
Note the following details of the code protection feature on PICmicro® MCUs.

- The PICmicro family meets the specifications contained in the Microchip Data Sheet.
- Microchip believes that its family of PICmicro microcontrollers is one of the most secure products 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 PICmicro microcontroller in a manner outside the operating specifications contained in the data sheet. The person doing so may be 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 product.

If you have any further questions about this matter, please contact the local sales office nearest to you.

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contained in this publication regarding device applications and the like is intended through suggestion only and may be superseded by updates.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

Trademarks

The Microchip name and logo, the Microchip logo, FilterLab, KEELOQ, MPLAB, PIC, PICmicro, PICMASTER, PICSTART, PRO MATE, SEEVAL and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

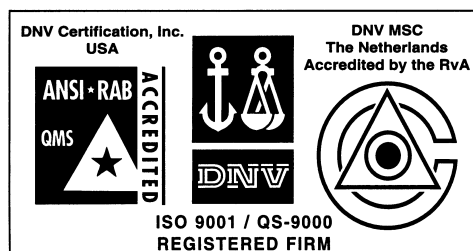
dsPIC, ECONOMONITOR, FanSense, FlexROM, fuzzyLAB, In-Circuit Serial Programming, ICSP, ICEPIC, microID, microPort, Migratable Memory, MPASM, MPLIB, MPLINK, MPSIM, MXDEV, PICC, PICDEM, PICDEM.net, rfPIC, Select Mode and Total Endurance are trademarks of Microchip Technology Incorporated in the U.S.A.

Serialized Quick Term Programming (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.

© 2002, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

 Printed on recycled paper.



Microchip received QS-9000 quality system certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona in July 1999. The Company's quality system processes and procedures are QS-9000 compliant for its PICmicro® 8-bit MCUs, KEELOQ® code hopping devices, Serial EEPROMs and microperipheral products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001 certified.

28/40-Pin 8-Bit CMOS FLASH Microcontrollers

Devices Included in this Data Sheet:

- PIC16F73 • PIC16F76
- PIC16F74 • PIC16F77

High Performance RISC CPU:

- High performance RISC CPU
- Only 35 single word instructions to learn
- All single cycle instructions except for program branches which are two-cycle
- Operating speed: DC - 20 MHz clock input
DC - 200 ns instruction cycle
- Up to 8K x 14 words of FLASH Program Memory, Up to 368 x 8 bytes of Data Memory (RAM)
- Pinout compatible to the PIC16C73B/74B/76/77
- Pinout compatible to the PIC16F873/874/876/877
- Interrupt capability (up to 12 sources)
- Eight level deep hardware stack
- Direct, Indirect and Relative Addressing modes
- Processor read access to program memory

Special Microcontroller Features:

- Power-on Reset (POR)
- Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Programmable code protection
- Power saving SLEEP mode
- Selectable oscillator options
- In-Circuit Serial Programming™ (ICSP™) via two pins

Peripheral Features:

- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler, can be incremented during SLEEP via external crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period register, prescaler and postscaler
- Two 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-bit
- 8-bit, up to 8-channel Analog-to-Digital converter
- Synchronous Serial Port (SSP) with SPI™ (Master mode) and I²C™ (Slave)
- Universal Synchronous Asynchronous Receiver Transmitter (USART/SCI)
- Parallel Slave Port (PSP), 8-bits wide with external $\overline{RD}$, $\overline{WR}$ and $\overline{CS}$ controls (40/44-pin only)
- Brown-out detection circuitry for Brown-out Reset (BOR)

CMOS Technology:

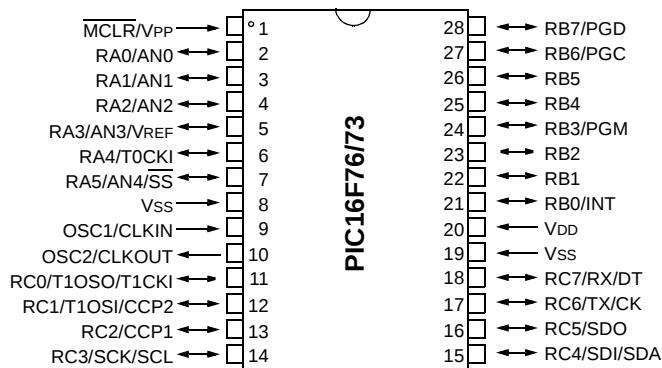
- Low power, high speed CMOS FLASH technology
- Fully static design
- Wide operating voltage range: 2.0V to 5.5V
- High Sink/Source Current: 25 mA
- Industrial temperature range
- Low power consumption:
 - < 2 mA typical @ 5V, 4 MHz
 - 20 μ A typical @ 3V, 32 kHz
 - < 1 μ A typical standby current

Device	Program Memory (# Single Word Instructions)	Data SRAM (Bytes)	I/O	Interrupts	8-bit A/D (ch)	CCP (PWM)	SSP		USART	Timers 8/16-bit
							SPI (Master)	I²C (Slave)		
PIC16F73	4096	192	22	11	5	2	Yes	Yes	Yes	2 / 1
PIC16F74	4096	192	33	12	8	2	Yes	Yes	Yes	2 / 1
PIC16F76	8192	368	22	11	5	2	Yes	Yes	Yes	2 / 1
PIC16F77	8192	368	33	12	8	2	Yes	Yes	Yes	2 / 1

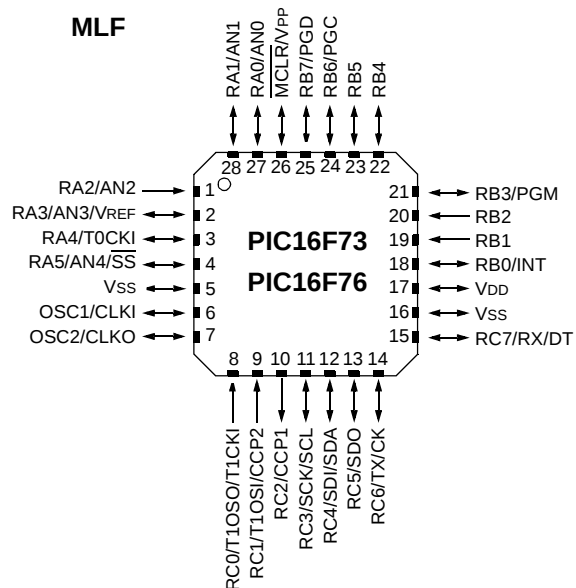
PIC16F7X

Pin Diagrams

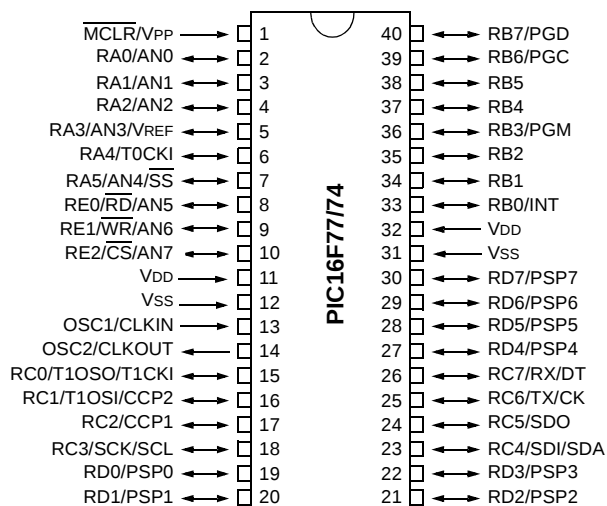
DIP, SOIC, SSOP



MLF

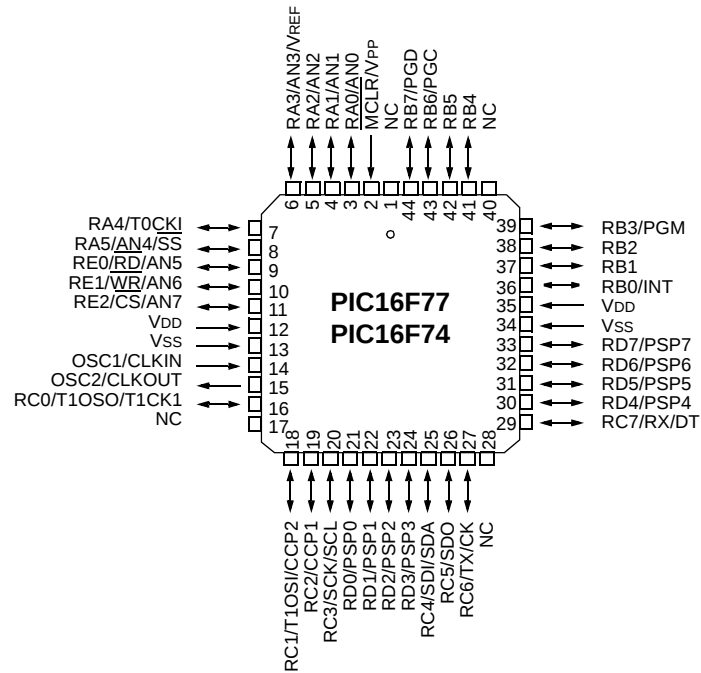


PDIP

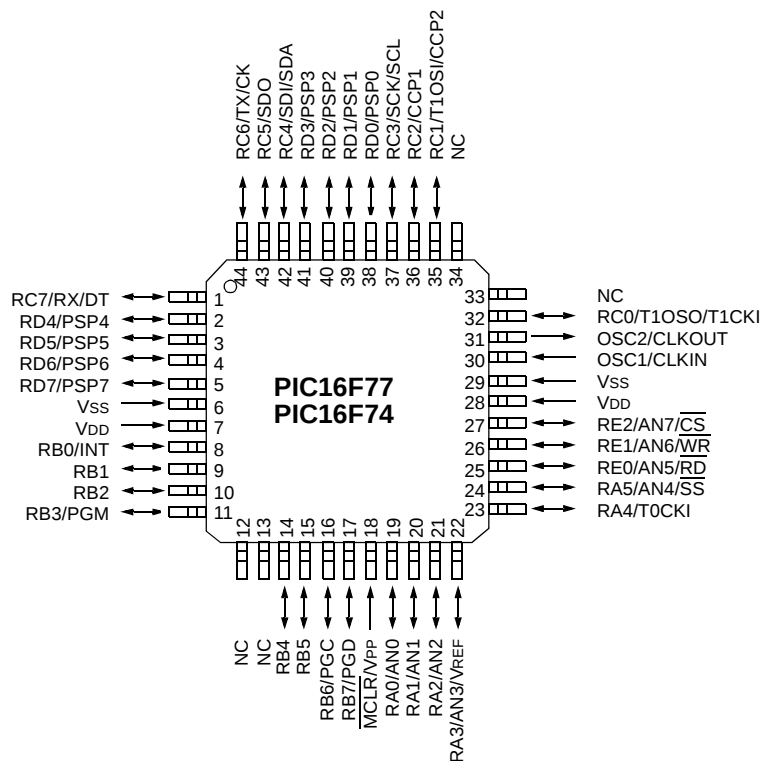


Pin Diagrams (Continued)

PLCC



QFP



PIC16F7X

Table of Contents

1.0	Device Overview	5
2.0	Memory Organization.....	13
3.0	Reading Program Memory.....	29
4.0	I/O Ports.....	31
5.0	Timer0 Module.....	43
6.0	Timer1 Module.....	47
7.0	Timer2 Module.....	51
8.0	Capture/Compare/PWM Modules.....	53
9.0	Synchronous Serial Port (SSP) Module.....	59
10.0	Universal Synchronous Asynchronous Receiver Transmitter (USART)	69
11.0	Analog-to-Digital Converter (A/D) Module	83
12.0	Special Features of the CPU	89
13.0	Instruction Set Summary.....	105
14.0	Development Support	113
15.0	Electrical Characteristics.....	119
16.0	DC and AC Characteristics Graphs and Tables	141
17.0	Packaging Information	151
Appendix A:	Revision History	161
Appendix B:	Device Differences	161
Appendix C:	Conversion Considerations	162
Index		163
On-Line Support.....		169
Reader Response		170
PIC16F7X Product Identification System		171

TO OUR VALUED CUSTOMERS

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

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

Most Current Data Sheet

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

<http://www.microchip.com>

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

Errata

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

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

- Microchip's Worldwide Web site; <http://www.microchip.com>
- Your local Microchip sales office (see last page)
- The Microchip Corporate Literature Center; U.S. FAX: (480) 792-7277

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

Customer Notification System

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

1.0 DEVICE OVERVIEW

This document contains device specific information about the following devices:

- PIC16F73
- PIC16F74
- PIC16F76
- PIC16F77

PIC16F73/76 devices are available only in 28-pin packages, while PIC16F74/77 devices are available in 40-pin and 44-pin packages. All devices in the PIC16F7X family share common architecture, with the following differences:

- The PIC16F73 and PIC16F76 have one-half of the total on-chip memory of the PIC16F74 and PIC16F77
- The 28-pin devices have 3 I/O ports, while the 40/44-pin devices have 5
- The 28-pin devices have 11 interrupts, while the 40/44-pin devices have 12
- The 28-pin devices have 5 A/D input channels, while the 40/44-pin devices have 8
- The Parallel Slave Port is implemented only on the 40/44-pin devices

The available features are summarized in Table 1-1. Block diagrams of the PIC16F73/76 and PIC16F74/77 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 Reference Manual (DS33023), which may be obtained from your local Microchip Sales Representative or downloaded from the Microchip website. 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: PIC16F7X DEVICE FEATURES

Key Features	PIC16F73	PIC16F74	PIC16F76	PIC16F77
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)	192	192	368	368
Interrupts	11	12	11	12
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	2	2	2	2
Serial Communications	SSP, USART	SSP, USART	SSP, USART	SSP, USART
Parallel Communications	—	PSP	—	PSP
8-bit Analog-to-Digital Module	5 Input Channels	8 Input Channels	5 Input Channels	8 Input Channels
Instruction Set	35 Instructions	35 Instructions	35 Instructions	35 Instructions
Packaging	28-pin DIP 28-pin SOIC 28-pin SSOP 28-pin MLF	40-pin PDIP 44-pin PLCC 44-pin TQFP	28-pin DIP 28-pin SOIC 28-pin SSOP 28-pin MLF	40-pin PDIP 44-pin PLCC 44-pin TQFP

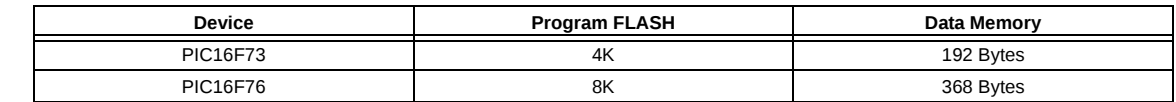
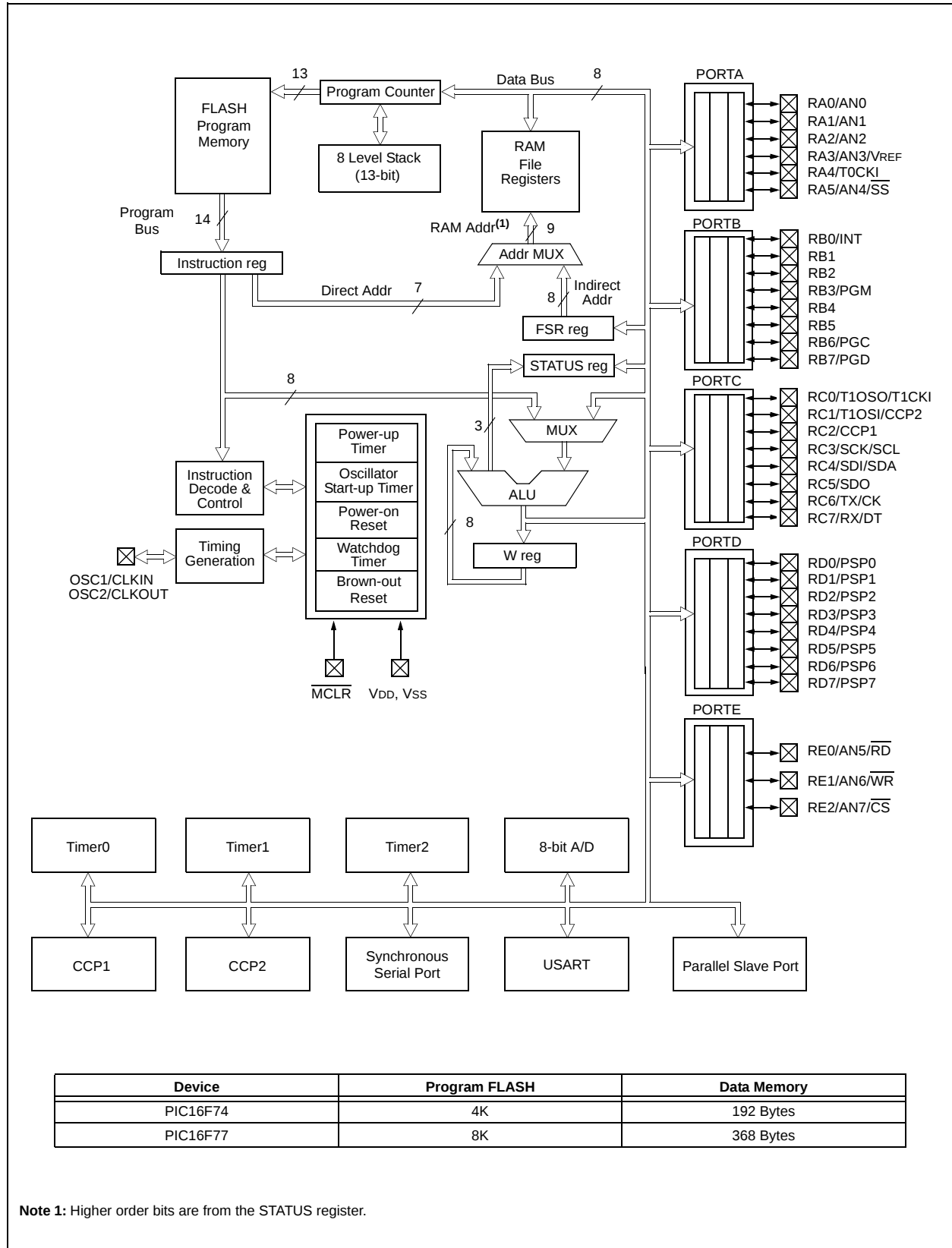


FIGURE 1-2: PIC16F74 AND PIC16F77 BLOCK DIAGRAM



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.			

TABLE 1-2: PIC16F73 AND PIC16F76 PINOUT DESCRIPTION (CONTINUED)

Pin Name	DIP SSOP SOIC Pin#	MLF Pin#	I/O/P Type	Buffer Type	Description
RB0/INT RB0 INT	21	18	I/O I	TTL/ST ⁽¹⁾	PORTB is a bi-directional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs. Digital I/O. External interrupt.
RB1	22	19	I/O	TTL	Digital I/O.
RB2	23	20	I/O	TTL	Digital I/O.
RB3/PGM RB3 PGM	24	21	I/O I/O	TTL	Digital I/O. Low voltage ICSP programming enable pin.
RB4	25	22	I/O	TTL	Digital I/O.
RB5	26	23	I/O	TTL	Digital I/O.
RB6/PGC RB6 PGC	27	24	I/O I/O	TTL/ST ⁽²⁾	Digital I/O. In-Circuit Debugger and ICSP programming clock.
RB7/PGD RB7 PGD	28	25	I/O I/O	TTL/ST ⁽²⁾	Digital I/O. In-Circuit Debugger and ICSP programming data.
RC0/T1OSO/T1CKI RC0 T1OSO T1CKI	11	8	I/O O I	ST	PORTC is a bi-directional I/O port. Digital I/O. Timer1 oscillator output. Timer1 external clock input.
RC1/T1OSI/CCP2 RC1 T1OSI CCP2	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 ² C mode.
RC4/SDI/SDA RC4 SDI SDA	15	12	I/O I I/O	ST	Digital I/O. SPI data in. I ² 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. USART asynchronous transmit. USART 1 synchronous clock.
RC7/RX/DT RC7 RX DT	18	15	I/O I I/O	ST	Digital I/O. USART asynchronous receive. USART synchronous data.
Vss	8, 19	5, 16	P	—	Ground reference for logic and I/O pins.
VDD	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.

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 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.			

TABLE 1-3: PIC16F74 AND PIC16F77 PINOUT DESCRIPTION (CONTINUED)

Pin Name	DIP Pin#	PLCC Pin#	QFP Pin#	I/O/P Type	Buffer Type	Description
RB0/INT RB0 INT	33	36	8	I/O I	TTL/ST ⁽⁴⁾	PORTB is a bi-directional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs. Digital I/O. External interrupt.
RB1	34	37	9	I/O	TTL	Digital I/O.
RB2	35	38	10	I/O	TTL	Digital I/O.
RB3/PGM RB3 PGM	36	39	11	I/O I/O	TTL	Digital I/O. Low voltage ICSP programming enable pin.
RB4	37	41	14	I/O	TTL	Digital I/O.
RB5	38	42	15	I/O	TTL	Digital I/O.
RB6/PGC RB6 PGC	39	43	16	I/O I/O	TTL/ST ⁽²⁾	Digital I/O. In-Circuit Debugger and ICSP programming clock.
RB7/PGD RB7 PGD	40	44	17	I/O I/O	TTL/ST ⁽²⁾	Digital I/O. In-Circuit Debugger and ICSP programming data.
RC0/T1OSO/T1CKI RC0 T1OSO T1CKI	15	16	32	I/O O I	ST	PORTC is a bi-directional I/O port. Digital I/O. Timer1 oscillator output. Timer1 external clock input.
RC1/T1OSI/CCP2 RC1 T1OSI CCP2	16	18	35	I/O I I/O	ST	Digital I/O. Timer1 oscillator input. Capture2 input, Compare2 output, PWM2 output.
RC2/CCP1 RC2 CCP1	17	19	36	I/O I/O	ST	Digital I/O. Capture1 input/Compare1 output/PWM1 output
RC3/SCK/SCL RC3 SCK SCL	18	20	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 ² C mode.
RC4/SDI/SDA RC4 SDI SDA	23	25	42	I/O I I/O	ST	Digital I/O. SPI data in. I ² C data I/O.
RC5/SDO RC5 SDO	24	26	43	I/O O	ST	Digital I/O. SPI data out.
RC6/TX/CK RC6 TX CK	25	27	44	I/O O I/O	ST	Digital I/O. USART asynchronous transmit. USART 1 synchronous clock.
RC7/RX/DT RC7 RX DT	26	29	1	I/O I I/O	ST	Digital I/O. USART asynchronous receive. USART 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 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.

PIC16F7X

TABLE 1-3: PIC16F74 AND PIC16F77 PINOUT DESCRIPTION (CONTINUED)

Pin Name	DIP Pin#	PLCC Pin#	QFP Pin#	I/O/P Type	Buffer Type	Description
RD0/PSP0 RD0 PSP0	19	21	38	I/O I/O	ST/TTL ⁽³⁾	PORTD is a bi-directional 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	22	39	I I/O I/O	ST/TTL ⁽³⁾	Digital I/O. Parallel Slave Port data.
RD2/PSP2 RD2 PSP2	21	23	40	I I/O I/O	ST/TTL ⁽³⁾	Digital I/O. Parallel Slave Port data.
RD3/PSP3 RD3 PSP3	22	24	41	I/O I/O	ST/TTL ⁽³⁾	Digital I/O. Parallel Slave Port data.
RD4/PSP4 RD4 PSP4	27	30	2	I/O I/O	ST/TTL ⁽³⁾	Digital I/O. Parallel Slave Port data.
RD5/PSP5 RD5 PSP5	28	31	3	I/O I/O	ST/TTL ⁽³⁾	Digital I/O. Parallel Slave Port data.
RD6/PSP6 RD6 PSP6	29	32	4	I/O I/O	ST/TTL ⁽³⁾	Digital I/O. Parallel Slave Port data.
RD7/PSP7 RD7 PSP7	30	33	5	I/O I/O	ST/TTL ⁽³⁾	Digital I/O. Parallel Slave Port data.
RE0/RD/AN5 RE0 RD AN5	8	9	25	I/O I I	ST/TTL ⁽³⁾	PORTE is a bi-directional I/O port. Digital I/O. Read control for parallel slave port . Analog input 5.
RE1/WR/AN6 RE1 WR AN6	9	10	26	I/O I I	ST/TTL ⁽³⁾	Digital I/O. Write control for parallel slave port . Analog input 6.
RE2/CS/AN7 RE2 CS AN7	10	11	27	I/O I I	ST/TTL ⁽³⁾	Digital I/O. Chip select control for parallel slave port . Analog input 7.
VSS	12,31	13,34	6,29	P	—	Ground reference for logic and I/O pins.
VDD	11,32	12,35	7,28	P	—	Positive supply for logic and I/O pins.
NC	—	1,17,28, 40	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**
- 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 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.

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).

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

2.1 Program Memory Organization

The PIC16F7X devices have a 13-bit program counter capable of addressing an 8K word x 14-bit program memory space. The PIC16F77/76 devices have 8K words of FLASH program memory and the PIC16F73/74 devices have 4K words. The program memory maps for PIC16F7X 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:

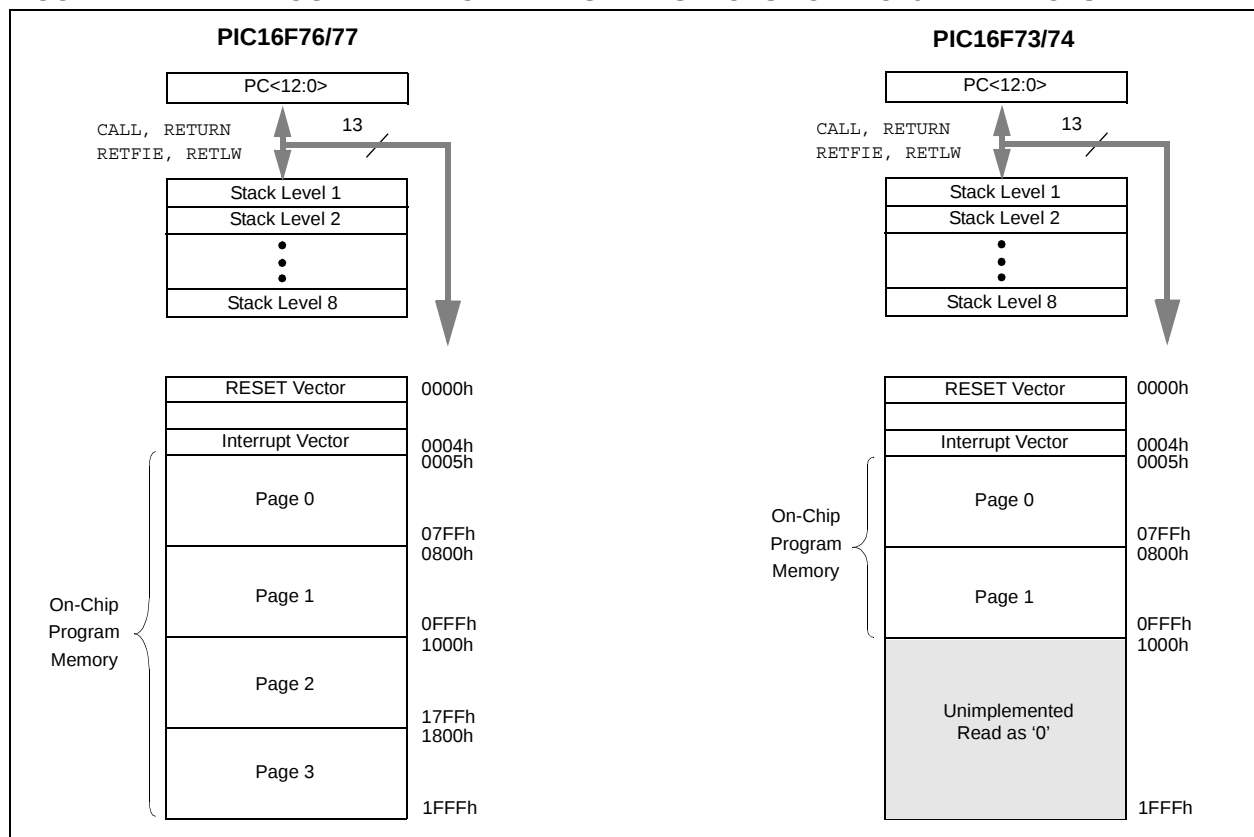
RP1:RP0	Bank
00	0
01	1
10	2
11	3

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 PIC16F7X DEVICES



PIC16F7X

FIGURE 2-2: PIC16F77/76 REGISTER FILE MAP

File Address	File Address	File Address	File Address
Indirect addr.(*) 00h	Indirect addr.(*) 80h	Indirect addr.(*) 100h	Indirect addr.(*) 180h
TMR0 01h	OPTION_REG 81h	TMR0 101h	OPTION_REG 181h
PCL 02h	PCL 82h	PCL 102h	PCL 182h
STATUS 03h	STATUS 83h	STATUS 103h	STATUS 183h
FSR 04h	FSR 84h	FSR 104h	FSR 184h
PORTA 05h	TRISA 85h	105h	185h
PORTB 06h	TRISB 86h	PORTB 106h	TRISB 186h
PORTC 07h	TRISC 87h	107h	187h
PORTD ⁽¹⁾ 08h	TRISD ⁽¹⁾ 88h	108h	188h
PORTE ⁽¹⁾ 09h	TRISE ⁽¹⁾ 89h	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	8Fh	PMADRH 10Fh	18Fh
T1CON 10h	90h	110h	190h
TMR2 11h	91h	111h	191h
T2CON 12h	PR2 92h	112h	192h
SSPBUF 13h	SSPADD 93h	113h	193h
SSPCON 14h	SSPSTAT 94h	114h	194h
CCPR1L 15h	95h	115h	195h
CCPR1H 16h	96h	116h	196h
CCP1CON 17h	97h	117h	197h
RCSTA 18h	TXSTA 98h	118h	198h
TXREG 19h	SPBRG 99h	119h	199h
RCREG 1Ah	9Ah	11Ah	19Ah
CCPR2L 1Bh	9Bh	11Bh	19Bh
CCPR2H 1Ch	9Ch	11Ch	19Ch
CCP2CON 1Dh	9Dh	11Dh	19Dh
ADRES 1Eh	9Eh	11Eh	19Eh
ADCON0 1Fh	ADCON1 9Fh	11Fh	19Fh
20h	A0h	120h	1A0h
General Purpose Register 96 Bytes	General Purpose Register 80 Bytes	General Purpose Register 80 Bytes	General Purpose Register 80 Bytes
7Fh	EFh	16Fh	1EFh
Bank 0	accesses 70h-7Fh	accesses 70h-7Fh	accesses 70h - 7Fh
	F0h	170h	1F0h
	FFh	17Fh	1FFh
Bank 1	Bank 2	Bank 3	

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

Note 1: These registers are not implemented on 28-pin devices.

FIGURE 2-3: PIC16F74/73 REGISTER FILE MAP

File Address	File Address	File Address	File Address
Indirect addr.(*) 00h	Indirect addr.(*) 80h	Indirect addr.(*) 100h	Indirect addr.(*) 180h
TMR0 01h	OPTION_REG 81h	TMR0 101h	OPTION_REG 181h
PCL 02h	PCL 82h	PCL 102h	PCL 182h
STATUS 03h	STATUS 83h	STATUS 103h	STATUS 183h
FSR 04h	FSR 84h	FSR 104h	FSR 184h
PORTA 05h	TRISA 85h		
PORTB 06h	TRISB 86h	PORTB 106h	TRISB 186h
PORTC 07h	TRISC 87h		
PORTD ⁽¹⁾ 08h	TRISD ⁽¹⁾ 88h		
PORTE ⁽¹⁾ 09h	TRISE ⁽¹⁾ 89h		
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	
TMR1L 0Eh	PCON 8Eh	PMDATH 10Eh	
TMR1H 0Fh		PMADRH 10Fh	
T1CON 10h			
TMR2 11h			
T2CON 12h	PR2 92h		
SSPBUF 13h	SSPADD 93h		
SSPCON 14h	SSPSTAT 94h		
CCPR1L 15h			
CCPR1H 16h			
CCP1CON 17h			
RCSTA 18h	TXSTA 98h		
TXREG 19h	SPBRG 99h		
RCREG 1Ah			
CCPR2L 1Bh			
CCPR2H 1Ch			
CCP2CON 1Dh			
ADRES 1Eh			
ADCON0 1Fh	ADCON1 9Fh		
20h	A0h	120h	1A0h
General Purpose Register 96 Bytes	General Purpose Register 96 Bytes	accesses 20h-7Fh	accesses A0h - FFh
Bank 0 7Fh	Bank 1 FFh	16Fh 170h 17Fh	1EFh 1F0h 1FFh
		Bank 2	Bank 3

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

Note 1: These registers are not implemented on 28-pin devices.

PIC16F7X

2.2.2 SPECIAL FUNCTION REGISTERS

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

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

TABLE 2-1: SPECIAL FUNCTION REGISTER SUMMARY

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Details on page
Bank 0											
00h ⁽⁴⁾	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	27, 96
01h	TMR0	Timer0 Module Register								xxxx xxxx	45, 96
02h ⁽⁴⁾	PCL	Program Counter (PC) Least Significant Byte								0000 0000	26, 96
03h ⁽⁴⁾	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxxx	19, 96
04h ⁽⁴⁾	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	27, 96
05h	PORTA	—	—	PORTA Data Latch when written: PORTA pins when read						--0x 0000	32, 96
06h	PORTB	PORTB Data Latch when written: PORTB pins when read								xxxx xxxx	34, 96
07h	PORTC	PORTC Data Latch when written: PORTC pins when read								xxxx xxxx	35, 96
08h ⁽⁵⁾	PORTD	PORTD Data Latch when written: PORTD pins when read								xxxx xxxx	36, 96
09h ⁽⁵⁾	PORTE	—	—	—	—	—	RE2	RE1	RE0	---- -xxx	39, 96
0Ah ^(1,4)	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	26, 96
0Bh ⁽⁴⁾	INTCON	GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBIF	0000 000x	21, 96
0Ch	PIR1	PSPIF ⁽³⁾	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	23, 96
0Dh	PIR2	—	—	—	—	—	—	—	CCP2IF	---- --0	24, 96
0Eh	TMR1L	Holding Register for the Least Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	50, 96
0Fh	TMR1H	Holding Register for the Most Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	50, 96
10h	T1CON	—	—	T1CKPS1	T1CKPS0	T1OSCEN	T1SYN $\overline{C}$	TMR1CS	TMR1ON	--00 0000	47, 96
11h	TMR2	Timer2 Module Register								0000 0000	52, 96
12h	T2CON	—	TOUTPS3	TOUTPS2	TOUTPS	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	52, 96
13h	SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register								xxxx xxxx	64, 68, 96
14h	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	61, 96
15h	CCPR1L	Capture/Compare/PWM Register1 (LSB)								xxxx xxxx	56, 96
16h	CCPR1H	Capture/Compare/PWM Register1 (MSB)								xxxx xxxx	56, 96
17h	CCP1CON	—	—	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	54, 96
18h	RCSTA	SPEN	RX9	SREN	CREN	—	FERR	OERR	RX9D	0000 -00x	70, 96
19h	TXREG	USART Transmit Data Register								0000 0000	74, 96
1Ah	RCREG	USART Receive Data Register								0000 0000	76, 96
1Bh	CCPR2L	Capture/Compare/PWM Register2 (LSB)								xxxx xxxx	58, 96
1Ch	CCPR2H	Capture/Compare/PWM Register2 (MSB)								xxxx xxxx	58, 96
1Dh	CCP2CON	—	—	CCP2X	CCP2Y	CCP2M3	CCP2M2	CCP2M1	CCP2M0	--00 0000	54, 96
1Eh	ADRES	A/D Result Register Byte								xxxx xxxx	88, 96
1Fh	ADCON0	ADCS1	ADCS0	CHS2	CHS1	CHS0	$\overline{GO/DONE}$	—	ADON	0000 00-0	83, 96

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>, 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, read as '0'.
- 6:** This bit always reads as a '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	
Bank 1												
80h ⁽⁴⁾	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	27, 96	
81h	OPTION_REG	RBP $\overline{U}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	20, 44, 96	
82h ⁽⁴⁾	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	26, 96	
83h ⁽⁴⁾	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	19, 96	
84h ⁽⁴⁾	FSR	Indirect data memory address pointer								xxxx xxxx	27, 96	
85h	TRISA	—	—	PORTA Data Direction Register							--11 1111	32, 96
86h	TRISB	PORTB Data Direction Register								1111 1111	34, 96	
87h	TRISC	PORTC Data Direction Register								1111 1111	35, 96	
88h ⁽⁵⁾	TRISD	PORTD Data Direction Register								1111 1111	36, 96	
89h ⁽⁵⁾	TRISE	IBF	OBF	IBOV	PSPMODE	—	PORTE Data Direction Bits			0000 -111	38, 96	
8Ah ^(1,4)	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	21, 96	
8Bh ⁽⁴⁾	INTCON	GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBIF	0000 000x	23, 96	
8Ch	PIE1	PSPIE ⁽³⁾	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	22, 96	
8Dh	PIE2	—	—	—	—	—	—	—	CCP2IE	---- ---0	24, 97	
8Eh	PCON	—	—	—	—	—	—	POR	BOR	---- --qg	25, 97	
8Fh	—	Unimplemented								—	—	
90h	—	Unimplemented								—	—	
91h	—	Unimplemented								—	—	
92h	PR2	Timer2 Period Register								1111 1111	52, 97	
93h	SSPADD	Synchronous Serial Port (I ² C mode) Address Register								0000 0000	68, 97	
94h	SSPSTAT	SMP	CKE	D/ $\overline{A}$	P	S	R/ $\overline{W}$	UA	BF	0000 0000	60, 97	
95h	—	Unimplemented								—	—	
96h	—	Unimplemented								—	—	
97h	—	Unimplemented								—	—	
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	69, 97	
99h	SPBRG	Baud Rate Generator Register								0000 0000	71, 97	
9Ah	—	Unimplemented								—		
9Bh	—	Unimplemented								—		
9Ch	—	Unimplemented								—		
9Dh	—	Unimplemented								—		
9Eh	—	Unimplemented								—		
9Fh	ADCON1	—	—	—	—	—	PCFG2	PCFG1	PCFG0	---- -000	84, 97	

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>, 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, read as '0'.
- 6:** This bit always reads as a '1'.

PIC16F7X

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

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Details on page
Bank 2											
100h ⁽⁴⁾	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	27, 96
101h	TMR0	Timer0 Module Register								xxxx xxxx	45, 96
102h ⁽⁴⁾	PCL	Program Counter (PC) Least Significant Byte								0000 0000	26, 96
103h ⁽⁴⁾	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	19, 96
104h ⁽⁴⁾	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	27, 96
105h	—	Unimplemented								—	—
106h	PORTB	PORTB Data Latch when written: PORTB pins when read								xxxx xxxx	34, 96
107h	—	Unimplemented								—	—
108h	—	Unimplemented								—	—
109h	—	Unimplemented								—	—
10Ah ^(1,4)	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	21, 96
10Bh ⁽⁴⁾	INTCON	GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBIF	0000 000x	23, 96
10Ch	PMDATA	Data Register Low Byte								xxxx xxxx	29, 97
10Dh	PMADR	Address Register Low Byte								xxxx xxxx	29, 97
10Eh	PMDATH	—	—	Data Register High Byte						xxxx xxxx	29, 97
10Fh	PMADRH	—	—	—	Address Register High Byte					xxxx xxxx	29, 97
Bank 3											
180h ⁽⁴⁾	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	27, 96
181h	OPTION_REG	$\overline{RBPU}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	20, 44, 96
182h ⁽⁴⁾	PCL	Program Counter (PC) Least Significant Byte								0000 0000	26, 96
183h ⁽⁴⁾	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	19, 96
184h ⁽⁴⁾	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	27, 96
185h	—	Unimplemented								—	—
186h	TRISB	PORTB Data Direction Register								1111 1111	34, 96
187h	—	Unimplemented								—	—
188h	—	Unimplemented								—	—
189h	—	Unimplemented								—	—
18Ah ^(1,4)	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	21, 96
18Bh ⁽⁴⁾	INTCON	GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBIF	0000 000x	23, 96
18Ch	PMCON1	— ⁽⁶⁾	—	—	—	—	—	—	RD	1--- ---0	29, 97
18Dh	—	Unimplemented								—	
18Eh	—	Reserved maintain clear								0000 0000	
18Fh	—	Reserved maintain clear								0000 0000	

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

Shaded locations are unimplemented, read as '0'.

- Note**
- 1: The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8>, 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, read as '0'.
 - 6: This bit always reads as a '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 the "Instruction Set Summary."

Note 1: The $\overline{\text{C}}$ and $\overline{\text{DC}}$ bits operate as a borrow and digit borrow bit, respectively, in subtraction. See the `SUBLW` and `SUBWF` instructions for examples.

REGISTER 2-1: STATUS REGISTER (ADDRESS 03h, 83h, 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 $\overline{\text{borrow}}$, 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 reset	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

PIC16F7X

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 known also as the prescaler), the External INT Interrupt, TMR0 and the weak pull-ups on PORTB.

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

REGISTER 2-2: OPTION_REG 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
RBP	U	T0CS	T0SE	PSA	PS2	PS1	PS0
bit 7							bit 0

- bit 7 **RBP**: 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 (CLKOUT)
- 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 reset '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 enable bit, GIE (INTCON<7>). User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

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

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

- bit 7 **GIE:** Global Interrupt Enable bit
1 = Enables all unmasked interrupts
0 = Disables all interrupts
- bit 6 **PEIE:** Peripheral Interrupt Enable bit
1 = Enables all unmasked peripheral interrupts
0 = Disables all peripheral interrupts
- bit 5 **TMR0IE:** TMR0 Overflow Interrupt Enable bit
1 = Enables the TMR0 interrupt
0 = Disables the TMR0 interrupt
- bit 4 **INTE:** RB0/INT External Interrupt Enable bit
1 = Enables the RB0/INT external interrupt
0 = Disables the RB0/INT external interrupt
- bit 3 **RBIE:** RB Port Change Interrupt Enable bit
1 = Enables the RB port change interrupt
0 = Disables the RB port change interrupt
- bit 2 **TMR0IF:** TMR0 Overflow Interrupt Flag bit
1 = TMR0 register has overflowed (must be cleared in software)
0 = TMR0 register did not overflow
- bit 1 **INTF:** RB0/INT External Interrupt Flag bit
1 = The RB0/INT external interrupt occurred (must be cleared in software)
0 = The RB0/INT external interrupt did not occur
- bit 0 **RBIF:** RB Port Change Interrupt Flag bit
A mismatch condition will continue to set flag bit RBIF. Reading PORTB will end the mismatch condition and allow flag bit RBIF to be cleared.
1 = At least one of the RB7:RB4 pins changed state (must be cleared in software)
0 = None of the RB7:RB4 pins have changed state

Legend:

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

PIC16F7X

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 REGISTER (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 ⁽¹⁾	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE
bit 7							bit 0

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

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

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**: USART Receive Interrupt Enable bit

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

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

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

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

Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

- n = Value at POR reset

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

REGISTER 2-5: PIR1 REGISTER (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 ⁽¹⁾	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF
bit 7							bit 0

- bit 7 **PSPIF⁽¹⁾:** Parallel Slave Port Read/Write Interrupt Flag bit
 1 = A read or a write operation has taken place (must be cleared in software)
 0 = No read or write has occurred
- 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:** USART Receive Interrupt Flag bit
 1 = The USART receive buffer is full
 0 = The USART receive buffer is empty
- bit 4 **TXIF:** USART Transmit Interrupt Flag bit
 1 = The USART transmit buffer is empty
 0 = The USART transmit buffer is full
- bit 3 **SSPIF:** Synchronous Serial Port (SSP) Interrupt Flag
 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.
 I²C Slave
 A transmission/reception has taken place.
 I²C 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

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

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

PIC16F7X

2.2.2.6 PIE2 Register

The PIE2 register contains the individual enable bits for the CCP2 peripheral interrupt.

REGISTER 2-6: PIE2 REGISTER (ADDRESS 8Dh)

	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
	—	—	—	—	—	—	CCP2IE
	bit 7						bit 0
bit 7-1	Unimplemented: Read as '0'						
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 reset

'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 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 REGISTER (ADDRESS 0Dh)

	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
	—	—	—	—	—	—	CCP2IF
	bit 7						bit 0
bit 7-1	Unimplemented: Read as '0'						
bit 0	CCP2IF: CCP2 Interrupt Flag bit						
	<u>Capture mode:</u>						
	1 = A TMR1 register capture occurred (must be cleared in software)						
	0 = No TMR1 register capture occurred						
	<u>Compare mode:</u>						
	1 = A TMR1 register compare match occurred (must be cleared in software)						
	0 = No TMR1 register compare match occurred						
	<u>PWM mode:</u>						
	Unused						

Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

- n = Value at POR reset

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

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: 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 BODEN bit in the configuration word).

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

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-1
—	—	—	—	—	—	POR	BOR
bit 7						bit 0	

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

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

1 = No Power-on Reset occurred

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

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

1 = No Brown-out Reset occurred

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

Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

- n = Value at POR reset

'1' = Bit is set

'0' = Bit is cleared

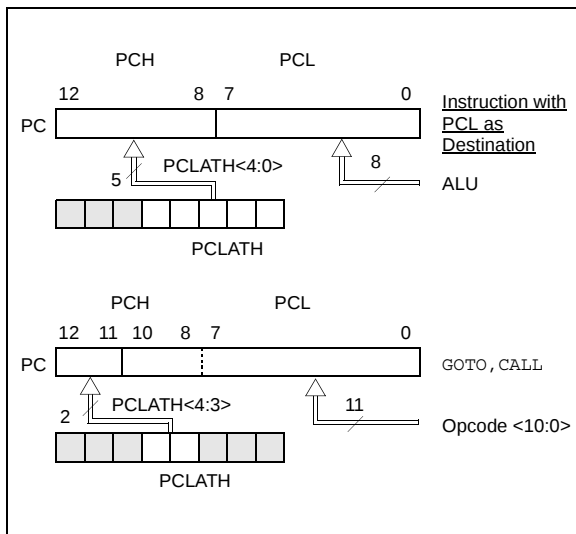
x = Bit is unknown

PIC16F7X

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, "Implementing a Table Read" (AN556).

2.3.2 STACK

The PIC16F7X 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

PIC16F7X devices are capable of addressing a continuous 8K word block of program memory. The CALL and GOTO instructions provide only 11 bits of address to allow branching within any 2K program memory page. When doing a CALL or GOTO instruction, the upper 2 bits of the address are provided by PCLATH<4:3>. When doing a CALL or GOTO instruction, the user must ensure that the page select bits are programmed so that the desired program memory page is addressed. If a return from a CALL instruction (or interrupt) is executed, the entire 13-bit PC is popped off the stack. Therefore, manipulation of the PCLATH<4:3> bits are 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 setup 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)
```

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.

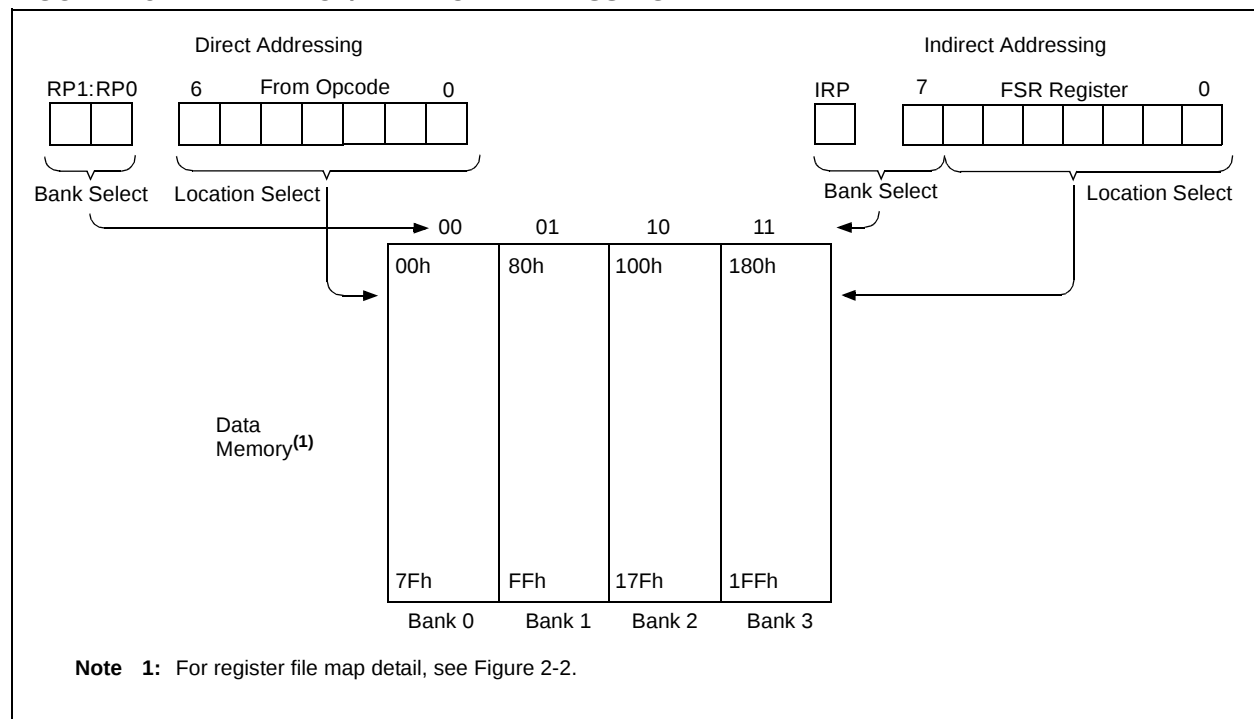
A simple program to clear RAM locations 20h-2Fh using indirect addressing is shown in Example 2-2.

EXAMPLE 2-2: INDIRECT ADDRESSING

```

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

FIGURE 2-5: DIRECT/INDIRECT ADDRESSING



PIC16F7X

NOTES:

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 MSByte of the address is written to the PMADRH register and the LSByte is written to the PMADR register. The upper MSbits 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 REGISTER (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 reset

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

PIC16F7X

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
Required Sequence	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 PMDATA

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	Address Register Low Byte								xxxx xxxx	uuuu uuuu
10Fh	PMADRH	—	—	—	Address Register High Byte					xxxx xxxx	uuuu uuuu
10Ch	PMDATA	Data Register Low Byte								xxxx xxxx	uuuu uuuu
10Eh	PMDATH	—	—	Data Register High Byte					xxxx xxxx	uuuu uuuu	
18Ch	PMCON1	— ⁽¹⁾	—	—	—	—	—	—	RD	1--- ---0	1--- ---0

Legend: x = unknown, u = unchanged, r = reserved, - = unimplemented read as '0'. Shaded cells are not used during FLASH access.

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

4.0 I/O PORTS

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

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

4.1 PORTA and the TRISA Register

PORTA is a 6-bit wide, bi-directional 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 Hi-Impedance mode). Clearing a TRISA bit (= '0') will make the corresponding PORTA pin an output (i.e., put the contents of the output latch on the selected pin).

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

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

Other PORTA pins are multiplexed with analog inputs and analog VREF input. The operation of each pin is selected by clearing/setting the control bits in the ADCON1 register (A/D Control Register1).

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

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

EXAMPLE 4-1: 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	0x06	; 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'.

FIGURE 4-1: BLOCK DIAGRAM OF RA3:RA0 AND RA5 PINS

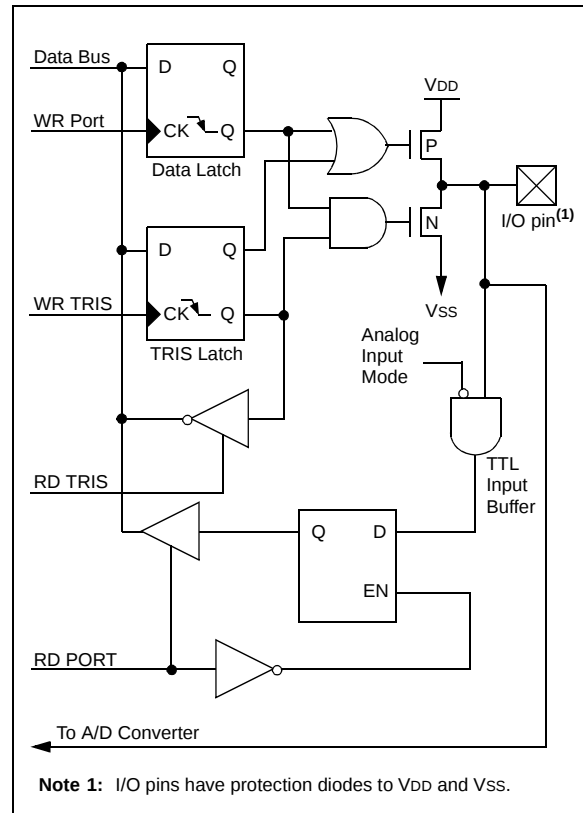
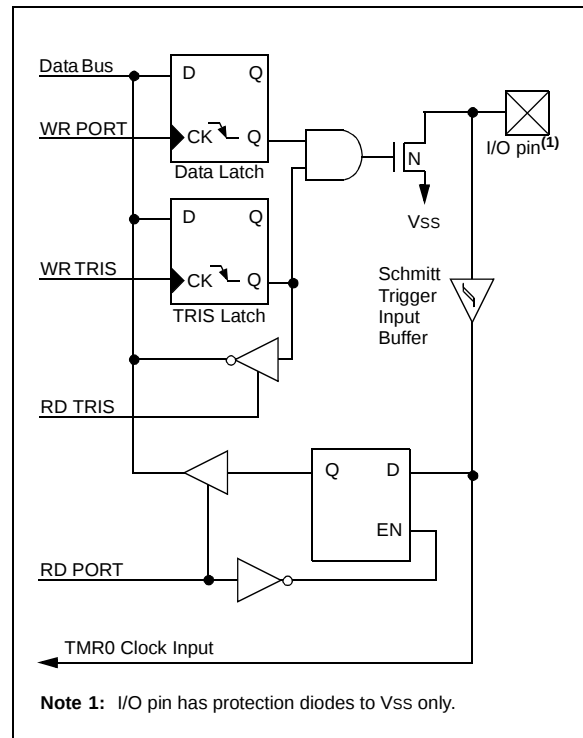


FIGURE 4-2: BLOCK DIAGRAM OF RA4/T0CKI PIN



PIC16F7X

TABLE 4-1: PORTA FUNCTIONS

Name	Bit#	Buffer	Function
RA0/AN0	bit0	TTL	Input/output or analog input.
RA1/AN1	bit1	TTL	Input/output or analog input.
RA2/AN2	bit2	TTL	Input/output or analog input.
RA3/AN3/VREF	bit3	TTL	Input/output or analog input or VREF.
RA4/T0CKI	bit4	ST	Input/output or external clock input for Timer0. Output is open drain type.
RA5/ $\overline{SS}$ /AN4	bit5	TTL	Input/output or slave select input for synchronous serial port or analog input.

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

TABLE 4-2: 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	—	—	RA5	RA4	RA3	RA2	RA1	RA0	--0x 0000	--0u 0000
85h	TRISA	—	—	PORTA Data Direction Register						--11 1111	--11 1111
9Fh	ADCON1	—	—	—	—	—	PCFG2	PCFG1	PCFG0	---- -000	---- -000

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

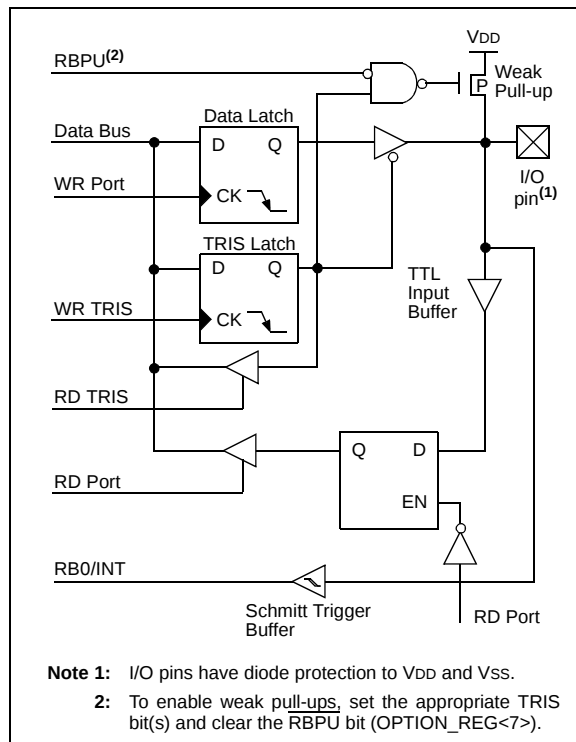
Note: When using the SSP module in SPI Slave mode and $\overline{SS}$ enabled, the A/D converter must be set to one of the following modes where PCFG2:PCFG0 = 100, 101, 11x.

4.2 PORTB and the TRISB Register

PORTB is an 8-bit wide, bi-directional 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 Hi-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.

FIGURE 4-3: BLOCK DIAGRAM OF RB3:RB0 PINS



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:

- Any read or write of PORTB. This will end the mismatch condition.
- 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.

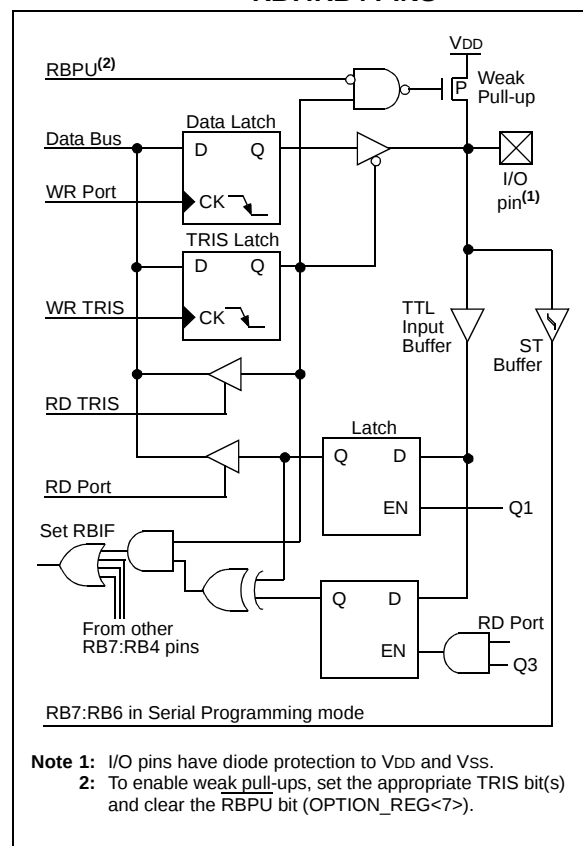
The interrupt-on-change feature is recommended for wake-up on key depression operation and operations where PORTB is only used for the interrupt-on-change feature. Polling of PORTB is not recommended while using the interrupt-on-change feature.

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 Embedded Control Handbook, "Implementing Wake-up on Key Stroke" (AN552).

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 12.11.1.

FIGURE 4-4: BLOCK DIAGRAM OF RB7:RB4 PINS



PIC16F7X

TABLE 4-3: PORTB FUNCTIONS

Name	Bit#	Buffer	Function
RB0/INT	bit0	TTL/ST ⁽¹⁾	Input/output pin or external interrupt input. Internal software programmable weak pull-up.
RB1	bit1	TTL	Input/output pin. Internal software programmable weak pull-up.
RB2	bit2	TTL	Input/output pin. Internal software programmable weak pull-up.
RB3	bit3	TTL	Input/output pin. Internal software programmable weak pull-up.
RB4	bit4	TTL	Input/output pin (with interrupt-on-change). Internal software programmable weak pull-up.
RB5	bit5	TTL	Input/output pin (with interrupt-on-change). Internal software programmable weak pull-up.
RB6	bit6	TTL/ST ⁽²⁾	Input/output pin (with interrupt-on-change). Internal software programmable weak pull-up. Serial programming clock.
RB7	bit7	TTL/ST ⁽²⁾	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 4-4: SUMMARY OF REGISTERS ASSOCIATED WITH PORTB

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

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

4.3 PORTC and the TRISC Register

PORTC is an 8-bit wide, bi-directional 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 Hi-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 4-5). 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 13.1 for additional information on read-modify-write operations.

FIGURE 4-5: PORTC BLOCK DIAGRAM (PERIPHERAL OUTPUT OVERRIDE)

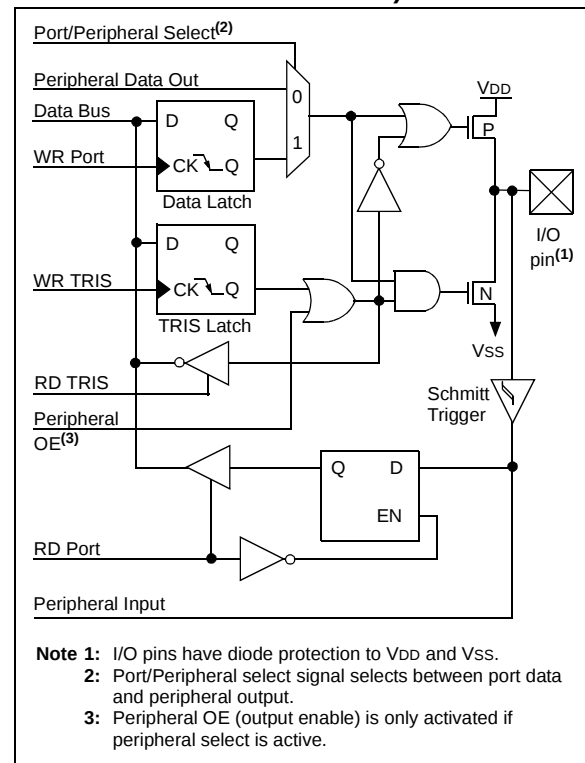


TABLE 4-5: PORTC FUNCTIONS

Name	Bit#	Buffer Type	Function
RC0/T1OSO/T1CKI	bit0	ST	Input/output port pin or Timer1 oscillator output/Timer1 clock input.
RC1/T1OSI/CCP2	bit1	ST	Input/output port pin or Timer1 oscillator input or Capture2 input/Compare2 output/PWM2 output.
RC2/CCP1	bit2	ST	Input/output port pin or Capture1 input/Compare1 output/PWM1 output.
RC3/SCK/SCL	bit3	ST	RC3 can also be the synchronous serial clock for both SPI and I ² C modes.
RC4/SDI/SDA	bit4	ST	RC4 can also be the SPI Data In (SPI mode) or Data I/O (I ² C mode).
RC5/SDO	bit5	ST	Input/output port pin or Synchronous Serial Port data output.
RC6/TX/CK	bit6	ST	Input/output port pin or USART Asynchronous Transmit or Synchronous Clock.
RC7/RX/DT	bit7	ST	Input/output port pin or USART Asynchronous Receive or Synchronous Data.

Legend: ST = Schmitt Trigger input

TABLE 4-6: 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

PIC16F7X

4.4 PORTD and TRISD Registers

This section is not applicable to the PIC16F73 or PIC16F76.

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

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

FIGURE 4-6: PORTD BLOCK DIAGRAM (IN I/O PORT MODE)

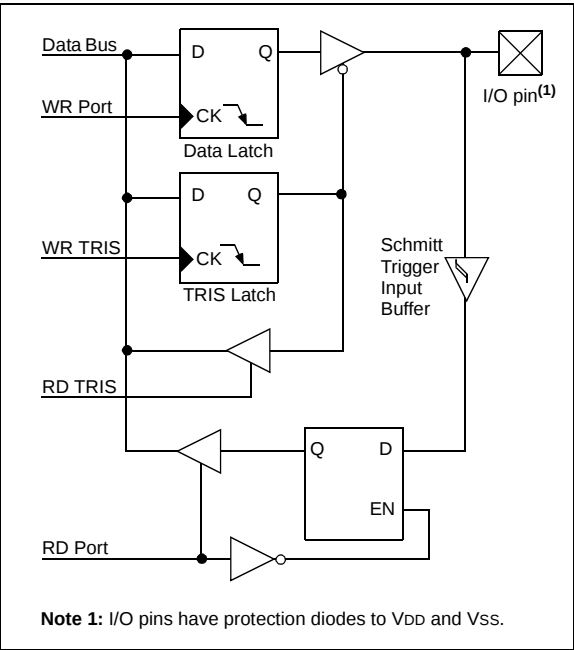


TABLE 4-7: PORTD FUNCTIONS

Name	Bit#	Buffer Type	Function
RD0/PSP0	bit0	ST/TTL ⁽¹⁾	Input/output port pin or parallel slave port bit0
RD1/PSP1	bit1	ST/TTL ⁽¹⁾	Input/output port pin or parallel slave port bit1
RD2/PSP2	bit2	ST/TTL ⁽¹⁾	Input/output port pin or parallel slave port bit2
RD3/PSP3	bit3	ST/TTL ⁽¹⁾	Input/output port pin or parallel slave port bit3
RD4/PSP4	bit4	ST/TTL ⁽¹⁾	Input/output port pin or parallel slave port bit4
RD5/PSP5	bit5	ST/TTL ⁽¹⁾	Input/output port pin or parallel slave port bit5
RD6/PSP6	bit6	ST/TTL ⁽¹⁾	Input/output port pin or parallel slave port bit6
RD7/PSP7	bit7	ST/TTL ⁽¹⁾	Input/output port pin or parallel slave port bit7

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 4-8: 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	—	PORTE Data Direction bits			0000 -111	0000 -111

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

4.5 PORTE and TRISE Register

This section is not applicable to the PIC16F73 or PIC16F76.

PORTE has three pins, RE0/ $\overline{\text{RD}}$ /AN5, RE1/ $\overline{\text{WR}}$ /AN6 and RE2/ $\overline{\text{CS}}$ /AN7, which are individually configurable as inputs or outputs. These pins have Schmitt Trigger input buffers.

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.

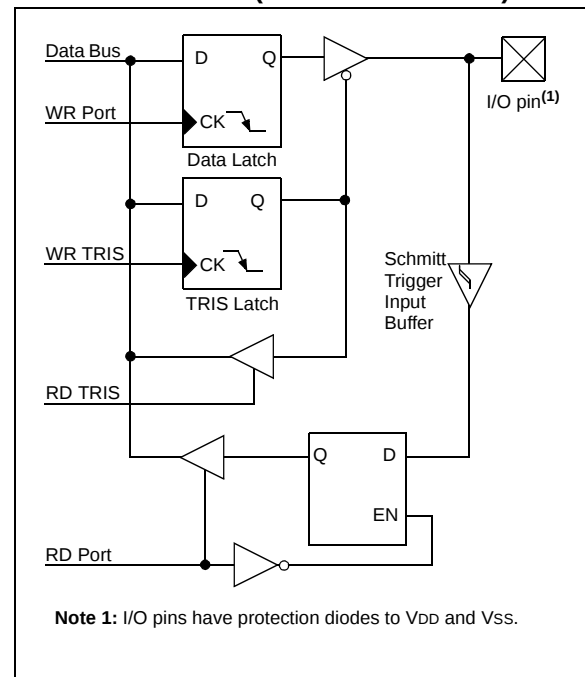
Register 4-1 shows the TRISE register, which also controls the parallel slave port operation.

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 4-7: PORTE BLOCK DIAGRAM (IN I/O PORT MODE)



PIC16F7X

REGISTER 4-1: 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	—	Bit2	Bit1	Bit0

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 '0'

bit 2 **PORTE Data Direction bits:**

Bit2: Direction Control bit for pin RE2/ $\overline{CS}$ /AN7

1 = Input

0 = Output

bit 1 **Bit1:** Direction Control bit for pin RE1/ $\overline{WR}$ /AN6

1 = Input

0 = Output

bit 0 **Bit0:** Direction Control bit for pin RE0/ $\overline{RD}$ /AN5

1 = Input

0 = Output

Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

- n = Value at POR reset

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

TABLE 4-9: PORTE FUNCTIONS

Name	Bit#	Buffer Type	Function
RE0/ $\overline{\text{RD}}$ /AN5	bit0	ST/TTL ⁽¹⁾	Input/output port pin or read control input in Parallel Slave Port mode or analog input. For $\overline{\text{RD}}$ (PSP mode): 1 = IDLE 0 = Read operation. Contents of PORTD register output to PORTD I/O pins (if chip selected).
RE1/ $\overline{\text{WR}}$ /AN6	bit1	ST/TTL ⁽¹⁾	Input/output port pin or write control input in Parallel Slave Port mode or analog input. For $\overline{\text{WR}}$ (PSP mode): 1 = IDLE 0 = Write operation. Value of PORTD I/O pins latched into PORTD register (if chip selected).
RE2/ $\overline{\text{CS}}$ /AN7	bit2	ST/TTL ⁽¹⁾	Input/output port pin or chip select control input in Parallel Slave Port mode or analog input. For $\overline{\text{CS}}$ (PSP mode): 1 = Device is not selected 0 = Device is selected

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 4-10: 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	—	—	—	—	—	RE2	RE1	RE0	---- -xxx	---- -uuu
89h	TRISE	IBF	OBF	IBOV	PSPMODE	—	PORTE Data Direction bits			0000 -111	0000 -111
9Fh	ADCON1	—	—	—	—	—	PCFG2	PCFG1	PCFG0	---- -000	---- -000

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

PIC16F7X

4.6 Parallel Slave Port

The Parallel Slave Port (PSP) is not implemented on the PIC16F73 or PIC16F76.

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/RD, the write control input pin RE1/WR, and the chip select control input pin RE2/CS.

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/RD to be the RD input, RE1/WR to be the WR input and RE2/CS to be the 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 CS and WR lines are both detected low. Firmware can read the actual data on the PORTD pins during this time. When either the CS or 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 4-9). 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 CS and 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 4-10), 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 CS or 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 4-8: PORTD AND PORTE BLOCK DIAGRAM (PARALLEL SLAVE PORT)

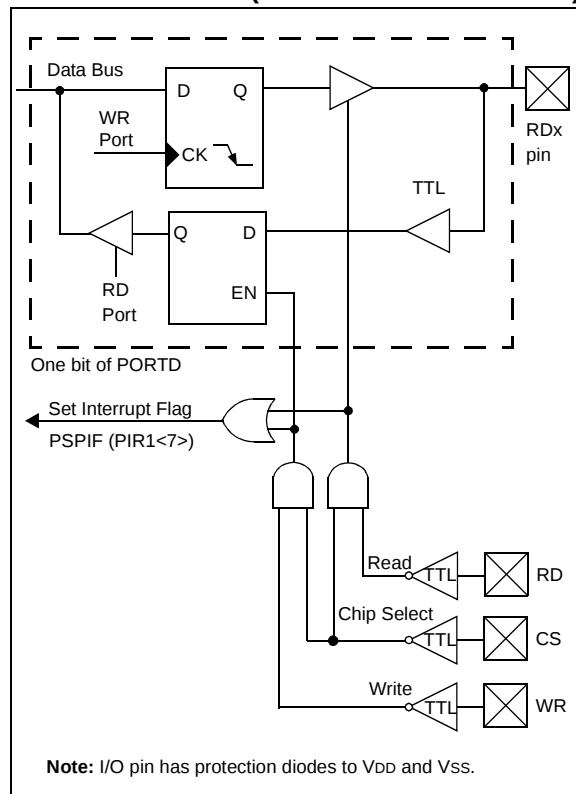


FIGURE 4-9: PARALLEL SLAVE PORT WRITE WAVEFORMS

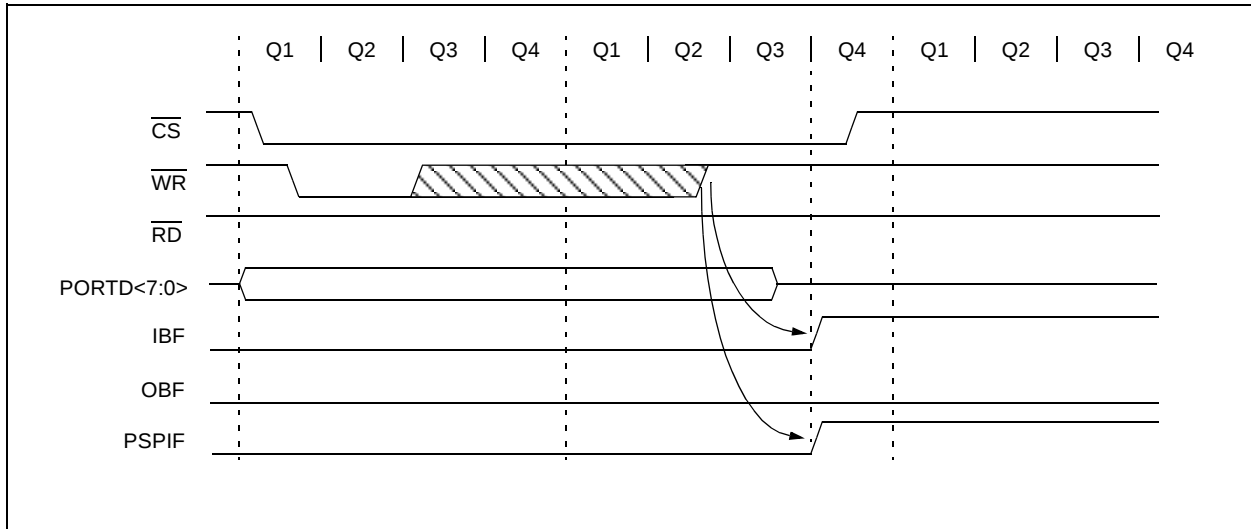


FIGURE 4-10: PARALLEL SLAVE PORT READ WAVEFORMS

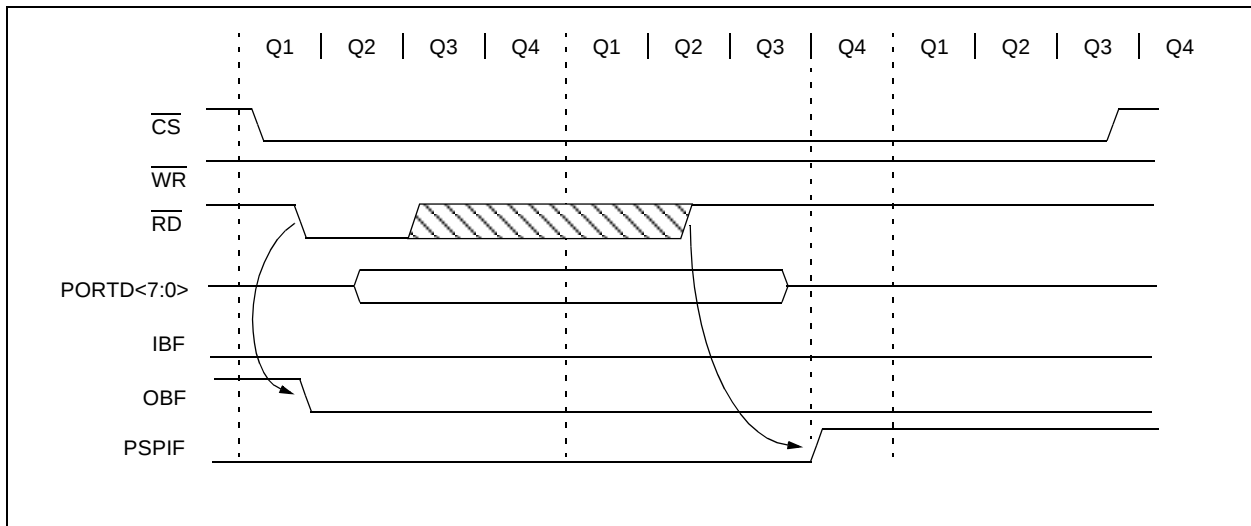


TABLE 4-11: REGISTERS ASSOCIATED WITH PARALLEL SLAVE PORT

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	—	—	—	—	—	RE2	RE1	RE0	---- -xxx	---- -uuu
89h	TRISE	IBF	OBF	IBOV	PSPMODE	—	PORTE Data Direction Bits			0000 -111	0000 -111
0Ch	PIR1	PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
8Ch	PIE1	PSPIE ⁽¹⁾	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
9Fh	ADCON1	—	—	—	—	—	PCFG2	PCFG1	PCFG0	---- -000	---- -000

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 PIC16F73/76; always maintain these bits clear.

PIC16F7X

NOTES:

5.0 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 5-1 is a block diagram of the Timer0 module and the prescaler shared with the WDT.

Timer0 operation is controlled through the OPTION_REG register (Register 5-1 on the following page). 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.

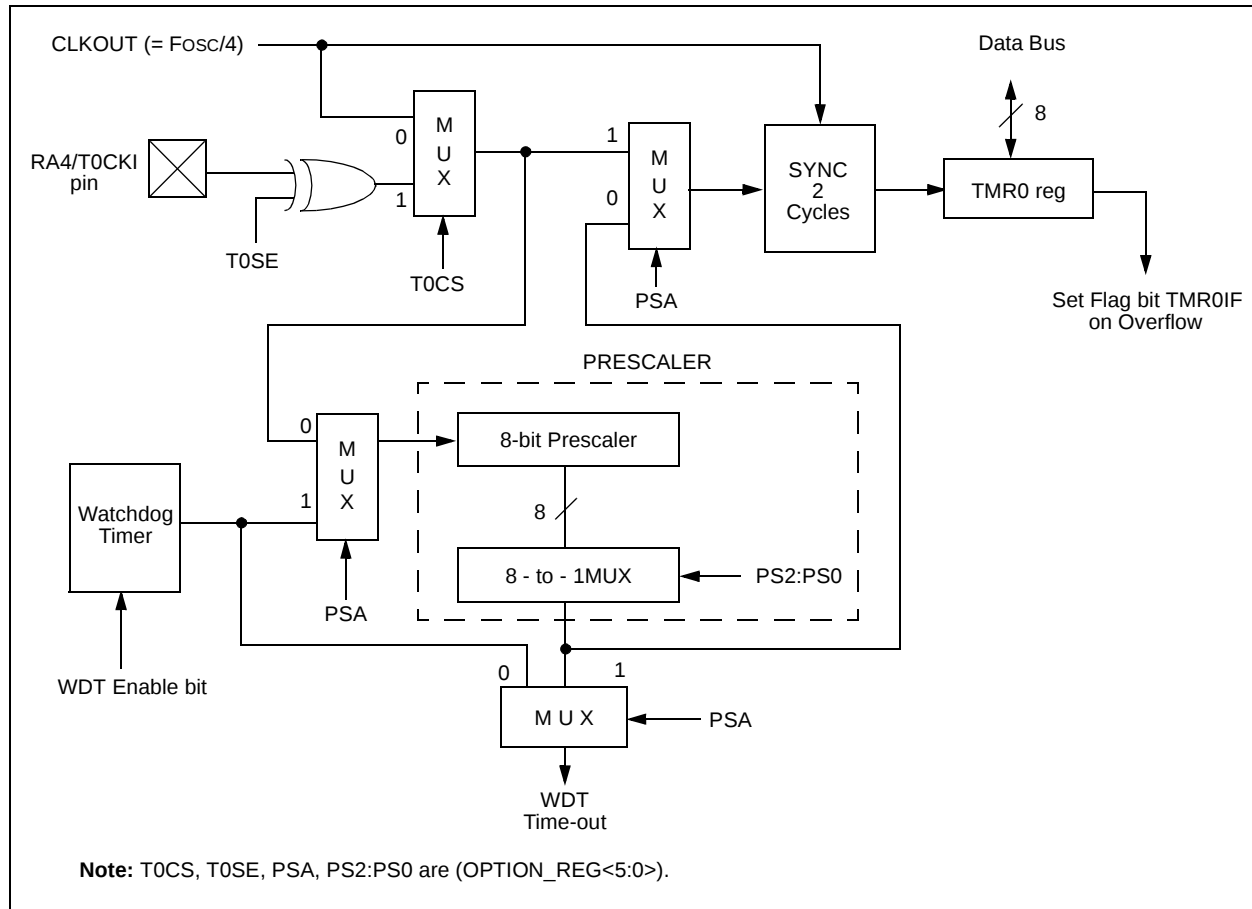
Counter mode is selected by setting bit T0CS (OPTION_REG<5>). In Counter mode, Timer0 will increment, either on every rising or falling edge of pin RA4/T0CKI. The incrementing edge is determined by the Timer0 Source Edge Select bit T0SE (OPTION_REG<4>). Clearing bit T0SE selects the rising edge. Restrictions on the external clock input are discussed in detail in Section 5.2.

The prescaler is mutually exclusively shared between the Timer0 module and the Watchdog Timer. The prescaler is not readable or writable. Section 5.3 details the operation of the prescaler.

5.1 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 5-1: BLOCK DIAGRAM OF THE TIMER0 MODULE AND PRESCALER



PIC16F7X

5.2 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 2Tosc (and a small RC delay of 20 ns) and low for at least 2Tosc (and a small RC delay of 20 ns). Refer to the electrical specification of the desired device.

REGISTER 5-1: OPTION_REG REGISTER

	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
	RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS0
bit 7							bit 0
bit 7	RBPU: PORTB Pull-up Enable bit (see Section 2.2.2.2)						
bit 6	INTEDG: Interrupt Edge Select bit (see Section 2.2.2.2)						
bit 5	T0CS: TMR0 Clock Source Select bit						
	1 = Transition on T0CKI pin						
	0 = Internal instruction cycle clock (CLKOUT)						
bit 4	T0SE: TMR0 Source Edge Select bit						
	1 = Increment on high-to-low transition on T0CKI pin						
	0 = Increment on low-to-high transition on T0CKI pin						
bit 3	PSA: Prescaler Assignment bit						
	1 = Prescaler is assigned to the WDT						
	0 = Prescaler is assigned to the Timer0 module						
bit 2-0	PS2:PS0: Prescaler Rate Select bits						
	Bit Value	TMR0 Rate	WDT Rate				
	000	1 : 2	1 : 1				
	001	1 : 4	1 : 2				
	010	1 : 8	1 : 4				
	011	1 : 16	1 : 8				
	100	1 : 32	1 : 16				
	101	1 : 64	1 : 32				
	110	1 : 128	1 : 64				
	111	1 : 256	1 : 128				

Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

- n = Value at POR reset

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

Note: To avoid an unintended device RESET, the instruction sequences shown in Example 5-1 and Example 5-2 (page 45) must be executed when changing the prescaler assignment between Timer0 and the WDT. This sequence must be followed even if the WDT is disabled.

5.3 Prescaler

There is only one prescaler available on the microcontroller; it is shared exclusively between the Timer0 module and the Watchdog Timer. The usage of the prescaler is also mutually exclusive: that is, a prescaler assignment for the Timer0 module means that there is no prescaler for the Watchdog Timer, and vice versa. This prescaler is not readable or writable (see Figure 5-1).

The PSA and PS2:PS0 bits (OPTION_REG<3:0>) determine the prescaler assignment and prescale ratio. Examples of code for assigning the prescaler assignment are shown in Example 5-1 and Example 5-2. Note that when the prescaler is being assigned to the WDT with ratios other than 1:1, lines 2 and 3 (highlighted) are optional. If a prescale ratio of 1:1 is to be used,

however, these lines must be used to set a temporary value. The final 1:1 value is then set in lines 10 and 11 (highlighted). (Line numbers are included in the example for illustrative purposes only, and are not part of the actual code.)

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 CLRWDW instruction will clear the prescaler along with the Watchdog Timer.

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

EXAMPLE 5-1: CHANGING THE PRESCALER ASSIGNMENT FROM TIMER0 TO WDT

```

1) BSF      STATUS, RP0      ; Bank1
2) MOVLW    b'xx0x0xxx'      ; Select clock source and prescale value of
3) MOVWF    OPTION_REG       ; other than 1:1
4) BCF      STATUS, RP0      ; Bank0
5) CLRF     TMR0              ; Clear TMR0 and prescaler
6) BSF      STATUS, RP1      ; Bank1
7) MOVLW    b'xxxx1xxx'      ; Select WDT, do not change prescale value
8) MOVWF    OPTION_REG
9) CLRWDW                    ; Clears WDT and prescaler
10) MOVLW    b'xxxx1xxx'      ; Select new prescale value and WDT
11) MOVWF    OPTION_REG
12) BCF      STATUS, RP0      ; Bank0

```

EXAMPLE 5-2: CHANGING THE PRESCALER ASSIGNMENT FROM WDT TO TIMER0

```

CLRWDW      ; Clear WDT and prescaler
BSF          STATUS, RP0      ; Bank1
MOVLW        b'xxxx0xxx'      ; Select TMR0, new prescale
MOVWF        OPTION_REG       ; value and clock source
BCF          STATUS, RP0      ; Bank0

```

TABLE 5-1: REGISTERS ASSOCIATED WITH TIMER0

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other RESETS
01h,101h	TMR0	Timer0 Module Register								xxxx xxxx	uuuu uuuu
0Bh,8Bh, 10Bh,18Bh	INTCON	GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBIF	0000 000x	0000 000u
81h,181h	OPTION_REG	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.

PIC16F7X

NOTES:

6.0 TIMER1 MODULE

The Timer1 module is a 16-bit timer/counter consisting of two 8-bit registers (TMR1H and TMR1L), which are readable and writable. The TMR1 Register pair (TMR1H:TMR1L) increments from 0000h to FFFFh and rolls over to 0000h. The TMR1 Interrupt, if enabled, is generated on overflow, which is latched in interrupt flag bit TMR1IF (PIR1<0>). This interrupt can be enabled/disabled by setting/clearing TMR1 interrupt enable bit TMR1IE (PIE1<0>).

Timer1 can operate in one of two modes:

- As a timer
- As a 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 either of the two CCP modules as the special event trigger (see Sections 8.1 and 8.2). Register 6-1 shows the Timer1 Control register.

When the Timer1 oscillator is enabled (T1OSCEN is set), the RC1/T1OSI/CCP2 and RC0/T1OSO/T1CKI pins become inputs. That is, the TRISC<1:0> value is ignored and these pins read as '0'.

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

REGISTER 6-1: T1CON: TIMER1 CONTROL REGISTER (ADDRESS 10h)

	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	—	—	T1CKPS1	T1CKPS0	T1OSCEN	T1SYN $\overline{C}$	TMR1CS	TMR1ON
	bit 7							bit 0
bit 7-6	Unimplemented: Read as '0'							
bit 5-4	T1CKPS1:T1CKPS0: Timer1 Input Clock Prescale Select bits 11 = 1:8 Prescale value 10 = 1:4 Prescale value 01 = 1:2 Prescale value 00 = 1:1 Prescale value							
bit 3	T1OSCEN: Timer1 Oscillator Enable Control bit 1 = Oscillator is enabled 0 = Oscillator is shut-off (the oscillator inverter is turned off to eliminate power drain)							
bit 2	T1SYN$\overline{C}$: 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 reset	'1' = Bit is set	'0' = Bit is cleared x = Bit is unknown

PIC16F7X

6.1 Timer1 Operation in Timer Mode

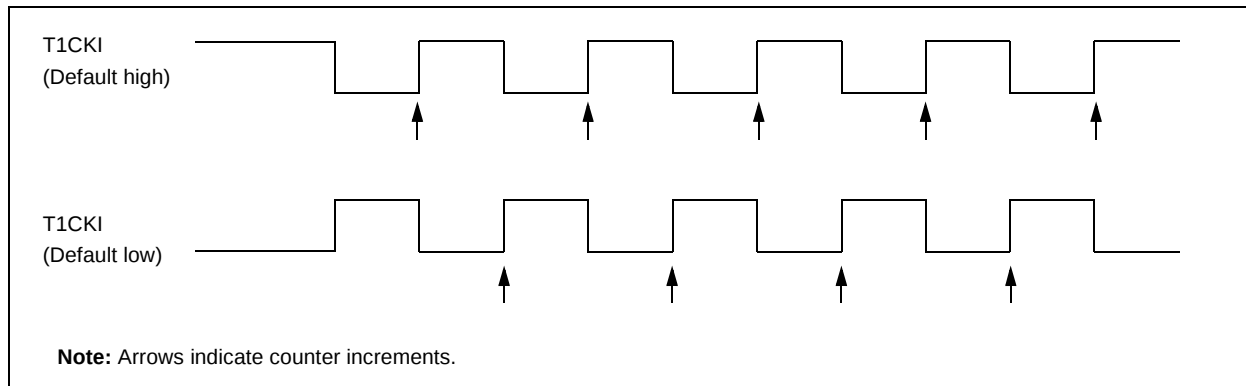
Timer mode is selected by clearing the TMR1CS (T1CON<1>) bit. In this mode, the input clock to the timer is $F_{osc}/4$. The synchronize control bit $\overline{T1SYNC}$ (T1CON<2>) has no effect, since the internal clock is always in sync.

6.2 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.

FIGURE 6-1: TIMER1 INCREMENTING EDGE



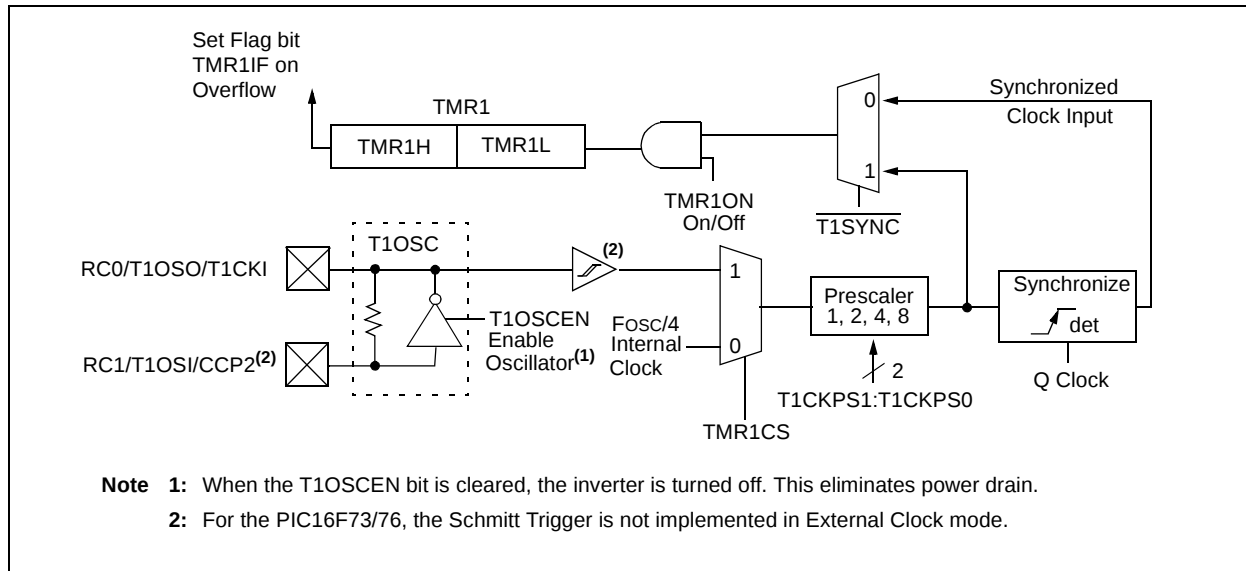
6.3 Timer1 Operation in Synchronized Counter Mode

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 T1OSCEN is set, or on pin RC0/T1OSO/T1CKI, when bit T1OSCEN is cleared.

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

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

FIGURE 6-2: TIMER1 BLOCK DIAGRAM



6.4 Timer1 Operation in Asynchronous Counter Mode

If control bit $\overline{\text{T1SYNC}}$ ($\text{T1CON}<2>$) 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, which will wake-up the processor. However, special precautions in software are needed to read/write the timer (Section 6.4.1).

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

6.4.1 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 code provided in Example 6-1 and Example 6-2 demonstrates how to write to and read Timer1 while it is running in Asynchronous mode.

EXAMPLE 6-1: 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 6-2: READING A 16-BIT FREE-RUNNING TIMER

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

PIC16F7X

6.5 Timer1 Oscillator

A crystal oscillator circuit is built-in between pins T1OSI (input) and T1OSO (amplifier output). It is enabled by setting control bit T1OSCEN (T1CON<3>). The oscillator is a low power oscillator rated up to 200 kHz. It will continue to run during SLEEP. It is primarily intended for use with a 32 kHz crystal. Table 6-1 shows the capacitor selection for the Timer1 oscillator.

The Timer1 oscillator is identical to the LP oscillator. The user must provide a software time delay to ensure proper oscillator start-up.

6.6 Resetting Timer1 using a CCP Trigger Output

If the CCP1 or CCP2 module is configured in Compare mode to generate a “special event trigger” (CCP1M3:CCP1M0 = ‘1011’), this signal will reset Timer1.

Note: The special event triggers from the CCP1 and CCP2 modules 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 or CCP2, the write will take precedence.

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

6.7 Resetting of 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 and CCP2 special event triggers.

TABLE 6-1: CAPACITOR SELECTION FOR THE TIMER1 OSCILLATOR

Osc Type	Frequency	Capacitors Used:	
		OSC1	OSC2
LP	32 kHz	47 pF	47 pF
	100 kHz	33 pF	33 pF
	200 kHz	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 (below) table for additional information.			
Commonly Used Crystals:			
32.768 kHz	Epson C-001R32.768K-A		
100 kHz	Epson C-2 100.00 KC-P		
200 kHz	STD XTL 200.000 kHz		
Note 1: Higher capacitance increases the stability of the oscillator, but also increases the start-up time.			
2: Since each resonator/crystal has its own characteristics, the user should consult the resonator/crystal manufacturer for appropriate values of external components.			

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

6.8 Timer1 Prescaler

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

TABLE 6-2: REGISTERS ASSOCIATED WITH TIMER1 AS A TIMER/COUNTER

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other RESETS
0Bh,8Bh, 10Bh,18Bh	INTCON	GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
8Ch	PIE1	PSPIE ⁽¹⁾	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	—	—	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON	--00 0000	--uu uuuu

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

Note 1: Bits PSPIE and PSPIF are reserved on the PIC16F73/76; always maintain these bits clear.

7.0 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 7-1 shows the Timer2 control register.

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

7.1 Timer2 Prescaler and Postscaler

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

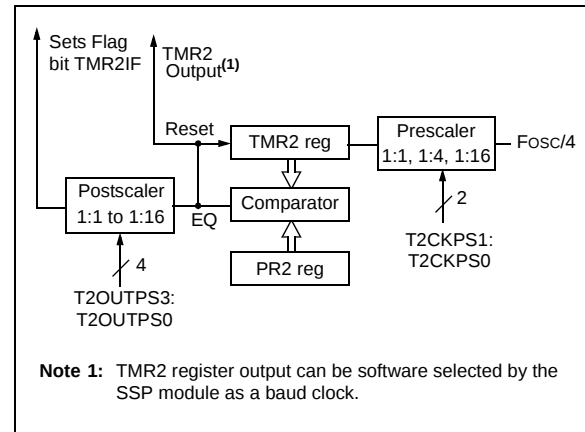
- a write to the TMR2 register
- a write to the T2CON register
- any device RESET (POR, $\overline{MCLR}$ Reset, WDT Reset or BOR)

TMR2 is not cleared when T2CON is written.

7.2 Output of TMR2

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

FIGURE 7-1: TIMER2 BLOCK DIAGRAM



PIC16F7X

REGISTER 7-1: 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 reset

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

TABLE 7-1: REGISTERS ASSOCIATED WITH TIMER2 AS A TIMER/COUNTER

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other RESETS
0Bh, 8Bh, 10Bh, 18Bh	INTCON	GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
8Ch	PIE1	PSPIE ⁽¹⁾	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 PIC16F73/76; always maintain these bits clear.

8.0 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

Both the CCP1 and CCP2 modules are identical in operation, with the exception being the operation of the special event trigger. Table 8-1 and Table 8-2 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 operates the same as CCP1, except where noted.

8.1 CCP1 Module

Capture/Compare/PWM Register1 (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.

8.2 CCP2 Module

Capture/Compare/PWM Register1 (CCPR1) is comprised of two 8-bit registers: CCPR1L (low byte) and CCPR1H (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 Modules" (DS00594).

TABLE 8-1: CCP MODE - TIMER RESOURCES REQUIRED

CCP Mode	Timer Resource
Capture	Timer1
Compare	Timer1
PWM	Timer2

TABLE 8-2: 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.

PIC16F7X

REGISTER 8-1: CCP1CON REGISTER/CCP2CON REGISTER (ADDRESS: 17h/1Dh)

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 reset

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

8.3 Capture Mode

In Capture mode, CCP1H:CCP1L 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

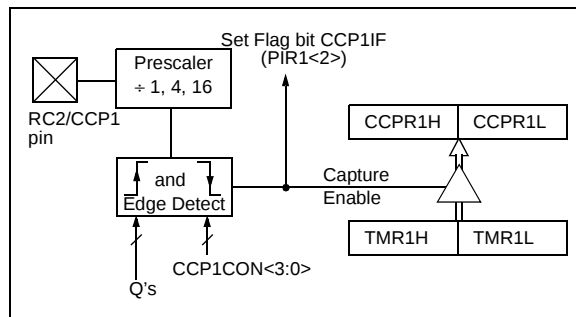
An event is selected by control bits CCP1M3:CCP1M0 (CCP1CON<3:0>). When a capture is made, the interrupt request flag bit CCP1IF (PIR1<2>) is set. The interrupt flag must be cleared in software. If another capture occurs before the value in register CCPR1 is read, the old captured value is overwritten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

8.3.1 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 8-1: CAPTURE MODE OPERATION BLOCK DIAGRAM



8.3.2 TIMER1 MODE SELECTION

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

8.3.3 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.

8.3.4 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 8-1 shows the recommended method for switching between capture prescalers. This example also clears the prescaler counter and will not generate the "false" interrupt.

EXAMPLE 8-1: CHANGING BETWEEN CAPTURE PRESCALERS

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

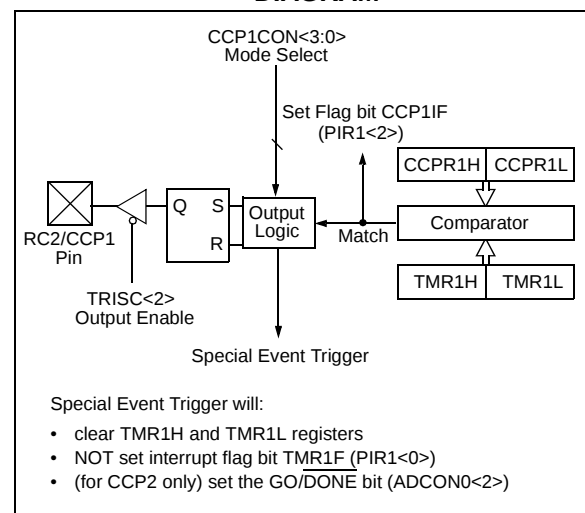
8.4 Compare Mode

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 8-2: COMPARE MODE OPERATION BLOCK DIAGRAM



PIC16F7X

8.4.1 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.

8.4.2 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.

8.4.3 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).

8.4.4 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 8-3: 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	INTE	RBIE	TMR0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
0Dh	PIR2	—	—	—	—	—	—	—	CCP2IF	---- --0	---- --0
8Ch	PIE1	PSPIE ⁽¹⁾	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
8Dh	PIE2	—	—	—	—	—	—	—	CCP2IE	---- --0	---- --0
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	—	—	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON	--00 0000	--uu uuuu
15h	CCPR1L	Capture/Compare/PWM Register1 (LSB)								xxxx xxxx	uuuu uuuu
16h	CCPR1H	Capture/Compare/PWM Register1 (MSB)								xxxx xxxx	uuuu uuuu
17h	CCP1CON	—	—	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	--00 0000
1Bh	CCPR2L	Capture/Compare/PWM Register2 (LSB)								xxxx xxxx	uuuu uuuu
1Ch	CCPR2H	Capture/Compare/PWM Register2 (MSB)								xxxx xxxx	uuuu uuuu
1Dh	CCP2CON	—	—	CCP2X	CCP2Y	CCP2M3	CCP2M2	CCP2M1	CCP2M0	--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 PIC16F73/76; always maintain these bits clear.

8.5 PWM Mode (PWM)

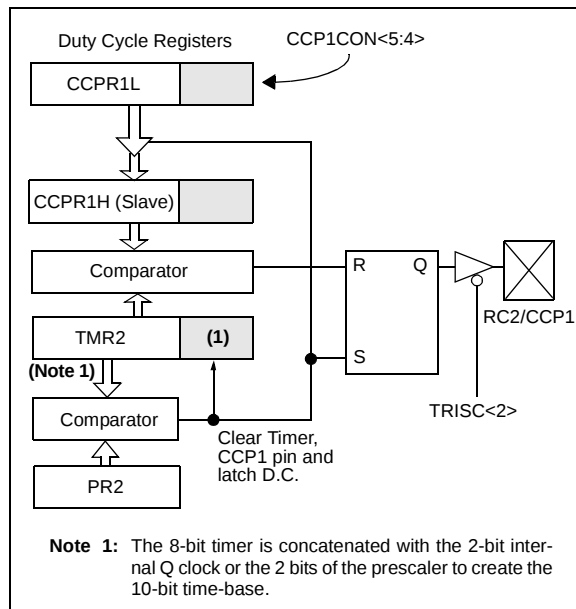
In Pulse Width Modulation mode, the CCPx pin produces up to a 10-bit resolution PWM output. Since the CCP1 pin is multiplexed with the PORTC data latch, the TRISC<2> bit must be cleared to make the CCP1 pin an output.

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 8-3 shows a simplified block diagram of the CCP module in PWM mode.

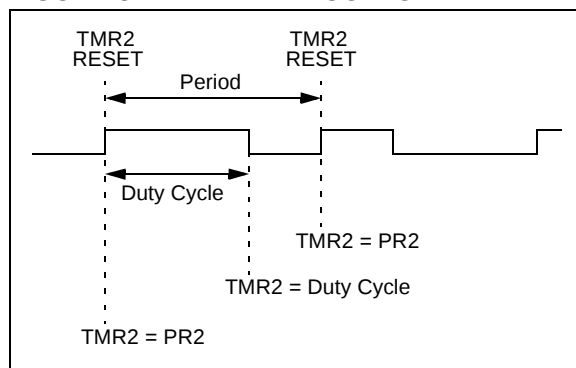
For a step-by-step procedure on how to set up the CCP module for PWM operation, see Section 8.5.3.

FIGURE 8-3: SIMPLIFIED PWM BLOCK DIAGRAM



A PWM output (Figure 8-4) 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 8-4: PWM OUTPUT



8.5.1 PWM PERIOD

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

$$\text{PWM period} = \frac{[(PR2) + 1] \cdot 4 \cdot T_{osc}}{(TMR2 \text{ prescale value})}$$

PWM frequency is defined as $1 / [\text{PWM period}]$.

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

- TMR2 is cleared
- The CCP1 pin is set (exception: if PWM duty cycle = 0%, the CCP1 pin will not be set)
- The PWM duty cycle is latched from CCPR1L into CCPR1H

Note: The Timer2 postscaler (see Section 8.3) 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.

8.5.2 PWM DUTY CYCLE

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

$$\text{PWM duty cycle} = \frac{(\text{CCPR1L:CCP1CON<5:4>}) \cdot T_{osc}}{(TMR2 \text{ prescale value})}$$

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

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

When the CCPR1H and 2-bit latch match TMR2, concatenated with an internal 2-bit Q clock or 2 bits of the TMR2 prescaler, the CCP1 pin is cleared.

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

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

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

PIC16F7X

8.5.3 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 8-4: 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	5.5

TABLE 8-5: 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	INTE	RBIE	TMR0IF	INTF	RBFIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
0Dh	PIR2	—	—	—	—	—	—	—	CCP2IF	---- ---0	---- ---0
8Ch	PIE1	PSPIE ⁽¹⁾	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
8Dh	PIE2	—	—	—	—	—	—	—	CCP2IE	---- ---0	---- ---0
87h	TRISC	PORTC Data Direction Register								1111 1111	1111 1111
11h	TMR2	Timer2 Module Register								0000 0000	0000 0000
92h	PR2	Timer2 Module Period Register								1111 1111	1111 1111
12h	T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	-000 0000
15h	CCPR1L	Capture/Compare/PWM Register1 (LSB)								xxxx xxxx	uuuu uuuu
16h	CCPR1H	Capture/Compare/PWM Register1 (MSB)								xxxx xxxx	uuuu uuuu
17h	CCP1CON	—	—	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	--00 0000
18h	CCPR2L	Capture/Compare/PWM Register2 (LSB)								xxxx xxxx	uuuu uuuu
1Ch	CCPR2H	Capture/Compare/PWM Register2 (MSB)								xxxx xxxx	uuuu uuuu
1Dh	CCP2CON	—	—	CCP2X	CCP2Y	CCP2M3	CCP2M2	CCP2M1	CCP2M0	--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 PIC16F73/76; always maintain these bits clear.

9.0 SYNCHRONOUS SERIAL PORT (SSP) MODULE

9.1 SSP Module Overview

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

- Serial Peripheral Interface (SPI)
- Inter-Integrated Circuit (I²C)

An overview of I²C operations and additional information on the SSP module can be found in the PICmicro™ Mid-Range MCU Family Reference Manual (DS33023).

Refer to Application Note AN578, "Use of the SSP Module in the I²C Multi-Master Environment" (DS00578).

9.2 SPI Mode

This section contains register definitions and operational characteristics of the SPI module. Additional information on the SPI module can be found in the PICmicro™ Mid-Range MCU Family Reference Manual (DS33023A).

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

- Serial Data Out (SDO) 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/ $\overline{SS}$ /AN4

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

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

PIC16F7X

REGISTER 9-1: SSPSTAT: SYNC SERIAL PORT STATUS REGISTER (ADDRESS 94h)

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

- bit 7 **SMP:** SPI Data Input Sample Phase bit
SPI Master mode:
 1 = Input data sampled at end of data output time
 0 = Input data sampled at middle of data output time (Microwire®)
SPI Slave mode:
 SMP must be cleared when SPI is used in Slave mode
I²C mode:
 This bit must be maintained clear
- bit 6 **CKE:** SPI Clock Edge Select bit (Figure 9-2, Figure 9-3, and Figure 9-4)
SPI mode, CKP = 0:
 1 = Data transmitted on rising edge of SCK (Microwire® alternate)
 0 = Data transmitted on falling edge of SCK
SPI mode, CKP = 1:
 1 = Data transmitted on falling edge of SCK (Microwire® default)
 0 = Data transmitted on rising edge of SCK
I²C mode:
 This bit must be maintained clear
- bit 5 **D $\overline{A}$:** Data/Address bit (I²C mode only)
 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 (I²C mode only)
 This bit is cleared when the SSP module is disabled, or when the START bit is detected last. SSPEN is cleared.
 1 = Indicates that a STOP bit has been detected last (this bit is '0' on RESET)
 0 = STOP bit was not detected last
- bit 3 **S:** START bit (I²C mode only)
 This bit is cleared when the SSP module is disabled, or when the STOP bit is detected last. SSPEN is cleared.
 1 = Indicates that a START bit has been detected last (this bit is '0' on RESET)
 0 = START bit was not detected last
- bit 2 **R $\overline{W}$:** Read/Write bit Information (I²C mode only)
 This bit holds the R/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 ACK bit.
 1 = Read
 0 = Write
- bit 1 **UA:** Update Address bit (10-bit I²C mode only)
 1 = Indicates that the user needs to update the address in the SSPADD register
 0 = Address does not need to be updated
- bit 0 **BF:** Buffer Full Status bit
Receive (SPI and I²C modes):
 1 = Receive complete, SSPBUF is full
 0 = Receive not complete, SSPBUF is empty
Transmit (I²C mode only):
 1 = Transmit in progress, SSPBUF is full
 0 = Transmit complete, SSPBUF is empty

Legend:

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

REGISTER 9-2: SSPCON: SYNC SERIAL PORT CONTROL REGISTER (ADDRESS 14h)

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

- bit 7 **WCOL**: Write Collision Detect bit
 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
In SPI mode:
 1 = A new byte is received while the SSPBUF register is still holding the previous data. In case of overflow, the data in SSPSR is lost. Overflow can only occur in Slave mode. The user must read the SSPBUF, even if only transmitting data, to avoid setting overflow. In Master mode, the overflow bit is not set since each new reception (and transmission) is initiated by writing to the SSPBUF register.
 0 = No overflow
In I²C mode:
 1 = A byte is received while the SSPBUF register is still holding the previous byte. SSPOV is a "don't care" in Transmit mode. SSPOV must be cleared in software in either mode.
 0 = No overflow
- bit 5 **SSPEN**: Synchronous Serial Port Enable bit
In SPI mode:
 1 = Enables serial port and configures SCK, SDO, and SDI as serial port pins
 0 = Disables serial port and configures these pins as I/O port pins
In I²C mode:
 1 = Enables the serial port and configures the SDA and SCL pins as serial port pins
 0 = Disables serial port and configures these pins as I/O port pins
 In both modes, when enabled, these pins must be properly configured as input or output.
- bit 4 **CKP**: Clock Polarity Select bit
In SPI mode:
 1 = IDLE state for clock is a high level (Microwire[®] default)
 0 = IDLE state for clock is a low level (Microwire[®] alternate)
In I²C mode:
 SCK release control
 1 = Enable clock
 0 = Holds clock low (clock stretch). (Used to ensure data setup time.)
- bit 3-0 **SSPM3:SSPM0**: Synchronous Serial Port Mode Select bits
 0000 = SPI Master mode, clock = Fosc/4
 0001 = SPI Master mode, clock = Fosc/16
 0010 = SPI Master mode, clock = Fosc/64
 0011 = SPI Master mode, clock = TMR2 output/2
 0100 = SPI Slave mode, clock = SCK pin. $\overline{SS}$ pin control enabled.
 0101 = SPI Slave mode, clock = SCK pin. $\overline{SS}$ pin control disabled. $\overline{SS}$ can be used as I/O pin.
 0110 = I²C Slave mode, 7-bit address
 0111 = I²C Slave mode, 10-bit address
 1011 = I²C Firmware Controlled Master mode (slave IDLE)
 1110 = I²C Slave mode, 7-bit address with START and STOP bit interrupts enabled
 1111 = I²C Slave mode, 10-bit address with START and STOP bit interrupts enabled

Legend:

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

[illegible]

- SDI must have TRISC<4> set
- SDO must have TRISC<5> cleared
- SCK (Master mode) must have TRISC<3> cleared
- SCK (Slave mode) must have TRISC<3> set
- $\overline{SS}$ must have TRISA<5> set and ADCON must be configured such that RA5 is a digital I/O

- Note 1:** When the SPI is in Slave mode with SS pin control enabled ($\text{SSPCON}\langle 3:0 \rangle = 0100$), the SPI module will reset if the SS pin is set to VDD.
- 2:** If the SPI is used in Slave mode with $\text{CKE} = '1'$, then the SS pin control must be enabled.
- 3:** When the SPI is in Slave mode with $\overline{\text{SS}}$ pin control enabled ($\text{SSPCON}\langle 3:0 \rangle = '0100'$), the state of the $\overline{\text{SS}}$ pin can affect the state read back from the $\text{TRISC}\langle 5 \rangle$ bit. The Peripheral OE signal from the SSP module into PORTC controls the state that is read back from the $\text{TRISC}\langle 5 \rangle$ bit (see Section 4.3 for information on PORTC). If Read-Modify-Write instructions, such as BSF are performed on the TRISC register while the $\overline{\text{SS}}$ pin is high, this will cause the $\text{TRISC}\langle 5 \rangle$ bit to be set, thus disabling the SDO output.

FIGURE 9-2: SPI MODE TIMING, MASTER MODE

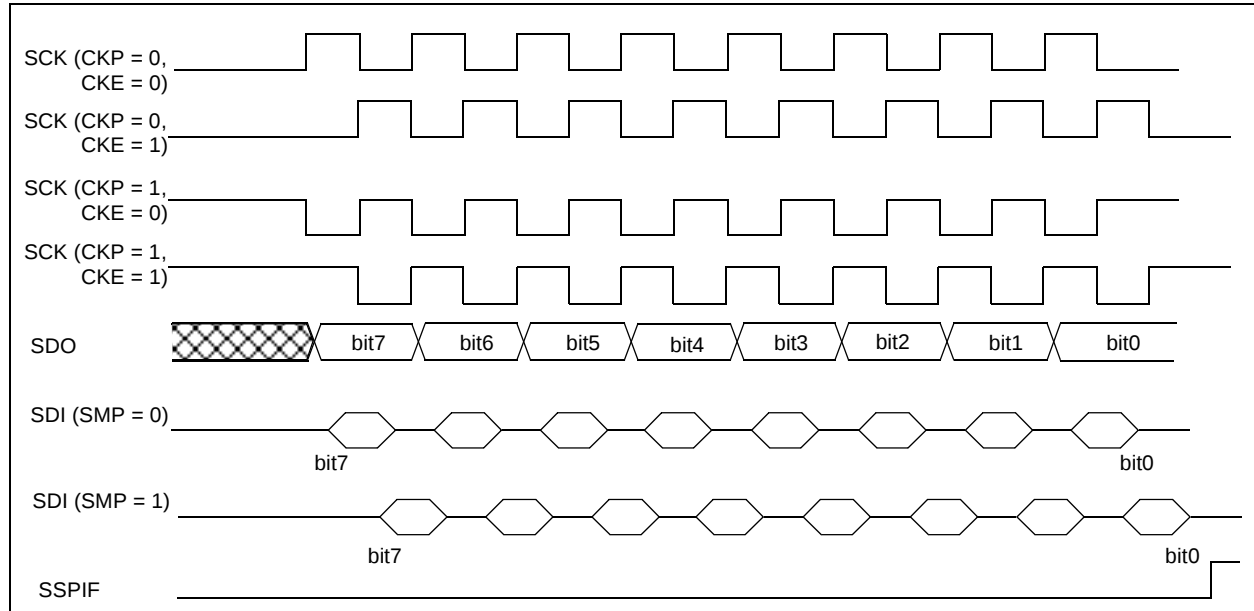


FIGURE 9-3: SPI MODE TIMING (SLAVE MODE WITH CKE = 0)

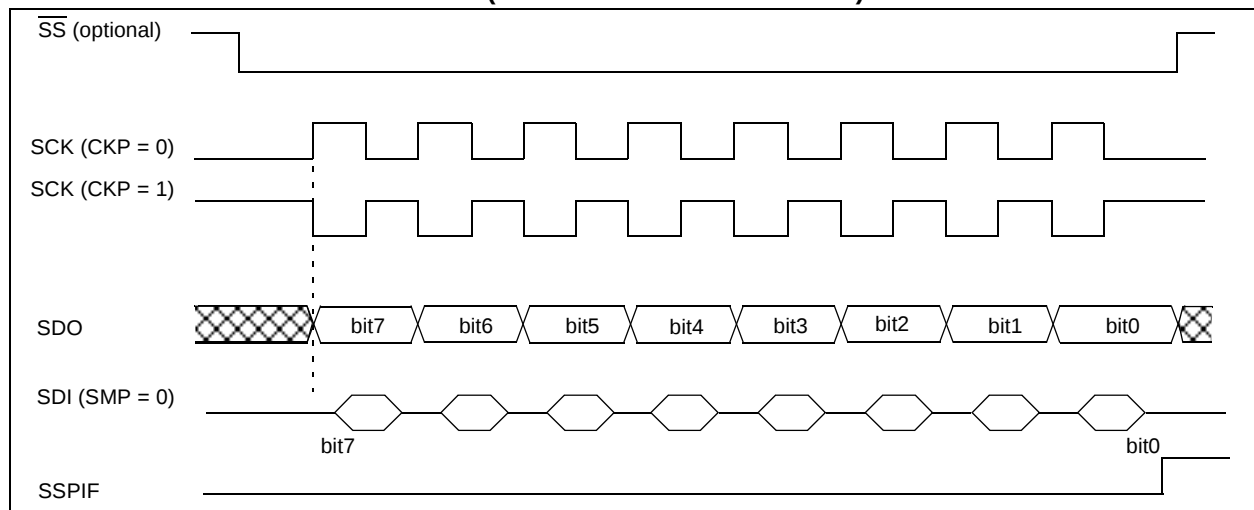
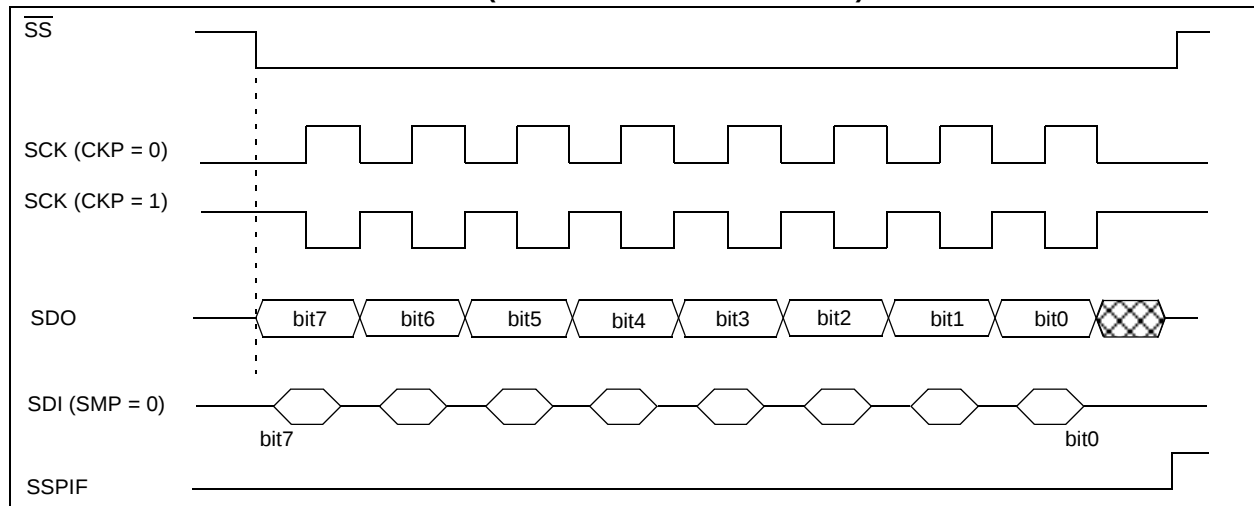


FIGURE 9-4: SPI MODE TIMING (SLAVE MODE WITH CKE = 1)



PIC16F7X

TABLE 9-1: REGISTERS ASSOCIATED WITH SPI OPERATION

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other RESETS
0Bh,8Bh. 10Bh,18Bh	INTCON	GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
8Ch	PIE1	PSPIE ⁽¹⁾	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
87h	TRISC	PORTC Data Direction Register								1111 1111	1111 1111
13h	SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register								xxxx xxxx	uuuu uuuu
14h	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	0000 0000
85h	TRISA	—	—	PORTA Data Direction Register						--11 1111	--11 1111
94h	SSPSTAT	SMP	CKE	D/ $\overline{A}$	P	S	R/ $\overline{W}$	UA	BF	0000 0000	0000 0000

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

Note 1: Bits PSPIE and PSPIF are reserved on the PIC16F73/76; always maintain these bits clear.

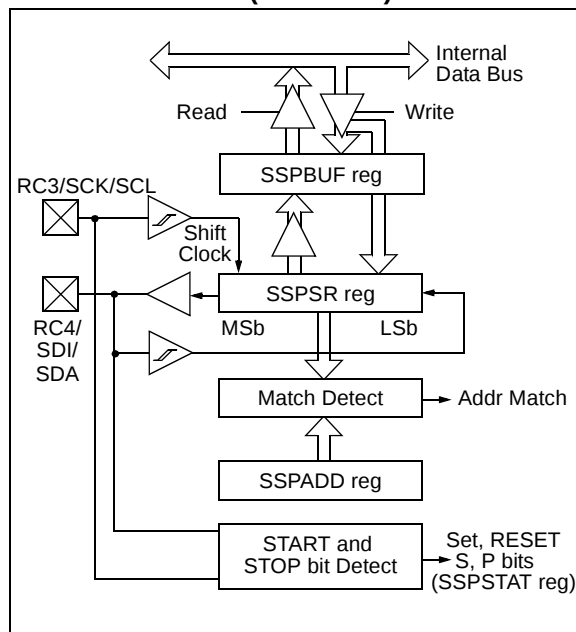
9.3 SSP I²C Operation

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

Two pins are used for data transfer. These are the RC3/SCK/SCL pin, which is the clock (SCL), and the RC4/SDI/SDA pin, which is the data (SDA). The user must configure these pins as inputs or outputs through the TRISC<4:3> bits.

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

FIGURE 9-5: SSP BLOCK DIAGRAM (I²C MODE)



The SSP module has five registers for I²C operation. These are the:

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

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

- I²C Slave mode (7-bit address)
- I²C Slave mode (10-bit address)
- I²C Slave mode (7-bit address), with START and STOP bit interrupts enabled to support Firmware Master mode
- I²C Slave mode (10-bit address), with START and STOP bit interrupts enabled to support Firmware Master mode
- I²C START and STOP bit interrupts enabled to support Firmware Master mode, Slave is IDLE

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

Additional information on SSP I²C operation can be found in the PICmicro™ Mid-Range MCU Family Reference Manual (DS33023A).

9.3.1 SLAVE MODE

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

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

There are certain conditions that will cause the SSP module not to give this $\overline{ACK}$ pulse. They include (either or both):

- 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. Table 9-2 shows what happens when a data transfer byte is received, given the status of bits BF and SSPOV. The shaded cells show the condition where user software did not properly clear the overflow condition. Flag bit BF is cleared by reading the SSPBUF register, while bit SSPOV is cleared through software.

The SCL clock input must have a minimum high and low for proper operation. The high and low times of the I²C specification, as well as the requirements of the SSP module, are shown in timing parameter #100 and parameter #101.

PIC16F7X

9.3.1.1 Addressing

Once the SSP 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:

- The SSPSR register value is loaded into the SSPBUF register.
- The buffer full bit, BF is set.
- An $\overline{\text{ACK}}$ pulse is generated.
- SSP interrupt flag bit, SSPIF (PIR1<3>) is set (interrupt is generated if enabled) - on the falling edge of the ninth SCL pulse.

In 10-bit Address mode, two address bytes need to be received by the slave (Figure 9-7). The five Most Significant bits (MSBs) of the first address byte specify if this is a 10-bit address. Bit $\overline{\text{R/W}}$ (SSPSTAT<2>) must specify a write so the slave device will receive the second address byte. For a 10-bit address, the first byte would equal '1111 0 A9 A8 0', where A9 and A8 are the two MSBs of the address.

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

- Receive first (high) byte of address (bits SSPIF, BF, and bit UA (SSPSTAT<1>) are set).
- Update the SSPADD register with second (low) byte of address (clears bit UA and releases the SCL line).
- Read the SSPBUF register (clears bit BF) and clear flag bit SSPIF.
- Receive second (low) byte of address (bits SSPIF, BF, and UA are set).
- Update the SSPADD register with the first (high) byte of address, if match releases SCL line, this will clear bit UA.
- Read the SSPBUF register (clears bit BF) and clear flag bit SSPIF.
- Receive Repeated START condition.
- Receive first (high) byte of address (bits SSPIF and BF are set).
- Read the SSPBUF register (clears bit BF) and clear flag bit SSPIF.

TABLE 9-2: DATA TRANSFER RECEIVED BYTE ACTIONS

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

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

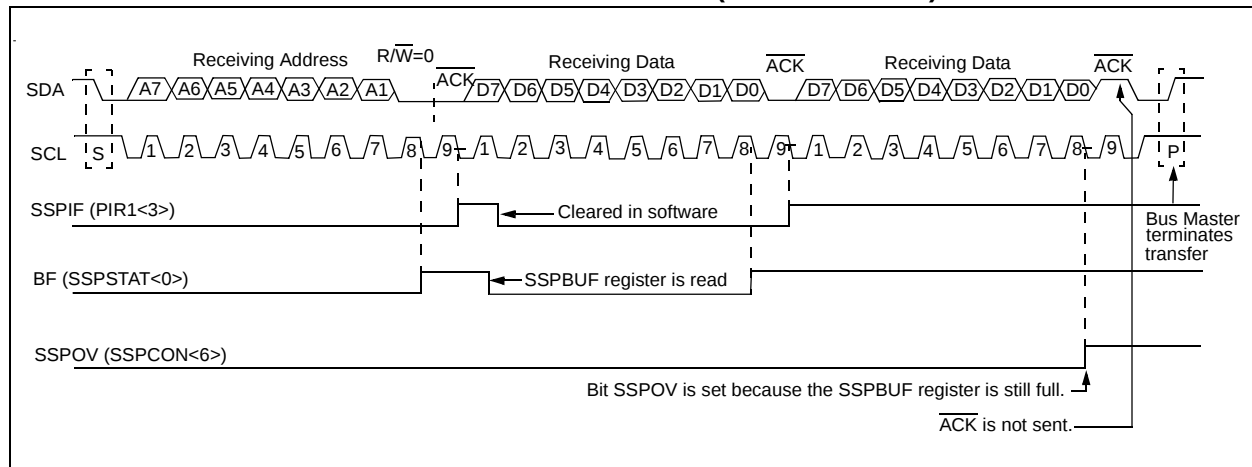
9.3.1.2 Reception

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

When the address byte overflow condition exists, then no Acknowledge (ACK) pulse is given. An overflow condition is defined as either bit BF (SSPSTAT<0>) is set, or bit SSPOV (SSPCON<6>) is set. This is an error condition due to the user's firmware.

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

FIGURE 9-6: I²C WAVEFORMS FOR RECEPTION (7-BIT ADDRESS)



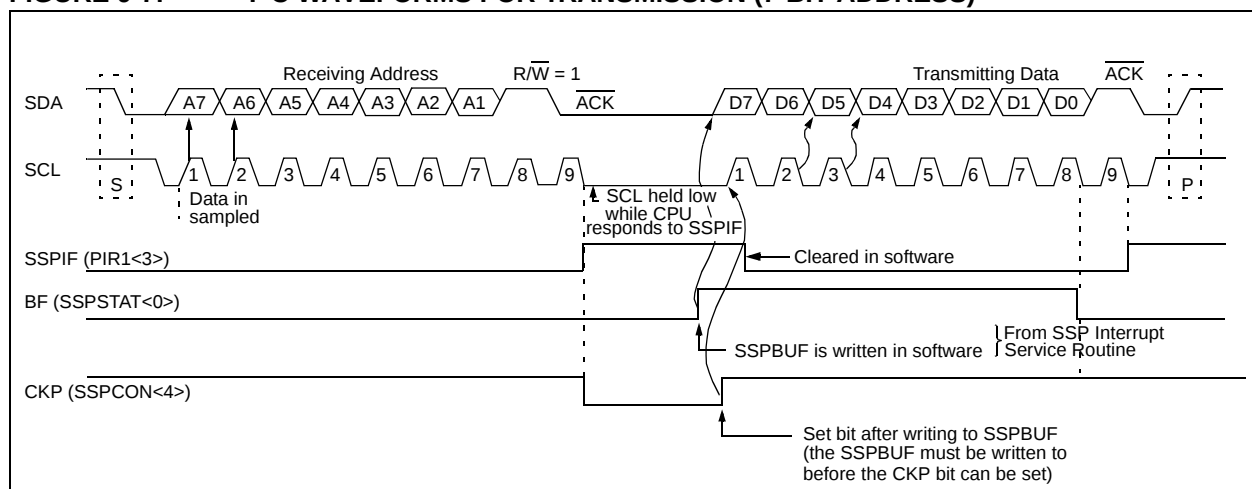
9.3.1.3 Transmission

When the R/W bit of the incoming address byte is set and an address match occurs, the R/W bit of the SSPSTAT register is set. The received address is loaded into the SSPBUF register. The ACK pulse will be sent on the ninth bit, and pin RC3/SCK/SCL is held low. 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 master must monitor the SCL pin prior to asserting another clock pulse. The slave devices may be holding off the master by stretching the clock. The eight data bits are shifted out on the falling edge of the SCL input. This ensures that the SDA signal is valid during the SCL high time (Figure 9-7).

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

As a slave-transmitter, the ACK pulse from the master-receiver is latched on the rising edge of the ninth SCL input pulse. If the SDA line was high (not ACK), then the data transfer is complete. When the ACK is latched by the slave, the slave logic is reset (resets SSPSTAT register) and the slave then monitors for another occurrence of the START bit. If the SDA line was low (ACK), the transmit data must be loaded into the SSPBUF register, which also loads the SSPSR register. Then pin RC3/SCK/SCL should be enabled by setting bit CKP.

FIGURE 9-7: I²C WAVEFORMS FOR TRANSMISSION (7-BIT ADDRESS)



PIC16F7X

9.3.2 MASTER MODE

Master mode of operation is supported in firmware using interrupt generation on the detection of the START and STOP conditions. The STOP (P) and START (S) bits are cleared from a RESET or when the SSP module is disabled. The STOP (P) and START (S) bits will toggle based on the START and STOP conditions. Control of the I²C bus may be taken when the P bit is set, or the bus is IDLE and both the S and P bits are clear.

In Master mode, the SCL and SDA lines are manipulated by clearing the corresponding TRISC<4:3> bit(s). The output level is always low, irrespective of the value(s) in PORTC<4:3>. So when transmitting data, a '1' data bit must have the TRISC<4> bit set (input) and a '0' data bit must have the TRISC<4> bit cleared (output). The same scenario is true for the SCL line with the TRISC<3> bit. Pull-up resistors must be provided externally to the SCL and SDA pins for proper operation of the I²C module.

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

- START condition
- STOP condition
- Data transfer byte transmitted/received

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

9.3.3 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 SSP module is disabled. The STOP (P) and START (S) bits will toggle based on the START and STOP conditions. Control of the I²C bus may be taken when bit P (SSPSTAT<4>) is set, or the bus is IDLE and both the S and P bits clear. When the bus is busy, enabling the SSP Interrupt will generate the interrupt when the STOP condition occurs.

In Multi-Master operation, the SDA line must be monitored to see if the signal level is the expected output level. This check only needs to be done when a high level is output. If a high level is expected and a low level is present, the device needs to release the SDA and SCL lines (set TRISC<4:3>). There are two stages where this arbitration can be lost, these are:

- Address Transfer
- Data Transfer

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

TABLE 9-3: REGISTERS ASSOCIATED WITH I²C OPERATION

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other RESETS
0Bh, 8Bh, 10Bh, 18Bh	INTCON	GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
8Ch	PIE1	PSPIE ⁽¹⁾	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
13h	SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register								xxxx xxxx	uuuu uuuu
93h	SSPAD	Synchronous Serial Port (I ² C mode) Address Register								0000 0000	0000 0000
14h	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	0000 0000
94h	SSPSTAT	SMP ⁽²⁾	CKE ⁽²⁾	D/A	P	S	R/W	UA	BF	0000 0000	0000 0000
87h	TRISC	PORTC Data Direction Register								1111 1111	1111 1111

Legend: x = unknown, u = unchanged, - = unimplemented locations read as '0'. Shaded cells are not used by SSP module in I²C mode.

Note 1: PSPIF and PSPIE are reserved on the PIC16F73/76; always maintain these bits clear.

2: Maintain these bits clear in I²C mode.

10.0 UNIVERSAL SYNCHRONOUS ASYNCHRONOUS RECEIVER TRANSMITTER (USART)

The Universal Synchronous Asynchronous Receiver Transmitter (USART) module is one of the two serial I/O modules. (USART is also known as a Serial Communications Interface or SCI.) The USART 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 USART 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.

REGISTER 10-1: 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 <u>Asynchronous mode:</u> Don't care <u>Synchronous mode:</u> 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: USART Mode Select bit 1 = Synchronous mode 0 = Asynchronous mode							
bit 3	Unimplemented: Read as '0'							
bit 2	BRGH: High Baud Rate Select bit <u>Asynchronous mode:</u> 1 = High speed 0 = Low speed <u>Synchronous mode:</u> 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 reset	'1' = Bit is set	'0' = Bit is cleared x = Bit is unknown

PIC16F7X

REGISTER 10-2: RCSTA: RECEIVE STATUS AND CONTROL REGISTER (ADDRESS 18h)

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R-0	R-0	R-x
SPEN	RX9	SREN	CREN	—	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 **Unimplemented:** Read as '0'
- bit 2 **FERR:** Framing Error bit
 1 = Framing error (can be updated by reading RCREG register and receive next valid byte)
 0 = No framing error
- bit 1 **OERR:** Overrun Error bit
 1 = Overrun error (can be cleared by clearing bit CREN)
 0 = No overrun error
- bit 0 **RX9D:** 9th bit of Received Data
 Can be parity bit (parity to be calculated by firmware)

Legend:

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

10.1 USART Baud Rate Generator (BRG)

The BRG supports both the Asynchronous and Synchronous modes of the USART. 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 10-1 shows the formula for computation of the baud rate for different USART 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 10-1. 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.

10.1.1 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 10-1: 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

X = value in SPBRG (0 to 255)

TABLE 10-2: 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	—	FERR	OERR	RX9D	0000 -00x	0000 -00x
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.

PIC16F7X

TABLE 10-3: BAUD RATES FOR ASYNCHRONOUS MODE (BRGH = 0)

BAUD RATE	Fosc = 20 MHz			Fosc = 16 MHz			Fosc = 10 MHz		
	BAUD	% ERROR	SPBRG VALUE (DECIMAL)	BAUD	% ERROR	SPBRG VALUE (DECIMAL)	BAUD	% ERROR	SPBRG VALUE (DECIMAL)
1200	1,221	1.73%	255	1,202	0.16%	207	1,202	0.16%	129
2400	2,404	0.16%	129	2,404	0.16%	103	2,404	0.16%	64
9600	9,470	-1.36%	32	9,615	0.16%	25	9,766	1.73%	15
19,200	19,531	1.73%	15	19,231	0.16%	12	19,531	1.73%	7
38,400	39,063	1.73%	7	35,714	-6.99%	6	39,063	1.73%	3
57,600	62,500	8.51%	4	62,500	8.51%	3	52,083	-9.58%	2
76,800	78,125	1.73%	3	83,333	8.51%	2	78,125	1.73%	1
96,000	104,167	8.51%	2	83,333	-13.19%	2	78,125	-18.62%	1
115,200	104,167	-9.58%	2	125,000	8.51%	1	78,125	-32.18%	1
250,000	312,500	25.00%	0	250,000	0.00%	0	156,250	-37.50%	0

BAUD RATE	Fosc = 4 MHz			Fosc = 3.6864 MHz			Fosc = 3.579545 MHz		
	BAUD	% ERROR	SPBRG VALUE (DECIMAL)	BAUD	% ERROR	SPBRG VALUE (DECIMAL)	BAUD	% ERROR	SPBRG VALUE (DECIMAL)
300	300	0.16%	207	300	0.00%	191	301	0.23%	185
1200	1,202	0.16%	51	1,200	0.00%	47	1,190	-0.83%	46
2400	2,404	0.16%	25	2,400	0.00%	23	2,432	1.32%	22
9600	8,929	-6.99%	6	9,600	0.00%	5	9,322	-2.90%	5
19,200	20,833	8.51%	2	19,200	0.00%	2	18,643	-2.90%	2
38,400	31,250	-18.62%	1	28,800	-25.00%	1	27,965	-27.17%	1
57,600	62,500	8.51%	0	57,600	0.00%	0	55,930	-2.90%	0
76,800	62,500	-18.62%	0	—	—	—	—	—	—

TABLE 10-4: BAUD RATES FOR ASYNCHRONOUS MODE (BRGH = 1)

BAUD RATE	Fosc = 20 MHz			Fosc = 16 MHz			Fosc = 10 MHz		
	BAUD	% ERROR	SPBRG VALUE (DECIMAL)	BAUD	% ERROR	SPBRG VALUE (DECIMAL)	BAUD	% ERROR	SPBRG VALUE (DECIMAL)
2400	—	—	—	—	—	—	2,441	1.73%	255
9600	9,615	0.16%	129	9,615	0.16%	103	9,615	0.16%	64
19,200	19,231	0.16%	64	19,231	0.16%	51	18,939	-1.36%	32
38,400	37,879	-1.36%	32	38,462	0.16%	25	39,063	1.73%	15
57,600	56,818	-1.36%	21	58,824	2.12%	16	56,818	-1.36%	10
76,800	78,125	1.73%	15	76,923	0.16%	12	78,125	1.73%	7
96,000	96,154	0.16%	12	100,000	4.17%	9	89,286	-6.99%	6
115,200	113,636	-1.36%	10	111,111	-3.55%	8	125,000	8.51%	4
250,000	250,000	0.00%	4	250,000	0.00%	3	208,333	-16.67%	2
300,000	312,500	4.17%	3	333,333	11.11%	2	312,500	4.17%	1

BAUD RATE (K)	Fosc = 4 MHz			Fosc = 3.6864 MHz			Fosc = 3.579545 MHz		
	BAUD	% ERROR	SPBRG VALUE (DECIMAL)	BAUD	% ERROR	SPBRG VALUE (DECIMAL)	BAUD	% ERROR	SPBRG VALUE (DECIMAL)
1200	1,202	0.16%	207	1,200	0.00%	191	1,203	0.23%	185
2400	2,404	0.16%	103	2,400	0.00%	95	2,406	0.23%	92
9600	9,615	0.16%	25	9,600	0.00%	23	9,727	1.32%	22
19,200	19,231	0.16%	12	19,200	0.00%	11	18,643	-2.90%	11
38,400	35,714	-6.99%	6	38,400	0.00%	5	37,287	-2.90%	5
57,600	62,500	8.51%	3	57,600	0.00%	3	55,930	-2.90%	3
76,800	83,333	8.51%	2	76,800	0.00%	2	74,574	-2.90%	2
96,000	83,333	-13.19%	2	115,200	20.00%	1	111,861	16.52%	1
115,200	125,000	8.51%	1	115,200	0.00%	1	111,861	-2.90%	1
250,000	250,000	0.00%	0	230,400	-7.84%	0	223,722	-10.51%	0

10.2 USART Asynchronous Mode

In this mode, the USART 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 USART transmits and receives the LSb first. The USART's 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 USART Asynchronous module consists of the following important elements:

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

10.2.1 USART ASYNCHRONOUS TRANSMITTER

The USART transmitter block diagram is shown in Figure 10-1. The heart of the transmitter is the transmit (serial) shift register (TSR). The shift register obtains its data from the read/write transmit buffer, TXREG. The TXREG register is loaded with data by firmware. 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, the TXREG register is empty. One instruction cycle later, flag bit TXIF (PIR1<4>) and flag bit TRMT (TXSTA<1>)

are set. The TXIF 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 one instruction cycle after the TSR register becomes empty, and is cleared one instruction cycle after the TSR register is loaded. 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.

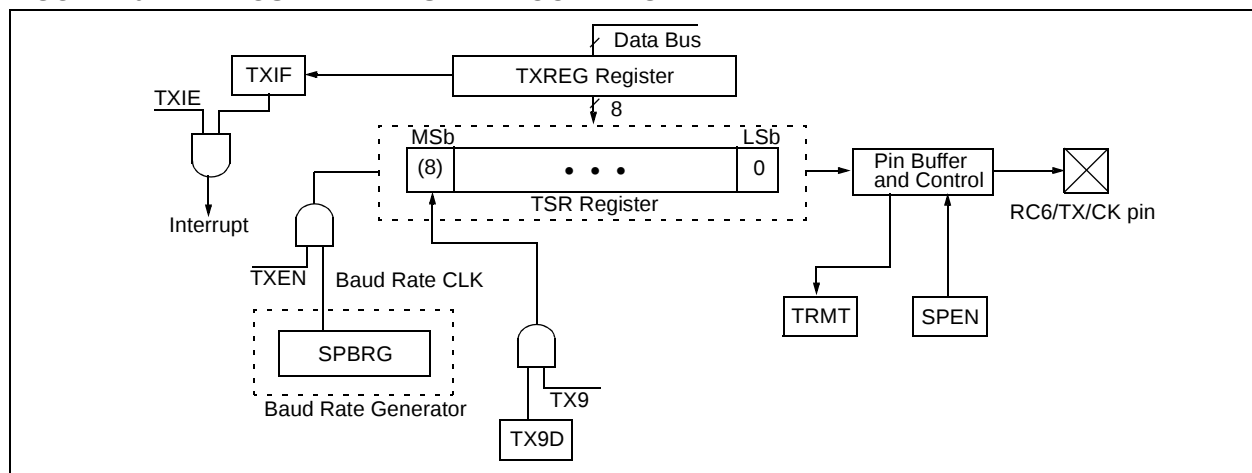
Note 1: The TSR register is not mapped in data memory, so it is not available to the user.

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 10-2). 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 10-3). 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 hi-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 10-1: USART TRANSMIT BLOCK DIAGRAM



PIC16F7X

Steps to follow when setting up an Asynchronous Transmission:

1. Initialize the SPBRG register for the appropriate baud rate. If a high speed baud rate is desired, set bit BRGH (Section 10.1).
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 in the INTCON register are set.

FIGURE 10-2: ASYNCHRONOUS MASTER TRANSMISSION

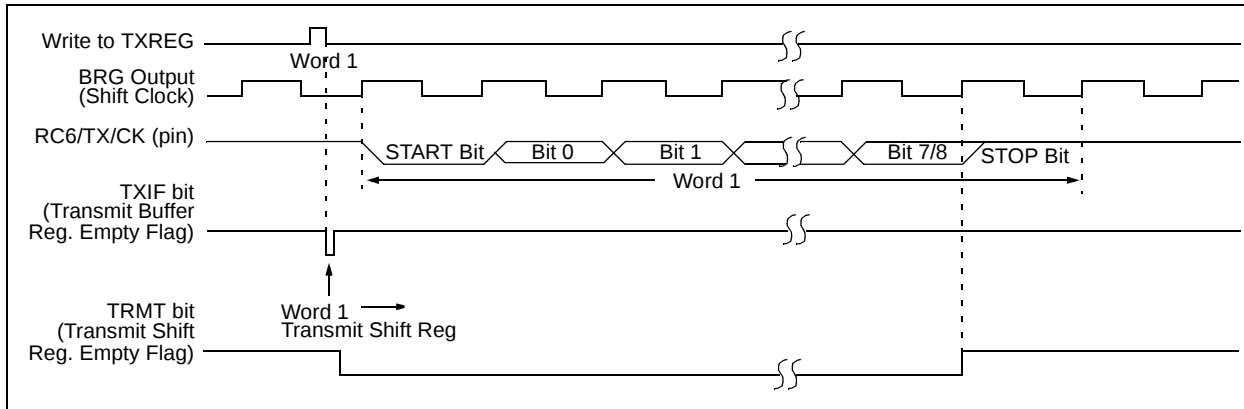


FIGURE 10-3: ASYNCHRONOUS MASTER TRANSMISSION (BACK TO BACK)

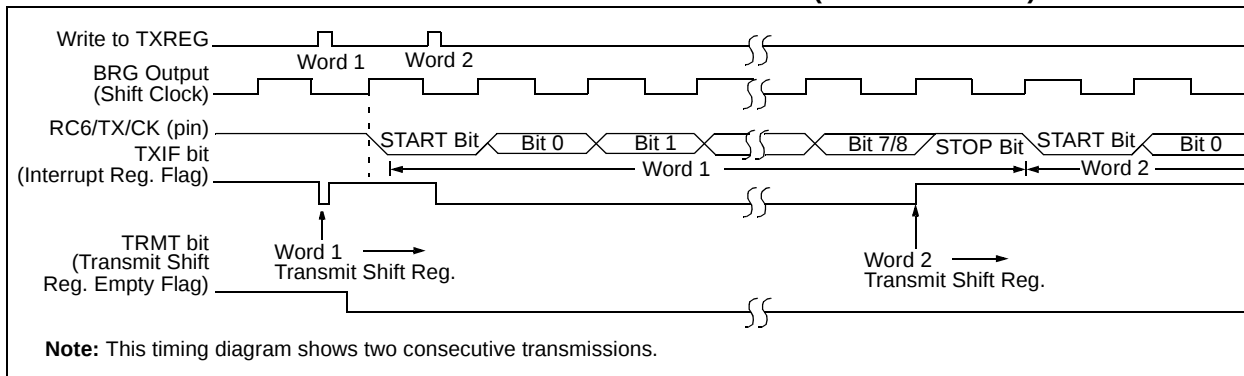


TABLE 10-5: REGISTERS ASSOCIATED WITH ASYNCHRONOUS TRANSMISSION

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	INTE	RBIE	TMR0IF	INTF	RBI	0000 000x	0000 000u
0Ch	PIR1	PSPIE ⁽¹⁾	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
18h	RCSTA	SPEN	RX9	SREN	CREN	—	FERR	OERR	RX9D	0000 -00x	0000 -00x
19h	TXREG	USART Transmit Register								0000 0000	0000 0000
8Ch	PIE1	PSPIE ⁽¹⁾	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 SSPIF are reserved on the PIC16F73/76; always maintain these bits clear.

10.2.2 USART ASYNCHRONOUS RECEIVER

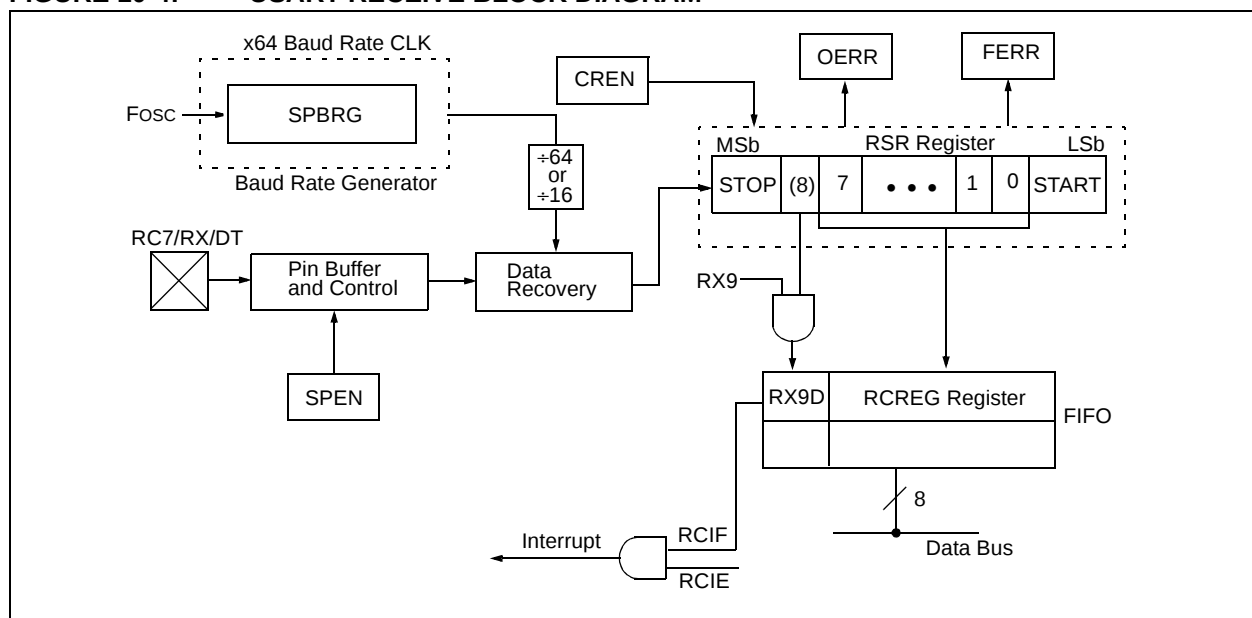
The receiver block diagram is shown in Figure 10-4. The data is received on the RC7/RX/DT pin and drives the data recovery block. The data recovery block is actually a high speed shifter operating at x16 times the baud rate, whereas the main receive serial shifter operates at the bit rate, or at Fosc.

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

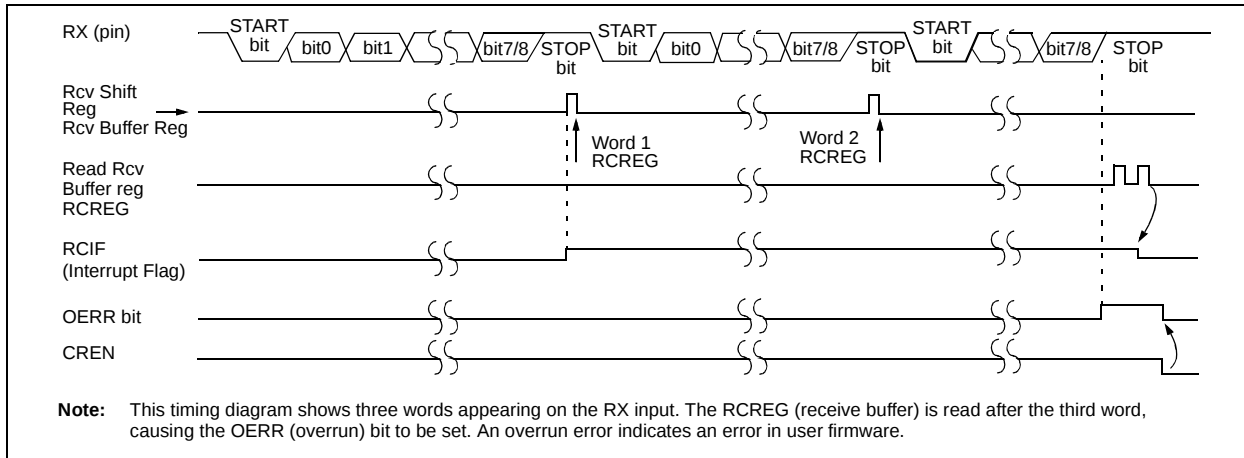
is possible for two bytes of data to be received and transferred to the RCREG FIFO and a third byte to begin shifting to the RSR register. On the detection of the STOP bit of the third byte, if the RCREG register is still full, the overrun error bit OERR (RCSTA<1>) will be set. The word in the RSR will 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, therefore, it is 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 RCREG register, in order not to lose the old FERR and RX9D information.

FIGURE 10-4: USART RECEIVE BLOCK DIAGRAM



PIC16F7X

FIGURE 10-5: ASYNCHRONOUS RECEPTION



Steps to follow when setting up an Asynchronous Reception:

1. Initialize the SPBRG register for the appropriate baud rate. If a high speed baud rate is desired, set bit BRGH (Section 10.1).
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 in the INTCON register are set.

TABLE 10-6: 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	INTE	RBIE	TMR0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
18h	RCSTA	SPEN	RX9	SREN	CREN	—	FERR	OERR	RX9D	0000 -00x	0000 -00x
1Ah	RCREG	USART Receive Register								0000 0000	0000 0000
8Ch	PIE1	PSPIE ⁽¹⁾	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 the PIC16F73/76 devices; always maintain these bits clear.

10.3 USART 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>).

10.3.1 USART SYNCHRONOUS MASTER TRANSMISSION

The USART transmitter block diagram is shown in Figure 10-1. 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 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 Tcycle), 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 10-6). The transmission can also be started by first loading the TXREG register and then setting bit TXEN (Figure 10-7). 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 hi-impedance. If either bit CREN or bit SREN is set during a transmission, the transmission is aborted and the DT pin reverts to a hi-impedance state (for a reception). The CK pin will remain an output if bit CSRC is set (internal clock). The transmitter logic, however, is not reset, although it is disconnected from the pins. In order to reset the transmitter, the user has to clear bit TXEN. If bit SREN is set (to interrupt an on-going transmission and receive a single word), then after the single word is received, bit SREN will be cleared and the serial port will revert back to transmitting, since bit TXEN is still set. The DT line will immediately switch from Hi-impedance Receive mode to transmit and start driving. To avoid this, bit TXEN should be cleared.

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

Steps to follow when setting up a Synchronous Master Transmission:

1. Initialize the SPBRG register for the appropriate baud rate (Section 10.1).
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 in the INTCON register are set.

PIC16F7X

FIGURE 10-6: SYNCHRONOUS TRANSMISSION

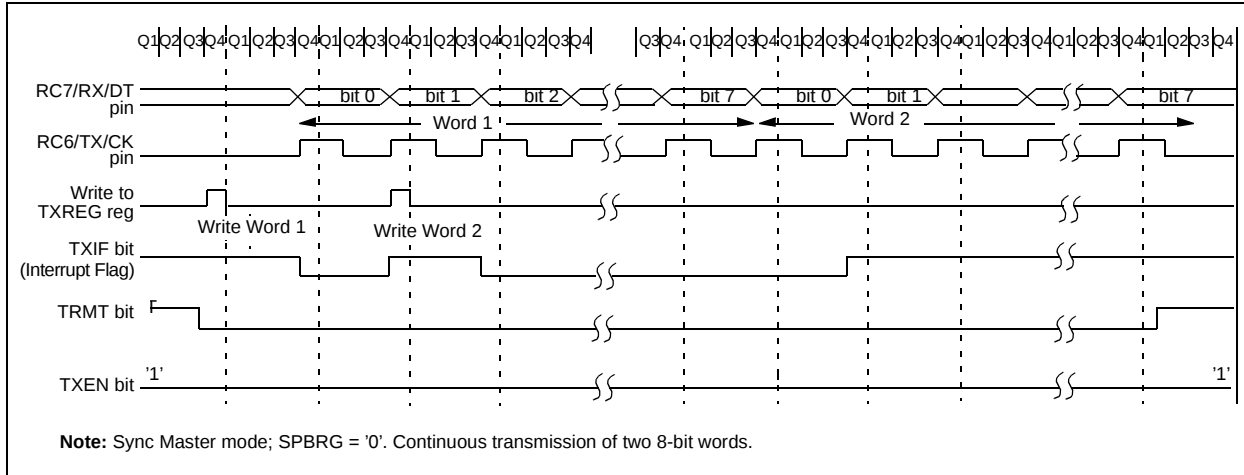


FIGURE 10-7: SYNCHRONOUS TRANSMISSION (THROUGH TXEN)

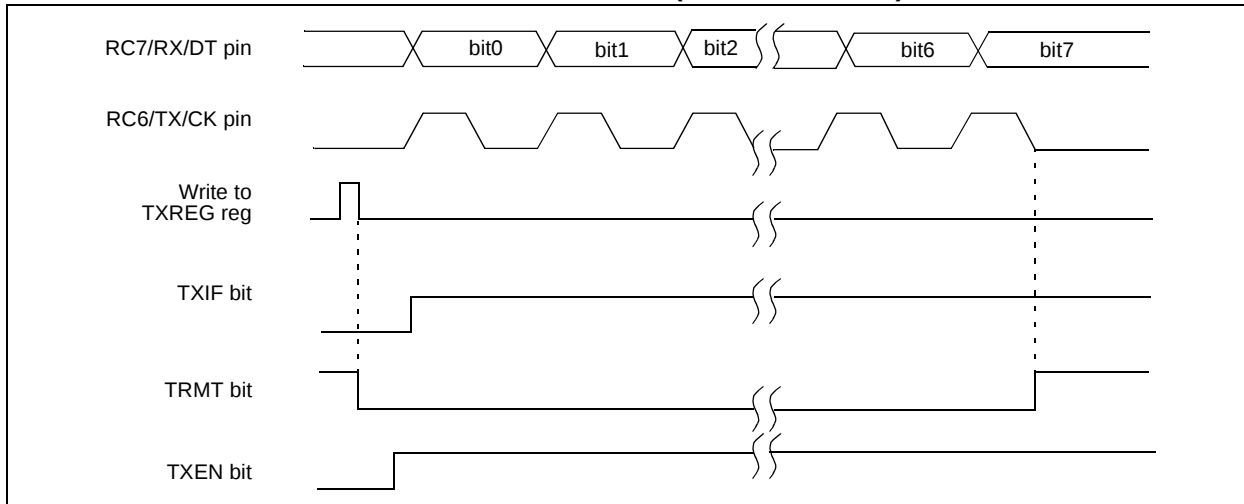


TABLE 10-7: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER TRANSMISSION

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	INTE	RBIE	TMR0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
18h	RCSTA	SPEN	RX9	SREN	CREN	—	FERR	OERR	RX9D	0000 -00x	0000 -00x
19h	TXREG	USART Transmit Register								0000 0000	0000 0000
8Ch	PIE1	PSPIE ⁽¹⁾	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 the PIC16F73/76 devices; always maintain these bits clear.

10.3.2 USART SYNCHRONOUS MASTER RECEPTION

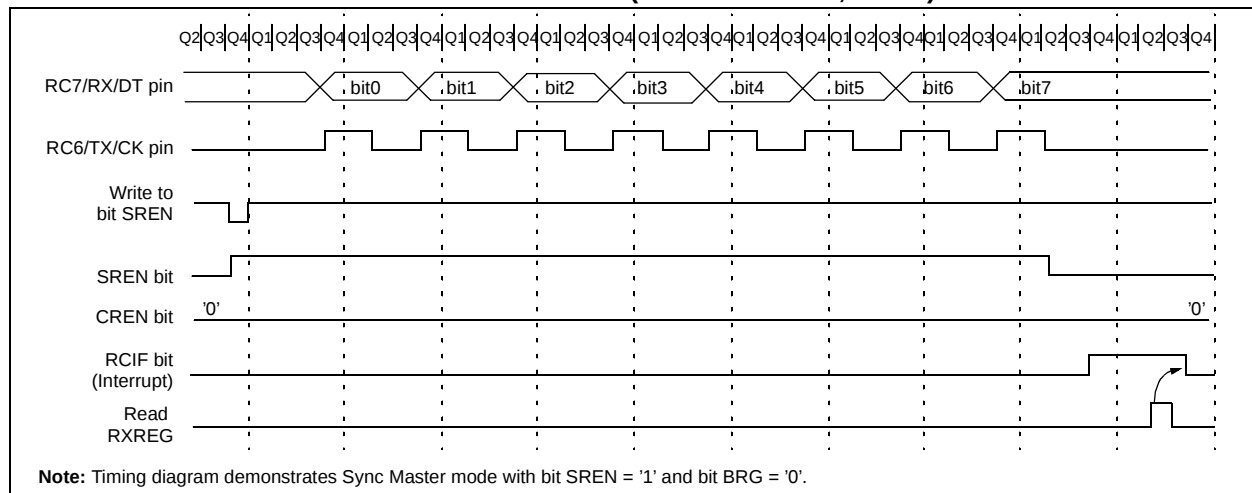
Once synchronous mode is selected, reception is enabled by setting either enable bit SREN (RCSTA<5>), or enable bit CREN (RCSTA<4>). Data is sampled on the RC7/RX/DT pin on the falling edge of the clock. If enable bit SREN is set, then only a single word is received. If enable bit CREN is set, the reception 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.

Steps to follow when setting up a Synchronous Master Reception:

1. Initialize the SPBRG register for the appropriate baud rate (Section 10.1).
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 in the INTCON register are set.

FIGURE 10-8: SYNCHRONOUS RECEPTION (MASTER MODE, SREN)



PIC16F7X

TABLE 10-8: 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	INTE	RBIE	TMR0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
18h	RCSTA	SPEN	RX9	SREN	CREN	—	FERR	OERR	RX9D	0000 -00x	0000 -00x
1Ah	RCREG	USART Receive Register								0000 0000	0000 0000
8Ch	PIE1	PSPIE ⁽¹⁾	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 the PIC16F73/76 devices; always maintain these bits clear.

10.4 USART Synchronous Slave Mode

Synchronous Slave mode differs from the Master mode, in 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>).

10.4.1 USART SYNCHRONOUS SLAVE TRANSMIT

The operation of the Synchronous Master and Slave modes are identical except in the case of the SLEEP mode.

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

- The first word will immediately transfer to the TSR register and transmit when the master device drives the CK line.
- The second word will remain in 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).

Follow these steps when setting up a Synchronous Slave Transmission:

- 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 in the INTCON register are set.

TABLE 10-9: 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	INTE	RBIE	TMR0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF ⁽¹⁾	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	USART Transmit Register								0000 0000	0000 0000
8Ch	PIE1	PSPIE ⁽¹⁾	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 the PIC16F73/76 devices; always maintain these bits clear.

10.4.2 USART SYNCHRONOUS SLAVE RECEPTION

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

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

Follow these steps when setting up a Synchronous Slave Reception:

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 in the INTCON register are set.

TABLE 10-10: 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	INTE	RBIE	TMR0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF ⁽¹⁾	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	USART Receive Register								0000 0000	0000 0000
8Ch	PIE1	PSPIE ⁽¹⁾	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 the PIC16F73/76 devices; always maintain these bits clear.

PIC16F7X

NOTES:

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

The 8-bit analog-to-digital (A/D) converter module has five inputs for the PIC16F73/76 and eight for the PIC16F74/77.

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

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

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

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

The ADCON0 register, shown in Register 11-1, controls the operation of the A/D module. The ADCON1 register, shown in Register 11-2, configures the functions of the port pins. The port pins can be configured as analog inputs (RA3 can also be a voltage reference), or as digital I/O.

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 (DS00546).

REGISTER 11-1: ADCON0 REGISTER (ADDRESS 1Fh)

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

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

00 = Fosc/2
 01 = Fosc/8
 10 = Fosc/32
 11 = FRC (clock derived from the internal A/D module RC oscillator)

bit 5-3 **CHS2:CHS0:** Analog Channel Select bits

000 = Channel 0 (RA0/AN0)
 001 = Channel 1 (RA1/AN1)
 010 = Channel 2 (RA2/AN2)
 011 = Channel 3 (RA3/AN3)
 100 = Channel 4 (RA5/AN4)
 101 = Channel 5 (RE0/AN5)⁽¹⁾
 110 = Channel 6 (RE1/AN6)⁽¹⁾
 111 = Channel 7 (RE2/AN7)⁽¹⁾

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

If ADON = 1:

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

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

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

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

Note 1: A/D channels 5, 6 and 7 are implemented on the PIC16F74/77 only.

Legend:

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

PIC16F7X

REGISTER 11-2: ADCON1 REGISTER (ADDRESS 9Fh)

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	PCFG2	PCFG1	PCFG0
bit 7					bit 0		

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

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

PCFG2:PCFG0	RA0	RA1	RA2	RA5	RA3	RE0 ⁽¹⁾	RE1 ⁽¹⁾	RE2 ⁽¹⁾	VREF
000	A	A	A	A	A	A	A	A	VDD
001	A	A	A	A	VREF	A	A	A	RA3
010	A	A	A	A	A	D	D	D	VDD
011	A	A	A	A	VREF	D	D	D	RA3
100	A	A	D	D	A	D	D	D	VDD
101	A	A	D	D	VREF	D	D	D	RA3
11x	D	D	D	D	D	D	D	D	VDD

A = Analog input

D = Digital I/O

Note 1: RE0, RE1 and RE2 are implemented on the PIC16F74/77 only.

Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

- n = Value at POR reset

'1' = Bit is set

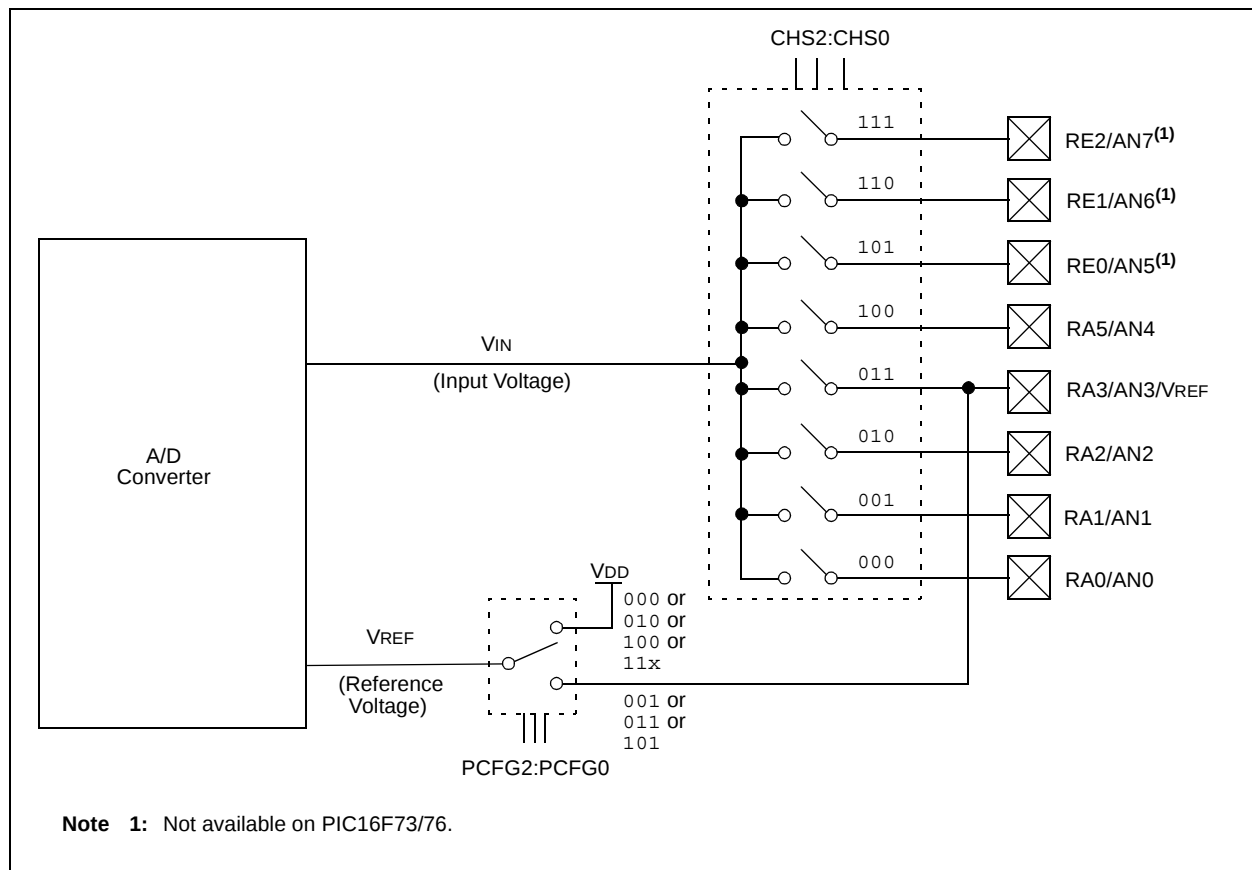
'0' = Bit is cleared

x = Bit is unknown

The following steps should be followed for doing an A/D conversion:

1. Configure the A/D module:
 - Configure analog pins, voltage reference, and digital I/O (ADCON1)
 - Select A/D conversion clock (ADCON0)
 - Turn on A/D module (ADCON0)
2. Configure the A/D interrupt (if desired):
 - Clear ADIF bit
 - Set ADIE bit
 - Set PEIE bit
 - Set GIE bit
3. Select an A/D input channel (ADCON0).
4. Wait for at least an appropriate acquisition period.
5. Start conversion:
 - Set GO/DONE bit (ADCON0)
6. Wait for the A/D conversion to complete, by either:
 - Polling for the GO/DONE bit to be cleared (interrupts disabled)
- OR
- Waiting for the A/D interrupt
7. Read A/D result register (ADRES), and clear bit ADIF if required.
8. For next conversion, go to step 3 or step 4, as required.

FIGURE 11-1: A/D BLOCK DIAGRAM



11.1 A/D Acquisition Requirements

For the A/D converter to meet its specified accuracy, the charge holding capacitor (CHOLD) must be allowed to fully charge to the input channel voltage level. The analog input model is shown in Figure 11-2. The source impedance (RS) and the internal sampling switch (RSS) impedance directly affect the time required to charge the capacitor CHOLD. The sampling switch (RSS) impedance varies over the device voltage (VDD), see Figure 11-2. The source impedance affects the offset voltage at the analog input (due to pin leakage current).

The maximum recommended impedance for analog sources is 10 kΩ. After the analog input channel is selected (changed), the acquisition period must pass before the conversion can be started.

To calculate the minimum acquisition time, TACQ, see the PICmicro™ Mid-Range MCU Family Reference Manual (DS33023). In general, however, given a maximum source impedance of 10 kΩ and at a temperature of 100°C, TACQ will be no more than 16 μsec.

FIGURE 11-2: ANALOG INPUT MODEL

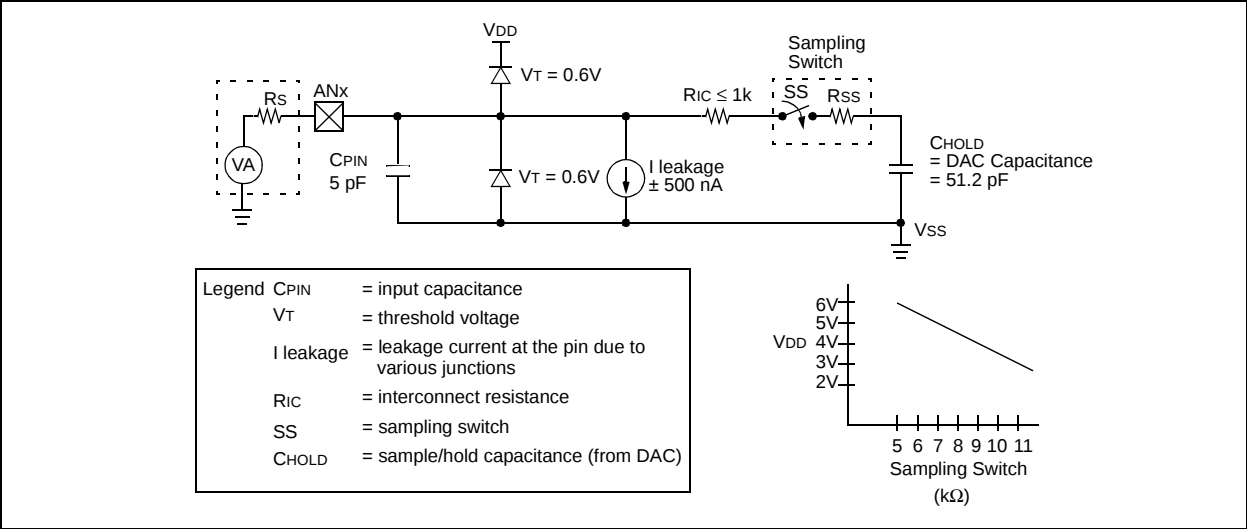


TABLE 11-1: TAD vs. MAXIMUM DEVICE OPERATING FREQUENCIES (STANDARD DEVICES (C))

AD Clock Source (TAD)		Maximum Device Frequency
Operation	ADCS1:ADCS0	Max.
2Tosc	00	1.25 MHz
8Tosc	01	5 MHz
32Tosc	10	20 MHz
RC ^(1, 2, 3)	11	(Note 1)

- Note 1:** The RC source has a typical TAD time of 4 μs but can vary between 2-6 μs.
- 2:** When the device frequencies are greater than 1 MHz, the RC A/D conversion clock source is only recommended for SLEEP operation.
- 3:** For extended voltage devices (LC), please refer to the Electrical Specifications section.

11.2 Selecting the A/D Conversion Clock

The A/D conversion time per bit is defined as T_{AD} . The A/D conversion requires 9.0 T_{AD} per 8-bit conversion. The source of the A/D conversion clock is software selectable. The four possible options for T_{AD} are:

- 2 T_{osc} ($F_{osc}/2$)
- 8 T_{osc} ($F_{osc}/8$)
- 32 T_{osc} ($F_{osc}/32$)
- Internal RC oscillator (2-6 μs)

For correct A/D conversions, the A/D conversion clock (T_{AD}) must be selected to ensure a minimum T_{AD} time as small as possible, but no less than 1.6 μs .

11.3 Configuring Analog Port Pins

The ADCON1, TRISA 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 (V_{OH} or V_{OL}) will be converted.

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

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.

11.4 A/D Conversions

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

Setting the $\overline{GO/DONE}$ bit begins an A/D conversion. When the conversion completes, the 8-bit result is placed in the ADRES register, the $\overline{GO/DONE}$ bit is cleared, and the ADIF flag (PIR<6>) is set.

If both the A/D interrupt bit ADIE (PIE1<6>) and the peripheral interrupt enable bit PEIE (INTCON<6>) are set, the device will wake from SLEEP whenever ADIF is set by hardware. In addition, an interrupt will also occur if the global interrupt bit GIE (INTCON<7>) is set.

Clearing the $\overline{GO/DONE}$ bit during a conversion will abort the current conversion. The ADRES register will NOT be changed, and the ADIF flag will not be set.

After the $\overline{GO/DONE}$ bit is cleared at either the end of a conversion, or by firmware, another conversion can be initiated by setting the $\overline{GO/DONE}$ bit. Users must still take into account the appropriate acquisition time for the application.

11.5 A/D Operation During SLEEP

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

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

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

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

11.6 Effects of a RESET

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

The ADRES register will contain unknown data after a Power-on Reset.

PIC16F7X

11.7 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 ADRES 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 11-2: 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	INTE	RBIE	TMR0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
0Dh	PIR2	—	—	—	—	—	—	—	CCP2IF	---- --0	---- --0
8Ch	PIE1	PSPIE ⁽¹⁾	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
8Dh	PIE2	—	—	—	—	—	—	—	CCP2IE	---- --0	---- --0
1Eh	ADRES	A/D Result Register								xxxx xxxx	uuuu uuuu
1Fh	ADCON0	ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	—	ADON	0000 00-0	0000 00-0
9Fh	ADCON1	—	—	—	—	—	PCFG2	PCFG1	PCFG0	---- -000	---- -000
05h	PORTA	—	—	RA5	RA4	RA3	RA2	RA1	RA0	--0x 0000	--0u 0000
85h	TRISA	—	—	PORTA Data Direction Register						--11 1111	--11 1111
09h	PORTE ⁽²⁾	—	—	—	—	—	RE2	RE1	RE0	---- -xxx	---- -uuu
89h	TRISE ⁽²⁾	IBF	OBF	IBOV	PSPMODE	—	PORTE Data Direction Bits			0000 -111	0000 -111

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 PIC16F73/76; always maintain these bits clear.

Note 2: These registers are reserved on the PIC16F73/76.

12.0 SPECIAL FEATURES OF THE CPU

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

- Oscillator Selection
- RESET
 - Power-on Reset (POR)
 - Power-up Timer (PWRT)
 - Oscillator Start-up Timer (OST)
 - Brown-out Reset (BOR)
- Interrupts
- Watchdog Timer (WDT)
- SLEEP
- Code Protection
- ID Locations
- In-Circuit Serial Programming

These devices have a Watchdog Timer, which can be enabled or disabled, using a configuration bit. It runs off its own RC oscillator for added reliability.

There are two timers that offer necessary delays on power-up. One is the Oscillator Start-up Timer (OST), intended to keep the chip in RESET until the crystal oscillator is stable. The other is the Power-up Timer (PWRT), which provides a fixed delay 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 Reference Manual (DS33023).

12.1 Configuration Bits

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

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

PIC16F7X

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

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/P-1	U-0	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1
—	—	—	—	—	—	—	BOREN	—	CP0	PWRTEN	WDTEN	FOSC1	FOSC0
bit13							bit0						

bit 13-7	Unimplemented: Read as '1'
bit 6	BOREN: Brown-out Reset Enable bit 1 = BOR enabled 0 = BOR disabled
bit 5	Unimplemented: Read as '1'
bit 4	CP0: FLASH Program Memory Code Protection bit 1 = Code protection off 0 = All memory locations code protected
bit 3	PWRTEN: Power-up Timer Enable bit 1 = PWRT disabled 0 = PWRT enabled
bit 2	WDTEN: Watchdog Timer Enable bit 1 = WDT enabled 0 = WDT disabled
bit 1-0	FOSC1:FOSC0: Oscillator Selection bits 11 = RC oscillator 10 = HS oscillator 01 = XT oscillator 00 = LP oscillator

Note 1: The erased (unprogrammed) value of the configuration word is 3FFFh.

Legend:

R = Readable bit	P = Programmable bit	U = Unimplemented bit, read as '0'
- n = Value when device is unprogrammed		u = Unchanged from programmed state

12.2 Oscillator Configurations

12.2.1 OSCILLATOR TYPES

The PIC16F7X can be operated in four different oscillator modes. The user can program two configuration bits (FOSC1 and FOSC0) to select one of these four modes:

- LP Low Power Crystal
- XT Crystal/Resonator
- HS High Speed Crystal/Resonator
- RC Resistor/Capacitor

12.2.2 CRYSTAL OSCILLATOR/CERAMIC RESONATORS

In XT, LP or HS modes, a crystal or ceramic resonator is connected to the OSC1/CLKIN and OSC2/CLKOUT pins to establish oscillation (Figure 12-1). The PIC16F7X oscillator design requires the use of a parallel cut crystal. Use of a series cut crystal may give a frequency out of the crystal manufacturers specifications. When in HS mode, the device can accept an external clock source to drive the OSC1/CLKIN pin (Figure 12-2). See Figure 15-1 or Figure 15-2 (depending on the part number and VDD range) for valid external clock frequencies.

FIGURE 12-1: CRYSTAL/CERAMIC RESONATOR OPERATION (HS, XT OR LP OSC CONFIGURATION)

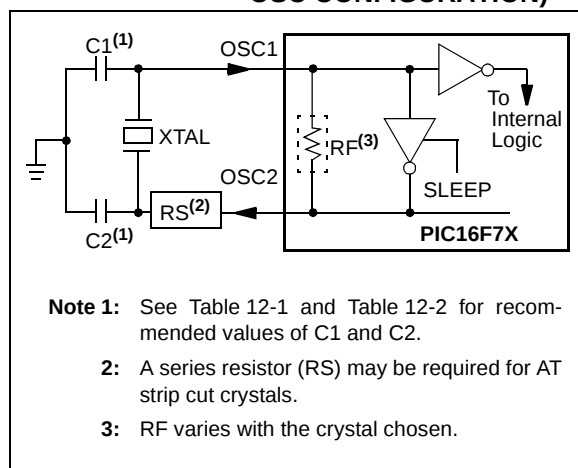


FIGURE 12-2: EXTERNAL CLOCK INPUT OPERATION (HS OSC CONFIGURATION)

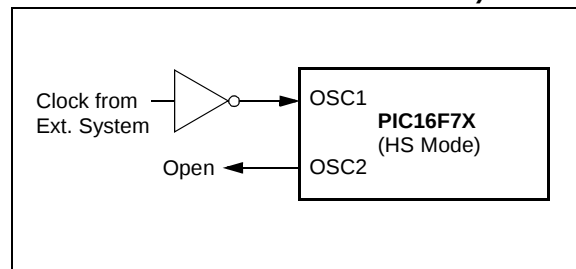


TABLE 12-1: CERAMIC RESONATORS (FOR DESIGN GUIDANCE ONLY)

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 at the bottom of page 92 for additional information.

Resonators Used:	
455 kHz	Panasonic EFO-A455K04B
2.0 MHz	Murata Erie CSA2.00MG
4.0 MHz	Murata Erie CSA4.00MG
8.0 MHz	Murata Erie CSA8.00MT
16.0 MHz	Murata Erie CSA16.00MX

PIC16F7X

TABLE 12-2: 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.

Crystals Used:

32 kHz	Epson C-001R32.768K-A
200 kHz	STD XTL 200.000KHz
1 MHz	ECS ECS-10-13-1
4 MHz	ECS ECS-40-20-1
8 MHz	EPSON CA-301 8.000M-C
20 MHz	EPSON CA-301 20.000M-C

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

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

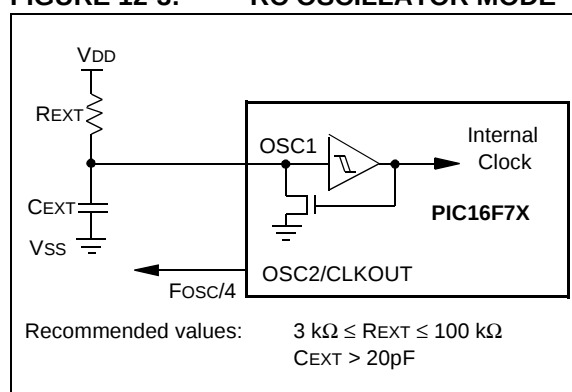
3: Rs may be required in HS mode, as well as XT mode, to avoid overdriving crystals with low drive level specification.

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

12.2.3 RC OSCILLATOR

For timing insensitive applications, the "RC" device option offers additional cost savings. The RC oscillator frequency is a function of the supply voltage, the resistor (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 process parameter 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 12-3 shows how the R/C combination is connected to the PIC16F7X.

FIGURE 12-3: RC OSCILLATOR MODE



12.3 RESET

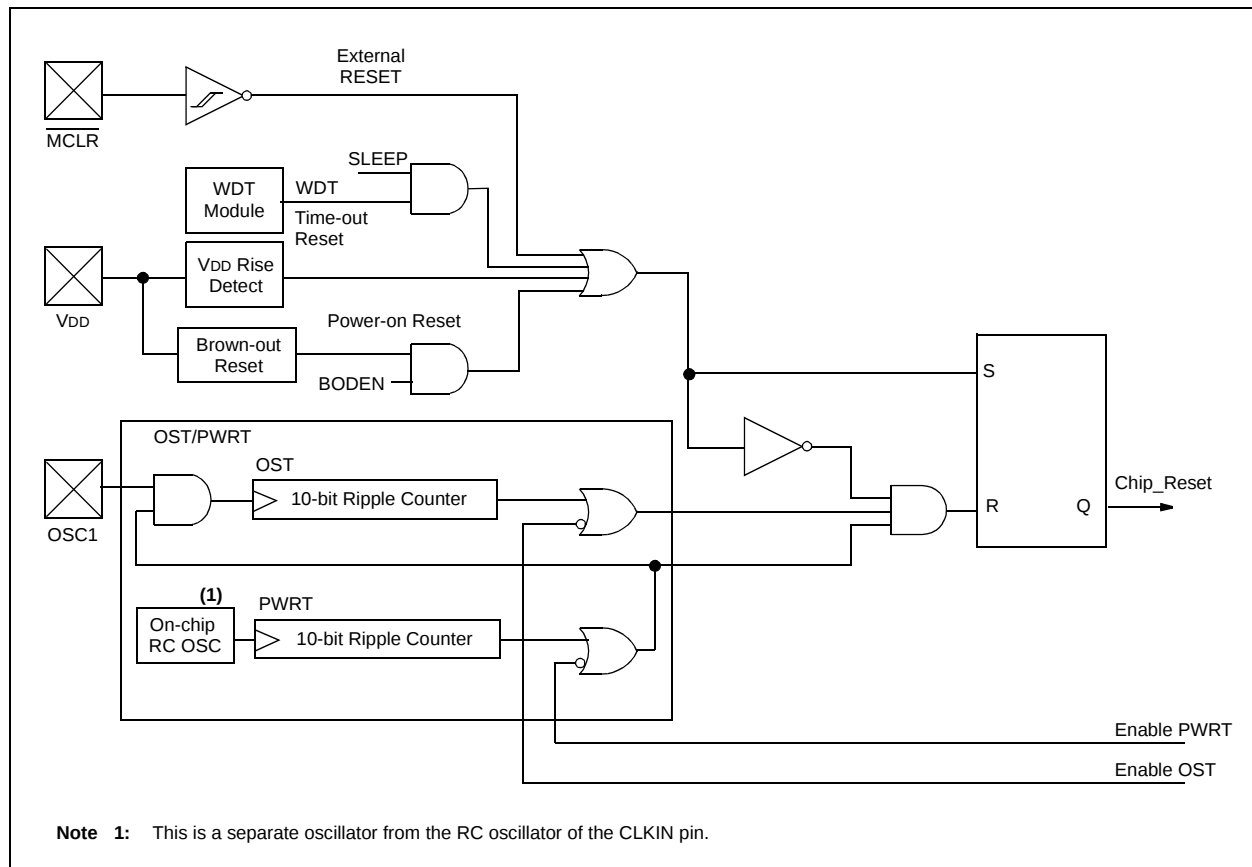
The PIC16F7X differentiates between various kinds of RESET:

- 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 12-4. These bits are used in software to determine the nature of the RESET. See Table 12-6 for a full description of RESET states of all registers.

A simplified block diagram of the on-chip RESET circuit is shown in Figure 12-4.

FIGURE 12-4: SIMPLIFIED BLOCK DIAGRAM OF ON-CHIP RESET CIRCUIT



PIC16F7X

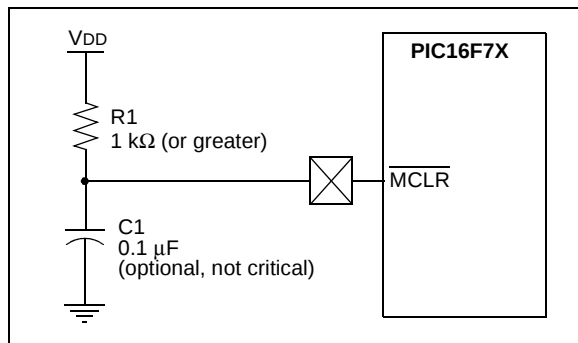
12.4 MCLR

PIC16F7X devices have a noise filter in the $\overline{\text{MCLR}}$ Reset path. The filter will detect and ignore small pulses.

It should be noted that a WDT Reset does not drive $\overline{\text{MCLR}}$ pin low.

The behavior of the ESD protection on the $\overline{\text{MCLR}}$ pin has been altered from previous devices of this family. Voltages applied to the pin that exceed its specification can result in both $\overline{\text{MCLR}}$ Resets and excessive current beyond the device specification during the ESD event. For this reason, Microchip recommends that the $\overline{\text{MCLR}}$ pin no longer be tied directly to V_{DD} . The use of an RC network, as shown in Figure 12-5, is suggested.

FIGURE 12-5: RECOMMENDED $\overline{\text{MCLR}}$ CIRCUIT



12.5 Power-on Reset (POR)

A Power-on Reset pulse is generated on-chip when V_{DD} rise is detected (in the range of 1.2V - 1.7V). To take advantage of the POR, tie the $\overline{\text{MCLR}}$ pin to V_{DD} as described in Section 12.4. A maximum rise time for V_{DD} is specified. See the Electrical Specifications 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 additional information, refer to Application Note, AN607, "Power-up Trouble Shooting" (DS00607).

12.6 Power-up Timer (PWRT)

The Power-up Timer provides a fixed 72 ms nominal time-out on power-up only from the POR. The Power-up Timer operates on an internal RC oscillator. The chip is kept in RESET as long as the PWRT is active. The PWRT's time delay allows V_{DD} to rise to an acceptable level. A configuration bit is provided to enable/disable the PWRT.

The power-up time delay will vary from chip to chip, due to V_{DD} , temperature and process variation. See DC parameters for details (TPWRT, parameter #33).

12.7 Oscillator Start-up Timer (OST)

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

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

12.8 Brown-out Reset (BOR)

The configuration bit, BODEN, can enable or disable the Brown-out Reset circuit. If V_{DD} falls below V_{BOR} (parameter D005, about 4V) for longer than T_{BOR} (parameter #35, about 100 μ S), the brown-out situation will reset the device. If V_{DD} falls below V_{BOR} for less than T_{BOR} , a RESET may not occur.

Once the brown-out occurs, the device will remain in Brown-out Reset until V_{DD} rises above V_{BOR} . The Power-up Timer then keeps the device in RESET for T_{PWRT} (parameter #33, about 72 mS). If V_{DD} should fall below V_{BOR} during T_{PWRT} , the Brown-out Reset process will restart when V_{DD} rises above V_{BOR} , with the Power-up Timer Reset. The Power-up Timer is always enabled when the Brown-out Reset circuit is enabled, regardless of the state of the PWRT configuration bit.

12.9 Time-out Sequence

On power-up, the time-out sequence is as follows: the PWRT delay starts (if enabled) when a POR Reset 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 PIC16F7X device operating in parallel.

Table 12-5 shows the RESET conditions for the STATUS, PCON and PC registers, while Table 12-6 shows the RESET conditions for all the registers.

12.10 Power Control/Status Register (PCON)

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

Bit0 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.

Bit1 is $\overline{\text{POR}}$ (Power-on Reset Status bit). It is cleared on a Power-on Reset and unaffected otherwise. The user must set this bit following a Power-on Reset.

TABLE 12-3: TIME-OUT IN VARIOUS SITUATIONS

Oscillator Configuration	Power-up		Brown-out	Wake-up from SLEEP
	$\overline{\text{PWRTE}} = 0$	$\overline{\text{PWRTE}} = 1$		
XT, HS, LP	72 ms + 1024 TOSC	1024 TOSC	72 ms + 1024 TOSC	1024 TOSC
RC	72 ms	—	72 ms	—

TABLE 12-4: STATUS BITS AND THEIR SIGNIFICANCE

$\overline{\text{POR}}$ (PCON<1>)	$\overline{\text{BOR}}$ (PCON<0>)	$\overline{\text{TO}}$ (STATUS<4>)	$\overline{\text{PD}}$ (STATUS<3>)	Significance
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	MCLR Reset during normal operation
1	1	1	0	MCLR Reset during SLEEP or interrupt wake-up from SLEEP

TABLE 12-5: RESET CONDITION FOR SPECIAL REGISTERS

Condition	Program Counter	STATUS Register	PCON Register
Power-on Reset	000h	0001 1xxx	---- --0x
MCLR Reset during normal operation	000h	000u uuuu	---- --uu
MCLR Reset during SLEEP	000h	0001 0uuu	---- --uu
WDT Reset	000h	0000 1uuu	---- --uu
WDT Wake-up	PC + 1	uuu0 0uuu	---- --uu
Brown-out Reset	000h	0001 1uuu	---- --u0
Interrupt wake-up from SLEEP	PC + 1 ⁽¹⁾	uuu1 0uuu	---- --uu

Legend: u = unchanged, x = unknown, - = unimplemented bit, read as '0'

Note 1: When the wake-up is due to an interrupt and the GIE bit is set, the PC is loaded with the interrupt vector (0004h).

PIC16F7X

TABLE 12-6: INITIALIZATION CONDITIONS FOR ALL REGISTERS

Register	Devices				Power-on Reset, Brown-out Reset	MCLR Reset, WDT Reset	Wake-up via WDT or Interrupt
W	73	74	76	77	xxxx xxxx	uuuu uuuu	uuuu uuuu
INDF	73	74	76	77	N/A	N/A	N/A
TMR0	73	74	76	77	xxxx xxxx	uuuu uuuu	uuuu uuuu
PCL	73	74	76	77	0000h	0000h	PC + 1 ⁽²⁾
STATUS	73	74	76	77	0001 1xxx	000q quuu ⁽³⁾	uuuq quuu ⁽³⁾
FSR	73	74	76	77	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTA	73	74	76	77	--0x 0000	--0u 0000	--uu uuuu
PORTB	73	74	76	77	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTC	73	74	76	77	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTD	73	74	76	77	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTE	73	74	76	77	---- -xxx	---- -uuu	---- -uuu
PCLATH	73	74	76	77	---0 0000	---0 0000	---u uuuu
INTCON	73	74	76	77	0000 000x	0000 000u	uuuu uuuu ⁽¹⁾
PIR1	73	74	76	77	r000 0000	r000 0000	ruuu uuuu ⁽¹⁾
	73	74	76	77	0000 0000	0000 0000	uuuu uuuu ⁽¹⁾
PIR2	73	74	76	77	---- --0	---- --0	---- --u ⁽¹⁾
TMR1L	73	74	76	77	xxxx xxxx	uuuu uuuu	uuuu uuuu
TMR1H	73	74	76	77	xxxx xxxx	uuuu uuuu	uuuu uuuu
T1CON	73	74	76	77	--00 0000	--uu uuuu	--uu uuuu
TMR2	73	74	76	77	0000 0000	0000 0000	uuuu uuuu
T2CON	73	74	76	77	-000 0000	-000 0000	-uuu uuuu
SSPBUF	73	74	76	77	xxxx xxxx	uuuu uuuu	uuuu uuuu
SSPCON	73	74	76	77	0000 0000	0000 0000	uuuu uuuu
CCPR1L	73	74	76	77	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCPR1H	73	74	76	77	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCP1CON	73	74	76	77	--00 0000	--00 0000	--uu uuuu
RCSTA	73	74	76	77	0000 -00x	0000 -00x	uuuu -uuu
TXREG	73	74	76	77	0000 0000	0000 0000	uuuu uuuu
RCREG	73	74	76	77	0000 0000	0000 0000	uuuu uuuu
CCPR2L	73	74	76	77	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCPR2H	73	74	76	77	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCP2CON	73	74	76	77	0000 0000	0000 0000	uuuu uuuu
ADRES	73	74	76	77	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADCON0	73	74	76	77	0000 00-0	0000 00-0	uuuu uu-u
OPTION_REG	73	74	76	77	1111 1111	1111 1111	uuuu uuuu
TRISA	73	74	76	77	--11 1111	--11 1111	--uu uuuu
TRISB	73	74	76	77	1111 1111	1111 1111	uuuu uuuu
TRISC	73	74	76	77	1111 1111	1111 1111	uuuu uuuu
TRISD	73	74	76	77	1111 1111	1111 1111	uuuu uuuu
TRISE	73	74	76	77	0000 -111	0000 -111	uuuu -uuu
PIE1	73	74	76	77	r000 0000	r000 0000	ruuu uuuu
	73	74	76	77	0000 0000	0000 0000	uuuu uuuu

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

Note 1: One or more bits in INTCON, PIR1 and/or PIR2 will be affected (to cause wake-up).

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

3: See Table 12-5 for RESET value for specific condition.

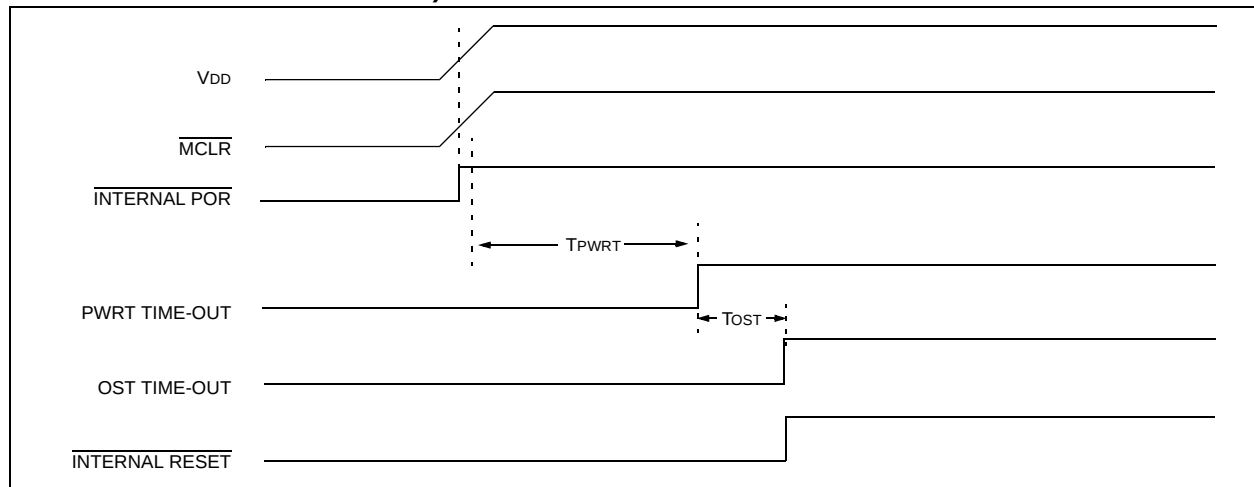
TABLE 12-6: INITIALIZATION CONDITIONS FOR ALL REGISTERS (CONTINUED)

Register	Devices				Power-on Reset, Brown-out Reset	MCLR Reset, WDT Reset	Wake-up via WDT or Interrupt
PIE2	73	74	76	77	---- ---0	---- ---0	---- ---u
PCON	73	74	76	77	---- --qq	---- --uu	---- --uu
PR2	73	74	76	77	1111 1111	1111 1111	1111 1111
SSPSTAT	73	74	76	77	--00 0000	--00 0000	--uu uuuu
SSPADD	73	74	76	77	0000 0000	0000 0000	uuuu uuuu
TXSTA	73	74	76	77	0000 -010	0000 -010	uuuu -uuu
SPBRG	73	74	76	77	0000 0000	0000 0000	uuuu uuuu
ADCON1	73	74	76	77	---- -000	---- -000	---- -uuu
PMDATA	73	74	76	77	0--- 0000	0--- 0000	u--- uuuu
PMADR	73	74	76	77	xxxx xxxx	uuuu uuuu	uuuu uuuu
PMDATH	73	74	76	77	xxxx xxxx	uuuu uuuu	uuuu uuuu
PMADRH	73	74	76	77	xxxx xxxx	uuuu uuuu	uuuu uuuu
PMCON1	73	74	76	77	1--- ---0	1--- ---0	1--- ---u

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

- Note 1:** One or more bits in INTCON, PIR1 and/or PIR2 will be affected (to cause wake-up).
Note 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).
Note 3: See Table 12-5 for RESET value for specific condition.

FIGURE 12-6: TIME-OUT SEQUENCE ON POWER-UP (MCLR TIED TO V_{DD} THROUGH RC NETWORK)



PIC16F7X

FIGURE 12-7: TIME-OUT SEQUENCE ON POWER-UP ($\overline{\text{MCLR}}$ NOT TIED TO V_{DD}): CASE 1

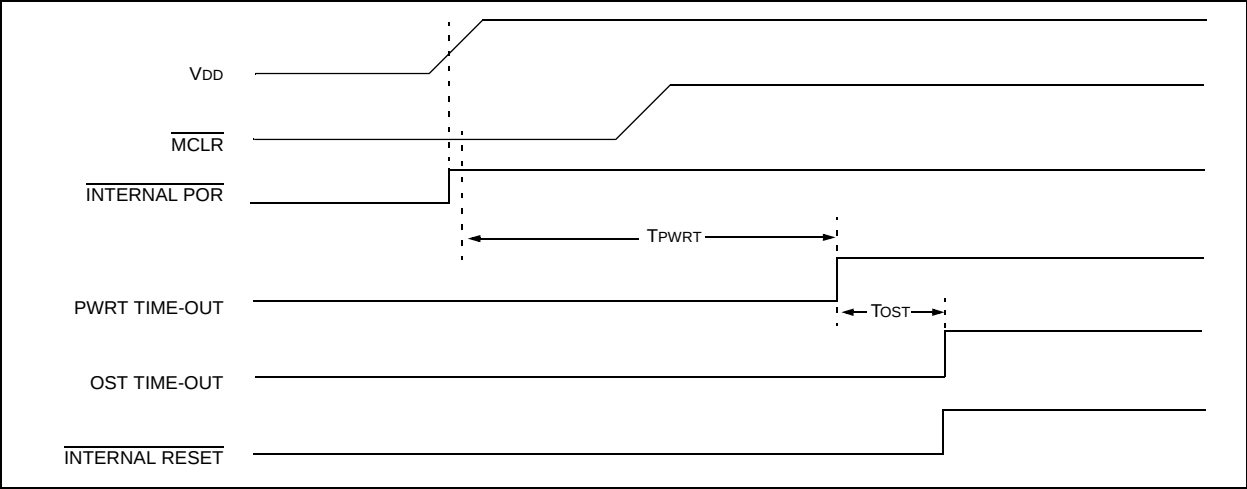


FIGURE 12-8: TIME-OUT SEQUENCE ON POWER-UP ($\overline{\text{MCLR}}$ NOT TIED TO V_{DD}): CASE 2

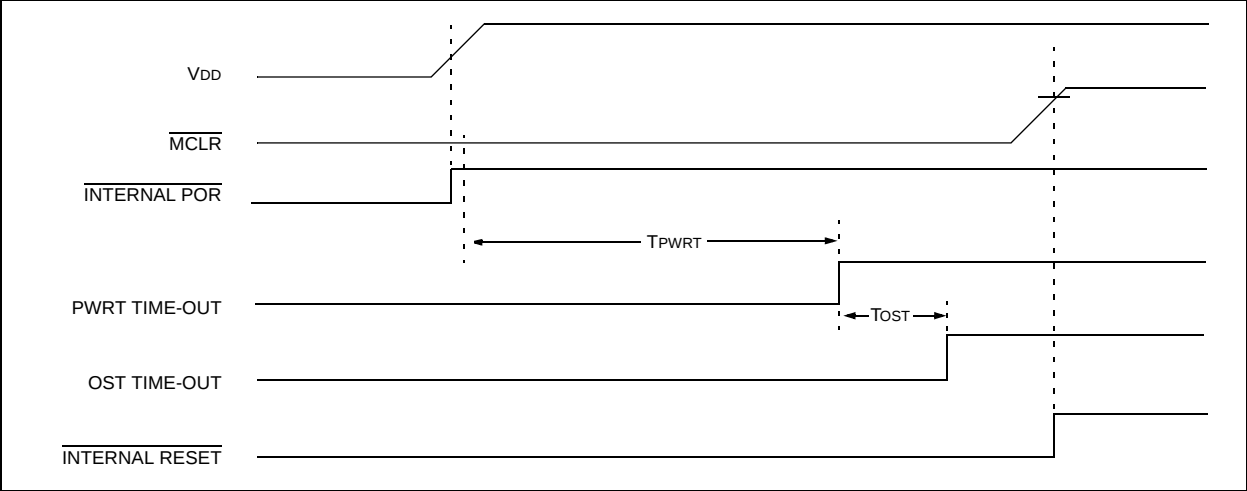
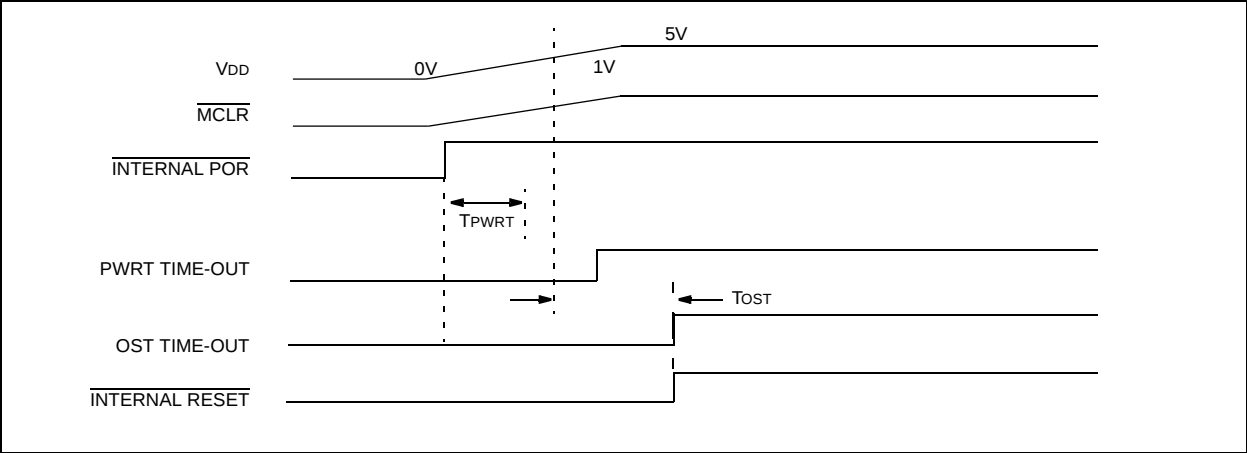


FIGURE 12-9: SLOW RISE TIME ($\overline{\text{MCLR}}$ TIED TO V_{DD} THROUGH RC NETWORK)



12.11 Interrupts

The PIC16F7X family has up to 12 sources of interrupt. The interrupt control register (INTCON) records individual interrupt requests in flag bits. It also has individual and global interrupt enable bits.

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

A global interrupt enable bit, GIE (INTCON<7>) enables (if set) all unmasked interrupts, or disables (if cleared) all interrupts. When bit GIE is enabled and an interrupt's flag bit and mask bit are set, the interrupt will vector immediately. Individual interrupts can be disabled through their corresponding enable bits in various registers. Individual interrupt bits are set, regardless of the status of the GIE bit. The GIE bit is cleared on RESET.

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

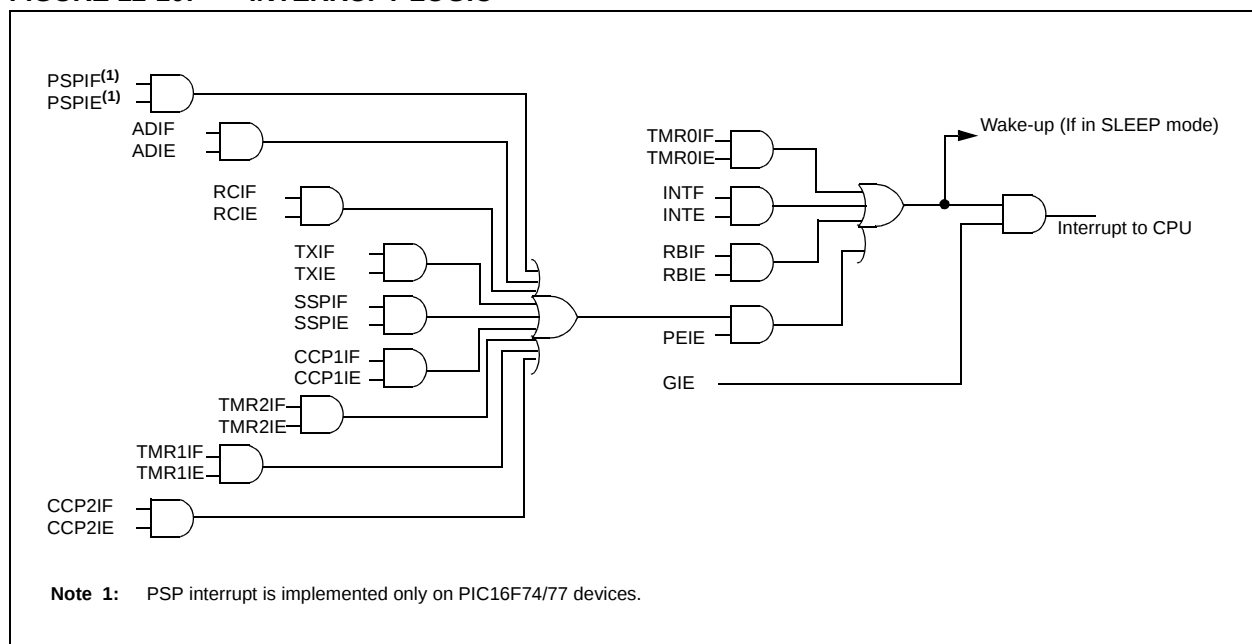
The RB0/INT pin interrupt, the RB port change interrupt and the TMR0 overflow interrupt flags are contained in the INTCON register.

The peripheral interrupt flags are contained in the Special Function Registers, PIR1 and PIR2. The corresponding interrupt enable bits are contained in Special Function Registers, PIE1 and PIE2, and the peripheral interrupt enable bit is contained in Special Function Register, INTCON.

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

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

FIGURE 12-10: INTERRUPT LOGIC



PIC16F7X

12.11.1 INT INTERRUPT

External interrupt on the RB0/INT pin is edge triggered, either rising, if bit INTEDG (OPTION_REG<6>) is set, or falling, if the INTEDG bit is clear. When a valid edge appears on the RB0/INT pin, flag bit INTF (INTCON<1>) is set. This interrupt can be disabled by clearing enable bit INTE (INTCON<4>). Flag bit INTF must be cleared in software in the Interrupt Service Routine before re-enabling this interrupt. The INT interrupt can wake-up the processor from SLEEP, if bit INTE was set prior to going into SLEEP. The status of global interrupt enable bit GIE decides whether or not the processor branches to the interrupt vector following wake-up. See Section 12.14 for details on SLEEP mode.

12.11.2 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>). (Section 5.0)

12.11.3 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 4.2.

12.12 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, PCLATH and STATUS registers). This will have to be implemented in software, as shown in Example 12-1.

For the PIC16F73/74 devices, the register W_TEMP must be defined in both banks 0 and 1 and must be defined at the same offset from the bank base address (i.e., If W_TEMP is defined at 20h in bank 0, it must also be defined at A0h in bank 1.). The registers, PCLATH_TEMP and STATUS_TEMP, are only defined in bank 0.

Since the upper 16 bytes of each bank are common in the PIC16F76/77 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 12-1 can be used.

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

```
MOVWF  W_TEMP          ;Copy W to TEMP register
SWAPF  STATUS,W        ;Swap status to be saved into W
CLRF   STATUS          ;bank 0, regardless of current bank, Clears IRP,RP1,RP0
MOVWF  STATUS_TEMP     ;Save status to bank zero STATUS_TEMP register
MOVF   PCLATH, W       ;Only required if using pages 1, 2 and/or 3
MOVWF  PCLATH_TEMP     ;Save PCLATH into W
CLRF   PCLATH          ;Page zero, regardless of current page
:
: (ISR)                ;Insert user code here
:
MOVF   PCLATH_TEMP, W   ;Restore PCLATH
MOVWF  PCLATH          ;Move W into PCLATH
SWAPF  STATUS_TEMP,W   ;Swap STATUS_TEMP register into W
                        ;(sets bank to original state)
MOVWF  STATUS          ;Move W into STATUS register
SWAPF  W_TEMP,F        ;Swap W_TEMP
SWAPF  W_TEMP,W        ;Swap W_TEMP into W
```

12.13 Watchdog Timer (WDT)

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

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

The WDT can be permanently disabled by clearing configuration bit, WDTE (Section 12.1).

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

Note 1: The `CLRWDT` and `SLEEP` instructions clear the WDT and the postscaler, if assigned to the WDT, and prevent it from timing out and generating a device RESET condition.

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

FIGURE 12-11: WATCHDOG TIMER BLOCK DIAGRAM

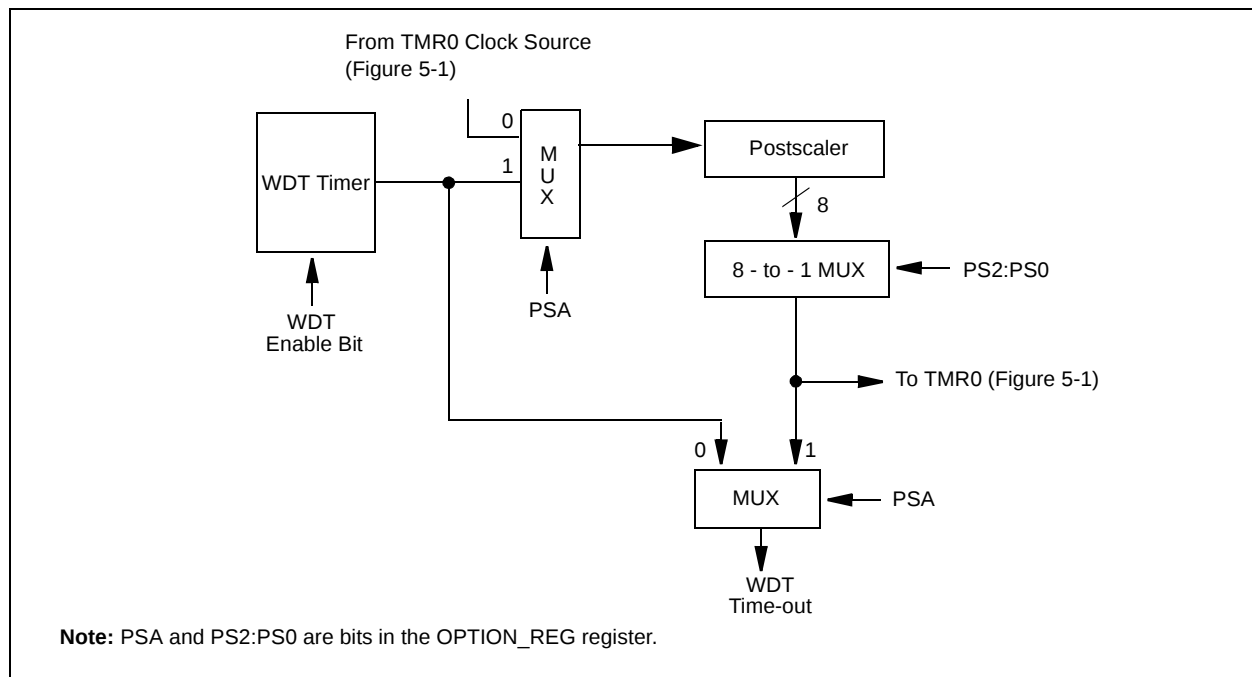


TABLE 12-7: SUMMARY OF WATCHDOG TIMER REGISTERS

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
2007h	Config. bits	(1)	BODEN ⁽¹⁾	—	CP0	PWRTE ⁽¹⁾	WDTE	FOSC1	FOSC0
81h,181h	OPTION_REG	RBPUR	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0

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

Note 1: See Register 12-1 for operation of these bits.

12.14 Power-down Mode (SLEEP)

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

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

For lowest current consumption in this mode, place all I/O pins at either V_{DD} or V_{SS} , 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 hi-impedance inputs, high or low externally, to avoid switching currents caused by floating inputs. The `T0CKI` input should also be at V_{DD} or V_{SS} for lowest current consumption. The contribution from on-chip pull-ups on `PORTB` should also be considered.

The $\overline{MCLR}$ pin must be at a logic high level (V_{IHMC}).

12.14.1 WAKE-UP FROM SLEEP

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

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

External $\overline{MCLR}$ Reset will cause a device `RESET`. All other events are considered a continuation of program execution and cause a "wake-up". The $\overline{TO}$ and $\overline{PD}$ bits in the `STATUS` register can be used to determine the cause of device `RESET`. The $\overline{PD}$ bit, which is set on power-up, is cleared when `SLEEP` is invoked. The $\overline{TO}$ bit is cleared if a `WDT` time-out occurred and caused wake-up.

The following peripheral interrupts can wake the device from `SLEEP`:

1. `PSP` read or write (`PIC16F74/77` only).
2. `TMR1` interrupt. `Timer1` must be operating as an asynchronous counter.
3. `CCP` Capture mode interrupt.
4. Special event trigger (`Timer1` in Asynchronous mode, using an external clock).
5. `SSP` (`START/STOP`) bit detect interrupt.
6. `SSP` transmit or receive in Slave mode (`SPI/I2C`).
7. `USART` `RX` or `TX` (Synchronous Slave mode).
8. A/D conversion (when A/D clock source is `RC`).

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

When the `SLEEP` instruction is being executed, the next instruction (`PC + 1`) is pre-fetched. For the device to wake-up through an interrupt event, the corresponding interrupt enable bit must be set (enabled). Wake-up occurs, regardless of the state of the `GIE` bit. If the `GIE` bit is clear (disabled), the device continues execution at the instruction after the `SLEEP` instruction. If the `GIE` bit is set (enabled), the device executes the instruction after the `SLEEP` instruction and then branches to the interrupt address (`0004h`). In cases where the execution of the instruction following `SLEEP` is not desirable, the user should have a `NOP` after the `SLEEP` instruction.

12.14.2 WAKE-UP USING INTERRUPTS

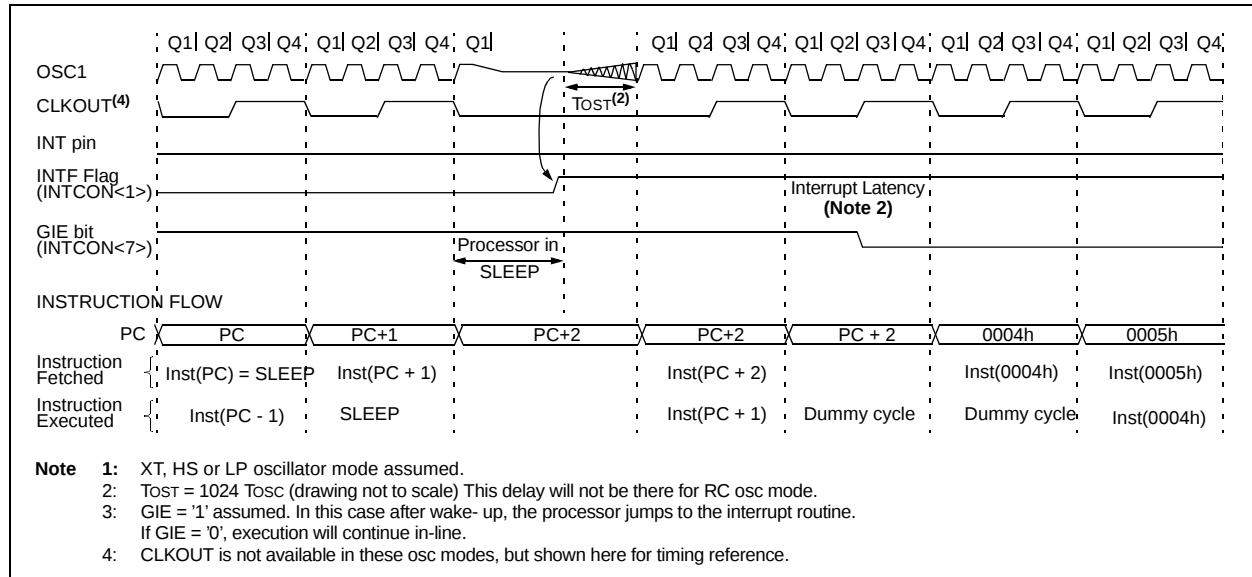
When global interrupts are disabled (`GIE` cleared) and any interrupt source has both its interrupt enable bit and interrupt flag bit set, one of the following will occur:

- If the interrupt occurs **before** the execution of a `SLEEP` instruction, the `SLEEP` instruction will complete as a `NOP`. Therefore, the `WDT` and `WDT` postscaler will not be cleared, the $\overline{TO}$ bit will not be set and $\overline{PD}$ bits will not be cleared.
- If the interrupt occurs **during or after** the execution of a `SLEEP` instruction, the device will immediately wake-up from `SLEEP`. The `SLEEP` instruction will be completely executed before the wake-up. Therefore, the `WDT` and `WDT` postscaler will be cleared, the $\overline{TO}$ bit will be set and the $\overline{PD}$ bit will be cleared.

Even if the flag bits were checked before executing a `SLEEP` instruction, it may be possible for flag bits to become set before the `SLEEP` instruction completes. To determine whether a `SLEEP` instruction executed, test the $\overline{PD}$ bit. If the $\overline{PD}$ 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 12-12: WAKE-UP FROM SLEEP THROUGH INTERRUPT



12.15 Program Verification/Code Protection

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

12.16 ID Locations

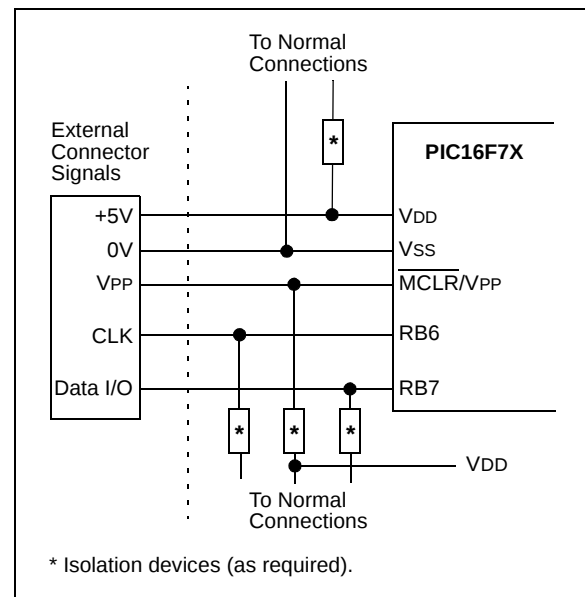
Four memory locations (2000h - 2003h) are designated as ID locations, where the user can store checksum or other code identification numbers. These locations are not accessible during normal execution, but are readable and writable during program/verify. It is recommended that only the 4 Least Significant bits of the ID location are used.

12.17 In-Circuit Serial Programming

PIC16F7X microcontrollers can be serially programmed while in the end application circuit. This is simply done, with two lines for clock and data and three other lines for power, ground, and the programming voltage (see Figure 12-13 for an example). This allows customers to manufacture boards with unprogrammed devices, and then program the microcontroller just before shipping the product. This also allows the most recent firmware or a custom firmware to be programmed.

For general information of serial programming, please refer to the In-Circuit Serial Programming (ICSP™) Guide (DS30277). For specific details on programming commands and operations for the PIC16F7X devices, please refer to the latest version of the PIC16F7X FLASH Program Memory Programming Specification (DS30324).

FIGURE 12-13: TYPICAL IN-CIRCUIT SERIAL PROGRAMMING CONNECTION



PIC16F7X

NOTES:

13.0 INSTRUCTION SET SUMMARY

The PIC16 instruction set is highly orthogonal and is comprised of three basic categories:

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

Each PIC16 instruction is a 14-bit word divided into an **opcode**, which specifies the instruction type and one or more **operands**, which further specify the operation of the instruction. The formats for each of the categories are presented in Figure 13-1, while the various opcode fields are summarized in Table 13-1.

Table 13-2 lists the instructions recognized by the MPASM™ Assembler. A complete description of each instruction is also available in the PICmicro™ Mid-Range 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 PIC16F7X products, do not use the **OPTION** and **TRIS** instructions.

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

13.1 READ-MODIFY-WRITE OPERATIONS

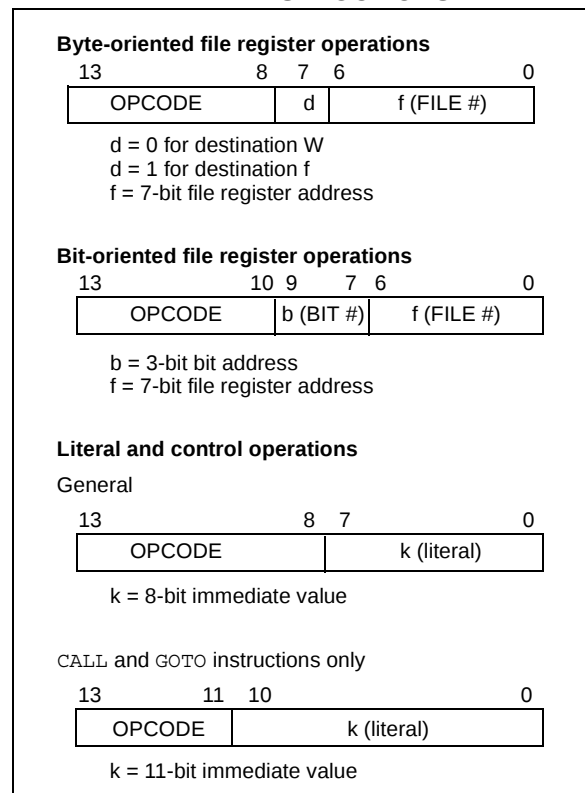
Any instruction that specifies a file register as part of the instruction performs a Read-Modify-Write (R-M-W) operation. The register is read, the data is modified, and the result is stored according to either the instruction, or the destination designator 'd'. A read operation is performed on a register even if the instruction writes to that register.

For example, a "clrf PORTB" instruction will read PORTB, clear all the data bits, then write the result back to PORTB. This example would have the unintended result that the condition that sets the RBIF flag would be cleared for pins configured as inputs and using the PORTB interrupt-on-change feature.

TABLE 13-1: OPCODE FIELD DESCRIPTIONS

Field	Description
f	Register file address (0x00 to 0x7F)
W	Working register (accumulator)
b	Bit address within an 8-bit file register
k	Literal field, constant data or label
x	Don't care location (= 0 or 1). 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 13-1: GENERAL FORMAT FOR INSTRUCTIONS



PIC16F7X

TABLE 13-2: PIC16F7X INSTRUCTION SET

Mnemonic, Operands		Description	Cycles	14-Bit Opcode				Status Affected	Notes
				MSb		LSb			
BYTE-ORIENTED FILE REGISTER OPERATIONS									
ADDWF	f, d	Add W and f	1	00	0111	dfff	ffff	C,DC,Z	1,2
ANDWF	f, d	AND W with f	1	00	0101	dfff	ffff	Z	1,2
CLRF	f	Clear f	1	00	0001	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
BIT-ORIENTED FILE REGISTER OPERATIONS									
BCF	f, b	Bit Clear f	1	01	00bb	bfff	ffff		1,2
BSF	f, b	Bit Set f	1	01	01bb	bfff	ffff		1,2
BTFSC	f, b	Bit Test f, Skip if Clear	1 (2)	01	10bb	bfff	ffff		3
BTFSS	f, b	Bit Test f, Skip if Set	1 (2)	01	11bb	bfff	ffff		3
LITERAL AND CONTROL OPERATIONS									
ADDLW	k	Add literal and W	1	11	111x	kkkk	kkkk	C,DC,Z	
ANDLW	k	AND literal with W	1	11	1001	kkkk	kkkk	Z	
CALL	k	Call subroutine	2	10	0kkk	kkkk	kkkk		
CLRWDT	-	Clear Watchdog Timer	1	00	0000	0110	0100	$\overline{TO}, \overline{PD}$	
GOTO	k	Go to address	2	10	1kkk	kkkk	kkkk		
IORLW	k	Inclusive OR literal with W	1	11	1000	kkkk	kkkk	Z	
MOVLW	k	Move literal to W	1	11	00xx	kkkk	kkkk		
RETFIE	-	Return from interrupt	2	00	0000	0000	1001		
RETLW	k	Return with literal in W	2	11	01xx	kkkk	kkkk		
RETURN	-	Return from Subroutine	2	00	0000	0000	1000		
SLEEP	-	Go into Standby mode	1	00	0000	0110	0011	$\overline{TO}, \overline{PD}$	
SUBLW	k	Subtract W from literal	1	11	110x	kkkk	kkkk	C,DC,Z	
XORLW	k	Exclusive OR literal with W	1	11	1010	kkkk	kkkk	Z	

- Note 1:** When an I/O register is modified as a function of itself (e.g., `MOVF PORTB, 1`), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as input and is driven low by an external device, the data will be written back with a '0'.
- 2:** If this instruction is executed on the TMR0 register (and, where applicable, d = 1), the prescaler will be cleared if assigned to the Timer0 module.
- 3:** If 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).

13.2 Instruction Descriptions

ADDLW Add Literal and W

Syntax:	[<i>label</i>] ADDLW <i>k</i>
Operands:	$0 \leq k \leq 255$
Operation:	$(W) + k \rightarrow (W)$
Status Affected:	C, DC, Z
Description:	The contents of the W register are added to the eight-bit literal 'k' and the result is placed in the W register.

ADDWF Add W and f

Syntax:	[<i>label</i>] ADDWF <i>f</i> , <i>d</i>
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'.

ANDLW AND Literal with W

Syntax:	[<i>label</i>] ANDLW <i>k</i>
Operands:	$0 \leq k \leq 255$
Operation:	$(W) .\text{AND.} (k) \rightarrow (W)$
Status Affected:	Z
Description:	The contents of W register are AND'ed with the eight-bit literal 'k'. The result is placed in the W register.

ANDWF AND W with f

Syntax:	[<i>label</i>] ANDWF <i>f</i> , <i>d</i>
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'.

BCF Bit Clear f

Syntax:	[<i>label</i>] BCF <i>f</i> , <i>b</i>
Operands:	$0 \leq f \leq 127$ $0 \leq b \leq 7$
Operation:	$0 \rightarrow (f)$
Status Affected:	None
Description:	Bit 'b' in register 'f' is cleared.

BSF Bit Set f

Syntax:	[<i>label</i>] BSF <i>f</i> , <i>b</i>
Operands:	$0 \leq f \leq 127$ $0 \leq b \leq 7$
Operation:	$1 \rightarrow (f)$
Status Affected:	None
Description:	Bit 'b' in register 'f' is set.

BTFSS Bit Test f, Skip if Set

Syntax:	[<i>label</i>] BTFSS <i>f</i> , <i>b</i>
Operands:	$0 \leq f \leq 127$ $0 \leq b < 7$
Operation:	skip if $(f) = 1$
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 2Tcy instruction.

BTFSC Bit Test, Skip if Clear

Syntax:	[<i>label</i>] BTFSC <i>f</i> , <i>b</i>
Operands:	$0 \leq f \leq 127$ $0 \leq b \leq 7$
Operation:	skip if $(f) = 0$
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 2Tcy instruction.

PIC16F7X

CALL Call Subroutine

Syntax: [*label*] CALL *k*

Operands: $0 \leq k \leq 2047$

Operation: (PC)+1 → TOS,
 $k \rightarrow PC<10:0>$,
(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.

CLRWDTClear Watchdog Timer

Syntax: [*label*] CLRWDTClear Watchdog Timer

Operands: None

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

Status Affected: $\overline{TO}$, $\overline{PD}$

Description: CLRWDTClear Watchdog Timer instruction resets the Watchdog Timer. It also resets the prescaler of the WDT. Status bits $\overline{TO}$ and $\overline{PD}$ are set.

CLRF Clear f

Syntax: [*label*] CLRF *f*

Operands: $0 \leq f \leq 127$

Operation: 00h → (f)
 $1 \rightarrow Z$

Status Affected: Z

Description: The contents of register 'f' are cleared and the Z bit is set.

COMF Complement f

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

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

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

Status Affected: Z

Description: The contents of register 'f' are complemented. If 'd' is 0, the result is stored in W. If 'd' is 1, the result is stored back in register 'f'.

CLRWClear W

Syntax: [*label*] CLRW

Operands: None

Operation: 00h → (W)
 $1 \rightarrow Z$

Status Affected: Z

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

DECF Decrement f

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

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

Operation: $(f) - 1 \rightarrow (\text{destination})$

Status Affected: Z

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

DECFSZ Decrement f, Skip if 0

Syntax: [*label*] DECFSZ f,d

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

Operation: $(f) - 1 \rightarrow (\text{destination});$
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 2Tcy 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 2Tcy 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 $\text{PCLATH}\langle 4:3 \rangle$. 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 OR'ed with the eight-bit literal 'k'. The result is placed in the W register.

INCF Increment f

Syntax: [*label*] INCF f,d

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

Operation: $(f) + 1 \rightarrow (\text{destination})$

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'.

PIC16F7X

MOVF **Move f**

Syntax: [*label*] MOVF f,d

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

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

Status Affected: Z

Description: The contents of register f are moved to a destination dependant upon the status of d. If d = 0, 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.

RETIE **Return from Interrupt**

Syntax: [*label*] RETIE

Operands: None

Operation: TOS $\rightarrow$ PC,
 1 $\rightarrow$ GIE

Status Affected: None

MOVWF **Move W to f**

Syntax: [*label*] MOVWF f

Operands: $0 \leq f \leq 127$

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

Status Affected: None

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

RETLW **Return with Literal in W**

Syntax: [*label*] RETLW k

Operands: $0 \leq k \leq 255$

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

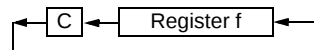
Syntax: [*label*] RLF f,d

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

Syntax: [*label*] SLEEP

Operands: None

Operation: 00h $\rightarrow$ WDT,
 0 $\rightarrow$ WDT prescaler,
 1 $\rightarrow \overline{TO}$,
 0 $\rightarrow \overline{PD}$

Status Affected: $\overline{TO}$, $\overline{PD}$

Description: The power-down status bit, $\overline{PD}$ is cleared. Time-out status bit, $\overline{TO}$ is set. Watchdog Timer and its prescaler are cleared. The processor is put into SLEEP mode with the oscillator stopped.

RETURN Return from Subroutine

Syntax: [*label*] RETURN

Operands: None

Operation: TOS $\rightarrow$ 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

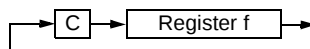
Syntax: [*label*] RRF f,d

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'.

PIC16F7X

SWAPF **Swap Nibbles in f**

Syntax: `[label] SWAPF f,d`

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

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

Status Affected: None

Description: The upper and lower nibbles of register 'f' are exchanged. If 'd' 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: $(W) .XOR. (f) \rightarrow (\text{destination})$

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

Status Affected: Z

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

14.0 DEVELOPMENT SUPPORT

The PICmicro® microcontrollers are supported with a full range of hardware and software development tools:

- Integrated Development Environment
 - MPLAB® IDE Software
- Assemblers/Compilers/Linkers
 - MPASM™ Assembler
 - MPLAB C17 and MPLAB C18 C Compilers
 - MPLINK™ Object Linker/
MPLIB™ Object Librarian
- Simulators
 - MPLAB SIM Software Simulator
- Emulators
 - MPLAB ICE 2000 In-Circuit Emulator
 - ICEPIC™ In-Circuit Emulator
- In-Circuit Debugger
 - MPLAB ICD
- Device Programmers
 - PRO MATE® II Universal Device Programmer
 - PICSTART® Plus Entry-Level Development Programmer
- Low Cost Demonstration Boards
 - PICDEM™ 1 Demonstration Board
 - PICDEM 2 Demonstration Board
 - PICDEM 3 Demonstration Board
 - PICDEM 17 Demonstration Board
 - KEELQ® Demonstration Board

14.1 MPLAB Integrated Development Environment Software

The MPLAB IDE software brings an ease of software development previously unseen in the 8-bit microcontroller market. The MPLAB IDE is a Windows®-based application that contains:

- An interface to debugging tools
 - simulator
 - programmer (sold separately)
 - emulator (sold separately)
 - in-circuit debugger (sold separately)
- A full-featured editor
- A project manager
- Customizable toolbar and key mapping
- A status bar
- 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
 - absolute listing file
 - machine code

The ability to use MPLAB IDE with multiple debugging tools allows users to easily switch from the cost-effective simulator to a full-featured emulator with minimal retraining.

14.2 MPASM Assembler

The MPASM assembler is a full-featured universal macro assembler for all PICmicro MCU's.

The MPASM assembler has a command line interface and a Windows shell. It can be used as a stand-alone application on a Windows 3.x or greater system, or it can be used through MPLAB IDE. The MPASM assembler generates relocatable object files for the MPLINK object linker, Intel® standard HEX files, MAP files to detail memory usage and symbol reference, an absolute LST file that contains source lines and generated machine code, and a COD file for debugging.

The MPASM assembler features include:

- Integration into MPLAB IDE projects.
- User-defined macros to streamline assembly code.
- Conditional assembly for multi-purpose source files.
- Directives that allow complete control over the assembly process.

14.3 MPLAB 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, respectively. These compilers provide powerful integration capabilities and ease of use not found with other compilers.

For easier source level debugging, the compilers provide symbol information that is compatible with the MPLAB IDE memory display.

14.4 MPLINK Object Linker/ MPLIB Object Librarian

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

The MPLIB object librarian is a librarian for pre-compiled code to be used with the MPLINK object linker. When a routine from a library is called from another 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 MPLIB object librarian manages the creation and modification of library files.

The MPLINK object linker features include:

- Integration with MPASM assembler and MPLAB C17 and MPLAB C18 C compilers.
- Allows all memory areas to be defined as sections to provide link-time flexibility.

The MPLIB object librarian features include:

- Easier linking because single libraries can be included instead of many smaller files.
- Helps keep code maintainable by grouping related modules together.
- Allows libraries to be created and modules to be added, listed, replaced, deleted or extracted.

14.5 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 of the pins. 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 the MPLAB C18 C compilers and the MPASM assembler. The software simulator offers the flexibility to develop and debug code outside of the laboratory environment, making it an excellent multi-project software development tool.

14.6 MPLAB ICE High Performance Universal In-Circuit Emulator with MPLAB IDE

The MPLAB ICE universal in-circuit emulator is intended to provide the product development engineer with a complete microcontroller design tool set for PICmicro microcontrollers (MCUs). Software control of the MPLAB ICE in-circuit emulator is provided by the MPLAB Integrated Development Environment (IDE), 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 in-circuit emulator system has been designed as a real-time emulation system, with advanced features that are generally found on more expensive development tools. The PC platform and Microsoft® Windows environment were chosen to best make these features available to you, the end user.

14.7 ICEPIC In-Circuit Emulator

The ICEPIC low cost, in-circuit emulator is a solution for the Microchip Technology PIC16C5X, PIC16C6X, PIC16C7X and PIC16CXXX families of 8-bit One-Time-Programmable (OTP) microcontrollers. The modular system can support different subsets of PIC16C5X or PIC16CXXX products through the use of interchangeable personality modules, or daughter boards. The emulator is capable of emulating without target application circuitry being present.

14.8 MPLAB ICD In-Circuit Debugger

Microchip's In-Circuit Debugger, MPLAB ICD, is a powerful, low cost, run-time development tool. This tool is based on the FLASH PICmicro MCUs and can be used to develop for this and other PICmicro microcontrollers. The MPLAB ICD utilizes the in-circuit debugging capability built into the FLASH devices. This feature, along with Microchip's In-Circuit Serial Programming™ 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 watching variables, single-stepping and setting break points. Running at full speed enables testing hardware in real-time.

14.9 PRO MATE II Universal Device Programmer

The PRO MATE II universal device programmer is a full-featured programmer, capable of operating in stand-alone mode, as well as PC-hosted mode. The PRO MATE II device programmer is CE compliant.

The PRO MATE II device programmer has programmable VDD and VPP supplies, which allow it to verify programmed memory at VDD min and VDD max for maximum reliability. It has an LCD display for instructions and error messages, keys to enter commands and a modular detachable socket assembly to support various package types. In stand-alone mode, the PRO MATE II device programmer can read, verify, or program PICmicro devices. It can also set code protection in this mode.

14.10 PICSTART Plus Entry Level 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 all PICmicro devices with up to 40 pins. Larger pin count devices, such as the PIC16C92X and PIC17C76X, may be supported with an adapter socket. The PICSTART Plus development programmer is CE compliant.

14.11 PICDEM 1 Low Cost PICmicro Demonstration Board

The PICDEM 1 demonstration board is a simple board which demonstrates the capabilities of several of Microchip's microcontrollers. The microcontrollers supported are: 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 user can program the sample microcontrollers provided with the PICDEM 1 demonstration board on a PRO MATE II device programmer, or a PICSTART Plus development programmer, and easily test firmware. The user can also connect the PICDEM 1 demonstration board to the MPLAB ICE in-circuit emulator and download the firmware to the emulator for testing. A prototype area is available for the user to build some additional hardware and connect it to the microcontroller socket(s). Some of the features include an RS-232 interface, a potentiometer for simulated analog input, push button switches and eight LEDs connected to PORTB.

14.12 PICDEM 2 Low Cost PIC16CXX Demonstration Board

The PICDEM 2 demonstration board is a simple demonstration board that supports the PIC16C62, PIC16C64, PIC16C65, PIC16C73 and PIC16C74 microcontrollers. All the necessary hardware and software is included to run the basic demonstration programs. The user can program the sample microcontrollers provided with the PICDEM 2 demonstration board on a PRO MATE II device programmer, or a PICSTART Plus development programmer, and easily test firmware. The MPLAB ICE in-circuit emulator may also be used with the PICDEM 2 demonstration board to test firmware. A prototype area has been provided to the user for adding additional hardware and connecting it to the microcontroller socket(s). Some of the features include a RS-232 interface, push button switches, a potentiometer for simulated analog input, a serial EEPROM to demonstrate usage of the I²C™ bus and separate headers for connection to an LCD module and a keypad.

14.13 PICDEM 3 Low Cost PIC16CXXX Demonstration Board

The PICDEM 3 demonstration board is a simple demonstration board that supports the PIC16C923 and PIC16C924 in the PLCC package. It will also support future 44-pin PLCC microcontrollers with an LCD Module. All the necessary hardware and software is included to run the basic demonstration programs. The user can program the sample microcontrollers provided with the PICDEM 3 demonstration board on a PRO MATE II device programmer, or a PICSTART Plus development programmer with an adapter socket, and easily test firmware. The MPLAB ICE in-circuit emulator may also be used with the PICDEM 3 demonstration board to test firmware. A prototype area has been provided to the user for adding hardware and connecting it to the microcontroller socket(s). Some of the features include a RS-232 interface, push button switches, a potentiometer for simulated analog input, a thermistor and separate headers for connection to an external LCD module and a keypad. Also provided on the PICDEM 3 demonstration board is a LCD panel, with 4 commons and 12 segments, that is capable of displaying time, temperature and day of the week. The PICDEM 3 demonstration board provides an additional RS-232 interface and Windows software for showing the demultiplexed LCD signals on a PC. A simple serial interface allows the user to construct a hardware demultiplexer for the LCD signals.

14.14 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. All necessary hardware is included to run basic demo programs, which are supplied on a 3.5-inch disk. A programmed sample is included and the user may erase it and program it with the other sample programs using the PRO MATE II device programmer, or the PICSTART Plus development programmer, and easily debug and test the sample code. In addition, the PICDEM 17 demonstration board supports downloading of programs to and executing out of external FLASH memory on board. The PICDEM 17 demonstration board is also usable with the MPLAB ICE in-circuit emulator, or the PICMASTER emulator and all of the sample programs can be run and modified using either emulator. Additionally, a generous prototype area is available for user hardware.

14.15 KEELOQ Evaluation and Programming Tools

KEELOQ evaluation and programming tools support Microchip's HCS Secure Data Products. The HCS evaluation kit includes a LCD display to show changing codes, a decoder to decode transmissions and a programming interface to program test transmitters.

TABLE 14-1: DEVELOPMENT TOOLS FROM MICROCHIP

	PIC12CXXX	PIC14000	PIC16C5X	PIC16C6X	PIC16CXXX	PIC16F62X	PIC16C7X	PIC16C7XX	PIC16C8X	PIC16F8XX	PIC16C9XX	PIC17C4X	PIC17C7XX	PIC18CXX2	PIC18FXXX	24CXX/ 25CXX/ 93CXX	HC5XX	MCRFXX	MCP2510
Software Tools	MPLAB® Integrated Development Environment	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		
	MPLAB® C17 C Compiler											✓	✓	✓					
	MPLAB® C18 C Compiler													✓	✓				
Emulators	MPASM™ Assembler/ MPLINK™ Object Linker	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		
	MPLAB® ICE In-Circuit Emulator	✓	✓	✓	✓	✓**	✓	✓	✓	✓	✓	✓	✓	✓	✓				
	ICEPIC™ In-Circuit Emulator	✓		✓	✓		✓	✓	✓		✓								
Debugger	MPLAB® ICD In-Circuit Debugger				✓*		✓*			✓					✓				
Programmers	PICSTART® Plus Entry Level Development Programmer	✓	✓	✓	✓	✓**	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
	PRO MATE® II Universal Device Programmer	✓	✓	✓	✓	✓**	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		
Demo Boards and Eval Kits	PICDEM™ 1 Demonstration Board		✓	✓			†		✓			✓							
	PICDEM™ 2 Demonstration Board				†		†							✓	✓				
	PICDEM™ 3 Demonstration Board										✓								
	PICDEM™ 14A Demonstration Board	✓																	
	PICDEM™ 17 Demonstration Board												✓						
	KEELOQ® Evaluation Kit																✓		
	KEELOQ® Transponder Kit																✓		
	microID™ Programmer's Kit																	✓	
	125 kHz microID™ Developer's Kit																	✓	
	125 kHz Anticollision Developer's Kit																	✓	
	13.56 MHz Anticollision microID™ Developer's Kit																	✓	
	MCP2510 CAN Developer's Kit																	✓	✓

* Contact the Microchip Technology Inc. web site at www.microchip.com for information on how to use the MPLAB® ICD In-Circuit Debugger (DV164001) with PIC16C62, 63, 64, 65, 72, 73, 74, 76, 77.

** Contact Microchip Technology Inc. for availability date.

† Development tool is available on select devices.

PIC16F7X

NOTES:

15.0 ELECTRICAL CHARACTERISTICS

Absolute Maximum Ratings †

Ambient temperature under bias	-55 to +125°C
Storage temperature	-65°C to +150°C
Voltage on any pin with respect to V _{SS} (except V _{DD} , $\overline{\text{MCLR}}$, and RA4)	-0.3V to (V _{DD} + 0.3V)
Voltage on V _{DD} with respect to V _{SS}	-0.3 to +6.5V
Voltage on $\overline{\text{MCLR}}$ with respect to V _{SS} (Note 2)	0 to +13.5V
Voltage on RA4 with respect to V _{SS}	0 to +12V
Total power dissipation (Note 1)	1.0W
Maximum current out of V _{SS} pin	300 mA
Maximum current into V _{DD} pin	250 mA
Input clamp current, I _{IK} (V _I < 0 or V _I > V _{DD})	± 20 mA
Output clamp current, I _{OK} (V _O < 0 or V _O > V _{DD})	± 20 mA
Maximum output current sunk by any I/O pin	25 mA
Maximum output current sourced by any I/O pin	25 mA
Maximum current sunk by PORTA, PORTB, and PORTE (combined) (Note 3)	200 mA
Maximum current sourced by PORTA, PORTB, and PORTE (combined) (Note 3)	200 mA
Maximum current sunk by PORTC and PORTD (combined) (Note 3)	200 mA
Maximum current sourced by PORTC and PORTD (combined) (Note 3)	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 latchup. A series resistor of greater than 1 k Ω should be used to pull $\overline{\text{MCLR}}$ to V_{DD}, rather than tying the pin directly to V_{DD}.

3: PORTD and PORTE are not implemented on the PIC16F73/76 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.

PIC16F7X

FIGURE 15-1: PIC16F7X VOLTAGE-FREQUENCY GRAPH

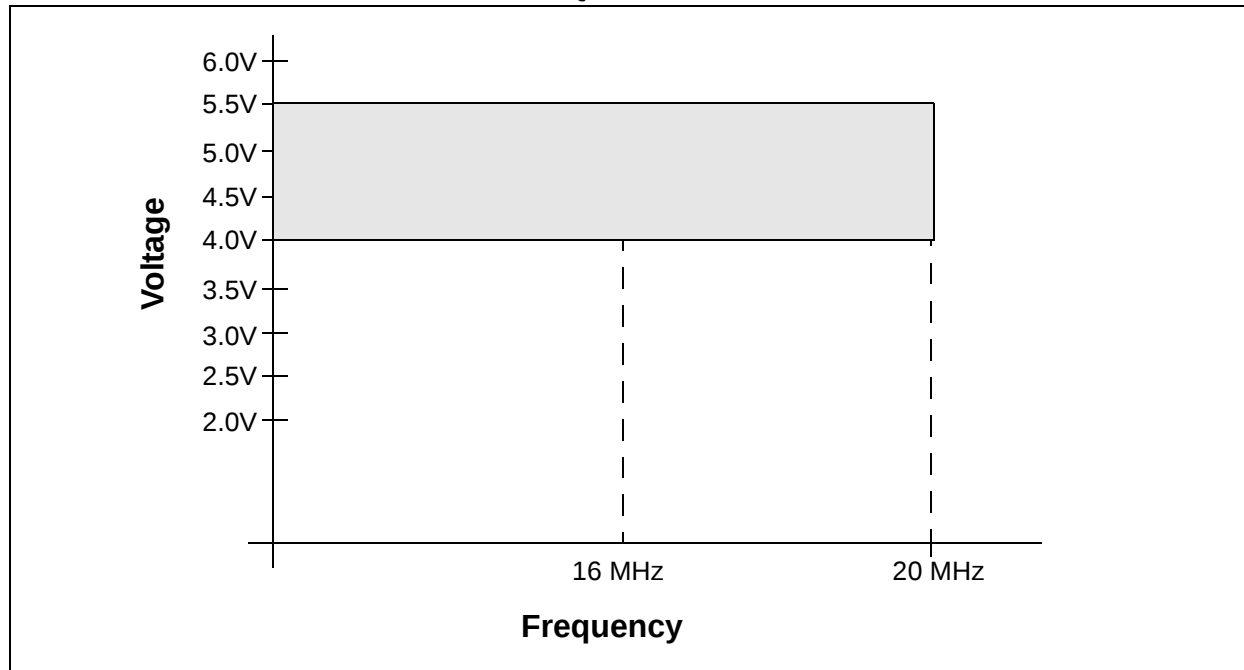
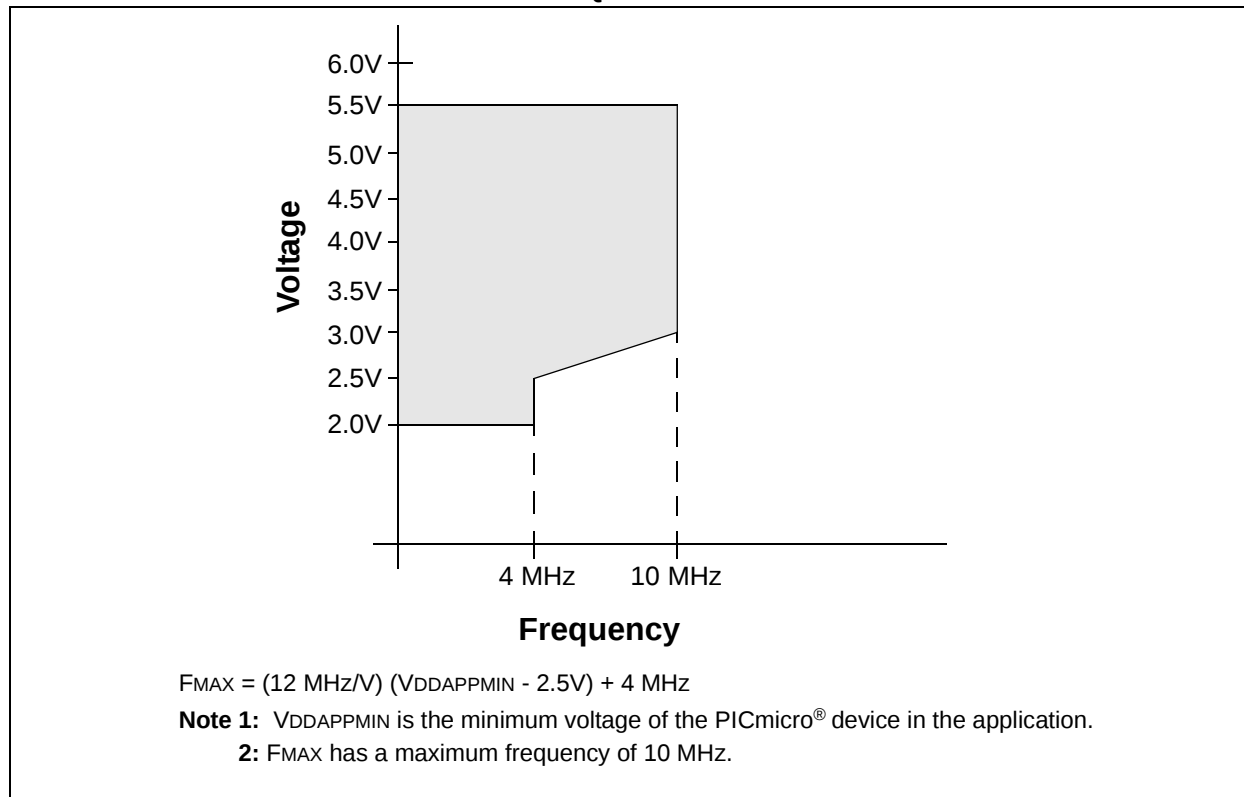


FIGURE 15-2: PIC16LF7X VOLTAGE-FREQUENCY GRAPH



15.1 DC Characteristics: PIC16F73/74/76/77 (Industrial, Extended) PIC16LF73/74/76/77 (Industrial)

PIC16LF73/74/76/77 (Industrial)		Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial					
PIC16F73/74/76/77 (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	Supply Voltage					
		PIC16F7X	2.5	—	5.5	V	A/D in use, -40°C to $+85^{\circ}\text{C}$
			2.2	—	5.5	V	A/D in use, 0°C to $+85^{\circ}\text{C}$
			2.0	—	5.5	V	A/D not used, -40°C to $+85^{\circ}\text{C}$
D001 D001A		PIC16F7X	4.0 VBOR*	- -	5.5 5.5	V V	All configurations BOR enabled (Note 7)
D002*	VDR	RAM Data Retention Voltage (Note 1)	-	1.5	-	V	
D003	VPOR	VDD Start Voltage to ensure internal Power-on Reset signal	-	VSS	-	V	See section on Power-on Reset for details
D004*	SVDD	VDD Rise Rate to ensure internal Power-on Reset signal	0.05	-	-	V/ms	See section on Power-on Reset for details
D005	VBOR	Brown-out Reset Voltage	3.65	4.0	4.35	V	BODEN bit in configuration word enabled

Legend: Shading of rows is to assist in readability of the table.

* These parameters are characterized but not tested.

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

Note 1: This is the limit to which VDD can be lowered without losing RAM data.

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.

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 hi-impedance state and tied to VDD and VSS.

4: For RC osc 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 kOhm.

5: Timer1 oscillator (when enabled) adds approximately 20 μA to the specification. This value is from characterization and is for design guidance only. This is not tested.

6: 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.

7: When BOR is enabled, the device will operate correctly until the VBOR voltage trip point is reached.

PIC16F7X

15.1 DC Characteristics: PIC16F73/74/76/77 (Industrial, Extended) PIC16LF73/74/76/77 (Industrial) (Continued)

PIC16LF73/74/76/77 (Industrial)			Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial				
PIC16F73/74/76/77 (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.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
D010 D010A	IDD	Supply Current (Notes 2, 5)					
		PIC16LF7X	—	0.4	2.0	mA	XT, RC osc configuration FOSC = 4 MHz, VDD = 3.0V (Note 4)
D010 D013		PIC16F7X	-	0.9	4	mA	XT, RC osc configuration FOSC = 4 MHz, VDD = 5.5V (Note 4)
			—	5.2	15	mA	HS osc configuration FOSC = 20 MHz, VDD = 5.5V
D015*	ΔIBOR	Brown-out Reset Current (Note 6)	—	25	200	μA	BOR enabled, VDD = 5.0V
D020 D021	IPD	Power-down Current (Notes 3, 5)					
		PIC16LF7X	—	2.0	30	μA	VDD = 3.0V, WDT enabled, -40°C to +85°C
D020 D021 D021A		PIC16F7X	—	0.1	5	μA	VDD = 3.0V, WDT disabled, -40°C to +85°C
			—	5.0	42	μA	VDD = 4.0V, WDT enabled, -40°C to +85°C
			—	0.1	19	μA	VDD = 4.0V, WDT disabled, -40°C to +85°C
			—	10.5	57	μA	VDD = 4.0V, WDT enabled, -40°C to +125°C
D023*	ΔIBOR	Brown-out Reset Current (Note 6)	—	25	200	μA	BOR enabled, VDD = 5.0V

Legend: Shading of rows is to assist in readability of the table.

* These parameters are characterized but not tested.

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

Note 1: This is the limit to which VDD can be lowered without losing RAM data.

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 hi-impedance state and tied to VDD and VSS.

Note 4: For RC osc 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 kOhm.

Note 5: Timer1 oscillator (when enabled) adds approximately 20 μA to the specification. This value is from characterization and is for design guidance only. This is not tested.

Note 6: 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 7: When BOR is enabled, the device will operate correctly until the VBOR voltage trip point is reached.

15.2 DC Characteristics: PIC16F73/74/76/77 (Industrial, Extended) PIC16LF73/74/76/77 (Industrial)

DC CHARACTERISTICS				Standard Operating Conditions (unless otherwise stated)			
				Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ for extended			
				Operating voltage V_{DD} range as described in DC Specification, Section 15.1.			
Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
D030 D030A D031 D032 D033	V_{IL}	Input Low Voltage					
		I/O ports:					
		with TTL buffer	V_{SS}	—	$0.15V_{DD}$	V	For entire V_{DD} range
			V_{SS}	—	0.8V	V	$4.5V \leq V_{DD} \leq 5.5V$
		with Schmitt Trigger buffer	V_{SS}	—	$0.2V_{DD}$	V	
		$\overline{\text{MCLR}}$, OSC1 (in RC mode)	V_{SS}	—	$0.2V_{DD}$	V	(Note 1)
		OSC1 (in XT and LP mode)	V_{SS}	—	0.3V	V	
D040 D040A D041 D042 D042A D043	V_{IH}	Input High Voltage					
		I/O ports:					
		with TTL buffer	2.0	—	V_{DD}	V	$4.5V \leq V_{DD} \leq 5.5V$
			$0.25V_{DD} + 0.8V$	—	V_{DD}	V	For entire V_{DD} range
		with Schmitt Trigger buffer	$0.8V_{DD}$	—	V_{DD}	V	For entire V_{DD} range
		$\overline{\text{MCLR}}$	$0.8V_{DD}$	—	V_{DD}	V	
		OSC1 (in XT and LP mode)	1.6V	—	V_{DD}	V	
D042A D043		OSC1 (in HS mode)	$0.7V_{DD}$	—	V_{DD}	V	
		OSC1 (in RC mode)	$0.9V_{DD}$	—	V_{DD}	V	(Note 1)
D070	I_{PURB}	PORTB Weak Pull-up Current	50	250	400	μA	$V_{DD} = 5V$, $V_{PIN} = V_{SS}$
D060 D061 D063	I_{IL}	Input Leakage Current (Notes 2, 3)					
		I/O ports	—	—	± 1	μA	$V_{SS} \leq V_{PIN} \leq V_{DD}$, pin at hi-impedance
		$\overline{\text{MCLR}}$, RA4/T0CKI	—	—	± 5	μA	$V_{SS} \leq V_{PIN} \leq V_{DD}$
		OSC1	—	—	± 5	μA	$V_{SS} \leq V_{PIN} \leq V_{DD}$, XT, HS and LP osc configuration

* These parameters are characterized but not tested.

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

- Note 1:** In RC oscillator configuration, the OSC1/CLKIN pin is a Schmitt Trigger input. It is not recommended that the PIC16F7X 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.

PIC16F7X

15.2 DC Characteristics: PIC16F73/74/76/77 (Industrial, Extended) PIC16LF73/74/76/77 (Industrial) (Continued)

DC CHARACTERISTICS			Standard Operating Conditions (unless otherwise stated)				
			Operating temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended				
			Operating voltage VDD range as described in DC Specification, Section 15.1.				
Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
D080 D083	VOL	Output Low Voltage					
		I/O ports	—	—	0.6	V	IoL = 8.5 mA, VDD = 4.5V, -40°C to +125°C
		OSC2/CLKOUT (RC osc config)	—	—	0.6	V	IoL = 1.6 mA, VDD = 4.5V, -40°C to +125°C
			—	—	0.6	V	IoL = 1.2 mA, VDD = 4.5V, -40°C to +125°C
D090 D092	VOH	Output High Voltage					
		I/O ports (Note 3)	VDD - 0.7	—	—	V	IoH = -3.0 mA, VDD = 4.5V, -40°C to +125°C
		OSC2/CLKOUT (RC osc config)	VDD - 0.7	—	—	V	IoH = -1.3 mA, VDD = 4.5V, -40°C to +125°C
			VDD - 0.7	—	—	V	IoH = -1.0 mA, VDD = 4.5V, -40°C to +125°C
D150*	VOD	Open Drain High Voltage	—	—	12	V	RA4 pin
D100 D101 D102	Cosc2 Cio Cb	Capacitive Loading Specs on Output Pins					
		OSC2 pin	—	—	15	pF	In XT, HS and LP modes when external clock is used to drive OSC1
		All I/O pins and OSC2 (in RC mode)	—	—	50	pF	
SCL, SDA in I²C mode	—	—	400	pF			
D130 D131	EP VPR	Program FLASH Memory					
		Endurance	100	1000	—	E/W	25°C at 5V
		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/CLKIN pin is a Schmitt Trigger input. It is not recommended that the PIC16F7X be driven with external clock in RC mode.
- 2:** The leakage current on the MCLR pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.
- 3:** Negative current is defined as current sourced by the pin.

15.3 Timing Parameter Symbology

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

1. TppS2ppS
2. TppS
3. TCC:ST (I²C specifications only)
4. Ts (I²C specifications only)

T			
F	Frequency	T	Time

Lowercase letters (pp) and their meanings:

pp			
cc	CCP1	osc	OSC1
ck	CLKOUT	rd	$\overline{RD}$
cs	$\overline{CS}$	rw	$\overline{RD}$ or $\overline{WR}$
di	SDI	sc	SCK
do	SDO	ss	$\overline{SS}$
dt	Data in	t0	T0CKI
io	I/O port	t1	T1CKI
mc	$\overline{MCLR}$	wr	$\overline{WR}$

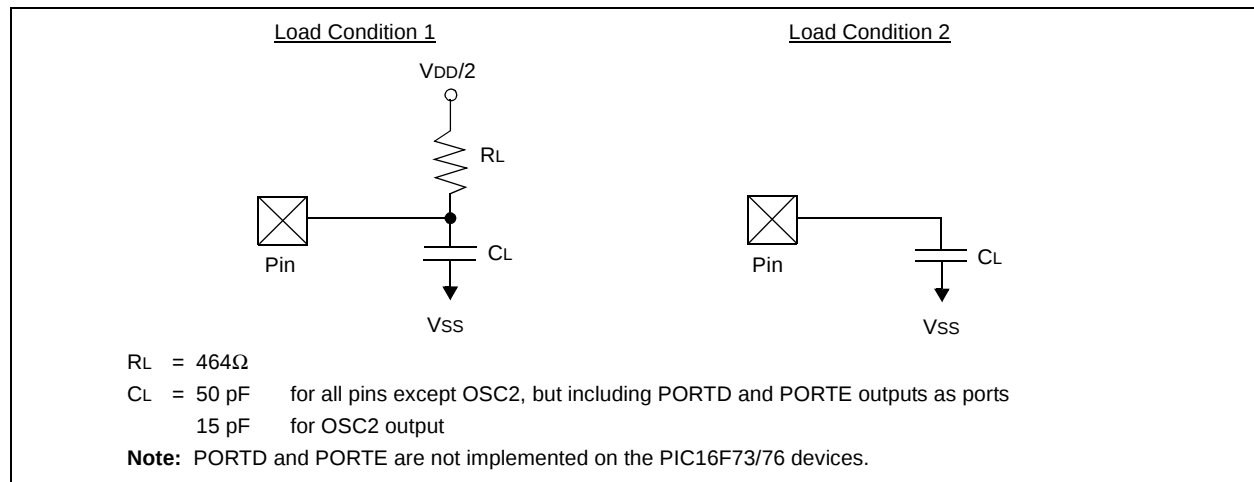
Uppercase letters and their meanings:

S			
F	Fall	P	Period
H	High	R	Rise
I	Invalid (Hi-impedance)	V	Valid
L	Low	Z	Hi-impedance
I²C only			
AA	output access	High	High
BUF	Bus free	Low	Low

TCC:ST (I²C specifications only)

CC			
HD	Hold	SU	Setup
ST			
DAT	DATA input hold	STO	STOP condition
STA	START condition		

FIGURE 15-3: LOAD CONDITIONS



PIC16F7X

FIGURE 15-4: EXTERNAL CLOCK TIMING

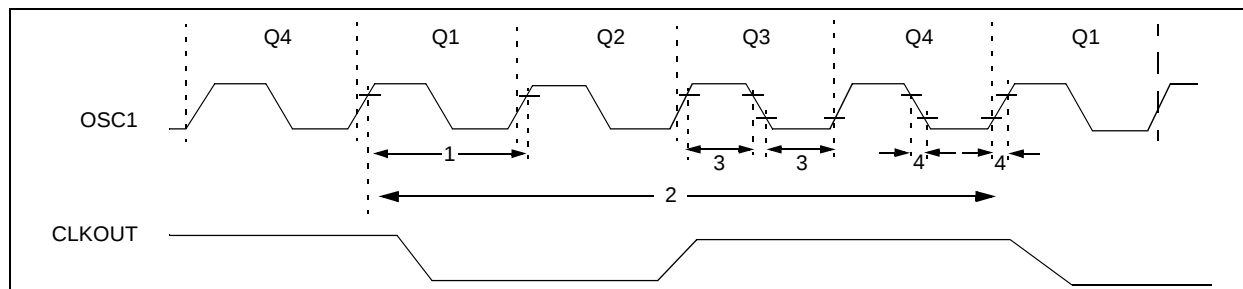


TABLE 15-1: EXTERNAL CLOCK TIMING REQUIREMENTS

Parameter No.	Symbol	Characteristic	Min	Typ†	Max	Units	Conditions
	Fosc	External CLKIN Frequency (Note 1)	DC	—	1	MHz	XT osc mode
			DC	—	20	MHz	HS osc mode
			DC	—	32	kHz	LP osc mode
		Oscillator Frequency (Note 1)	DC	—	4	MHz	RC osc mode
			0.1	—	4	MHz	XT osc mode
			4	—	20	MHz	HS osc mode
			5	—	200	kHz	LP osc mode
1	Tosc	External CLKIN Period (Note 1)	1000	—	—	ns	XT osc mode
			50	—	—	ns	HS osc mode
			5	—	—	ms	LP osc mode
		Oscillator Period (Note 1)	250	—	—	ns	RC osc mode
			250	—	10,000	ns	XT osc mode
			50	—	250	ns	HS osc mode
			5	—	—	ms	LP osc 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/CLKIN pin. When an external clock input is used, the "max." cycle time limit is "DC" (no clock) for all devices.

FIGURE 15-5: CLKOUT AND I/O TIMING

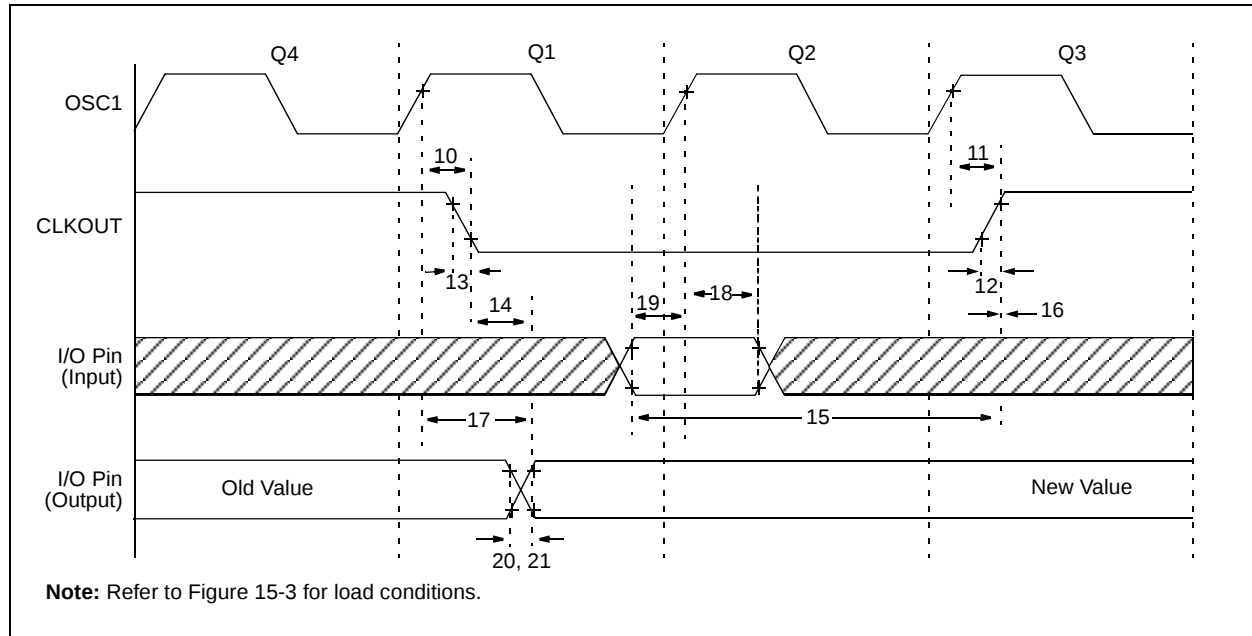


TABLE 15-2: CLKOUT AND I/O TIMING REQUIREMENTS

Param No.	Symbol	Characteristic	Min	Typ†	Max	Units	Conditions
10*	TosH2ckL	OSC1↑ to CLKOUT↓	—	75	200	ns	(Note 1)
11*	TosH2ckH	OSC1↑ to CLKOUT↑	—	75	200	ns	(Note 1)
12*	TckR	CLKOUT rise time	—	35	100	ns	(Note 1)
13*	TckF	CLKOUT fall time	—	35	100	ns	(Note 1)
14*	TckL2ioV	CLKOUT↓ to Port out valid	—	—	0.5T _{CY} + 20	ns	(Note 1)
15*	TioV2ckH	Port in valid before CLKOUT↑	T _{OSC} + 200	—	—	ns	(Note 1)
16*	TckH2ioI	Port in hold after CLKOUT↑	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)	Standard (F)	100	—	ns	
			Extended (LF)	200	—	ns	
19*	TioV2osH	Port input valid to OSC1↑ (I/O in setup time)	0	—	—	ns	
20*	TioR	Port output rise time	Standard (F)	—	10	ns	
			Extended (LF)	—	—	145	ns
21*	TioF	Port output fall time	Standard (F)	—	10	ns	
			Extended (LF)	—	—	145	ns
22††	Tinp	INT pin high or low time	T _{CY}	—	—	ns	
23††	Trbp	RB7:RB4 change INT high or low time	T _{CY}	—	—	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 CLKOUT output is 4 x T_{OSC}.

PIC16F7X

FIGURE 15-6: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING

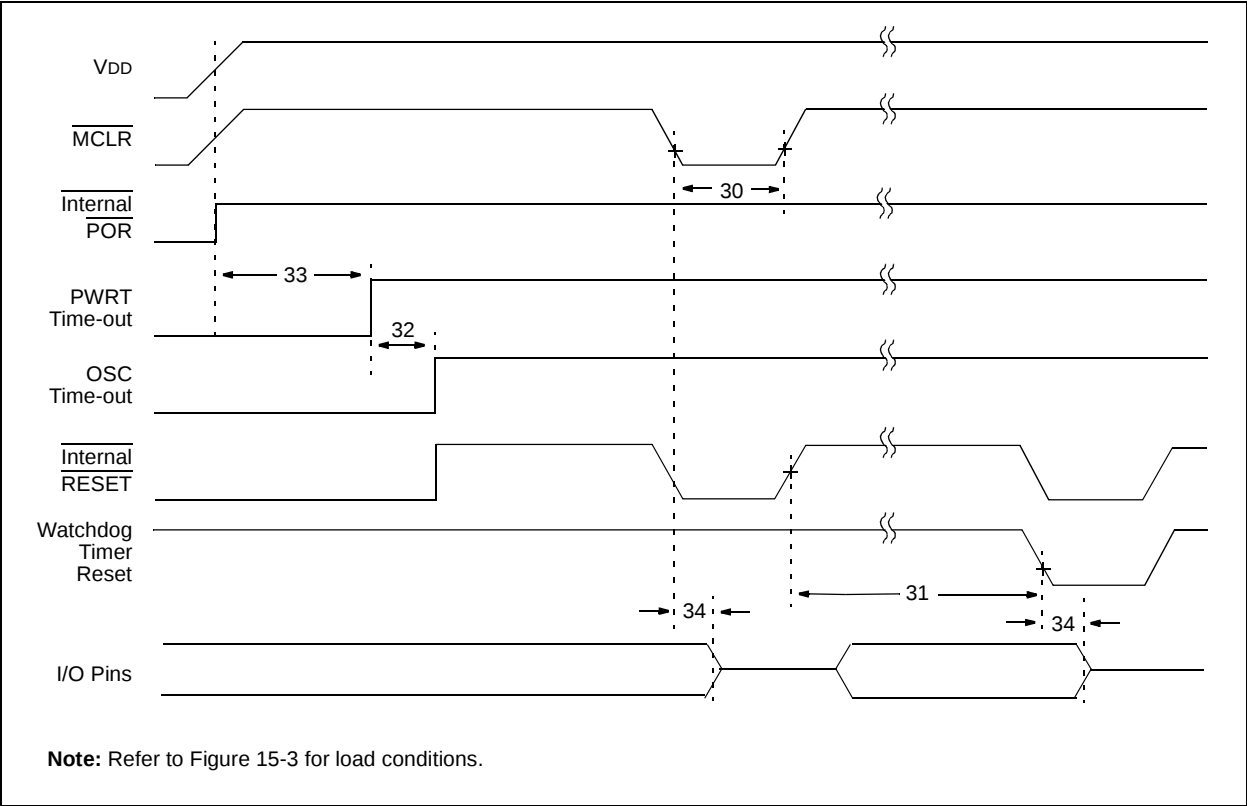


FIGURE 15-7: BROWN-OUT RESET TIMING

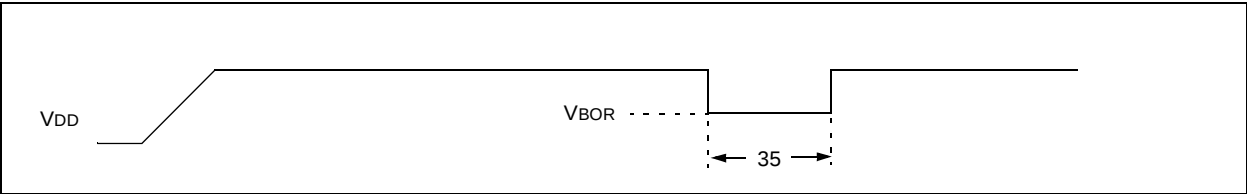


TABLE 15-3: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER, POWER-UP TIMER, AND BROWN-OUT RESET REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
30	TmCL	MCLR Pulse Width (low)	2	—	—	μs	VDD = 5V, -40°C to +85°C
31*	TWDT	Watchdog Timer Time-out Period (No Prescaler)	7	18	33	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	28	72	132	ms	VDD = 5V, -40°C to +85°C
34	TIOZ	I/O Hi-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.

FIGURE 15-8: TIMER0 AND TIMER1 EXTERNAL CLOCK TIMINGS

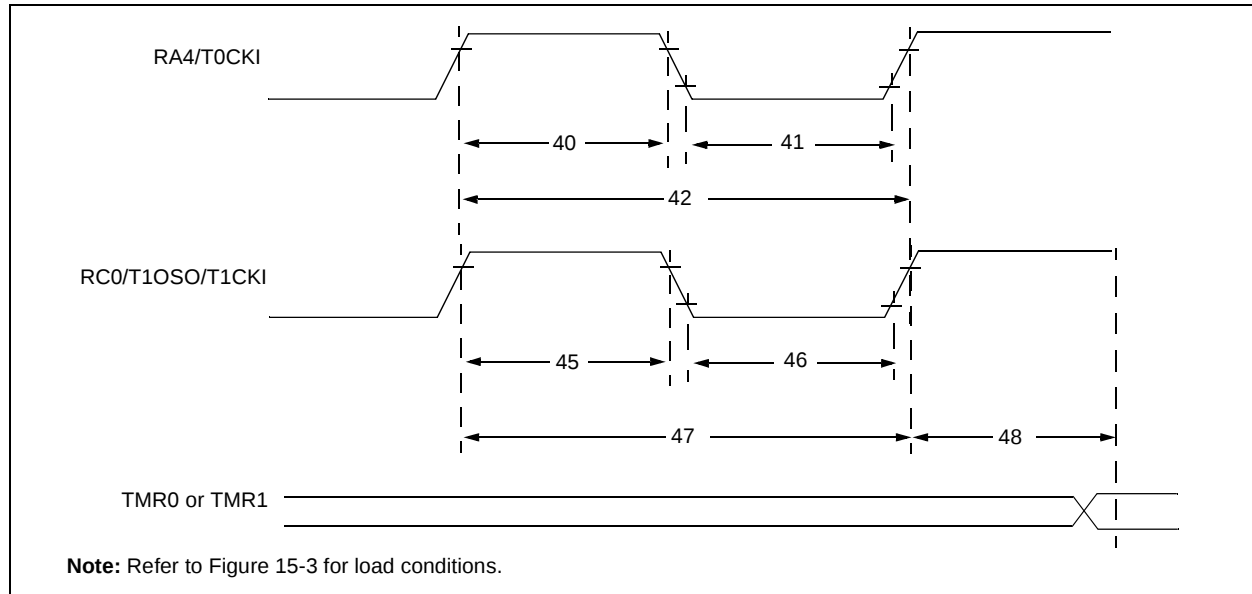


TABLE 15-4: TIMER0 AND TIMER1 EXTERNAL CLOCK REQUIREMENTS

Param No.	Symbol	Characteristic			Min	Typ†	Max	Units	Conditions
40*	Tt0H	T0CKI High Pulse Width		No Prescaler	0.5Tcy + 20	—	—	ns	Must also meet parameter 42
				With Prescaler	10	—	—	ns	
41*	Tt0L	T0CKI Low Pulse Width		No Prescaler	0.5Tcy + 20	—	—	ns	Must also meet parameter 42
				With Prescaler	10	—	—	ns	
42*	Tt0P	T0CKI Period		No Prescaler	Tcy + 40	—	—	ns	N = prescale value (2, 4, ..., 256)
				With Prescaler	Greater of: 20 or $\frac{Tcy + 40}{N}$	—	—	ns	
45*	Tt1H	T1CKI High Time	Synchronous, Prescaler = 1		0.5Tcy + 20	—	—	ns	Must also meet parameter 47
			Synchronous, Prescaler = 2,4,8	Standard(F)	15	—	—	ns	
				Extended(LF)	25	—	—	ns	
			Asynchronous	Standard(F)	30	—	—	ns	
				Extended(LF)	50	—	—	ns	
46*	Tt1L	T1CKI Low Time	Synchronous, Prescaler = 1		0.5Tcy + 20	—	—	ns	Must also meet parameter 47
			Synchronous, Prescaler = 2,4,8	Standard(F)	15	—	—	ns	
				Extended(LF)	25	—	—	ns	
			Asynchronous	Standard(F)	30	—	—	ns	
				Extended(LF)	50	—	—	ns	
47*	Tt1P	T1CKI Input Period	Synchronous	Standard(F)	Greater of: 30 or $\frac{Tcy + 40}{N}$	—	—	ns	N = prescale value (1, 2, 4, 8)
				Extended(LF)	Greater of: 50 or $\frac{Tcy + 40}{N}$				N = prescale value (1, 2, 4, 8)
			Asynchronous	Standard(F)	60	—	—	ns	
				Extended(LF)	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 Tosc	—	7 Tosc	—	

* These parameters are characterized but not tested.

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

PIC16F7X

FIGURE 15-9: CAPTURE/COMPARE/PWM TIMINGS (CCP1 AND CCP2)

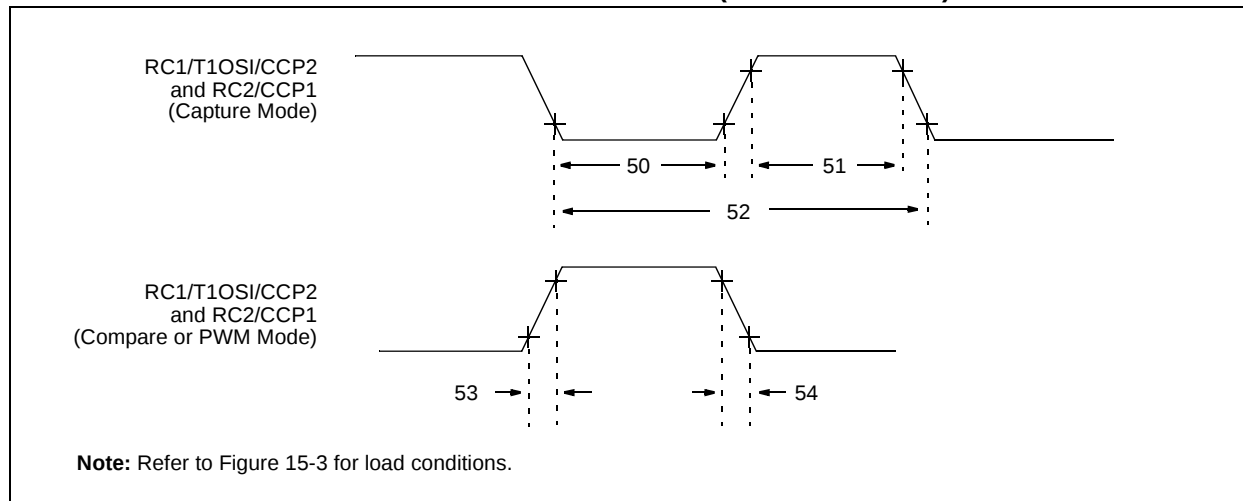


TABLE 15-5: CAPTURE/COMPARE/PWM REQUIREMENTS (CCP1 AND CCP2)

Param No.	Symbol	Characteristic		Min	Typ†	Max	Units	Conditions	
50*	TccL	CCP1 and CCP2 input low time	No Prescaler		0.5Tcy + 20	—	—	ns	
			With Prescaler	Standard(F)	10	—	—	ns	
				Extended(LF)	20	—	—	ns	
51*	TccH	CCP1 and CCP2 input high time	No Prescaler		0.5Tcy + 20	—	—	ns	
			With Prescaler	Standard(F)	10	—	—	ns	
				Extended(LF)	20	—	—	ns	
52*	TccP	CCP1 and CCP2 input period			$\frac{3Tcy + 40}{N}$	—	—	ns	N = prescale value (1,4 or 16)
53*	TccR	CCP1 and CCP2 output rise time	Standard(F)	—	10	25	ns		
			Extended(LF)	—	25	50	ns		
54*	TccF	CCP1 and CCP2 output fall time	Standard(F)	—	10	25	ns		
			Extended(LF)	—	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.

FIGURE 15-10: PARALLEL SLAVE PORT TIMING (PIC16F74/77 DEVICES ONLY)

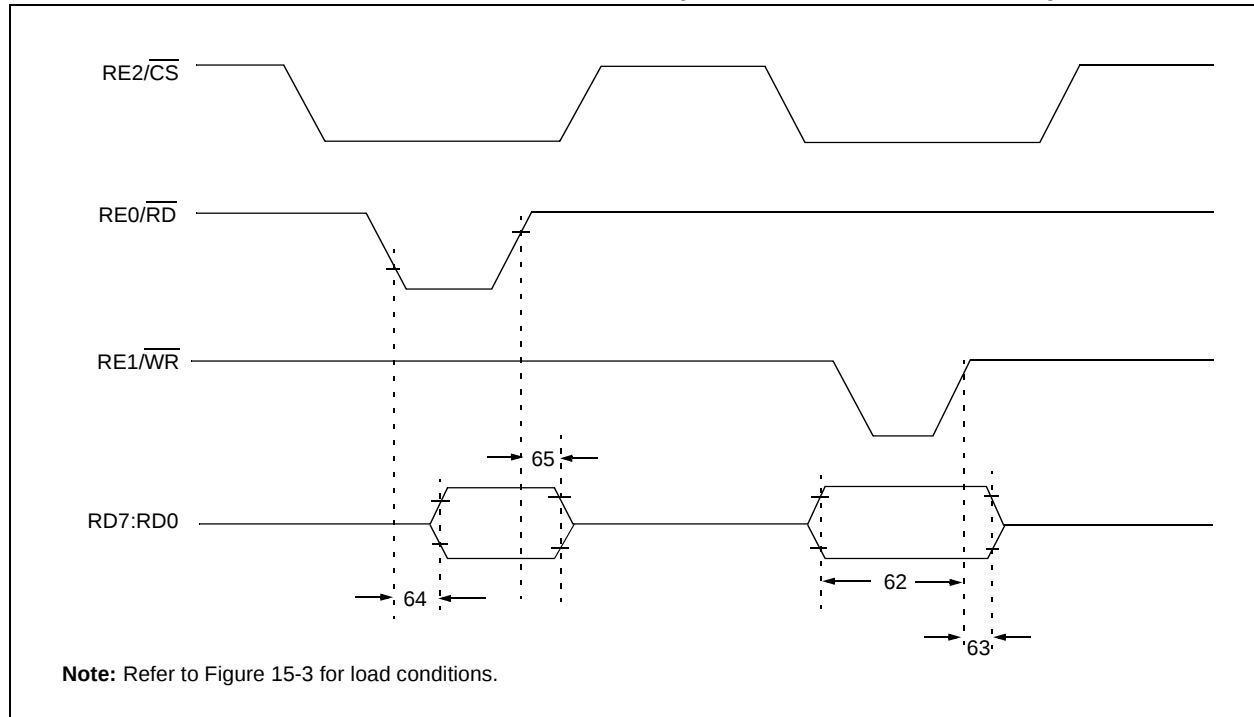


TABLE 15-6: PARALLEL SLAVE PORT REQUIREMENTS (PIC16F74/77 DEVICES ONLY)

Parameter 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 ns	Extended range only
63*	TwrH2dtI	$\overline{WR}\uparrow$ or $\overline{CS}\uparrow$ to data in invalid (hold time)	Standard(F) Extended(LF)	20 35	— —	ns ns	
64	TrdL2dtV	$\overline{RD}\downarrow$ and $\overline{CS}\downarrow$ to data out valid	— —	— —	80 90	ns 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.

PIC16F7X

FIGURE 15-11: SPI MASTER MODE TIMING (CKE = 0, SMP = 0)

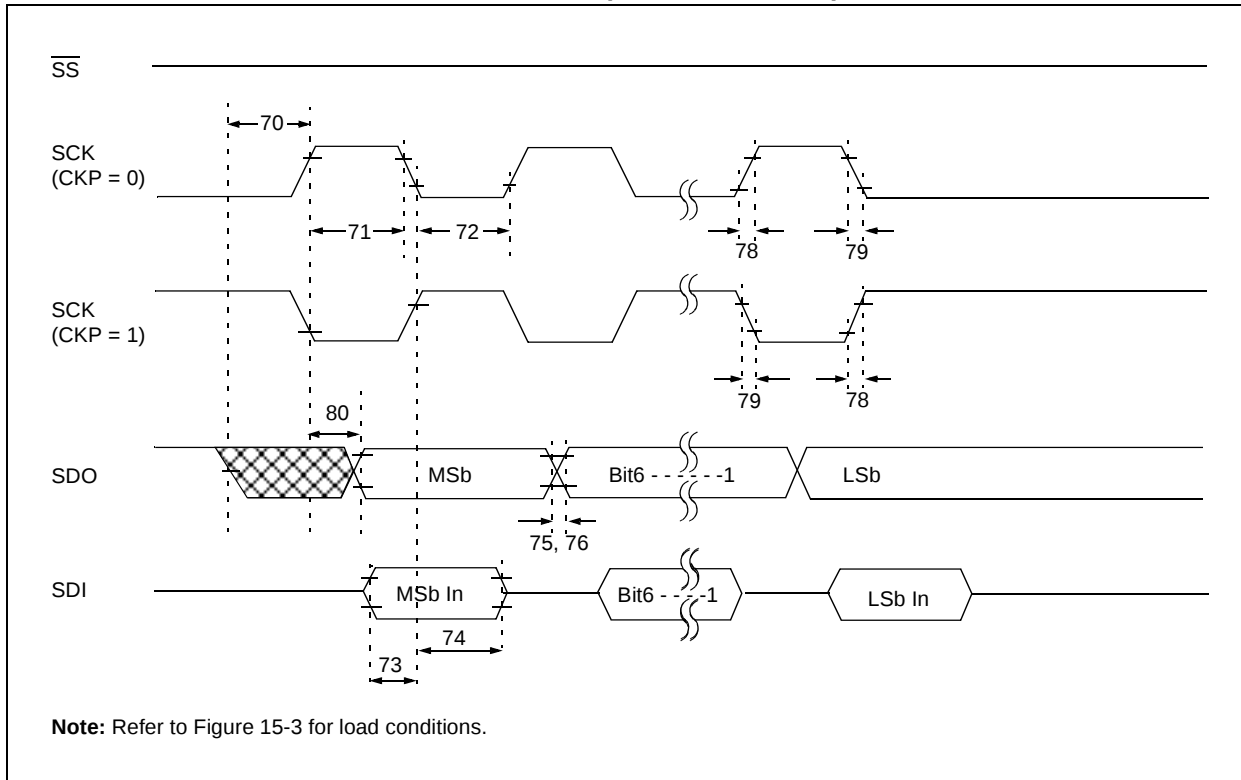


FIGURE 15-12: SPI MASTER MODE TIMING (CKE = 1, SMP = 1)

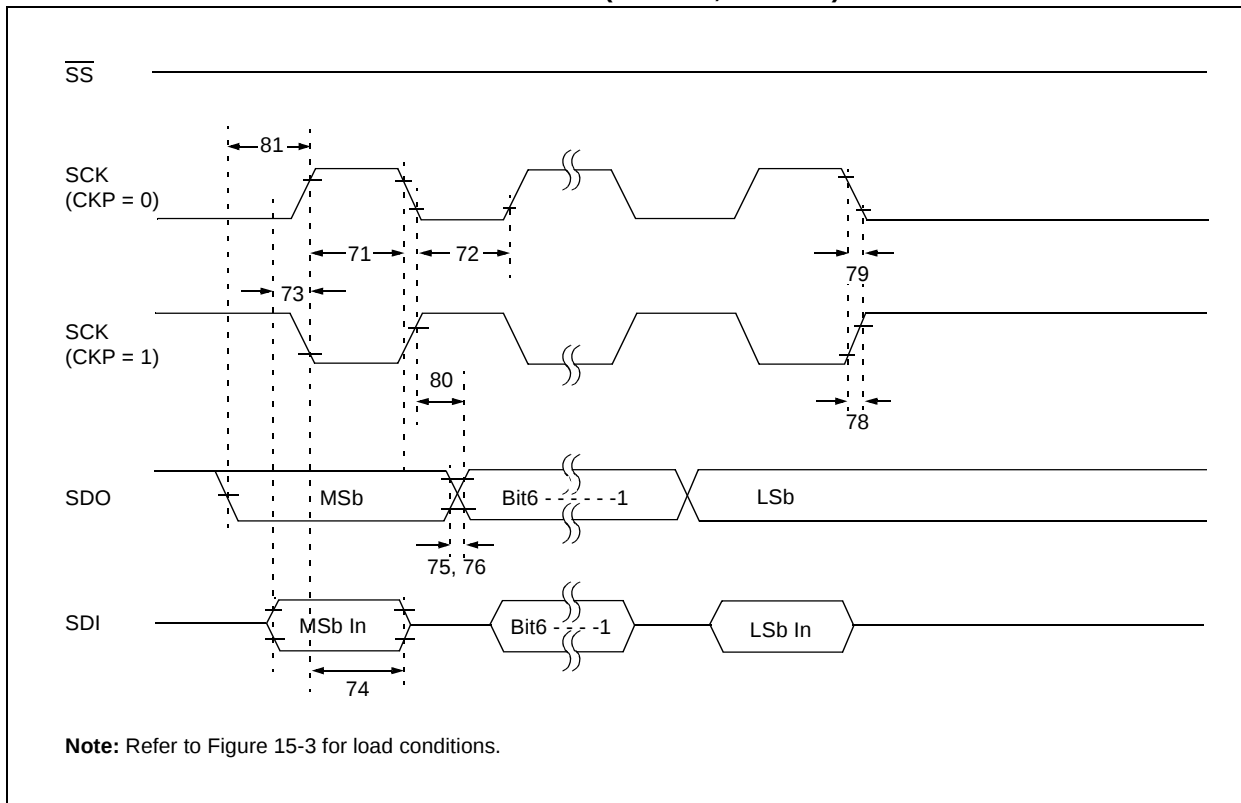


FIGURE 15-13: SPI SLAVE MODE TIMING (CKE = 0)

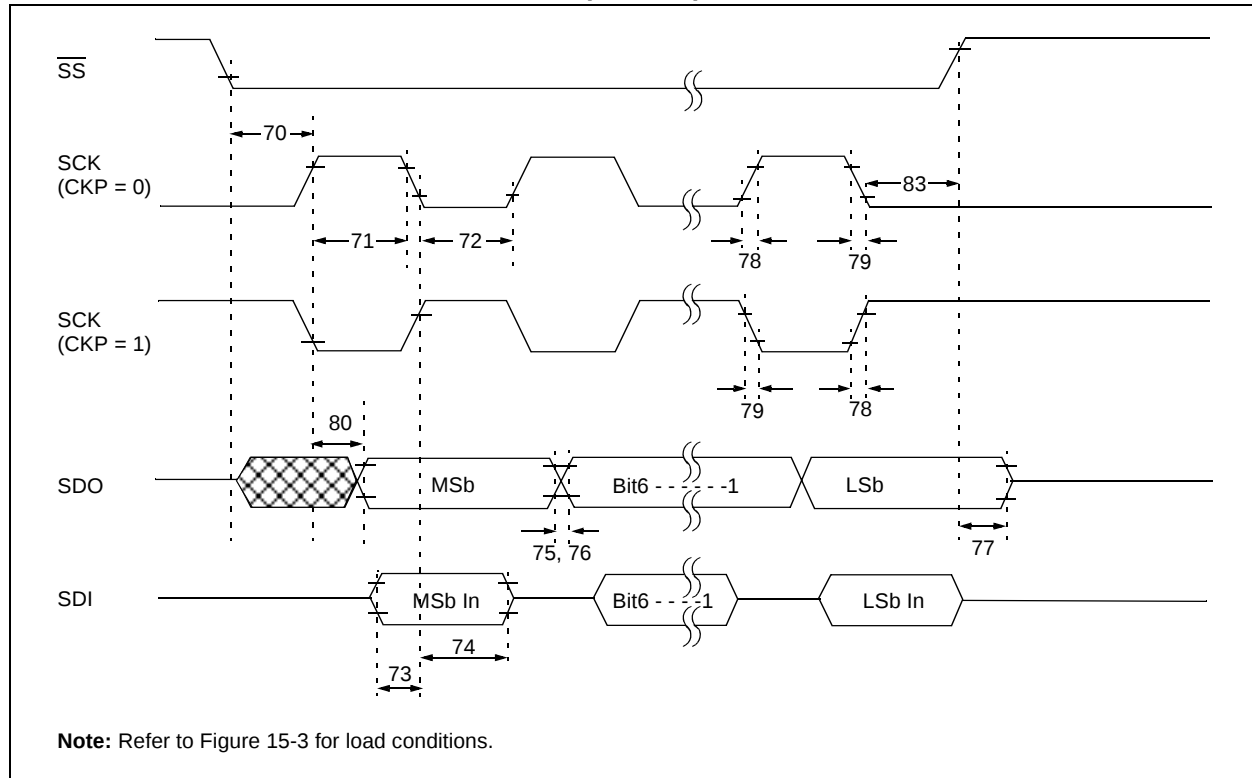
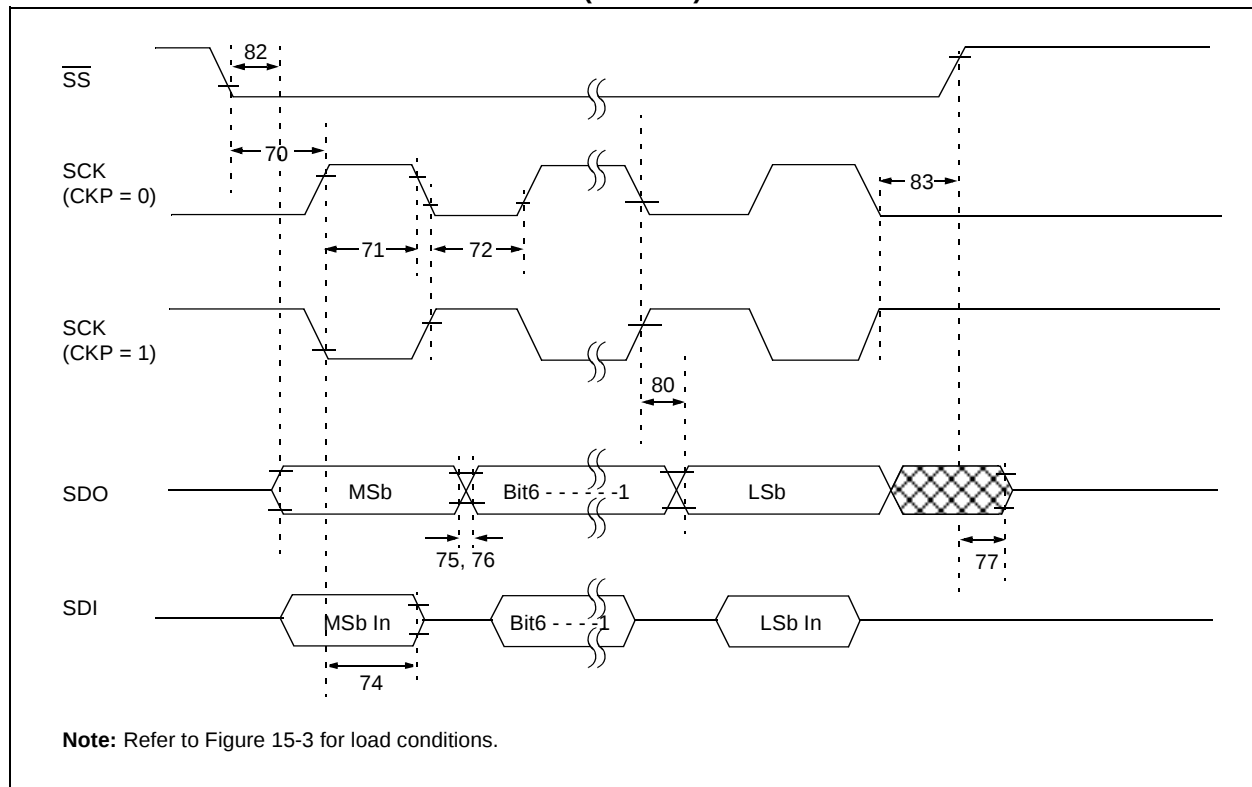


FIGURE 15-14: SPI SLAVE MODE TIMING (CKE = 1)



PIC16F7X

TABLE 15-7: 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	Standard(F) Extended(LF)	10 25	25 50	ns ns	
76*	TdoF	SDO data output fall time	—	10	25	ns	
77*	TssH2doZ	$\overline{SS}\uparrow$ to SDO output hi-impedance	10	—	50	ns	
78*	TscR	SCK output rise time (Master mode)	Standard(F) Extended(LF)	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	Standard(F) Extended(LF)	— —	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	1.5Tcy + 40	—	—	ns	

* These parameters are characterized but not tested.

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

FIGURE 15-15: I²C BUS START/STOP BITS TIMING

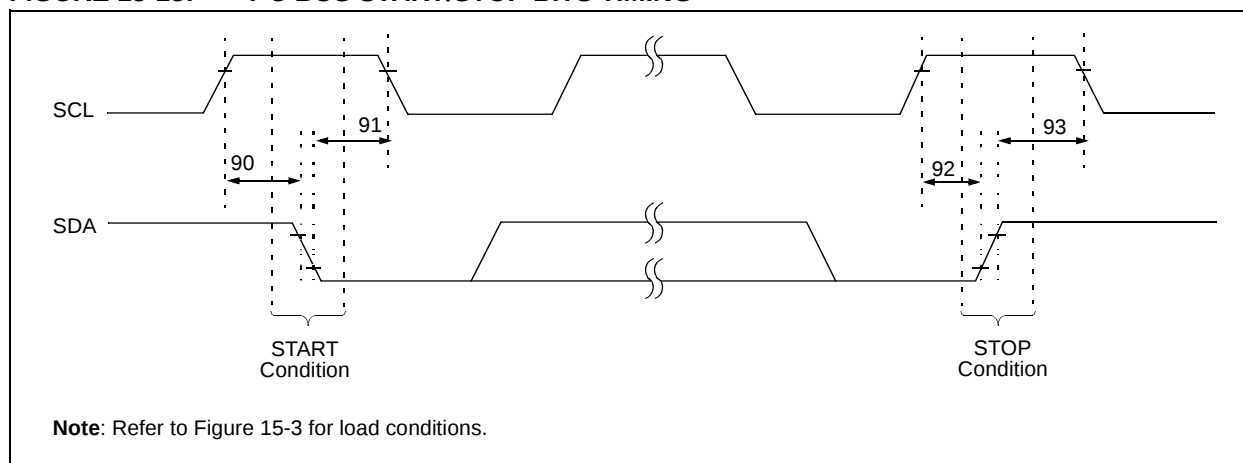
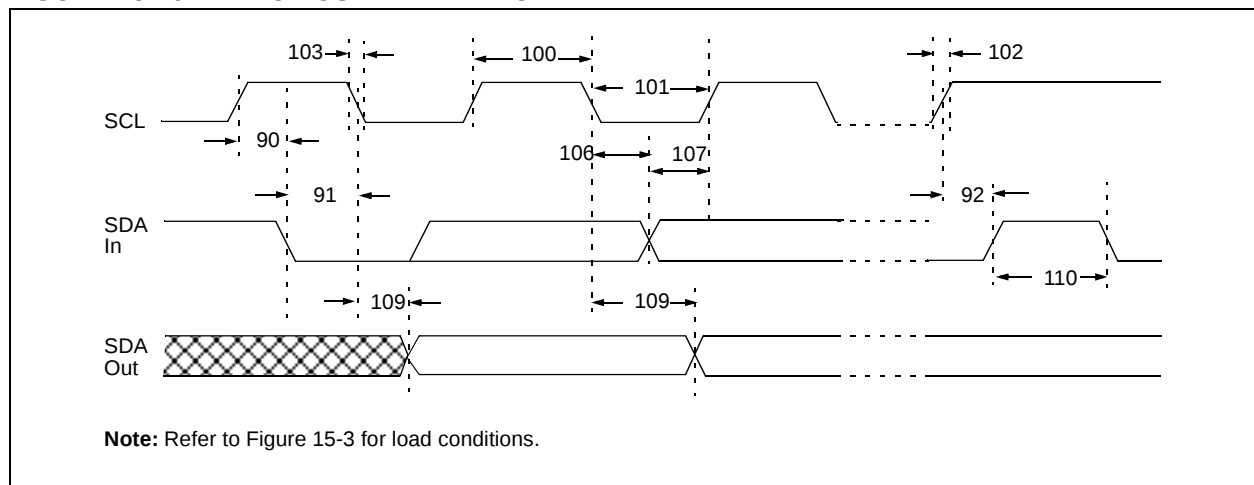


TABLE 15-8: I²C BUS START/STOP BITS REQUIREMENTS

Param No.	Symbol	Characteristic		Min	Typ	Max	Units	Conditions
90*	TSU:STA	START condition Setup time	100 kHz mode	4700	—	—	ns	Only relevant for Repeated START condition
			400 kHz mode	600	—	—		
91*	THD:STA	START condition Hold time	100 kHz mode	4000	—	—	ns	After this period, the first clock pulse is generated
			400 kHz mode	600	—	—		
92*	TSU:STO	STOP condition Setup time	100 kHz mode	4700	—	—	ns	
			400 kHz mode	600	—	—		
93	THD:STO	STOP condition Hold time	100 kHz mode	4000	—	—	ns	
			400 kHz mode	600	—	—		

* These parameters are characterized but not tested.

FIGURE 15-16: I²C BUS DATA TIMING



PIC16F7X

TABLE 15-9: I²C 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	1.5TCY	—		
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	1.5TCY	—		
102*	TR	SDA and SCL rise time	100 kHz mode	—	1000	ns	
			400 kHz mode	20 + 0.1CB	300	ns	CB is specified to be from 10 - 400 pF
103*	TF	SDA and SCL fall time	100 kHz mode	—	300	ns	
			400 kHz mode	20 + 0.1CB	300	ns	CB is specified to be from 10 - 400 pF
90*	TSU:STA	START condition setup time	100 kHz mode	4.7	—	μs	Only relevant for Repeated START condition
			400 kHz mode	0.6	—	μs	
91*	THD:STA	START condition hold time	100 kHz mode	4.0	—	μs	After this period the first clock pulse is generated
			400 kHz mode	0.6	—	μs	
106*	THD:DAT	Data input hold time	100 kHz mode	0	—	ns	
			400 kHz mode	0	0.9	μs	
107*	TSU:DAT	Data input setup time	100 kHz mode	250	—	ns	(Note 2)
			400 kHz mode	100	—	ns	
92*	TSU:STO	STOP condition setup time	100 kHz mode	4.7	—	μs	
			400 kHz mode	0.6	—	μs	
109*	TAA	Output valid from clock	100 kHz mode	—	3500	ns	(Note 1)
			400 kHz mode	—	—	ns	
110*	TBUF	Bus free time	100 kHz mode	4.7	—	μs	Time the bus must be free before a new transmission can start
			400 kHz mode	1.3	—	μs	
	CB	Bus capacitive loading		—	400	pF	

* These parameters are characterized but not tested.

- Note 1:** As a transmitter, the device must provide this internal minimum delay time to bridge the undefined region (min. 300 ns) of the falling edge of SCL to avoid unintended generation of START or STOP conditions.
- Note 2:** A Fast mode (400 kHz) I²C bus device can be used in a Standard mode (100 kHz) I²C bus system, but the requirement T_{SU: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 T_R max. + T_{SU:DAT} = 1000 + 250 = 1250 ns (according to the Standard mode I²C bus specification), before the SCL line is released.

FIGURE 15-17: USART SYNCHRONOUS TRANSMISSION (MASTER/SLAVE) TIMING

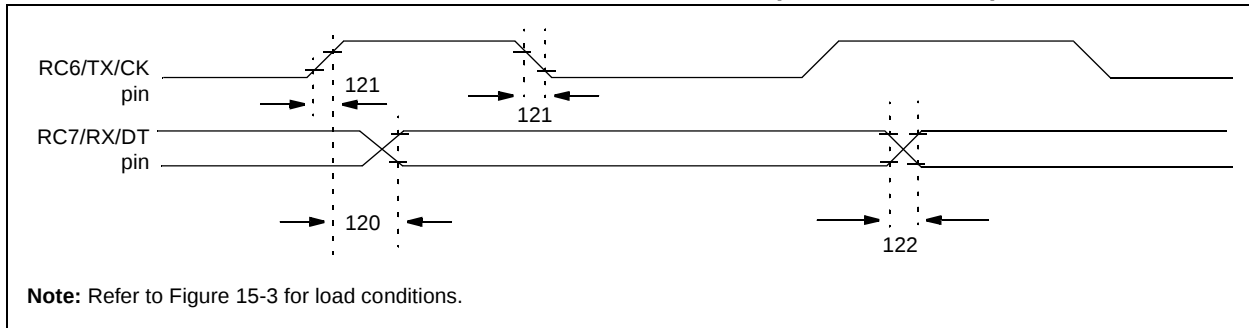


TABLE 15-10: USART SYNCHRONOUS TRANSMISSION REQUIREMENTS

Param No.	Symbol	Characteristic	Min	Typ†	Max	Units	Conditions
120	TckH2dV	<u>SYNC XMIT (MASTER & SLAVE)</u> Clock high to data out valid	—	—	80	ns	
			—	—	100	ns	
121	Tckrf	Clock out rise time and fall time (Master mode)	—	—	45	ns	
			—	—	50	ns	
122	Tdtrf	Data out rise time and fall time	—	—	45	ns	
			—	—	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 15-18: USART SYNCHRONOUS RECEIVE (MASTER/SLAVE) TIMING

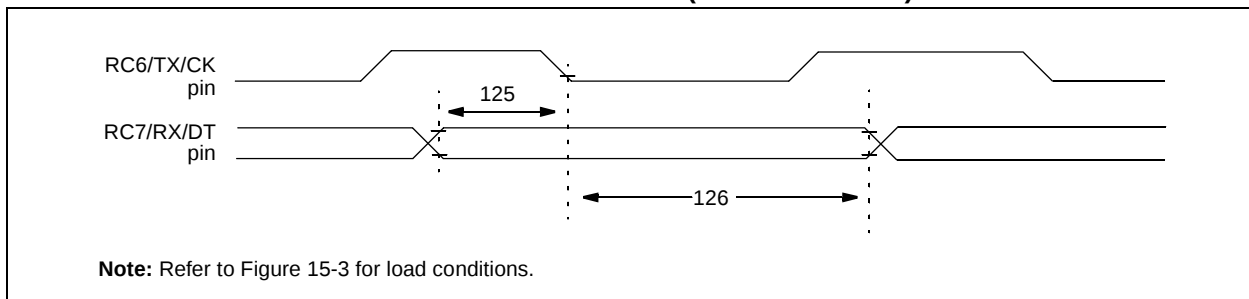


TABLE 15-11: USART SYNCHRONOUS RECEIVE REQUIREMENTS

Parameter No.	Symbol	Characteristic	Min	Typ†	Max	Units	Conditions
125	TdtV2ckL	<u>SYNC RCV (MASTER & SLAVE)</u> Data setup before CK↓ (DT setup time)	15	—	—	ns	
126	TckL2dtH	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.

PIC16F7X

**TABLE 15-12: A/D CONVERTER CHARACTERISTICS: PIC16F7X (INDUSTRIAL, EXTENDED)
PIC16LF7X (INDUSTRIAL)**

Param No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions
A01	NR	Resolution	PIC16F7X	—	—	8 bits	bit	VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF
			PIC16LF7X	—	—	8 bits	bit	VREF = VDD = 2.2V
A02	EABS	Total absolute error		—	—	< ±1	LSb	VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF
A03	EIL	Integral linearity error		—	—	< ±1	LSb	VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF
A04	EDL	Differential linearity error		—	—	< ±1	LSb	VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF
A05	EFS	Full scale error		—	—	< ±1	LSb	VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF
A06	EOFF	Offset error		—	—	< ±1	LSb	VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF
A10	—	Monotonicity (Note 3)		—	guaranteed	—	—	VSS ≤ VAIN ≤ VREF
A20	VREF	Reference voltage		2.5	—	5.5	V	-40°C to +125°C
				2.2	—	5.5	V	0°C to +125°C
A25	VAIN	Analog input voltage		VSS - 0.3	—	VREF + 0.3	V	
A30	ZAIN	Recommended impedance of analog voltage source		—	—	10.0	kΩ	
A40	IAD	A/D conversion current (VDD)	PIC16F7X	—	180	—	μA	Average current consumption when A/D is on (Note 1).
			PIC16LF7X	—	90	—	μA	
A50	IREF	VREF input current (Note 2)		N/A —	— —	±5 500	μA μA	During VAIN acquisition. During A/D Conversion cycle.

* These parameters are characterized but not tested.

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

Note 1: When A/D is off, it will not consume any current other than minor leakage current. The power-down current spec includes any such leakage from the A/D module.

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

3: The A/D conversion result never decreases with an increase in the input voltage and has no missing codes.

FIGURE 15-19: A/D CONVERSION TIMING

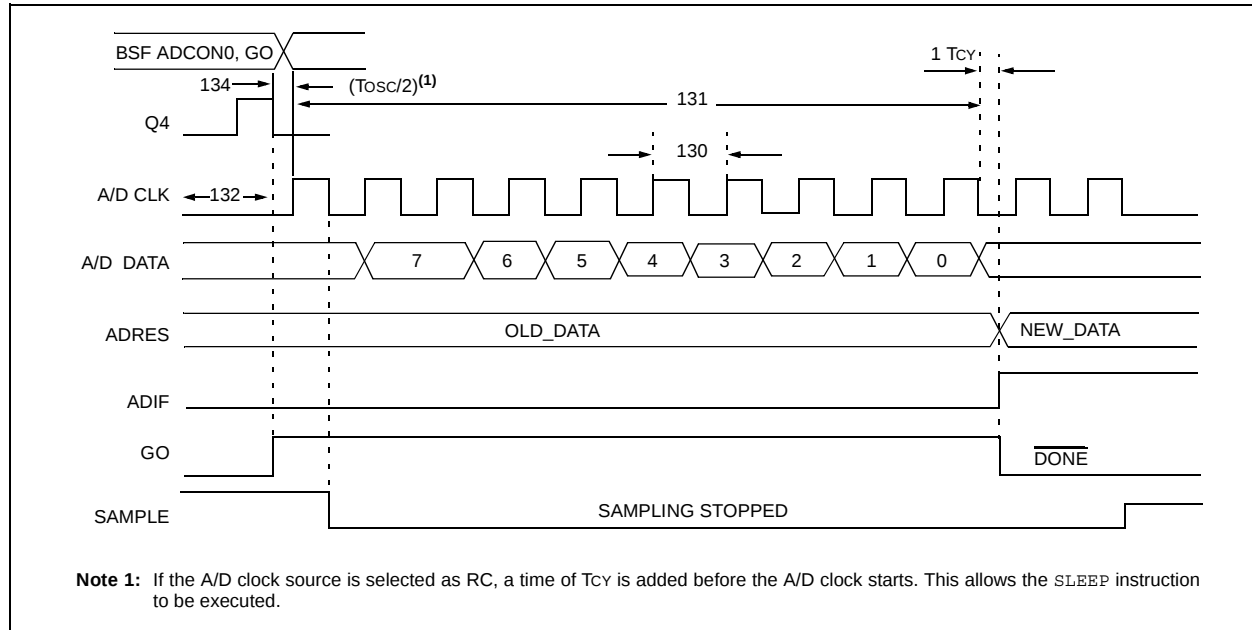


TABLE 15-13: A/D CONVERSION REQUIREMENTS

Param No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions
130	TAD	A/D clock period	PIC16F7X	1.6	—	—	μs	TOSC based, $V_{REF} \geq 3.0V$
			PIC16LF7X	2.0	—	—	μs	TOSC based, $2.0V \leq V_{REF} \leq 5.5V$
			PIC16F7X	2.0	4.0	6.0	μs	A/D RC mode
			PIC16LF7X	3.0	6.0	9.0	μs	A/D RC mode
131	TCNV	Conversion time (not including S/H time) (Note 1)		9	—	9	TAD	
132	TACQ	Acquisition time		5*	—	—	μs	The minimum time is the amplifier settling time. This may be used if the "new" input voltage has not changed by more than 1 LSb (i.e., 20.0 mV @ 5.12V) from the last sampled voltage (as stated on CHOLD).
134	TGO	Q4 to A/D clock start		—	$T_{osc}/2$	—	—	If the A/D clock source is selected as RC, a time of T_{cy} is added before the A/D clock starts. This allows the <code>SLEEP</code> instruction to be executed.

* These parameters are characterized but not tested.

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

Note 1: ADRES register may be read on the following T_{cy} cycle.

2: See Section 11.1 for minimum conditions.

PIC16F7X

NOTES:

16.0 DC AND AC CHARACTERISTICS GRAPHS AND TABLES

Note: The graphs and tables provided following this note are a statistical summary based on a limited number of samples and are provided for informational purposes only. The performance characteristics listed herein are not tested or guaranteed. In some graphs or tables, the data presented may be outside the specified operating range (e.g., outside specified power supply range) and therefore, outside the warranted range.

“Typical” represents the mean of the distribution at 25°C. “Maximum” or “minimum” represents (mean + 3 σ) or (mean - 3 σ) respectively, where σ is a standard deviation, over the whole temperature range.

FIGURE 16-1: TYPICAL I_{DD} vs. F_{osc} OVER V_{DD} (HS MODE)

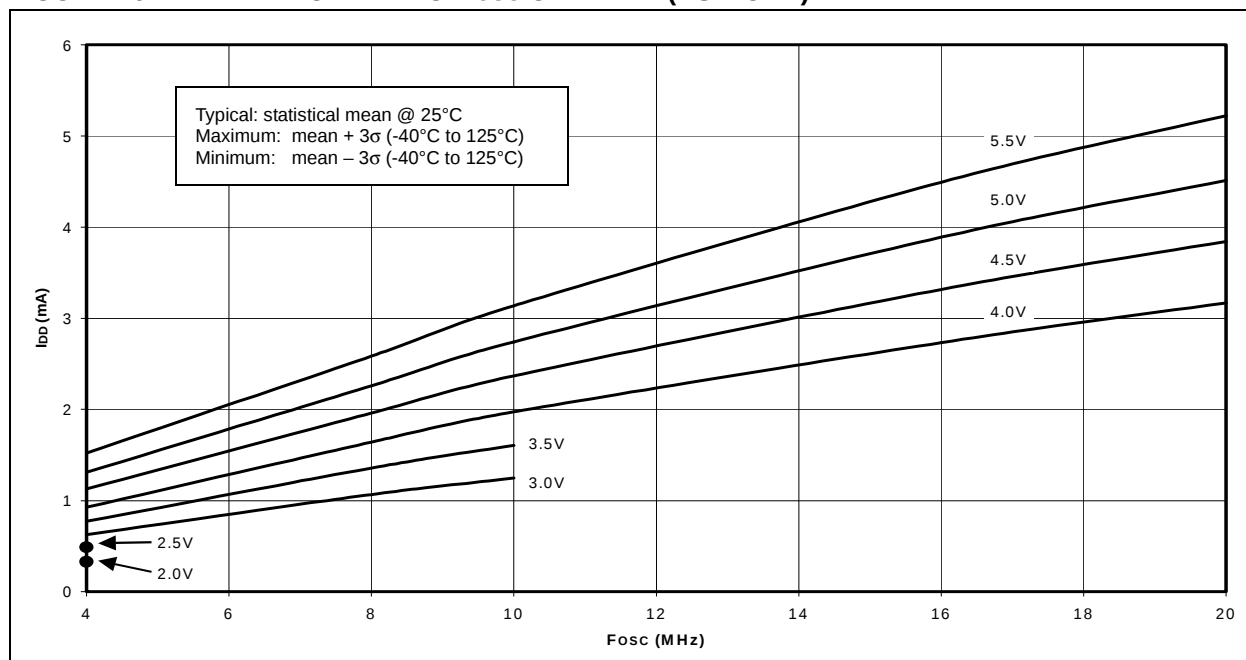
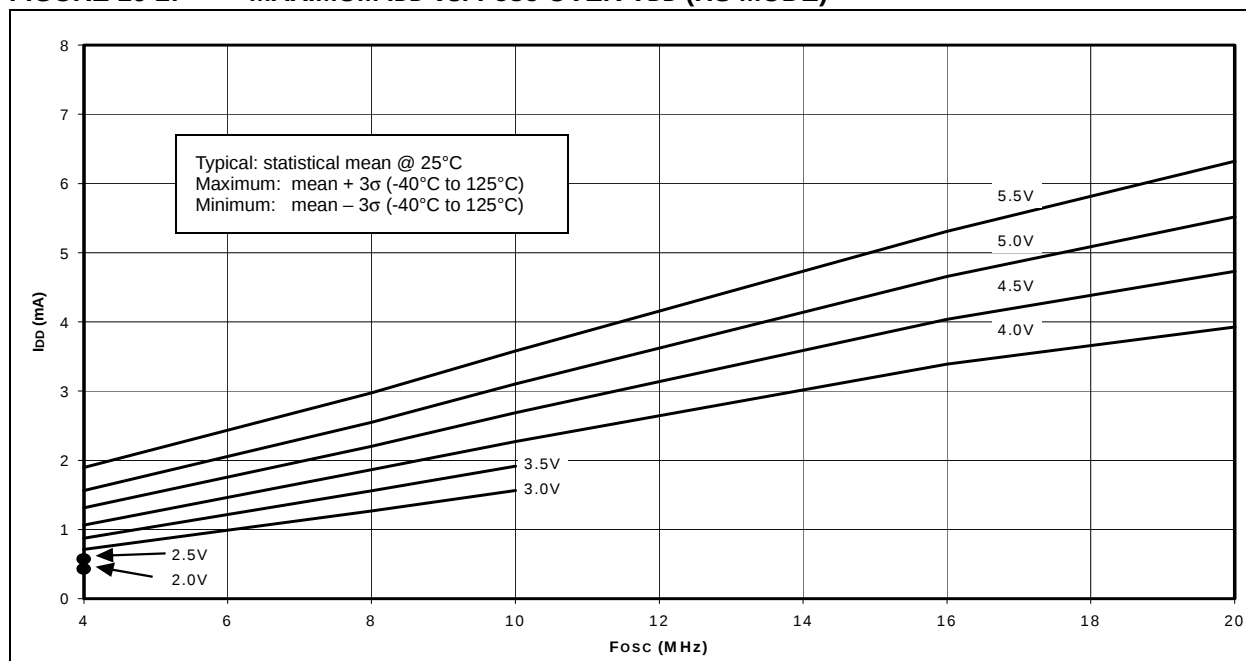


FIGURE 16-2: MAXIMUM I_{DD} vs. F_{osc} OVER V_{DD} (HS MODE)



PIC16F7X

FIGURE 16-3: TYPICAL I_{DD} vs. F_{osc} OVER V_{DD} (XT MODE)

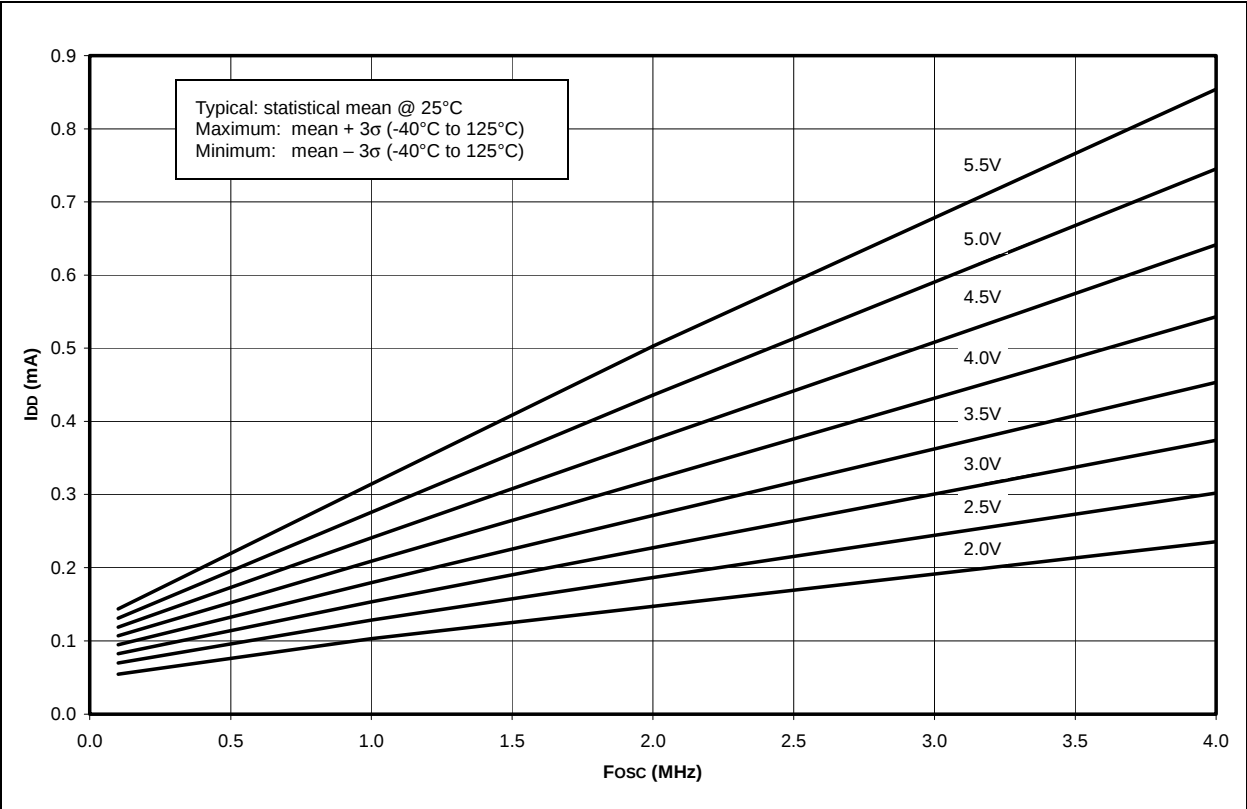


FIGURE 16-4: MAXIMUM I_{DD} vs. F_{osc} OVER V_{DD} (XT MODE)

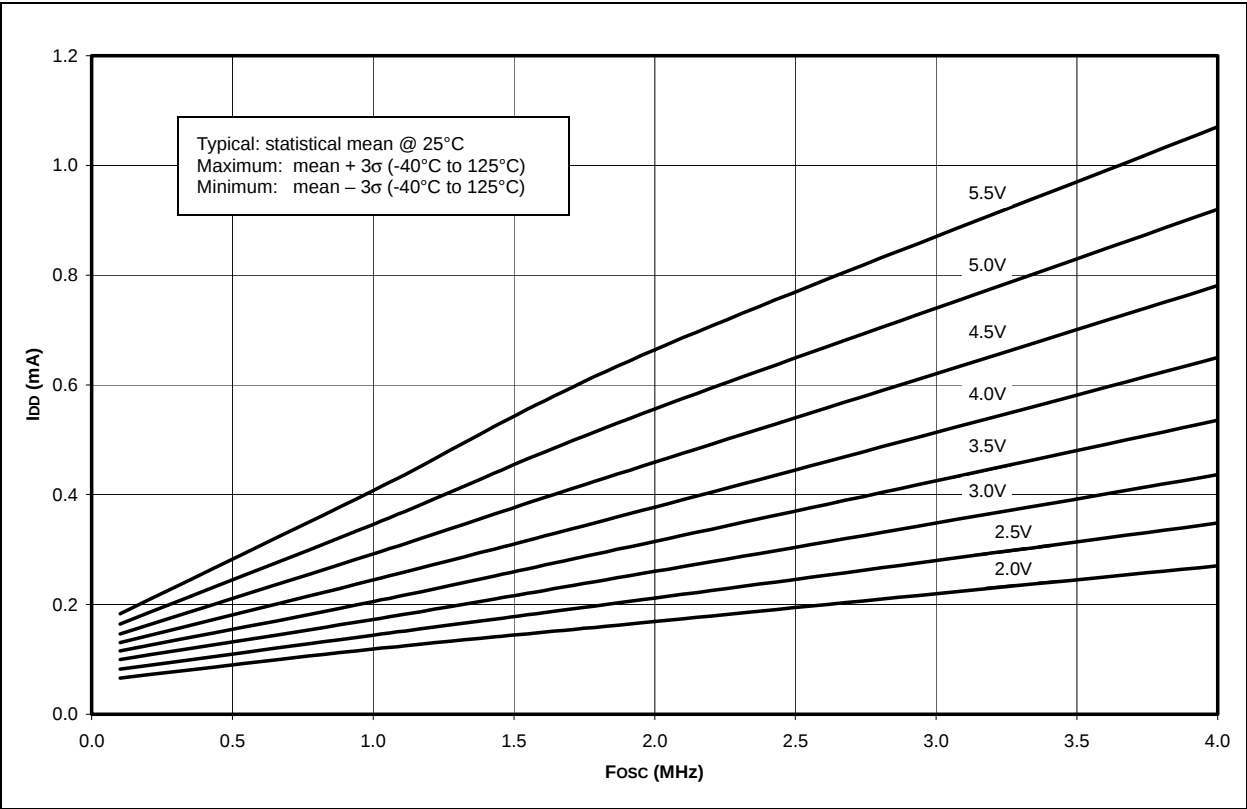


FIGURE 16-5: TYPICAL I_{DD} vs. F_{OSC} OVER V_{DD} (LP MODE)

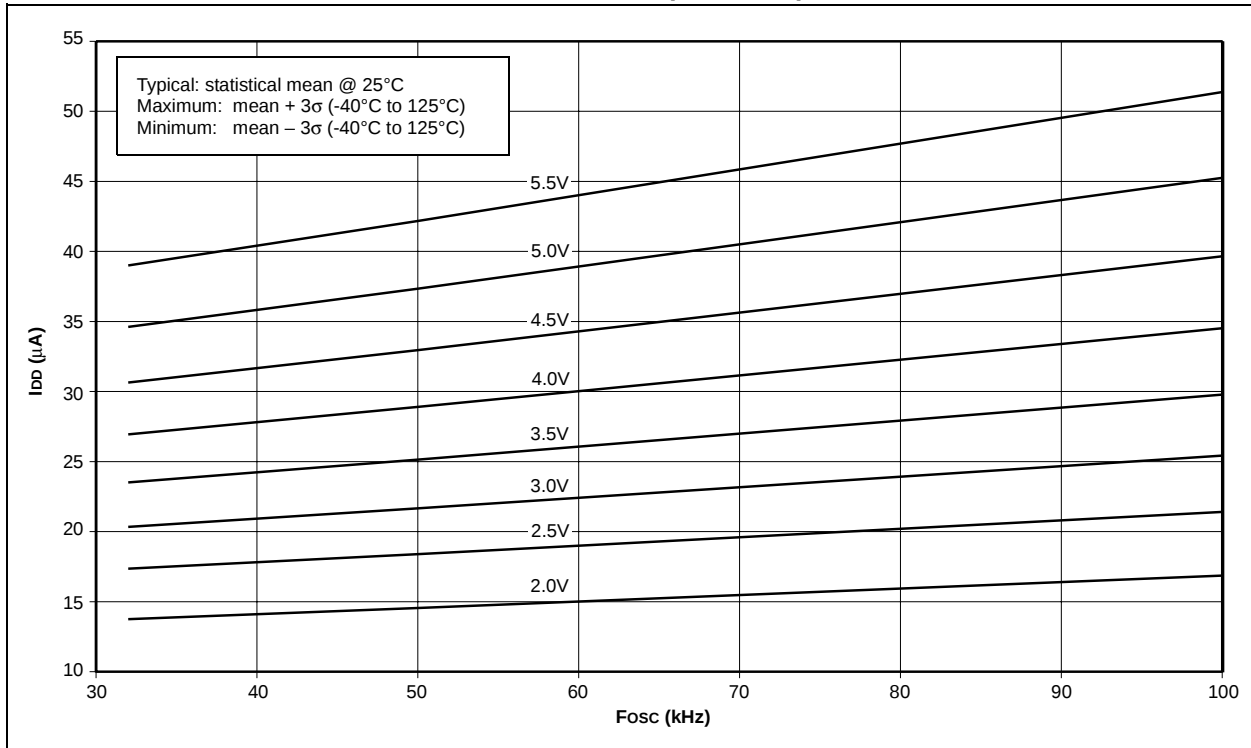
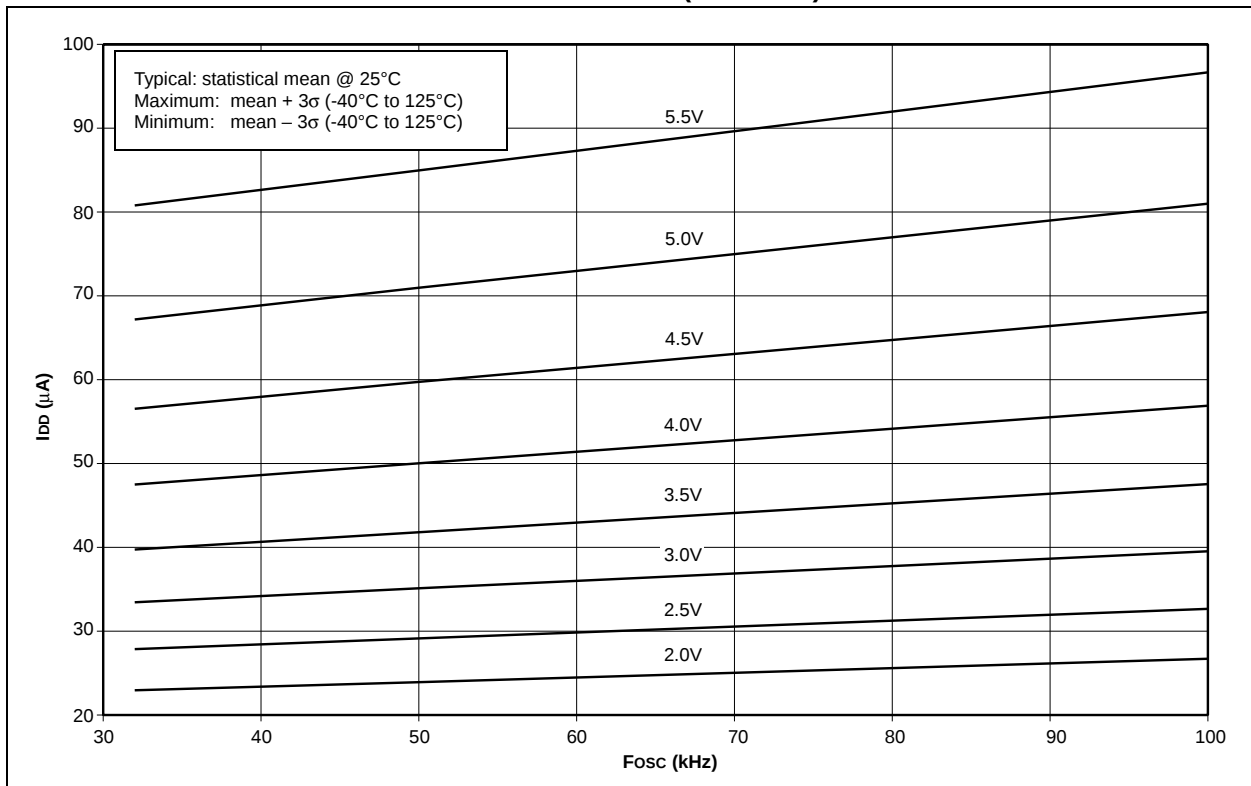


FIGURE 16-6: MAXIMUM I_{DD} vs. F_{OSC} OVER V_{DD} (LP MODE)



PIC16F7X

FIGURE 16-7: AVERAGE Fosc vs. VDD FOR VARIOUS VALUES OF R (RC MODE, C = 20 pF, 25°C)

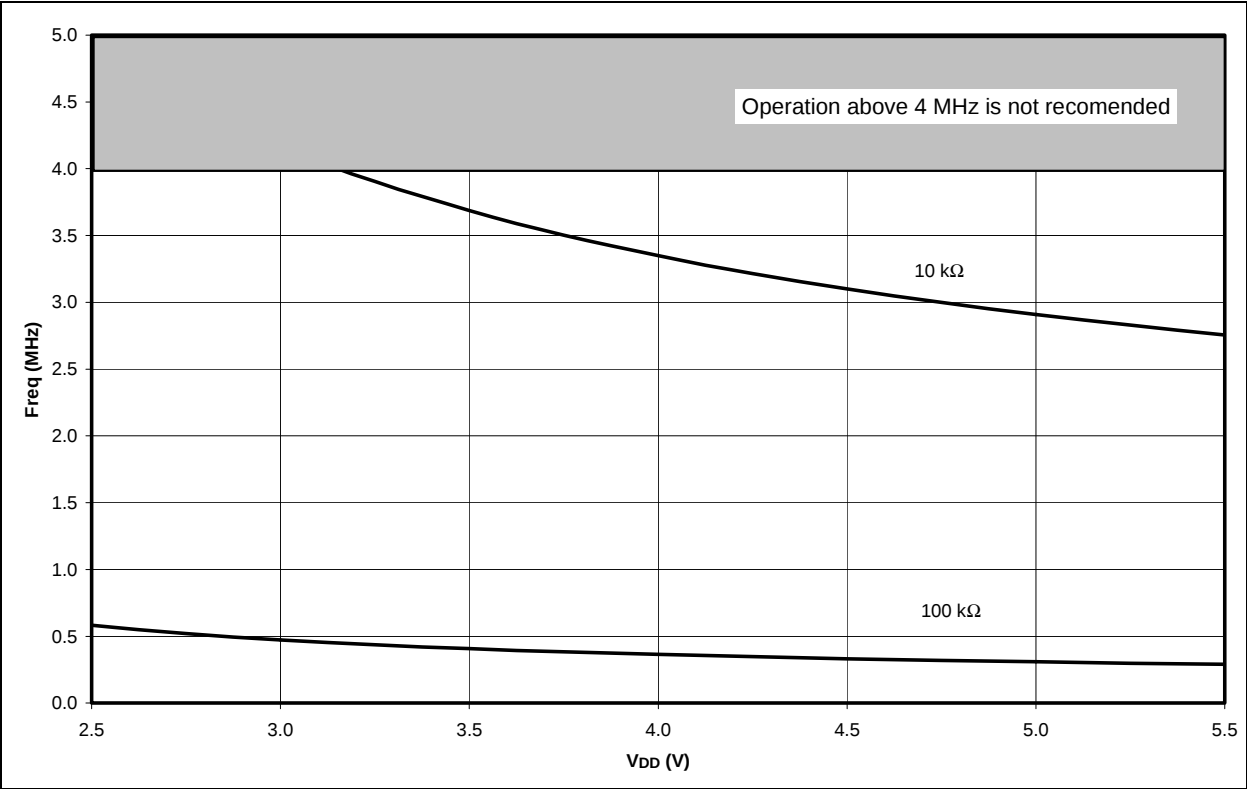
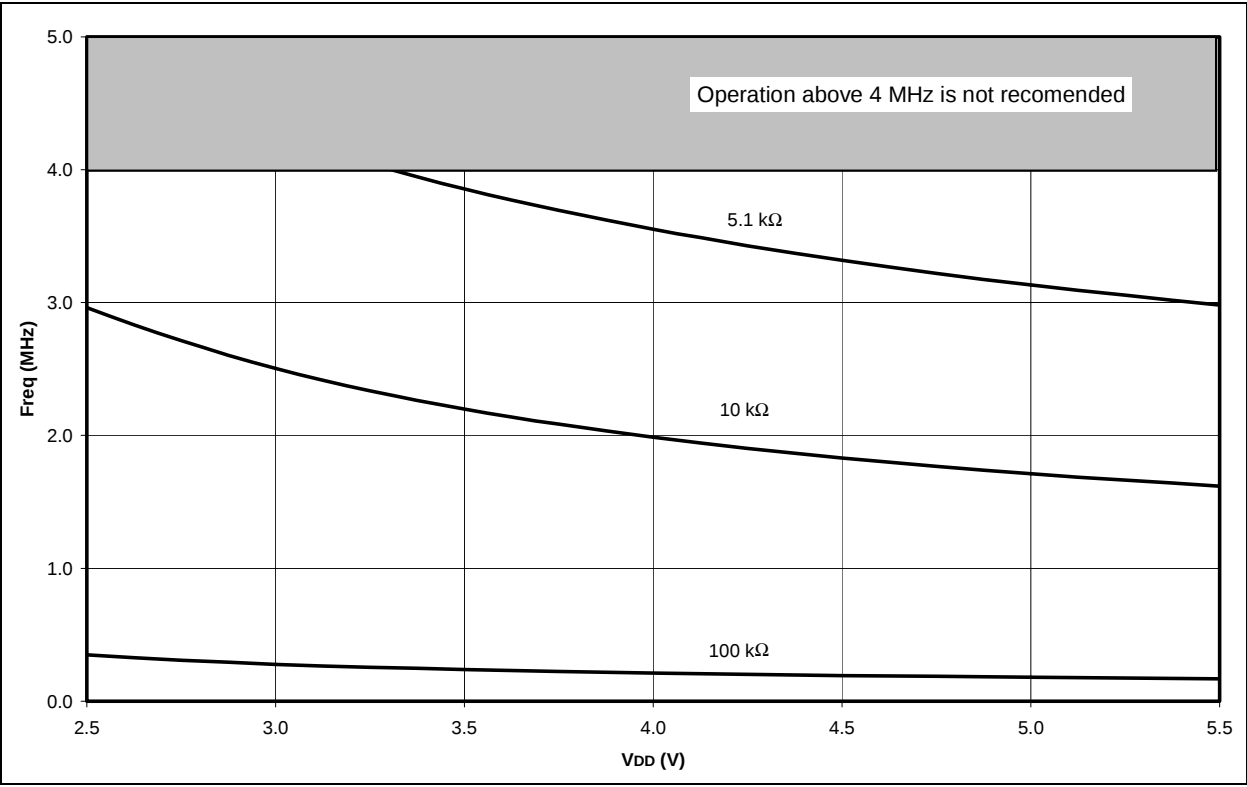


FIGURE 16-8: AVERAGE Fosc vs. VDD FOR VARIOUS VALUES OF R (RC MODE, C = 100 pF, 25°C)



**FIGURE 16-9: AVERAGE F_{osc} vs. V_{DD} FOR VARIOUS VALUES OF R
(RC MODE, $C = 300$ pF, 25°C)**

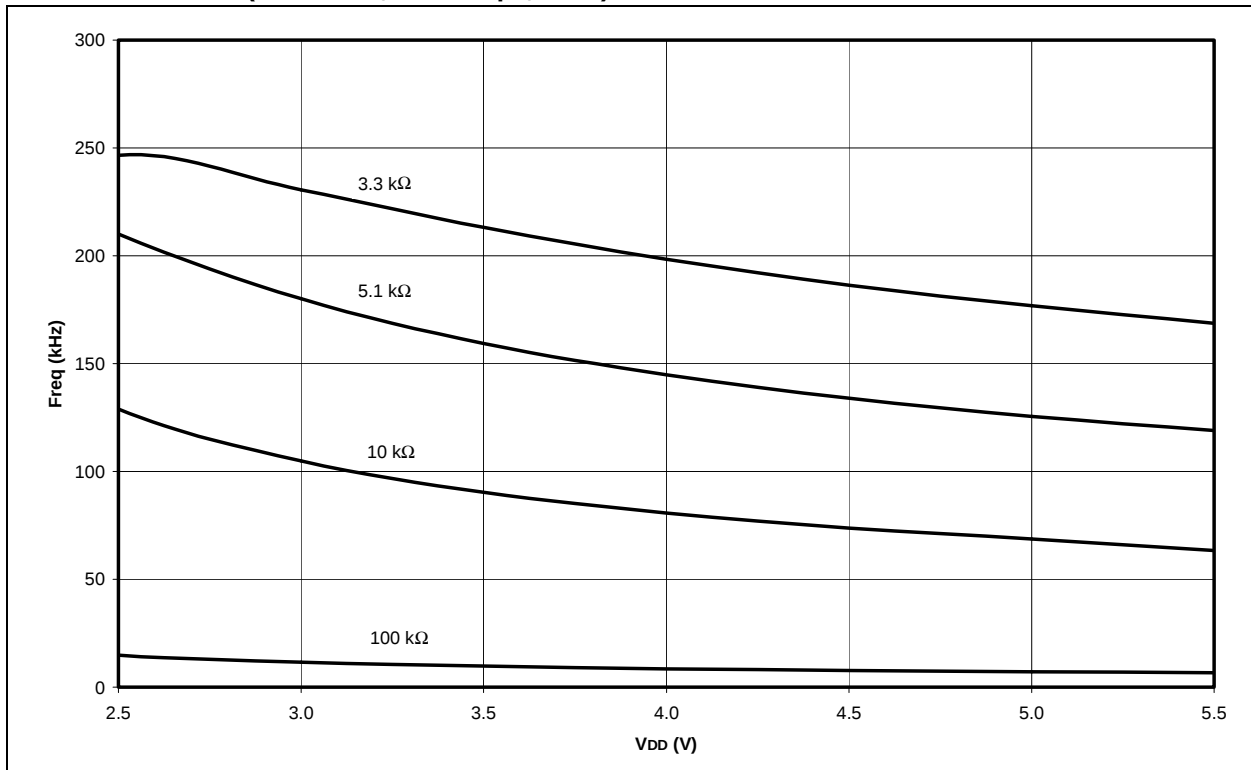
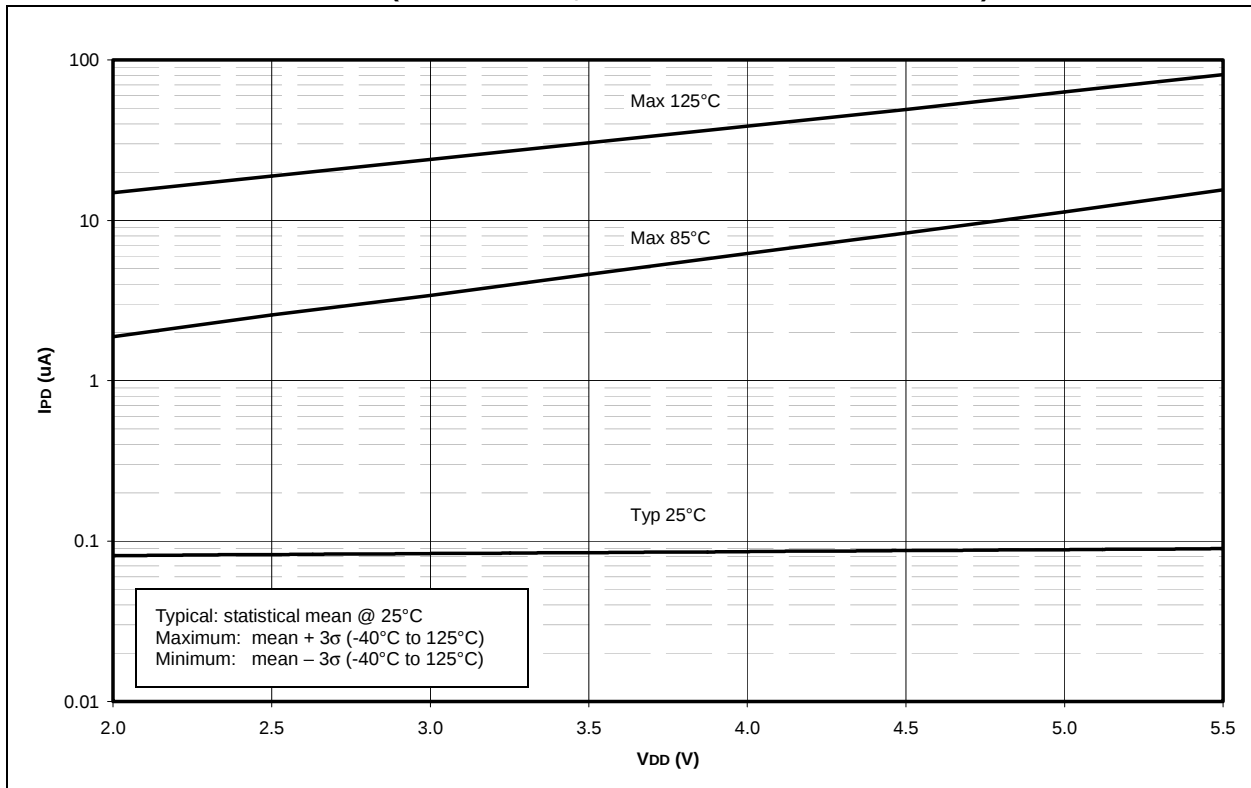


FIGURE 16-10: I_{PD} vs. V_{DD} (SLEEP MODE, ALL PERIPHERALS DISABLED)



PIC16F7X

FIGURE 16-11: ΔI_{BOR} vs. V_{DD} OVER TEMPERATURE

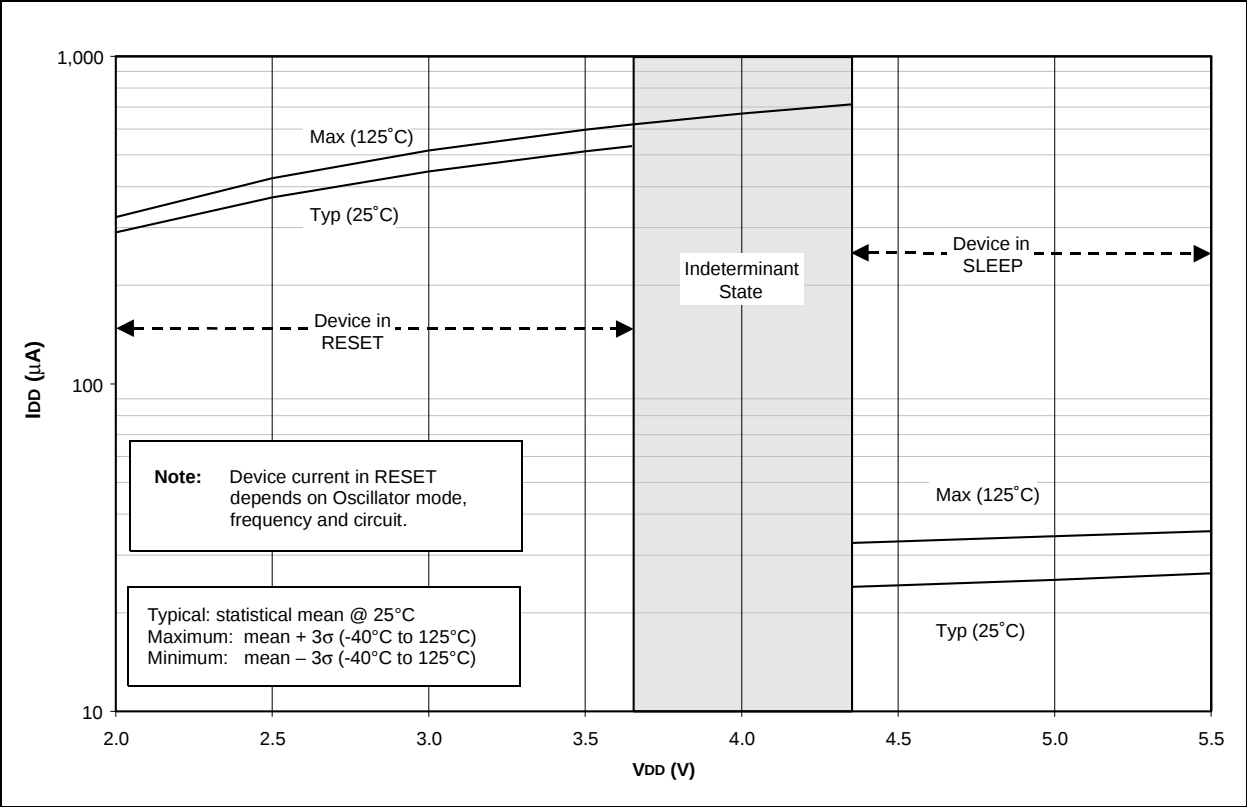


FIGURE 16-12: TYPICAL AND MAXIMUM ΔI_{WDT} vs. V_{DD} OVER TEMPERATURE

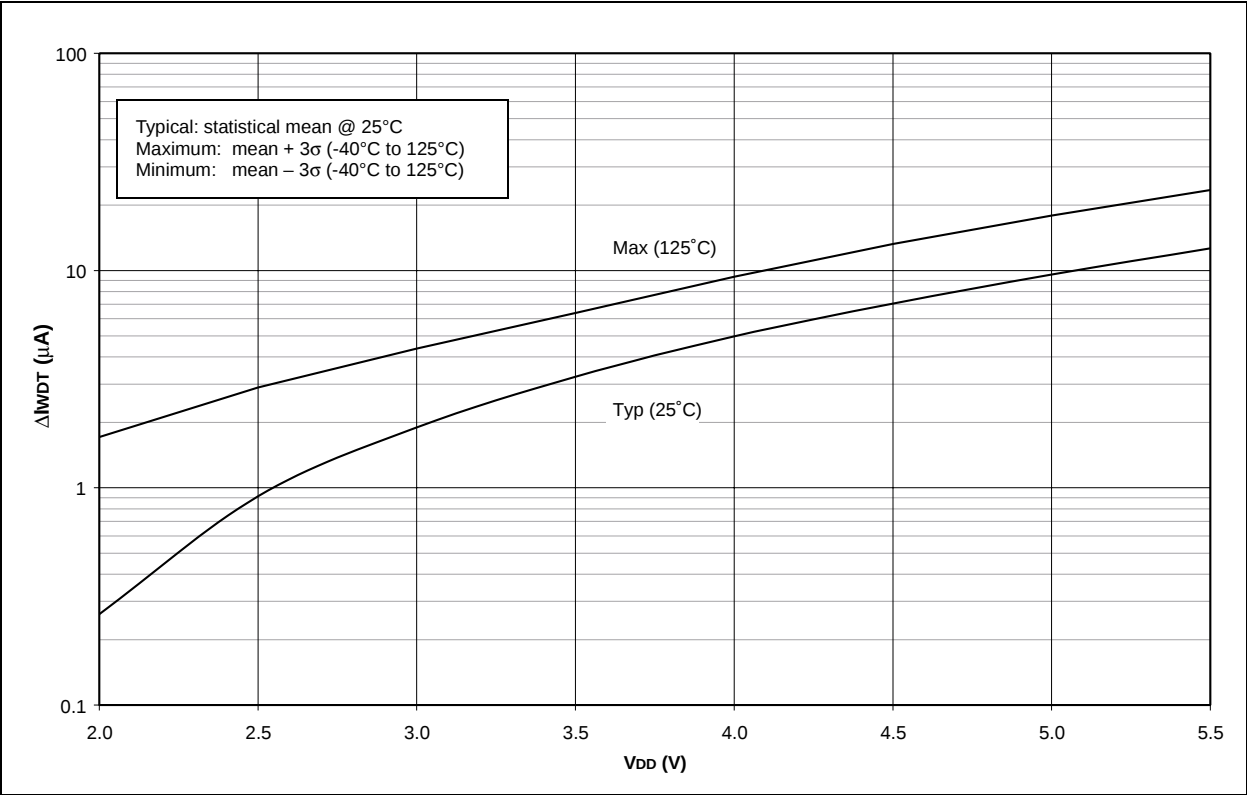


FIGURE 16-13: TYPICAL, MINIMUM AND MAXIMUM WDT PERIOD vs. V_{DD} (-40°C TO 125°C)

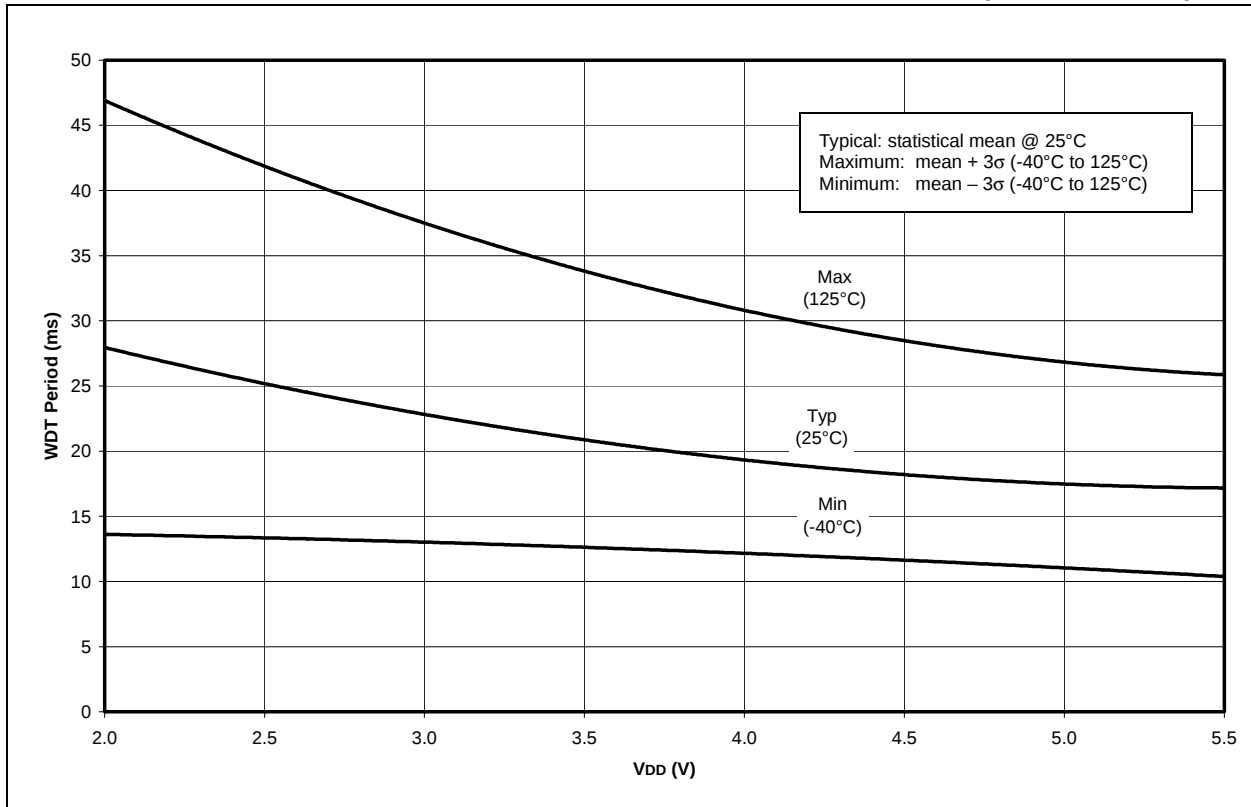
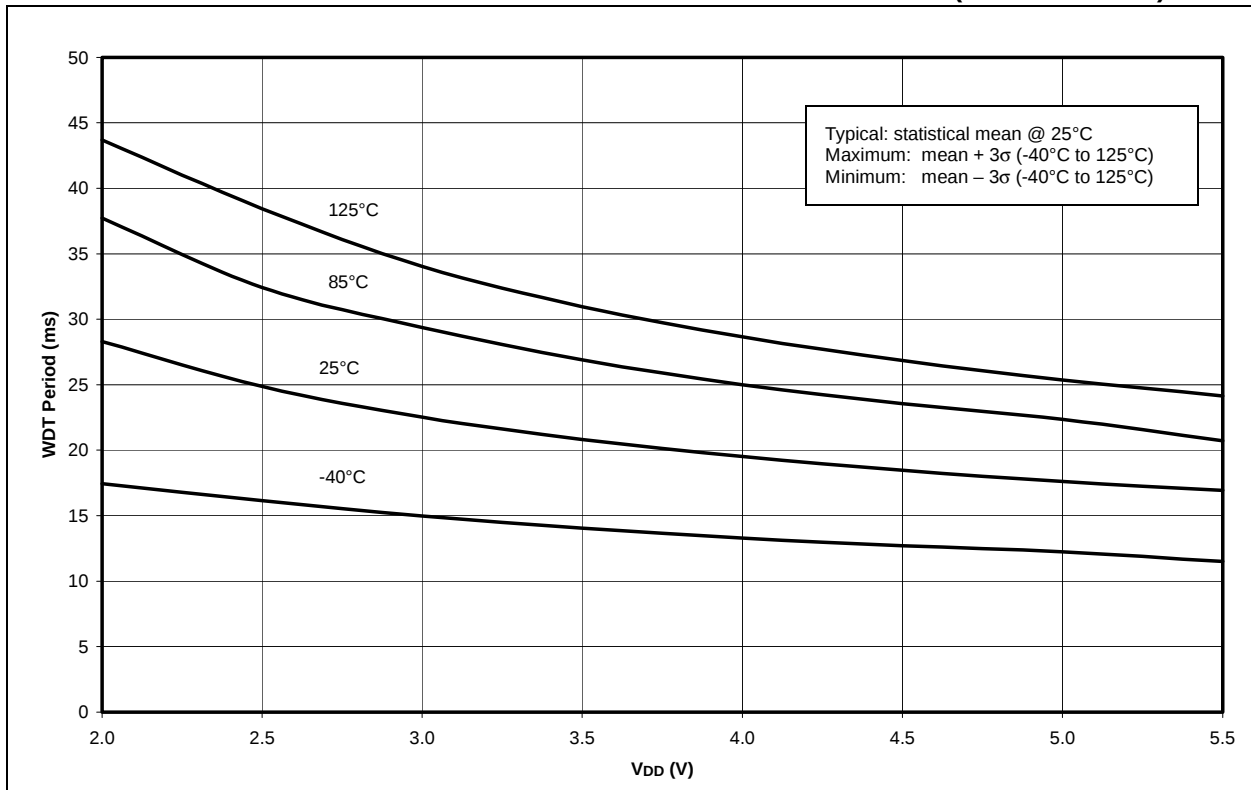


FIGURE 16-14: AVERAGE WDT PERIOD vs. V_{DD} OVER TEMPERATURE (-40°C TO 125°C)



PIC16F7X

FIGURE 16-15: TYPICAL, MINIMUM AND MAXIMUM V_{OH} vs. I_{OH} ($V_{DD} = 5V$, $-40^{\circ}C$ TO $125^{\circ}C$)

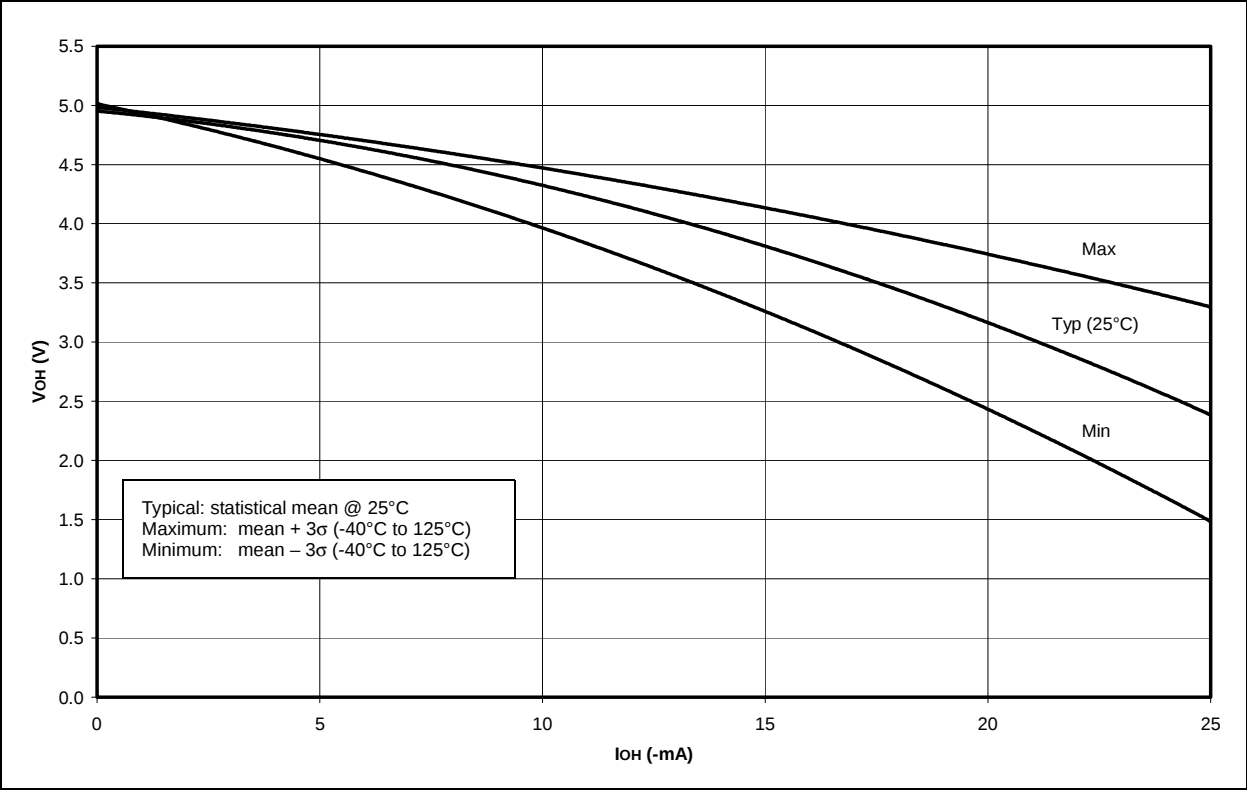


FIGURE 16-16: TYPICAL, MINIMUM AND MAXIMUM V_{OH} vs. I_{OH} ($V_{DD} = 3V$, $-40^{\circ}C$ TO $125^{\circ}C$)

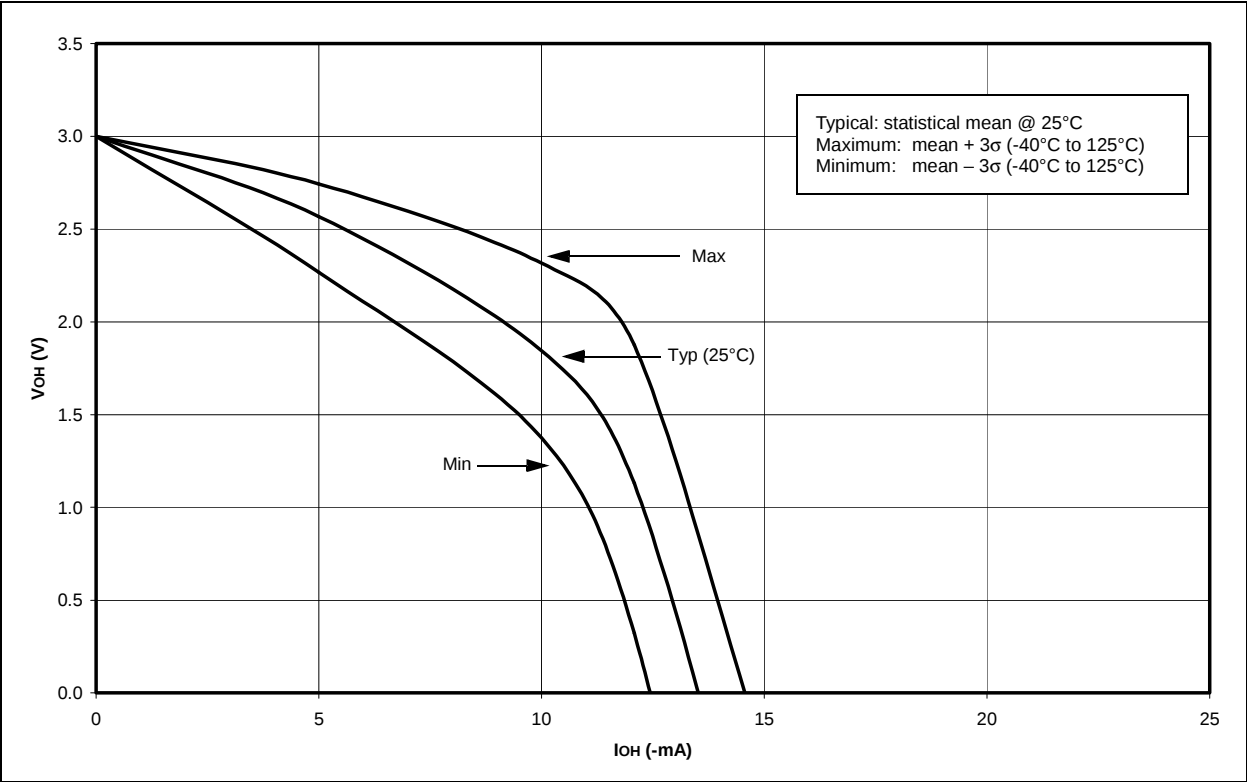


FIGURE 16-17: TYPICAL, MINIMUM AND MAXIMUM V_{OL} vs. I_{OL} ($V_{DD} = 5V$, $-40^{\circ}C$ TO $125^{\circ}C$)

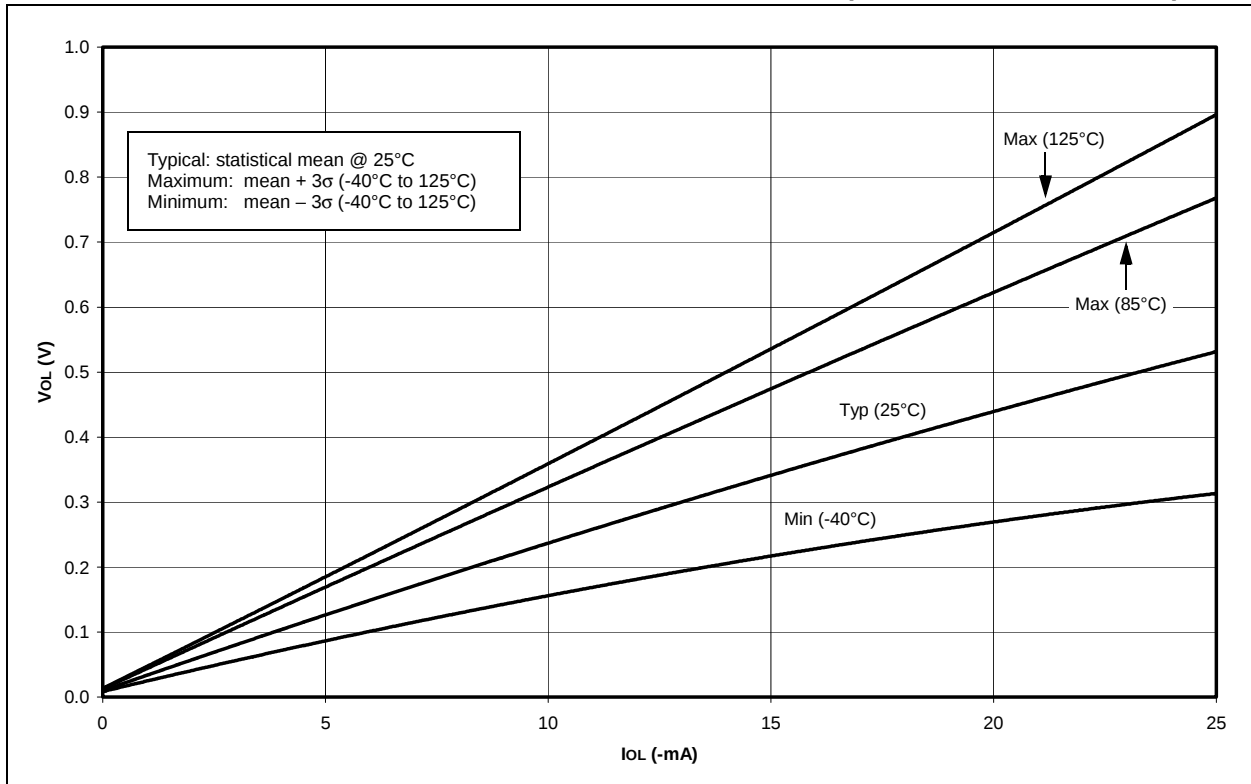
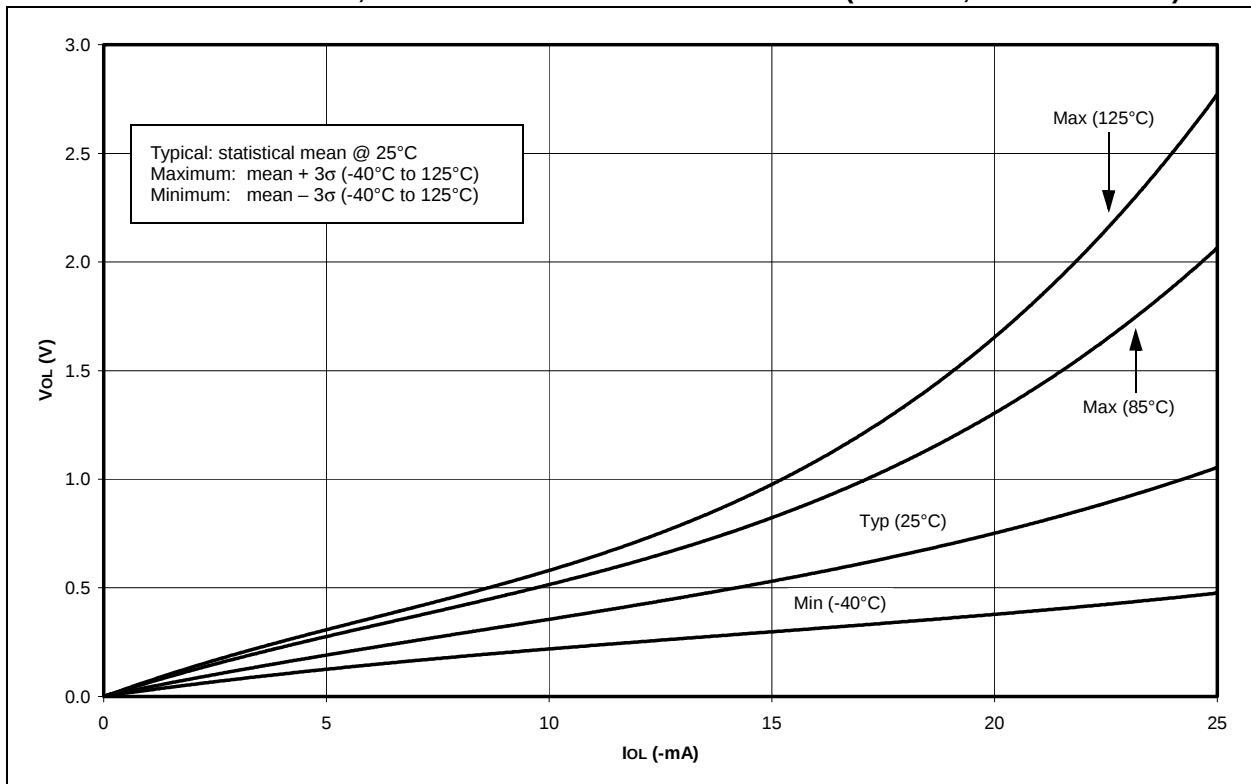


FIGURE 16-18: TYPICAL, MINIMUM AND MAXIMUM V_{OL} vs. I_{OL} ($V_{DD} = 3V$, $-40^{\circ}C$ TO $125^{\circ}C$)



PIC16F7X

FIGURE 16-19: MINIMUM AND MAXIMUM V_{IN} vs. V_{DD} , (TTL INPUT, -40°C TO 125°C)

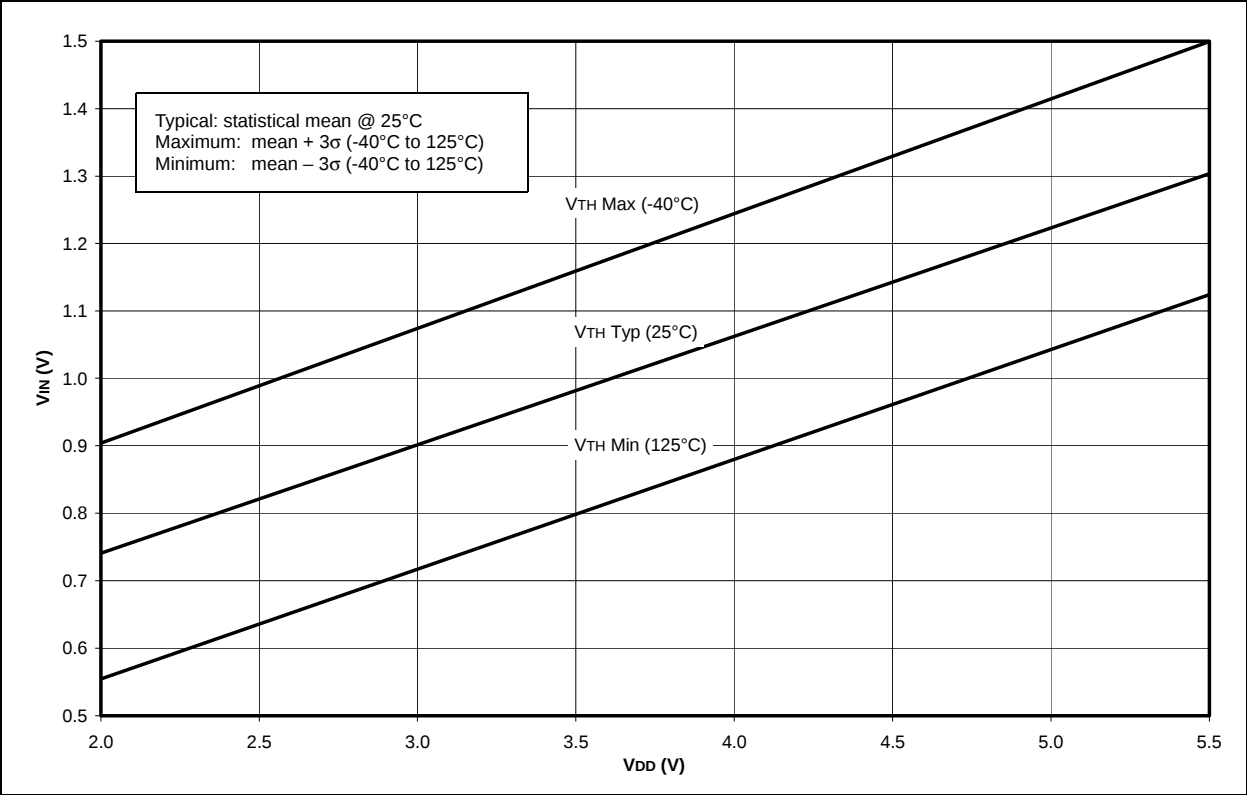
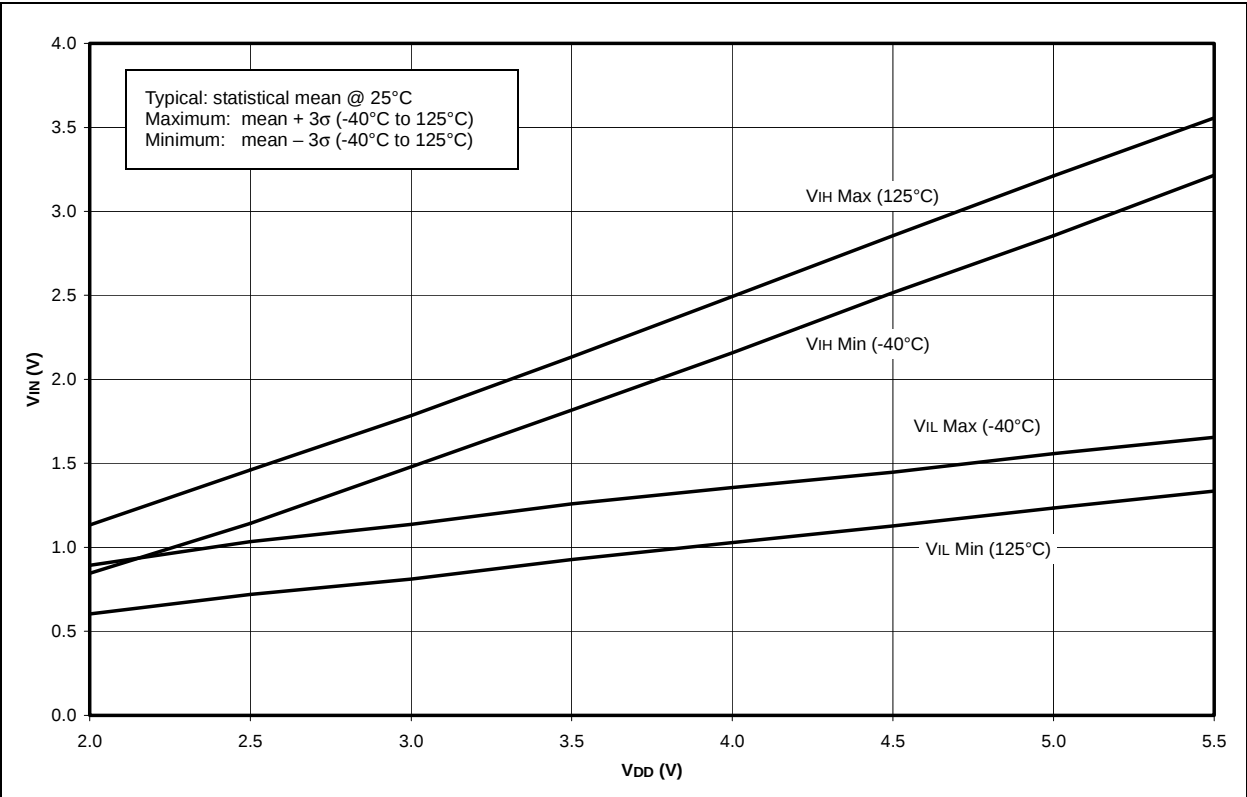


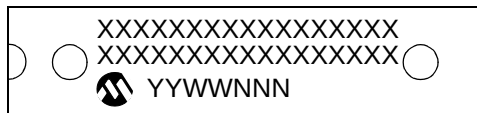
FIGURE 16-20: MINIMUM AND MAXIMUM V_{IN} vs. V_{DD} (ST INPUT, -40°C TO 125°C)



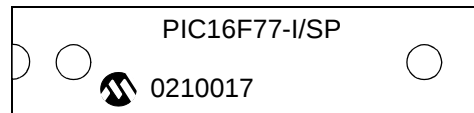
17.0 PACKAGING INFORMATION

17.1 Package Marking Information

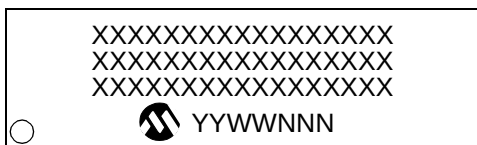
28-Lead PDIP (Skinny DIP)



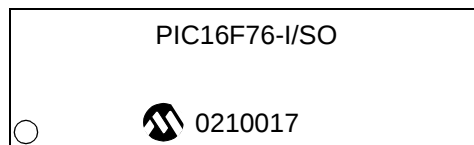
Example



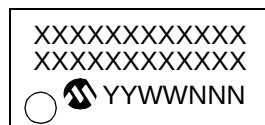
28-Lead SOIC



Example



28-Lead SSOP



Example



28-Lead MLF



Example



Legend:

XX...X	Customer specific information*
Y	Year code (last digit of calendar year)
YY	Year code (last 2 digits of calendar year)
WW	Week code (week of January 1 is week '01')
NNN	Alphanumeric traceability code

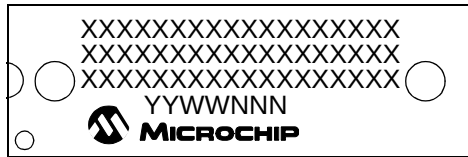
Note: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line thus limiting the number of available characters for customer specific information.

- * Standard PICmicro device marking consists of Microchip part number, year code, week code, and traceability code. For PICmicro device marking beyond this, certain price adders apply. Please check with your Microchip Sales Office. For QTP devices, any special marking adders are included in QTP price.

PIC16F7X

Package Marking Information (Cont'd)

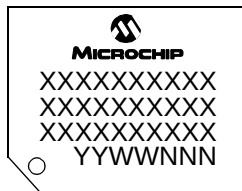
40-Lead PDIP



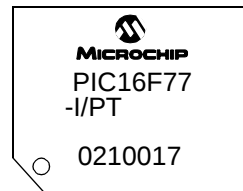
Example



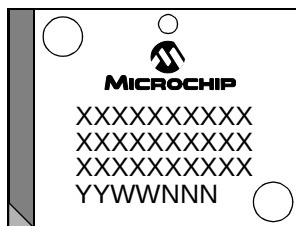
44-Lead TQFP



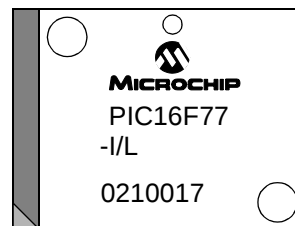
Example



44-Lead PLCC



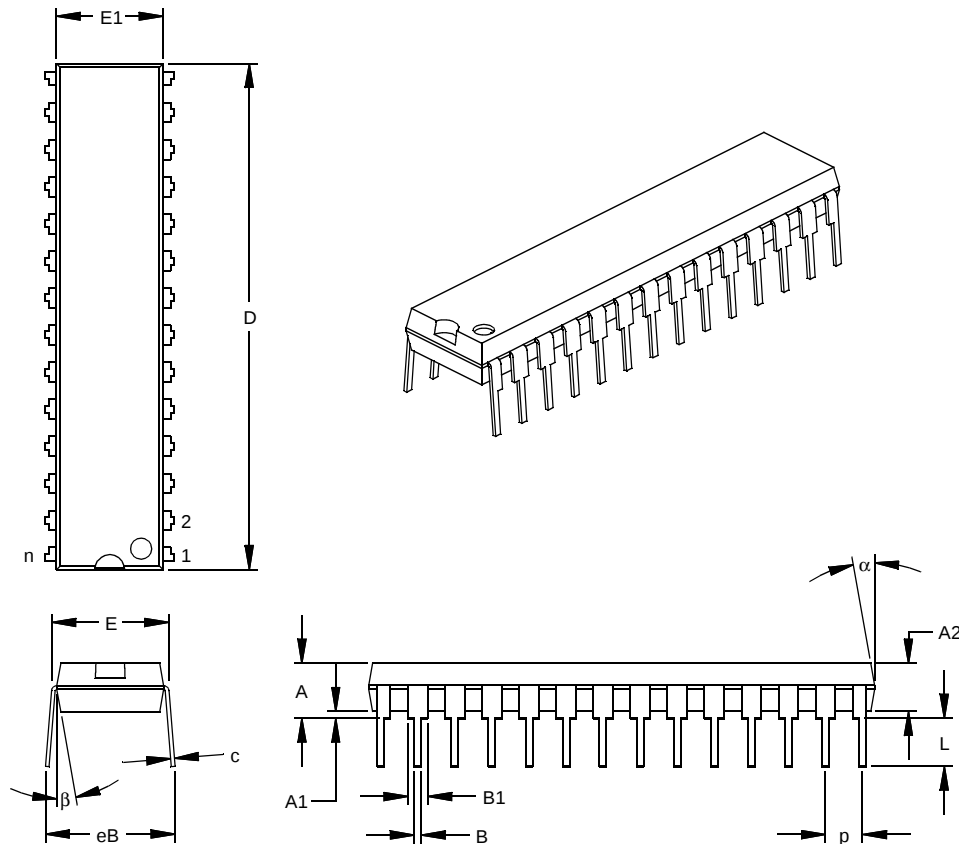
Example



17.2 Package Details

The following sections give the technical details of the packages.

28-Lead Skinny Plastic Dual In-line (SP) – 300 mil (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:

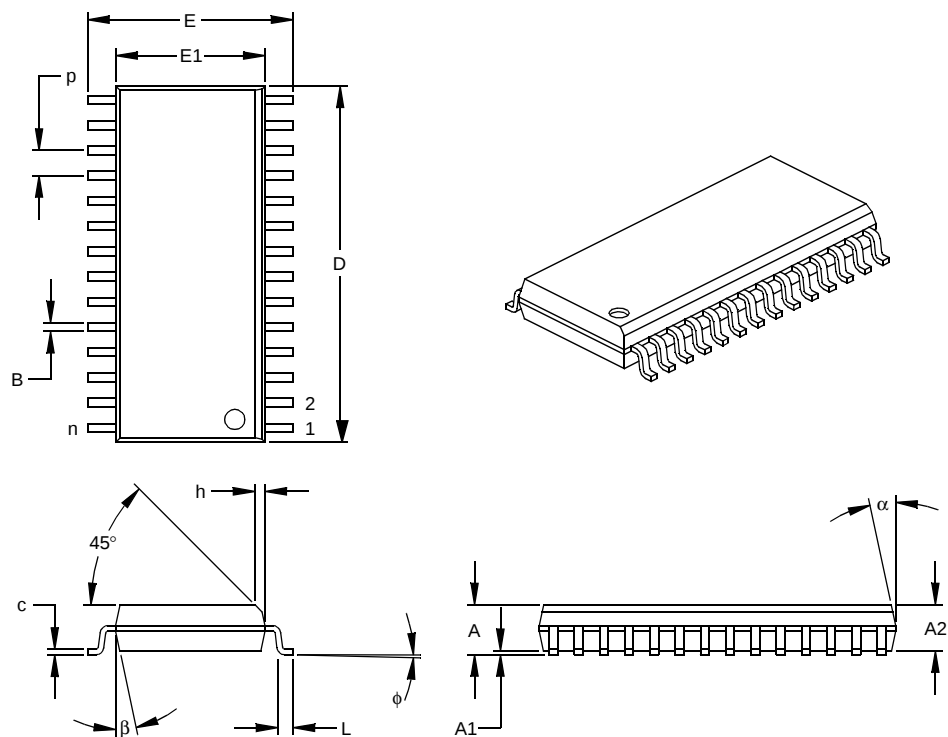
Dimension D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" (0.254mm) per side.

JEDEC Equivalent: MO-095

Drawing No. C04-070

PIC16F7X

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



Units		INCHES*			MILLIMETERS		
Dimension Limits		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

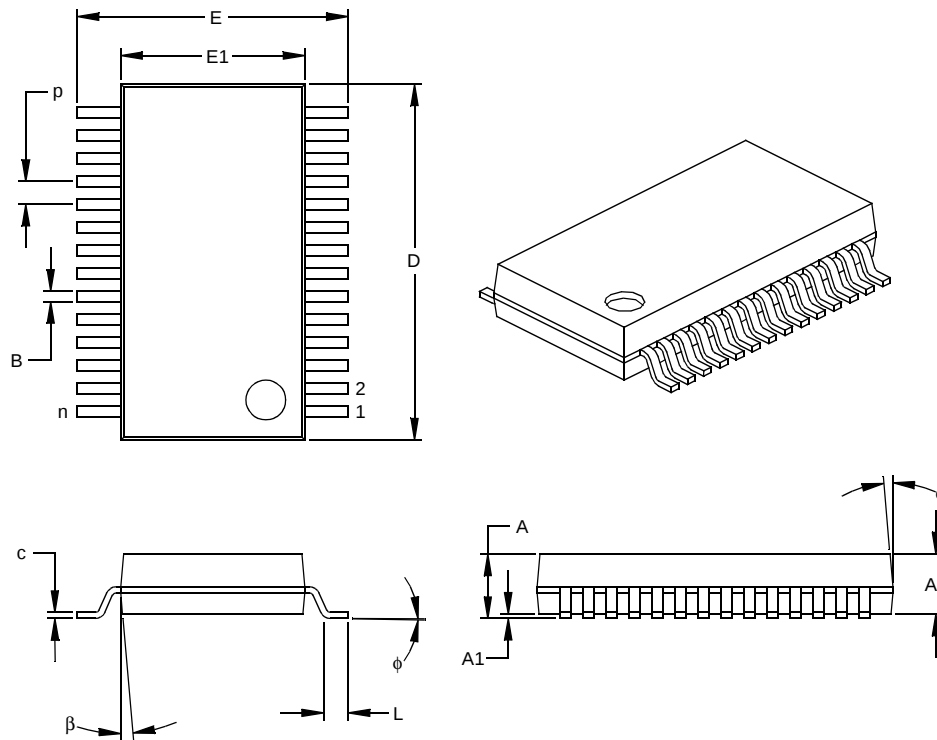
Notes:

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, 5.30 mm (SSOP)



Units		INCHES			MILLIMETERS*		
Dimension Limits		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		28			28	
Pitch	p		.026			0.65	
Overall Height	A	.068	.073	.078	1.73	1.85	1.98
Molded Package Thickness	A2	.064	.068	.072	1.63	1.73	1.83
Standoff §	A1	.002	.006	.010	0.05	0.15	0.25
Overall Width	E	.299	.309	.319	7.59	7.85	8.10
Molded Package Width	E1	.201	.207	.212	5.11	5.25	5.38
Overall Length	D	.396	.402	.407	10.06	10.20	10.34
Foot Length	L	.022	.030	.037	0.56	0.75	0.94
Lead Thickness	c	.004	.007	.010	0.10	0.18	0.25
Foot Angle	φ	0	4	8	0.00	101.60	203.20
Lead Width	B	.010	.013	.015	0.25	0.32	0.38
Mold Draft Angle Top	α	0	5	10	0	5	10
Mold Draft Angle Bottom	β	0	5	10	0	5	10

* Controlling Parameter

§ Significant Characteristic

Notes:

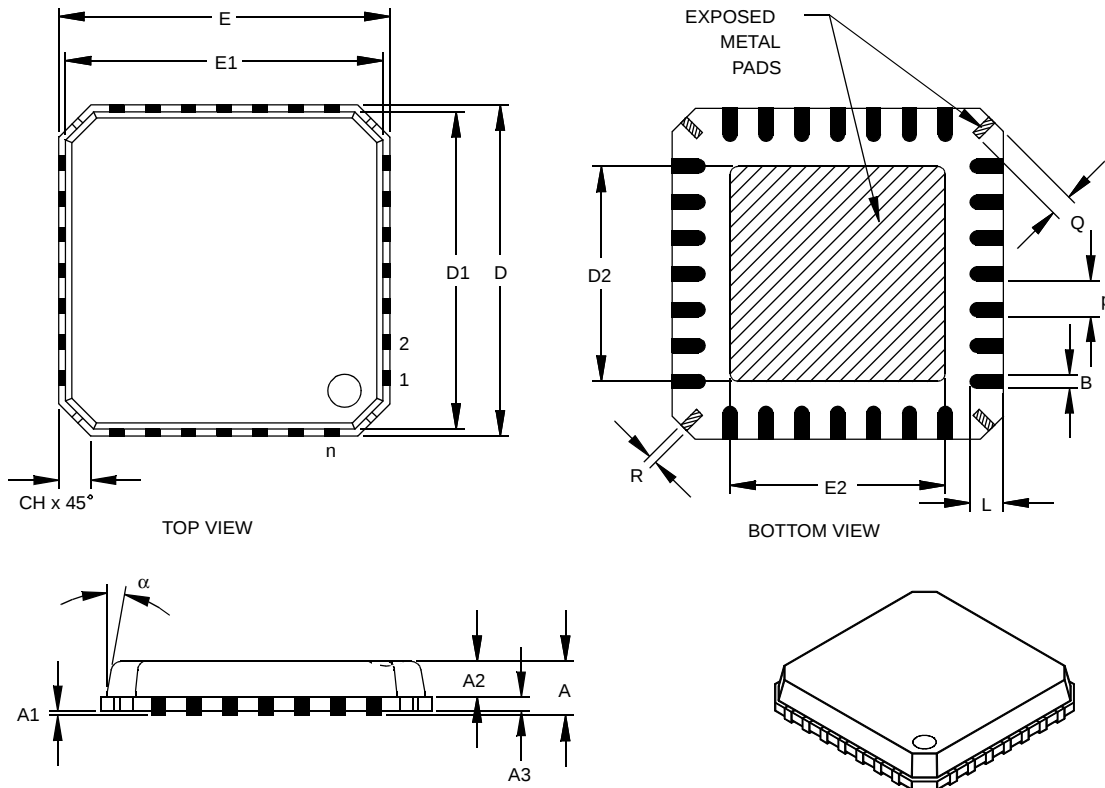
Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" (0.254mm) per side.

JEDEC Equivalent: MS-150

Drawing No. C04-073

PIC16F7X

28-Lead Plastic Micro Leadframe Package (MF) 6x6 mm Body (MLF)



Units		INCHES			MILLIMETERS*		
Dimension Limits		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		28			28	
Pitch	p	.026 BSC			0.65 BSC		
Overall Height	A		.033	.039		0.85	1.00
Molded Package Thickness	A2		.026	.031		0.65	0.80
Standoff	A1	.000	.0004	.002	0.00	0.01	0.05
Base Thickness	A3	.008 REF.			0.20 REF.		
Overall Width	E	.236 BSC			6.00 BSC		
Molded Package Width	E1	.226 BSC			5.75 BSC		
Exposed Pad Width	E2	.140	.146	.152	3.55	3.70	3.85
Overall Length	D	.236 BSC			6.00 BSC		
Molded Package Length	D1	.226 BSC			5.75 BSC		
Exposed Pad Length	D2	.140	.146	.152	3.55	3.70	3.85
Lead Width	B	.009	.011	.014	0.23	0.28	0.35
Lead Length	L	.020	.024	.030	0.50	0.60	0.75
Tie Bar Width	R	.005	.007	.010	0.13	0.17	0.23
Tie Bar Length	Q	.012	.016	.026	0.30	0.40	0.65
Chamfer	CH	.009	.017	.024	0.24	0.42	0.60
Mold Draft Angle Top	α			12°			12°

*Controlling Parameter

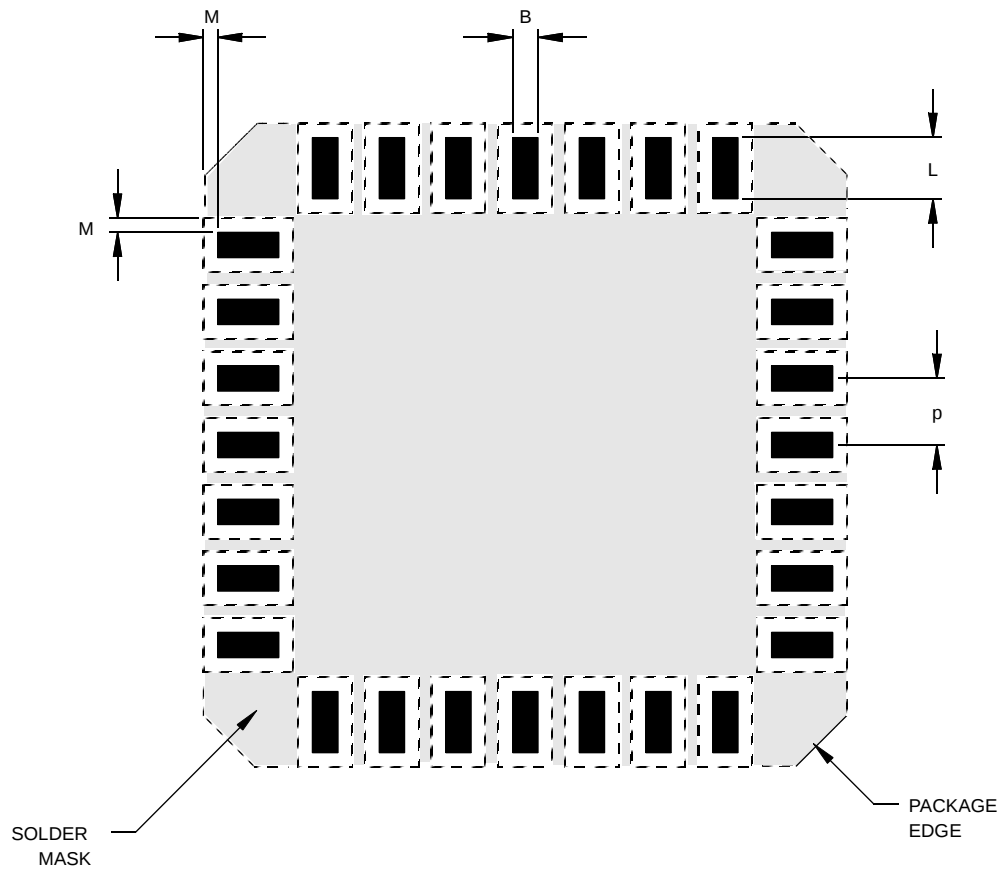
Notes:

Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" (0.254mm) per side.

JEDEC equivalent: pending

Drawing No. C04-114

28-Lead Plastic Micro Leadframe Package (MF) 6x6 mm Body (MLF) (Continued)



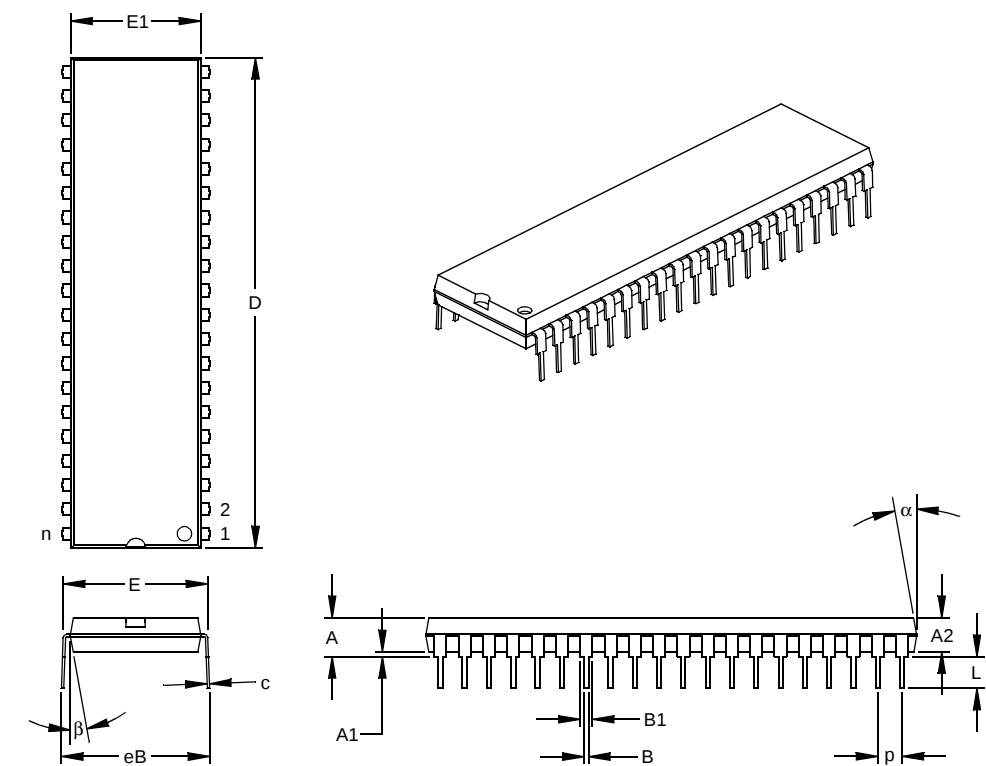
Units		INCHES			MILLIMETERS*		
Dimension Limits		MIN	NOM	MAX	MIN	NOM	MAX
Pitch	p	.026 BSC			0.65 BSC		
Pad Width	B	.009	.011	.014	0.23	0.28	0.35
Pad Length	L	.020	.024	.030	0.50	0.60	0.75
Pad to Solder Mask	M	.005		.006	0.13		0.15

*Controlling Parameter

Drawing No. C04-2114

PIC16F7X

40-Lead Plastic Dual In-line (P) – 600 mil (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

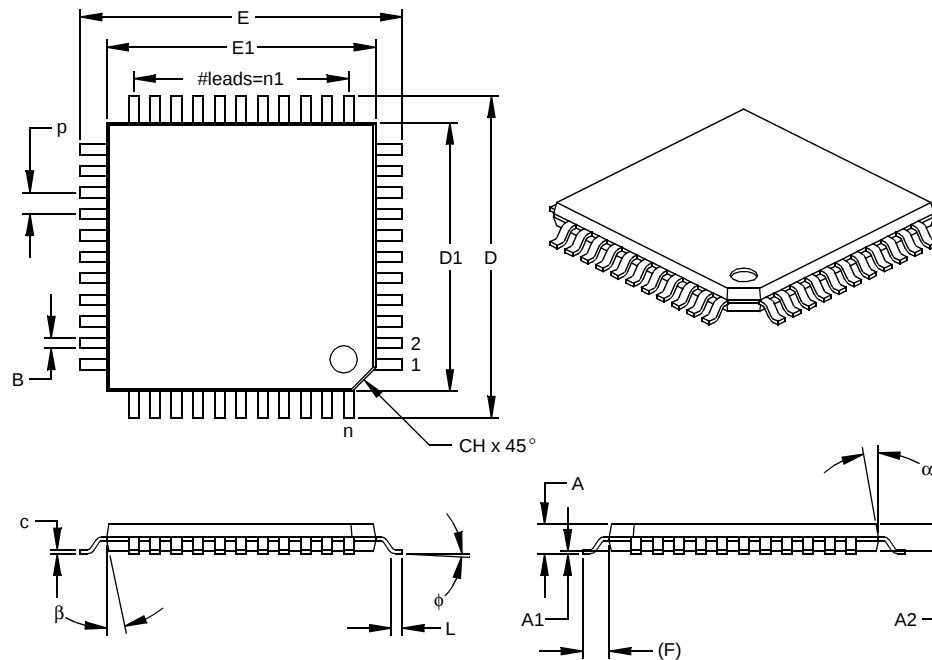
Notes:

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

44-Lead Plastic Thin Quad Flatpack (PT) 10x10x1 mm Body, 1.0/0.10 mm Lead Form (TQFP)



Units		INCHES			MILLIMETERS*		
Dimension Limits		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	phi	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	alpha	5	10	15	5	10	15
Mold Draft Angle Bottom	beta	5	10	15	5	10	15

* Controlling Parameter
§ Significant Characteristic

Notes:

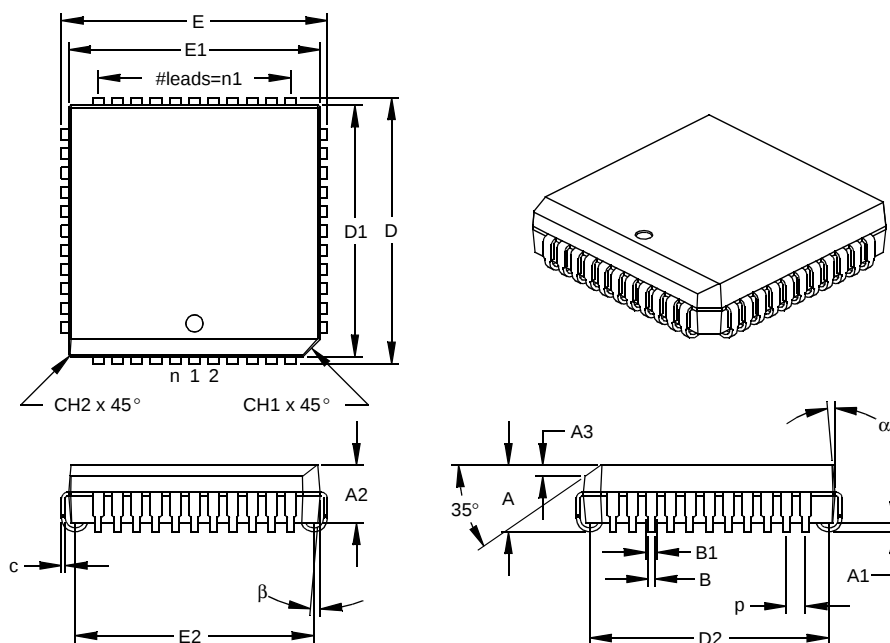
Dimensions D1 and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" (0.254mm) per side.

JEDEC Equivalent: MS-026

Drawing No. C04-076

PIC16F7X

44-Lead Plastic Leaded Chip Carrier (L) – Square (PLCC)



Units		INCHES*			MILLIMETERS		
Dimension Limits		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		44			44	
Pitch	p		.050			1.27	
Pins per Side	n1		11			11	
Overall Height	A	.165	.173	.180	4.19	4.39	4.57
Molded Package Thickness	A2	.145	.153	.160	3.68	3.87	4.06
Standoff §	A1	.020	.028	.035	0.51	0.71	0.89
Side 1 Chamfer Height	A3	.024	.029	.034	0.61	0.74	0.86
Corner Chamfer 1	CH1	.040	.045	.050	1.02	1.14	1.27
Corner Chamfer (others)	CH2	.000	.005	.010	0.00	0.13	0.25
Overall Width	E	.685	.690	.695	17.40	17.53	17.65
Overall Length	D	.685	.690	.695	17.40	17.53	17.65
Molded Package Width	E1	.650	.653	.656	16.51	16.59	16.66
Molded Package Length	D1	.650	.653	.656	16.51	16.59	16.66
Footprint Width	E2	.590	.620	.630	14.99	15.75	16.00
Footprint Length	D2	.590	.620	.630	14.99	15.75	16.00
Lead Thickness	c	.008	.011	.013	0.20	0.27	0.33
Upper Lead Width	B1	.026	.029	.032	0.66	0.74	0.81
Lower Lead Width	B	.013	.020	.021	0.33	0.51	0.53
Mold Draft Angle Top	α	0	5	10	0	5	10
Mold Draft Angle Bottom	β	0	5	10	0	5	10

* 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-047

Drawing No. C04-048

APPENDIX A: REVISION HISTORY

Version	Date	Revision Description
A	2000	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).
B	2001	Final data sheet. Includes device characterization data. Addition of extended temperature devices. Addition of 28-pin MLF package. Minor typographic revisions throughout.

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	PIC16F73	PIC16F74	PIC16F76	PIC16F77
FLASH Program Memory (14-bit words)	4K	4K	8K	8K
Data Memory (bytes)	192	192	368	368
I/O Ports	3	5	3	5
A/D	5 channels, 8 bits	8 channels, 8 bits	5 channels, 8 bits	8 channels, 8 bits
Parallel Slave Port	no	yes	no	yes
Interrupt Sources	11	12	11	12
Packages	28-pin PDIP 28-pin SOIC 28-pin SSOP 28-pin MLF	40-pin PDIP 44-pin TQFP 44-pin PLCC	28-pin PDIP 28-pin SOIC 28-pin SSOP 28-pin MLF	40-pin PDIP 44-pin TQFP 44-pin PLCC

PIC16F7X

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	PIC16F7X
Pins	28/40	28/40	28/40
Timers	3	3	3
Interrupts	11 or 12	13 or 14	11 or 12
Communication	PSP, USART, SSP (SPI, I ² C Slave)	PSP, USART, SSP (SPI, I ² C Master/Slave)	PSP, USART, SSP (SPI, I ² C Slave)
Frequency	20 MHz	20 MHz	20 MHz
A/D	8-bit	10-bit	8-bit
CCP	2	2	2
Program Memory	4K, 8K EPROM	4K, 8K FLASH (1,000 E/W cycles)	4K, 8K FLASH (100 E/W cycles typical)
RAM	192, 368 bytes	192, 368 bytes	192, 368 bytes
EEPROM Data	None	128, 256 bytes	None
Other	—	In-Circuit Debugger, Low Voltage Programming	—

INDEX

A

A/D

A/D Conversion Status (GO/DONE Bit)	83
Acquisition Requirements	86
ADCON0 Register	83
ADCON1 Register	83
ADRES Register	83
Analog Port Pins	8, 10, 12, 39
Analog-to-Digital Converter	83
Associated Registers	88
Configuring Analog Port Pins	87
Configuring the Interrupt	85
Configuring the Module	85
Conversion Clock	87
Conversion Requirements	139
Conversions	87
Converter Characteristics	138
Effects of a RESET	87
Faster Conversion - Lower Resolution	
Trade-off	87
Internal Sampling Switch (Rss) Impedance	86
Operation During SLEEP	87
Source Impedance	86
Using the CCP Trigger	88
Absolute Maximum Ratings	119
ACK Pulse	65, 66
ADCON0 Register	83
GO/DONE Bit	83
ADCON1 Register	83
ADRES Register	83
Analog Port Pins. See A/D	
Application Notes	
AN552 (Implementing Wake-up on Key Strokes	
Using PIC16F7X)	33
AN556 (Implementing a Table Read)	26
AN578 (Use of the SSP Module in the I ² C	
Multi-Master Environment)	59
AN607 (Power-up Trouble Shooting)	94
Assembler	
MPASM Assembler	113

B

Banking, Data Memory	13
BF bit	60
Block Diagrams	
A/D	85
Analog Input Model	86
Capture Mode Operation	55
Compare	55
Crystal/Ceramic Resonator Operation (HS, XT	
or LP Osc Configuration)	91
External Clock Input Operation	
(HS Osc Configuration)	91
Interrupt Logic	99
PIC16F73 and PIC16F76	6
PIC16F74 and PIC16F77	7
PORTA	
RA3:RA0 and RA5 Port Pins	31
RA4/T0CKI Pin	31
PORTB	
RB3:RB0 Port Pins	33
RB7:RB4 Port Pins	33
PORTC (Peripheral Output Override)	35

PORTD (In I/O Port Mode)	36
PORTD and PORTE (Parallel Slave Port)	40
PORTE (In I/O Port Mode)	37
PWM Mode	57
RC Oscillator Mode	92
Recommended MCLR Circuit	94
Reset Circuit	93
SSP (I ² C Mode)	65
SSP (SPI Mode)	62
Timer0/WDT Prescaler	43
Timer1	48
Timer2	51
Typical In-Circuit Serial Programming	
Connection	103
USART	
Receive	75
USART Transmit	73
Watchdog Timer (WDT)	101
BOR. See Brown-out Reset	
BRGH bit	71
Brown-out Reset (BOR)	89, 93, 94, 95, 96

C

Capture/Compare/PWM (CCP)	
Associated Registers	56, 58
Capture Mode	55
Prescaler	55
CCP Pin Configuration	55, 56
CCP1	
RC2/CCP1 Pin	9, 11
CCP2	
RC1/T1OSI/CCP2 Pin	9, 11
Compare Mode	55
Software Interrupt Mode	56
Special Trigger Output	56
Timer1 Mode Selection	56
Example PWM Frequencies and Resolutions	58
Interaction of Two CCP Modules	53
PWM Duty Cycle	57
PWM Mode	57
PWM Period	57
Setup for PWM Operation	58
Special Event Trigger and A/D Conversions	56
Timer Resources	53
CCP1 Module	53
CCP2 Module	53
CCPR1H Register	53
CCPR1L Register	53
CCPxM<3:0> bits	54
CCPxX and CCPxY bits	54
CKE bit	60
CKP bit	61
Code Examples	
Call of a Subroutine in Page 1 from Page 0	26
Changing Between Capture Prescalers	55
Changing Prescaler Assignment to Timer0	45
Changing Prescaler Assignment to WDT	45
FLASH Program Read	30
Indirect Addressing	27
Initializing PORTA	31
Reading a 16-bit Free-Running Timer	49
Saving STATUS, W, and PCLATH Registers	
in RAM	100
Writing a 16-bit Free-Running Timer	49

PIC16F7X

Code Protection	89, 103
Computed GOTO	26
Configuration Bits	89
Continuous Receive Enable (CREN Bit)	70
Conversion Considerations	162

D

D/A bit	60
Data Memory	13
Bank Select (RP1:RP0 bits)	13
General Purpose Registers	13
Register File Map, PIC16F74/73	15
Register File Map, PIC16F77/76	14
Special Function Registers	16
Data/Address bit (D/A)	60
DC and AC Characteristics	
Graphs and Tables	141
DC Characteristics	121
Development Support	113
Device Differences	161
Device Overview	5
Features	5
Direct Addressing	27

E

Electrical Characteristics	119
Errata	4
External Clock Input (RA4/T0CKI). See Timer0	
External Interrupt Input (RB0/INT). See Interrupt Sources	

F

Firmware Instructions	105
FSR Register	27

I

I/O Ports	31
I ² C Mode	
Addressing	66
Associated Registers	68
Master Mode	68
Mode Selection	65
Multi-Master Mode	68
Operation	65
Reception	66
Slave Mode	
SCL and SDA pins	65
Transmission	67
ICEPIC In-Circuit Emulator	114
ID Locations	103
In-Circuit Serial Programming (ICSP)	103
INDF Register	27
Indirect Addressing	27
FSR Register	13
Instruction Format	105
Instruction Set	105
ADDLW	107
ADDWF	107
ANDLW	107
ANDWF	107
BCF	107
BSF	107
BTFSC	107
BTFSS	107
CALL	108
CLRF	108
CLRWF	108

CLRWDW	108
COMF	108
DECF	108
DECFSZ	109
GOTO	109
INCF	109
INCFSZ	109
IORLW	109
IORWF	109
MOVF	110
MOVLW	110
MOVWF	110
NOP	110
RETFIE	110
RETLW	110
RETURN	111
RLF	111
RRF	111
SLEEP	111
SUBLW	111
SUBWF	111
SWAPF	112
XORLW	112
XORWF	112
Summary Table	106

INT Interrupt (RB0/INT). See Interrupt Sources

INTCON Register	21
GIE bit	21
INTE bit	21
INTF bit	21
RBIF bit	21, 33
TMR0IE bit	21

Inter-Integrated Circuit (I²C). See I²C Mode

Interrupt Sources	89, 99
Interrupt-on-Change (RB7:RB4)	33
RB0/INT Pin, External	9, 11, 100
TMR0 Overflow	100
USART Receive/Transmit Complete	69

Interrupts

Synchronous Serial Port Interrupt	23
---	----

Interrupts, Context Saving During	100
---	-----

Interrupts, Enable bits

Global Interrupt Enable (GIE bit)	21, 99
Interrupt-on-Change (RB7:RB4) Enable (RBIE bit)	100
RB0/INT Enable (INTE bit)	21
TMR0 Overflow Enable (TMR0IE bit)	21

Interrupts, Flag bits

Interrupt-on Change (RB7:RB4) Flag	
(RBIF bit)	21
Interrupt-on-Change (RB7:RB4) Flag	
(RBIF bit)	21, 33, 100
RB0/INT Flag (INTF bit)	21
TMR0 Overflow Flag (TMR0IF bit)	100

K

KEELOQ Evaluation and Programming Tools	116
---	-----

L

Load Conditions	125
Loading of PC	26

M

Master Clear (MCLR)	8, 10
MCLR Reset, Normal Operation	93, 95, 96
MCLR Reset, SLEEP	93, 95, 96
Operation and ESD Protection	94
MCLR/VPP Pin	8
MCLR/VPP Pin	10
Memory Organization	13
Data Memory	13
Program Memory	13
Program Memory and Stack Maps	13
MPLAB C17 and MPLAB C18 C Compilers	113
MPLAB ICD In-Circuit Debugger	115
MPLAB ICE High Performance Universal In-Circuit Emulator with MPLAB IDE	114
MPLAB Integrated Development Environment Software	113
MPLINK Object Linker/MPLIB Object Librarian	114

O

OPCODE Field Descriptions	105
OPTION_REG Register	20
INTEDG bit	20
PS2:PS0 bits	20
PSA bit	20
RBPV bit	20
T0CS bit	20
T0SE bit	20
OSC1/CLKI Pin	8, 10
OSC2/CLKO Pin	8, 10
Oscillator Configuration	89
Oscillator Configurations	91
Crystal Oscillator/Ceramic Resonators	91
HS	91, 95
LP	91, 95
RC	91, 92, 95
XT	91, 95
Oscillator, WDT	101

P

P (STOP) bit	60
Packaging	151
Paging, Program Memory	26
Parallel Slave Port	41
Associated Registers	41
Parallel Slave Port (PSP)	36, 40
RE0/RD/AN5 Pin	12, 39
RE1/WR/AN6 Pin	12, 39
RE2/CS/AN7 Pin	12, 39
Select (PSPMODE bit)	36, 37
PCFG0 bit	84
PCFG1 bit	84
PCFG2 bit	84
PCL Register	26
PCLATH Register	26
PCON Register	25, 95
POR Bit	25
PICDEM 1 Low Cost PICmicro Demonstration Board	115
PICDEM 17 Demonstration Board	116
PICDEM 2 Low Cost PIC16CXX Demonstration Board	115
PICDEM 3 Low Cost PIC16CXXX Demonstration Board	116

PICSTART Plus Entry Level

Development Programmer	115
PIE1 Register	22
PIE2 Register	24
Pinout Descriptions	8–9
PIC16F73/PIC16F76	8–9
PIC16F74/PIC16F77	10–12
PIR1 Register	23
PIR2 Register	24
PMADR Register	29
PMADRH Register	29
POP	26
POR. See Power-on Reset	
PORTA	8, 10
Analog Port Pins	8, 10
Associated Registers	32
PORTA Register	31
RA4/T0CKI Pin	8, 10
RA5/SS/AN4 Pin	8, 10
TRISA Register	31
PORTA Register	31
PORTB	9, 11
Associated Registers	34
PORTB Register	33
Pull-up Enable (RBPV bit)	20
RB0/INT Edge Select (INTEDG bit)	20
RB0/INT Pin, External	9, 11, 100
RB7:RB4 Interrupt-on-Change	100
RB7:RB4 Interrupt-on-Change Enable (RBIE bit)	100
RB7:RB4 Interrupt-on-Change Flag (RBIF bit)	21, 33, 100
TRISB Register	33
PORTB Register	33
PORTC	9, 11
Associated Registers	35
PORTC Register	35
RC0/T1OSO/T1CKI Pin	9, 11
RC1/T1OSI/CCP2 Pin	9, 11
RC2/CCP1 Pin	9, 11
RC3/SCK/SCL Pin	9, 11
RC4/SDI/SDA Pin	9, 11
RC5/SDO Pin	9, 11
RC6/TX/CK Pin	9, 11, 70
RC7/RX/DT Pin	9, 11, 70, 71
TRISC Register	35
PORTC Register	35
PORTD	12
Associated Registers	36
Parallel Slave Port (PSP) Function	36
PORTD Register	36
TRISD Register	36
PORTD Register	36
PORTE	12
Analog Port Pins	12, 39
Associated Registers	39
Input Buffer Full Status (IBF bit)	38
Input Buffer Overflow (IBOV bit)	38
PORTE Register	37
PSP Mode Select (PSPMODE bit)	36, 37
RE0/RD/AN5 Pin	12, 39
RE1/WR/AN6 Pin	12, 39
RE2/CS/AN7 Pin	12, 39
TRISE Register	37

PIC16F7X

PORTE Register	37
Postscaler, WDT	
Assignment (PSA bit)	20
Rate Select (PS2:PS0 bits)	20
Power-down Mode. See SLEEP	
Power-on Reset (POR)	89, 93, 95, 96
Oscillator Start-up Timer (OST)	89, 94
POR Status ($\overline{\text{POR}}$ bit)	25
Power Control (PCON) Register	95
Power-down (PD bit)	93
Power-up Timer (PWRT)	89, 94
Time-out (TO bit)	19, 93
PR2 Register	51
Prescaler, Timer0	
Assignment (PSA bit)	20
Rate Select (PS2:PS0 bits)	20
PRO MATE II Universal Device Programmer	115
Program Counter	
RESET Conditions	95
Program Memory	29
Associated Registers	30
Interrupt Vector	13
Memory and Stack Maps	13
Operation During Code Protect	30
Organization	13
Paging	26
PMADR Register	29
PMADRH Register	29
Reading FLASH	30
Reading, PMADR Register	29
Reading, PMADRH Register	29
Reading, PMCON1 Register	29
Reading, PMDATA Register	29
Reading, PMDATA Register	29
RESET Vector	13
Program Verification	103
Programming Pin (VPP)	8, 10
Programming, Device Instructions	105
PUSH	26

R

R/W bit	60, 66, 67
RA0/AN0 Pin	8, 10
RA1/AN1 Pin	8, 10
RA2/AN2 Pin	8, 10
RA3/AN3/VREF Pin	8, 10
RA4/T0CKI Pin	8, 10
RA5/SS/AN4 Pin	8, 10
RAM. See Data Memory	
RB0/INT Pin	9, 11
RB1 Pin	9, 11
RB2 Pin	9, 11
RB3/PGM Pin	9, 11
RB4 Pin	9, 11
RB5 Pin	9, 11
RB6/PGC Pin	9, 11
RB7/PGD Pin	9, 11
RC0/T1OSO/T1CKI Pin	9, 11
RC1/T1OSI/CCP2 Pin	9, 11
RC2/CCP1 Pin	9, 11
RC3/SCK/SCL Pin	9, 11
RC4/SDI/SDA Pin	9, 11
RC5/SDO Pin	9, 11
RC6/TX/CK Pin	9, 11
RC7/RX/DT Pin	9, 11

RCSTA Register	
CREN bit	70
OERR bit	70
SPEN bit	69
SREN bit	70
RD0/PSP0 Pin	12
RD1/PSP1 Pin	12
RD2/PSP2 Pin	12
RD3/PSP3 Pin	12
RD4/PSP4 Pin	12
RD5/PSP5 Pin	12
RD6/PSP6 Pin	12
RD7/PSP7 Pin	12
RE0/RD/AN5 Pin	12
RE1/WR/AN6 Pin	12
RE2/ $\overline{\text{CS}}$ /AN7 Pin	12
Read-Modify-Write Operations	105
Receive Overflow Indicator bit (SSPOV)	61
Register File	13
Registers	
ADCON0 (A/D Control 0)	83
ADCON0 (A/D Control 0) Register	83
ADCON1 (A/D Control 1)	83
ADCON1 (A/D Control 1) Register	84
ADRES (A/D Result)	83
CCP1CON/CCP2CON (CCP Control) Registers	54
Configuration Word Register	90
Initialization Conditions (table)	96–97
INTCON (Interrupt Control)	21
INTCON (Interrupt Control) Register	21
OPTION_REG	20
OPTION_REG Register	20, 44
PCON (Power Control)	25
PCON (Power Control) Register	25
PIE1 (Peripheral Interrupt Enable 1)	22
PIE1 (Peripheral Interrupt Enable 1) Register	22
PIE2 (Peripheral Interrupt Enable 2)	24
PIE2 (Peripheral Interrupt Enable 2) Register	24
PIR1 (Peripheral Interrupt Request 1)	23
PIR1 (Peripheral Interrupt Request 1) Register	23
PIR2 (Peripheral Interrupt Request 2)	24
PIR2 (Peripheral Interrupt Request 2) Register	24
PMCON1 (Program Memory Control 1) Register	29
RCSTA (Receive Status and Control) Register	70
Special Function, Summary	16–18
SSPCON (Sync Serial Port Control) Register	61
SSPSTAT (Sync Serial Port Status) Register	60
STATUS Register	19
T1CON (Timer 1 Control) Register	47
T2CON (Timer2 Control) Register	52
TRISE Register	38
TXSTA (Transmit Status and Control) Register	69
RESET	89, 93
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	96
RESET Conditions for PCON Register	95
RESET Conditions for Program Counter	95
RESET Conditions for STATUS Register	95
RESET	
WDT Reset. See Watchdog Timer (WDT)	
Revision History	161

S

S (START) bit	60
SCI. See USART	
SCL	65
Serial Communication Interface. See USART	
SLEEP	89, 93, 102
SMP bit	60
Software Simulator (MPLAB SIM)	114
Special Features of the CPU	89
Special Function Registers	16, 16–18
Speed, Operating	1
SPI Mode	59
Associated Registers	64
Serial Clock (SCK pin)	59
Serial Data In (SDI pin)	59
Serial Data Out (SDO pin)	59
Slave Select	59
SSP	
Overview	
RA5/SS/AN4 Pin	8, 10
RC3/SCK/SCL Pin	9, 11
RC4/SDI/SDA Pin	9, 11
RC5/SDO Pin	9, 11
SSP I ² C Operation	65
Slave Mode	65
SSPEN bit	61
SSPIF bit	23
SSPM<3:0> bits	61
SSPOV bit	61
Stack	26
Overflows	26
Underflow	26
STATUS Register	
DC Bit	19
IRP Bit	19
PD Bit	93
TO Bit	19, 93
Z Bit	19
Synchronous Serial Port Enable bit (SSPEN)	61
Synchronous Serial Port Interrupt bit (SSPIF)	23
Synchronous Serial Port Mode Select bits (SSPM<3:0>)	61
Synchronous Serial Port. See SSP	

T

T1CKPS0 bit	47
T1CKPS1 bit	47
T1OSCEN bit	47
T1SYNC bit	47
T2CKPS0 bit	52
T2CKPS1 bit	52
TAD	87
Time-out Sequence	94
Timer0	43
Associated Registers	45
Clock Source Edge Select (T0SE bit)	20
Clock Source Select (T0CS bit)	20
External Clock	44
Interrupt	43
Overflow Enable (TMR0IE bit)	21
Overflow Flag (TMR0IF bit)	100
Overflow Interrupt	100
Prescaler	45
RA4/T0CKI Pin, External Clock	8, 10
T0CKI	44

Timer1	47
Associated Registers	50
Asynchronous Counter Mode	49
Capacitor Selection	50
Counter Operation	48
Operation in Timer Mode	48
Oscillator	50
Prescaler	50
RC0/T1OSO/T1CKI Pin	9, 11
RC1/T1OSI/CCP2 Pin	9, 11
Resetting of Timer1 Registers	50
Resetting Timer1 using a CCP Trigger Output	50
Synchronized Counter Mode	48
TMR1H Register	49
TMR1L Register	49
Timer2	51
Associated Registers	52
Output	51
Postscaler	51
Prescaler	51
Prescaler and Postscaler	51
Timing Diagrams	
A/D Conversion	139
Brown-out Reset	128
Capture/Compare/PWM (CCP1 and CCP2)	130
CLKOUT and I/O	127
External Clock	126
I ² C Bus Data	135
I ² C Bus START/STOP bits	134
I ² C Reception (7-bit Address)	67
I ² C Transmission (7-bit Address)	67
Parallel Slave Port	131
Parallel Slave Port Read Waveforms	41
Parallel Slave Port Write Waveforms	41
Power-up Timer	128
PWM Output	57
RESET	128
Slow Rise Time (MCLR Tied to VDD Through RC Network)	98
SPI Master Mode (CKE = 0, SMP = 0)	132
SPI Master Mode (CKE = 1, SMP = 1)	132
SPI Mode (Master Mode)	63
SPI Mode (Slave Mode with CKE = 0)	63
SPI Mode (Slave Mode with CKE = 1)	63
SPI Slave Mode (CKE = 0)	133
SPI Slave Mode (CKE = 1)	133
Start-up Timer	128
Time-out Sequence on Power-up (MCLR Not Tied to VDD)	98
Case 1	98
Case 2	98
Time-out Sequence on Power-up (MCLR Tied to VDD Through RC Network)	97
Timer0	129
Timer1	129
USART Asynchronous Master Transmission	74
USART Asynchronous Master Transmission (Back to Back)	74
USART Asynchronous Reception	76
USART Synchronous Receive (Master/Slave)	137
USART Synchronous Reception (Master Mode, SREN)	79
USART Synchronous Transmission	78
USART Synchronous Transmission (Master/Slave)	137

PIC16F7X

USART Synchronous Transmission	
(Through TXEN)	78
Wake-up from SLEEP via Interrupt	103
Watchdog Timer	128
Timing Parameter Symbology	125
Timing Requirements	
Capture/Compare/PWM (CCP1 and CCP2)	130
CLKOUT and I/O	127
External Clock	126
I ² C Bus Data	136
I ² C Bus START/STOP Bits	135
Parallel Slave Port	131
RESET, Watchdog Timer, Oscillator	
Start-up Timer, Power-up Timer	
and Brown-out Reset	128
SPI Mode	134
Timer0 and Timer1 External Clock	129
USART Synchronous Receive	137
USART Synchronous Transmission	137
TMR1CS bit	47
TMR1ON bit	47
TMR2ON bit	52
TOUTPS<3:0> bits	52
TRISA Register	31
TRISB Register	33
TRISC Register	35
TRISD Register	36
TRISE Register	37
IBF Bit	38
IBOV Bit	38
PSPMODE bit	36, 37
TXSTA Register	
SYNC bit	69
TRMT bit	69
TX9 bit	69
TX9D bit	69
TXEN bit	69

U

UA	60
Universal Synchronous Asynchronous	
Receiver Transmitter. See USART	
Update Address bit, UA	60
USART	69
Asynchronous Mode	73
Asynchronous Receiver	75
Asynchronous Reception	76
Associated Registers	76
Asynchronous Transmission	
Associated Registers	74
Asynchronous Transmitter	73

Baud Rate Generator (BRG)	71
Baud Rate Formula	71
Baud Rates, Asynchronous Mode	
(BRGH = 0)	72
Baud Rates, Asynchronous Mode	
(BRGH = 1)	72
Sampling	71
Mode Select (SYNC Bit)	69
Overrun Error (OERR Bit)	70
RC6/TX/CK Pin	9, 11
RC7/RX/DT Pin	9, 11
Serial Port Enable (SPEN Bit)	69
Single Receive Enable (SREN Bit)	70
Synchronous Master Mode	77
Synchronous Master Reception	79
Associated Registers	80
Synchronous Master Transmission	77
Associated Registers	78
Synchronous Slave Mode	80
Synchronous Slave Reception	81
Associated Registers	81
Synchronous Slave Transmission	80
Associated Registers	81
Transmit Data, 9th Bit (TX9D)	69
Transmit Enable (TXEN bit)	69
Transmit Enable, Nine-bit (TX9 bit)	69
Transmit Shift Register Status (TRMT bit)	69

W

Wake-up from SLEEP	89, 102
Interrupts	95, 96
MCLR Reset	96
WDT Reset	96
Wake-up Using Interrupts	102
Watchdog Timer (WDT)	89, 101
Associated Registers	101
Enable (WDTE Bit)	101
Postscaler. See Postscaler, WDT	
Programming Considerations	101
RC Oscillator	101
Time-out Period	101
WDT Reset, Normal Operation	93, 95, 96
WDT Reset, SLEEP	93, 95, 96
WCOL bit	61
Write Collision Detect bit (WCOL)	61
WWW, On-Line Support	4

ON-LINE SUPPORT

Microchip provides on-line support on the Microchip World Wide Web (WWW) 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 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 by using your favorite Internet browser to attach to:

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 any currently available 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.

013001

PIC16F7X

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 Data Sheet.

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: PIC16F7X Literature Number: DS30325B

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 data sheet easy to follow? If not, why?

4. What additions to the data sheet do you think would enhance the structure and subject?

5. What deletions from the data sheet 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?

8. How would you improve our software, systems, and silicon products?

PIC16F7X PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

PART NO.	X	/XX	XXX
Device	Temperature Range	Package	Pattern
Device	PIC16F7X ⁽¹⁾ , PIC16F7XT ⁽¹⁾ ; V _{DD} range 4.0V to 5.5V PIC16LF7X ⁽¹⁾ , PIC16LF7XT ⁽¹⁾ ; V _{DD} range 2.0V to 5.5V		
Temperature Range	I = -40°C to +85°C (Industrial) E = -40°C to +125°C (Extended)		
Package	ML = MLF (Micro Lead Frame) PT = TQFP (Thin Quad Flatpack) SO = SOIC SP = Skinny Plastic DIP P = PDIP L = PLCC SS = SSOP		
Pattern	QTP, SQTP, Code or Special Requirements (blank otherwise)		

Examples:

- a) PIC16F77-I/P 301 = Industrial temp., PDIP package, normal V_{DD} limits, QTP pattern #301.
- b) PIC16LF76-I/SO = Industrial temp., SOIC package, Extended V_{DD} limits.
- c) PIC16F74-E/P = Extended temp., PDIP package, normal V_{DD} limits.

Note 1: F = CMOS FLASH
LF = Low Power CMOS FLASH

2: T = in tape and reel - SOIC, PLCC, SSOP, TQFP packages only.

Sales and Support

Data Sheets

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

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

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

New Customer Notification System

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



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: 480-792-7627
Web Address: <http://www.microchip.com>

Rocky Mountain

2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7966 Fax: 480-792-7456

Atlanta

500 Sugar Mill Road, Suite 200B
Atlanta, GA 30350
Tel: 770-640-0034 Fax: 770-640-0307

Boston

2 Lan Drive, Suite 120
Westford, MA 01886
Tel: 978-692-3848 Fax: 978-692-3821

Chicago

333 Pierce Road, Suite 180
Itasca, IL 60143
Tel: 630-285-0071 Fax: 630-285-0075

Dallas

4570 Westgrove Drive, Suite 160
Addison, TX 75001
Tel: 972-818-7423 Fax: 972-818-2924

Detroit

Tri-Atria Office Building
32255 Northwestern Highway, Suite 190
Farmington Hills, MI 48334
Tel: 248-538-2250 Fax: 248-538-2260

Kokomo

2767 S. Albright Road
Kokomo, Indiana 46902
Tel: 765-864-8360 Fax: 765-864-8387

Los Angeles

18201 Von Karman, Suite 1090
Irvine, CA 92612
Tel: 949-263-1888 Fax: 949-263-1338

New York

150 Motor Parkway, Suite 202
Hauppauge, NY 11788
Tel: 631-273-5305 Fax: 631-273-5335

San Jose

Microchip Technology Inc.
2107 North First Street, Suite 590
San Jose, CA 95131
Tel: 408-436-7950 Fax: 408-436-7955

Toronto

6285 Northam Drive, Suite 108
Mississauga, Ontario L4V 1X5, Canada
Tel: 905-673-0699 Fax: 905-673-6509

ASIA/PACIFIC

Australia

Microchip Technology Australia Pty Ltd
Suite 22, 41 Rawson Street
Epping 2121, NSW
Australia
Tel: 61-2-9868-6733 Fax: 61-2-9868-6755

China - Beijing

Microchip Technology Consulting (Shanghai)
Co., Ltd., Beijing Liaison Office
Unit 915
Bei Hai Wan Tai Bldg.
No. 6 Chaoyangmen Beidajie
Beijing, 100027, No. China
Tel: 86-10-85282100 Fax: 86-10-85282104

China - Chengdu

Microchip Technology Consulting (Shanghai)
Co., Ltd., Chengdu Liaison Office
Rm. 2401, 24th Floor,
Ming Xing Financial Tower
No. 88 TIDU Street
Chengdu 610016, China
Tel: 86-28-6766200 Fax: 86-28-6766599

China - Fuzhou

Microchip Technology Consulting (Shanghai)
Co., Ltd., Fuzhou Liaison Office
Unit 28F, World Trade Plaza
No. 71 Wusi Road
Fuzhou 350001, China
Tel: 86-591-7503506 Fax: 86-591-7503521

China - Shanghai

Microchip Technology Consulting (Shanghai)
Co., Ltd.
Room 701, Bldg. B
Far East International Plaza
No. 317 Xian Xia Road
Shanghai, 200051
Tel: 86-21-6275-5700 Fax: 86-21-6275-5060

China - Shenzhen

Microchip Technology Consulting (Shanghai)
Co., Ltd., Shenzhen Liaison Office
Rm. 1315, 13/F, Shenzhen Kerry Centre,
Renminnan Lu
Shenzhen 518001, China
Tel: 86-755-2350361 Fax: 86-755-2366086

Hong Kong

Microchip Technology Hongkong Ltd.
Unit 901-6, Tower 2, Metroplaza
223 Hing Fong Road
Kwai Fong, N.T., Hong Kong
Tel: 852-2401-1200 Fax: 852-2401-3431

India

Microchip Technology Inc.
India Liaison Office
Divyasree Chambers
1 Floor, Wing A (A3/A4)
No. 11, O'Shaugnessey Road
Bangalore, 560 025, India
Tel: 91-80-2290061 Fax: 91-80-2290062

Japan

Microchip Technology Japan K.K.
Benex S-1 6F
3-18-20, Shinyokohama
Kohoku-Ku, Yokohama-shi
Kanagawa, 222-0033, Japan
Tel: 81-45-471- 6166 Fax: 81-45-471-6122

Korea

Microchip Technology Korea
168-1, Youngbo Bldg. 3 Floor
Samsung-Dong, Kangnam-Ku
Seoul, Korea 135-882
Tel: 82-2-554-7200 Fax: 82-2-558-5934

Singapore

Microchip Technology Singapore Pte Ltd.
200 Middle Road
#07-02 Prime Centre
Singapore, 188980
Tel: 65-334-8870 Fax: 65-334-8850

Taiwan

Microchip Technology Taiwan
11F-3, No. 207
Tung Hua North Road
Taipei, 105, Taiwan
Tel: 886-2-2717-7175 Fax: 886-2-2545-0139

EUROPE

Denmark

Microchip Technology Nordic ApS
Regus Business Centre
Lautrup høj 1-3
Ballerup DK-2750 Denmark
Tel: 45 4420 9895 Fax: 45 4420 9910

France

Microchip Technology SARL
Parc d'Activite du Moulin de Massy
43 Rue du Saule Trapu
Batiment A - 1er Etage
91300 Massy, France
Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79

Germany

Microchip Technology GmbH
Gustav-Heinemann Ring 125
D-81739 Munich, Germany
Tel: 49-89-627-144 0 Fax: 49-89-627-144-44

Italy

Microchip Technology SRL
Centro Direzionale Colleoni
Palazzo Taurus 1 V. Le Colleoni 1
20041 Agrate Brianza
Milan, Italy
Tel: 39-039-65791-1 Fax: 39-039-6899883

United Kingdom

Arizona Microchip Technology Ltd.
505 Eskdale Road
Winnersh Triangle
Wokingham
Berkshire, England RG41 5TU
Tel: 44 118 921 5869 Fax: 44-118 921-5820

01/18/02