\overline{TO}$ , 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.

**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.

**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 (2's complement method) from the eight-bit literal 'k'. The result is placed in the W register.

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

---

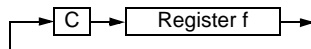
Syntax:                [ *label* ] RRF f,d

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' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'.



**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 (2's complement method) W register from register 'f'. If 'd' is '0', the result is stored in the W register. If 'd' is '1', the result is stored back in register 'f'.

# PIC16F7X7

---

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

---

Syntax:      [ *label* ] SWAPF f,d

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

Operation:    (f<3:0>) → (destination<7:4>),  
              (f<7:4>) → (destination<3:0>)

Status Affected: None

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

Status Affected: Z

Description:    Exclusive OR the contents of the W register with register 'f'. If 'd' is '0', the result is stored in the W register. If 'd' is '1', the result is stored back in register 'f'.

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

---

Syntax:      [ *label* ] XORLW k

Operands:     $0 \leq k \leq 255$

Operation:    (W) .XOR. k → (W)

Status Affected: Z

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

## 17.0 DEVELOPMENT SUPPORT

The PICmicro<sup>®</sup> microcontrollers are supported with a full range of hardware and software development tools:

- Integrated Development Environment
  - MPLAB<sup>®</sup> IDE Software
- Assemblers/Compilers/Linkers
  - MPASM<sup>™</sup> Assembler
  - MPLAB C17 and MPLAB C18 C Compilers
  - MPLINK<sup>™</sup> Object Linker/  
MPLIB<sup>™</sup> 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<sup>®</sup> II Universal Device Programmer
  - PICSTART<sup>®</sup> Plus Development Programmer
  - MPLAB PM3 Device Programmer
- Low-Cost Demonstration Boards
  - PICDEM<sup>™</sup> 1 Demonstration Board
  - PICDEM.net<sup>™</sup> 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<sup>®</sup> Evaluation and Programming Tools
  - PICDEM MSC
  - microID<sup>®</sup> Developer Kits
  - CAN
  - PowerSmart<sup>®</sup> 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<sup>®</sup> 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<sup>®</sup> 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

# PIC16F7X7

---

## 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/library features include:

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

## 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.

# PIC16F7X7

---

## 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.

# PIC16F7X7

---

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, $\overline{\text{MCLR}}$ and RA4) .....	-0.3V to (VDD + 0.3V)
Voltage on VDD with respect to VSS .....	-0.3 to +6.5V
Voltage on $\overline{\text{MCLR}}$ with respect to VSS ( <b>Note 2</b> ) .....	0 to +13.5V
Voltage on RA4 with respect to VSS.....	0 to +12V
Total power dissipation ( <b>Note 1</b> ) .....	1.0W
Maximum current out of VSS pin .....	300 mA
Maximum current into VDD pin .....	250 mA
Input clamp current, I <sub>IK</sub> (V <sub>I</sub> < 0 or V <sub>I</sub> > V <sub>DD</sub> ).....	±20 mA
Output clamp current, I <sub>OK</sub> (V <sub>O</sub> < 0 or V <sub>O</sub> > V <sub>DD</sub> ) .....	±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, PORTB and PORTE (combined) ( <b>Note 3</b> ).....	200 mA
Maximum current sourced by PORTA, PORTB and PORTE (combined) ( <b>Note 3</b> ).....	200 mA
Maximum current sunk by PORTC and PORTD (combined) ( <b>Note 3</b> ) .....	200 mA
Maximum current sourced by PORTC and PORTD (combined) ( <b>Note 3</b> ) .....	200 mA

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

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

**3:** PORTD and PORTE are not implemented on the PIC16F737/767 devices.

† 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.

# PIC16F7X7

FIGURE 18-1: PIC16F7X7 VOLTAGE-FREQUENCY GRAPH (INDUSTRIAL, EXTENDED)

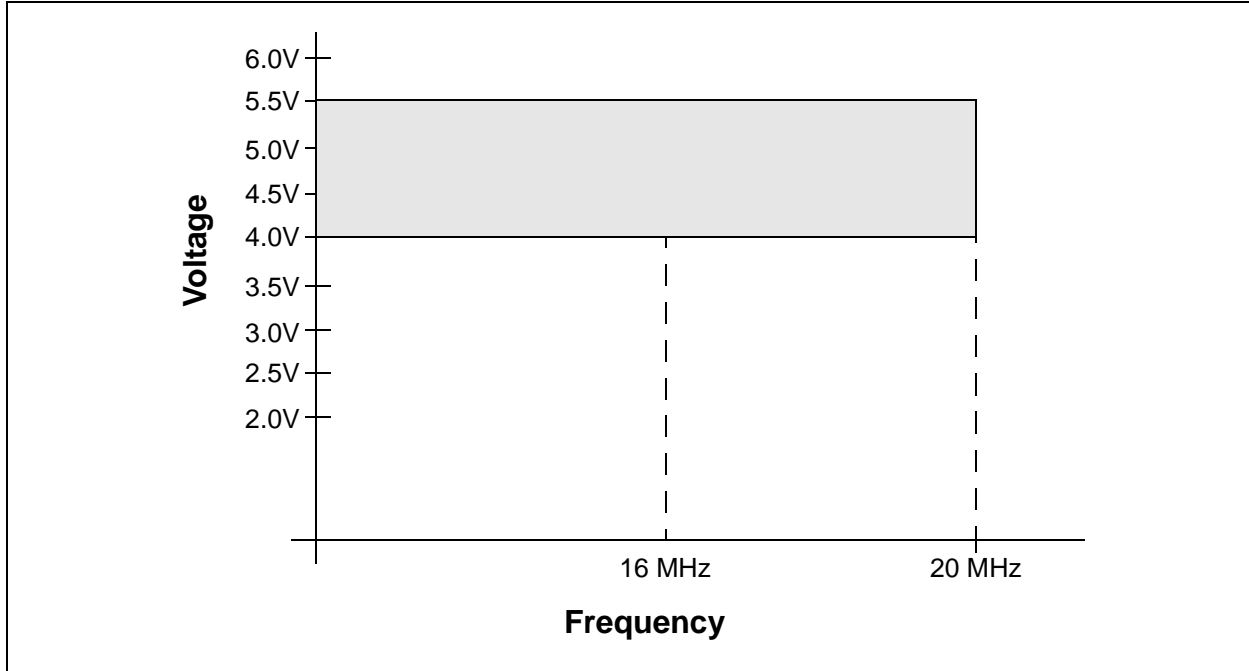
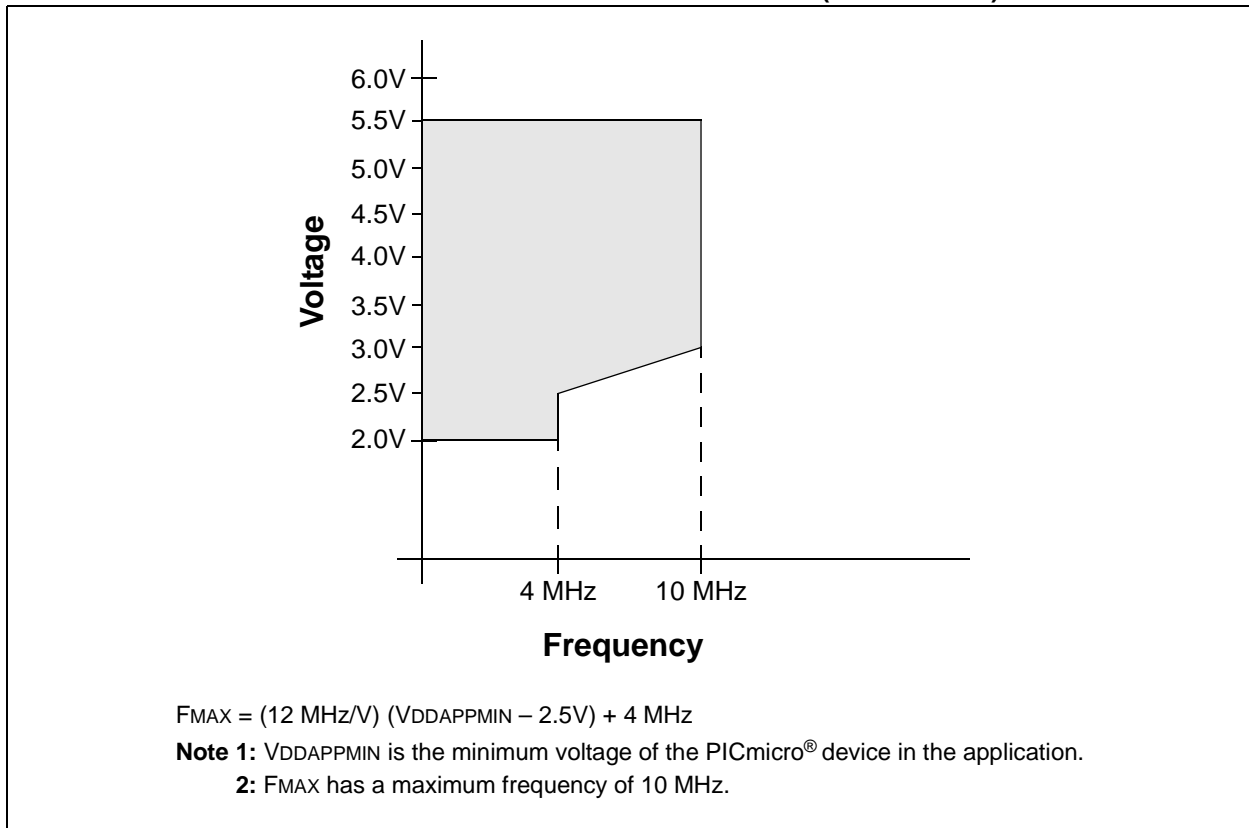


FIGURE 18-2: PIC16LF7X7 VOLTAGE-FREQUENCY GRAPH (INDUSTRIAL)



## 18.1 DC Characteristics: PIC16F737/747/767/777 (Industrial, Extended) PIC16LF737/747/767/777 (Industrial)

PIC16LF737/747/767/777 (Industrial)		Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial						
PIC16F737/747/767/777 (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.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions	
D001	VDD	<b>Supply Voltage</b>						
		PIC16LF7X7	2.5	—	5.5	V	A/D in use, $-40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$	
			2.2	—	5.5	V	A/D in use, $0^{\circ}\text{C}$ to $+85^{\circ}\text{C}$	
D001 D001A		PIC16F7X7	4.0	—	5.5	V	All configurations	
			VBOR*	—	5.5	V	BOR enabled ( <b>Note 6</b> )	
D002*	VDR	<b>RAM Data Retention Voltage (Note 1)</b>	—	1.5	—	V		
D003	VPOR	<b>VDD Start Voltage</b> to ensure internal Power-on Reset signal	—	VSS	—	V	See section on Power-on Reset for details	
D004*	SVDD	<b>VDD Rise Rate</b> to ensure internal Power-on Reset signal	0.05	—	—	V/ms	See section on Power-on Reset for details	
D005	VBOR	<b>Brown-out Reset Voltage</b>						
		PIC16LF7X7						
		BORV1:BORV0 = 11	1.96	2.06	2.16	V	$85^{\circ}\text{C} \geq T \geq 25^{\circ}\text{C}$	
		BORV1:BORV0 = 10	2.64	2.78	2.92	V		
		BORV1:BORV0 = 01	4.11	4.33	4.55	V		
		BORV1:BORV0 = 00	4.41	4.64	4.87	V		
D005		PIC16F7X7	Industrial					
		BORV1:BORV0 = 1x	N.A.	—	N.A.	V	Not in operating voltage range of device	
		BORV1:BORV0 = 01	4.16	—	4.5	V		
		BORV1:BORV0 = 00	4.45	—	4.83	V		
D005		PIC16F7X7	Extended					
		BORV1:BORV0 = 1x	N.A.	—	N.A.	V	Not in operating voltage range of device	
		BORV1:BORV0 = 01	4.07	—	4.59	V		
		BORV1:BORV0 = 00	4.36	—	4.92	V		

**Legend:** Shading of rows is to assist in readability of of the table.

\* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V,  $25^{\circ}\text{C}$  unless otherwise stated. These parameters are for design guidance only and are not tested.

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

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

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.

**Note 3:** 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 and VSS.

**Note 4:** For RC oscillator configuration, 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.

**Note 5:** The Δ current is the additional current consumed when this peripheral is enabled. This current should be added to the base IDD or IPD measurement.

**Note 6:** When BOR is enabled, the device will operate correctly until the VBOR voltage trip point is reached.

# PIC16F7X7

## 18.2 DC Characteristics: Power-Down and Supply Current PIC16F737/747/767/777 (Industrial, Extended) PIC16LF737/747/767/777 (Industrial)

PIC16LF737/747/767/777 (Industrial)		Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial				
PIC16F737/747/767/777 (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>Power-Down Current (IPD)<sup>(1)</sup></b>						
	PIC16LF7X7	0.1	0.4	μA	-40°C	VDD = 2.0V
		0.1	0.4	μA	+25°C	
		0.4	1.5	μA	+85°C	
	PIC16LF7X7	0.3	0.5	μA	-40°C	VDD = 3.0V
		0.3	0.5	μA	+25°C	
		0.7	1.7	μA	+85°C	
	All devices	0.6	1.0	μA	-40°C	VDD = 5.0V
		0.6	1.0	μA	+25°C	
		1.2	5.0	μA	+85°C	
	Extended devices	6	28	μA	+125°C	

**Legend:** Shading of rows is to assist in readability of the table.

- Note 1:** The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in high-impedance state and tied to VDD or VSS and all features that add delta current disabled (such as WDT, Timer1 Oscillator, BOR, etc.).
- 2:** The supply current is mainly a function of operating voltage, frequency and mode. Other factors, such as I/O pin loading and switching rate, oscillator type and circuit, internal code execution pattern and temperature, also have an impact on the current consumption.  
The test conditions for all IDD measurements in active operation mode are:  
OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to VDD;  
MCLR = VDD; WDT enabled/disabled as specified.
- 3:** For RC oscillator configurations, current through REXT is not included. The current through the resistor can be estimated by the formula  $I_r = V_{DD}/2R_{EXT}$  (mA) with REXT in kΩ.

## 18.2 DC Characteristics: Power-Down and Supply Current PIC16F737/747/767/777 (Industrial, Extended) PIC16LF737/747/767/777 (Industrial) (Continued)

PIC16LF737/747/767/777 (Industrial)		Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial				
PIC16F737/747/767/777 (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>						
	PIC16LF7X7	9	20	μA	-40°C	VDD = 2.0V
		7	15	μA	+25°C	
		7	15	μA	+85°C	
	PIC16LF7X7	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	
	PIC16LF7X7	72	95	μA	-40°C	VDD = 2.0V
		76	90	μA	+25°C	
		76	90	μA	+85°C	
	PIC16LF7X7	138	175	μA	-40°C	VDD = 3.0V
		136	170	μA	+25°C	
		136	170	μA	+85°C	
	All devices	310	380	μA	-40°C	VDD = 5.0V
		290	360	μA	+25°C	
		280	360	μA	+85°C	
	Extended devices	330	500	μA	+125°C	
		FOSC = 32 kHz (LP Oscillator)				
		FOSC = 1 MHz (RC Oscillator) <sup>(3)</sup>				

**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.

# PIC16F7X7

## 18.2 DC Characteristics: Power-Down and Supply Current PIC16F737/747/767/777 (Industrial, Extended) PIC16LF737/747/767/777 (Industrial) (Continued)

PIC16LF737/747/767/777 (Industrial)		Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial					
PIC16F737/747/767/777 (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>							
	PIC16LF7X7	270	315	$\mu\text{A}$	$-40^{\circ}\text{C}$	$V_{DD} = 2.0\text{V}$  $V_{DD} = 3.0\text{V}$  $V_{DD} = 5.0\text{V}$  $V_{DD} = 4.0\text{V}$  $V_{DD} = 5.0\text{V}$	$F_{OSC} = 4\text{ MHz}$ (RC Oscillator) <sup>(3)</sup>
		280	310	$\mu\text{A}$	$+25^{\circ}\text{C}$		
		285	310	$\mu\text{A}$	$+85^{\circ}\text{C}$		
	PIC16LF7X7	460	610	$\mu\text{A}$	$-40^{\circ}\text{C}$		
		450	600	$\mu\text{A}$	$+25^{\circ}\text{C}$		
		450	600	$\mu\text{A}$	$+85^{\circ}\text{C}$		
	All devices	900	1060	$\mu\text{A}$	$-40^{\circ}\text{C}$		
		890	1050	$\mu\text{A}$	$+25^{\circ}\text{C}$		
		890	1050	$\mu\text{A}$	$+85^{\circ}\text{C}$		
	Extended devices	.920	1.5	$\text{mA}$	$+125^{\circ}\text{C}$		
	All devices	1.8	2.3	$\text{mA}$	$-40^{\circ}\text{C}$	$F_{OSC} = 20\text{ MHz}$ (HS Oscillator)	
		1.6	2.2	$\text{mA}$	$+25^{\circ}\text{C}$		
		1.3	2.2	$\text{mA}$	$+85^{\circ}\text{C}$		
	All devices	3.0	4.2	$\text{mA}$	$-40^{\circ}\text{C}$		
		2.5	4.0	$\text{mA}$	$+25^{\circ}\text{C}$		
		2.5	4.0	$\text{mA}$	$+85^{\circ}\text{C}$		
	Extended devices	3.0	5.0	$\text{mA}$	$+125^{\circ}\text{C}$		

**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:  
 $OSC1 =$  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  $R_{EXT}$  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 PIC16F737/747/767/777 (Industrial, Extended) PIC16LF737/747/767/777 (Industrial) (Continued)

PIC16F737/747/767/777 (Industrial)		Standard Operating Conditions (unless otherwise stated)					
		Operating temperature		-40°C ≤ TA ≤ +85°C for industrial			
PIC16F737/747/767/777 (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>							
	PIC16LF7X7	8	20	μA	-40°C	VDD = 2.0V	FOSC = 31.25 kHz (RC_RUN mode, Internal RC Oscillator)
		7	15	μA	+25°C		
		7	15	μA	+85°C		
PIC16LF7X7	16	30	μA	-40°C	VDD = 3.0V		
	14	25	μA	+25°C			
	14	25	μA	+85°C			
All devices		32	40	μA	-40°C	VDD = 5.0V	
		29	35	μA	+25°C		
		29	35	μA	+85°C		
Extended devices		35	45	μA	+125°C		
PIC16LF7X7		132	160	μA	-40°C	VDD = 2.0V	FOSC = 1 MHz (RC_RUN mode, Internal RC Oscillator)
		126	155	μA	+25°C		
		126	155	μA	+85°C		
PIC16LF7X7		260	310	μA	-40°C	VDD = 3.0V	
		230	300	μA	+25°C		
		230	300	μA	+85°C		
All devices		560	690	μA	-40°C	VDD = 5.0V	
		500	650	μA	+25°C		
		500	650	μA	+85°C		
Extended devices		570	710	μA	+125°C		
PIC16LF7X7		310	420	μA	-40°C	VDD = 2.0V	FOSC = 4 MHz (RC_RUN mode, Internal RC Oscillator)
		300	410	μA	+25°C		
		300	410	μA	+85°C		
PIC16LF7X7		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.

# PIC16F7X7

## 18.2 DC Characteristics: Power-Down and Supply Current PIC16F737/747/767/777 (Industrial, Extended) PIC16LF737/747/767/777 (Industrial) (Continued)

PIC16LF737/747/767/777 (Industrial)		Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial						
PIC16F737/747/767/777 (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>								
	PIC16LF7X7	.950	1.3	mA	-40°C	V <sub>DD</sub> = 3.0V	F <sub>OSC</sub> = 8 MHz ( <b>RC_RUN</b> 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			
	PIC16LF7X7	9	13	μA	-10°C	V <sub>DD</sub> = 2.0V		F <sub>OSC</sub> = 32 kHz ( <b>SEC_RUN</b> mode, Timer1 as Clock)
		9	14	μA	+25°C			
11		16	μA	+70°C				
PIC16LF7X7	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 PIC16F737/747/767/777 (Industrial, Extended) PIC16LF737/747/767/777 (Industrial) (Continued)

PIC16LF737/747/767/777 (Industrial)		Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial				
PIC16F737/747/767/777 (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 (ΔI <sub>WDT</sub> )	<b>Watchdog Timer</b>	1.5	3.8	μA	-40°C	VDD = 2.0V
		2.2	3.8	μA	+25°C	
		2.7	4.0	μA	+85°C	
		2.3	4.6	μA	-40°C	VDD = 3.0V
		2.7	4.6	μA	+25°C	
		3.1	4.8	μA	+85°C	
		3.0	10.0	μA	-40°C	VDD = 5.0V
		3.3	10.0	μA	+25°C	
		3.9	13.0	μA	+85°C	
		Extended devices		5.0	21.0	μA
D022A (ΔI <sub>BOR</sub> )	<b>Brown-out Reset</b>	17	35	μA	-40°C to +85°C	VDD = 3.0V
		47	45	μA	-40°C to +85°C	VDD = 5.0V
		0	0	μA	-40°C to +85°C	VDD = 2.0V VDD = 3.0V VDD = 5.0V
	Extended devices		48	50	μA	-40°C to +125°C
D022B (ΔI <sub>LVD</sub> )	<b>Low-Voltage Detect</b>	14	25	μA	-40°C to +85°C	VDD = 2.0V
		18	35	μA	-40°C to +85°C	VDD = 3.0V
		21	45	μA	-40°C to +85°C	VDD = 5.0V
	Extended devices		24	50	μA	-40°C to +125°C

**Legend:** Shading of rows is to assist in readability of the table.

**Note 1:** The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in high-impedance state and tied to VDD or VSS and all features that add delta current disabled (such as WDT, Timer1 Oscillator, BOR, etc.).

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

The test conditions for all IDD measurements in active operation mode are:

OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to VDD;

MCLR = VDD; WDT enabled/disabled as specified.

**3:** For RC oscillator configurations, current through 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.

# PIC16F7X7

## 18.2 DC Characteristics: Power-Down and Supply Current PIC16F737/747/767/777 (Industrial, Extended) PIC16LF737/747/767/777 (Industrial) (Continued)

PIC16LF737/747/767/777 (Industrial)		Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial					
PIC16F737/747/767/777 (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		
D025 (ΔIOSCB)	Timer1 Oscillator	Module Differential Currents (ΔI <sub>WDT</sub> , ΔI <sub>BOR</sub> , ΔI <sub>LVD</sub> , ΔI <sub>OSCB</sub> , ΔI <sub>AD</sub> )					
		1.7	2.3	μA	-40°C	VDD = 2.0V	32 kHz on Timer1
		1.8	2.3	μA	+25°C		
		2.0	2.3	μA	+85°C		
		2.2	3.8	μA	-40°C	VDD = 3.0V	
		2.6	3.8	μA	+25°C		
		2.9	3.8	μA	+85°C		
		3.0	6.0	μA	-40°C	VDD = 5.0V	
		3.2	6.0	μA	+25°C		
		3.4	7.0	μA	+85°C		
D026 (ΔIAD)	A/D Converter	0.001	2.0	μA	-40°C to +85°C	VDD = 2.0V	A/D on, Sleep, not converting
		0.001	2.0	μA	-40°C to +85°C	VDD = 3.0V	
		0.003	2.0	μA	-40°C to +85°C	VDD = 5.0V	
	Extended devices	4	8	mA	-40°C to +125°C	VDD = 5.0V	

**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.

## 18.3 DC Characteristics: Internal RC Accuracy PIC16F737/747/767/777 (Industrial, Extended) PIC16LF737/747/767/777 (Industrial)

<b>PIC16LF737/747/767/777</b> (Industrial)		<b>Standard Operating Conditions (unless otherwise stated)</b> Operating temperature -40°C ≤ TA ≤ +85°C for industrial					
<b>PIC16F737/747/767/777</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	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>							
	PIC16LF7X7	-2	±1	2	%	+25°C	VDD = 2.7V-3.3V
		-5	—	5	%	-10°C to +85°C	
		-10	—	10	%	-40°C to +85°C	
	PIC16F7X7	-2	±1	2	%	+25°C	VDD = 4.5V-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	
<b>INTRC Accuracy @ Freq = 31 kHz<sup>(2)</sup></b>							
	PIC16LF7X7	26.562	—	35.938	kHz	-40°C to +85°C	VDD = 2.7V-3.3V
	PIC16F7X7	26.562	—	35.938	kHz	-40°C to +85°C	VDD = 4.5V-5.5V

**Legend:** Shading of rows is to assist in readability of the table.

**Note 1:** Frequency calibrated at 25°C. OSCUNE register can be used to compensate for temperature drift.

**2:** INTRC is used to calibrate INTOSC.

# PIC16F7X7

## 18.4 DC Characteristics: PIC16F737/747/767/777 (Industrial, Extended) PIC16LF737/747/767/777 (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 Section 18.1 "DC Characteristics".				
Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
D030	$V_{IL}$	<b>Input Low Voltage</b> I/O ports: with TTL buffer	$V_{SS}$	—	0.15 $V_{DD}$	V	For entire $V_{DD}$ range
D030A			$V_{SS}$	—	0.8V	V	$4.5\text{V} \leq V_{DD} \leq 5.5\text{V}$
D031		with Schmitt Trigger buffer	$V_{SS}$	—	0.2 $V_{DD}$	V	
D032		$\overline{\text{MCLR}}$ , OSC1 (in RC mode)	$V_{SS}$	—	0.2 $V_{DD}$	V	
D033		OSC1 (in XT and LP modes) OSC1 (in HS mode)	$V_{SS}$	—	0.3V 0.3 $V_{DD}$	V	(Note 1)
D034		Ports RC3 and RC4: with Schmitt Trigger buffer	$V_{SS}$	—	0.3 $V_{DD}$	V	For entire $V_{DD}$ range
D034A		with SMBus	-0.5	—	0.6	V	For $V_{DD} = 4.5$ to 5.5V
D040	$V_{IH}$	<b>Input High Voltage</b> I/O ports: with TTL buffer	2.0	—	$V_{DD}$	V	$4.5\text{V} \leq V_{DD} \leq 5.5\text{V}$
D040A			0.25 $V_{DD} + 0.8\text{V}$	—	$V_{DD}$	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	
D042A		OSC1 (in XT and LP modes) OSC1 (in HS mode)	1.6V 0.7 $V_{DD}$	—	$V_{DD}$ $V_{DD}$	V	(Note 1)
D043		OSC1 (in RC mode)	0.9 $V_{DD}$	—	$V_{DD}$	V	
D044		Ports RC3 and RC4: with Schmitt Trigger buffer	0.7 $V_{DD}$	—	$V_{DD}$	V	For entire $V_{DD}$ range
D044A		with SMBus	1.4	—	5.5	V	For $V_{DD} = 4.5$ to 5.5V
D070	$I_{PURB}$	<b>PORTB Weak Pull-up Current</b>	50	250	400	$\mu\text{A}$	$V_{DD} = 5\text{V}$ , $V_{PIN} = V_{SS}$ , $-40^{\circ}\text{C}$ TO $+85^{\circ}\text{C}$

\* 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 PIC16F7X7 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.

## 18.4 DC Characteristics: PIC16F737/747/767/777 (Industrial, Extended) PIC16LF737/747/767/777 (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 Section 18.1 "DC Characteristics".					
Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
D060	IIL	<b>Input Leakage Current</b> <sup>(2, 3)</sup> I/O ports	—	—	$\pm 1$	$\mu\text{A}$	$V_{SS} \leq V_{PIN} \leq V_{DD}$ , pin at high-impedance
D061		$\overline{\text{MCLR}}$ , RA4/T0CKI	—	—	$\pm 5$	$\mu\text{A}$	$V_{SS} \leq V_{PIN} \leq V_{DD}$
D063		OSC1	—	—	$\pm 5$	$\mu\text{A}$	$V_{SS} \leq V_{PIN} \leq V_{DD}$ , XT, HS and LP oscillator configuration
D080	VOL	<b>Output Low Voltage</b> 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 $+125^{\circ}\text{C}$
			—	—	0.6	V	$I_{OL} = 1.2 \text{ mA}$ , $V_{DD} = 4.5\text{V}$ , $-40^{\circ}\text{C}$ to $+125^{\circ}\text{C}$
D090	VOH	<b>Output High Voltage</b> I/O ports ( <b>Note 3</b> )	$V_{DD} - 0.7$	—	—	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 $+125^{\circ}\text{C}$
			$V_{DD} - 0.7$	—	—	V	$I_{OH} = -1.0 \text{ mA}$ , $V_{DD} = 4.5\text{V}$ , $-40^{\circ}\text{C}$ to $+125^{\circ}\text{C}$
D150*	VOD	<b>Open-Drain High Voltage</b>	—	—	12	V	RA4 pin
		<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>Program Flash Memory</b>					
D130	EP	Endurance	100	1000	—	E/W	25°C at 5V
D131	VPR	VDD for Read	2.0	—	5.5	V	

\* 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 PIC16F7X7 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.

# PIC16F7X7

**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	PIC16F7X7 PIC16LF7X7
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<sub>SS</sub> to V<sub>DD</sub>.

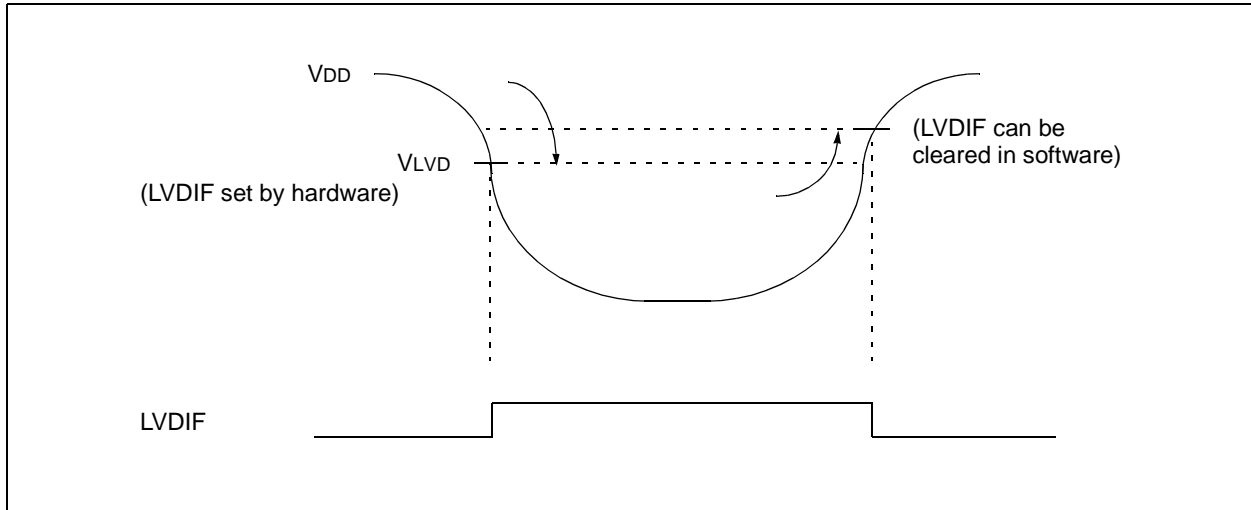
**TABLE 18-2: VOLTAGE REFERENCE SPECIFICATIONS**

Operating Conditions: $3.0V < V_{DD} < 5.5V$ , $-40^{\circ}C < T_A < +85^{\circ}C$ (unless otherwise stated).							
Param 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	— —	— —	1/4 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> transition from '0000' to '1111'.

**FIGURE 18-3: LOW-VOLTAGE DETECT CHARACTERISTICS**



**TABLE 18-3: LOW-VOLTAGE DETECT 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								
Param No.	Symbol	Characteristic		Min	Typ†	Max	Units	Conditions
D420	VLVD	LVD Voltage on VDD Transition High-to-Low	LVDL<3:0> = 0000	N/A	N/A	N/A	V	Reserved
			LVDL<3:0> = 0001	1.96	2.06	2.16	V	$T \geq 25^{\circ}\text{C}$
			LVDL<3:0> = 0010	2.16	2.27	2.38	V	$T \geq 25^{\circ}\text{C}$
			LVDL<3:0> = 0011	2.35	2.47	2.59	V	$T \geq 25^{\circ}\text{C}$
			LVDL<3:0> = 0100	2.43	2.56	2.69	V	
			LVDL<3:0> = 0101	2.64	2.78	2.92	V	
			LVDL<3:0> = 0110	2.75	2.89	3.03	V	
			LVDL<3:0> = 0111	2.95	3.1	3.26	V	
			LVDL<3:0> = 1000	3.24	3.41	3.58	V	
			LVDL<3:0> = 1001	3.43	3.61	3.79	V	
			LVDL<3:0> = 1010	3.53	3.72	3.91	V	
			LVDL<3:0> = 1011	3.72	3.92	4.12	V	
			LVDL<3:0> = 1100	3.92	4.13	4.34	V	
			LVDL<3:0> = 1101	4.11	4.33	4.55	V	
LVDL<3:0> = 1110	4.41	4.64	4.87	V				

**Legend:** Shading of rows is to assist in readability of the table.

† Production tested at  $T_{AMB} = 25^{\circ}\text{C}$ . Specifications over temperature limits ensured by characterization.

# PIC16F7X7

## 18.5 Timing Parameter Symbology

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

1. TppS2ppS
2. TppS
3. TCC:ST (I<sup>2</sup>C specifications only)
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	$\overline{WR}$

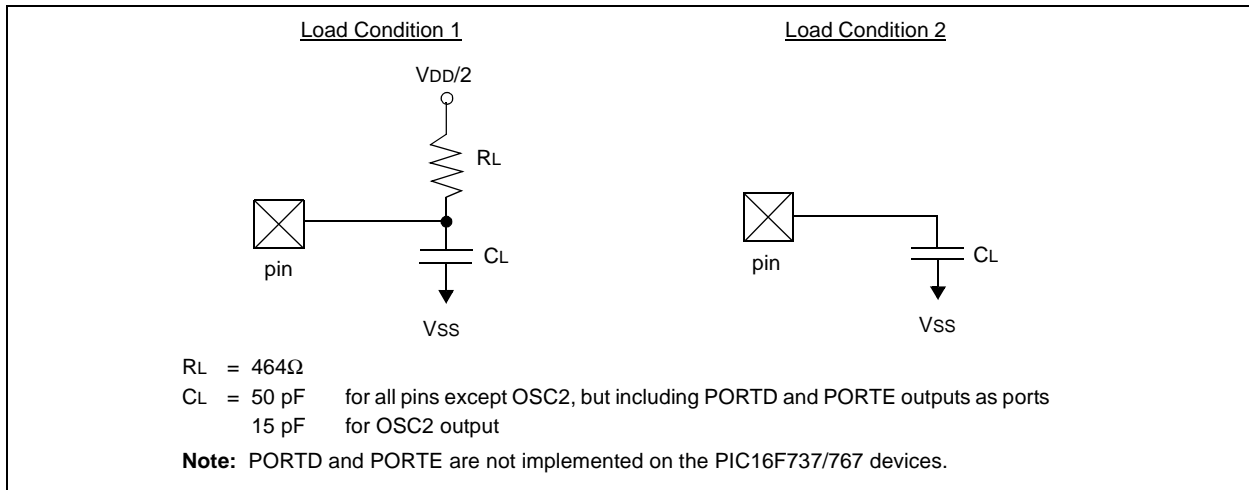
Uppercase letters and their meanings:

<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)

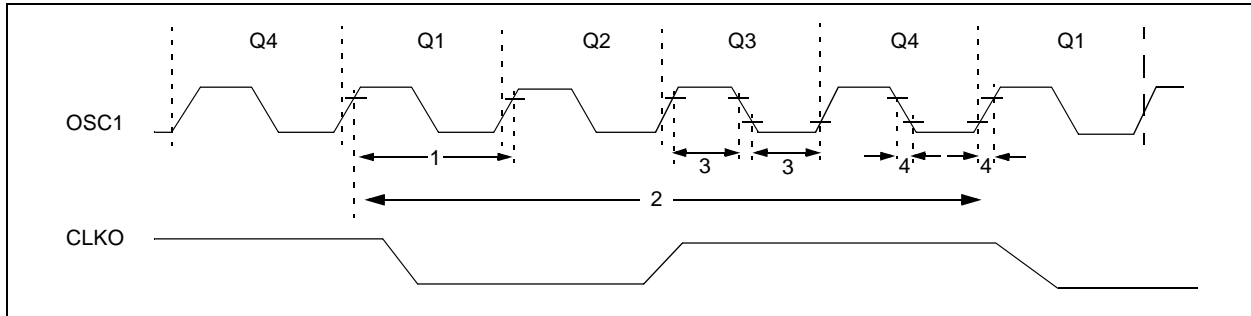
<b>CC</b>			
HD	Hold	SU	Setup
<b>ST</b>			
DAT	DATA input hold	STO	Stop condition
STA	Start condition		

**FIGURE 18-4: LOAD CONDITIONS**





**FIGURE 18-5: EXTERNAL CLOCK TIMING**



**TABLE 18-4: EXTERNAL CLOCK TIMING REQUIREMENTS**

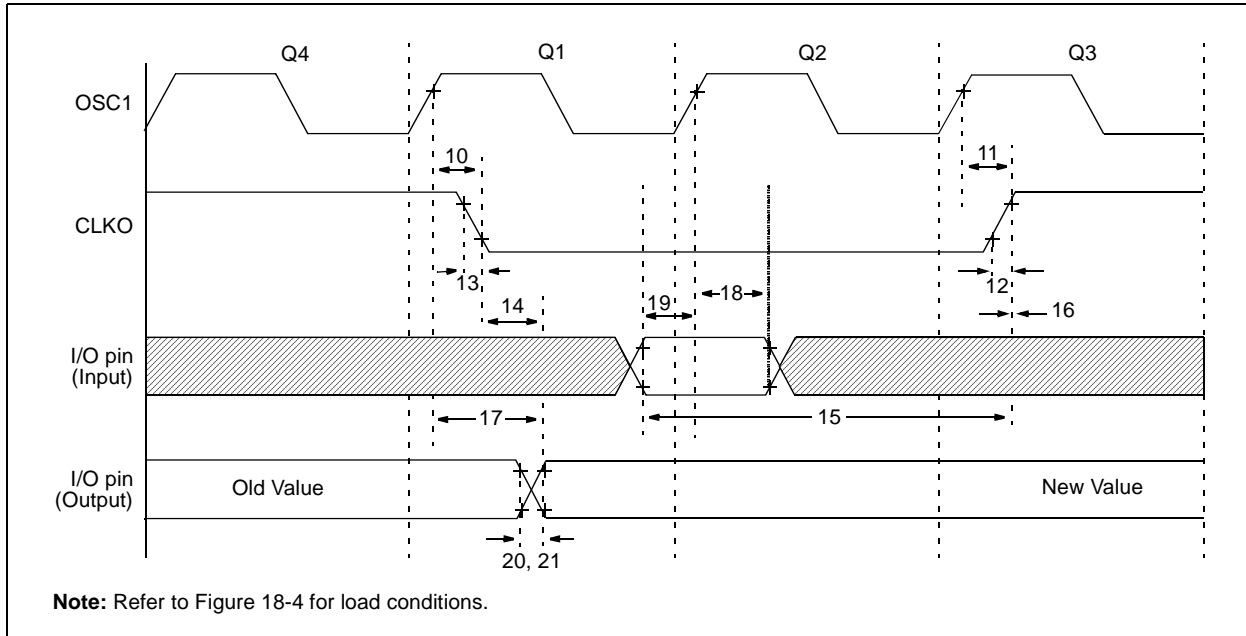
Param No.	Symbol	Characteristic	Min	Typ†	Max	Units	Conditions
	FOSC	External CLKI Frequency (Note 1)	DC	—	1	MHz	XT Oscillator mode
			DC	—	20	MHz	HS Oscillator mode
			DC	—	32	kHz	LP Oscillator mode
		Oscillator Frequency (Note 1)	DC	—	4	MHz	RC Oscillator mode
			0.1	—	4	MHz	XT Oscillator mode
4			—	20	MHz	HS Oscillator mode	
		5	—	200	kHz	LP Oscillator mode	
1	TOSC	External CLKI Period (Note 1)	1000	—	—	ns	XT Oscillator mode
			50	—	—	ns	HS Oscillator mode
			5	—	—	ms	LP Oscillator mode
		Oscillator Period (Note 1)	250	—	—	ns	RC Oscillator mode
			250	—	10,000	ns	XT Oscillator mode
50			—	250	ns	HS Oscillator mode	
		5	—	—	ms	LP Oscillator mode	
2	Tcy	Instruction Cycle Time (Note 1)	200	Tcy	DC	ns	Tcy = 4/FOSC
3	TosL, TosH	External Clock in (OSC1) High or Low Time	500	—	—	ns	XT oscillator
			2.5	—	—	ms	LP oscillator
			15	—	—	ns	HS oscillator
4	TosR, TosF	External Clock in (OSC1) Rise or Fall Time	—	—	25	ns	XT oscillator
			—	—	50	ns	LP oscillator
			—	—	15	ns	HS oscillator

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

**Note 1:** Instruction cycle period (Tcy) equals four times the input oscillator time base period. All specified values are based on characterization data for that particular oscillator type, under standard operating conditions, with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at "min." values with an external clock applied to the OSC1/CLKI pin. When an external clock input is used, the "max." cycle time limit is "DC" (no clock) for all devices.

# PIC16F7X7

**FIGURE 18-6: CLKO AND I/O TIMING**



**TABLE 18-5: 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 T <sub>CY</sub> + 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)	PIC16F7X7	100	—	—	ns	
			PIC16LF7X7	200	—	—	ns	
19*	TioV2osH	Port Input Valid to OSC1 ↑ (I/O in setup time)	0	—	—	ns		
20*	TioR	Port Output Rise Time	PIC16F7X7	—	10	40	ns	
			PIC16LF7X7	—	—	145	ns	
21*	TioF	Port Output Fall Time	PIC16F7X7	—	10	40	ns	
			PIC16LF7X7	—	—	145	ns	
22††*	TINP	INT pin High or Low Time	T <sub>CY</sub>	—	—	ns		
23††*	TRBP	RB7:RB4 Change INT High or Low Time	T <sub>CY</sub>	—	—	ns		

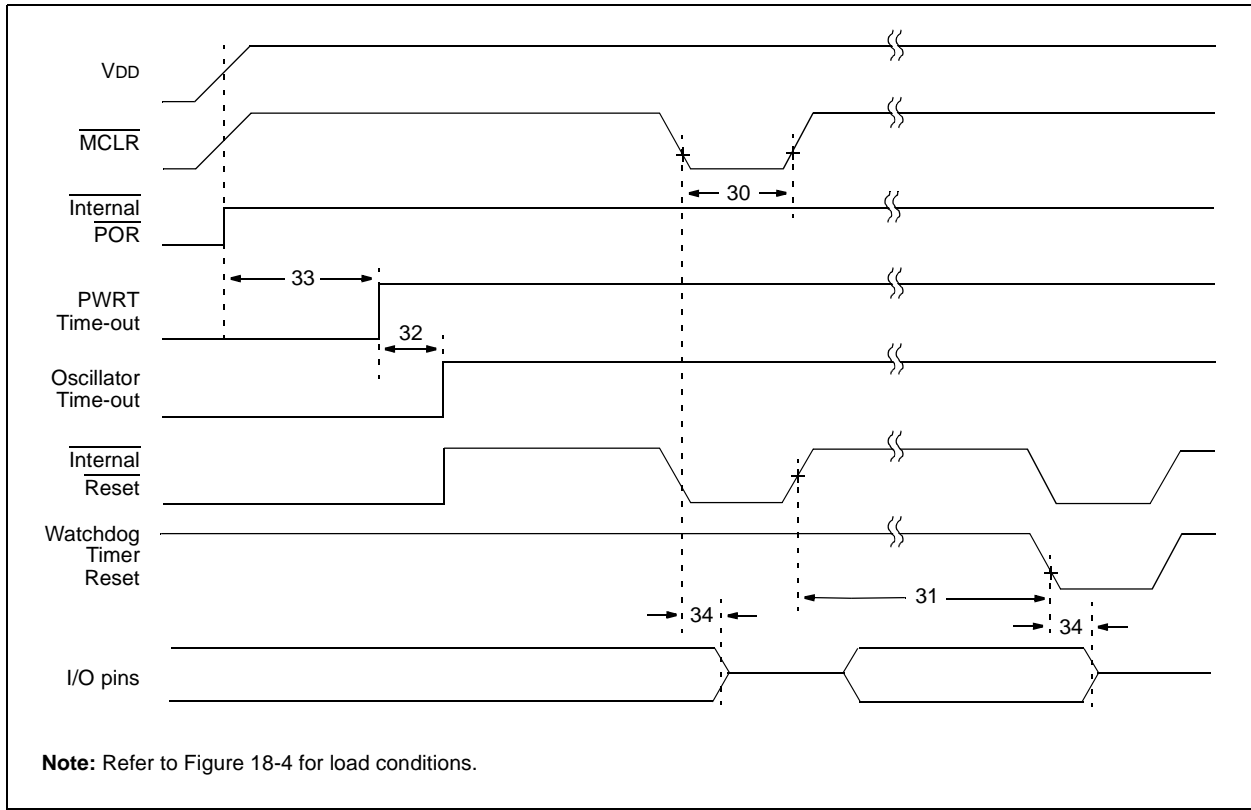
\* These parameters are characterized but not tested.

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

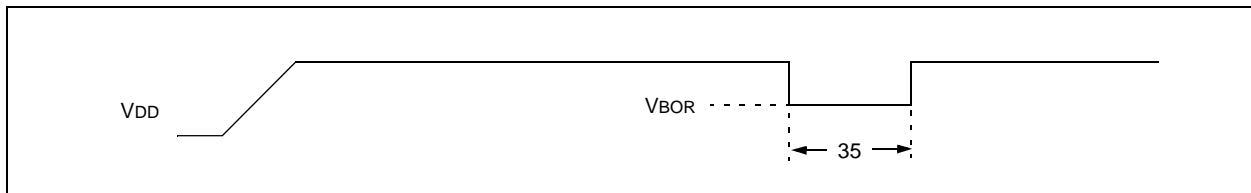
†† These parameters are asynchronous events not related to any internal clock edges.

**Note 1:** Measurements are taken in RC mode, where CLKO output is 4 x T<sub>osc</sub>.

**FIGURE 18-7: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING**



**FIGURE 18-8: BROWN-OUT RESET TIMING**



**TABLE 18-6: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER, POWER-UP TIMER AND BROWN-OUT RESET REQUIREMENTS**

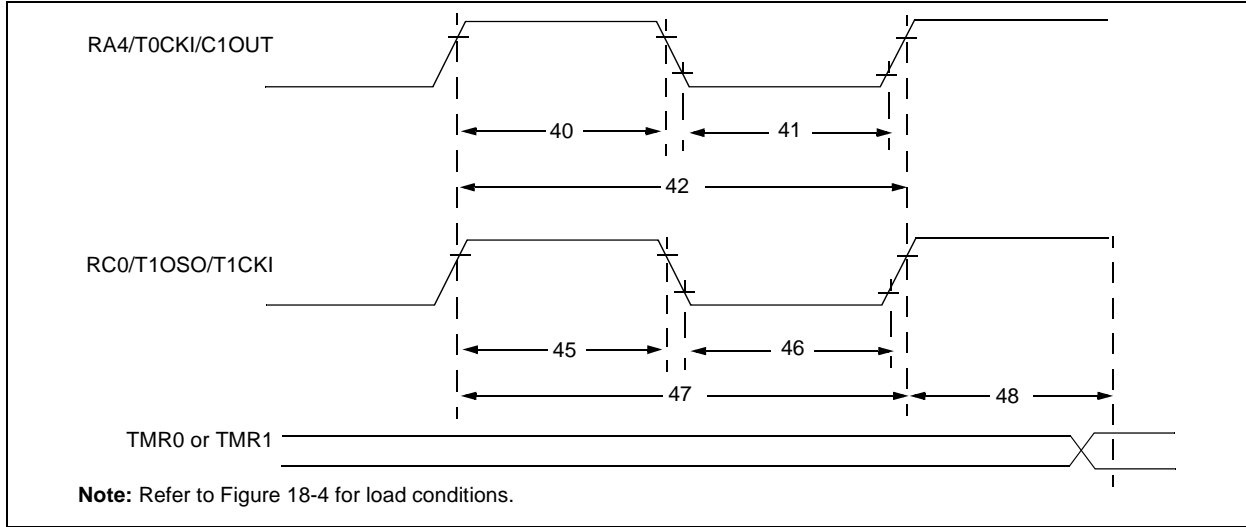
Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
30	TMCL	MCLR Pulse Width (low)	2	—	—	μs	VDD = 5V, -40°C to +85°C
31*	TWDT	Watchdog Timer Time-out Period (no prescaler)	13.6	16	18.4	ms	VDD = 5V, -40°C to +85°C
32	TOST	Oscillation Start-up Timer Period	—	1024 TOSC	—	—	TOSC = OSC1 period
33*	TPWRT	Power-up Timer Period	61.2	72	82.8	ms	VDD = 5V, -40°C to +85°C
34	TIOZ	I/O High-Impedance from MCLR Low or Watchdog Timer Reset	—	—	2.1	μs	
35	TBOR	Brown-out Reset Pulse Width	100	—	—	μs	VDD ≤ VBOR (D005)

\* These parameters are characterized but not tested.

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

# PIC16F7X7

**FIGURE 18-9: TIMER0 AND TIMER1 EXTERNAL CLOCK TIMINGS**



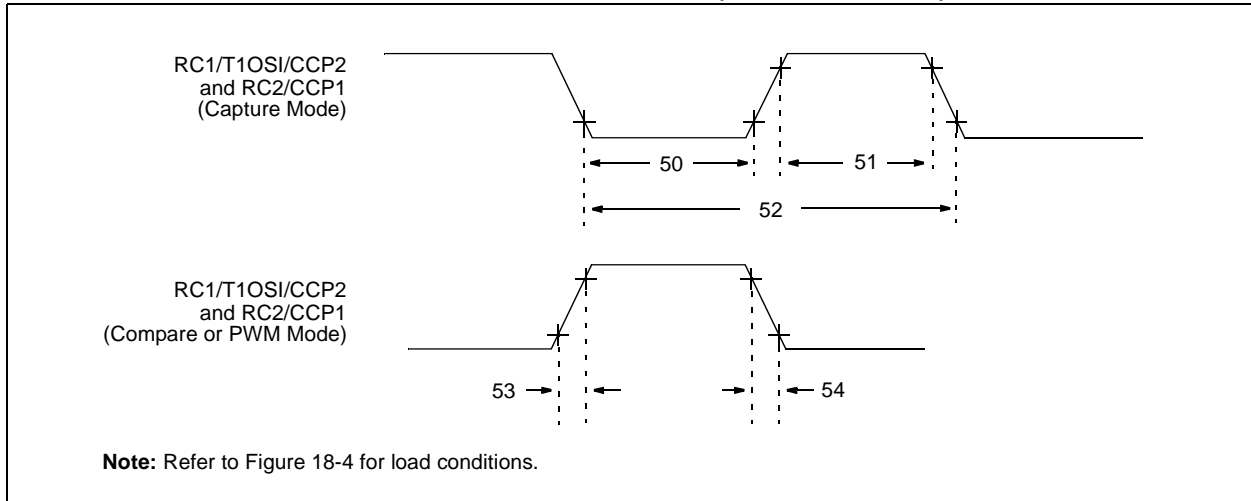
**TABLE 18-7: TIMER0 AND TIMER1 EXTERNAL CLOCK REQUIREMENTS**

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	N = prescale value (2, 4, ..., 256)	
			With prescaler	Greater of: $20$ or $\frac{T_{CY} + 40}{N}$	—	—	ns		
45*	Tt1H	T1CKI High Time	Synchronous, Prescaler = 1	$0.5 T_{CY} + 20$	—	—	ns	Must also meet parameter 47	
			Synchronous, Prescaler = 2, 4, 8	PIC16F7X7	15	—	—		ns
				PIC16LF7X7	25	—	—		ns
			Asynchronous	PIC16F7X7	30	—	—		ns
PIC16LF7X7	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	PIC16F7X7	15	—	—		ns
				PIC16LF7X7	25	—	—		ns
			Asynchronous	PIC16F7X7	30	—	—		ns
PIC16LF7X7	50	—		—	ns				
47*	Tt1P	T1CKI Input Period	Synchronous	PIC16F7X7	Greater of: $30$ or $\frac{T_{CY} + 40}{N}$	—	—	ns	N = prescale value (1, 2, 4, 8)
				PIC16LF7X7	Greater of: $50$ or $\frac{T_{CY} + 40}{N}$	—	—	ns	N = prescale value (1, 2, 4, 8)
			Asynchronous	PIC16F7X7	60	—	—	ns	
				PIC16LF7X7	100	—	—	ns	
	Ft1	Timer1 Oscillator Input Frequency Range (oscillator enabled by setting bit T1OSCEN)		DC	—	200	kHz		
48	TCKEZTMR1	Delay from External Clock Edge to Timer Increment		$2 T_{osc}$	—	$7 T_{osc}$	—		

\* 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 18-10: CAPTURE/COMPARE/PWM TIMINGS (CCP1 AND CCP2)**



**TABLE 18-8: CAPTURE/COMPARE/PWM REQUIREMENTS (ALL CCP MODULES)**

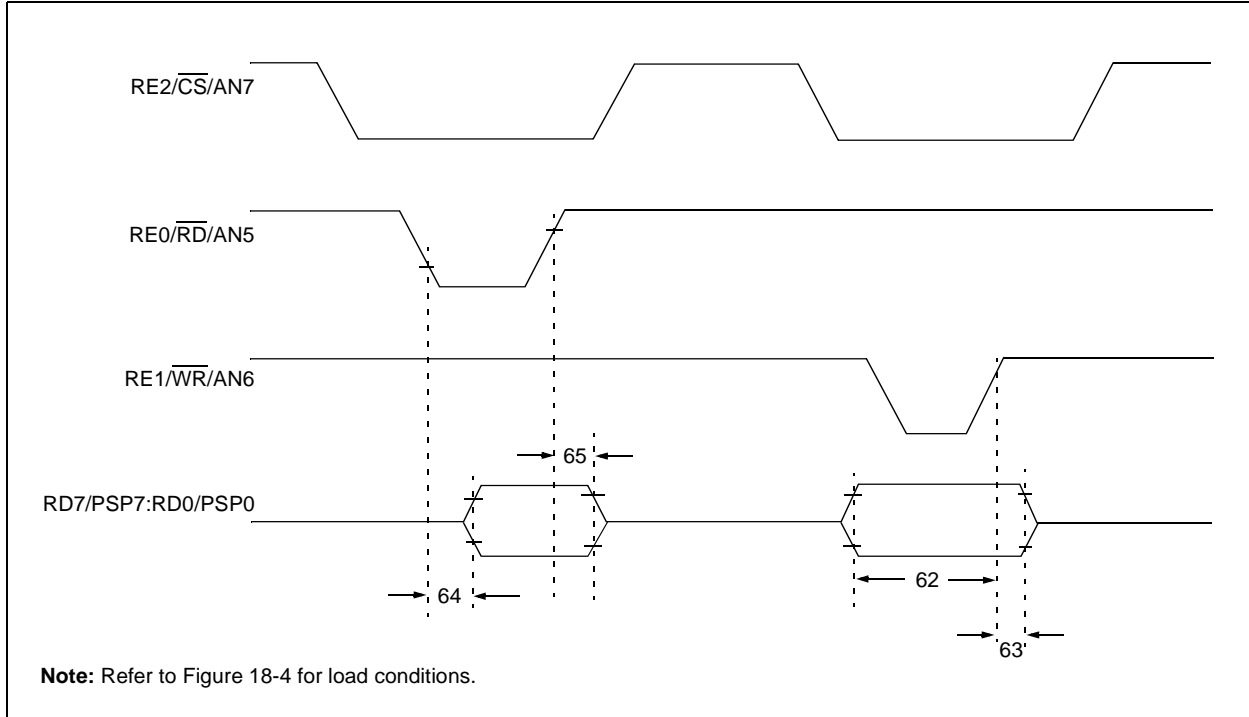
Param No.	Symbol	Characteristic		Min	Typ†	Max	Units	Conditions	
50*	TcCL	CCP1, CCP2 and CCP3 Input Low Time	No prescaler	$0.5 T_{CY} + 20$	—	—	ns		
			With prescaler	PIC16F7X7	10	—	—		ns
				PIC16LF7X7	20	—	—		ns
51*	TcCH	CCP1, CCP2 and CCP3 Input High Time	No prescaler	$0.5 T_{CY} + 20$	—	—	ns		
			With prescaler	PIC16F7X7	10	—	—		ns
				PIC16LF7X7	20	—	—		ns
52*	TcCP	CCP1, CCP2 and CCP3 Input Period		$\frac{3 T_{CY} + 40}{N}$	—	—	ns	N = prescale value (1, 4 or 16)	
53*	TcCR	CCP1, CCP2 and CCP3 Output Rise Time	PIC16F7X7	—	10	25	ns		
			PIC16LF7X7	—	25	50	ns		
54*	TcCF	CCP1, CCP2 and CCP3 Output Fall Time	PIC16F7X7	—	10	25	ns		
			PIC16LF7X7	—	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.

# PIC16F7X7

**FIGURE 18-11: PARALLEL SLAVE PORT TIMING (PIC16F747/777 DEVICES ONLY)**



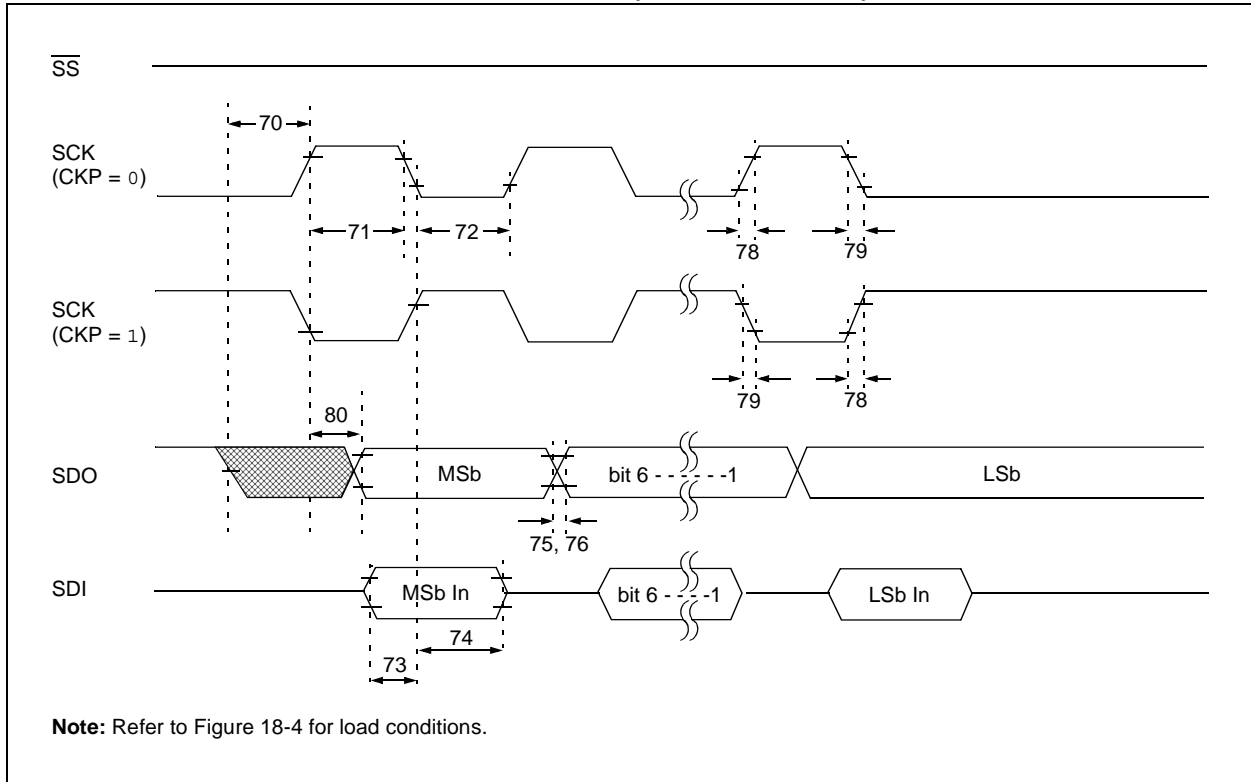
**TABLE 18-9: PARALLEL SLAVE PORT REQUIREMENTS (PIC16F747/777 DEVICES ONLY)**

Param No.	Symbol	Characteristic	Min	Typ†	Max	Units	Conditions
62	TDTV2WRH	Data In Valid before $\overline{WR}$ $\uparrow$ or $\overline{CS}$ $\uparrow$ (setup time)	20 25	—	—	ns	Extended range only
63*	TWRH2DTI	$\overline{WR}$ $\uparrow$ or $\overline{CS}$ $\uparrow$ to Data In Invalid (hold time)	20 35	—	—	ns	
64	TRDL2DTV	$\overline{RD}$ $\downarrow$ and $\overline{CS}$ $\downarrow$ to Data Out Valid	— —	—	80 90	ns	Extended range only
65	TRDH2DTI	$\overline{RD}$ $\uparrow$ or $\overline{CS}$ $\downarrow$ to Data Out Invalid	10	—	30	ns	

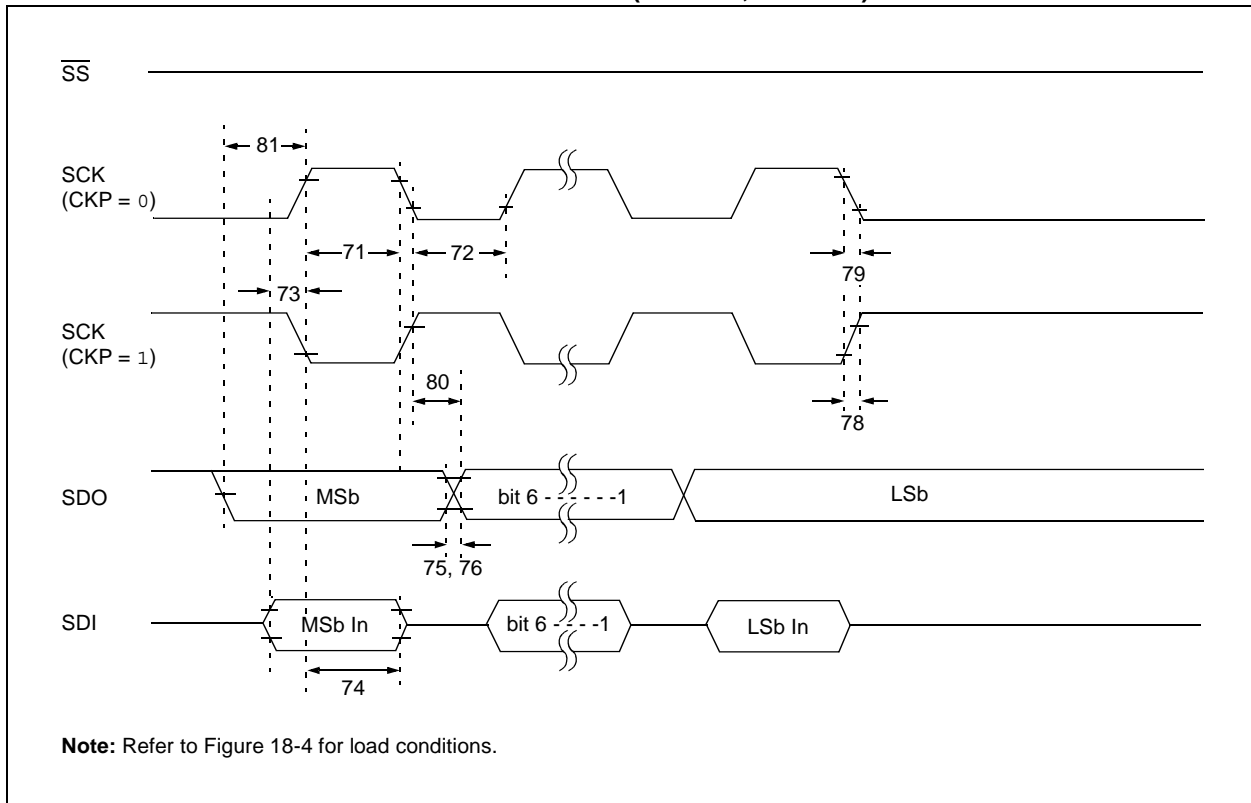
\* These parameters are characterized but not tested.

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

**FIGURE 18-12: SPI™ MASTER MODE TIMING (CKE = 0, SMP = 0)**

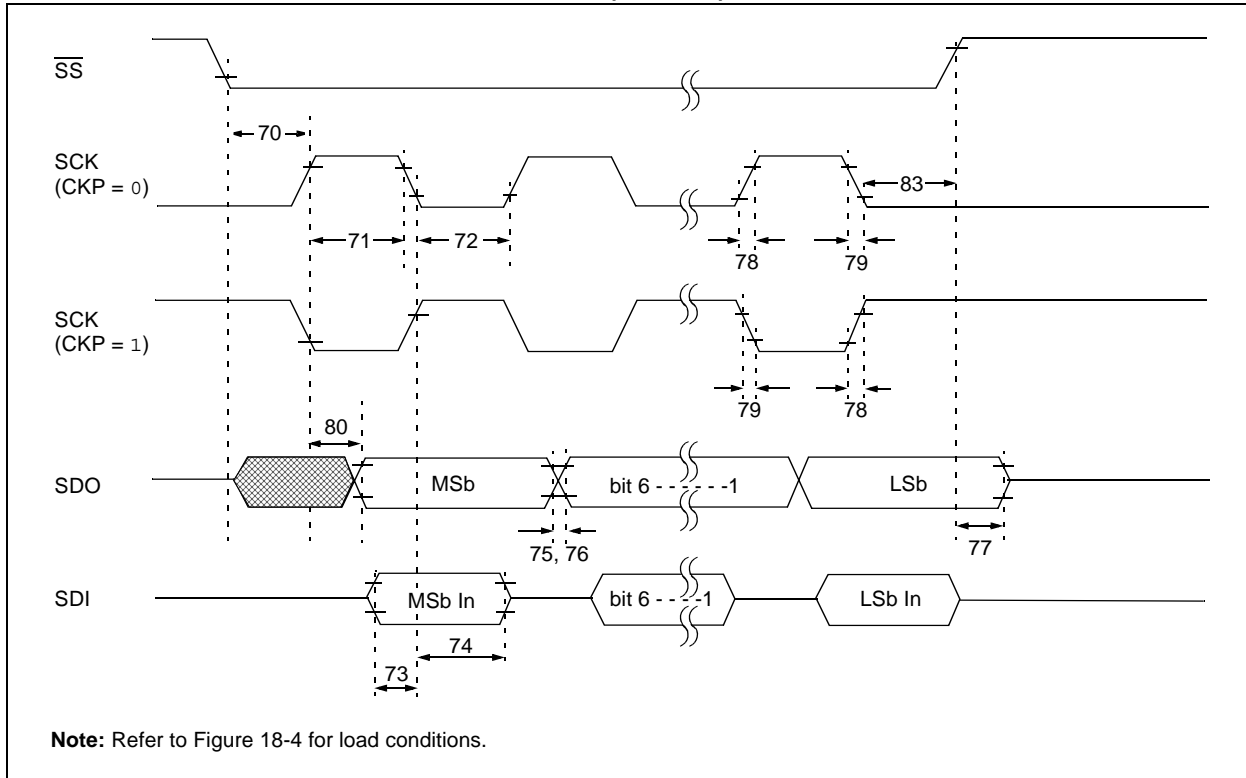


**FIGURE 18-13: SPI™ MASTER MODE TIMING (CKE = 1, SMP = 1)**

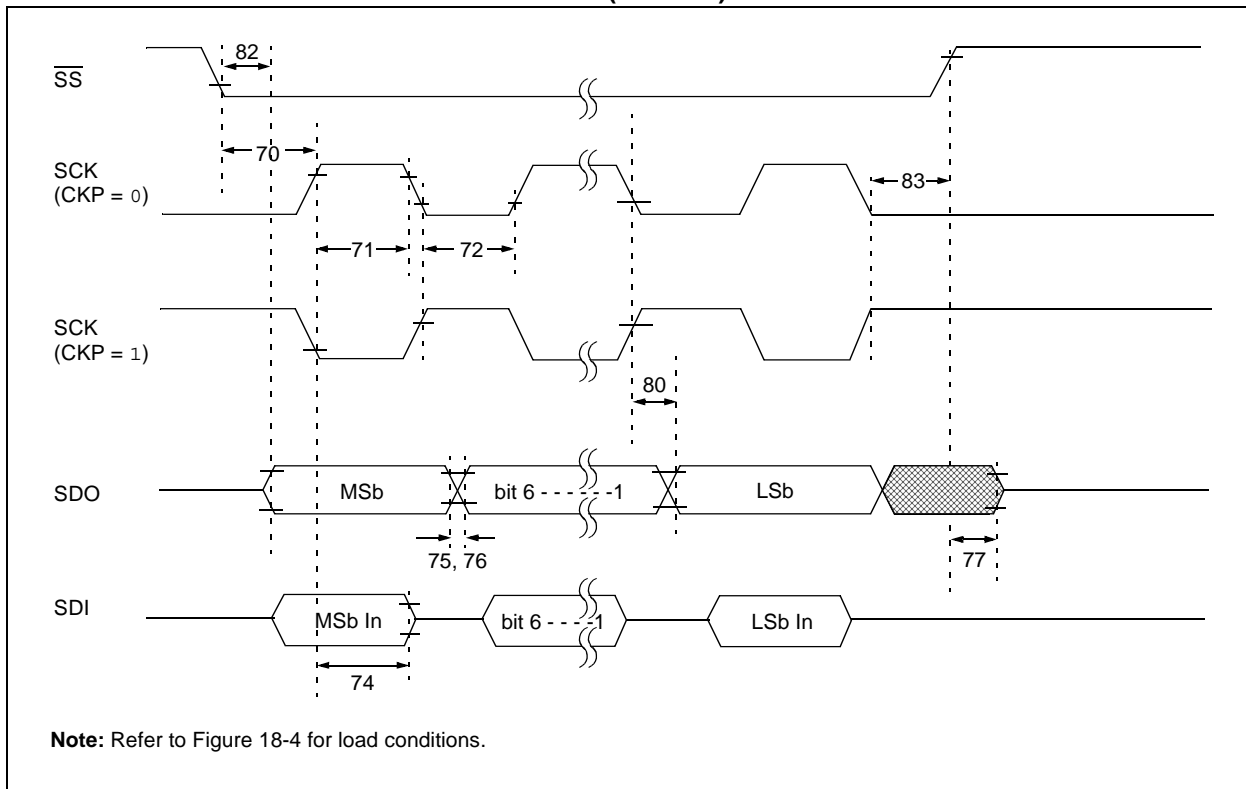


# PIC16F7X7

**FIGURE 18-14: SPI™ SLAVE MODE TIMING (CKE = 0)**



**FIGURE 18-15: SPI™ SLAVE MODE TIMING (CKE = 1)**





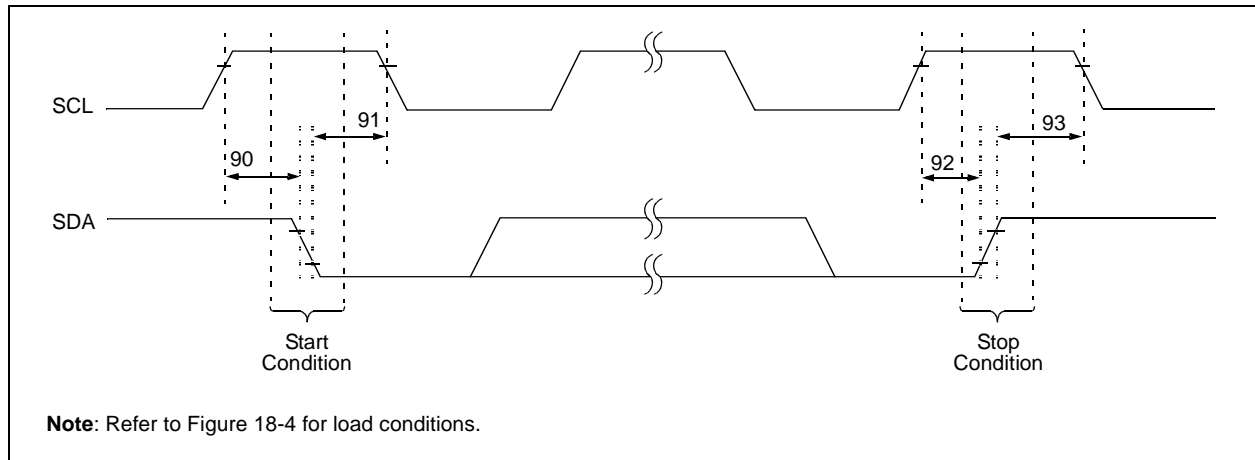
**TABLE 18-10: SPI™ MODE REQUIREMENTS**

Param No.	Symbol	Characteristic	Min	Typ†	Max	Units	Conditions
70*	TssL2sch, TssL2scl	$\overline{SS} \downarrow$ to SCK $\downarrow$ or SCK $\uparrow$ Input	Tcy	—	—	ns	
71*	Tsch	SCK Input High Time (Slave mode)	Tcy + 20	—	—	ns	
72*	Tscl	SCK Input Low Time (Slave mode)	Tcy + 20	—	—	ns	
73*	TdIV2sch, TdIV2scl	Setup Time of SDI Data Input to SCK Edge	100	—	—	ns	
74*	Tsch2dIL, Tscl2dIL	Hold Time of SDI Data Input to SCK Edge	100	—	—	ns	
75*	TdoR	SDO Data Output Rise Time	PIC16F7X7 —	10 25	25 50	ns ns	
76*	TdoF	SDO Data Output Fall Time	—	10	25	ns	
77*	TssH2doZ	$\overline{SS} \uparrow$ to SDO Output High-Impedance	10	—	50	ns	
78*	TscR	SCK Output Rise Time (Master mode)	PIC16F7X7 —	10 25	25 50	ns ns	
79*	TscF	SCK Output Fall Time (Master mode)	—	10	25	ns	
80*	Tsch2doV, Tscl2doV	SDO Data Output Valid after SCK Edge	PIC16F7X7 —	— —	50 145	ns ns	
81*	TdoV2sch, TdoV2scl	SDO Data Output Setup to SCK Edge	Tcy	—	—	ns	
82*	TssL2doV	SDO Data Output Valid after $\overline{SS} \downarrow$ Edge	—	—	50	ns	
83*	Tsch2ssH, Tscl2ssH	$\overline{SS} \uparrow$ after SCK Edge	1.5 Tcy + 40	—	—	ns	

\* These parameters are characterized but not tested.

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

**FIGURE 18-16: I<sup>2</sup>C™ BUS START/STOP BITS TIMING**



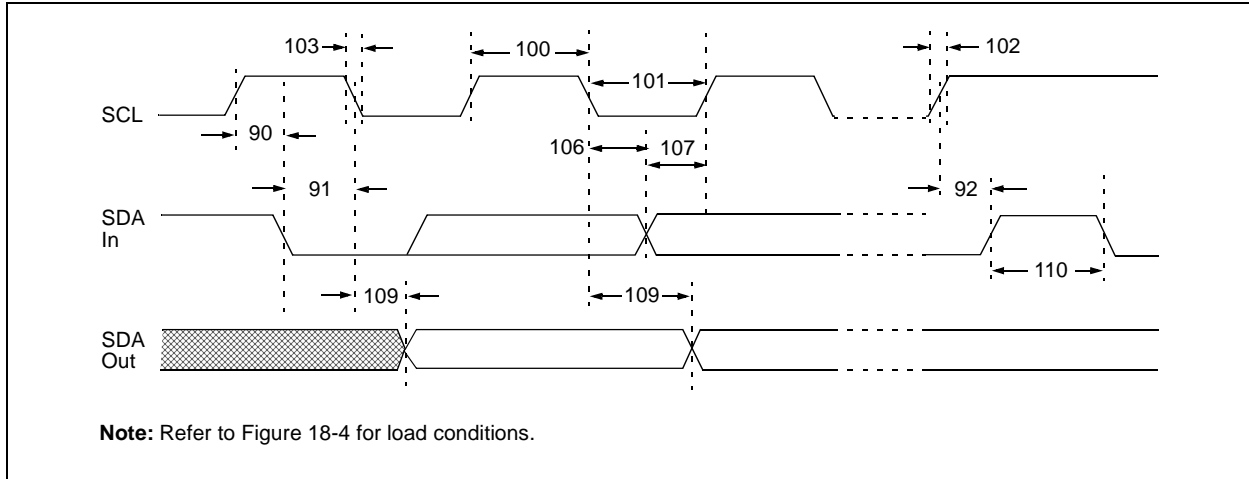
# PIC16F7X7

**TABLE 18-11: 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 18-17: I<sup>2</sup>C™ BUS DATA TIMING**



**TABLE 18-12: I<sup>2</sup>C™ BUS DATA REQUIREMENTS**

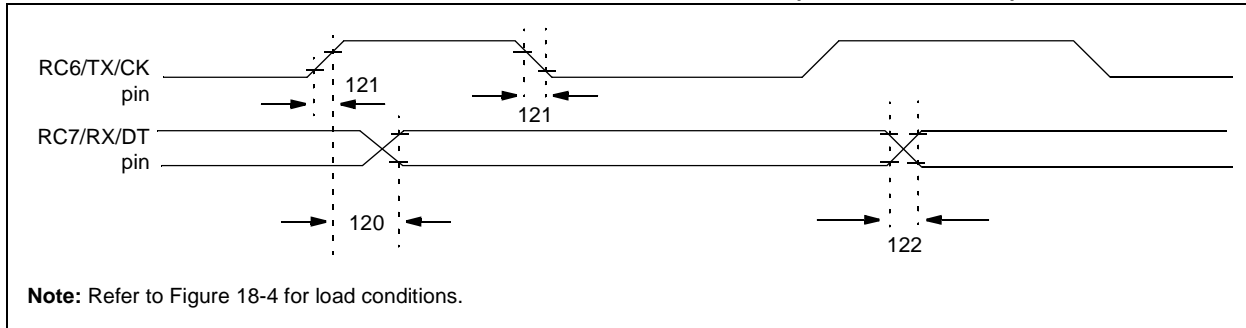
Param. No.	Symbol	Characteristic		Min	Max	Units	Conditions
100*	THIGH	Clock High Time	100 kHz mode	4.0	—	μs	Device must operate at a minimum of 1.5 MHz
			400 kHz mode	0.6	—	μs	Device must operate at a minimum of 10 MHz
			SSP module	1.5 T <sub>CY</sub>	—		
101*	TLOW	Clock Low Time	100 kHz mode	4.7	—	μs	Device must operate at a minimum of 1.5 MHz
			400 kHz mode	1.3	—	μs	Device must operate at a minimum of 10 MHz
			SSP module	1.5 T <sub>CY</sub>	—		
102*	TR	SDA and SCL Rise Time	100 kHz mode	—	1000	ns	
			400 kHz mode	20 + 0.1 C <sub>B</sub>	300	ns	C <sub>B</sub> 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 C <sub>B</sub>	300	ns	C <sub>B</sub> 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	
	C <sub>B</sub>	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.

# PIC16F7X7

**FIGURE 18-18: AUSART SYNCHRONOUS TRANSMISSION (MASTER/SLAVE) TIMING**

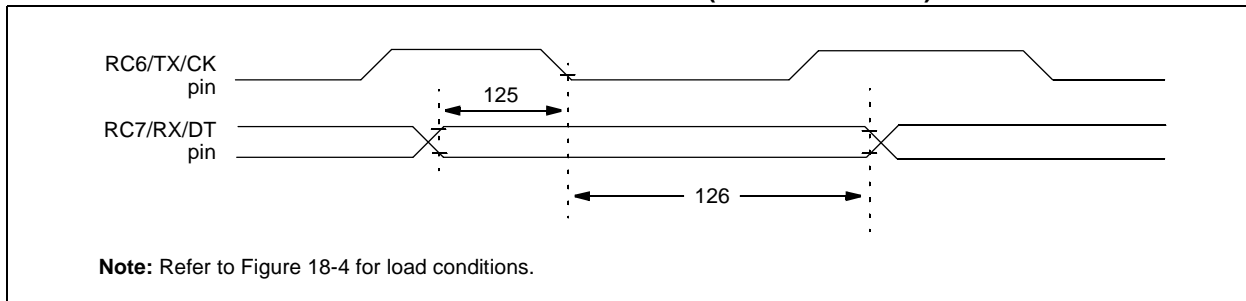


**TABLE 18-13: AUSART SYNCHRONOUS TRANSMISSION REQUIREMENTS**

Param No.	Symbol	Characteristic	Min	Typ†	Max	Units	Conditions
120	T <sub>CKH2DTV</sub>	SYNC XMIT (MASTER & SLAVE) Clock High to Data Out Valid	PIC16F7X7	—	—	80	ns
			PIC16LF7X7	—	—	100	ns
121	T <sub>CKRF</sub>	Clock Out Rise Time and Fall Time (Master mode)	PIC16F7X7	—	—	45	ns
			PIC16LF7X7	—	—	50	ns
122	T <sub>DTRF</sub>	Data Out Rise Time and Fall Time	PIC16F7X7	—	—	45	ns
			PIC16LF7X7	—	—	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 18-19: AUSART SYNCHRONOUS RECEIVE (MASTER/SLAVE) TIMING**



**TABLE 18-14: AUSART SYNCHRONOUS RECEIVE REQUIREMENTS**

Param No.	Symbol	Characteristic	Min	Typ†	Max	Units	Conditions
125	T <sub>DTV2CKL</sub>	SYNC RCV (MASTER & SLAVE) Data Setup before CK ↓ (DT setup time)	15	—	—	ns	
126	T <sub>CKL2DTL</sub>	Data Hold after CK ↓ (DT hold time)	15	—	—	ns	

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

**TABLE 18-15: A/D CONVERTER CHARACTERISTICS: PIC16F7X7 (INDUSTRIAL, EXTENDED)  
PIC16LF7X7 (INDUSTRIAL)**

Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions	
A01	NR	Resolution	—	—	10 bits	bit	$V_{REF} = V_{DD} = 5.12V$ , $V_{SS} \leq V_{AIN} \leq V_{REF}$	
A03	EIL	Integral Linearity Error	—	—	$<\pm 1$	LSb	$V_{REF} = V_{DD} = 5.12V$ , $V_{SS} \leq V_{AIN} \leq V_{REF}$	
A04	EDL	Differential Linearity Error	—	—	$<\pm 1$	LSb	$V_{REF} = V_{DD} = 5.12V$ , $V_{SS} \leq V_{AIN} \leq V_{REF}$	
A06	E <sub>OFF</sub>	Offset Error	—	—	$<\pm 2$	LSb	$V_{REF} = V_{DD} = 5.12V$ , $V_{SS} \leq V_{AIN} \leq V_{REF}$	
A07	E <sub>GN</sub>	Gain Error	—	—	$<\pm 1$	LSb	$V_{REF} = V_{DD} = 5.12V$ , $V_{SS} \leq V_{AIN} \leq V_{REF}$	
A10	—	Monotonicity	—	guaranteed <sup>(3)</sup>	—	—	$V_{SS} \leq V_{AIN} \leq V_{REF}$	
A20	V <sub>REF</sub>	Reference Voltage (V <sub>REF+</sub> – V <sub>REF-</sub> )	2.0	—	$V_{DD} + 0.3$	V		
A21	V <sub>REF+</sub>	Reference Voltage High	$AV_{DD} - 2.5V$	—	$AV_{DD} + 0.3V$	V		
A22	V <sub>REF-</sub>	Reference Voltage Low	$AV_{SS} - 0.3V$	—	$V_{REF+} - 2.0V$	V		
A25	V <sub>AIN</sub>	Analog Input Voltage	$V_{SS} - 0.3V$	—	$V_{REF} + 0.3V$	V		
A30	Z <sub>AIN</sub>	Recommended Impedance of Analog Voltage Source	—	—	2.5	k $\Omega$	<b>(Note 4)</b>	
A40	I <sub>AD</sub>	A/D Conversion Current (V <sub>DD</sub> )	PIC16F7X7	—	220	—	$\mu A$	Average current consumption when A/D is on <b>(Note 1)</b>
			PIC16LF7X7	—	90	—	$\mu A$	
A50	I <sub>REF</sub>	V <sub>REF</sub> Input Current <b>(Note 2)</b>	—	—	5	$\mu A$	During V <sub>AIN</sub> acquisition. Based on differential of V <sub>HOLD</sub> to V <sub>AIN</sub> to charge CHOLD, see <b>Section 12.1 “A/D Acquisition Requirements”</b> . During A/D conversion cycle	
			—	—	150	$\mu A$		

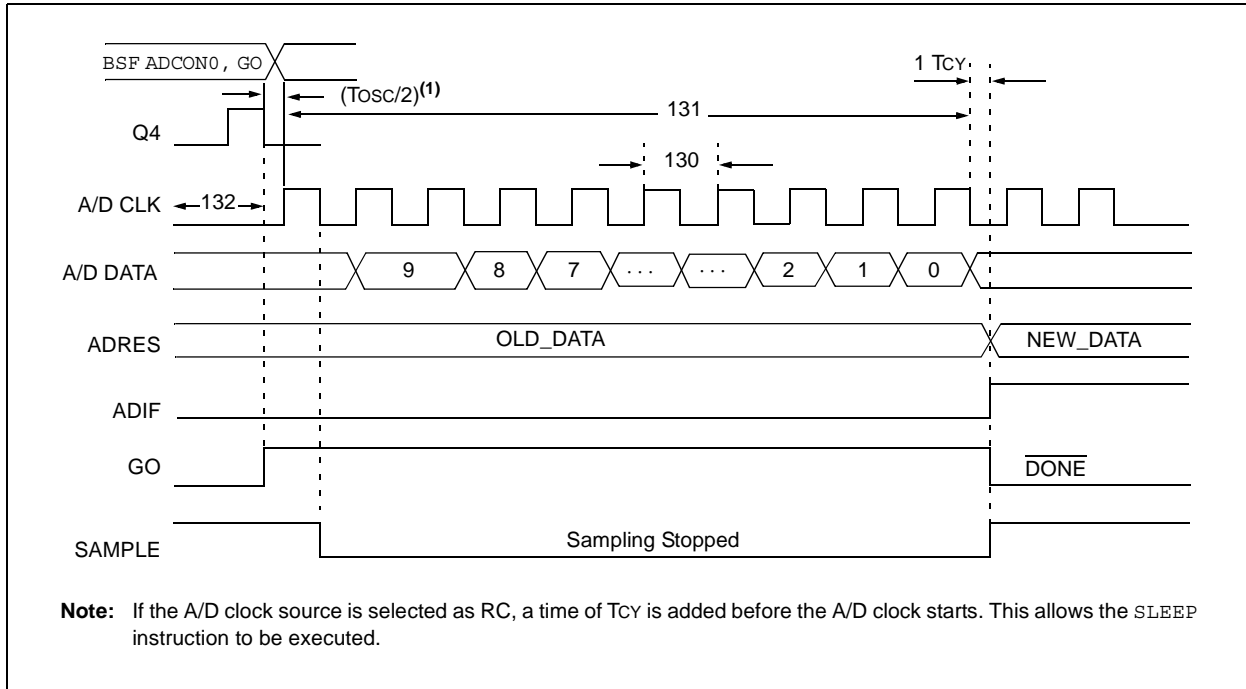
\* These parameters are characterized but not tested.

† 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:** V<sub>REF</sub> current is from RA3 pin or V<sub>DD</sub> 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 k $\Omega$ . This requires higher acquisition time.

# PIC16F7X7

**FIGURE 18-20: A/D CONVERSION TIMING**



**TABLE 18-16: A/D CONVERSION REQUIREMENTS**

Param No.	Symbol	Characteristic	Min	Typ†	Max	Units	Conditions	
130	TAD	A/D Clock Period	PIC16F7X7	1.6	—	—	$\mu\text{s}$	TOSC based, $V_{REF} \geq 3.0\text{V}$
			PIC16LF7X7	3.0	—	—	$\mu\text{s}$	TOSC based, $V_{REF} \geq 2.0\text{V}$
			PIC16F7X7	2.0	4.0	6.0	$\mu\text{s}$	A/D RC mode
			PIC16LF7X7	3.0	6.0	9.0	$\mu\text{s}$	A/D RC mode
131	TCNV	Conversion Time (not including S/H time) (Note 1)		—	12	TAD		
132	TACQ	Acquisition Time	(Note 2)	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 SLEEP instruction to be executed.	

\* These parameters are characterized but not tested.

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

§ This specification ensured by design.

**Note 1:** ADRES register may be read on the following  $T_{cy}$  cycle.

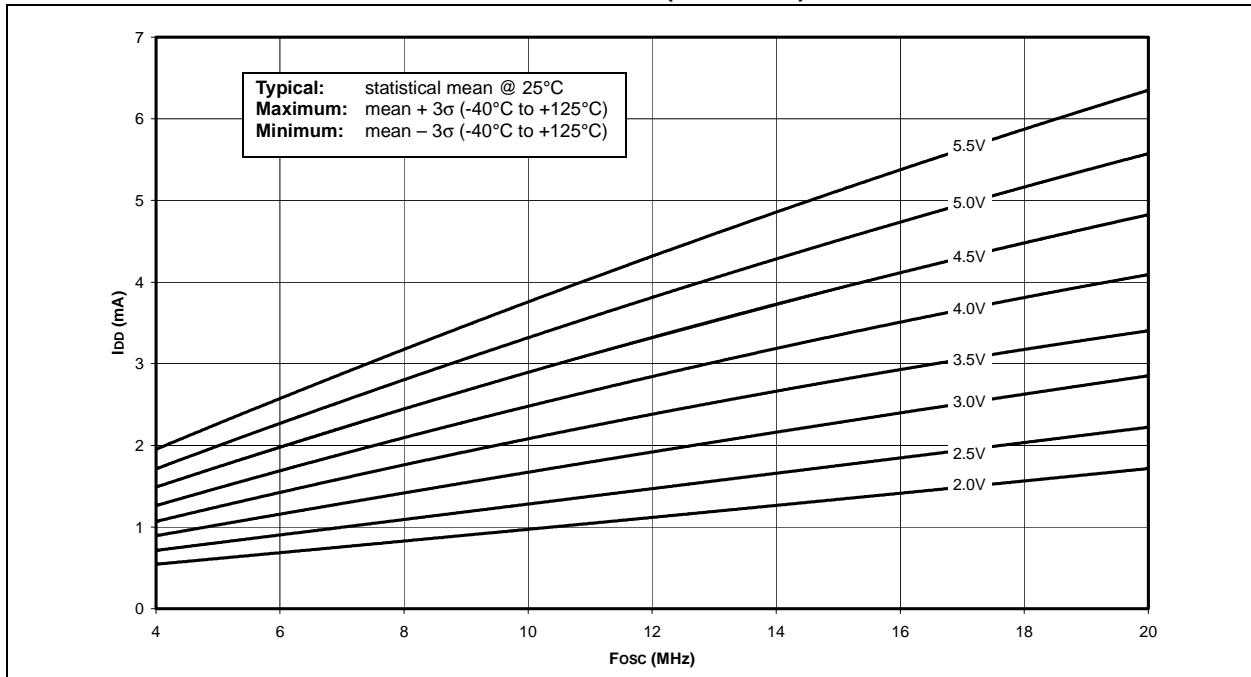
**Note 2:** See Section 12.1 "A/D Acquisition Requirements" for minimum conditions.

## 19.0 DC AND AC CHARACTERISTICS GRAPHS AND TABLES

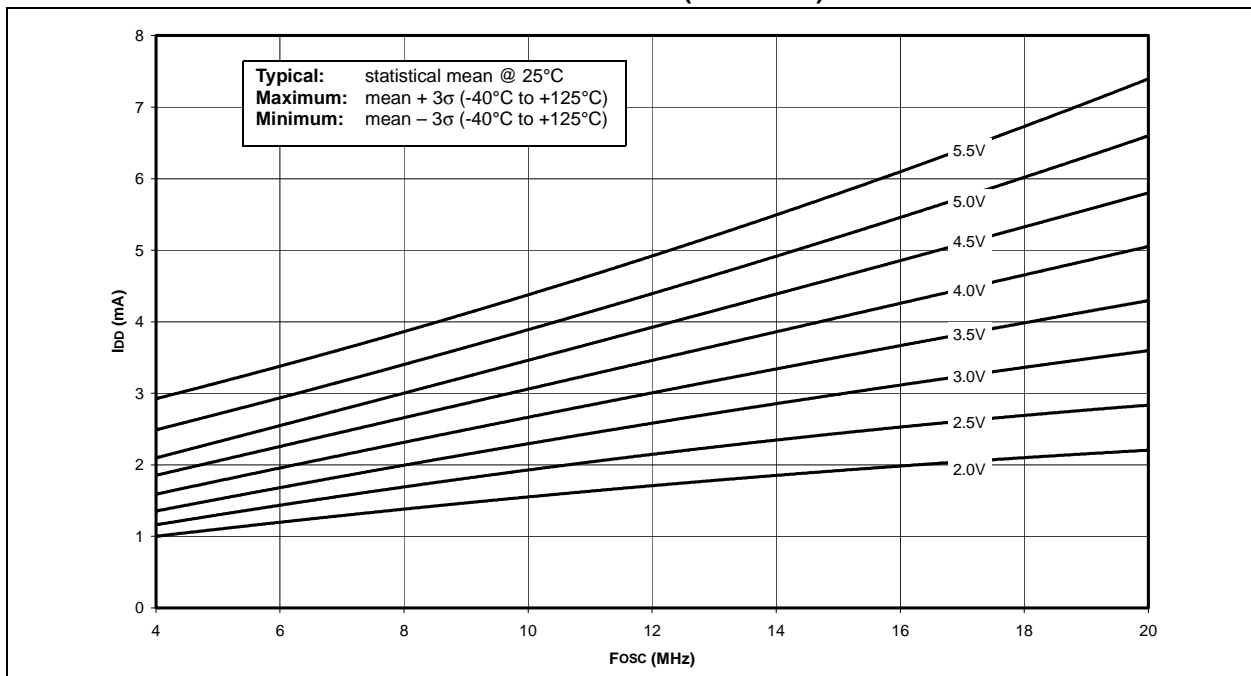
**Note:** The graphs and tables provided following this note are a statistical summary based on a limited number of samples and are provided for informational purposes only. The performance characteristics listed herein are not tested or guaranteed. In some graphs or tables, the data presented may be 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  $I_{DD}$  vs.  $F_{osc}$  OVER  $V_{DD}$  (HS MODE)**



**FIGURE 19-2: MAXIMUM  $I_{DD}$  vs.  $F_{osc}$  OVER  $V_{DD}$  (HS MODE)**



# PIC16F7X7

FIGURE 19-3: TYPICAL  $I_{DD}$  vs.  $F_{osc}$  OVER  $V_{DD}$  (XT MODE)

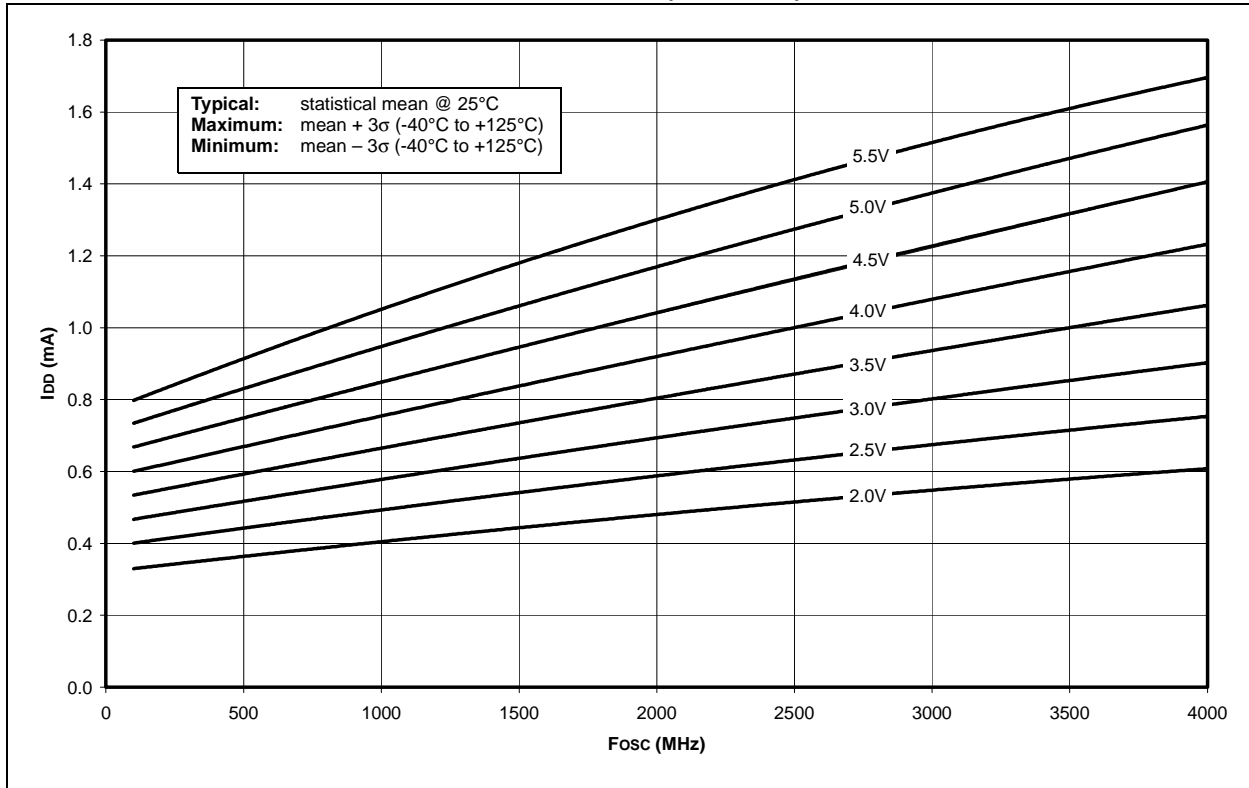
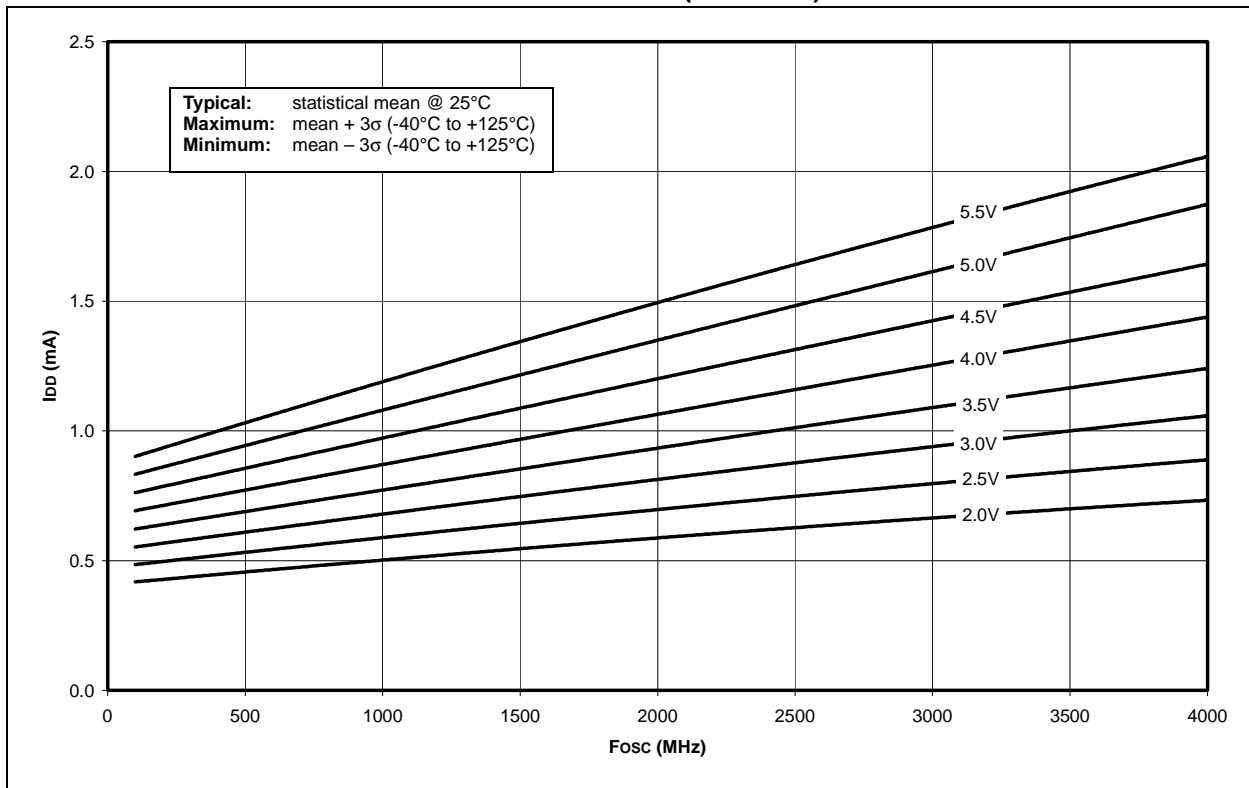
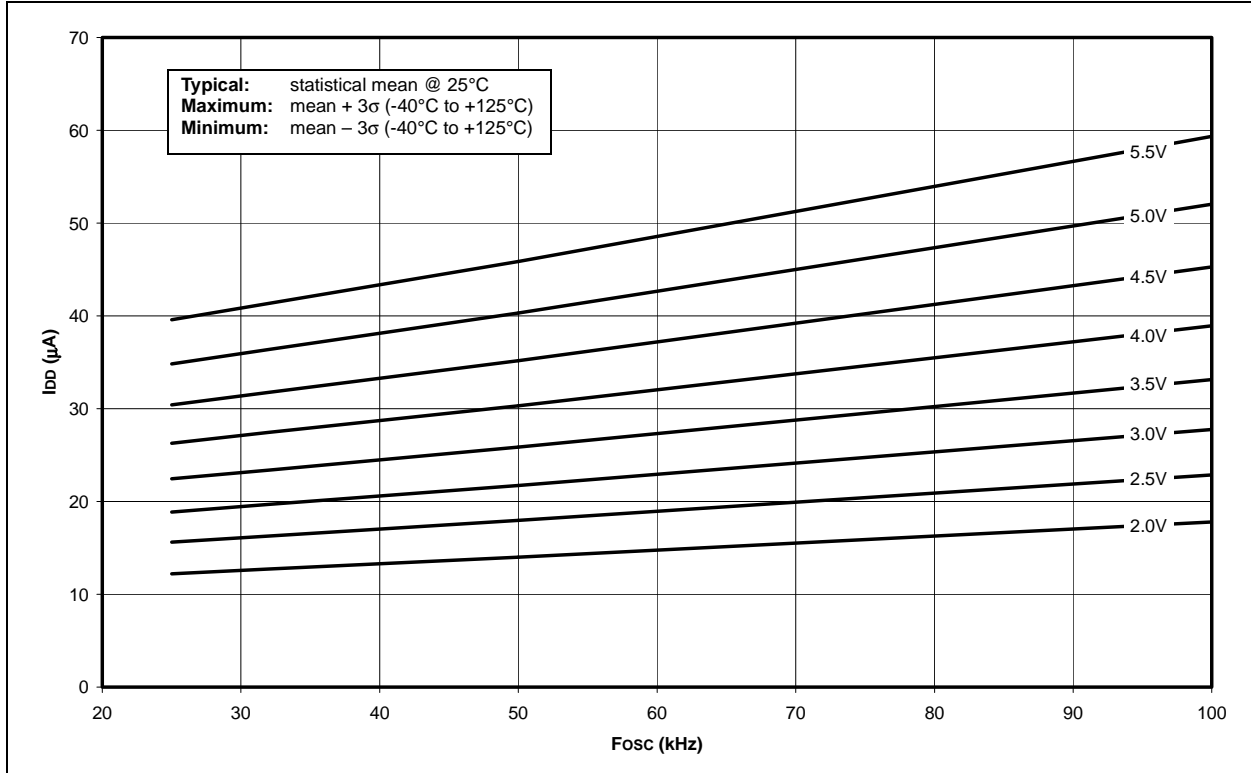


FIGURE 19-4: MAXIMUM  $I_{DD}$  vs.  $F_{osc}$  OVER  $V_{DD}$  (XT MODE)

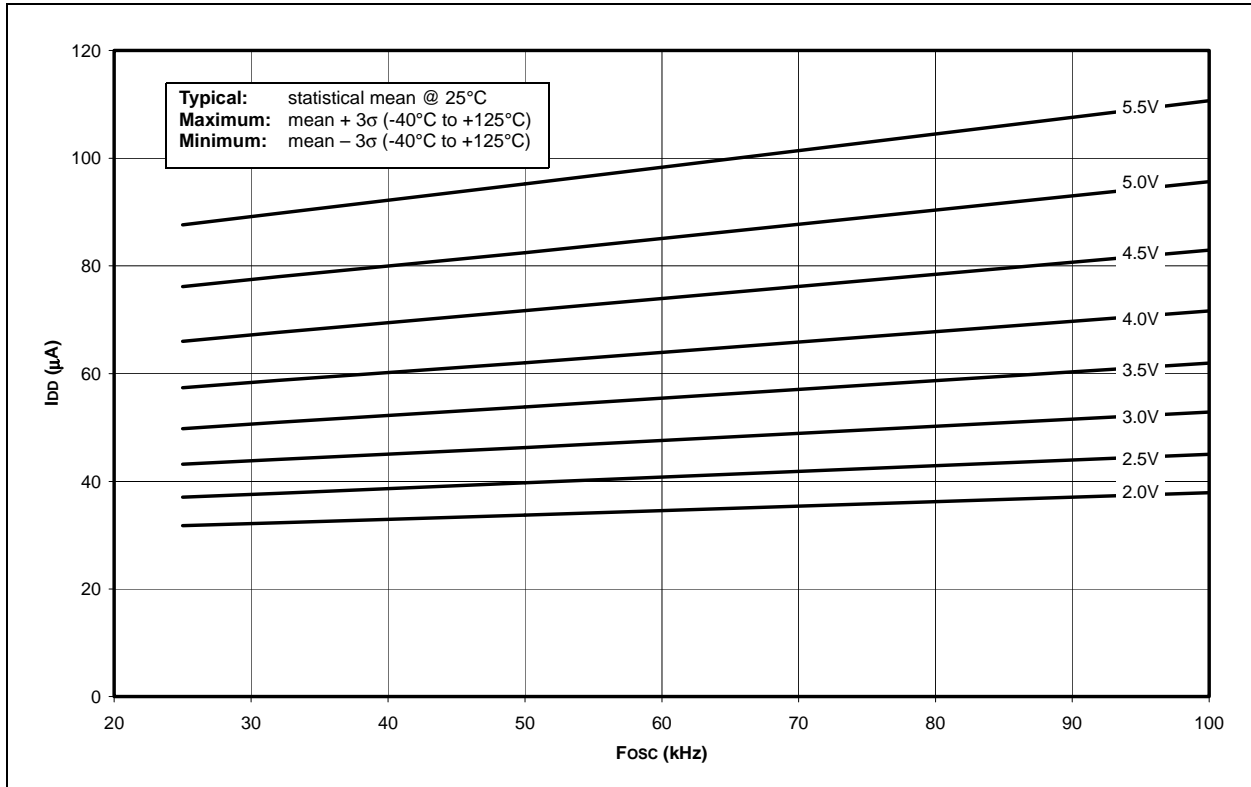




**FIGURE 19-5: TYPICAL  $I_{DD}$  vs.  $F_{osc}$  OVER  $V_{DD}$  (LP MODE)**

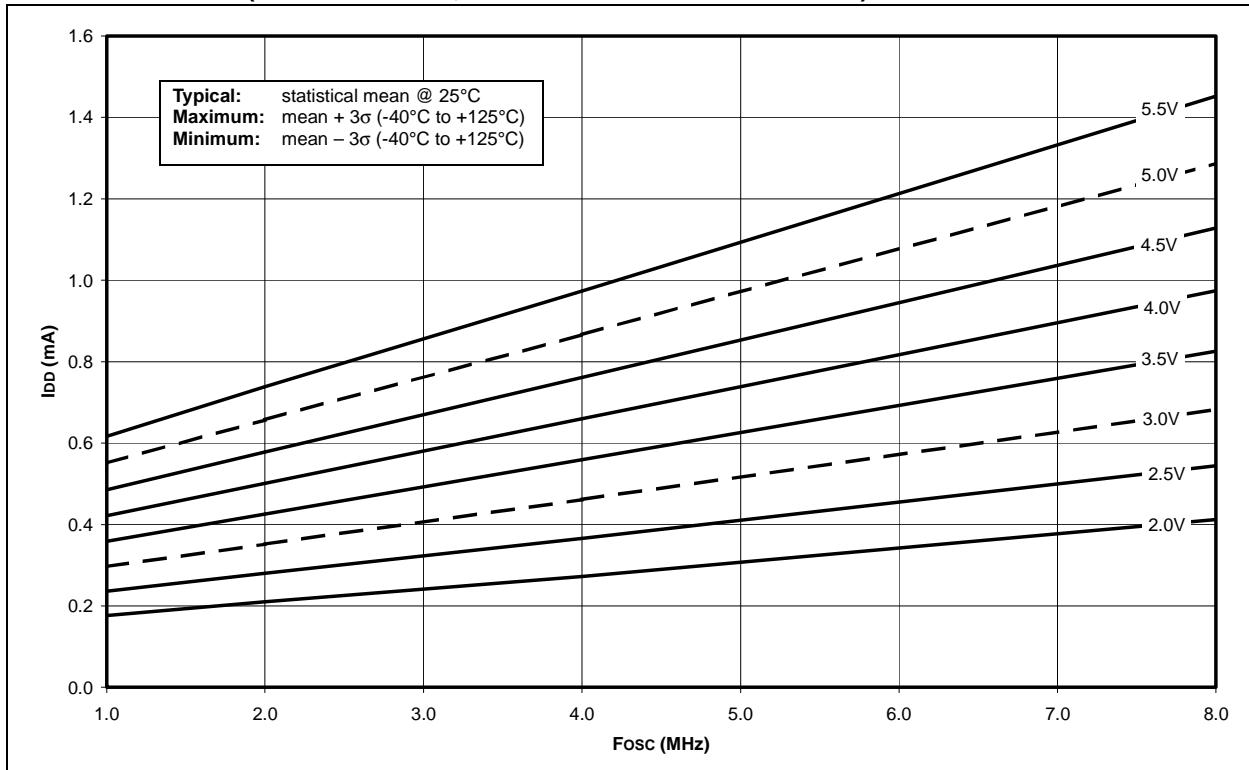


**FIGURE 19-6: MAXIMUM  $I_{DD}$  vs.  $F_{osc}$  OVER  $V_{DD}$  (LP MODE)**

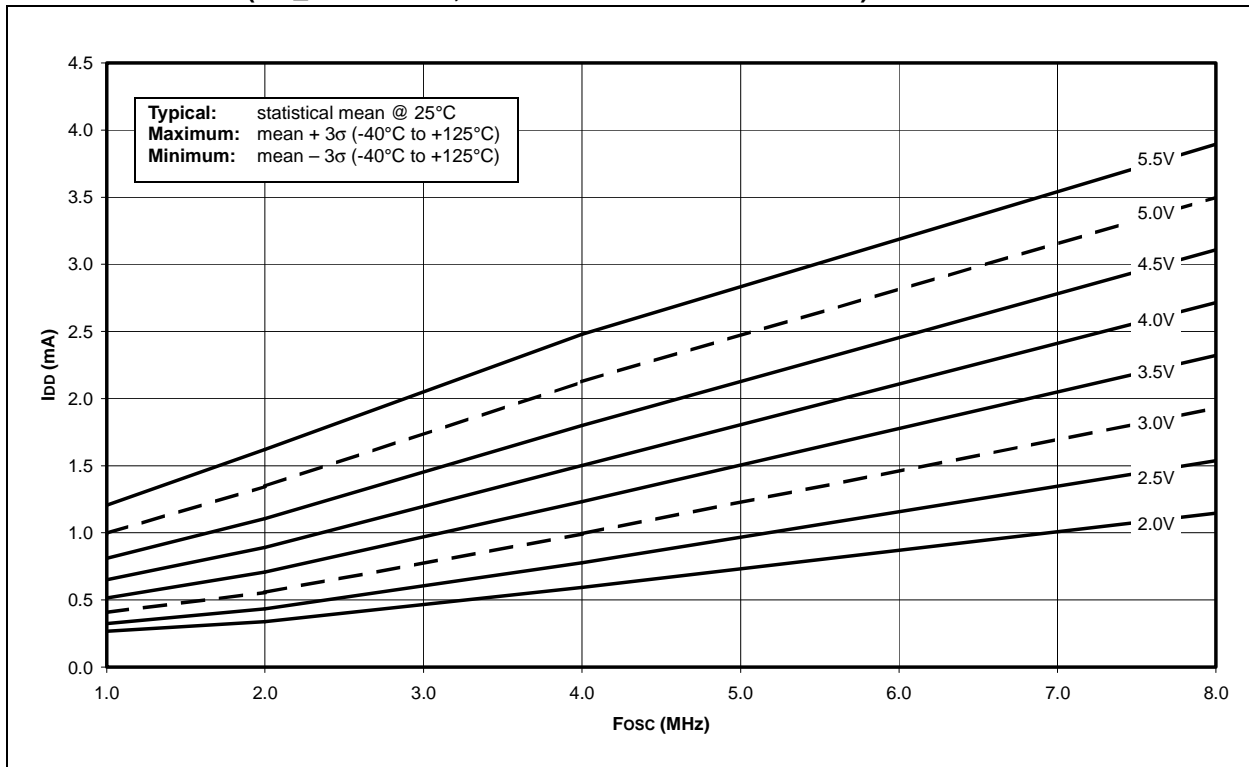


# PIC16F7X7

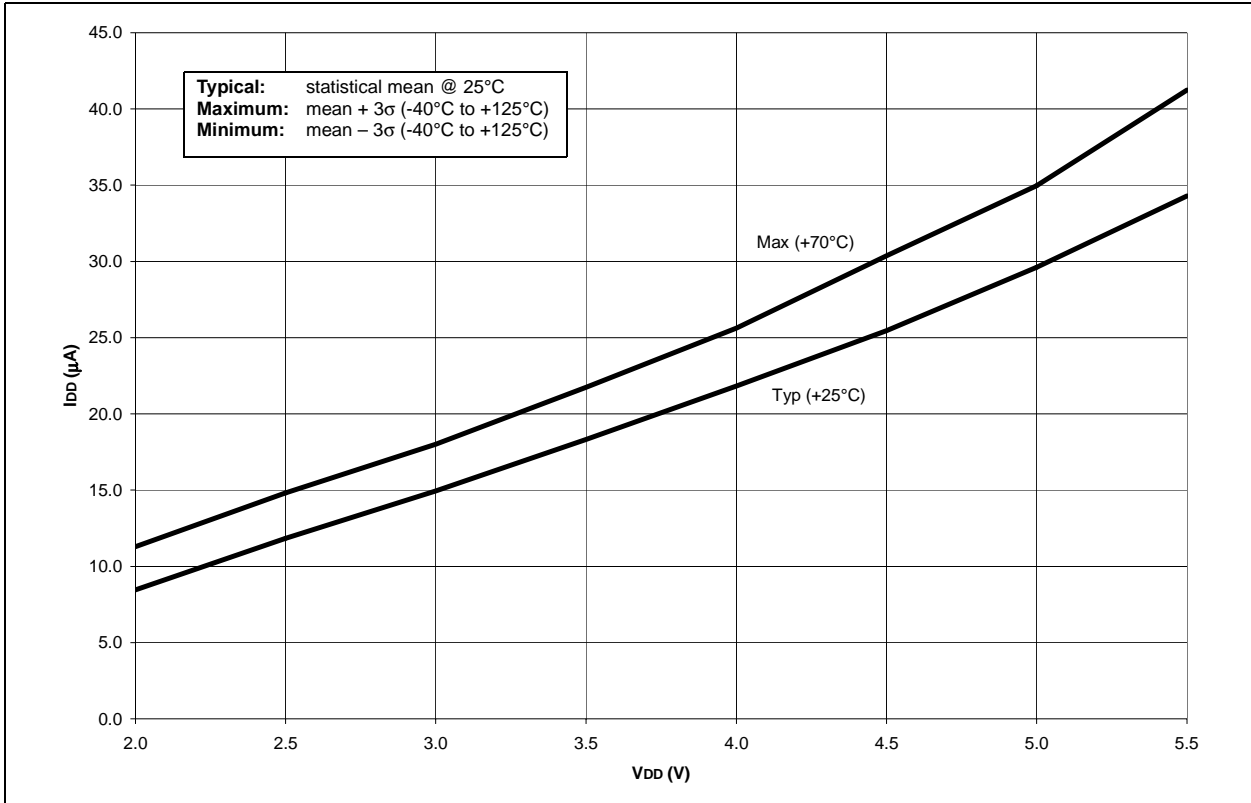
**FIGURE 19-7: TYPICAL  $I_{DD}$  vs.  $V_{DD}$ ,  $-40^{\circ}\text{C}$  TO  $+125^{\circ}\text{C}$ , 1 MHz TO 8 MHz (RC\_RUN MODE, ALL PERIPHERALS DISABLED)**



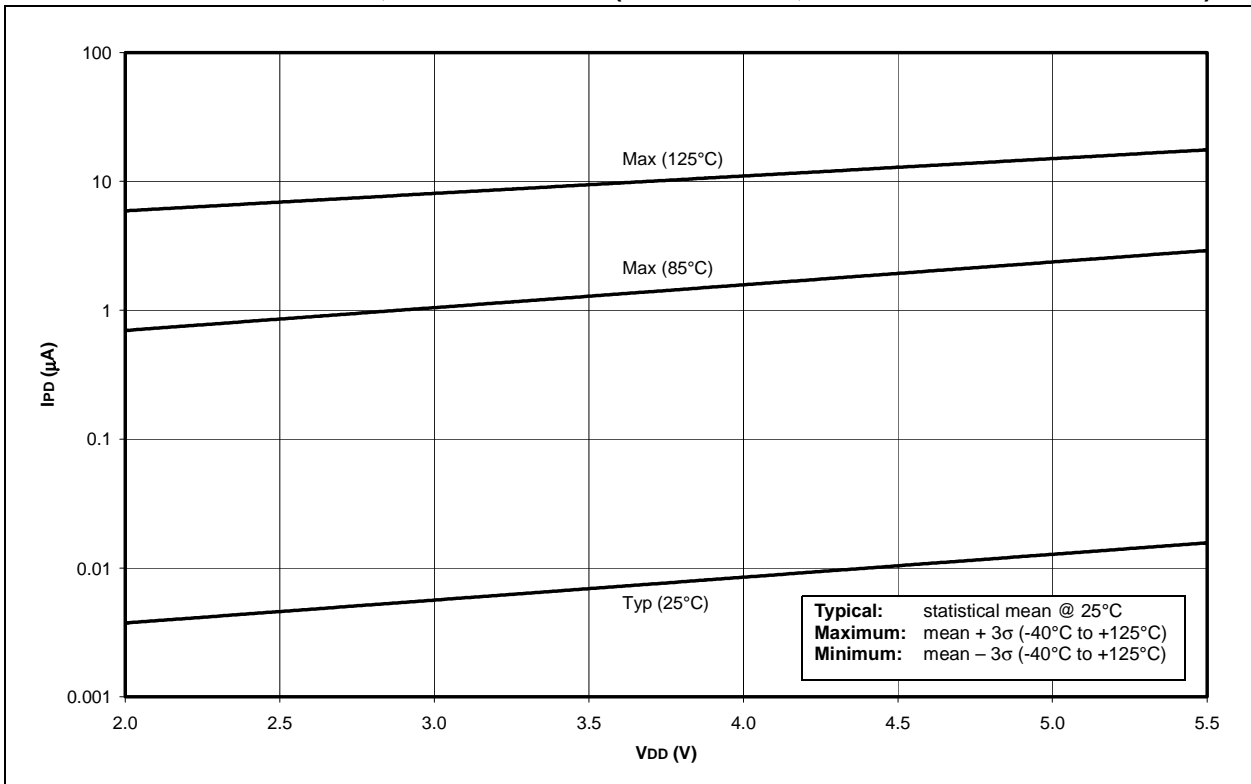
**FIGURE 19-8: MAXIMUM  $I_{DD}$  vs.  $V_{DD}$ ,  $-40^{\circ}\text{C}$  TO  $+125^{\circ}\text{C}$ , 1 MHz TO 8 MHz (RC\_RUN MODE, ALL PERIPHERALS DISABLED)**



**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)**



# PIC16F7X7

FIGURE 19-11: AVERAGE  $F_{osc}$  vs.  $V_{DD}$  FOR VARIOUS VALUES OF R (RC MODE,  $C = 20\text{ pF}$ ,  $+25^\circ\text{C}$ )

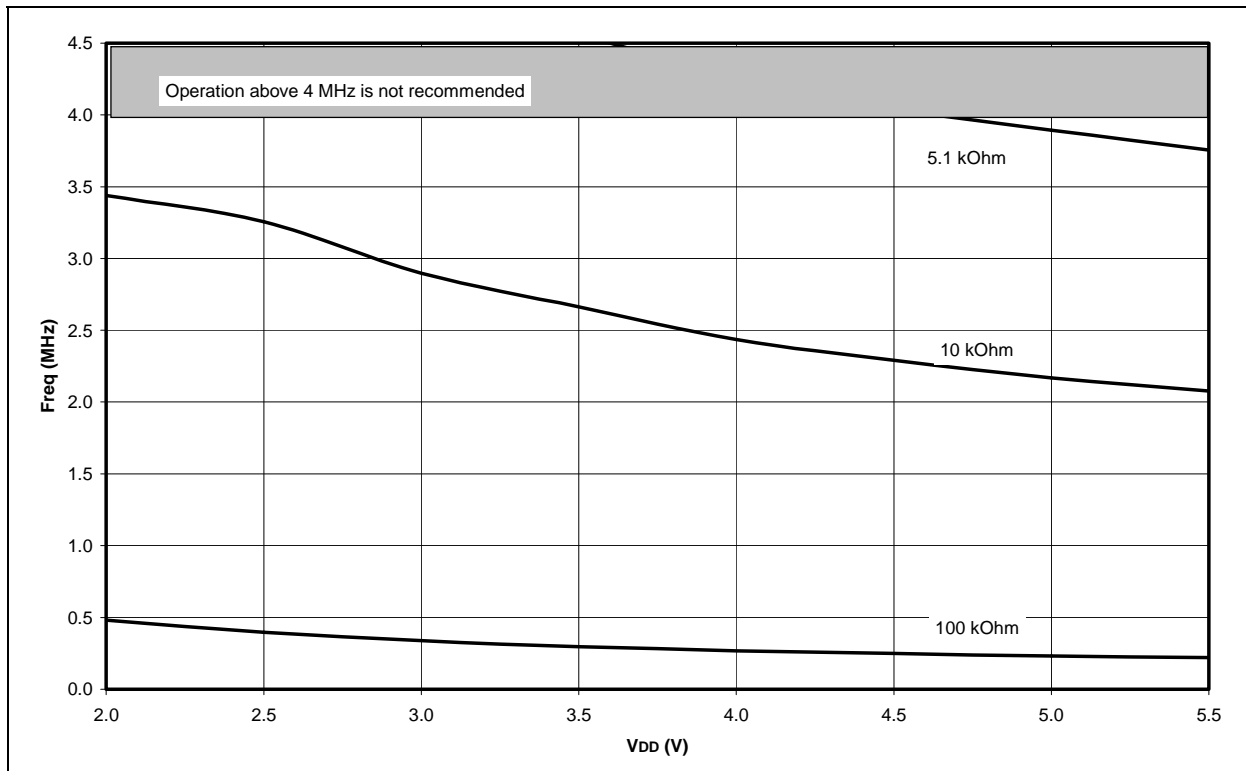
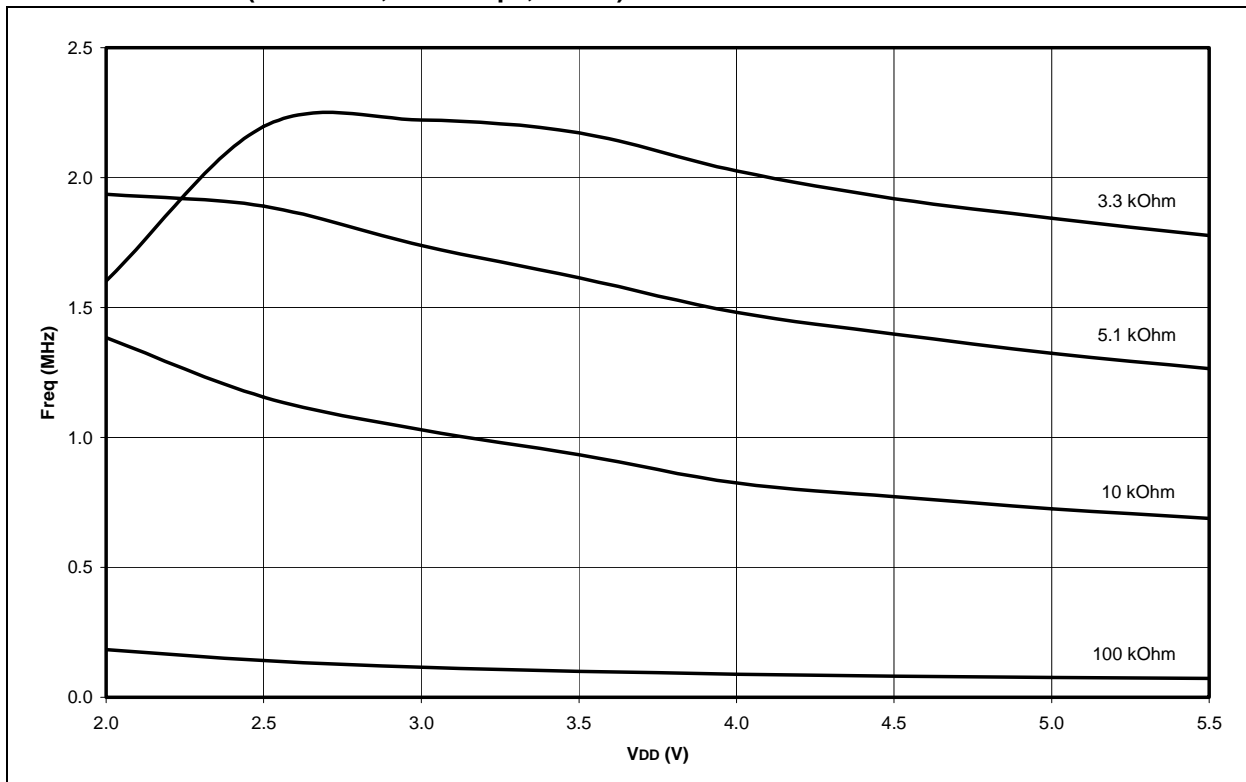
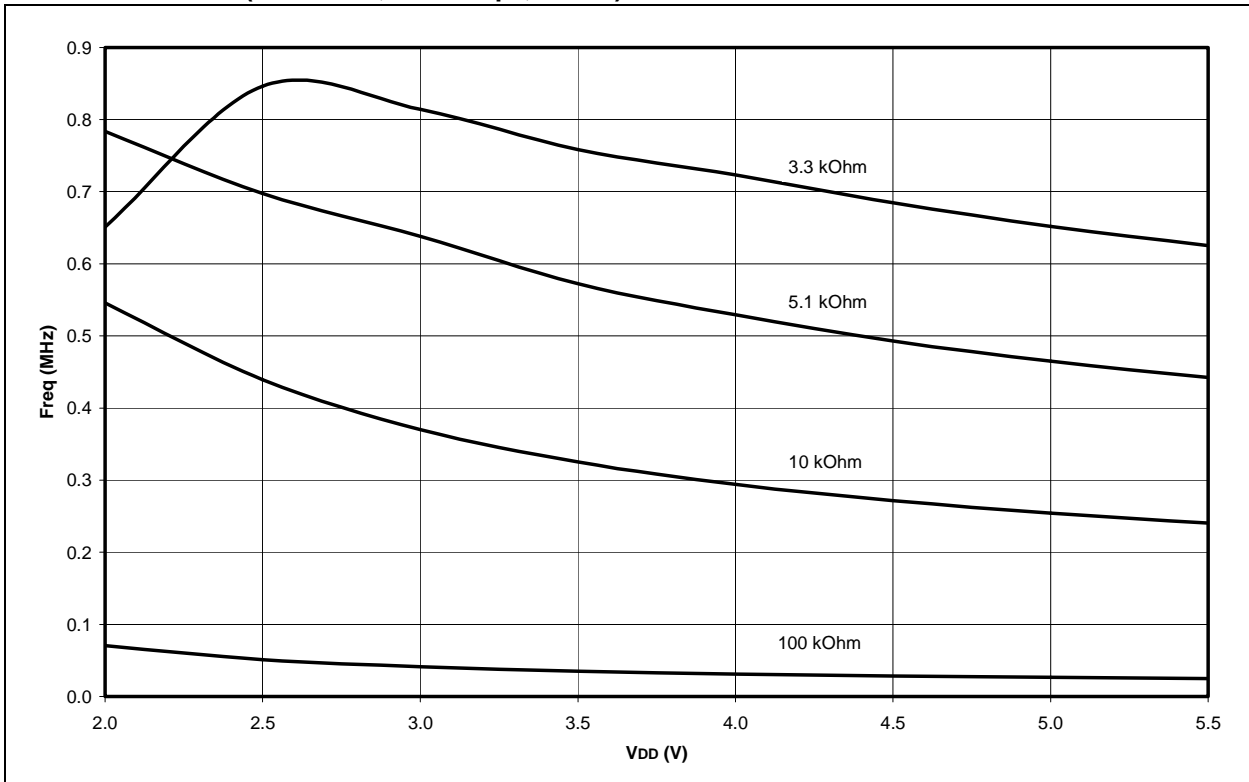


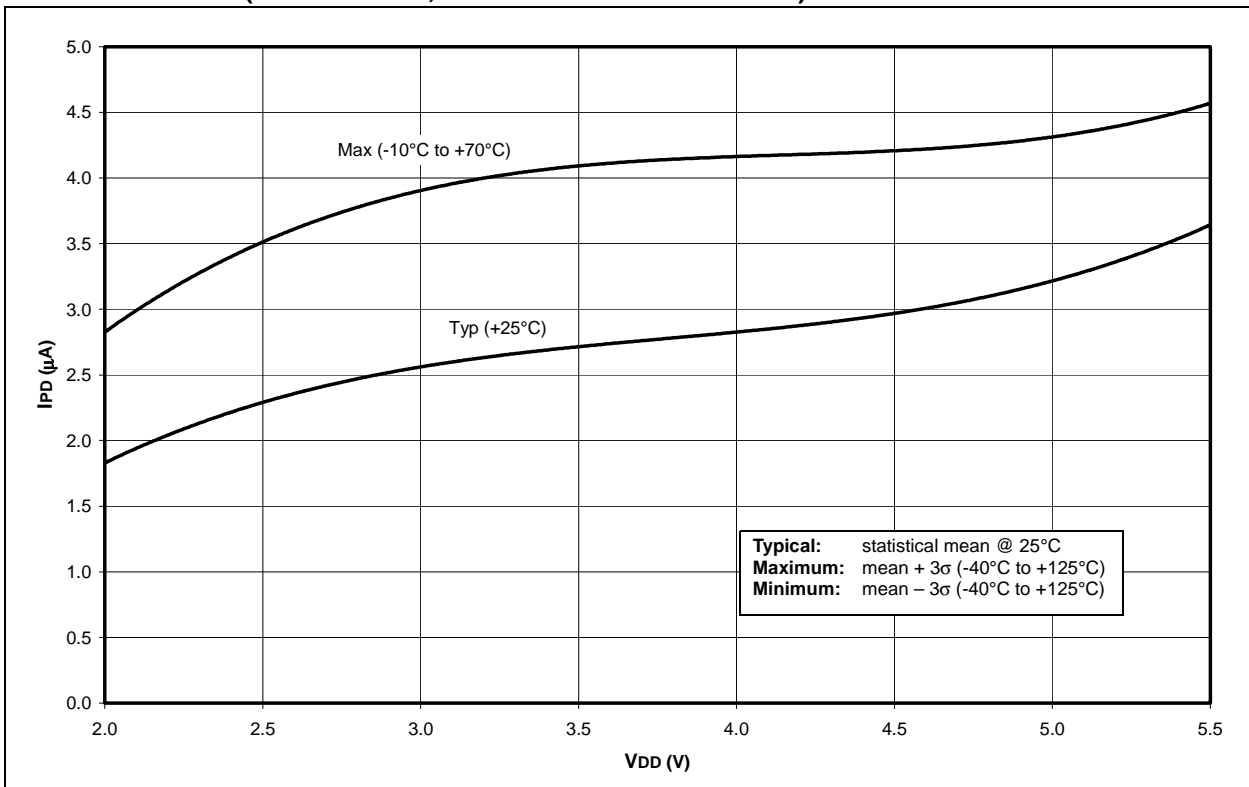
FIGURE 19-12: AVERAGE  $F_{osc}$  vs.  $V_{DD}$  FOR VARIOUS VALUES OF R (RC MODE,  $C = 100\text{ pF}$ ,  $+25^\circ\text{C}$ )



**FIGURE 19-13: AVERAGE  $F_{osc}$  vs.  $V_{DD}$  FOR VARIOUS VALUES OF R (RC MODE, C = 300 pF, +25°C)**

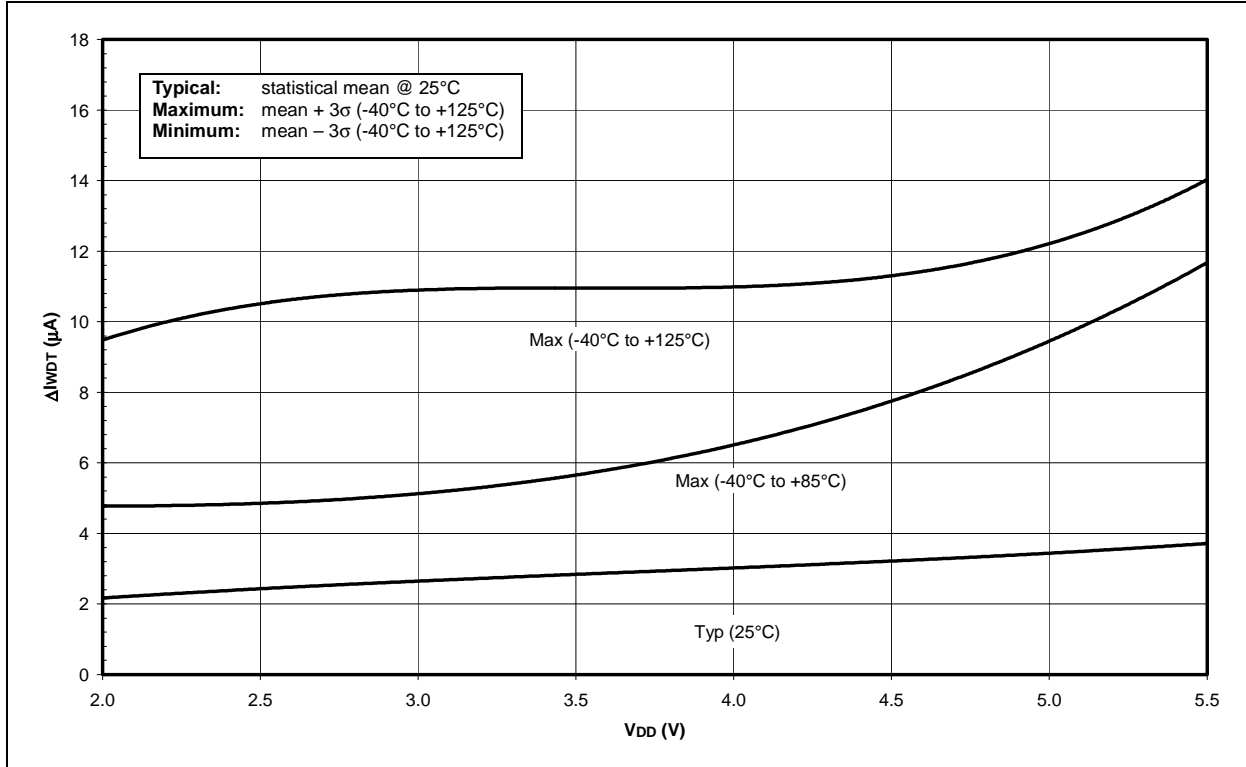


**FIGURE 19-14:  $\Delta I_{PD}$  TIMER1 OSCILLATOR, -10°C TO +70°C (SLEEP MODE, TMR1 COUNTER DISABLED)**

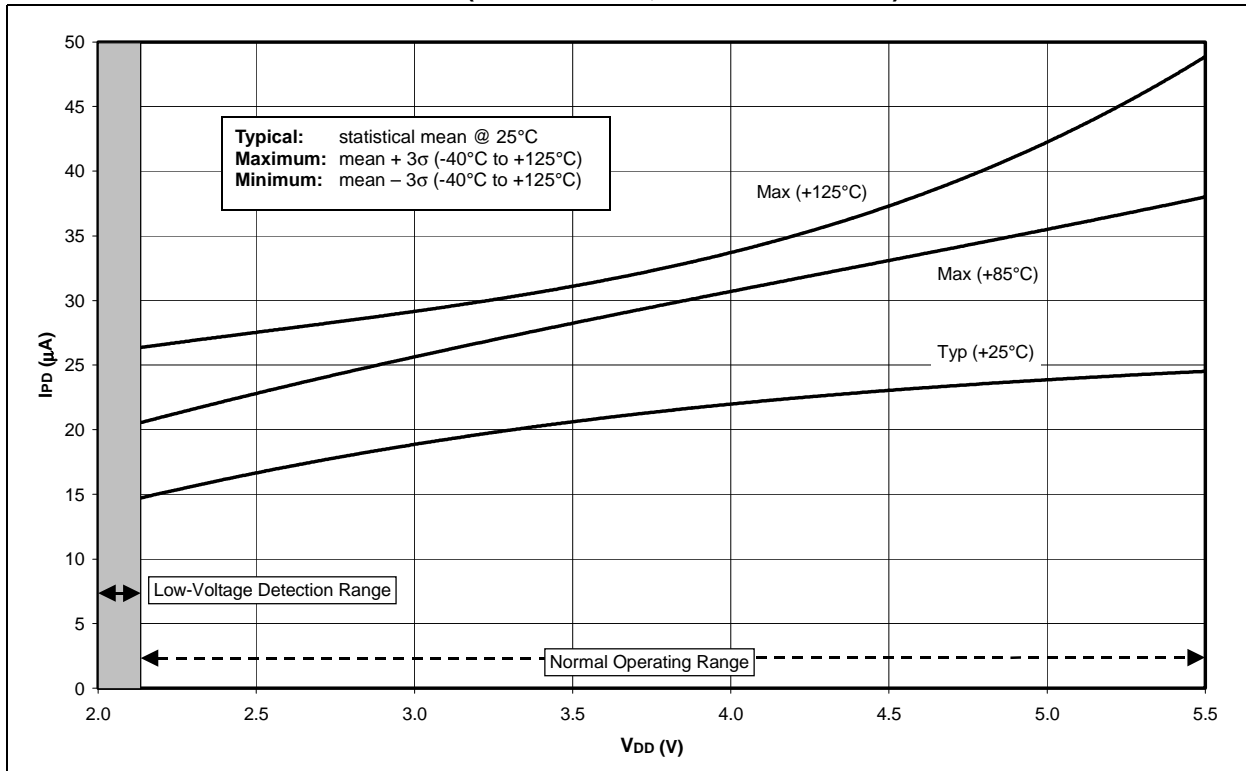


# PIC16F7X7

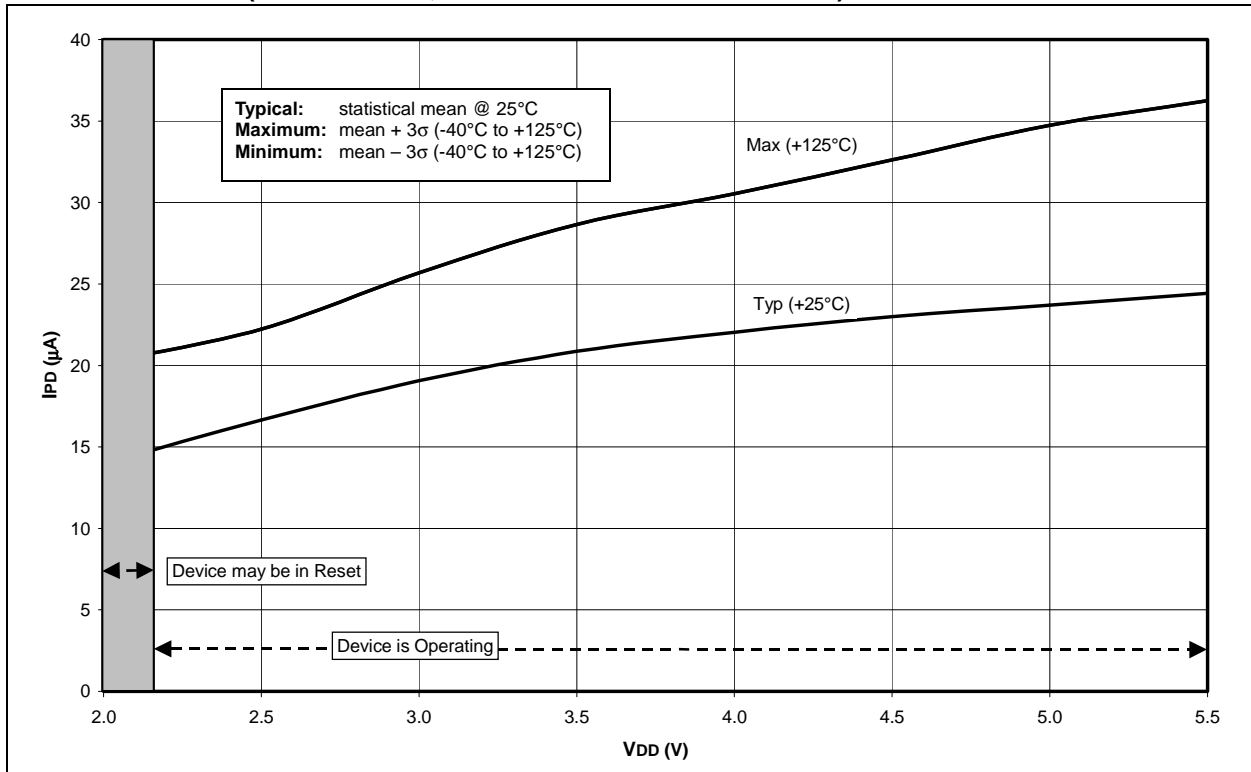
**FIGURE 19-15:  $\Delta I_{PD}$  WDT, -40°C TO +125°C (SLEEP MODE, ALL PERIPHERALS DISABLED)**



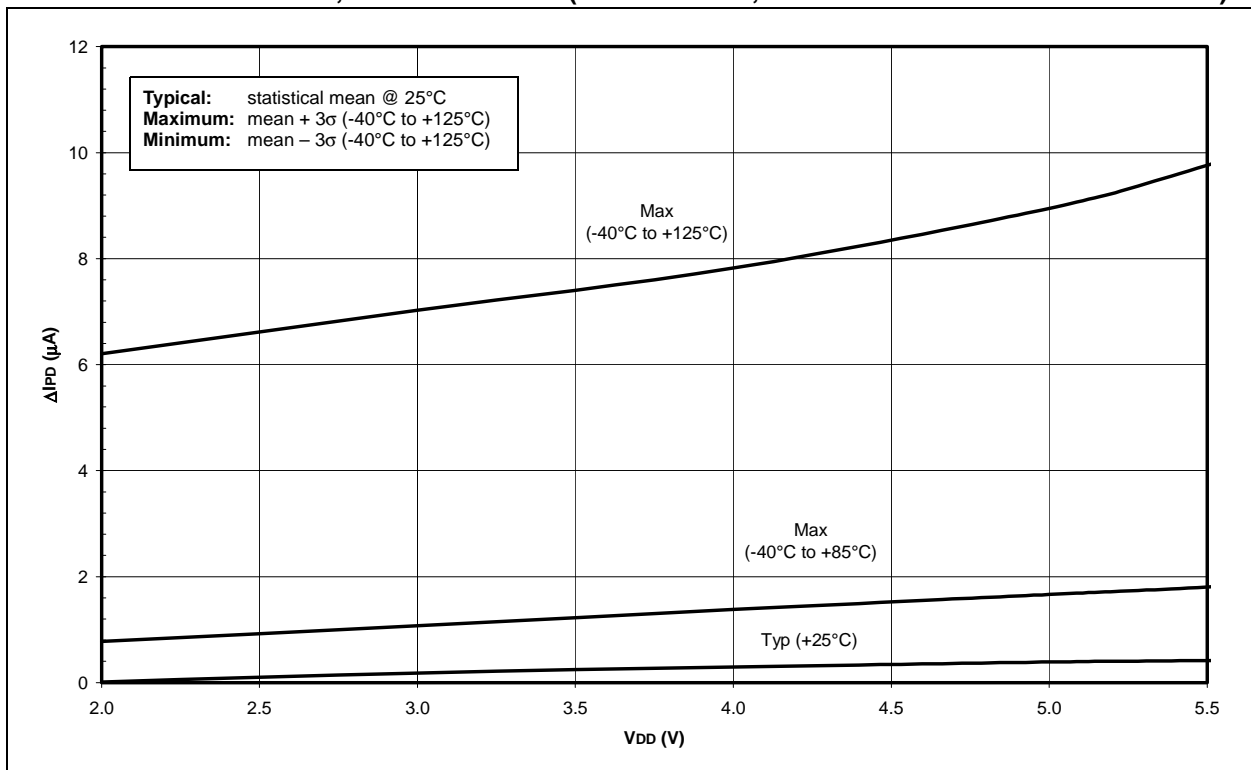
**FIGURE 19-16:  $I_{PD}$  LVD vs. V<sub>DD</sub> (SLEEP MODE, LVD = 2.00V-2.12V)**



**FIGURE 19-17:  $\Delta I_{PD}$  BOR vs.  $V_{DD}$ , -40°C TO +125°C  
(SLEEP MODE, BOR ENABLED AT 2.00V-2.16V)**



**FIGURE 19-18:  $\Delta I_{PD}$  A/D, -40°C TO +125°C (SLEEP MODE, A/D ENABLED – NOT CONVERTING)**



# PIC16F7X7

FIGURE 19-19: TYPICAL, MINIMUM AND MAXIMUM  $V_{OH}$  vs.  $I_{OH}$  ( $V_{DD} = 5V$ ,  $-40^{\circ}C$  TO  $+125^{\circ}C$ )

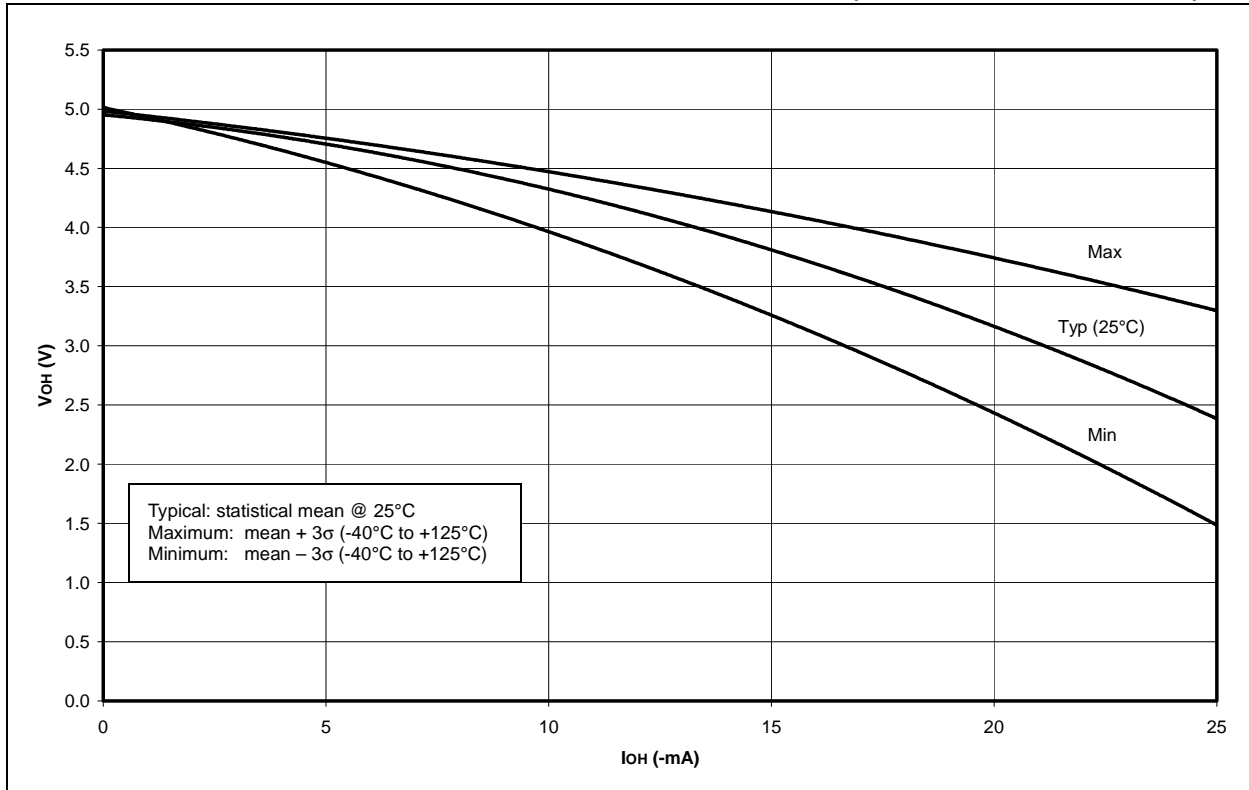
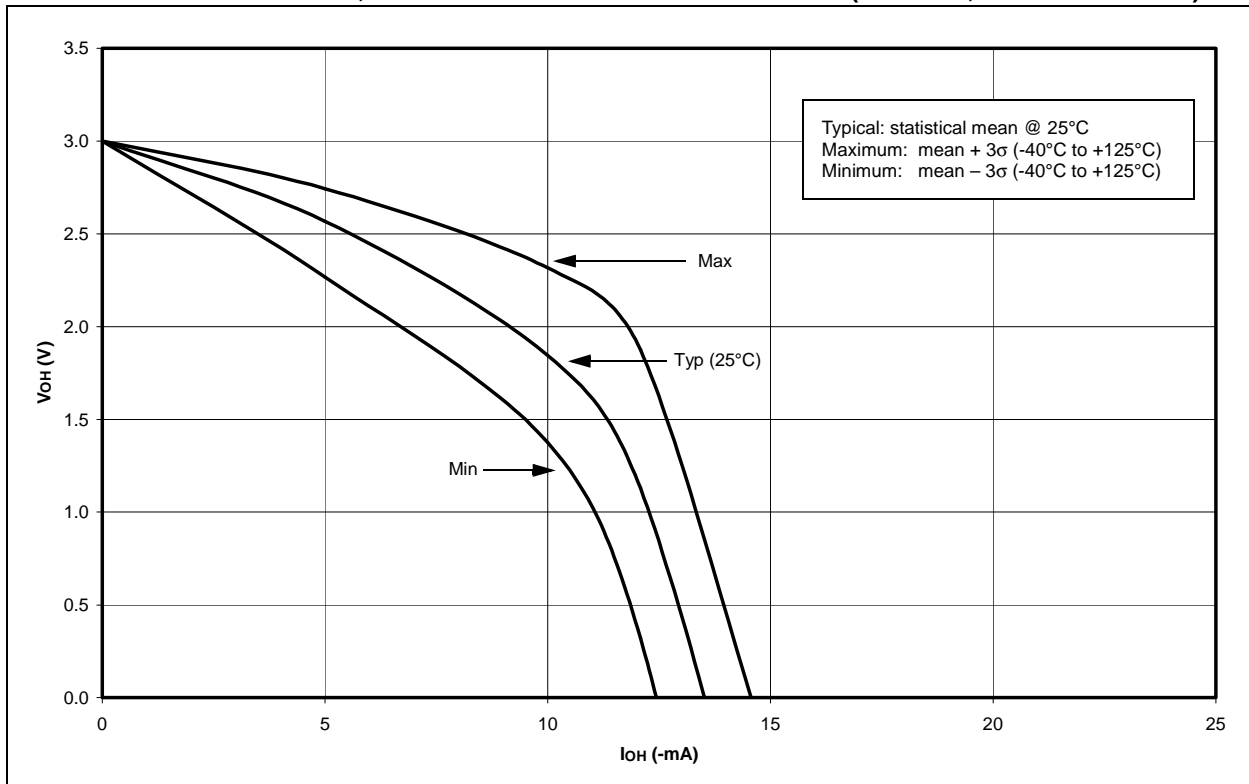
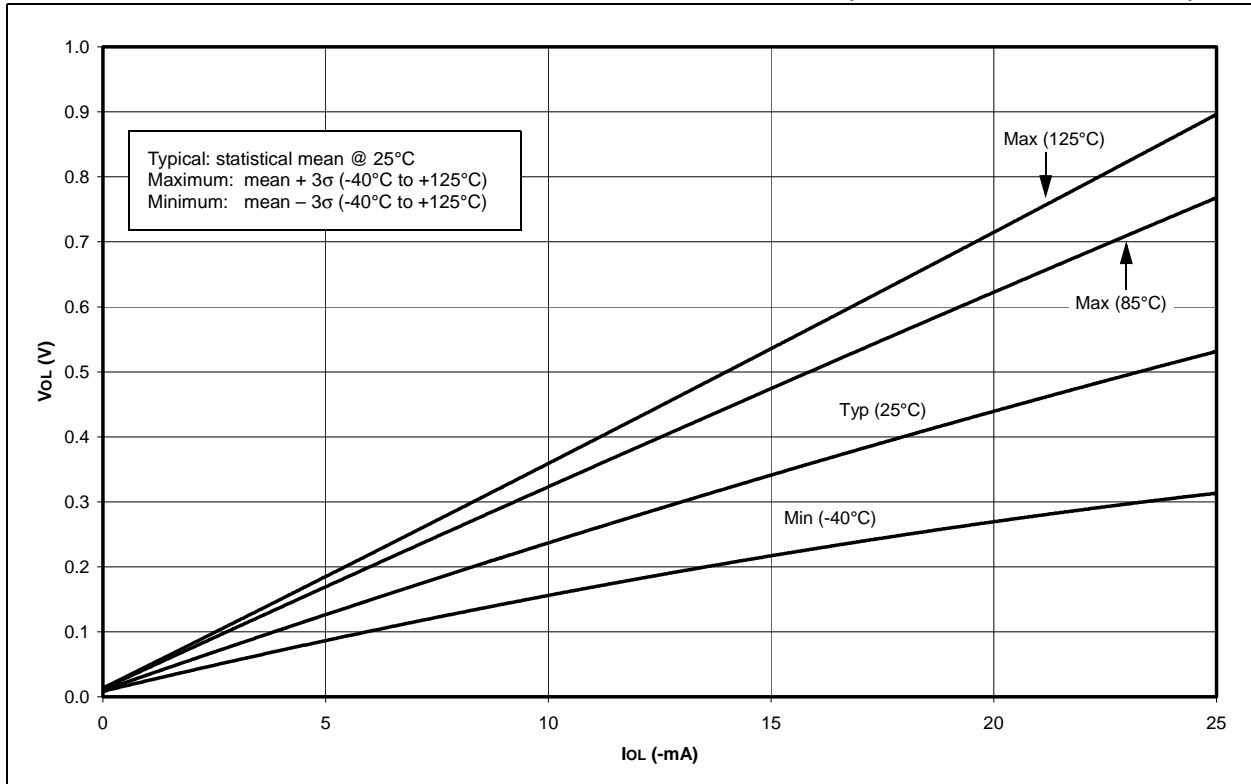


FIGURE 19-20: TYPICAL, MINIMUM AND MAXIMUM  $V_{OH}$  vs.  $I_{OH}$  ( $V_{DD} = 3V$ ,  $-40^{\circ}C$  TO  $+125^{\circ}C$ )

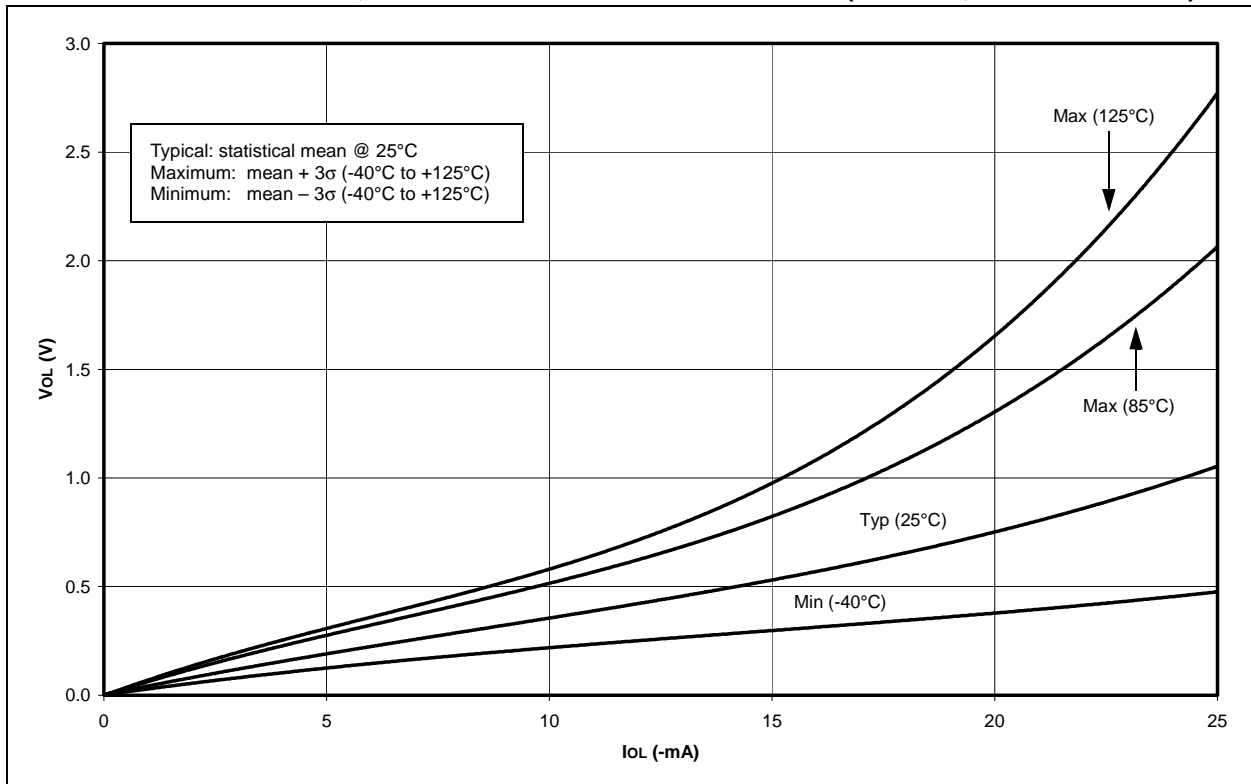




**FIGURE 19-21: TYPICAL, MINIMUM AND MAXIMUM  $V_{OL}$  vs.  $I_{OL}$  ( $V_{DD} = 5V$ ,  $-40^{\circ}C$  TO  $+125^{\circ}C$ )**



**FIGURE 19-22: TYPICAL, MINIMUM AND MAXIMUM  $V_{OL}$  vs.  $I_{OL}$  ( $V_{DD} = 3V$ ,  $-40^{\circ}C$  TO  $+125^{\circ}C$ )**



# PIC16F7X7

FIGURE 19-23: MINIMUM AND MAXIMUM  $V_{IN}$  vs.  $V_{DD}$  (TTL INPUT,  $-40^{\circ}\text{C}$  TO  $+125^{\circ}\text{C}$ )

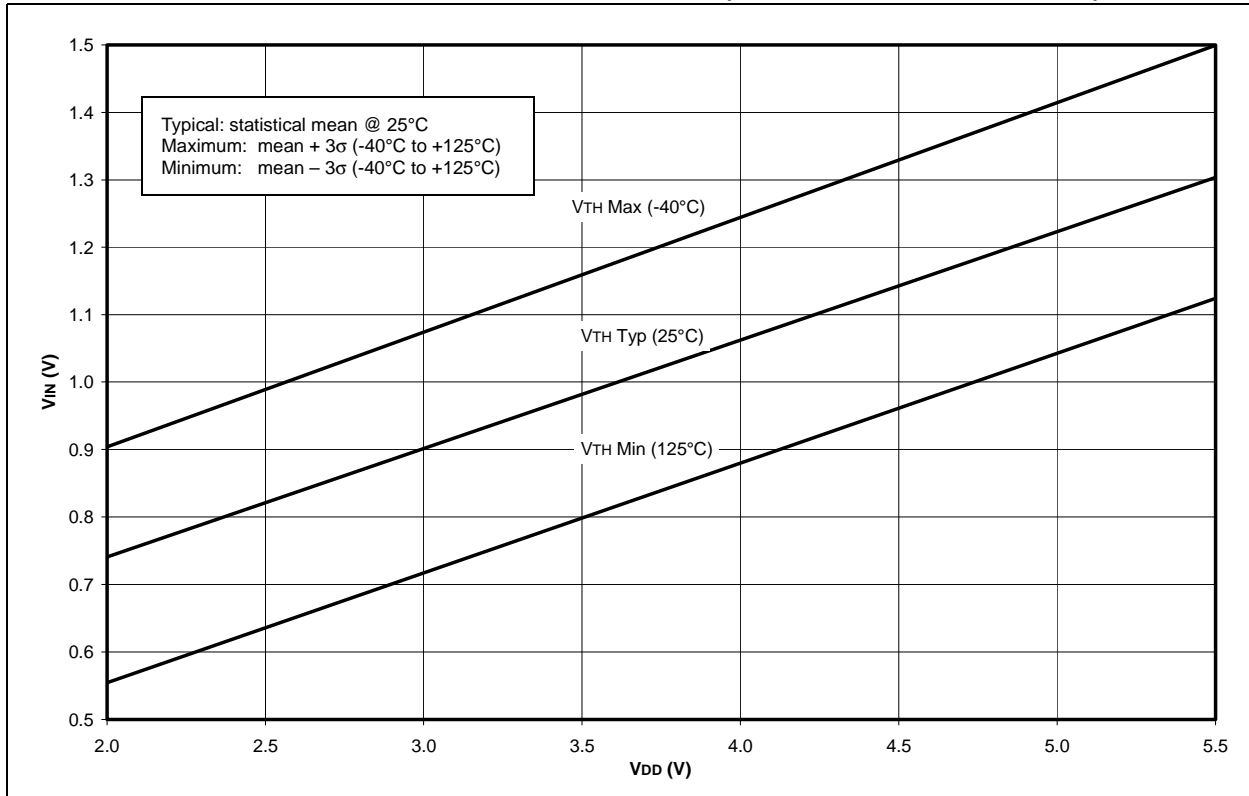
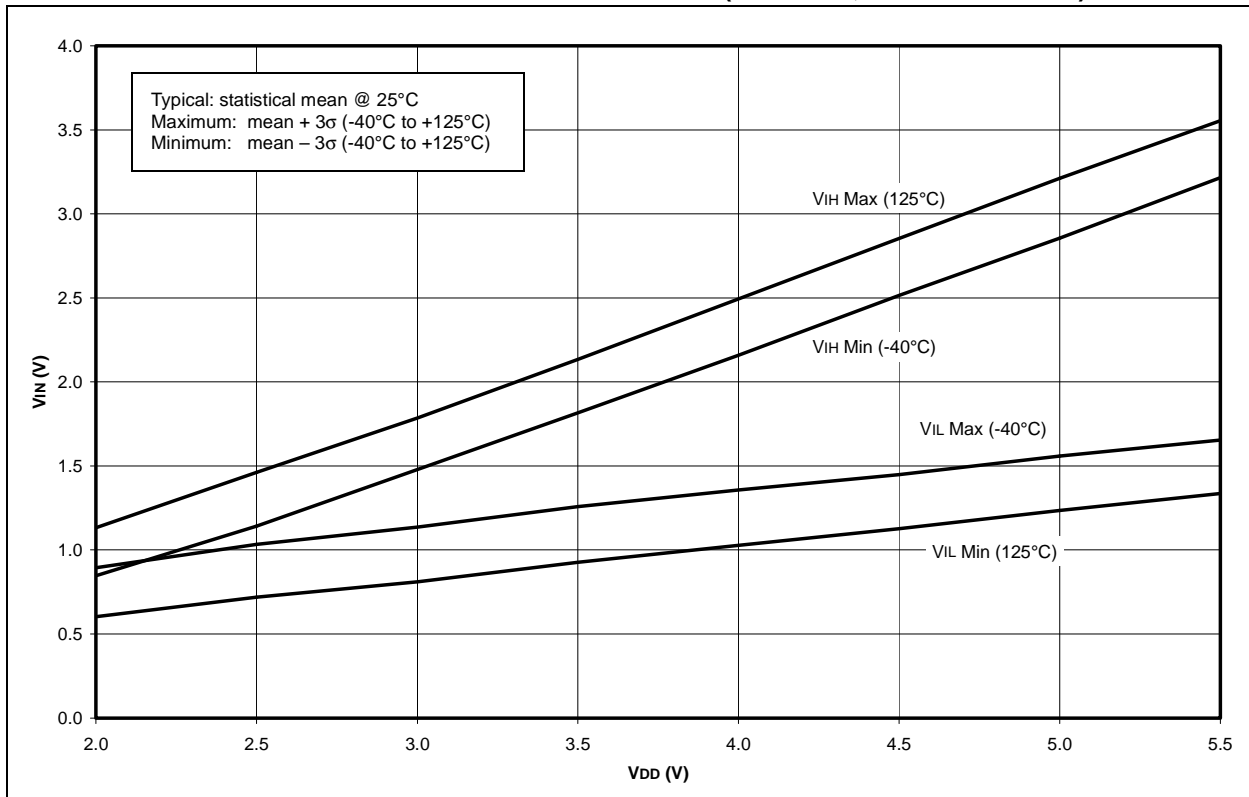
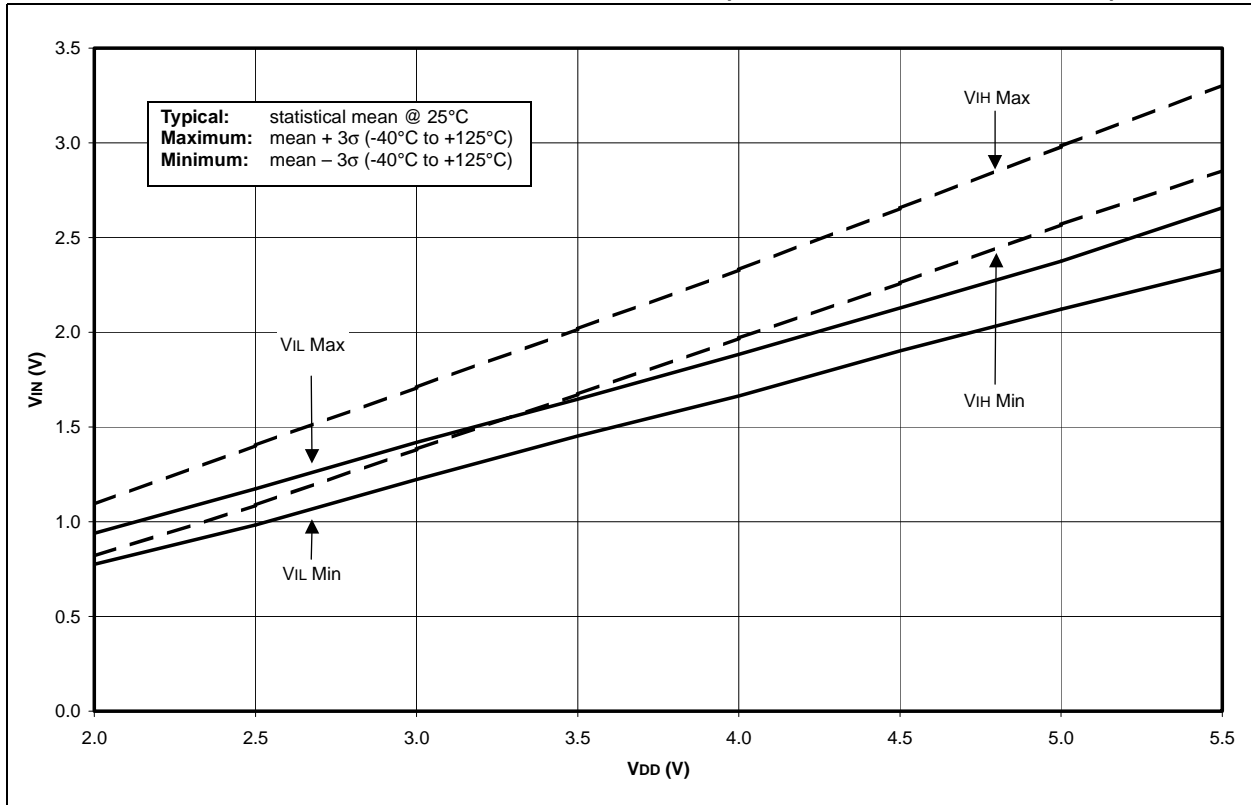


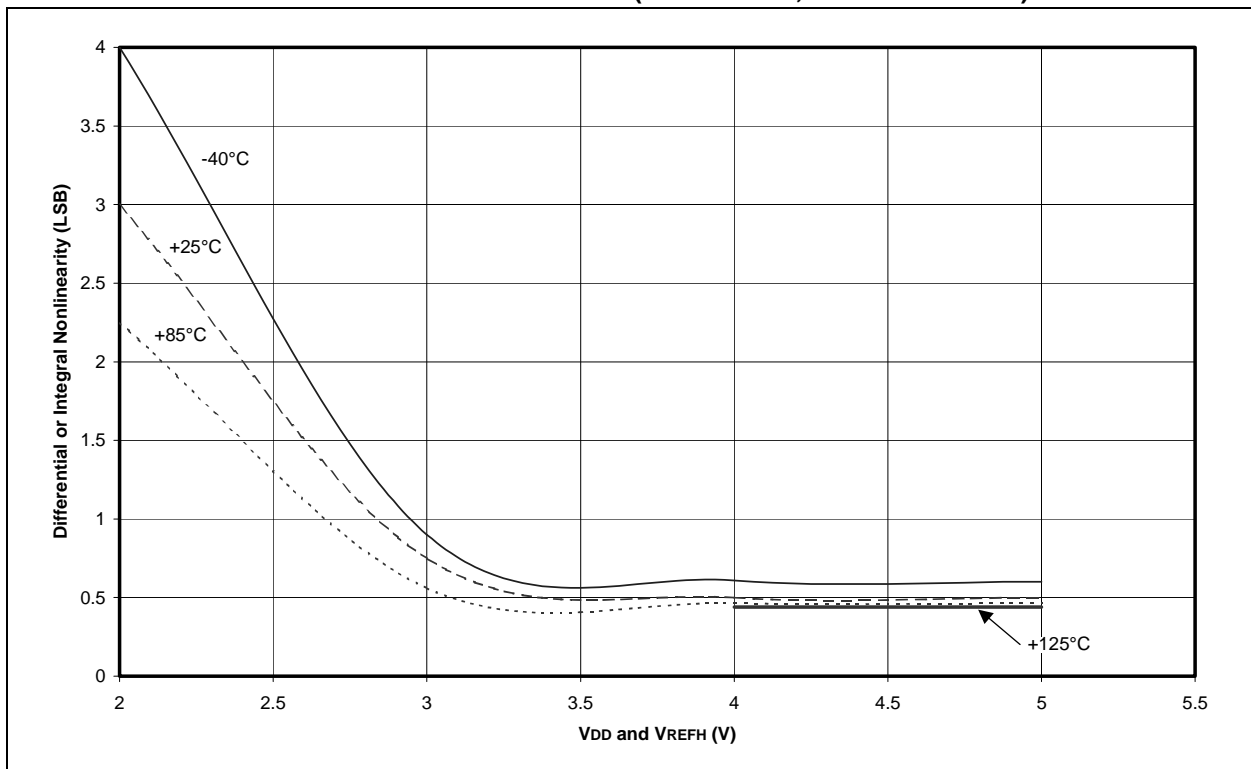
FIGURE 19-24: MINIMUM AND MAXIMUM  $V_{IN}$  vs.  $V_{DD}$  (ST INPUT,  $-40^{\circ}\text{C}$  TO  $+125^{\circ}\text{C}$ )



**FIGURE 19-25: MINIMUM AND MAXIMUM  $V_{IN}$  vs.  $V_{DD}$  ( $I^2C^{\text{TM}}$  INPUT,  $-40^{\circ}\text{C}$  TO  $+125^{\circ}\text{C}$ )**

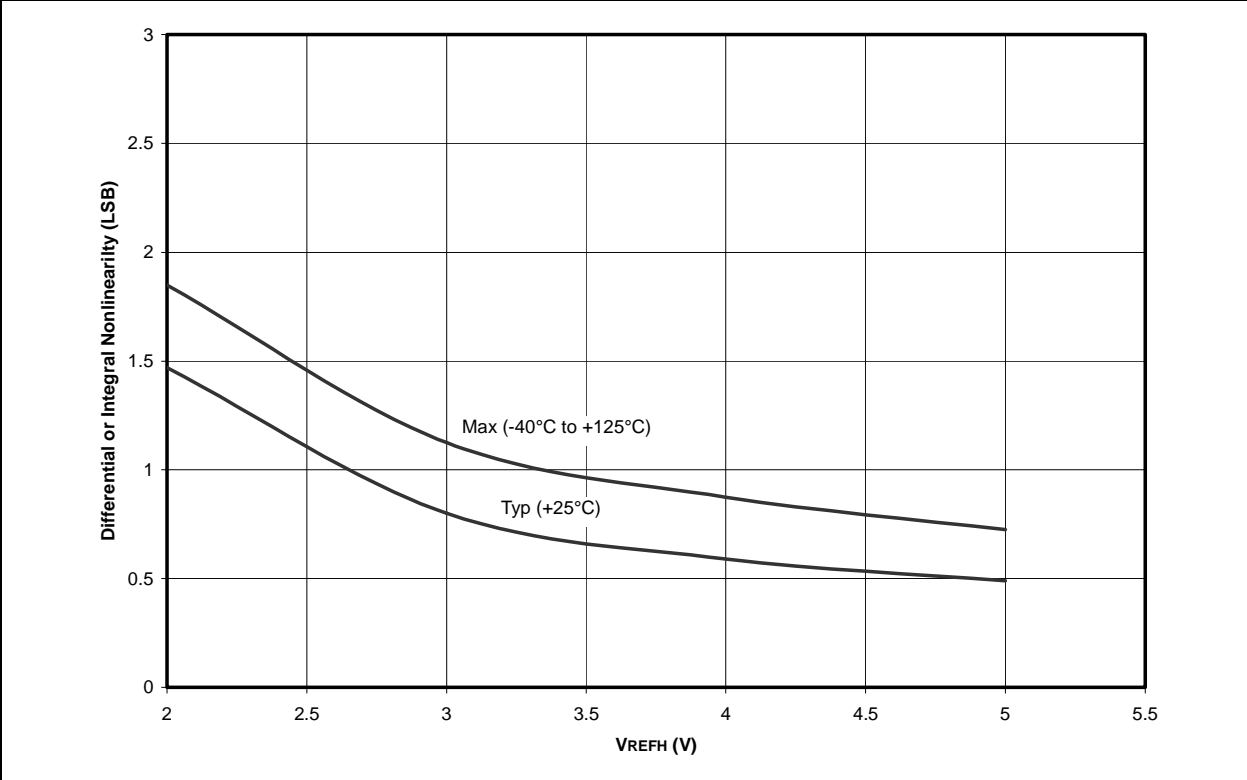


**FIGURE 19-26: A/D NONLINEARITY vs.  $V_{REFH}$  ( $V_{DD} = V_{REFH}$ ,  $-40^{\circ}\text{C}$  TO  $+125^{\circ}\text{C}$ )**



# PIC16F7X7

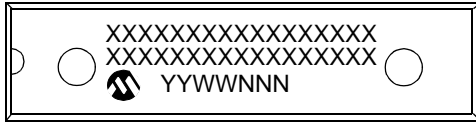
FIGURE 19-27: A/D NONLINEARITY vs. VREFH (VDD = 5V, -40°C TO +125°C)



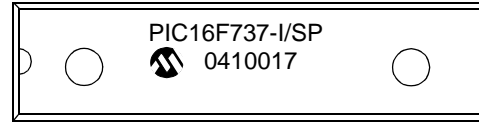
## 20.0 PACKAGING INFORMATION

### 20.1 Package Marking Information

28-Lead PDIP (Skinny DIP)



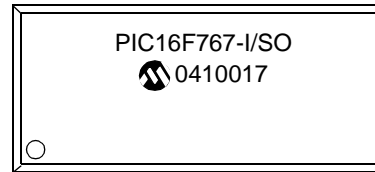
Example



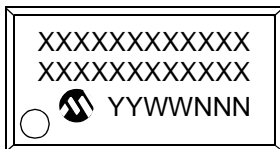
28-Lead SOIC



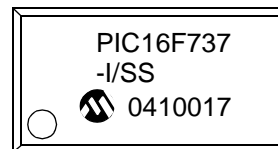
Example



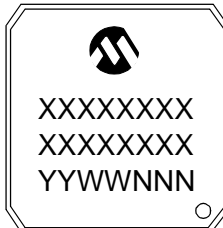
28-Lead SSOP



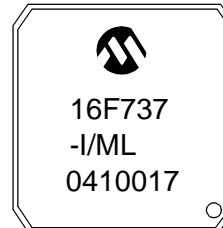
Example



28-Lead QFN



Example



<b>Legend:</b>	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
<b>Note:</b> In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line thus limiting the number of available characters for customer specific information.		

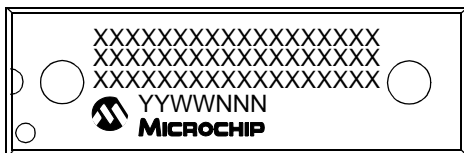
\* Standard PICmicro device marking consists of Microchip part number, year code, week code, and traceability code. For PICmicro device marking beyond this, certain price adders apply. Please check with your Microchip Sales Office. For QTP devices, any special marking adders are included in QTP price.

# PIC16F7X7

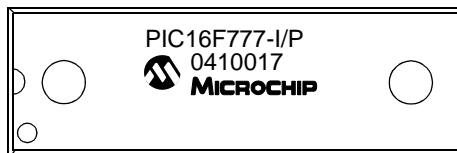
---

## Package Marking Information (Continued)

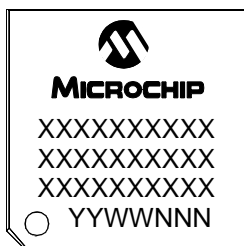
40-Lead PDIP



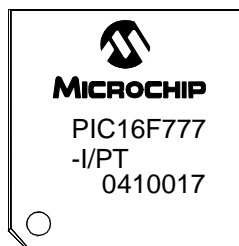
Example



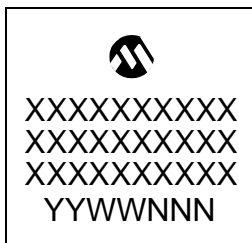
44-Lead TQFP



Example



44-Lead QFN



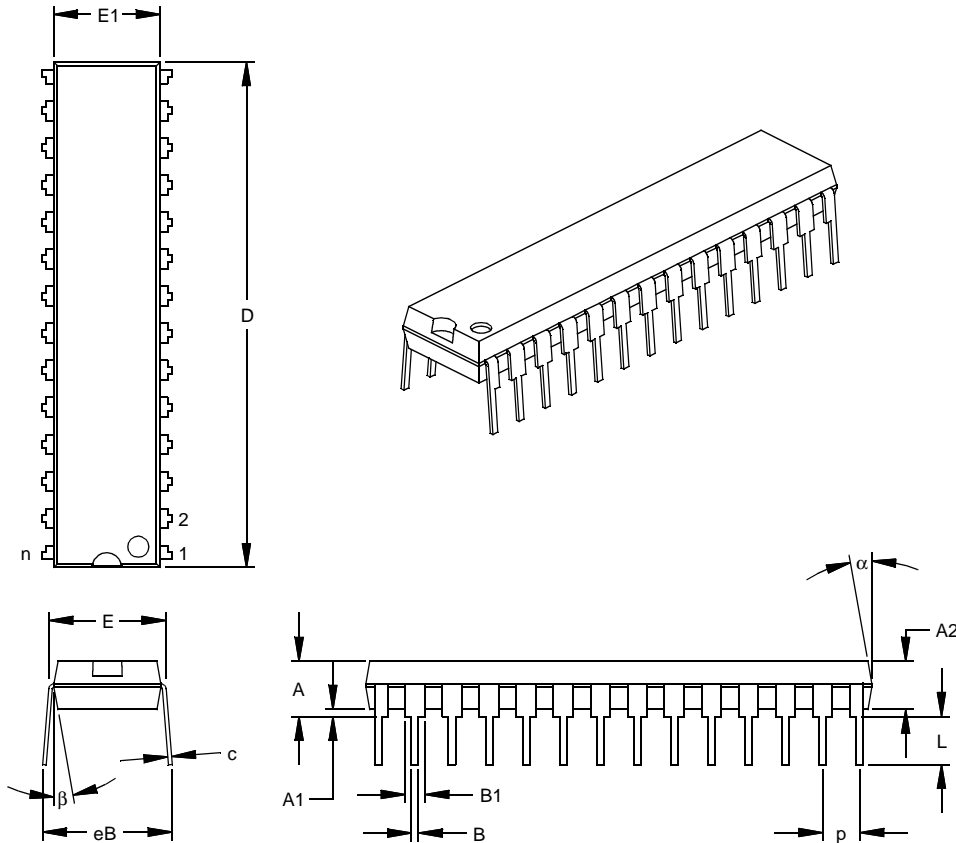
Example



## 20.2 Package Details

The following sections give the technical details of the packages.

### 28-Lead Skinny Plastic Dual In-line (SP) – 300 mil Body (PDIP)



		Units	INCHES*			MILLIMETERS		
Dimension Limits			MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		28			28		
Pitch	p			.100			2.54	
Top to Seating Plane	A		.140	.150	.160	3.56	3.81	4.06
Molded Package Thickness	A2		.125	.130	.135	3.18	3.30	3.43
Base to Seating Plane	A1		.015			0.38		
Shoulder to Shoulder Width	E		.300	.310	.325	7.62	7.87	8.26
Molded Package Width	E1		.275	.285	.295	6.99	7.24	7.49
Overall Length	D		1.345	1.365	1.385	34.16	34.67	35.18
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		.040	.053	.065	1.02	1.33	1.65
Lower Lead Width	B		.016	.019	.022	0.41	0.48	0.56
Overall Row Spacing	§ eB		.320	.350	.430	8.13	8.89	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:

Dimension D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed

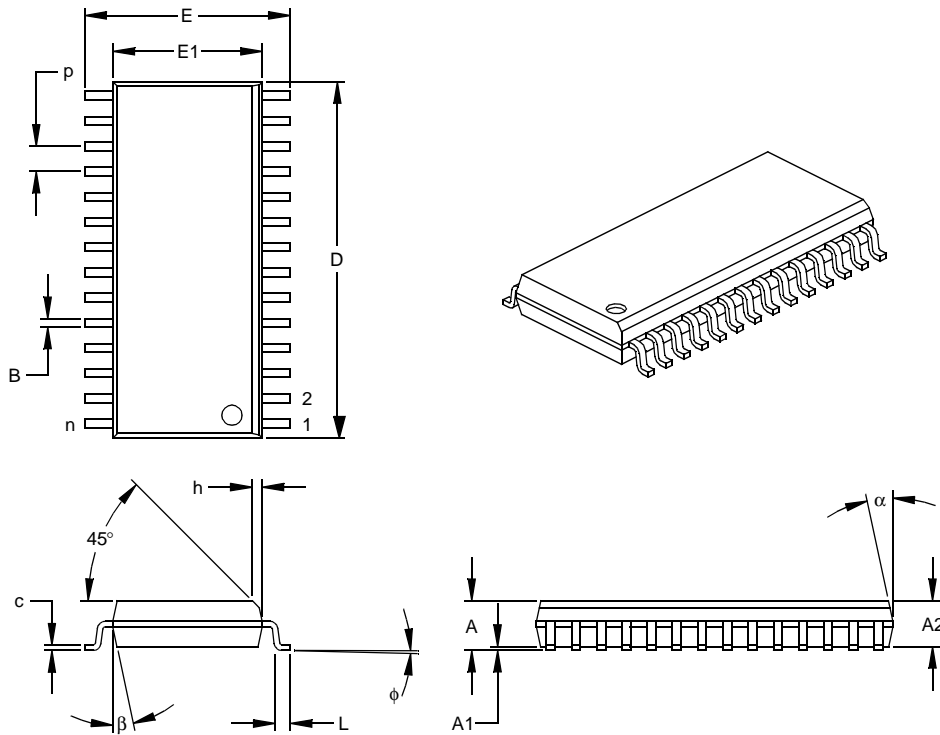
.010" (0.254mm) per side.

JEDEC Equivalent: MO-095

Drawing No. C04-070

# PIC16F7X7

## 28-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	28			28		
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	.288	.295	.299	7.32	7.49	7.59
Overall Length	D	.695	.704	.712	17.65	17.87	18.08
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 Top	φ	0	4	8	0	4	8
Lead Thickness	c	.009	.011	.013	0.23	0.28	0.33
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:

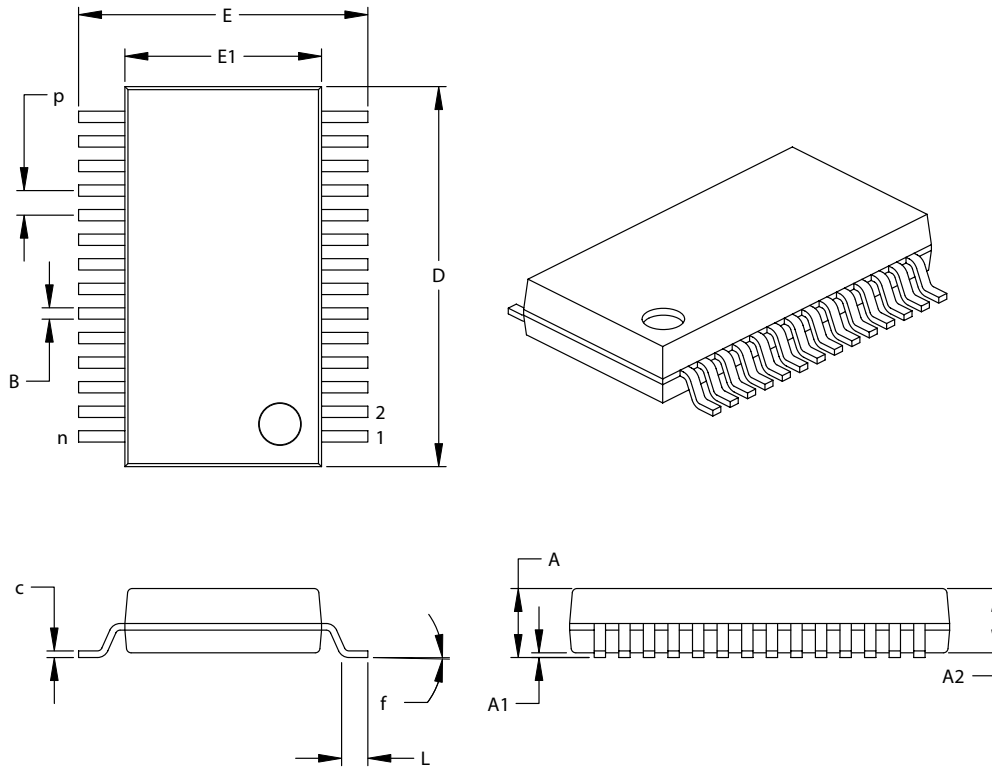
Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" (0.254mm) per side.

JEDEC Equivalent: MS-013

Drawing No. C04-052



## 28-Lead Plastic Shrink Small Outline (SS) – 209 mil Body, 5.30 mm (SSOP)



Dimension Limits	Units	INCHES			MILLIMETERS*		
		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n	28			28		
Pitch	P		.026			0.65	
Overall Height	A	-	-	.079	-	-	2.0
Molded Package Thickness	A2	.065	.069	.073	1.65	1.75	1.85
Standoff	A1	.002	-	-	0.05	-	-
Overall Width	E	.295	.307	.323	7.49	7.80	8.20
Molded Package Width	E1	.009	.209	.220	5.00	5.30	5.60
Overall Length	D	.390	.402	.413	9.90	10.20	10.50
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:

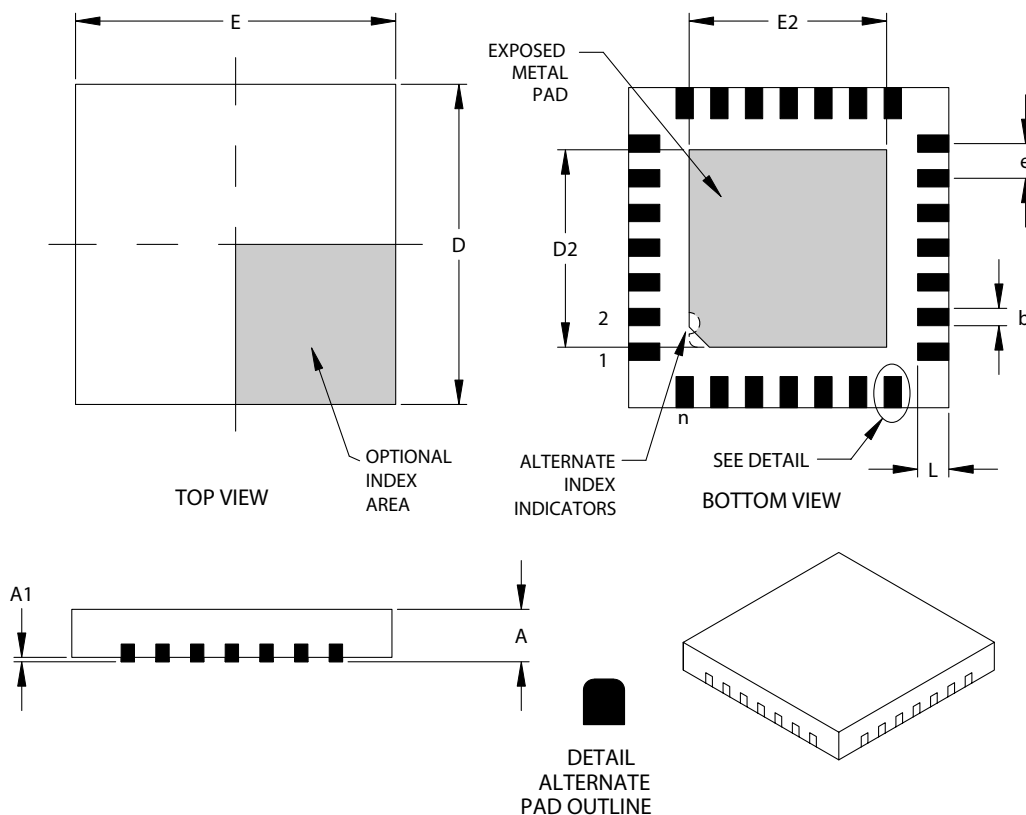
Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" (0.254mm) per side.

JEDEC Equivalent: MO-150

Drawing No. C04-073

# PIC16F7X7

## 28-Lead Plastic Quad Flat No Lead Package (ML) 6x6 mm Body (QFN) – With 0.55 mm Contact Length (Saw Singulated)



Dimension Limits	Units	INCHES			MILLIMETERS*		
		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n	28			28		
Pitch	e	.026 BSC			0.65 BSC		
Overall Height	A	.031	.035	.039	0.80	0.90	1.00
Standoff	A1	.000	.001	.002	0.00	0.02	0.05
Contact Thickness	A3	.008 REF			0.20 REF		
Overall Width	E	.232	.236	.240	5.90	6.00	6.10
Exposed Pad Width	E2	.140	.146	.152	3.55	3.70	3.85
Overall Length	D	.232	.236	.240	5.90	6.00	6.10
Exposed Pad Length	D2	.140	.146	.152	3.55	3.70	3.85
Contact Width	b	.009	.011	.013	0.23	0.28	0.33
Contact Length	L	.020	.024	.028	0.50	0.60	0.70

\*Controlling Parameter

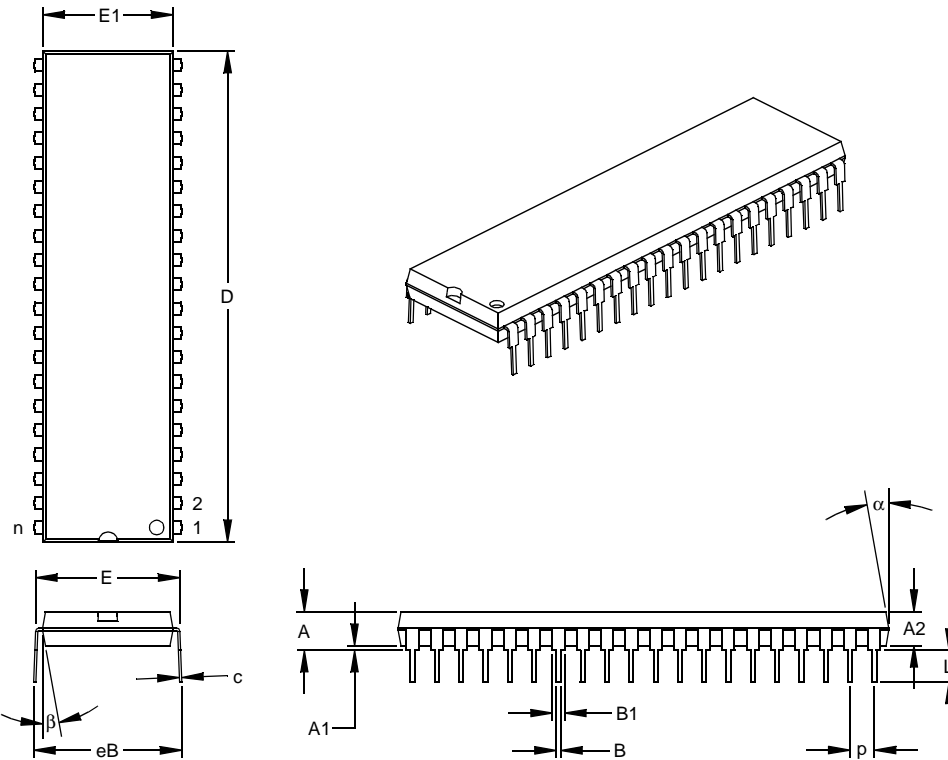
Notes:

JEDEC equivalent: MO-220

Drawing No. C04-105

Revised 05-24-04

## 40-Lead Plastic Dual In-line (P) – 600 mil Body (PDIP)



Units		INCHES*			MILLIMETERS		
Dimension Limits		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n	40			40		
Pitch	p		.100			2.54	
Top to Seating Plane	A	.160	.175	.190	4.06	4.45	4.83
Molded Package Thickness	A2	.140	.150	.160	3.56	3.81	4.06
Base to Seating Plane	A1	.015			0.38		
Shoulder to Shoulder Width	E	.595	.600	.625	15.11	15.24	15.88
Molded Package Width	E1	.530	.545	.560	13.46	13.84	14.22
Overall Length	D	2.045	2.058	2.065	51.94	52.26	52.45
Tip to Seating Plane	L	.120	.130	.135	3.05	3.30	3.43
Lead Thickness	c	.008	.012	.015	0.20	0.29	0.38
Upper Lead Width	B1	.030	.050	.070	0.76	1.27	1.78
Lower Lead Width	B	.014	.018	.022	0.36	0.46	0.56
Overall Row Spacing	§ eB	.620	.650	.680	15.75	16.51	17.27
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:

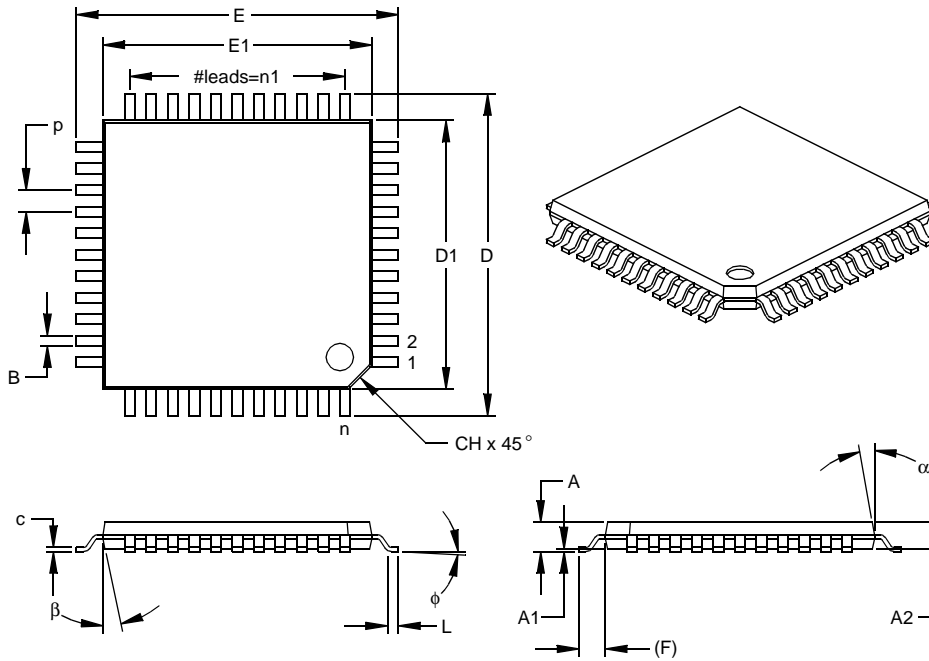
Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" (0.254mm) per side.

JEDEC Equivalent: MO-011

Drawing No. C04-016

# PIC16F7X7

## 44-Lead Plastic Thin Quad Flatpack (PT) 10x10x1 mm Body, 1.0/0.10 mm Lead Form (TQFP)



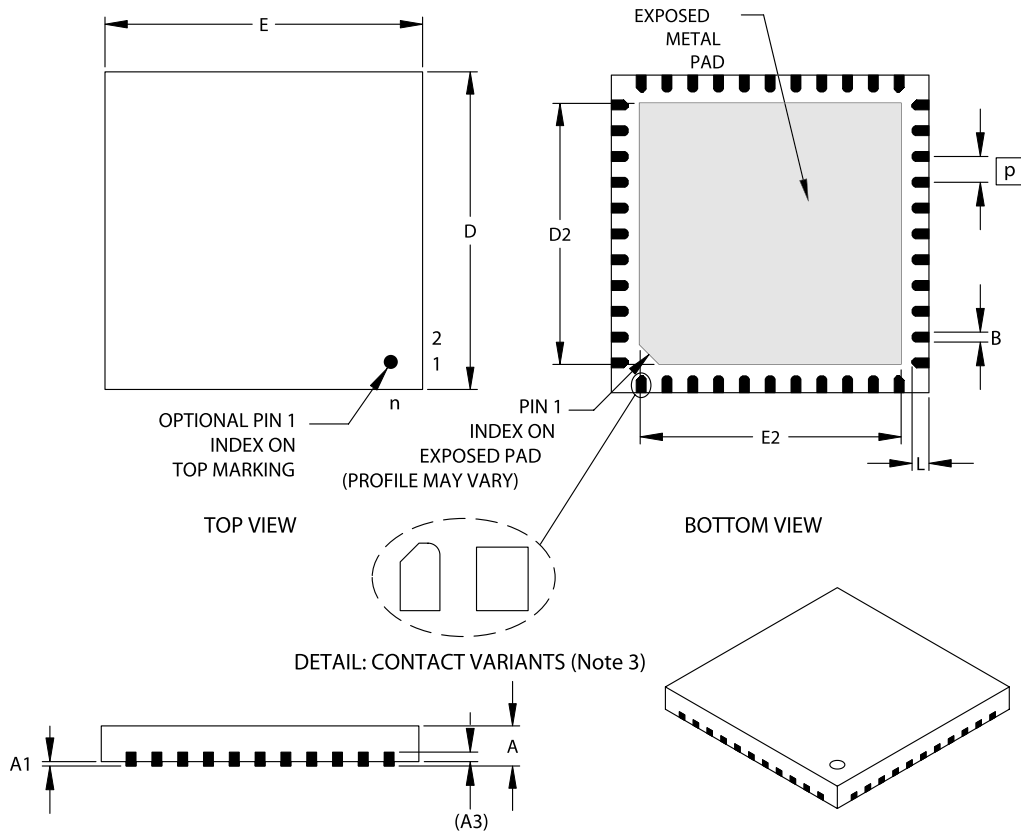
Dimension Limits	Units	INCHES			MILLIMETERS*		
		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n	44			44		
Pitch	p		.031			0.80	
Pins per Side	n1		11			11	
Overall Height	A	.039	.043	.047	1.00	1.10	1.20
Molded Package Thickness	A2	.037	.039	.041	0.95	1.00	1.05
Standoff §	A1	.002	.004	.006	0.05	0.10	0.15
Foot Length	L	.018	.024	.030	0.45	0.60	0.75
Footprint (Reference)	(F)		.039		1.00		
Foot Angle	φ	0	3.5	7	0	3.5	7
Overall Width	E	.463	.472	.482	11.75	12.00	12.25
Overall Length	D	.463	.472	.482	11.75	12.00	12.25
Molded Package Width	E1	.390	.394	.398	9.90	10.00	10.10
Molded Package Length	D1	.390	.394	.398	9.90	10.00	10.10
Lead Thickness	c	.004	.006	.008	0.09	0.15	0.20
Lead Width	B	.012	.015	.017	0.30	0.38	0.44
Pin 1 Corner Chamfer	CH	.025	.035	.045	0.64	0.89	1.14
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 D1 and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" (0.254mm) per side.  
 JEDEC Equivalent: MS-026  
 Drawing No. C04-076

## 44-Lead Plastic Quad Flat No Lead Package (ML) 8x8 mm Body (QFN)



Dimension Limits	Units	INCHES			MILLIMETERS*		
		MIN	NOM	MAX	MIN	NOM	MAX
Number of Contacts	n	44			44		
Pitch	p	.026 BSC <sup>1</sup>			0.65 BSC <sup>1</sup>		
Overall Height	A	.031	.035	.039	0.80	0.90	1.00
Standoff	A1	.000	.001	.002	0	0.02	0.05
Base Thickness	(A3)	.010 REF <sup>2</sup>			0.25 REF <sup>2</sup>		
Overall Width	E	.309	.315	.321	7.85	8.00	8.15
Exposed Pad Width	E2	.246	.268	.274	6.25	6.80	6.95
Overall Length	D	.309	.315	.321	7.85	8.00	8.15
Exposed Pad Length	D2	.246	.268	.274	6.25	6.80	6.95
Contact Width	B	.008	.013	.013	0.20	0.33	0.35
Contact Length	L	.014	.016	.019	0.35	0.40	0.48

\*Controlling Parameter

Notes:

1. BSC: Basic Dimension. Theoretically exact value shown without tolerances. See ASME Y14.5M
2. REF: Reference Dimension, usually without tolerance, for information purposes only. See ASME Y14.5M
3. Contact profiles may vary.

JEDEC equivalent: M0-220  
Drawing No. C04-103

# PIC16F7X7

---

NOTES:

## APPENDIX A: REVISION HISTORY

### Revision A (June 2003)

This is a new data sheet. However, these devices are similar to the PIC16C7X devices found in the PIC16C7X Data Sheet (DS30390) or the PIC16F87X devices (DS30292).

### Revision B (November 2003)

This revision includes updates to the Electrical Specifications in **Section 18.0 “Electrical Characteristics”** and minor corrections to the data sheet text.

### Revision C (October 2004)

This revision includes the DC and AC Characteristics Graphs and Tables. The Electrical Specifications in **Section 19.0 “DC and AC Characteristics Graphs and Tables”** have been updated and there have been minor corrections to the data sheet text.

## APPENDIX B: DEVICE DIFFERENCES

The differences between the devices in this data sheet are listed in Table B-1.

**TABLE B-1: DEVICE DIFFERENCES**

Difference	PIC16F737	PIC16F747	PIC16F767	PIC16F777
Flash Program Memory (14-bit words)	4K	4K	8K	8K
Data Memory (bytes)	368	368	368	368
I/O Ports	3	5	3	5
A/D	11 channels, 10 bits	14 channels, 10 bits	11 channels, 10 bits	14 channels, 10 bits
Parallel Slave Port	No	Yes	No	Yes
Interrupt Sources	16	17	16	17
Packages	28-pin PDIP 28-pin SOIC 28-pin SSOP 28-pin QFN	40-pin PDIP 44-pin QFN 44-pin TQFP	28-pin PDIP 28-pin SOIC 28-pin SSOP 28-pin QFN	40-pin PDIP 44-pin QFN 44-pin TQFP

# PIC16F7X7

---

## APPENDIX C: CONVERSION CONSIDERATIONS

Considerations for converting from previous versions of devices to the ones listed in this data sheet are listed in Table C-1.

**TABLE C-1: CONVERSION CONSIDERATIONS**

Characteristic	PIC16C7X	PIC16F87X	PIC16F7X7
Pins	28/40	28/40	28/40
Timers	3	3	3
Interrupts	11 or 12	13 or 14	16 or 17
Communication	PSP, USART, SSP (SPI™, I <sup>2</sup> C™ Master/Slave)	PSP, AUSART, MSSP (SPI, I <sup>2</sup> C Master/Slave)	PSP, AUSART, MSSP (SPI, I <sup>2</sup> C Master/Slave)
Frequency	20 MHz	20 MHz	20 MHz
A/D	8-bit	10-bit	10-bit
CCP	2	2	3
Program Memory	4K, 8K EPROM	4K, 8K Flash (1,000 E/W cycles)	4K, 8K Flash (100 E/W cycles)
RAM	192, 368 bytes	192, 368 bytes	368 bytes
EEPROM Data	None	128, 256 bytes	None
Other	—	In-Circuit Debugger, Low-Voltage Programming	In-Circuit Debugger



## INDEX

### A

A/D	
A/D Converter Interrupt, Configuring .....	155
Acquisition Requirements .....	156
ADRESH Register .....	154
Analog Port Pins .....	68
Analog-to-Digital Converter .....	151
Associated Registers .....	160
Automatic Acquisition Time .....	157
Calculating Acquisition Time .....	156
Configuring Analog Port Pins .....	158
Configuring the Module .....	155
Conversion Clock .....	157
Conversion Requirements .....	236
Conversion Status (GO/DONE Bit) .....	154
Conversions .....	159
Delays .....	156
Effects of a Reset .....	160
Internal Sampling Switch (Rss)	
Impedance .....	156
Operation During Sleep .....	160
Operation in Power-Managed Modes .....	158
Source Impedance .....	156
Time Delays .....	156
Use of the CCP Trigger .....	160
Absolute Maximum Ratings .....	207
ACKSTAT .....	123
ACKSTAT Status Flag .....	123
ADCON0 Register	
GO/DONE Bit .....	154
Addressable Universal Synchronous Asynchronous	
Receiver Transmitter. See AUSART	
ADRESL Register .....	154
Application Notes	
AN546 (Using the Analog-to-Digital (A/D)	
Converter) .....	151
AN552 (Implementing Wake-up	
on Key Stroke) .....	56
AN556 (Implementing a Table Read) .....	29
AN607 (Power-up Trouble Shooting) .....	173
Assembler	
MPASM Assembler .....	201
AUSART	
Address Detect Enable (ADDEN Bit) .....	134
Addressable Universal Synchronous	
Asynchronous Receiver Transmitter .....	133
Asynchronous	
Receiver	
(9-Bit Mode) .....	142
Asynchronous Mode .....	138
Receiver .....	140
Transmitter .....	138
Asynchronous Receive with Address Detect.	
See Asynchronous Receive (9-bit Mode).	
Asynchronous Reception	
Associated Registers .....	141, 143
Setup .....	141
Asynchronous Reception with Address Detect	
Setup .....	142

Asynchronous Transmission	
Associated Registers .....	139
Setup .....	139
Baud Rate Generator (BRG) .....	135
Associated Registers .....	135
Baud Rate Formula .....	135
Baud Rates, Asynchronous Mode	
(BRGH = 0) .....	136
Baud Rates, Asynchronous Mode	
(BRGH = 1) .....	136
High Baud Rate Select (BRGH Bit) .....	133
INTRC Baud Rates, Asynchronous Mode	
(BRGH = 0) .....	137
INTRC Baud Rates, Asynchronous Mode	
(BRGH = 1) .....	137
Sampling .....	135
Clock Source Select (CSRC Bit) .....	133
Continuous Receive Enable	
(CREN Bit) .....	134
Framing Error (FERR Bit) .....	134
Overrun Error (OERR Bit) .....	134
Receive Data, 9th Bit (RX9D Bit) .....	134
Receive Enable, 9-Bit (RX9 Bit) .....	134
Serial Port Enable (SPEN Bit) .....	133, 134
Single Receive Enable (SREN Bit) .....	134
Synchronous Master Mode .....	144
Reception .....	146
Transmission .....	144
Synchronous Master Reception	
Associated Registers .....	146
Setup .....	146
Synchronous Master Transmission	
Associated Registers .....	145
Setup .....	144
Synchronous Slave Mode .....	148
Reception .....	149
Transmit .....	148
Synchronous Slave Reception	
Associated Registers .....	149
Setup .....	149
Synchronous Slave Transmission	
Associated Registers .....	148
Setup .....	148
Transmit Data, 9th Bit (TX9D) .....	133
Transmit Enable (TXEN Bit) .....	133
Transmit Enable, 9-Bit (TX9 Bit) .....	133
Transmit Shift Register Status	
(TRMT Bit) .....	133

### B

Banking, Data Memory .....	15
Baud Rate Generator .....	119
BF .....	123
BF Status Flag .....	123

# PIC16F7X7

## Block Diagrams

A/D .....	155
Analog Input Model .....	156, 165
AUSART Receive .....	140, 142
AUSART Transmit .....	138
Baud Rate Generator .....	119
Capture Mode Operation .....	89
Comparator I/O Operating Modes .....	162
Comparator Output .....	164
Comparator Voltage Reference .....	168
Compare .....	89
Fail-Safe Clock Monitor .....	189
In-Circuit Serial Programming	
Connections .....	192
Interrupt Logic .....	184
Low-Voltage Detect (LVD) .....	175
Low-Voltage Detect (LVD) with	
External Input .....	175
Low-Voltage Detect Characteristics .....	221
MSSP (I <sup>2</sup> C Master Mode) .....	117
MSSP (I <sup>2</sup> C Mode) .....	102
MSSP (SPI Mode) .....	93
On-Chip Reset Circuit .....	172
OSC1/CLKI/RA7 Pin .....	54
OSC2/CLKO/RA6 Pin .....	53
PIC16F737 and PIC16F767 .....	6
PIC16F747 and PIC16F777 .....	7
PORTC (Peripheral Output Override)	
RC<2:0>, RC<7:5> Pins .....	65
PORTC (Peripheral Output Override)	
RC<4:3> Pins .....	65
PORTD (In I/O Port Mode) .....	67
PORTD and PORTE	
(Parallel Slave Port) .....	70
PORTE (In I/O Port Mode) .....	68
PWM Mode .....	91
RA0/AN0:RA1/AN1 Pins .....	50
RA2/AN2/VREF-/CVREF Pin .....	51
RA3/AN3/VREF+ Pin .....	50
RA4/T0CKI/C1OUT Pin .....	51
RA5/AN4/LVDIN/SS/C2OUT Pin .....	52
RB0/INT/AN12 Pin .....	57
RB1/AN10 Pin .....	57
RB2/AN8 Pin .....	58
RB3/CCP2/AN9 Pin .....	59
RB4/AN11 Pin .....	60
RB5/AN13/CCP3 Pin .....	61
RB6/PGC Pin .....	62
RB7/PGD Pin .....	63
Recommended MCLR Circuit .....	173
System Clock .....	39
Timer0/WDT Prescaler .....	73
Timer1 .....	79
Timer2 .....	85
Watchdog Timer (WDT) .....	186
BOR. See Brown-out Reset.	
BRG. See Baud Rate Generator.	
BRGH Bit .....	135
Brown-out Reset (BOR) .....	169, 172, 173, 179, 180

## C

C Compilers	
MPLAB C17 .....	202
MPLAB C18 .....	202
MPLAB C30 .....	202
Capture/Compare/PWM (CCP) .....	87
Capture Mode .....	89
CCP Pin Configuration .....	89
Prescaler .....	89
Compare Mode .....	89
CCP Pin Configuration .....	90
Software Interrupt Mode .....	90
Special Event Trigger .....	90
Special Event Trigger Output .....	90
Timer1 Mode Selection .....	90
Interaction of Two CCP Modules .....	87
PWM Mode .....	91
Duty Cycle .....	91
Example Frequencies and	
Resolutions .....	92
Period .....	91
Setup for Operation .....	92
Registers Associated with Capture,	
Compare and Timer1 .....	90
Registers Associated with PWM	
and Timer2 .....	92
Timer Resources .....	87
CCP1 Module .....	87
CCP2 Module .....	87
CCP3 Module .....	87
CCPR1H Register .....	87
CCPR1L Register .....	87
CCPR2H Register .....	87
CCPR2L Register .....	87
CCPR3H Register .....	87
CCPR3L Register .....	87
CCPxM<3:0> Bits .....	88
CCPxX and CCPxY Bits .....	88
Clock Sources .....	37
Selection Using OSCCON Register .....	37
Clock Switching .....	37
Modes (table) .....	47
Transition and the Watchdog Timer .....	38
Code Examples	
Call of a Subroutine in Page 1	
from Page 0 .....	29
Changing Between Capture Prescalers .....	89
Changing Prescaler Assignment from	
WDT to Timer0 .....	76
Flash Program Read .....	32
Implementing a Real-Time Clock Using a	
Timer1 Interrupt Service .....	82
Indirect Addressing .....	30
Initializing PORTA .....	49
Loading the SSPBUF (SSPSR) Register .....	96
Reading a 16-Bit Free Running Timer .....	80
Saving Status and W Registers in RAM .....	185
Writing a 16-Bit Free Running Timer .....	80
Code Protection .....	169, 192

Comparator Module .....	161	<b>I</b>	
Analog Input Connection		I/O Ports .....	49
Considerations .....	165	I <sup>2</sup> Mode	
Associated Registers .....	165	Operation .....	106
Configuration .....	162	I <sup>2</sup> Slave Mode	
Effects of a Reset .....	165	Clock Stretching, 10-bit Receive Mode	
Interrupts .....	164	(SEN = 1) .....	112
Operation .....	163	Clock Stretching, 10-bit Transmit Mode .....	112
Operation During Sleep .....	165	Clock Stretching, 7-bit Receive Mode	
Outputs .....	163	(SEN = 1) .....	112
Reference .....	163	Clock Stretching, 7-bit Transmit Mode .....	112
External Signal .....	163	I <sup>2</sup> C Master Mode .....	117
Internal Signal .....	163	Clock Arbitration .....	120
Response Time .....	163	Operation .....	118
Comparator Specifications .....	220	Reception .....	123
Comparator Voltage Reference .....	167	Repeated Start Condition Timing .....	122
Associated Registers .....	168	Start Condition Timing .....	121
Computed GOTO .....	29	Transmission .....	123
Configuration Bits .....	169	I <sup>2</sup> C Mode .....	102
Conversion Considerations .....	262	ACK Pulse .....	106, 107
Crystal and Ceramic Resonators .....	33	Acknowledge Sequence Timing .....	126
<b>D</b>		Baud Rate Generator .....	119
Data Memory .....	15	Bus Collision	
Bank Select (RP1:RP0 Bits) .....	15	Repeated Start Condition .....	130
General Purpose Registers .....	15	Start Condition .....	128
Map for PIC16F737 and		Stop Condition .....	131
PIC16F767 .....	16	Clock Synchronization and the CKP Bit .....	113
Map for PIC16F747 and		Effect of a Reset .....	127
PIC16F777 .....	17	General Call Address Support .....	116
Special Function Registers .....	18	Multi-Master Communication, Bus Collision	
DC and AC Characteristics		and Arbitration .....	127
Graphs and Tables .....	237	Multi-Master Mode .....	127
DC Characteristics .....	209, 218	Read/Write Bit Information (R/W Bit) .....	107
Internal RC Accuracy .....	217	Registers .....	102
Power-Down and Supply Current .....	210	Serial Clock (RC3/SCK/SCL) .....	107
Demonstration Boards		Sleep Operation .....	127
PICDEM 1 .....	204	Stop Condition Timing .....	126
PICDEM 17 .....	205	I <sup>2</sup> C Slave Mode .....	106
PICDEM 18R .....	205	Addressing .....	106
PICDEM 2 Plus .....	204	Clock Stretching .....	112
PICDEM 3 .....	204	Reception .....	107
PICDEM 4 .....	204	Transmission .....	107
PICDEM LIN .....	205	ID Locations .....	169, 192
PICDEM USB .....	205	In-Circuit Debugger .....	192
PICDEM.net Internet/Ethernet .....	204	In-Circuit Serial Programming .....	169
Development Support .....	201	In-Circuit Serial Programming (ICSP) .....	192
Device Differences .....	261	INDF Register .....	30
Device Overview .....	5	Indirect Addressing .....	30
Features .....	5	FSR Register .....	15
Direct Addressing .....	30	Instruction Set	
<b>E</b>		ADDLW .....	195
Electrical Characteristics .....	207	ADDWF .....	195
Errata .....	4	ANDLW .....	195
Evaluation and Programming Tools .....	205	ANDWF .....	195
External Clock Input .....	34	BCF .....	195
<b>F</b>		BSF .....	195
Fail-Safe Clock Monitor .....	169, 189	BTFSC .....	195
FSR Register .....	30	BTFSS .....	195
		CALL .....	196
		CLRf .....	196

# PIC16F7X7

CLRW .....	196	<b>L</b>	
CLRWDT .....	196	Load Conditions .....	222
COMF .....	196	Loading of PC .....	29
DECF .....	196	Low-Voltage Detect .....	174
DECFSZ .....	197	Characteristics .....	221
Firmware Instructions .....	193	Effects of a Reset .....	178
General Format .....	193	Operation .....	177
GOTO .....	197	Current Consumption .....	178
INCF .....	197	Reference Voltage Set Point .....	178
INCFSZ .....	197	Operation During Sleep .....	178
IORLW .....	197	Time-out Sequence .....	178
IORWF .....	197	Low-Voltage Detect (LVD) .....	169
MOVF .....	198	LVD. See Low-Voltage Detect.	
MOVLW .....	198	<b>M</b>	
MOVWF .....	198	Master Clear ( $\overline{\text{MCLR}}$ )	
NOP .....	198	$\overline{\text{MCLR}}$ Reset, Normal Operation .....	172, 179, 180
Opcode Field Descriptions .....	193	$\overline{\text{MCLR}}$ Reset, Sleep .....	172, 179, 180
Read-Modify-Write Operations .....	193	Master Synchronous Serial Port (MSSP).	
RETFIE .....	198	See MSSP.	
RETLW .....	198	Master Synchronous Serial Port. See MSSP	
RETURN .....	199	$\overline{\text{MCLR}}/\text{VPP}/\text{RE3}$ Pin .....	8
RLF .....	199	$\overline{\text{MCLR}}/\text{VPP}/\text{RE3}$ Pin .....	11
RRF .....	199	Memory Organization .....	15
SLEEP .....	199	Data Memory .....	15
SUBLW .....	199	Program Memory .....	15
SUBWF .....	199	Program Memory and Stack Maps .....	15
SWAPF .....	200	MPLAB ASM30 Assembler, Linker, Librarian .....	202
XORLW .....	200	MPLAB ICD 2 In-Circuit Debugger .....	203
XORWF .....	200	MPLAB ICE 2000 High-Performance	
Summary Table .....	194	Universal In-Circuit Emulator .....	203
INT Interrupt (RB0/INT). See Interrupt Sources.		MPLAB ICE 4000 High-Performance	
INTCON Register		Universal In-Circuit Emulator .....	203
GIE Bit .....	23	MPLAB Integrated Development	
INT0IE Bit .....	23	Environment Software .....	201
INT0IF Bit .....	23	MPLAB PM3 Device Programmer .....	203
PEIE Bit .....	23	MPLINK Object Linker/MPLIB Object Librarian .....	202
RBIF Bit .....	23, 56	MSSP .....	93
TMR0IE Bit .....	23	I <sup>2</sup> C Mode. See I <sup>2</sup> C.	
Inter-Integrated Circuit. See I <sup>2</sup> C.		SPI Mode. See SPI.	
Internal Oscillator Block .....	35	MSSP Module	
INTRC Modes .....	36	Control Registers (General) .....	93
Interrupt Sources .....	169, 184	Overview .....	93
A/D Conversion Complete .....	155	Multi-Master Mode .....	127
Interrupt-on-Change (RB7:RB4) .....	56	<b>O</b>	
RB0/INT Pin, External .....	185	OPTION_REG Register	
TMR0 Overflow .....	185	INTEDG Bit .....	22
Interrupts		PS2:PS0 Bits .....	22
Exiting Sleep with .....	48	PSA Bit .....	22
Synchronous Serial Port Interrupt .....	25	RBPJ Bit .....	22
Interrupts, Context Saving During .....	185	T0CS Bit .....	22
Interrupts, Enable Bits		T0SE Bit .....	22
Global Interrupt Enable (GIE Bit) .....	23, 184	OSC1/CLKI/RA7 Pin .....	8, 11
Interrupt-on-Change (RB7:RB4)		OSC2/CLKO/RA6 Pin .....	8, 11
Enable (RBIE Bit) .....	185	Oscillator Configuration .....	33
Peripheral Interrupt Enable (PEIE Bit) .....	23	ECIO .....	33
RB0/INT Enable (INT0IE Bit) .....	23	EXTRC .....	179
TMR0 Overflow Enable (TMR0IE Bit) .....	23	HS .....	33, 179
Interrupts, Flag Bits		INTIO1 .....	33
Interrupt-on Change (RB7:RB4)		INTIO2 .....	33
Flag (RBIF Bit) .....	23	INTRC .....	179
Interrupt-on-Change (RB7:RB4)		LP .....	33, 179
Flag (RBIF Bit) .....	23, 56, 185	RC .....	33, 35
RB0/INT Flag (INT0IF Bit) .....	23	RCIO .....	33
TMR0 Overflow Flag (TMR0IF Bit) .....	185	XT .....	33, 179
INTRC Modes			
Adjustment .....	36		

Oscillator Control Register		PORTD Register.....	67
Modifying IRCF Bits .....	39	PORTE .....	14
Clock Transition Sequence.....	40	Analog Port Pins.....	68
Oscillator Delay upon Power-up, Wake-up		Associated Registers.....	68
and Clock Switching.....	40	Input Buffer Full Status (IBF Bit).....	69
Oscillator Start-up Timer (OST) .....	169, 173	Input Buffer Overflow (IBOV Bit).....	69
Oscillator Switching.....	37	PORTE Register.....	68
<b>P</b>		PSP Mode Select (PSPMODE Bit).....	67, 68
Packaging .....	251	RE0/RD/AN5 Pin .....	68
Details .....	253	RE1/WR/AN6 Pin .....	68
Marking Information .....	251	RE2/CS/AN7 Pin .....	68
Paging, Program Memory .....	29	TRISE Register.....	68
Parallel Slave Port		PORTE Register.....	68
Associated Registers .....	71	Postscaler, WDT	
Parallel Slave Port (PSP).....	67, 70	Assignment (PSA Bit) .....	22
RE0/RD/AN5 Pin.....	68	Rate Select (PS2:PS0 Bits).....	22
RE1/WR/AN6 Pin.....	68	Power-Down Mode (Sleep).....	190
RE2/CS/AN7 Pin.....	68	Power-Down Mode. <i>See</i> Sleep.	
Select (PSPMODE Bit) .....	67, 68	Power-Managed Modes.....	41
PCL Register.....	29	RC_RUN.....	41
PCLATH Register .....	29	SEC_RUN .....	42
PCON Register .....	178	SEC_RUN/RC_RUN to	
POR Bit.....	28	Primary Clock Source.....	43
Peripheral Interrupt (PEIE Bit) .....	23	Power-on Reset (POR).....	169, 172, 173, 179, 180
PICkit 1 Flash Starter Kit.....	205	POR Status (POR Bit) .....	28
PICSTART Plus Development Programmer .....	204	Power Control/Status	
Pinout Descriptions		(PCON) Register .....	178
PIC16F737/PIC16F767.....	8–10	Power-Down (PD Bit) .....	172
PIC16F747/PIC16F777.....	11–14	Time-out (TO Bit).....	21, 172
PMADR Register.....	31	Power-up Timer (PWRT) .....	169, 173
POP .....	29	PR2 Register .....	85
POR. <i>See</i> Power-on Reset.		Prescaler, Timer0	
PORTA.....	8, 11	Assignment (PSA Bit) .....	22
Associated Registers .....	55	Rate Select (PS2:PS0 Bits).....	22
PORTA Register .....	49	PRO MATE II Universal Device	
TRISA Register .....	49	Programmer.....	203
PORTA Register .....	49	Program Counter	
PORTB.....	9, 12	Reset Conditions .....	179
Associated Registers .....	64	Program Memory	
PORTB Register .....	56	Flash	
Pull-up Enable (RBPU Bit) .....	22	Associated Registers.....	32
RB0/INT Edge Select (INTEDG Bit).....	22	Interrupt Vector.....	15
RB0/INT Pin, External.....	185	Memory and Stack Maps .....	15
RB7:RB4 Interrupt-on-Change.....	185	Operation During Code-Protect.....	32
RB7:RB4 Interrupt-on-Change		Organization .....	15
Enable (RBIE Bit).....	185	Paging .....	29
RB7:RB4 Interrupt-on-Change		PMADR Register .....	31
Flag (RBIF Bit) .....	23, 56, 185	PMADRH Register.....	31
TRISB Register .....	56	Reading .....	31
PORTB Register .....	56	Reading Flash .....	32
PORTC .....	10, 13	Reading, PMADR Register.....	31
Associated Registers .....	66	Reading, PMADRH Register .....	31
PORTC Register.....	65	Reading, PMCON1 Register .....	31
RC3/SCK/SCL Pin .....	107	Reading, PMDATA Register.....	31
RC6/TX/CK Pin.....	134	Reading, PMDATH Register.....	31
RC7/RX/DT Pin.....	134, 135	Reset Vector.....	15
TRISC Register.....	65, 133	Program Verification .....	192
PORTC Register .....	65	Programming, Device Instructions.....	193
PORTD .....	14	PUSH.....	29
Associated Registers .....	67		
Parallel Slave Port (PSP) Function.....	67		
PORTD Register.....	67		
TRISD Register.....	67		

# PIC16F7X7

## R

RA0/AN0 Pin .....	8, 11
RA1/AN1 Pin .....	8, 11
RA2/AN2/VREF-/CVREF Pin .....	8, 11
RA3/AN3/VREF+ Pin .....	8, 11
RA4/T0CKI/C1OUT Pin .....	8, 11
RA5/AN4/LVDIN/SS/C2OUT Pin .....	8, 11
RAM. See Data Memory.	
RB0/INT/AN12 Pin .....	9, 12
RB1/AN10 Pin .....	9, 12
RB2/AN8 Pin .....	9, 12
RB3/CCP2/AN9 Pin .....	9, 12
RB4/AN11 Pin .....	9, 12
RB5/AN13/CCP3 Pin .....	9, 12
RB6/PGC Pin .....	9, 12
RB7/PGD Pin .....	9, 12
RC0/T1OSO/T1CKI Pin .....	10, 13
RC1/T1OSI/CCP2 Pin .....	10, 13
RC2/CCP1 Pin .....	10, 13
RC3/SCK/SCL Pin .....	10, 13
RC4/SDI/SDA Pin .....	10, 13
RC5/SDO Pin .....	10, 13
RC6/TX/CK Pin .....	10, 13
RC7/RX/DT Pin .....	10, 13
RCIO Oscillator .....	35
RCSTA Register	
ADDEN Bit .....	134
CREN Bit .....	134
FERR Bit .....	134
OERR Bit .....	134
RX9 Bit .....	134
RX9D Bit .....	134
SPEN Bit .....	133, 134
SREN Bit .....	134
RD0/PSP0 Pin .....	14
RD1/PSP1 Pin .....	14
RD2/PSP2 Pin .....	14
RD3/PSP3 Pin .....	14
RD4/PSP4 Pin .....	14
RD5/PSP5 Pin .....	14
RD6/PSP6 Pin .....	14
RD7/PSP7 Pin .....	14
RE0/RD/AN5 Pin .....	14
RE1/WR/AN6 Pin .....	14
RE2/CS/AN7 Pin .....	14
Register File .....	15
Registers	
ADCON0 (A/D Control 0) .....	152
ADCON1 (A/D Control 1) .....	153
ADCON2 (A/D Control 2) .....	154
CCPxCON (CCPx Control) .....	88
CMCON (Comparator Control) .....	161
CVRCON (Comparator Voltage Reference Control) .....	167
Initialization Conditions (table) .....	180–181
INTCON (Interrupt Control) .....	23
LVDCON (Low-Voltage Detect Control) .....	176
OPTION_REG (Option Control) .....	22, 75
OSCCON (Oscillator Control) .....	38
OSCTUNE (Oscillator Tuning) .....	36
PCON (Power Control/Status) .....	28
PIE1 (Peripheral Interrupt Enable 1) .....	24
PIE2 (Peripheral Interrupt Enable 2) .....	26
PIR1 (Peripheral Interrupt Request (Flag) 1) .....	25
PIR2 (Peripheral Interrupt Request (Flag) 2) .....	27
PMCON1 (Program Memory Control 1) .....	31
RCSTA (Receive Status and Control) .....	134
Special Function, Summary .....	18–20
SSPCON (MSSP Control Register 1, I <sup>2</sup> C Mode) .....	104
SSPCON (MSSP Control Register 1, SPI Mode) .....	95
SSPCON2 (MSSP Control Register 2, I <sup>2</sup> C Mode) .....	105
SSPSTAT (MSSP Status, I <sup>2</sup> C Mode) .....	103
SSPSTAT (MSSP Status, SPI Mode) .....	94
Status .....	21
T1CON (Timer1 Control) .....	78
T2CON (Timer2 Control) .....	86
TRISE .....	69
TXSTA (Transmit Status and Control) .....	133
WDTCON (Watchdog Timer Control) .....	187
Reset .....	169, 172
Brown-out Reset (BOR). See Brown-out Reset (BOR).	
MCLR Reset. See MCLR.	
Power-on Reset (POR). See Power-on Reset (POR).	
Reset Conditions for All Registers .....	180, 181
Reset Conditions for PCON Register .....	179
Reset Conditions for Program Counter .....	179
Reset Conditions for Status Register .....	179
WDT Reset. See Watchdog Timer (WDT).	
Revision History .....	261

## S

SCI. See AUSART.	
SCK .....	93
SDI .....	93
SDO .....	93
Serial Clock, SCK .....	93
Serial Communication Interface. See AUSART.	
Serial Data In, SDI .....	93
Serial Data Out, SDO .....	93
Serial Peripheral Interface. See SPI.	
Slave Select, SS .....	93
Sleep .....	169, 172, 190
Software Simulator (MPLAB SIM) .....	202
Software Simulator (MPLAB SIM30) .....	202
Special Features of the CPU .....	169
Special Function Registers .....	18, 18–20
SPI Master Mode .....	98
SPI Mode .....	93
Associated Registers .....	101
Bus Mode Compatibility .....	101
Clock .....	98
Effects of a Reset .....	101
Enabling SPI I/O .....	97
Master/Slave Connection .....	97
Serial Clock .....	93
Serial Data In .....	93
Serial Data Out .....	93
Slave Select .....	93
Slave Select Synchronization .....	99
Sleep Operation .....	101
Typical Connection .....	97
SPI Slave Mode .....	99

$\overline{SS}$ .....	93	Timing Diagrams .....	
SSPBUF .....	98	A/D Conversion .....	236
SSPIF Bit .....	25	Acknowledge Sequence .....	126
SSPOV .....	123	Asynchronous Master Transmission .....	139
SSPOV Status Flag .....	123	Asynchronous Master Transmission	
SSPSR .....	98	(Back to Back) .....	139
SSPSTAT Register		Asynchronous Reception .....	140
R/W Bit .....	107	Asynchronous Reception with	
Stack .....	29	Address Byte First .....	143
Overflows .....	29	Asynchronous Reception with	
Underflows .....	29	Address Detect .....	143
Status Register		AUSART Synchronous Receive	
C Bit .....	21	(Master/Slave) .....	234
DC Bit .....	21	AUSART Synchronous Transmission	
IRP Bit .....	21	(Master/Slave) .....	234
PD Bit .....	21, 172	Baud Rate Generator with Clock Arbitration .....	120
TO Bit .....	21, 172	BRG Reset Due to SDA Arbitration	
Z Bit .....	21	During Start Condition .....	129
Synchronous Serial Port		Brown-out Reset .....	225
Interrupt Flag Bit (SSPIF) .....	25	Bus Collision During a Repeated	
<b>T</b>		Start Condition (Case 1) .....	130
T1CKPS0 Bit .....	78	Bus Collision During a Repeated	
T1CKPS1 Bit .....	78	Start Condition (Case 2) .....	130
T1OSCEN Bit .....	78	Bus Collision During a Stop Condition	
T1SYNC Bit .....	78	(Case 1) .....	131
T2CKPS0 Bit .....	86	Bus Collision During a Stop Condition	
T2CKPS1 Bit .....	86	(Case 2) .....	131
TAD .....	157	Bus Collision During Start Condition	
Timer0 .....	73	(SCL = 0) .....	129
Associated Registers .....	76	Bus Collision During Start Condition	
Clock Source Edge Select (T0SE Bit) .....	22	(SDA Only) .....	128
Clock Source Select (T0CS Bit) .....	22	Bus Collision for Transmit and	
Interrupt .....	73	Acknowledge .....	127
Operation .....	73	Capture/Compare/PWM	
Overflow Enable (TMR0IE Bit) .....	23	(CCP1 and CCP2) .....	227
Overflow Flag (TMR0IF Bit) .....	185	CLKO and I/O .....	224
Overflow Interrupt .....	185	Clock Synchronization .....	113
Prescaler .....	74	External Clock .....	223
T0CKI .....	74	Fail-Safe Clock Monitor .....	189
Use with External Clock .....	74	First Start Bit .....	121
Timer1 .....	77	I <sup>2</sup> C Bus Data .....	232
Associated Registers .....	83	I <sup>2</sup> C Bus Start/Stop Bits .....	231
Asynchronous Counter Mode .....	80	I <sup>2</sup> C Master Mode (Reception,	
Reading and Writing .....	80	7-bit Address) .....	125
Capacitor Selection .....	81	I <sup>2</sup> C Master Mode (Transmission,	
Counter Operation .....	79	7 or 10-bit Address) .....	124
Operation .....	77	I <sup>2</sup> C Slave Mode (Transmission,	
Operation in Synchronized		10-bit Address) .....	111
Counter Mode .....	79	I <sup>2</sup> C Slave Mode (Transmission,	
Operation in Timer Mode .....	79	7-bit Address) .....	109
Oscillator .....	81	I <sup>2</sup> C Slave Mode with SEN = 0 (Reception,	
Oscillator Layout Considerations .....	81	10-bit Address) .....	110
Prescaler .....	82	I <sup>2</sup> C Slave Mode with SEN = 0 (Reception,	
Resetting Timer1 Register Pair .....	82	7-bit Address) .....	108
Resetting Using a CCP Trigger Output .....	81	I <sup>2</sup> C Slave Mode with SEN = 1 (Reception,	
Use as a Real-Time Clock .....	82	10-bit Address) .....	115
Timer2 .....	85	I <sup>2</sup> C Slave Mode with SEN = 1 (Reception,	
Associated Registers .....	86	7-bit Address) .....	114
Output .....	85	Low-Voltage Detect .....	177
Postscaler .....	85	LP Clock to Primary System Clock after	
Prescaler .....	85	Reset (EC, RC, INTRC) .....	46
Prescaler and Postscaler .....	85	LP Clock to Primary System Clock after	
		Reset (HS, XT, LP) .....	45
		Parallel Slave Port .....	228
		Parallel Slave Port Read .....	71





## ON-LINE SUPPORT

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

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

### Connecting to the Microchip Internet Web Site

The Microchip web site is available at the following URL:

**[www.microchip.com](http://www.microchip.com)**

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

**<ftp://ftp.microchip.com>**

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

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

## SYSTEMS INFORMATION AND UPGRADE HOT LINE

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

1-800-755-2345 for U.S. and most of Canada, and

1-480-792-7302 for the rest of the world.

042003

# PIC16F7X7

---

---

## READER RESPONSE

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

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

To: Technical Publications Manager  
RE: Reader Response  
From: Name \_\_\_\_\_  
Company \_\_\_\_\_  
Address \_\_\_\_\_  
City / State / ZIP / Country \_\_\_\_\_  
Telephone: (\_\_\_\_\_) \_\_\_\_\_ - \_\_\_\_\_ FAX: (\_\_\_\_\_) \_\_\_\_\_ - \_\_\_\_\_

Application (optional):

Would you like a reply? \_\_\_Y \_\_\_N

Device: PIC16F7X7

Literature Number: DS30498C

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?

---

---

## PIC16F7X7 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	PIC16F7X7 <sup>(1)</sup> , PIC16F7X7T <sup>(1)</sup> ; VDD range 4.0V to 5.5V PIC16LF7X7 <sup>(1)</sup> , PIC16LF7X7T <sup>(1)</sup> ; VDD range 2.0V to 5.5V		
Temperature Range	I = -40°C to +85°C (Industrial) E = -40°C to +125°C (Extended)		
Package	ML = QFN (Micro Lead Frame) PT = TQFP (Thin Quad Flatpack) SO = SOIC SP = Skinny Plastic DIP P = PDIP SS = SSOP		
Pattern	QTP, SQTP, Code or Special Requirements (blank otherwise)		

**Examples:**

- a) PIC16F777-I/P 301 = Industrial temp., PDIP package, normal VDD limits, QTP pattern #301.
- b) PIC16LF767-I/SO = Industrial temp., SOIC package, extended VDD limits.
- c) PIC16F747-E/P = Extended temp., PDIP package, normal VDD limits.

**Note 1:** F = CMOS Flash  
LF = Low-Power CMOS Flash

**2:** T = in tape and reel – SOIC, SSOP, TQFP 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

10/20/04