



# PIC16C7X

## 8-Bit CMOS Microcontrollers with A/D Converter

### Devices included in this data sheet:

- PIC16C72
- PIC16C73
- PIC16C73A
- PIC16C74
- PIC16C74A
- PIC16C76
- PIC16C77

### PIC16C7X Microcontroller Core Features:

- High-performance RISC CPU
- Only 35 single word instructions to learn
- All single cycle instructions except for program branches which are two cycle
- Operating speed: DC - 20 MHz clock input  
DC - 200 ns instruction cycle
- Up to 8K x 14 words of Program Memory, up to 368 x 8 bytes of Data Memory (RAM)
- Interrupt capability
- Eight level deep hardware stack
- Direct, indirect, and relative addressing modes
- 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
- Low-power, high-speed CMOS EPROM technology
- Fully static design

- Wide operating voltage range: 2.5V to 6.0V
- High Sink/Source Current 25/25 mA
- Commercial, Industrial and Extended temperature ranges
- Low-power consumption:
  - < 2 mA @ 5V, 4 MHz
  - 15  $\mu$ A typical @ 3V, 32 kHz
  - < 1  $\mu$ A typical standby current

### PIC16C7X 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
- Capture, Compare, PWM module(s)
- 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 multichannel analog-to-digital converter
- Synchronous Serial Port (SSP) with SPI™ and I<sup>2</sup>C™
- Universal Synchronous Asynchronous Receiver Transmitter (USART/SCI)
- Parallel Slave Port (PSP) 8-bits wide, with external RD, WR and CS controls
- Brown-out detection circuitry for Brown-out Reset (BOR)

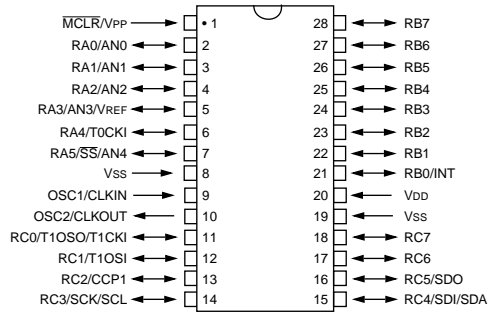
PIC16C7X Features	72	73	73A	74	74A	76	77
Program Memory (EPROM) x 14	2K	4K	4K	4K	4K	8K	8K
Data Memory (Bytes) x 8	128	192	192	192	192	368	368
I/O Pins	22	22	22	33	33	22	33
Parallel Slave Port	—	—	—	Yes	Yes	—	Yes
Capture/Compare/PWM Modules	1	2	2	2	2	2	2
Timer Modules	3	3	3	3	3	3	3
A/D Channels	5	5	5	8	8	5	8
Serial Communication	SPI/I <sup>2</sup> C	SPI/I <sup>2</sup> C, USART	SPI/I <sup>2</sup> C, USART	SPI/I <sup>2</sup> C, USART	SPI/I <sup>2</sup> C, USART	SPI/I <sup>2</sup> C, USART	SPI/I <sup>2</sup> C, USART
In-Circuit Serial Programming	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Brown-out Reset	Yes	—	Yes	—	Yes	Yes	Yes
Interrupt Sources	8	11	11	12	12	11	12



# PIC16C7X

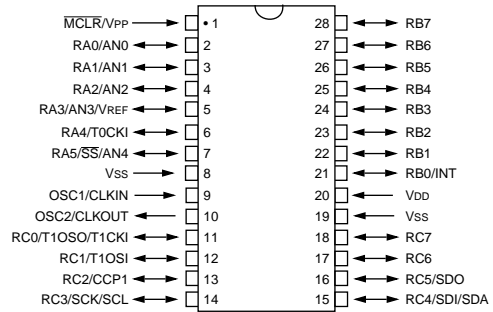
## Pin Diagrams

### SDIP, SOIC, Windowed Side Brazed Ceramic



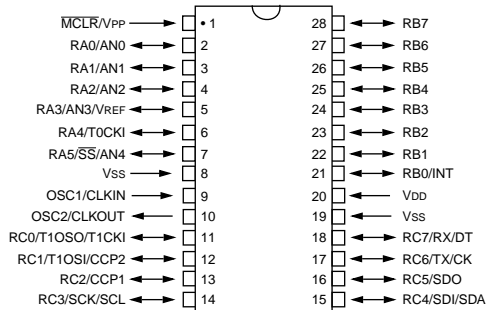
**PIC16C72**

### SSOP



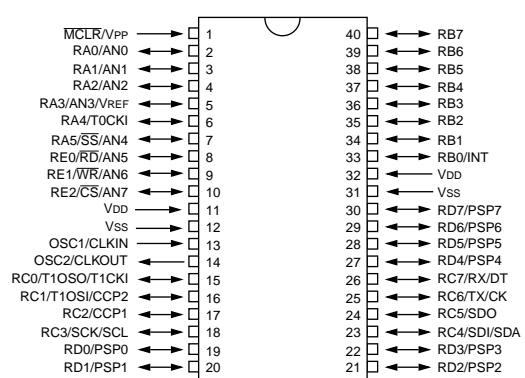
**PIC16C72**

### SDIP, SOIC, Windowed Side Brazed Ceramic



**PIC16C73  
PIC16C73A  
PIC16C76**

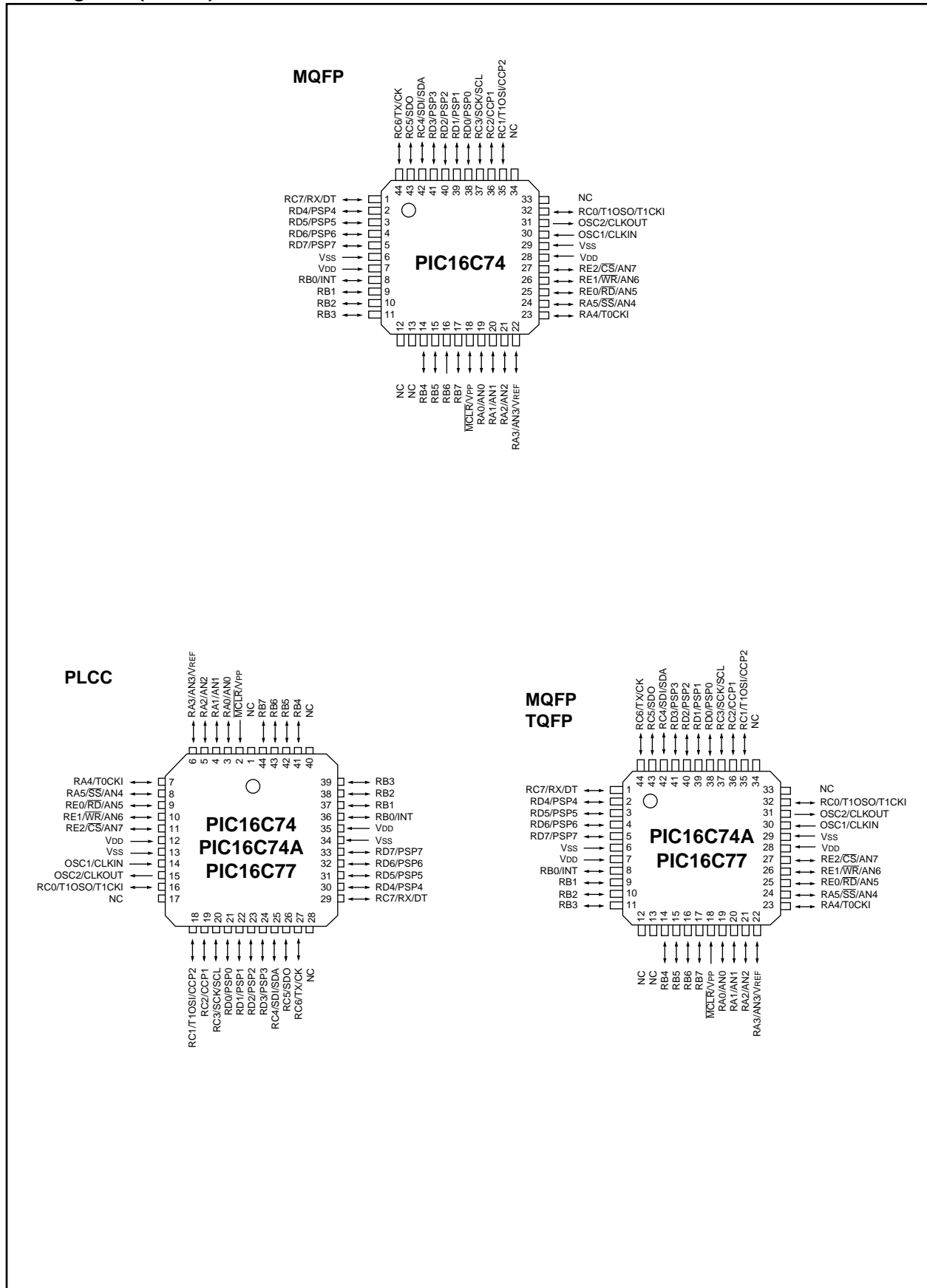
### PDIP, Windowed CERDIP



**PIC16C74  
PIC16C74A  
PIC16C77**

# PIC16C7X

## Pin Diagrams (Cont.'d)



# PIC16C7X

## Table of Contents

1.0	General Description .....	5
2.0	PIC16C7X Device Varieties .....	7
3.0	Architectural Overview .....	9
4.0	Memory Organization.....	19
5.0	I/O Ports.....	43
6.0	Overview of Timer Modules .....	57
7.0	Timer0 Module .....	59
8.0	Timer1 Module.....	65
9.0	Timer2 Module .....	69
10.0	Capture/Compare/PWM Module(s).....	71
11.0	Synchronous Serial Port (SSP) Module.....	77
12.0	Universal Synchronous Asynchronous Receiver Transmitter (USART) .....	99
13.0	Analog-to-Digital Converter (A/D) Module .....	117
14.0	Special Features of the CPU .....	129
15.0	Instruction Set Summary.....	147
16.0	Development Support .....	163
17.0	Electrical Characteristics for PIC16C72.....	167
18.0	Electrical Characteristics for PIC16C73/74.....	183
19.0	Electrical Characteristics for PIC16C73A/74A .....	201
20.0	Electrical Characteristics for PIC16C76/77 .....	219
21.0	DC and AC Characteristics Graphs and Tables .....	241
22.0	Packaging Information .....	251
Appendix A:	.....	263
Appendix B:	Compatibility .....	263
Appendix C:	What's New.....	264
Appendix D:	What's Changed .....	264
Appendix E:	PIC16/17 Microcontrollers .....	265
Pin Compatibility .....		271
Index .....		273
List of Examples.....		279
List of Figures.....		280
List of Tables.....		283
Reader Response .....		286
PIC16C7X Product Identification System.....		287

For register and module descriptions in this data sheet, device legends show which devices apply to those sections. As an example, the legend below would mean that the following section applies only to the PIC16C72, PIC16C73A and PIC16C74A devices.

Applicable Devices						
72	73	73A	74	74A	76	77

## To Our Valued Customers

We constantly strive to improve the quality of all our products and documentation. We have spent an exceptional amount of time to ensure that these documents are correct. However, we realize that we may have missed a few things. If you find any information that is missing or appears in error, please use the reader response form in the back of this data sheet to inform us. We appreciate your assistance in making this a better document.

# PIC16C7X

---

## 1.0 GENERAL DESCRIPTION

The PIC16C7X is a family of low-cost, high-performance, CMOS, fully-static, 8-bit microcontrollers with integrated analog-to-digital (A/D) converters, in the PIC16CXX mid-range family.

All PIC16/17 microcontrollers employ an advanced RISC architecture. The PIC16CXX microcontroller family has enhanced core features, eight-level deep stack, and multiple internal and external interrupt sources. The separate instruction and data buses of the Harvard architecture allow a 14-bit wide instruction word with the separate 8-bit wide data. The two stage instruction pipeline allows all instructions to execute in a single cycle, except for program branches which require two cycles. A total of 35 instructions (reduced instruction set) are available. Additionally, a large register set gives some of the architectural innovations used to achieve a very high performance.

PIC16CXX microcontrollers typically achieve a 2:1 code compression and a 4:1 speed improvement over other 8-bit microcontrollers in their class.

The **PIC16C72** has 128 bytes of RAM and 22 I/O pins. In addition several peripheral features are available including: three timer/counters, one Capture/Compare/PWM module and one serial port. The Synchronous Serial Port can be configured as either a 3-wire Serial Peripheral Interface (SPI) or the two-wire Inter-Integrated Circuit (I<sup>2</sup>C) bus. Also a 5-channel high-speed 8-bit A/D is provided. The 8-bit resolution is ideally suited for applications requiring low-cost analog interface, e.g. thermostat control, pressure sensing, etc.

The **PIC16C73/73A** devices have 192 bytes of RAM, while the **PIC16C76** has 368 bytes of RAM. Each device has 22 I/O pins. In addition, several peripheral features are available including: three timer/counters, two Capture/Compare/PWM modules and two serial ports. The Synchronous Serial Port can be configured as either a 3-wire Serial Peripheral Interface (SPI) or the two-wire Inter-Integrated Circuit (I<sup>2</sup>C) bus. The Universal Synchronous Asynchronous Receiver Transmitter (USART) is also known as the Serial Communications Interface or SCI. Also a 5-channel high-speed 8-bit A/D is provided. The 8-bit resolution is ideally suited for applications requiring low-cost analog interface, e.g. thermostat control, pressure sensing, etc.

The **PIC16C74/74A** devices have 192 bytes of RAM, while the **PIC16C77** has 368 bytes of RAM. Each device has 33 I/O pins. In addition several peripheral features are available including: three timer/counters, two Capture/Compare/PWM modules and two serial ports. The Synchronous Serial Port can be configured as either a 3-wire Serial Peripheral Interface (SPI) or the two-wire Inter-Integrated Circuit (I<sup>2</sup>C) bus. The Universal Synchronous Asynchronous Receiver Transmitter (USART) is also known as the Serial Communications Interface or SCI. An 8-bit Parallel Slave Port is provided. Also an 8-channel high-speed

8-bit A/D is provided. The 8-bit resolution is ideally suited for applications requiring low-cost analog interface, e.g. thermostat control, pressure sensing, etc.

The PIC16C7X family has special features to reduce external components, thus reducing cost, enhancing system reliability and reducing power consumption. There are four oscillator options, of which the single pin RC oscillator provides a low-cost solution, the LP oscillator minimizes power consumption, XT is a standard crystal, and the HS is for High Speed crystals. The SLEEP (power-down) feature provides a power saving mode. The user can wake up the chip from SLEEP through several external and internal interrupts and resets.

A highly reliable Watchdog Timer with its own on-chip RC oscillator provides protection against software lock-up.

A UV erasable CERDIP packaged version is ideal for code development while the cost-effective One-Time-Programmable (OTP) version is suitable for production in any volume.

The PIC16C7X family fits perfectly in applications ranging from security and remote sensors to appliance control and automotive. The EPROM technology makes customization of application programs (transmitter codes, motor speeds, receiver frequencies, etc.) extremely fast and convenient. The small footprint packages make this microcontroller series perfect for all applications with space limitations. Low cost, low power, high performance, ease of use and I/O flexibility make the PIC16C7X very versatile even in areas where no microcontroller use has been considered before (e.g. timer functions, serial communication, capture and compare, PWM functions and coprocessor applications).

### 1.1 Family and Upward Compatibility

Users familiar with the PIC16C5X microcontroller family will realize that this is an enhanced version of the PIC16C5X architecture. Please refer to Appendix A for a detailed list of enhancements. Code written for the PIC16C5X can be easily ported to the PIC16CXX family of devices (Appendix B).

### 1.2 Development Support

PIC16C7X devices are supported by the complete line of Microchip Development tools.

Please refer to Section 16.0 for more details about Microchip's development tools.

# PIC16C7X

TABLE 1-1: PIC16C7XX FAMILY OF DEVICES

		PIC16C710	PIC16C71	PIC16C711	PIC16C715	PIC16C72	PIC16CR72 <sup>(1)</sup>
Clock	Maximum Frequency of Operation (MHz)	20	20	20	20	20	20
	EPROM Program Memory (x14 words)	512	1K	1K	2K	2K	—
Memory	ROM Program Memory (14K words)	—	—	—	—	—	2K
	Data Memory (bytes)	36	36	68	128	128	128
Peripherals	Timer Module(s)	TMR0	TMR0	TMR0	TMR0	TMR0, TMR1, TMR2	TMR0, TMR1, TMR2
	Capture/Compare/PWM Module(s)	—	—	—	—	1	1
	Serial Port(s) (SPI/I <sup>2</sup> C, USART)	—	—	—	—	SPI/I <sup>2</sup> C	SPI/I <sup>2</sup> C
	Parallel Slave Port	—	—	—	—	—	—
Features	A/D Converter (8-bit) Channels	4	4	4	4	5	5
	Interrupt Sources	4	4	4	4	8	8
	I/O Pins	13	13	13	13	22	22
	Voltage Range (Volts)	3.0-6.0	3.0-6.0	3.0-6.0	3.0-5.5	2.5-6.0	3.0-5.5
	In-Circuit Serial Programming	Yes	Yes	Yes	Yes	Yes	Yes
	Brown-out Reset	Yes	—	Yes	Yes	Yes	Yes
	Packages	18-pin DIP, SOIC, 20-pin SSOP	18-pin DIP, SOIC	18-pin DIP, SOIC, 20-pin SSOP	18-pin DIP, SOIC, 20-pin SSOP	28-pin SDIP, SOIC, SSOP	28-pin SDIP, SOIC, SSOP

		PIC16C73A	PIC16C74A	PIC16C76	PIC16C77
Clock	Maximum Frequency of Operation (MHz)	20	20	20	20
	EPROM Program Memory (x14 words)	4K	4K	8K	8K
Memory	Data Memory (bytes)	192	192	368	368
	Timer Module(s)	TMR0, TMR1, TMR2	TMR0, TMR1, TMR2	TMR0, TMR1, TMR2	TMR0, TMR1, TMR2
Peripherals	Capture/Compare/PWM Module(s)	2	2	2	2
	Serial Port(s) (SPI/I <sup>2</sup> C, USART)	SPI/I <sup>2</sup> C, USART	SPI/I <sup>2</sup> C, USART	SPI/I <sup>2</sup> C, USART	SPI/I <sup>2</sup> C, USART
	Parallel Slave Port	—	Yes	—	Yes
	A/D Converter (8-bit) Channels	5	8	5	8
Features	Interrupt Sources	11	12	11	12
	I/O Pins	22	33	22	33
	Voltage Range (Volts)	2.5-6.0	2.5-6.0	2.5-6.0	2.5-6.0
	In-Circuit Serial Programming	Yes	Yes	Yes	Yes
	Brown-out Reset	Yes	Yes	Yes	Yes
	Packages	28-pin SDIP, SOIC	40-pin DIP; 44-pin PLCC, MQFP, TQFP	28-pin SDIP, SOIC	40-pin DIP; 44-pin PLCC, MQFP, TQFP

All PIC16/17 Family devices have Power-on Reset, selectable Watchdog Timer, selectable code protect and high I/O current capability. All PIC16C7XX Family devices use serial programming with clock pin RB6 and data pin RB7.

Note 1: Please contact your local Microchip sales office for availability of these devices.

## 2.0 PIC16C7X DEVICE VARIETIES

A variety of frequency ranges and packaging options are available. Depending on application and production requirements, the proper device option can be selected using the information in the PIC16C7X Product Identification System section at the end of this data sheet. When placing orders, please use that page of the data sheet to specify the correct part number.

For the PIC16C7X family, there are two device “types” as indicated in the device number:

1. **C**, as in PIC16**C**74. These devices have EPROM type memory and operate over the standard voltage range.
2. **LC**, as in PIC16**LC**74. These devices have EPROM type memory and operate over an extended voltage range.

### 2.1 UV Erasable Devices

The UV erasable version, offered in CERDIP package is optimal for prototype development and pilot programs. This version can be erased and reprogrammed to any of the oscillator modes.

Microchip's PICSTART<sup>®</sup> Plus and PRO MATE<sup>®</sup> II programmers both support programming of the PIC16C7X.

### 2.2 One-Time-Programmable (OTP) Devices

The availability of OTP devices is especially useful for customers who need the flexibility for frequent code updates and small volume applications.

The OTP devices, packaged in plastic packages, permit the user to program them once. In addition to the program memory, the configuration bits must also be programmed.

### 2.3 Quick-Turnaround-Production (QTP) Devices

Microchip offers a QTP Programming Service for factory production orders. This service is made available for users who choose not to program a medium to high quantity of units and whose code patterns have stabilized. The devices are identical to the OTP devices but with all EPROM locations and configuration options already programmed by the factory. Certain code and prototype verification procedures apply before production shipments are available. Please contact your local Microchip Technology sales office for more details.

### 2.4 Serialized Quick-Turnaround Production (SQTP<sup>SM</sup>) Devices

Microchip offers a unique programming service where a few user-defined locations in each device are programmed with different serial numbers. The serial numbers may be random, pseudo-random, or sequential.

Serial programming allows each device to have a unique number which can serve as an entry-code, password, or ID number.

# PIC16C7X

---

NOTES:



## 3.0 ARCHITECTURAL OVERVIEW

The high performance of the PIC16CXX family can be attributed to a number of architectural features commonly found in RISC microprocessors. To begin with, the PIC16CXX uses a Harvard architecture, in which, program and data are accessed from separate memories using separate buses. This improves bandwidth over traditional von Neumann architecture in which program and data are fetched from the same memory using the same bus. Separating program and data buses further allows instructions to be sized differently than the 8-bit wide data word. Instruction opcodes are 14-bits wide making it possible to have all single word instructions. A 14-bit wide program memory access bus fetches a 14-bit instruction in a single cycle. A two-stage pipeline overlaps fetch and execution of instructions (Example 3-1). Consequently, all instructions (35) execute in a single cycle (200 ns @ 20 MHz) except for program branches.

The table below lists program memory (EPROM) and data memory (RAM) for each PIC16C7X device.

Device	Program Memory	Data Memory
PIC16C72	2K x 14	128 x 8
PIC16C73	4K x 14	192 x 8
PIC16C73A	4K x 14	192 x 8
PIC16C74	4K x 14	192 x 8
PIC16C74A	4K x 14	192 x 8
PIC16C76	8K x 14	368 x 8
PIC16C77	8K x 14	386 x 8

The PIC16CXX can directly or indirectly address its register files or data memory. All special function registers, including the program counter, are mapped in the data memory. The PIC16CXX has an orthogonal (symmetrical) instruction set that makes it possible to carry out any operation on any register using any addressing mode. This symmetrical nature and lack of 'special optimal situations' make programming with the PIC16CXX simple yet efficient. In addition, the learning curve is reduced significantly.

PIC16CXX devices contain an 8-bit ALU and working register. The ALU is a general purpose arithmetic unit. It performs arithmetic and Boolean functions between the data in the working register and any register file.

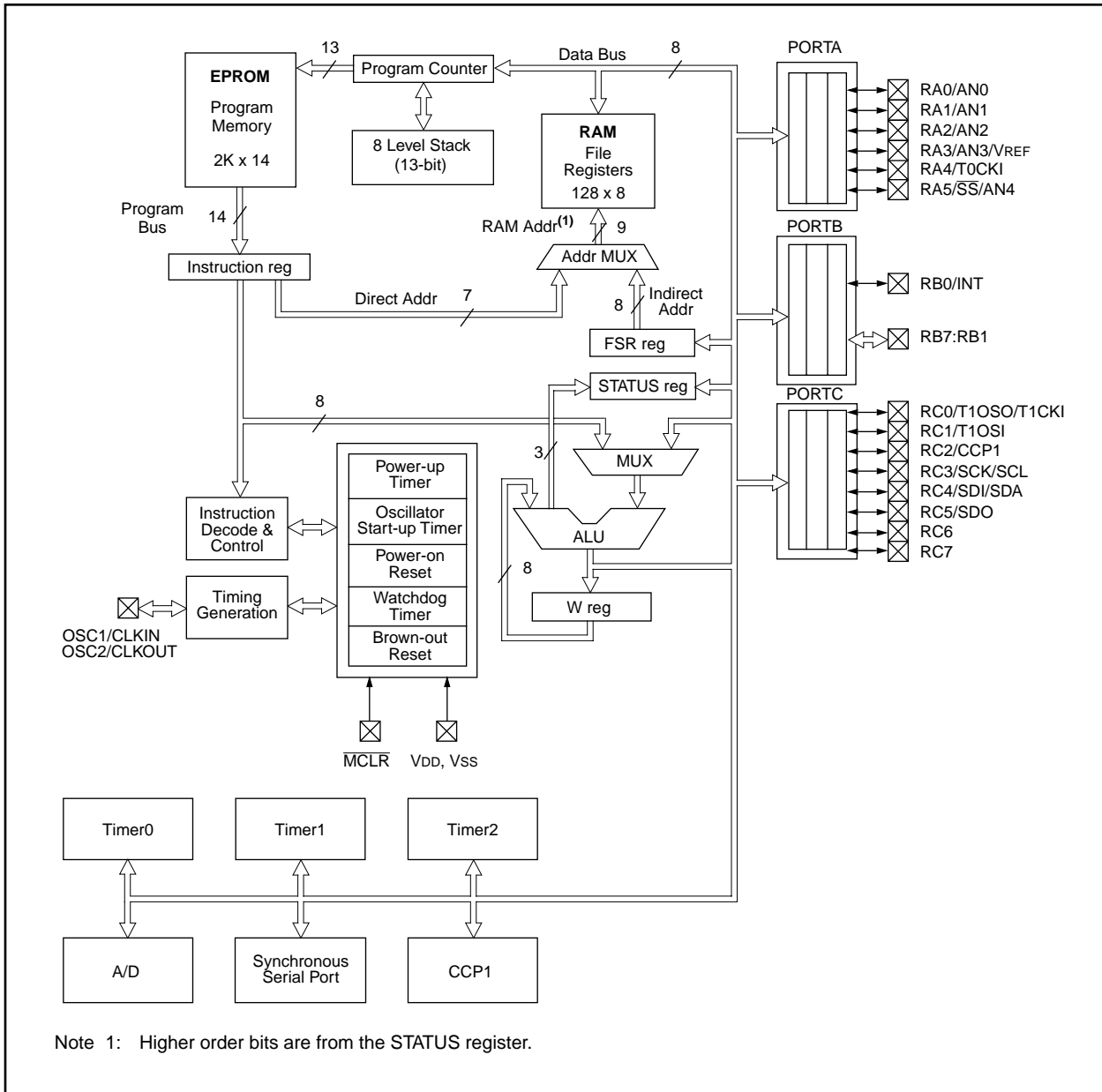
The ALU is 8-bits wide and capable of addition, subtraction, shift and logical operations. Unless otherwise mentioned, arithmetic operations are two's complement in nature. In two-operand instructions, typically one operand is the working register (W register). The other operand is a file register or an immediate constant. In single operand instructions, the operand is either the W register or a file register.

The W register is an 8-bit working register used for ALU operations. It is not an addressable register.

Depending on the instruction executed, the ALU may affect the values of the Carry (C), Digit Carry (DC), and Zero (Z) bits in the STATUS register. The C and DC bits operate as a borrow bit and a digit borrow out bit, respectively, in subtraction. See the `SUBLW` and `SUBWF` instructions for examples.

# PIC16C7X

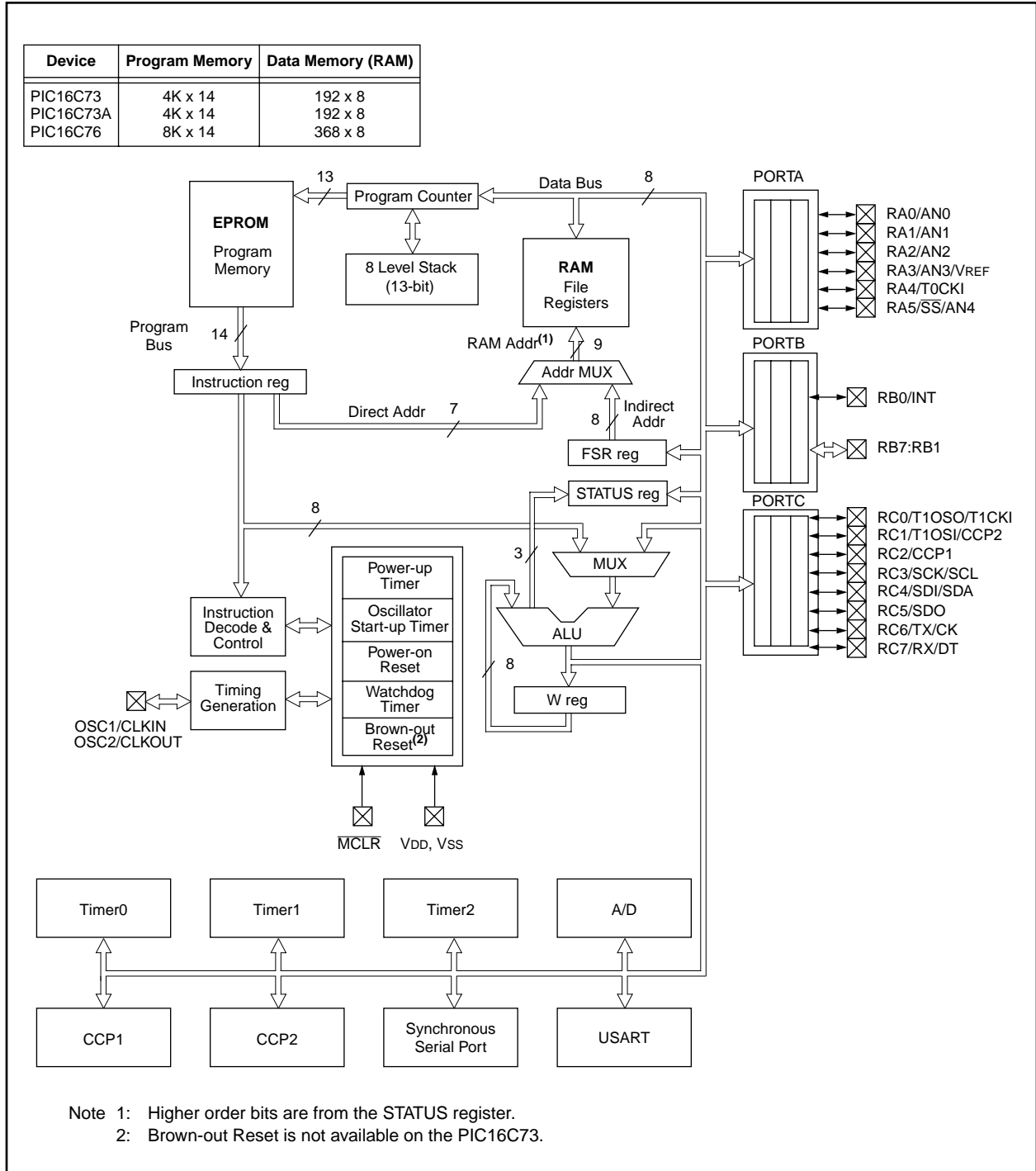
FIGURE 3-1: PIC16C72 BLOCK DIAGRAM



# PIC16C7X

FIGURE 3-2: PIC16C73/73A/76 BLOCK DIAGRAM

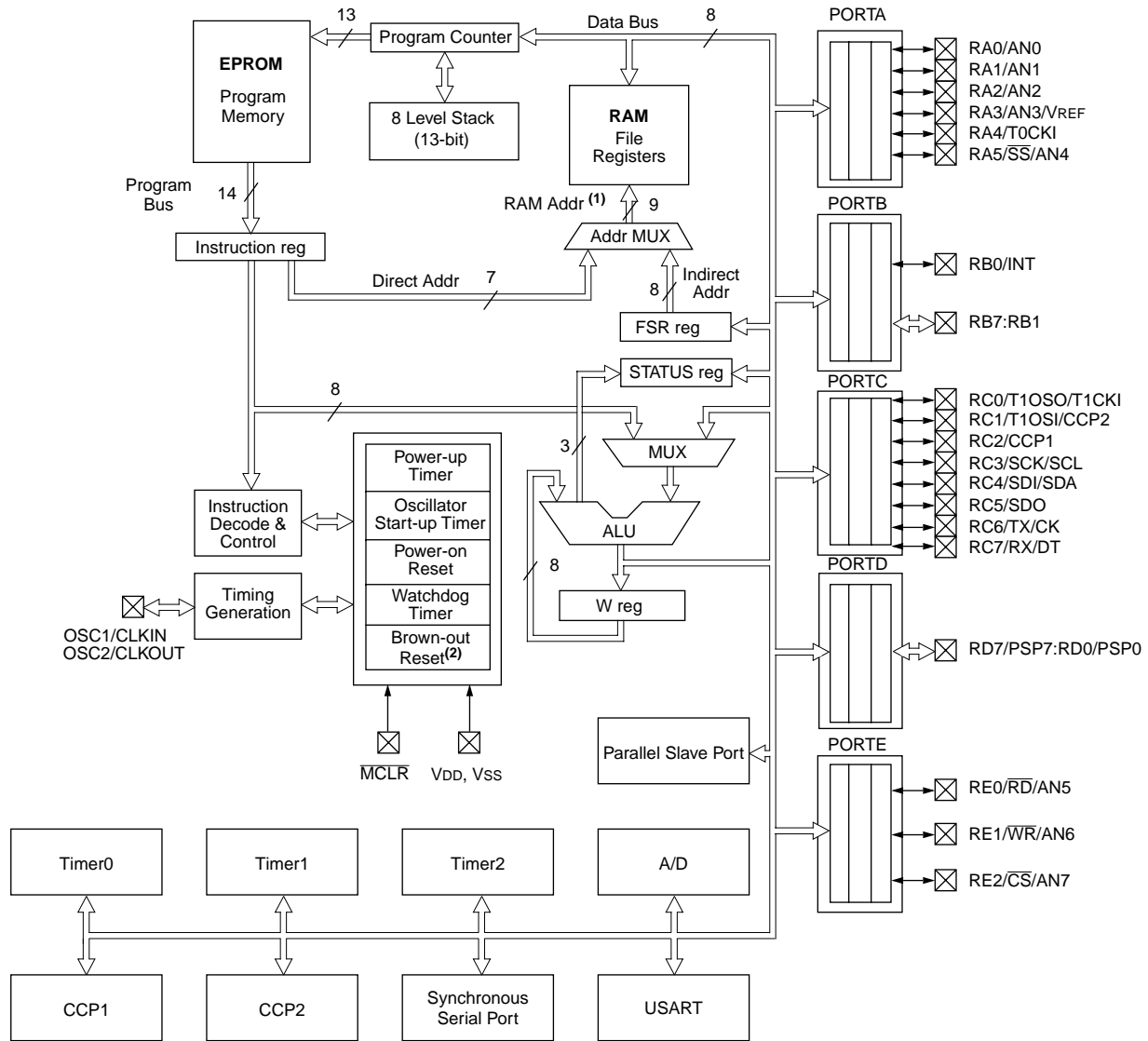
Device	Program Memory	Data Memory (RAM)
PIC16C73	4K x 14	192 x 8
PIC16C73A	4K x 14	192 x 8
PIC16C76	8K x 14	368 x 8



# PIC16C7X

FIGURE 3-3: PIC16C74/74A/77 BLOCK DIAGRAM

Device	Program Memory	Data Memory (RAM)
PIC16C74	4K x 14	192 x 8
PIC16C74A	4K x 14	192 x 8
PIC16C77	8K x 14	368 x 8



Note 1: Higher order bits are from the STATUS register.  
 Note 2: Brown-out Reset is not available on the PIC16C74.

# PIC16C7X

**TABLE 3-1: PIC16C72 PINOUT DESCRIPTION**

Pin Name	DIP Pin#	SSOP Pin#	SOIC Pin#	I/O/P Type	Buffer Type	Description
OSC1/CLKIN	9	9	9	I	ST/CMOS <sup>(3)</sup>	Oscillator crystal input/external clock source input.
OSC2/CLKOUT	10	10	10	O	—	Oscillator crystal output. Connects to crystal or resonator in crystal oscillator mode. In RC mode, the OSC2 pin outputs CLKOUT which has 1/4 the frequency of OSC1, and denotes the instruction cycle rate.
MCLR/VPP	1	1	1	I/P	ST	Master clear (reset) input or programming voltage input. This pin is an active low reset to the device.
RA0/AN0	2	2	2	I/O	TTL	PORTA is a bi-directional I/O port. RA0 can also be analog input0 RA1 can also be analog input1 RA2 can also be analog input2 RA3 can also be analog input3 or analog reference voltage RA4 can also be the clock input to the Timer0 module. Output is open drain type. RA5 can also be analog input4 or the slave select for the synchronous serial port.
RA1/AN1	3	3	3	I/O	TTL	
RA2/AN2	4	4	4	I/O	TTL	
RA3/AN3/VREF	5	5	5	I/O	TTL	
RA4/T0CKI	6	6	6	I/O	ST	
RA5/ $\overline{SS}$ /AN4	7	7	7	I/O	TTL	
RB0/INT	21	21	21	I/O	TTL/ST <sup>(1)</sup>	PORTB is a bi-directional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs. RB0 can also be the external interrupt pin.  Interrupt on change pin. Interrupt on change pin. Interrupt on change pin. Serial programming clock. Interrupt on change pin. Serial programming data.
RB1	22	22	22	I/O	TTL	
RB2	23	23	23	I/O	TTL	
RB3	24	24	24	I/O	TTL	
RB4	25	25	25	I/O	TTL	
RB5	26	26	26	I/O	TTL	
RB6	27	27	27	I/O	TTL/ST <sup>(2)</sup>	
RB7	28	28	28	I/O	TTL/ST <sup>(2)</sup>	
RC0/T1OSO/T1CKI	11	11	11	I/O	ST	PORTC is a bi-directional I/O port. RC0 can also be the Timer1 oscillator output or Timer1 clock input. RC1 can also be the Timer1 oscillator input. RC2 can also be the Capture1 input/Compare1 output/PWM1 output. RC3 can also be the synchronous serial clock input/output for both SPI and I <sup>2</sup> C modes. RC4 can also be the SPI Data In (SPI mode) or data I/O (I <sup>2</sup> C mode). RC5 can also be the SPI Data Out (SPI mode).
RC1/T1OSI	12	12	12	I/O	ST	
RC2/CCP1	13	13	13	I/O	ST	
RC3/SCK/SCL	14	14	14	I/O	ST	
RC4/SDI/SDA	15	15	15	I/O	ST	
RC5/SDO	16	16	16	I/O	ST	
RC6	17	17	17	I/O	ST	
RC7	18	18	18	I/O	ST	
V <sub>SS</sub>	8, 19	8, 19	8, 19	P	—	Ground reference for logic and I/O pins.
V <sub>DD</sub>	20	20	20	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.  
 Note 2: This buffer is a Schmitt Trigger input when used in serial programming mode.  
 Note 3: This buffer is a Schmitt Trigger input when configured in RC oscillator mode and a CMOS input otherwise.

# PIC16C7X

**TABLE 3-2: PIC16C73/73A/76 PINOUT DESCRIPTION**

Pin Name	DIP Pin#	SOIC Pin#	I/O/P Type	Buffer Type	Description
OSC1/CLKIN	9	9	I	ST/CMOS <sup>(3)</sup>	Oscillator crystal input/external clock source input.
OSC2/CLKOUT	10	10	O	—	Oscillator crystal output. Connects to crystal or resonator in crystal oscillator mode. In RC mode, the OSC2 pin outputs CLKOUT which has 1/4 the frequency of OSC1, and denotes the instruction cycle rate.
MCLR/VPP	1	1	I/P	ST	Master clear (reset) input or programming voltage input. This pin is an active low reset to the device.
RA0/AN0	2	2	I/O	TTL	PORTA is a bi-directional I/O port. RA0 can also be analog input0 RA1 can also be analog input1 RA2 can also be analog input2 RA3 can also be analog input3 or analog reference voltage RA4 can also be the clock input to the Timer0 module. Output is open drain type. RA5 can also be analog input4 or the slave select for the synchronous serial port.
RA1/AN1	3	3	I/O	TTL	
RA2/AN2	4	4	I/O	TTL	
RA3/AN3/VREF	5	5	I/O	TTL	
RA4/T0CKI	6	6	I/O	ST	
RA5/SS/AN4	7	7	I/O	TTL	
RB0/INT	21	21	I/O	TTL/ST <sup>(1)</sup>	PORTB is a bi-directional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs. RB0 can also be the external interrupt pin.  Interrupt on change pin. Interrupt on change pin. Interrupt on change pin. Serial programming clock. Interrupt on change pin. Serial programming data.
RB1	22	22	I/O	TTL	
RB2	23	23	I/O	TTL	
RB3	24	24	I/O	TTL	
RB4	25	25	I/O	TTL	
RB5	26	26	I/O	TTL	
RB6	27	27	I/O	TTL/ST <sup>(2)</sup>	
RB7	28	28	I/O	TTL/ST <sup>(2)</sup>	
RC0/T1OSO/T1CKI	11	11	I/O	ST	PORTC is a bi-directional I/O port. RC0 can also be the Timer1 oscillator output or Timer1 clock input. RC1 can also be the Timer1 oscillator input or Capture2 input/Compare2 output/PWM2 output. RC2 can also be the Capture1 input/Compare1 output/PWM1 output. RC3 can also be the synchronous serial clock input/output for both SPI and I <sup>2</sup> C modes. RC4 can also be the SPI Data In (SPI mode) or data I/O (I <sup>2</sup> C mode). RC5 can also be the SPI Data Out (SPI mode). RC6 can also be the USART Asynchronous Transmit or Synchronous Clock. RC7 can also be the USART Asynchronous Receive or Synchronous Data.
RC1/T1OSI/CCP2	12	12	I/O	ST	
RC2/CCP1	13	13	I/O	ST	
RC3/SCK/SCL	14	14	I/O	ST	
RC4/SDI/SDA	15	15	I/O	ST	
RC5/SDO	16	16	I/O	ST	
RC6/TX/CK	17	17	I/O	ST	
RC7/RX/DT	18	18	I/O	ST	
Vss	8, 19	8, 19	P	—	Ground reference for logic and I/O pins.
Vdd	20	20	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.

# PIC16C7X

**TABLE 3-3: PIC16C74/74A/77 PINOUT DESCRIPTION**

Pin Name	DIP Pin#	PLCC Pin#	QFP Pin#	I/O/P Type	Buffer Type	Description
OSC1/CLKIN	13	14	30	I	ST/CMOS <sup>(4)</sup>	Oscillator crystal input/external clock source input.
OSC2/CLKOUT	14	15	31	O	—	Oscillator crystal output. Connects to crystal or resonator in crystal oscillator mode. In RC mode, OSC2 pin outputs CLKOUT which has 1/4 the frequency of OSC1, and denotes the instruction cycle rate.
MCLR/VPP	1	2	18	I/P	ST	Master clear (reset) input or programming voltage input. This pin is an active low reset to the device.
RA0/AN0	2	3	19	I/O	TTL	PORTA is a bi-directional I/O port. RA0 can also be analog input0 RA1 can also be analog input1 RA2 can also be analog input2 RA3 can also be analog input3 or analog reference voltage RA4 can also be the clock input to the Timer0 timer/counter. Output is open drain type. RA5 can also be analog input4 or the slave select for the synchronous serial port.
RA1/AN1	3	4	20	I/O	TTL	
RA2/AN2	4	5	21	I/O	TTL	
RA3/AN3/VREF	5	6	22	I/O	TTL	
RA4/T0CKI	6	7	23	I/O	ST	
RA5/SS/AN4	7	8	24	I/O	TTL	
RB0/INT	33	36	8	I/O	TTL/ST <sup>(1)</sup>	PORTB is a bi-directional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs. RB0 can also be the external interrupt pin.  Interrupt on change pin. Interrupt on change pin. Interrupt on change pin. Serial programming clock. Interrupt on change pin. Serial programming data.
RB1	34	37	9	I/O	TTL	
RB2	35	38	10	I/O	TTL	
RB3	36	39	11	I/O	TTL	
RB4	37	41	14	I/O	TTL	
RB5	38	42	15	I/O	TTL	
RB6	39	43	16	I/O	TTL/ST <sup>(2)</sup>	
RB7	40	44	17	I/O	TTL/ST <sup>(2)</sup>	

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

- Note 1: This buffer is a Schmitt Trigger input when configured as an external interrupt.  
 Note 2: This buffer is a Schmitt Trigger input when used in serial programming mode.  
 Note 3: This buffer is a Schmitt Trigger input when configured as general purpose I/O and a TTL input when used in the Parallel Slave Port mode (for interfacing to a microprocessor bus).  
 Note 4: This buffer is a Schmitt Trigger input when configured in RC oscillator mode and a CMOS input otherwise.

# PIC16C7X

**TABLE 3-3: PIC16C74/74A/77 PINOUT DESCRIPTION (Cont.'d)**

Pin Name	DIP Pin#	PLCC Pin#	QFP Pin#	I/O/P Type	Buffer Type	Description
RC0/T1OSO/T1CKI	15	16	32	I/O	ST	PORTC is a bi-directional I/O port. RC0 can also be the Timer1 oscillator output or a Timer1 clock input.
RC1/T1OSI/CCP2	16	18	35	I/O	ST	RC1 can also be the Timer1 oscillator input or Capture2 input/Compare2 output/PWM2 output.
RC2/CCP1	17	19	36	I/O	ST	RC2 can also be the Capture1 input/Compare1 output/PWM1 output.
RC3/SCK/SCL	18	20	37	I/O	ST	RC3 can also be the synchronous serial clock input/output for both SPI and I <sup>2</sup> C modes.
RC4/SDI/SDA	23	25	42	I/O	ST	RC4 can also be the SPI Data In (SPI mode) or data I/O (I <sup>2</sup> C mode).
RC5/SDO	24	26	43	I/O	ST	RC5 can also be the SPI Data Out (SPI mode).
RC6/TX/CK	25	27	44	I/O	ST	RC6 can also be the USART Asynchronous Transmit or Synchronous Clock.
RC7/RX/DT	26	29	1	I/O	ST	RC7 can also be the USART Asynchronous Receive or Synchronous Data.
RD0/PSP0	19	21	38	I/O	ST/TTL <sup>(3)</sup>	PORTD is a bi-directional I/O port or parallel slave port when interfacing to a microprocessor bus.
RD1/PSP1	20	22	39	I/O	ST/TTL <sup>(3)</sup>	
RD2/PSP2	21	23	40	I/O	ST/TTL <sup>(3)</sup>	
RD3/PSP3	22	24	41	I/O	ST/TTL <sup>(3)</sup>	
RD4/PSP4	27	30	2	I/O	ST/TTL <sup>(3)</sup>	
RD5/PSP5	28	31	3	I/O	ST/TTL <sup>(3)</sup>	
RD6/PSP6	29	32	4	I/O	ST/TTL <sup>(3)</sup>	
RD7/PSP7	30	33	5	I/O	ST/TTL <sup>(3)</sup>	
RE0/RD $\bar{}$ /AN5	8	9	25	I/O	ST/TTL <sup>(3)</sup>	PORTE is a bi-directional I/O port. RE0 can also be read control for the parallel slave port, or analog input5.
RE1/WR $\bar{}$ /AN6	9	10	26	I/O	ST/TTL <sup>(3)</sup>	RE1 can also be write control for the parallel slave port, or analog input6.
RE2/CS $\bar{}$ /AN7	10	11	27	I/O	ST/TTL <sup>(3)</sup>	RE2 can also be select control for the parallel slave port, or analog input7.
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.



# PIC16C7X

## 3.1 Clocking Scheme/Instruction Cycle

The clock input (from OSC1) is internally divided by four to generate four non-overlapping quadrature clocks namely Q1, Q2, Q3 and Q4. Internally, the program counter (PC) is incremented every Q1, the instruction is fetched from the program memory and latched into the instruction register in Q4. The instruction is decoded and executed during the following Q1 through Q4. The clocks and instruction execution flow is shown in Figure 3-4.

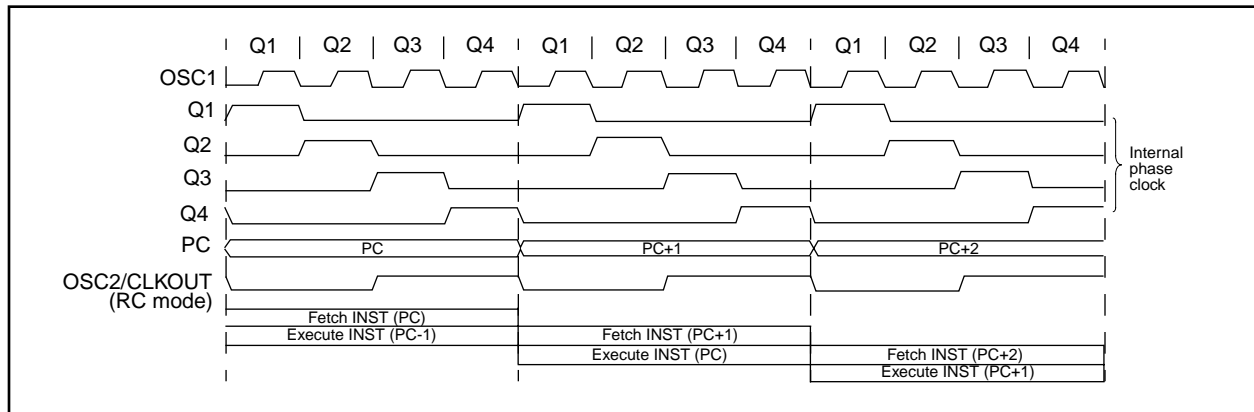
## 3.2 Instruction Flow/Pipelining

An "Instruction Cycle" consists of four Q cycles (Q1, Q2, Q3 and Q4). The instruction fetch and execute are pipelined such that fetch takes one instruction cycle while decode and execute takes another instruction cycle. However, due to the pipelining, each instruction effectively executes in one cycle. If an instruction causes the program counter to change (e.g. GOTO) then two cycles are required to complete the instruction (Example 3-1).

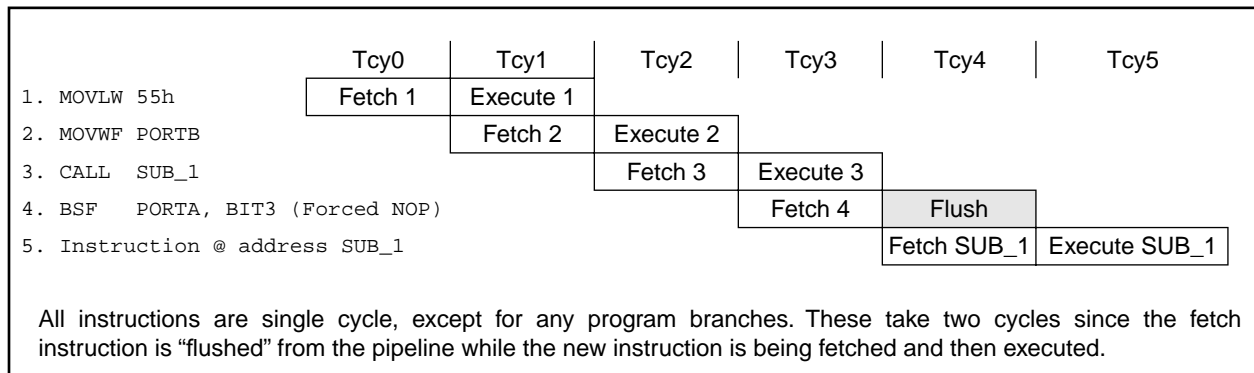
A fetch cycle begins with the program counter (PC) incrementing in Q1.

In the execution cycle, the fetched instruction is latched into the "Instruction Register" (IR) in cycle Q1. This instruction is then decoded and executed during the Q2, Q3, and Q4 cycles. Data memory is read during Q2 (operand read) and written during Q4 (destination write).

**FIGURE 3-4: CLOCK/INSTRUCTION CYCLE**



**EXAMPLE 3-1: INSTRUCTION PIPELINE FLOW**



# PIC16C7X

---

NOTES:

# PIC16C7X

## 4.0 MEMORY ORGANIZATION

Applicable Devices
72 73 73A 74 74A 76 77

### 4.1 Program Memory Organization

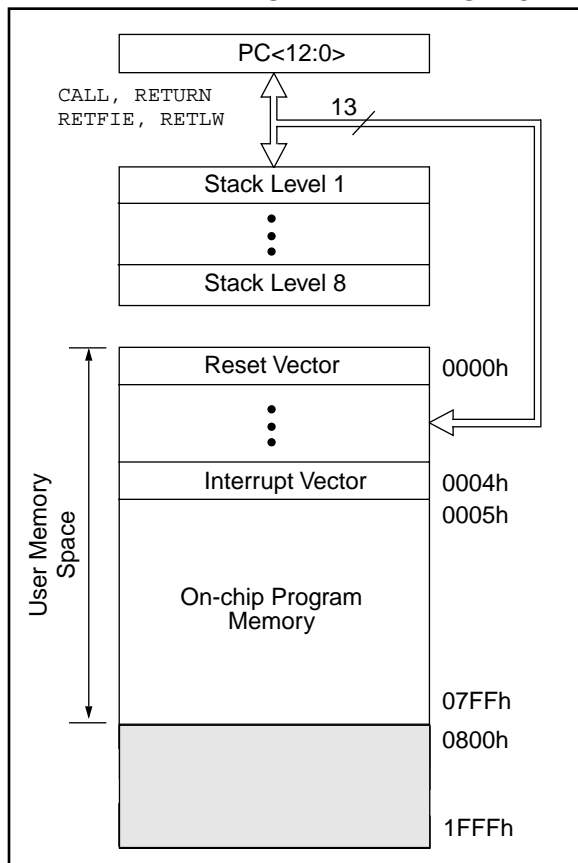
The PIC16C7X family has a 13-bit program counter capable of addressing an 8K x 14 program memory space. The amount of program memory available to each device is listed below:

Device	Program Memory	Address Range
PIC16C72	2K x 14	0000h-07FFh
PIC16C73	4K x 14	0000h-0FFFh
PIC16C73A	4K x 14	0000h-0FFFh
PIC16C74	4K x 14	0000h-0FFFh
PIC16C74A	4K x 14	0000h-0FFFh
PIC16C76	8K x 14	0000h-1FFFh
PIC16C77	8K x 14	0000h-1FFFh

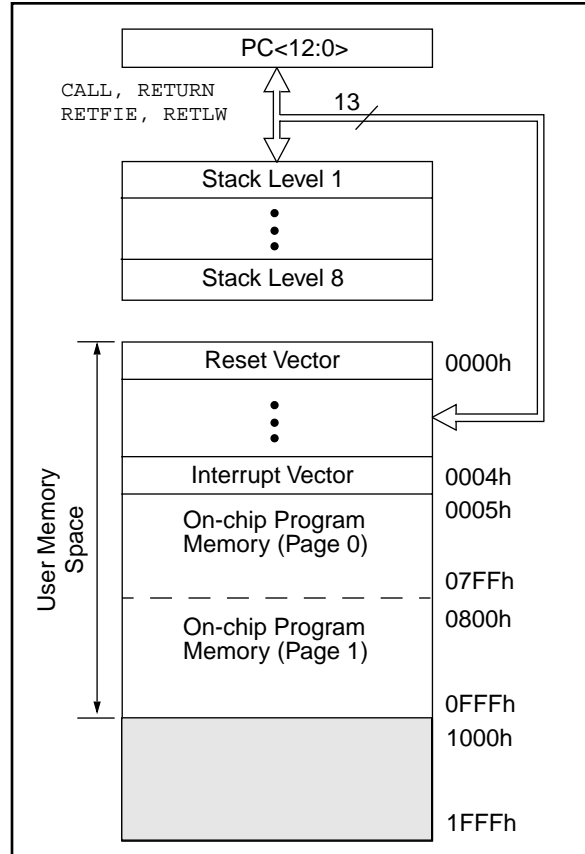
For those devices with less than 8K program memory, accessing a location above the physically implemented address will cause a wraparound.

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

**FIGURE 4-1: PIC16C72 PROGRAM MEMORY MAP AND STACK**

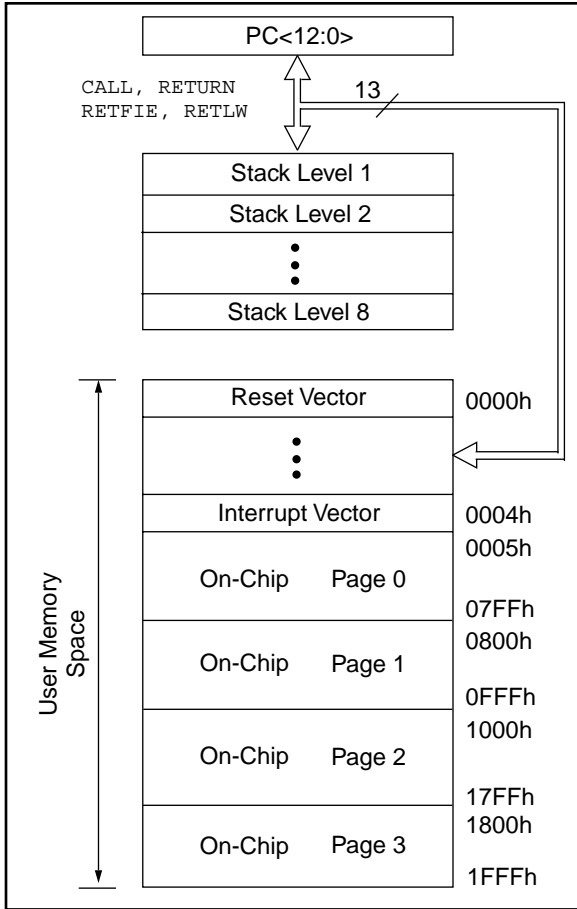


**FIGURE 4-2: PIC16C73/73A/74/74A PROGRAM MEMORY MAP AND STACK**



# PIC16C7X

**FIGURE 4-3: PIC16C76/77 PROGRAM MEMORY MAP AND STACK**



## 4.2 Data Memory Organization

Applicable Devices			
72	73	73A	74
74A	76	77	

The data memory is partitioned into multiple banks which contain the General Purpose Registers and the Special Function Registers. Bits RP1 and RP0 are the bank select bits.

- RP1:RP0 (STATUS<6:5>)
- = 00 → Bank0
  - = 01 → Bank1
  - = 10 → Bank2
  - = 11 → Bank3

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 "high use" special function registers from one bank may be mirrored in another bank for code reduction and quicker access.

### 4.2.1 GENERAL PURPOSE REGISTER FILE

The register file can be accessed either directly, or indirectly through the File Select Register FSR (Section 4.5).

# PIC16C7X

**FIGURE 4-4: PIC16C72 REGISTER FILE MAP**

File Address			File Address	
00h	INDF <sup>(1)</sup>	INDF <sup>(1)</sup>	80h	
01h	TMR0	OPTION	81h	
02h	PCL	PCL	82h	
03h	STATUS	STATUS	83h	
04h	FSR	FSR	84h	
05h	PORTA	TRISA	85h	
06h	PORTB	TRISB	86h	
07h	PORTC	TRISC	87h	
08h			88h	
09h			89h	
0Ah	PCLATH	PCLATH	8Ah	
0Bh	INTCON	INTCON	8Bh	
0Ch	PIR1	PIE1	8Ch	
0Dh			8Dh	
0Eh	TMR1L	PCON	8Eh	
0Fh	TMR1H		8Fh	
10h	T1CON		90h	
11h	TMR2		91h	
12h	T2CON	PR2	92h	
13h	SSPBUF	SSPADD	93h	
14h	SSPCON	SSPSTAT	94h	
15h	CCPR1L		95h	
16h	CCPR1H		96h	
17h	CCP1CON		97h	
18h			98h	
19h			99h	
1Ah			9Ah	
1Bh			9Bh	
1Ch			9Ch	
1Dh			9Dh	
1Eh	ADRES		9Eh	
1Fh	ADCON0	ADCON1	9Fh	
20h	General Purpose Register	General Purpose Register	A0h	
				Bf
				C0h
7Fh			Ff	

Bank 0                      Bank 1

Unimplemented data memory locations, read as '0'.  
 Note 1: Not a physical register.

**FIGURE 4-5: PIC16C73/73A/74/74A REGISTER FILE MAP**

File Address			File Address	
00h	INDF <sup>(1)</sup>	INDF <sup>(1)</sup>	80h	
01h	TMR0	OPTION	81h	
02h	PCL	PCL	82h	
03h	STATUS	STATUS	83h	
04h	FSR	FSR	84h	
05h	PORTA	TRISA	85h	
06h	PORTB	TRISB	86h	
07h	PORTC	TRISC	87h	
08h	PORTD <sup>(2)</sup>	TRISD <sup>(2)</sup>	88h	
09h	PORTE <sup>(2)</sup>	TRISE <sup>(2)</sup>	89h	
0Ah	PCLATH	PCLATH	8Ah	
0Bh	INTCON	INTCON	8Bh	
0Ch	PIR1	PIE1	8Ch	
0Dh	PIR2	PIE2	8Dh	
0Eh	TMR1L	PCON	8Eh	
0Fh	TMR1H		8Fh	
10h	T1CON		90h	
11h	TMR2		91h	
12h	T2CON	PR2	92h	
13h	SSPBUF	SSPADD	93h	
14h	SSPCON	SSPSTAT	94h	
15h	CCPR1L		95h	
16h	CCPR1H		96h	
17h	CCP1CON		97h	
18h	RCSTA	TXSTA	98h	
19h	TXREG	SPBRG	99h	
1Ah	RCREG		9Ah	
1Bh	CCPR2L		9Bh	
1Ch	CCPR2H		9Ch	
1Dh	CCP2CON		9Dh	
1Eh	ADRES		9Eh	
1Fh	ADCON0	ADCON1	9Fh	
20h	General Purpose Register	General Purpose Register	A0h	
7Fh			Ff	

Bank 0                      Bank 1

Unimplemented data memory locations, read as '0'.  
 Note 1: Not a physical register.  
 Note 2: These registers are not physically implemented on the PIC16C73/73A, read as '0'.

# PIC16C7X

FIGURE 4-6: PIC16C76/77 REGISTER FILE MAP

						File Address	
Indirect addr.(*)	00h	Indirect addr.(*)	80h	Indirect addr.(*)	100h	Indirect addr.(*)	180h
TMR0	01h	OPTION	81h	TMR0	101h	OPTION	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 <sup>(1)</sup>	08h	TRISD <sup>(1)</sup>	88h		108h		188h
PORTE <sup>(1)</sup>	09h	TRISE <sup>(1)</sup>	89h		109h		189h
PCLATH	0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah
INTCON	0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh
PIR1	0Ch	PIE1	8Ch		10Ch		18Ch
PIR2	0Dh	PIE2	8Dh		10Dh		18Dh
TMR1L	0Eh	PCON	8Eh		10Eh		18Eh
TMR1H	0Fh		8Fh		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	General Purpose Register 16 Bytes	117h	General Purpose Register 16 Bytes	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	
		accesses 70h-7Fh	EFh F0h	accesses 70h-7Fh	16Fh 170h	accesses 70h - 7Fh	1EFh 1F0h
Bank 0	7Fh	Bank 1	FFh	Bank 2	17Fh	Bank 3	1FFh

■ Unimplemented data memory locations, read as '0'.

\* Not a physical register.

Note 1: PORTD, PORTE, TRISD, and TRISE are unimplemented on the PIC16C76, read as '0'.

**Note:** The upper 16 bytes of data memory in banks 1, 2, and 3 are mapped in Bank 0. This may require relocation of data memory usage in the user application code if upgrading to the PIC16C76/77.

# PIC16C7X

## 4.2.2 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.

The special function registers can be classified into two sets (core and peripheral). Those registers associated with the “core” functions are described in this section, and those related to the operation of the peripheral features are described in the section of that peripheral feature.

**TABLE 4-1: PIC16C72 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	Value on all other resets (3)
<b>Bank 0</b>											
00h <sup>(1)</sup>	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	0000 0000
01h	TMR0	Timer0 module's register								xxxx xxxx	uuuu uuuu
02h <sup>(1)</sup>	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	0000 0000
03h <sup>(1)</sup>	STATUS	IRP <sup>(4)</sup>	RP1 <sup>(4)</sup>	RP0	T0	PD	Z	DC	C	0001 1xxx	000q quuu
04h <sup>(1)</sup>	FSR	Indirect data memory address pointer								xxxx xxxx	uuuu uuuu
05h	PORTA	—	—	PORTA Data Latch when written: PORTA pins when read						--0x 0000	--0u 0000
06h	PORTB	PORTB Data Latch when written: PORTB pins when read								xxxx xxxx	uuuu uuuu
07h	PORTC	PORTC Data Latch when written: PORTC pins when read								xxxx xxxx	uuuu uuuu
08h	—	Unimplemented								—	—
09h	—	Unimplemented								—	—
0Ah <sup>(1,2)</sup>	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	---0 0000
0Bh <sup>(1)</sup>	INTCON	GIE	PEIE	T0IE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	—	ADIF	—	—	SSPIF	CCP1IF	TMR2IF	TMR1IF	-0-- 0000	-0-- 0000
0Dh	—	Unimplemented								—	—
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
11h	TMR2	Timer2 module's register								0000 0000	0000 0000
12h	T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	-000 0000
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
15h	CCPR1L	Capture/Compare/PWM Register (LSB)								xxxx xxxx	uuuu uuuu
16h	CCPR1H	Capture/Compare/PWM Register (MSB)								xxxx xxxx	uuuu uuuu
17h	CCP1CON	—	—	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	--00 0000
18h	—	Unimplemented								—	—
19h	—	Unimplemented								—	—
1Ah	—	Unimplemented								—	—
1Bh	—	Unimplemented								—	—
1Ch	—	Unimplemented								—	—
1Dh	—	Unimplemented								—	—
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

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

Note 1: These registers can be addressed from either bank.

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

Note 3: Other (non power-up) resets include external reset through MCLR and Watchdog Timer Reset.

Note 4: The IRP and RP1 bits are reserved on the PIC16C72, always maintain these bits clear.

# PIC16C7X

**TABLE 4-1: PIC16C72 SPECIAL FUNCTION REGISTER SUMMARY (Cont'd)**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other resets (3)
<b>Bank 1</b>											
80h <sup>(1)</sup>	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	0000 0000
81h	OPTION	RBP $\bar{U}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
82h <sup>(1)</sup>	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	0000 0000
83h <sup>(1)</sup>	STATUS	IRP <sup>(4)</sup>	RP1 <sup>(4)</sup>	RP0	$\bar{T}O$	$\bar{P}D$	Z	DC	C	0001 1xxx	000q quuu
84h <sup>(1)</sup>	FSR	Indirect data memory address pointer								xxxx xxxx	uuuu uuuu
85h	TRISA	—	—	PORTA Data Direction Register						--11 1111	--11 1111
86h	TRISB	PORTB Data Direction Register								1111 1111	1111 1111
87h	TRISC	PORTC Data Direction Register								1111 1111	1111 1111
88h	—	Unimplemented								—	—
89h	—	Unimplemented								—	—
8Ah <sup>(1,2)</sup>	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the PC					---0 0000	---0 0000
8Bh <sup>(1)</sup>	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
8Ch	PIE1	—	ADIE	—	—	SSPIE	CCP1IE	TMR2IE	TMR1IE	-0-- 0000	-0-- 0000
8Dh	—	Unimplemented								—	—
8Eh	PCON	—	—	—	—	—	—	$\bar{P}OR$	$\bar{B}OR$	---- --qg	---- --uu
8Fh	—	Unimplemented								—	—
90h	—	Unimplemented								—	—
91h	—	Unimplemented								—	—
92h	PR2	Timer2 Period Register								1111 1111	1111 1111
93h	SSPADD	Synchronous Serial Port (I <sup>2</sup> C mode) Address Register								0000 0000	0000 0000
94h	SSPSTAT	—	—	D $\bar{A}$	P	S	R $\bar{W}$	UA	BF	--00 0000	--00 0000
95h	—	Unimplemented								—	—
96h	—	Unimplemented								—	—
97h	—	Unimplemented								—	—
98h	—	Unimplemented								—	—
99h	—	Unimplemented								—	—
9Ah	—	Unimplemented								—	—
9Bh	—	Unimplemented								—	—
9Ch	—	Unimplemented								—	—
9Dh	—	Unimplemented								—	—
9Eh	—	Unimplemented								—	—
9Fh	ADCON1	—	—	—	—	—	PCFG2	PCFG1	PCFG0	---- -000	---- -000

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

- Note 1: These registers can be addressed from either bank.  
 2: The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8> whose contents are transferred to the upper byte of the program counter.  
 3: Other (non power-up) resets include external reset through MCLR and Watchdog Timer Reset.  
 4: The IRP and RP1 bits are reserved on the PIC16C72, always maintain these bits clear.



# PIC16C7X

**TABLE 4-2: PIC16C73/73A/74/74A 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	Value on all other resets (2)	
<b>Bank 0</b>												
00h <sup>(4)</sup>	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	0000 0000	
01h	TMR0	Timer0 module's register								xxxx xxxx	uuuu uuuu	
02h <sup>(4)</sup>	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	0000 0000	
03h <sup>(4)</sup>	STATUS	IRP <sup>(7)</sup>	RP1 <sup>(7)</sup>	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxxx	000q quuu	
04h <sup>(4)</sup>	FSR	Indirect data memory address pointer								xxxx xxxx	uuuu uuuu	
05h	PORTA	—	—	PORTA Data Latch when written: PORTA pins when read							--0x 0000	--0u 0000
06h	PORTB	PORTB Data Latch when written: PORTB pins when read								xxxx xxxx	uuuu uuuu	
07h	PORTC	PORTC Data Latch when written: PORTC pins when read								xxxx xxxx	uuuu uuuu	
08h <sup>(5)</sup>	PORTD	PORTD Data Latch when written: PORTD pins when read								xxxx xxxx	uuuu uuuu	
09h <sup>(5)</sup>	PORTE	—	—	—	—	—	RE2	RE1	RE0	---- -xxx	---- -uuu	
0Ah <sup>(1,4)</sup>	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	---0 0000	
0Bh <sup>(4)</sup>	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000u	
0Ch	PIR1	PSPIF <sup>(3)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000	
0Dh	PIR2	—	—	—	—	—	—	—	CCP2IF	---- --0	---- --0	
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	
11h	TMR2	Timer2 module's register								0000 0000	0000 0000	
12h	T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	-000 0000	
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	
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	RCSTA	SPEN	RX9	SREN	CREN	—	FERR	OERR	RX9D	0000 -00x	0000 -00x	
19h	TXREG	USART Transmit Data Register								0000 0000	0000 0000	
1Ah	RCREG	USART Receive Data Register								0000 0000	0000 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	
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	

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented read as '0'.  
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.
- Other (non power-up) resets include external reset through MCLR and Watchdog Timer Reset.
  - Bits PSPIE and PSPIF are reserved on the PIC16C73/73A, always maintain these bits clear.
  - These registers can be addressed from either bank.
  - PORTD and PORTE are not physically implemented on the PIC16C73/73A, read as '0'.
  - Brown-out Reset is not implemented on the PIC16C73 or the PIC16C74, read as '0'.
  - The IRP and RP1 bits are reserved on the PIC16C73/73A/74/74A, always maintain these bits clear.

# PIC16C7X

**TABLE 4-2: PIC16C73/73A/74/74A SPECIAL FUNCTION REGISTER SUMMARY (Cont'd)**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other resets (2)
<b>Bank 1</b>											
80h <sup>(4)</sup>	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	0000 0000
81h	OPTION	RBP $\bar{U}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
82h <sup>(4)</sup>	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	0000 0000
83h <sup>(4)</sup>	STATUS	IRP <sup>(7)</sup>	RP1 <sup>(7)</sup>	RP0	TO	PD	Z	DC	C	0001 1xxx	000q quuu
84h <sup>(4)</sup>	FSR	Indirect data memory address pointer								xxxx xxxx	uuuu uuuu
85h	TRISA	—	—	PORTA Data Direction Register						--11 1111	--11 1111
86h	TRISB	PORTB Data Direction Register								1111 1111	1111 1111
87h	TRISC	PORTC Data Direction Register								1111 1111	1111 1111
88h <sup>(5)</sup>	TRISD	PORTD Data Direction Register								1111 1111	1111 1111
89h <sup>(5)</sup>	TRISE	IBF	OBF	IBOV	PSPMODE	—	PORTE Data Direction Bits			0000 -111	0000 -111
8Ah <sup>(1,4)</sup>	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	---0 0000
8Bh <sup>(4)</sup>	INTCON	GIE	PEIE	TOIE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
8Ch	PIE1	PSPIE <sup>(3)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
8Dh	PIE2	—	—	—	—	—	—	—	CCP2IE	---- -000	---- -000
8Eh	PCON	—	—	—	—	—	—	POR	BOR <sup>(6)</sup>	---- -qqq	---- -uuu
8Fh	—	Unimplemented								—	—
90h	—	Unimplemented								—	—
91h	—	Unimplemented								—	—
92h	PR2	Timer2 Period Register								1111 1111	1111 1111
93h	SSPADD	Synchronous Serial Port (I <sup>2</sup> C mode) Address Register								0000 0000	0000 0000
94h	SSPSTAT	—	—	D/ $\bar{A}$	P	S	R/ $\bar{W}$	UA	BF	--00 0000	--00 0000
95h	—	Unimplemented								—	—
96h	—	Unimplemented								—	—
97h	—	Unimplemented								—	—
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000
9Ah	—	Unimplemented								—	—
9Bh	—	Unimplemented								—	—
9Ch	—	Unimplemented								—	—
9Dh	—	Unimplemented								—	—
9Eh	—	Unimplemented								—	—
9Fh	ADCON1	—	—	—	—	—	PCFG2	PCFG1	PCFG0	---- -000	---- -000

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented read as '0'.  
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.
- 2: Other (non power-up) resets include external reset through MCLR and Watchdog Timer Reset.
- 3: Bits PSPIE and PSPIF are reserved on the PIC16C73/73A, always maintain these bits clear.
- 4: These registers can be addressed from either bank.
- 5: PORTD and PORTE are not physically implemented on the PIC16C73/73A, read as '0'.
- 6: Brown-out Reset is not implemented on the PIC16C73 or the PIC16C74, read as '0'.
- 7: The IRP and RP1 bits are reserved on the PIC16C73/73A/74/74A, always maintain these bits clear.

# PIC16C7X

**TABLE 4-3: PIC16C76/77 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	Value on all other resets (2)
<b>Bank 0</b>											
00h <sup>(4)</sup>	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	0000 0000
01h	TMR0	Timer0 module's register								xxxx xxxx	uuuu uuuu
02h <sup>(4)</sup>	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	0000 0000
03h <sup>(4)</sup>	STATUS	IRP	RP1	RP0	T $\bar{O}$	P $\bar{D}$	Z	DC	C	0001 1xxx	000q quuu
04h <sup>(4)</sup>	FSR	Indirect data memory address pointer								xxxx xxxx	uuuu uuuu
05h	PORTA	—	—	PORTA Data Latch when written: PORTA pins when read						--0x 0000	--0u 0000
06h	PORTB	PORTB Data Latch when written: PORTB pins when read								xxxx xxxx	uuuu uuuu
07h	PORTC	PORTC Data Latch when written: PORTC pins when read								xxxx xxxx	uuuu uuuu
08h <sup>(5)</sup>	PORTD	PORTD Data Latch when written: PORTD pins when read								xxxx xxxx	uuuu uuuu
09h <sup>(5)</sup>	PORTE	—	—	—	—	—	RE2	RE1	RE0	---- -xxx	---- -uuu
0Ah <sup>(1,4)</sup>	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	---0 0000
0Bh <sup>(4)</sup>	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF <sup>(3)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
0Dh	PIR2	—	—	—	—	—	—	—	CCP2IF	---- --0	---- --0
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
11h	TMR2	Timer2 module's register								0000 0000	0000 0000
12h	T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	-000 0000
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
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	RCSTA	SPEN	RX9	SREN	CREN	—	FERR	OERR	RX9D	0000 -00x	0000 -00x
19h	TXREG	USART Transmit Data Register								0000 0000	0000 0000
1Ah	RCREG	USART Receive Data Register								0000 0000	0000 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
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

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented read as '0'.  
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.
- 2: Other (non power-up) resets include external reset through MCLR and Watchdog Timer Reset.
- 3: Bits PSPIE and PSPIF are reserved on the PIC16C76, always maintain these bits clear.
- 4: These registers can be addressed from any bank.
- 5: PORTD and PORTE are not physically implemented on the PIC16C76, read as '0'.

# PIC16C7X

**TABLE 4-3: PIC16C76/77 SPECIAL FUNCTION REGISTER SUMMARY (Cont'd)**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other resets (2)	
<b>Bank 1</b>												
80h <sup>(4)</sup>	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	0000 0000	
81h	OPTION	RBP $\bar{U}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111	
82h <sup>(4)</sup>	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	0000 0000	
83h <sup>(4)</sup>	STATUS	IRP	RP1	RP0	$\bar{T}O$	$\bar{P}D$	Z	DC	C	0001 1xxx	000q quuu	
84h <sup>(4)</sup>	FSR	Indirect data memory address pointer								xxxx xxxx	uuuu uuuu	
85h	TRISA	—	—	PORTA Data Direction Register						--11 1111	--11 1111	
86h	TRISB	PORTB Data Direction Register								1111 1111	1111 1111	
87h	TRISC	PORTC Data Direction Register								1111 1111	1111 1111	
88h <sup>(5)</sup>	TRISD	PORTD Data Direction Register								1111 1111	1111 1111	
89h <sup>(5)</sup>	TRISE	IBF	OBF	IBOV	PSPMODE	—	PORTE Data Direction Bits				0000 -111	0000 -111
8Ah <sup>(1,4)</sup>	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter						---0 0000	---0 0000
8Bh <sup>(4)</sup>	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u	
8Ch	PIE1	PSPIE <sup>(3)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000	
8Dh	PIE2	—	—	—	—	—	—	—	CCP2IE	---- --0	---- --0	
8Eh	PCON	—	—	—	—	—	—	POR	BOR	---- --qq	---- --uu	
8Fh	—	Unimplemented								—	—	
90h	—	Unimplemented								—	—	
91h	—	Unimplemented								—	—	
92h	PR2	Timer2 Period Register								1111 1111	1111 1111	
93h	SSPADD	Synchronous Serial Port (I <sup>2</sup> C mode) Address Register								0000 0000	0000 0000	
94h	SSPSTAT	SMP	CKE	D/ $\bar{A}$	P	S	R/ $\bar{W}$	UA	BF	0000 0000	0000 0000	
95h	—	Unimplemented								—	—	
96h	—	Unimplemented								—	—	
97h	—	Unimplemented								—	—	
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010	
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000	
9Ah	—	Unimplemented								—	—	
9Bh	—	Unimplemented								—	—	
9Ch	—	Unimplemented								—	—	
9Dh	—	Unimplemented								—	—	
9Eh	—	Unimplemented								—	—	
9Fh	ADCON1	—	—	—	—	—	PCFG2	PCFG1	PCFG0	---- -000	---- -000	

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

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.
- 2: Other (non power-up) resets include external reset through MCLR and Watchdog Timer Reset.
- 3: Bits PSPIE and PSPIF are reserved on the PIC16C76, always maintain these bits clear.
- 4: These registers can be addressed from any bank.
- 5: PORTD and PORTE are not physically implemented on the PIC16C76, read as '0'.

# PIC16C7X

**TABLE 4-3: PIC16C76/77 SPECIAL FUNCTION REGISTER SUMMARY (Cont'd)**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other resets (2)
<b>Bank 2</b>											
100h <sup>(4)</sup>	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	0000 0000
101h	TMR0	Timer0 module's register								xxxx xxxx	uuuu uuuu
102h <sup>(4)</sup>	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	0000 0000
103h <sup>(4)</sup>	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	000q quuu
104h <sup>(4)</sup>	FSR	Indirect data memory address pointer								xxxx xxxx	uuuu uuuu
105h	—	Unimplemented								—	—
106h	PORTB	PORTB Data Latch when written: PORTB pins when read								xxxx xxxx	uuuu uuuu
107h	—	Unimplemented								—	—
108h	—	Unimplemented								—	—
109h	—	Unimplemented								—	—
10Ah <sup>(1,4)</sup>	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	---0 0000
10Bh <sup>(4)</sup>	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000u
10Ch-10Fh	—	Unimplemented								—	—
<b>Bank 3</b>											
180h <sup>(4)</sup>	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	0000 0000
181h	OPTION	$\overline{RBPU}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
182h <sup>(4)</sup>	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	0000 0000
183h <sup>(4)</sup>	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	000q quuu
184h <sup>(4)</sup>	FSR	Indirect data memory address pointer								xxxx xxxx	uuuu uuuu
185h	—	Unimplemented								—	—
186h	TRISB	PORTB Data Direction Register								1111 1111	1111 1111
187h	—	Unimplemented								—	—
188h	—	Unimplemented								—	—
189h	—	Unimplemented								—	—
18Ah <sup>(1,4)</sup>	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	---0 0000
18Bh <sup>(4)</sup>	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000u
18Ch-18Fh	—	Unimplemented								—	—

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

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.
- 2: Other (non power-up) resets include external reset through  $\overline{MCLR}$  and Watchdog Timer Reset.
- 3: Bits PSPIE and PSPIF are reserved on the PIC16C76, always maintain these bits clear.
- 4: These registers can be addressed from any bank.
- 5: PORTD and PORTE are not physically implemented on the PIC16C76, read as '0'.

# PIC16C7X

## 4.2.2.1 STATUS REGISTER

Applicable Devices						
72	73	73A	74	74A	76	77

The STATUS register, shown in Figure 4-7, contains the arithmetic status of the ALU, the RESET status and the bank select bits for data memory.

The STATUS register can be the destination for any instruction, as with any other register. If the STATUS register is the destination for an instruction that affects the Z, DC or C bits, then the write to these three bits is disabled. These bits are set or cleared according to the device logic. Furthermore, the  $\overline{TO}$  and  $\overline{PD}$  bits are not writable. Therefore, the result of an instruction with the STATUS register as destination may be different than intended.

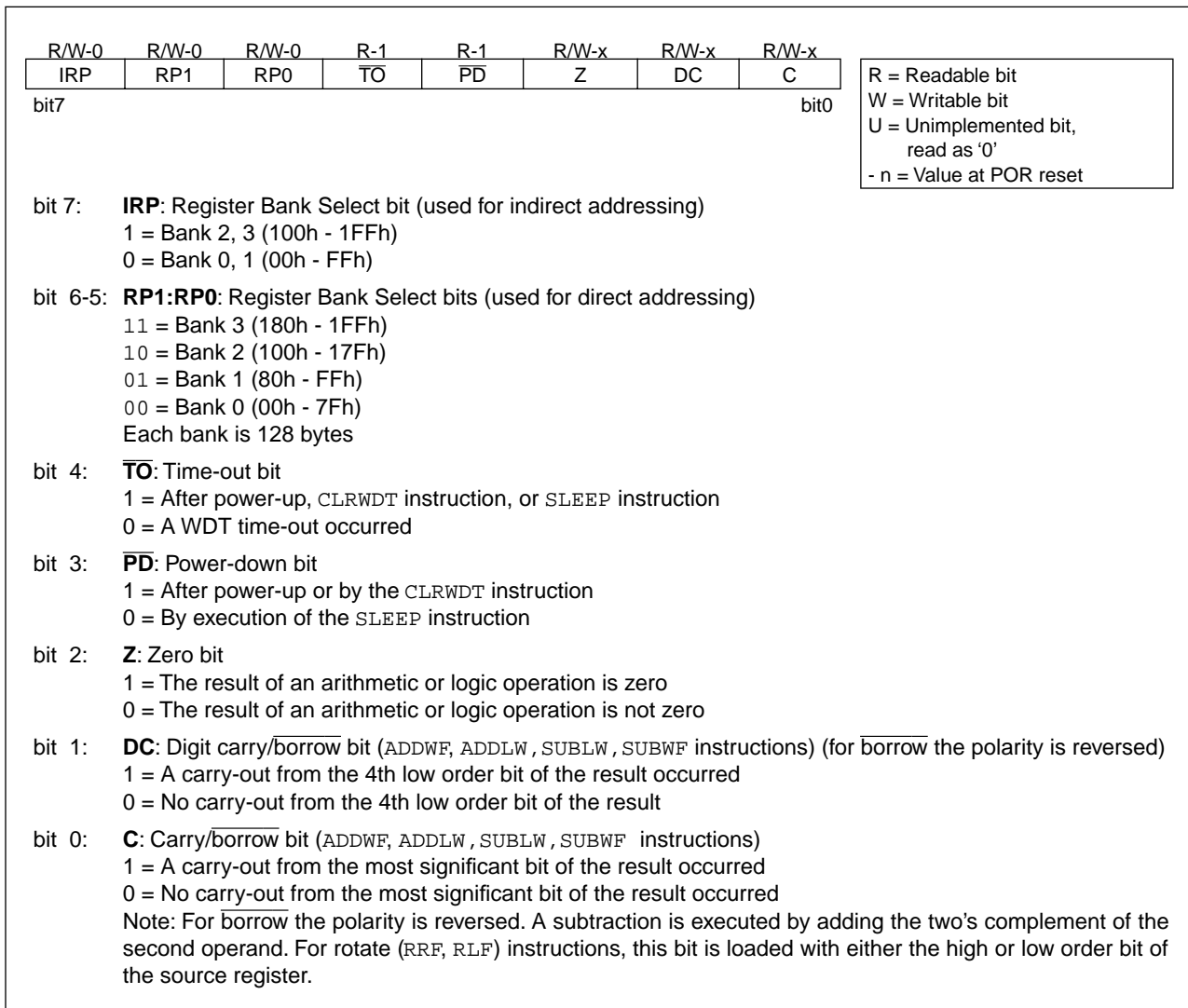
For example, `CLRF STATUS` will clear the upper-three bits and set the Z bit. This leaves the STATUS register as `000u u1uu` (where u = unchanged).

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

**Note 1:** For those devices that do not use bits IRP and RP1 (STATUS<7:6>), maintain these bits clear to ensure upward compatibility with future products.

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

**FIGURE 4-7: STATUS REGISTER (ADDRESS 03h, 83h, 103h, 183h)**



# PIC16C7X

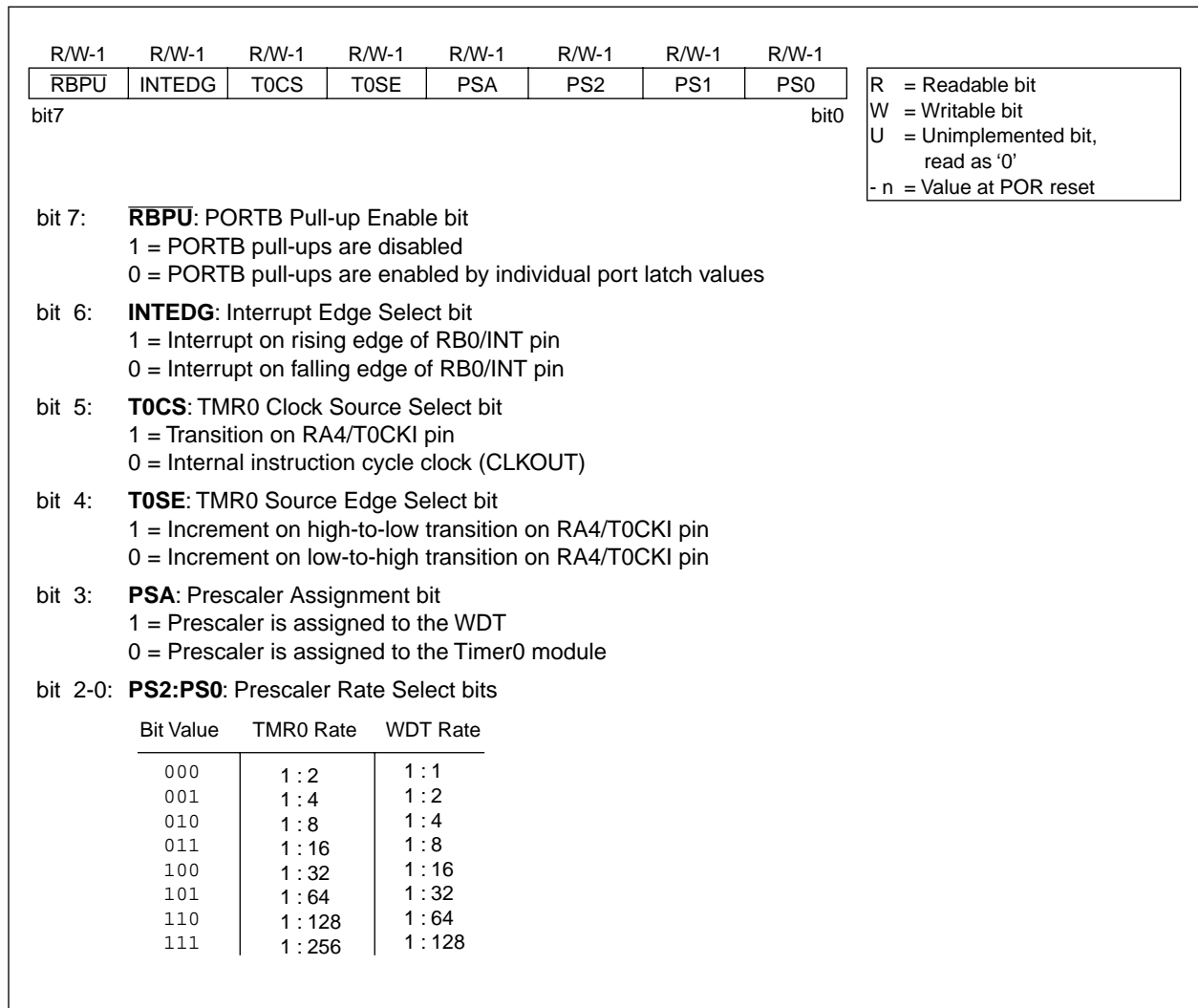
## 4.2.2.2 OPTION REGISTER

Applicable Devices							
72	73	73A	74	74A	76	77	

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

The OPTION register is a readable and writable register which contains various control bits to configure the TMR0/WDT prescaler, the External INT Interrupt, TMR0, and the weak pull-ups on PORTB.

**FIGURE 4-8: OPTION REGISTER (ADDRESS 81h, 181h)**



# PIC16C7X

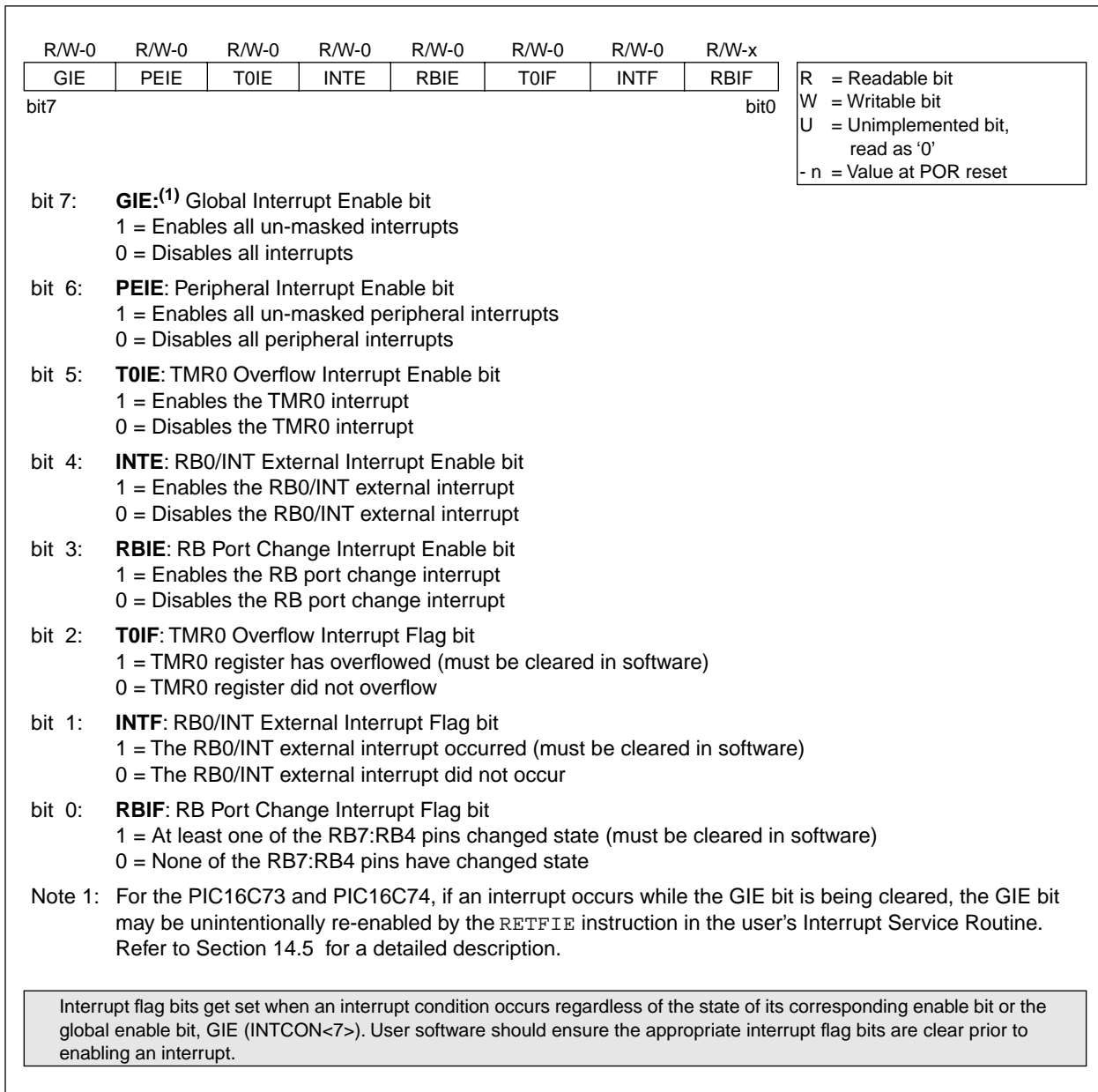
## 4.2.2.3 INTCON REGISTER

Applicable Devices							
72	73	73A	74	74A	76	77	

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

**Note:** Interrupt flag bits get set when an interrupt condition occurs regardless of the state of its corresponding enable bit or the global enable bit, GIE (INTCON<7>).

**FIGURE 4-9: INTCON REGISTER  
(ADDRESS 0Bh, 8Bh, 10Bh, 18Bh)**





## 4.2.2.4 PIE1 REGISTER

Applicable Devices							
72	73	73A	74	74A	76	77	

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

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

**FIGURE 4-10: PIE1 REGISTER PIC16C72 (ADDRESS 8Ch)**

U-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	ADIE	—	—	SSPIE	CCP1IE	TMR2IE	TMR1IE
bit7							bit0

R = Readable bit  
W = Writable bit  
U = Unimplemented bit, read as '0'  
- n = Value at POR reset

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

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

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

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

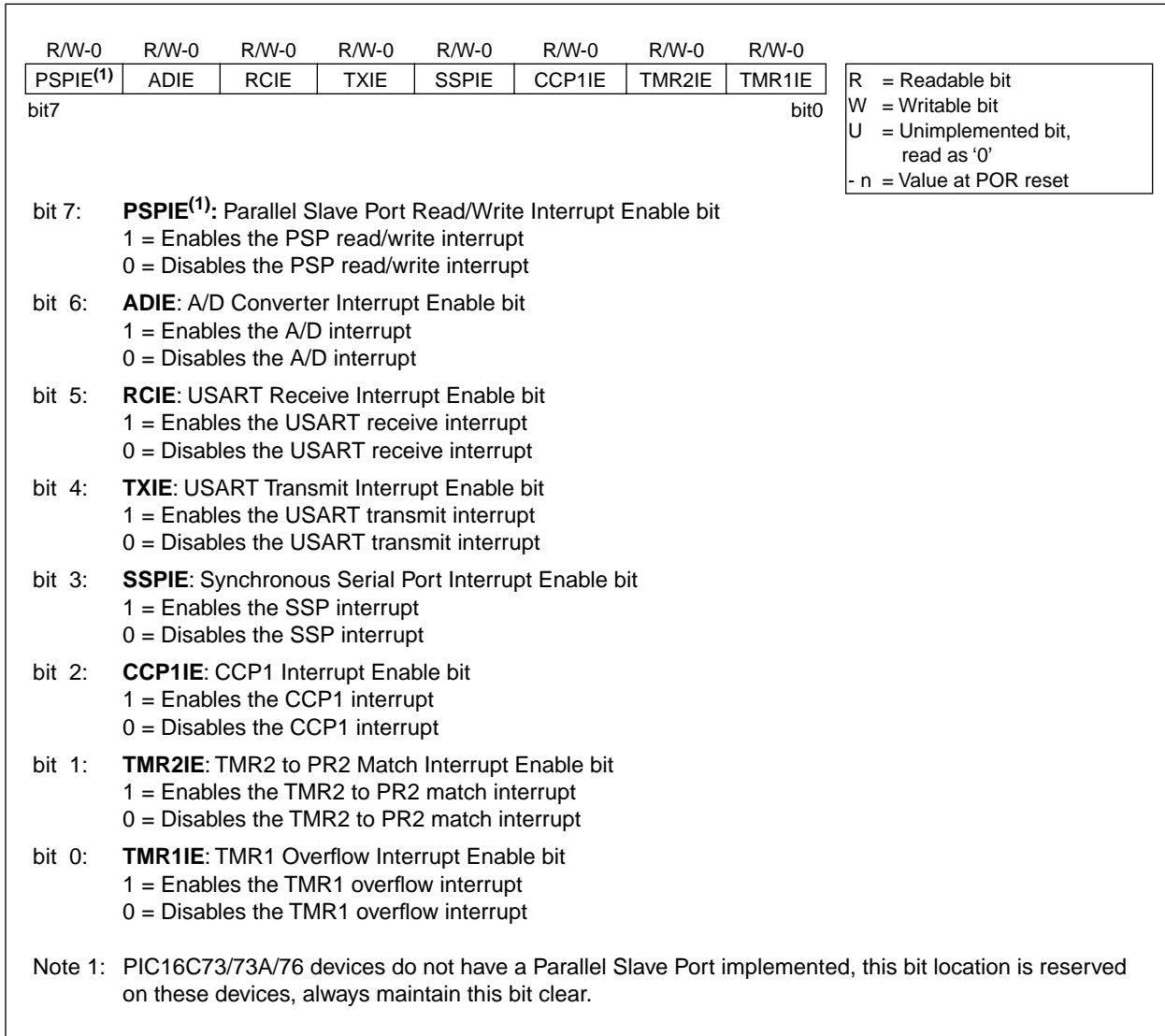
bit 2: **CCP1IE:** CCP1 Interrupt Enable bit  
1 = Enables the CCP1 interrupt  
0 = Disables the CCP1 interrupt

bit 1: **TMR2IE:** TMR2 to PR2 Match Interrupt Enable bit  
1 = Enables the TMR2 to PR2 match interrupt  
0 = Disables the TMR2 to PR2 match interrupt

bit 0: **TMR1IE:** TMR1 Overflow Interrupt Enable bit  
1 = Enables the TMR1 overflow interrupt  
0 = Disables the TMR1 overflow interrupt

# PIC16C7X

**FIGURE 4-11: PIE1 REGISTER PIC16C73/73A/74/74A/76/77 (ADDRESS 8Ch)**



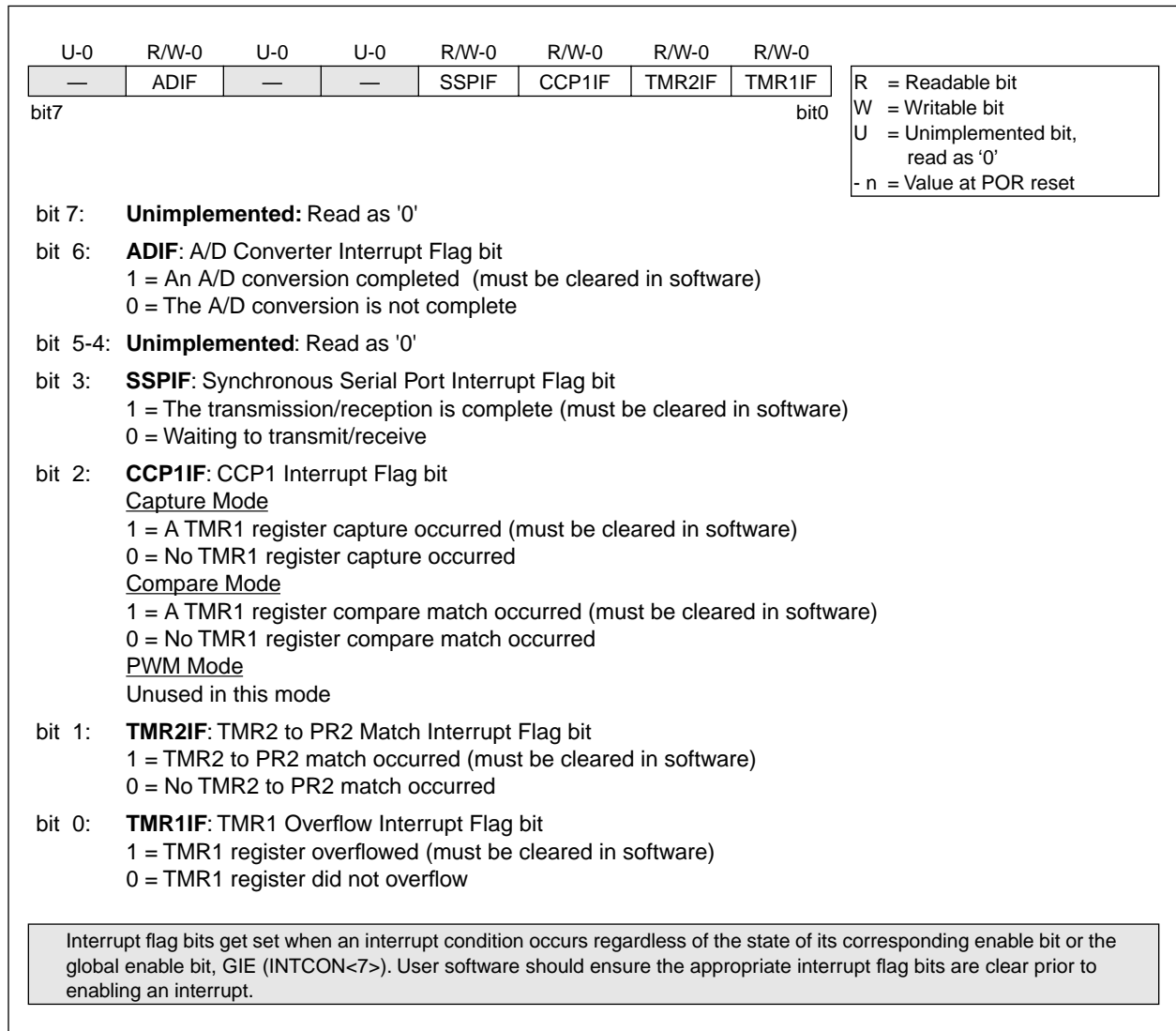
## 4.2.2.5 PIR1 REGISTER

Applicable Devices						
72	73	73A	74	74A	76	77

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

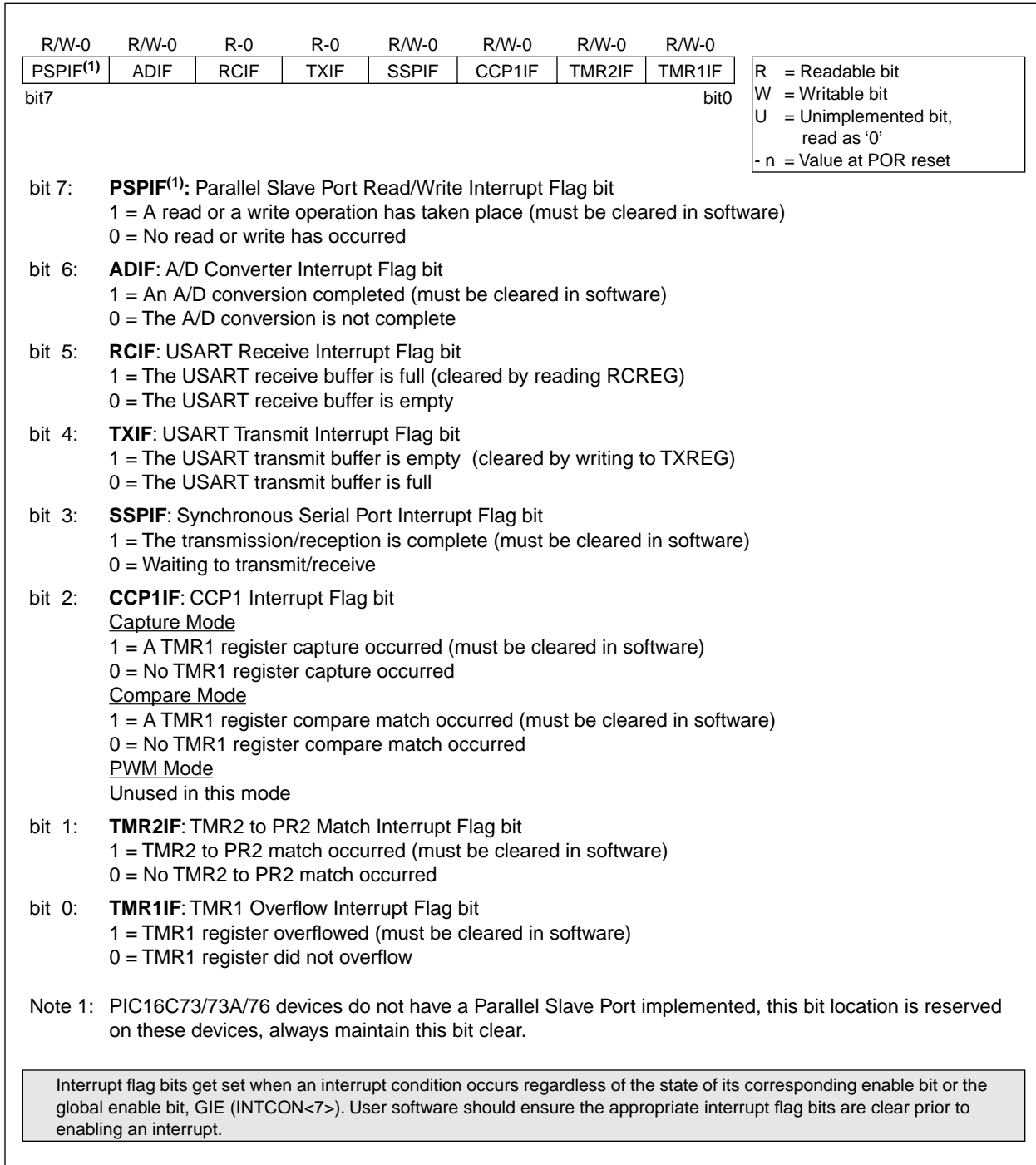
**Note:** Interrupt flag bits get set when an interrupt condition occurs regardless of the state of its corresponding enable bit or the global enable bit, GIE (INTCON<7>). User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

**FIGURE 4-12: PIR1 REGISTER PIC16C72 (ADDRESS 0Ch)**



# PIC16C7X

**FIGURE 4-13: PIR1 REGISTER PIC16C73/73A/74/74A/76/77 (ADDRESS 0Ch)**



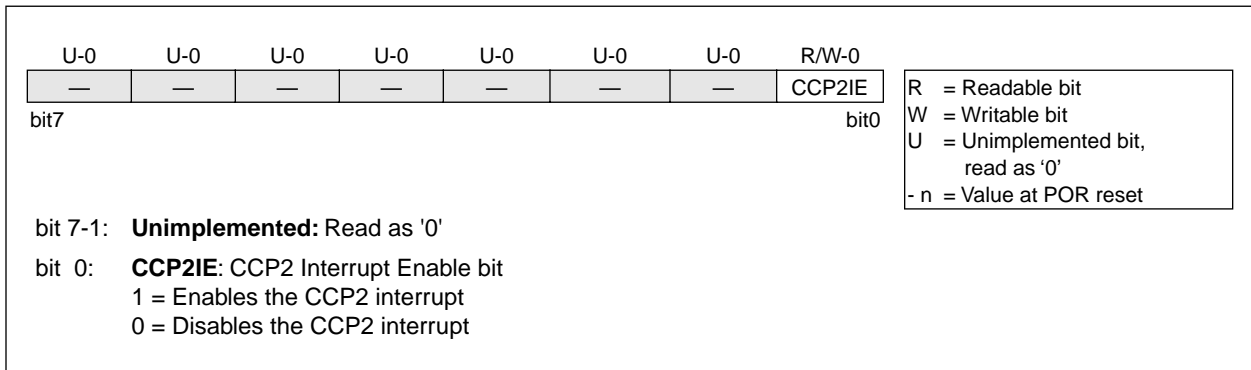
# PIC16C7X

## 4.2.2.6 PIE2 REGISTER

Applicable Devices						
72	73	73A	74	74A	76	77

This register contains the individual enable bit for the CCP2 peripheral interrupt.

**FIGURE 4-14: PIE2 REGISTER (ADDRESS 8Dh)**



# PIC16C7X

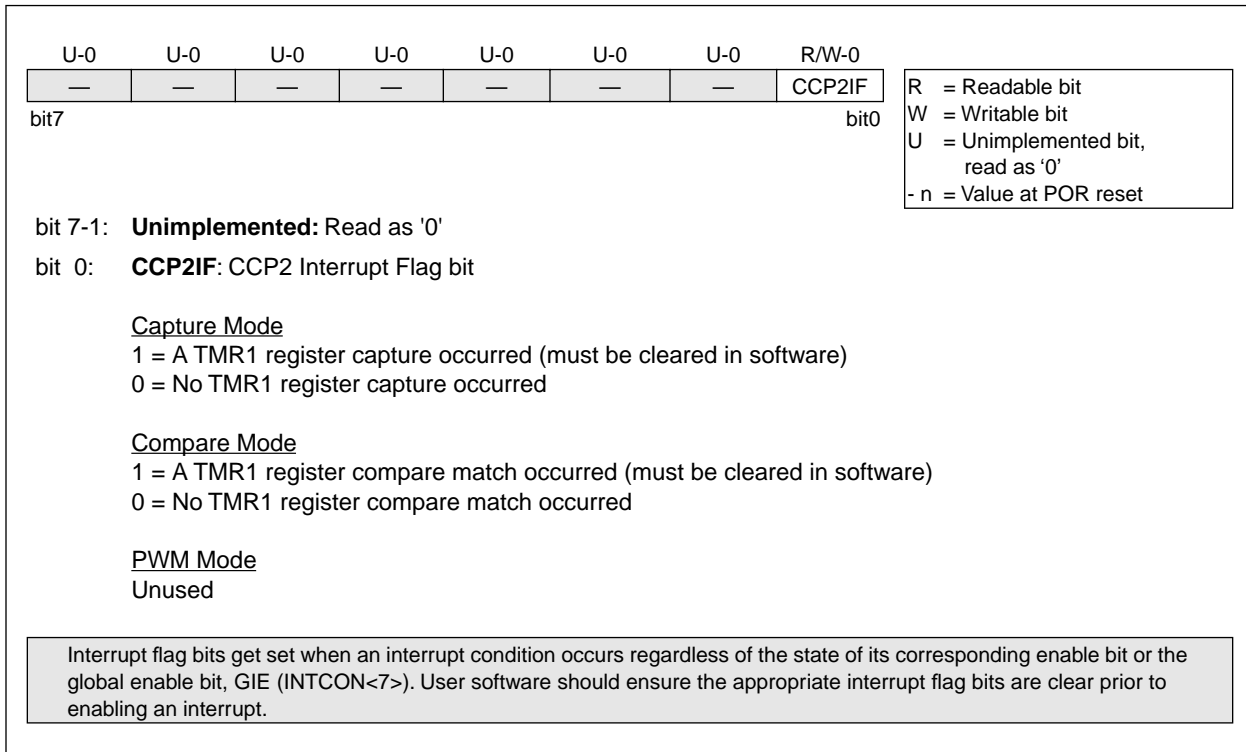
## 4.2.2.7 PIR2 REGISTER

Applicable Devices							
72	73	73A	74	74A	76	77	

This register contains the CCP2 interrupt flag bit.

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

**FIGURE 4-15: PIR2 REGISTER (ADDRESS 0Dh)**



# PIC16C7X

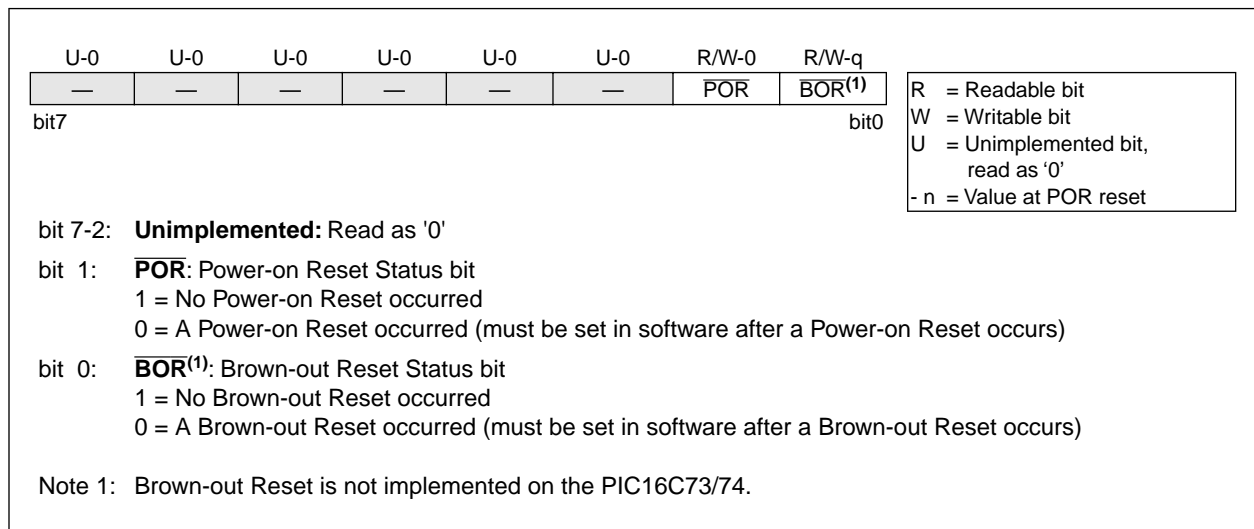
## 4.2.2.8 PCON REGISTER

Applicable Devices						
72	73	73A	74	74A	76	77

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

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

**FIGURE 4-16: PCON REGISTER (ADDRESS 8Eh)**



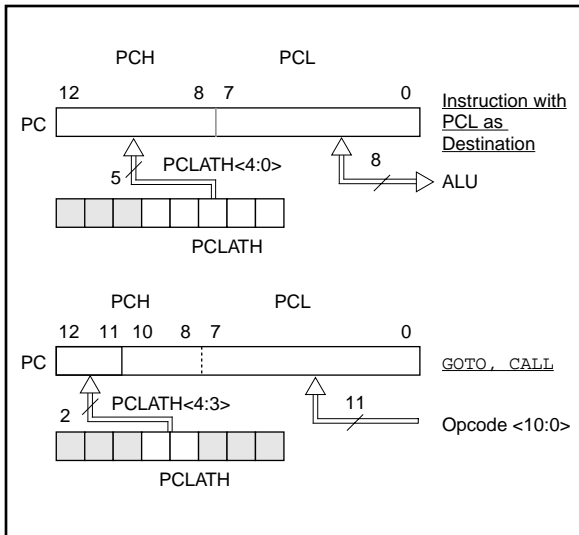
# PIC16C7X

## 4.3 PCL and PCLATH

Applicable Devices					
72	73	73A	74	74A	76/77

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 4-17 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 4-17: LOADING OF PC IN DIFFERENT SITUATIONS**



### 4.3.1 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).

### 4.3.2 STACK

The PIC16CXX 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.

**Note 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.

## 4.4 Program Memory Paging

Applicable Devices					
72	73	73A	74	74A	76/77

PIC16C7X 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 pushed onto 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:** PIC16C7X devices with 4K or less of program memory ignore paging bit PCLATH<4>. The use of PCLATH<4> as a general purpose read/write bit is not recommended since this may affect upward compatibility with future products.



Example 4-1 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 4-1: CALL OF A SUBROUTINE IN PAGE 1 FROM PAGE 0

```

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

### 4.5 Indirect Addressing, INDF and FSR Registers

Applicable Devices					
72	73	73A	74	74A	76/77

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 4-18.

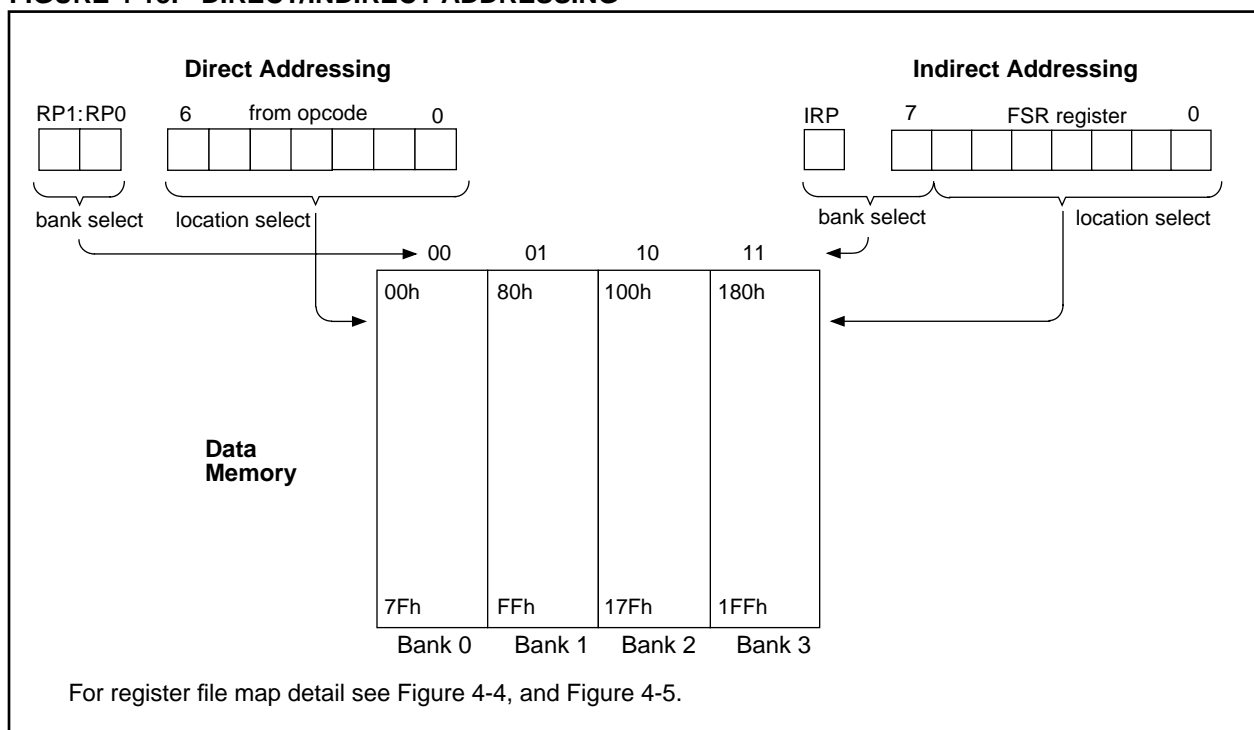
A simple program to clear RAM locations 20h-2Fh using indirect addressing is shown in Example 4-2.

### EXAMPLE 4-2: 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 4-18: DIRECT/INDIRECT ADDRESSING



# PIC16C7X

---

NOTES:

## 5.0 I/O PORTS

Applicable Devices					
72	73	73A	74	74A	76/77

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.

### 5.1 PORTA and TRISA Registers

Applicable Devices					
72	73	73A	74	74A	76/77

PORTA is a 6-bit latch.

The RA4/T0CKI pin is a Schmitt Trigger input and an open drain output. All other RA port pins have TTL input levels and full CMOS output drivers. All pins have data direction bits (TRIS registers) which can configure these pins as output or input.

Setting a TRISA register bit puts the corresponding output driver in a hi-impedance mode. Clearing a bit in the TRISA register puts the contents of the output latch on the selected pin(s).

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

Pin RA4 is multiplexed with the Timer0 module clock input to become the RA4/T0CKI pin.

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).

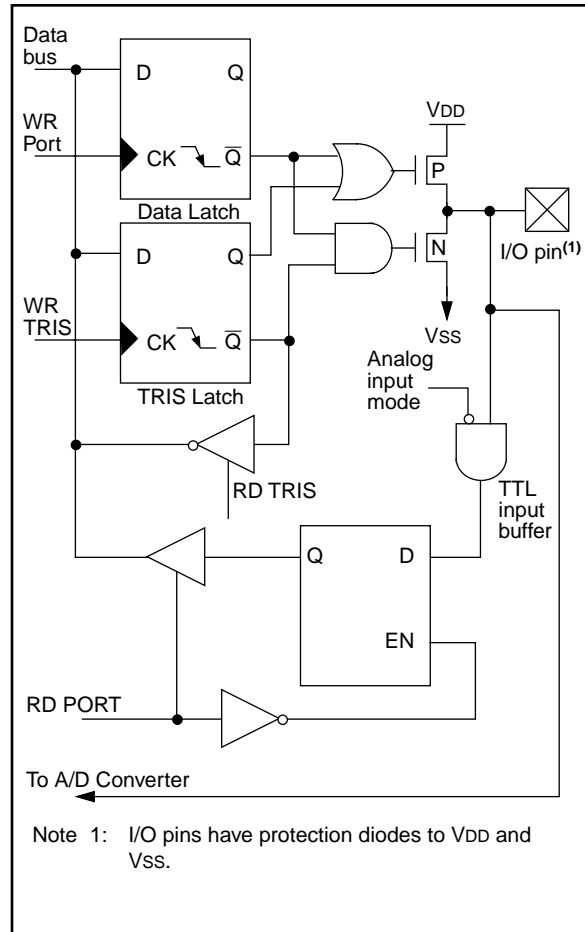
**Note:** On a Power-on Reset, these pins are configured as analog inputs and read as '0'.

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.

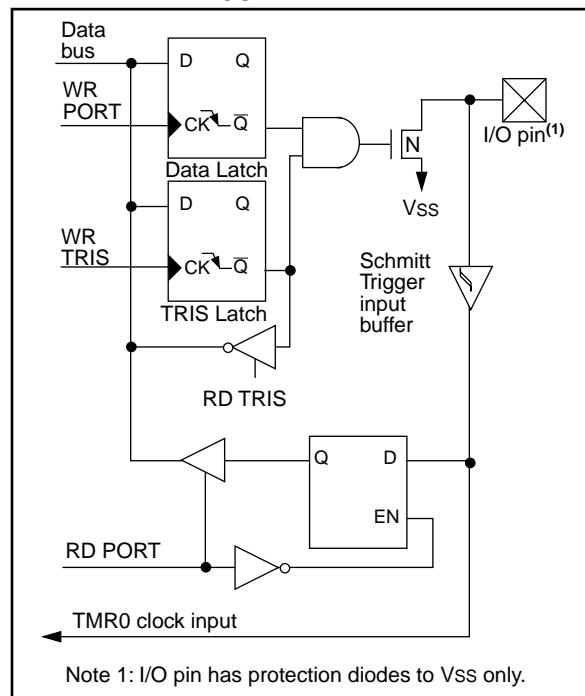
#### EXAMPLE 5-1: INITIALIZING PORTA

```
BCF STATUS, RP0 ;
BCF STATUS, RP1 ; PIC16C76/77 only
CLRF PORTA ; Initialize PORTA by
; clearing output
; data latches
BSF STATUS, RP0 ; Select Bank 1
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 5-1: BLOCK DIAGRAM OF RA3:RA0 AND RA5 PINS**



**FIGURE 5-2: BLOCK DIAGRAM OF RA4/T0CKI PIN**



# PIC16C7X

**TABLE 5-1: 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 5-2: SUMMARY OF REGISTERS ASSOCIATED WITH PORTA**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other resets
05h	PORTA	—	—	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.

## 5.2 PORTB and TRISB Registers

Applicable Devices				
72	73	73A	74	74A
76	77			

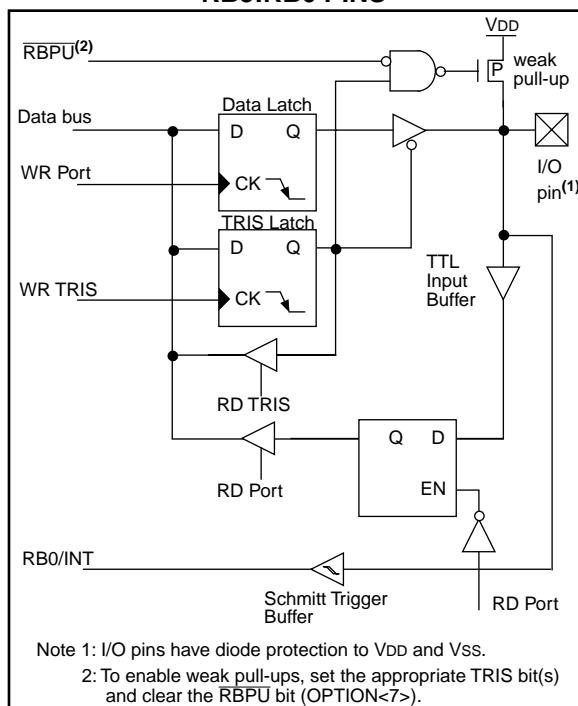
PORTB is an 8-bit wide bi-directional port. The corresponding data direction register is TRISB. Setting a bit in the TRISB register puts the corresponding output driver in a hi-impedance input mode. Clearing a bit in the TRISB register puts the contents of the output latch on the selected pin(s).

### EXAMPLE 5-2: INITIALIZING PORTB

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

Each of the PORTB pins has a weak internal pull-up. A single control bit can turn on all the pull-ups. This is performed by clearing bit  $\overline{\text{RBP}}\text{U}$  (OPTION<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 5-3: BLOCK DIAGRAM OF RB3:RB0 PINS**



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

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

- 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.

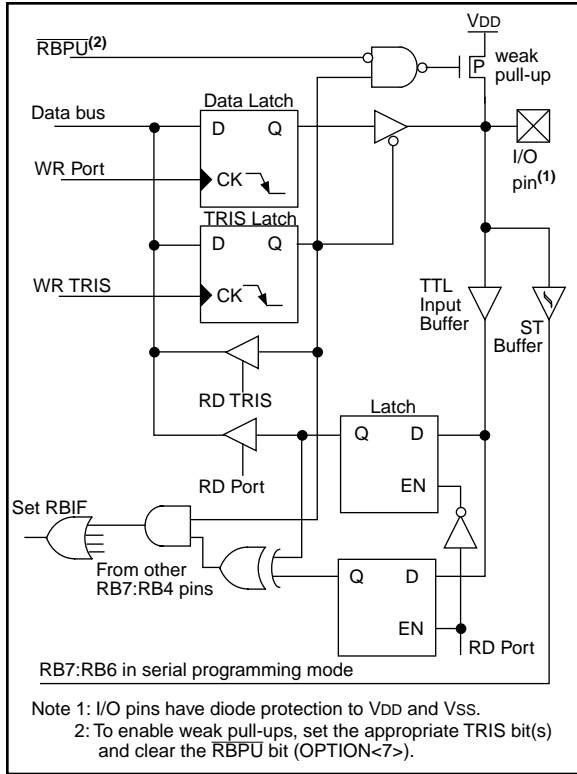
This interrupt on mismatch feature, together with software configurable pull-ups on these four pins allow easy interface to a keypad and make it possible for wake-up on key-depression. Refer to the Embedded Control Handbook, "Implementing Wake-Up on Key Stroke" (AN552).

**Note:** For the PIC16C73/74, if a change on the I/O pin should occur when the read operation is being executed (start of the Q2 cycle), then interrupt flag bit RBIF may not get set.

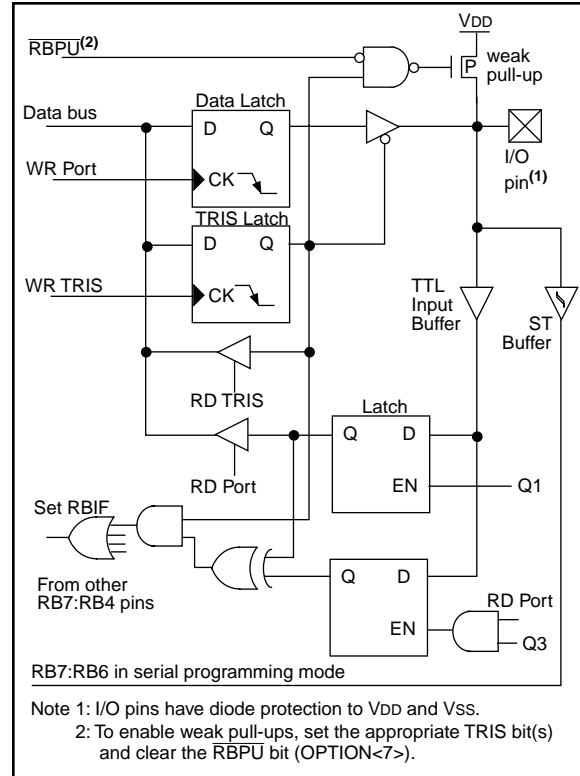
The interrupt on change feature is recommended for wake-up on key depression operation and operations where PORTB is only used for the interrupt on change feature. Polling of PORTB is not recommended while using the interrupt on change feature.

# PIC16C7X

**FIGURE 5-4: BLOCK DIAGRAM OF RB7:RB4 PINS (PIC16C73/74)**



**FIGURE 5-5: BLOCK DIAGRAM OF RB7:RB4 PINS (PIC16C72/73A/74A/76/77)**



**TABLE 5-3: PORTB FUNCTIONS**

Name	Bit#	Buffer	Function
RB0/INT	bit0	TTL/ST <sup>(1)</sup>	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 <sup>(2)</sup>	Input/output pin (with interrupt on change). Internal software programmable weak pull-up. Serial programming clock.
RB7	bit7	TTL/ST <sup>(2)</sup>	Input/output pin (with interrupt on change). Internal software programmable weak pull-up. Serial programming data.

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

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

2: This buffer is a Schmitt Trigger input when used in serial programming mode.

# PIC16C7X

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

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

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

# PIC16C7X

## 5.3 PORTC and TRISC Registers

Applicable Devices			
72	73	73A	74
74A	76	77	

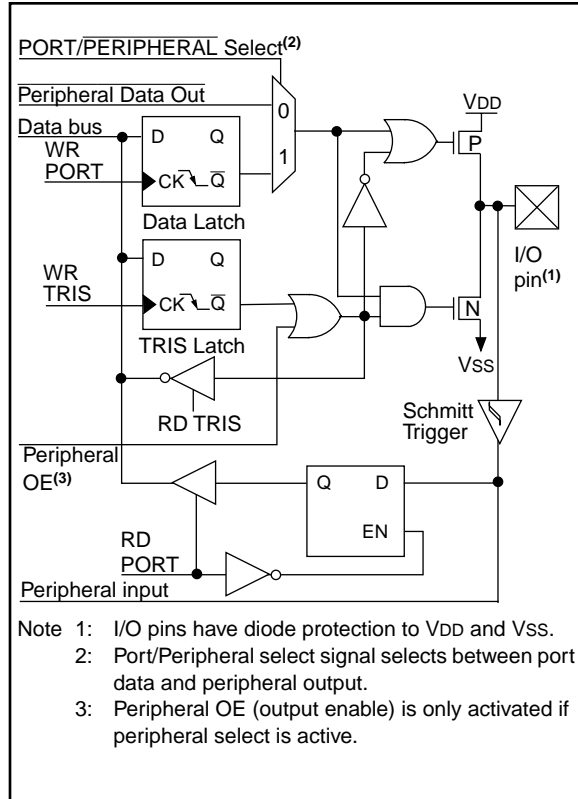
PORTC is an 8-bit bi-directional port. Each pin is individually configurable as an input or output through the TRISC register. PORTC is multiplexed with several peripheral functions (Table 5-5). PORTC pins have Schmitt Trigger input buffers.

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

### EXAMPLE 5-3: INITIALIZING PORTC

```
BCF STATUS, RP0 ; Select Bank 0
BCF STATUS, RP1 ; PIC16C76/77 only
CLRF PORTC ; Initialize PORTC by
; clearing output
; data latches
BSF STATUS, RP0 ; Select Bank 1
MOVLW 0xCF ; Value used to
; initialize data
; direction
MOVWF TRISC ; Set RC<3:0> as inputs
; RC<5:4> as outputs
; RC<7:6> as inputs
```

**FIGURE 5-6: PORTC BLOCK DIAGRAM (PERIPHERAL OUTPUT OVERRIDE)**



- Note 1: I/O pins have diode protection to VDD and VSS.
- Note 2: Port/Peripheral select signal selects between port data and peripheral output.
- Note 3: Peripheral OE (output enable) is only activated if peripheral select is active.

**TABLE 5-5: 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 <sup>(1)</sup>	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 <sup>2</sup> C modes.
RC4/SDI/SDA	bit4	ST	RC4 can also be the SPI Data In (SPI mode) or data I/O (I <sup>2</sup> C mode).
RC5/SDO	bit5	ST	Input/output port pin or Synchronous Serial Port data output
RC6/TX/CK <sup>(2)</sup>	bit6	ST	Input/output port pin or USART Asynchronous Transmit, or USART Synchronous Clock
RC7/RX/DT <sup>(2)</sup>	bit7	ST	Input/output port pin or USART Asynchronous Receive, or USART Synchronous Data

Legend: ST = Schmitt Trigger input

Note 1: The CCP2 multiplexed function is not enabled on the PIC16C72.

Note 2: The TX/CK and RX/DT multiplexed functions are not enabled on the PIC16C72.



# PIC16C7X

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

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

Legend: x = unknown, u = unchanged.

# PIC16C7X

## 5.4 PORTD and TRISD Registers

Applicable Devices						
72	73	73A	74	74A	76	77

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

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

FIGURE 5-7: PORTD BLOCK DIAGRAM (IN I/O PORT MODE)

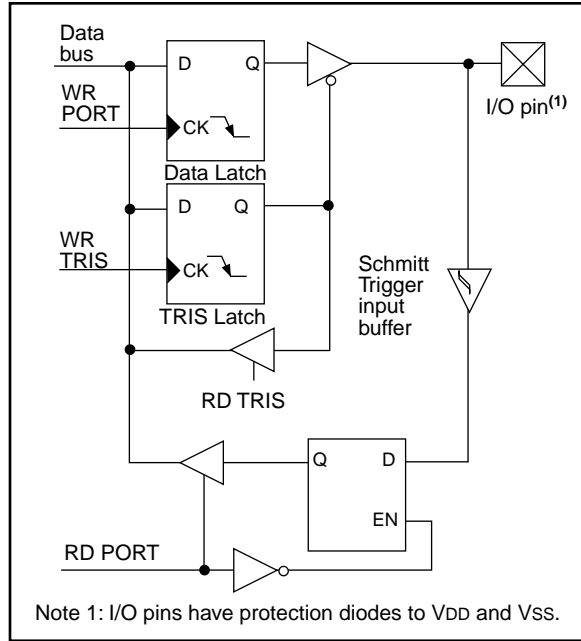


TABLE 5-7: PORTD FUNCTIONS

Name	Bit#	Buffer Type	Function
RD0/PSP0	bit0	ST/TTL <sup>(1)</sup>	Input/output port pin or parallel slave port bit0
RD1/PSP1	bit1	ST/TTL <sup>(1)</sup>	Input/output port pin or parallel slave port bit1
RD2/PSP2	bit2	ST/TTL <sup>(1)</sup>	Input/output port pin or parallel slave port bit2
RD3/PSP3	bit3	ST/TTL <sup>(1)</sup>	Input/output port pin or parallel slave port bit3
RD4/PSP4	bit4	ST/TTL <sup>(1)</sup>	Input/output port pin or parallel slave port bit4
RD5/PSP5	bit5	ST/TTL <sup>(1)</sup>	Input/output port pin or parallel slave port bit5
RD6/PSP6	bit6	ST/TTL <sup>(1)</sup>	Input/output port pin or parallel slave port bit6
RD7/PSP7	bit7	ST/TTL <sup>(1)</sup>	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 buffer when in Parallel Slave Port Mode.

TABLE 5-8: SUMMARY OF REGISTERS ASSOCIATED WITH PORTD

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other resets
08h	PORTD	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	xxxx xxxx	uuuu uuuu
88h	TRISD	PORTD Data Direction Register								1111 1111	1111 1111
89h	TRISE	IBF	OBF	IBOV	PSPMODE	—	PORTE Data Direction Bits			0000 -111	0000 -111

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

## 5.5 PORTE and TRISE Register

Applicable Devices					
72	73	73A	74	74A	76/77

PORTE has three pins RE0/RD/AN5, RE1/WR/AN6 and RE2/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 microprocessor 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) and that register ADCON1 is configured for digital I/O. In this mode the input buffers are TTL.

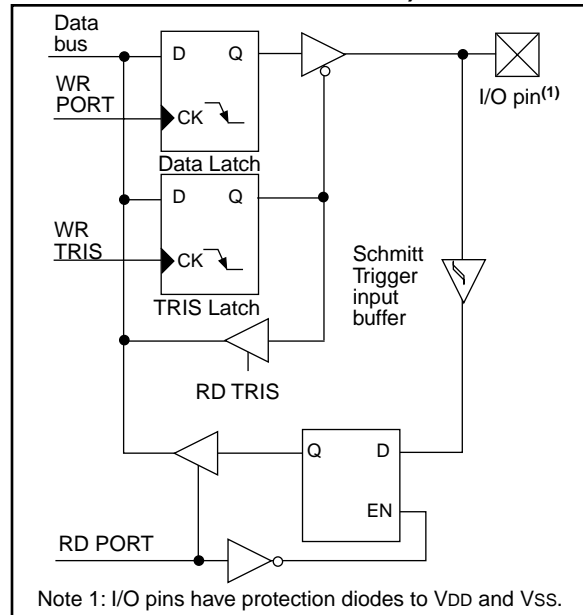
Figure 5-9 shows the TRISE register, which also controls the parallel slave port operation.

PORTE pins are multiplexed with analog inputs. The operation of these pins is selected by control bits in the ADCON1 register. 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.

**FIGURE 5-8: PORTE BLOCK DIAGRAM (IN I/O PORT MODE)**



**FIGURE 5-9: 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
bit7							bit0

R = Readable bit  
W = Writable bit  
U = Unimplemented bit, read as '0'  
-n = Value at POR reset

bit 7 : **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'

**PORTE Data Direction Bits**

bit 2 : **Bit2:** Direction Control bit for pin RE2/CS/AN7  
1 = Input  
0 = Output

bit 1 : **Bit1:** Direction Control bit for pin RE1/WR/AN6  
1 = Input  
0 = Output

bit 0 : **Bit0:** Direction Control bit for pin RE0/RD/AN5  
1 = Input  
0 = Output

# PIC16C7X

**TABLE 5-9: PORTE FUNCTIONS**

Name	Bit#	Buffer Type	Function
RE0/ $\overline{RD}$ /AN5	bit0	ST/TTL <sup>(1)</sup>	Input/output port pin or read control input in parallel slave port mode or analog input: $\overline{RD}$ 1 = Not a read operation 0 = Read operation. Reads PORTD register (if chip selected)
RE1/ $\overline{WR}$ /AN6	bit1	ST/TTL <sup>(1)</sup>	Input/output port pin or write control input in parallel slave port mode or analog input: $\overline{WR}$ 1 = Not a write operation 0 = Write operation. Writes PORTD register (if chip selected)
RE2/ $\overline{CS}$ /AN7	bit2	ST/TTL <sup>(1)</sup>	Input/output port pin or chip select control input in parallel slave port mode or analog input: $\overline{CS}$ 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 5-10: SUMMARY OF REGISTERS ASSOCIATED WITH PORTE**

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
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.

## 5.6 I/O Programming Considerations

Applicable Devices				
72	73	73A	74	74A
76	77			

### 5.6.1 BI-DIRECTIONAL I/O PORTS

Any instruction which writes, operates internally as a read followed by a write operation. The `BCF` and `BSF` instructions, for example, read the register into the CPU, execute the bit operation and write the result back to the register. Caution must be used when these instructions are applied to a port with both inputs and outputs defined. For example, a `BSF` operation on bit5 of `PORTB` will cause all eight bits of `PORTB` to be read into the CPU. Then the `BSF` operation takes place on bit5 and `PORTB` is written to the output latches. If another bit of `PORTB` is used as a bi-directional I/O pin (e.g., bit0) and it is defined as an input at this time, the input signal present on the pin itself would be read into the CPU and rewritten to the data latch of this particular pin, overwriting the previous content. As long as the pin stays in the input mode, no problem occurs. However, if bit0 is switched to an output, the content of the data latch may now be unknown.

Reading the port register, reads the values of the port pins. Writing to the port register writes the value to the port latch. When using read-modify-write instructions (ex. `BCF`, `BSF`, etc.) on a port, the value of the port pins is read, the desired operation is done to this value, and this value is then written to the port latch.

Example 5-4 shows the effect of two sequential read-modify-write instructions on an I/O port.

### EXAMPLE 5-4: READ-MODIFY-WRITE INSTRUCTIONS ON AN I/O PORT

```

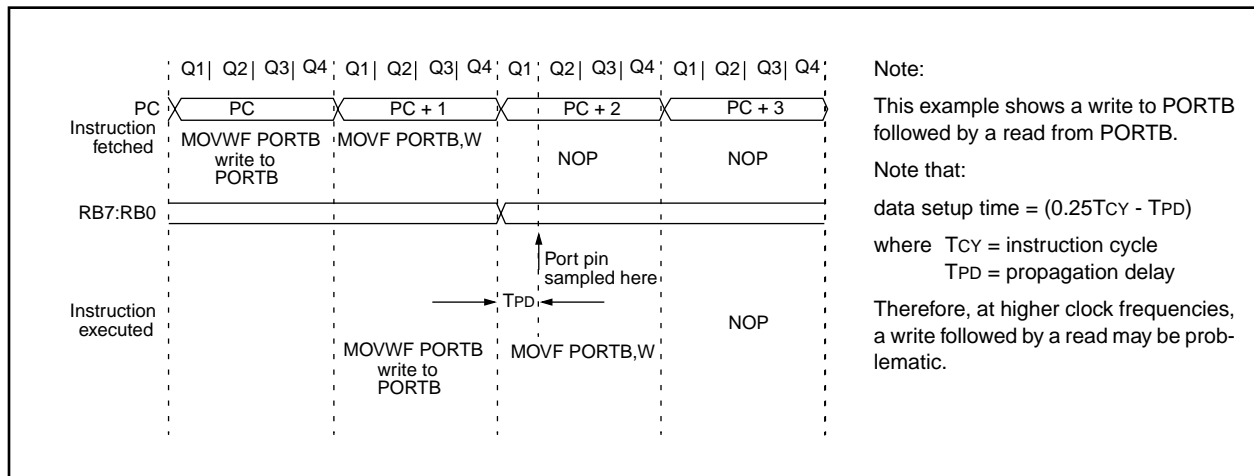
;Initial PORT settings: PORTB<7:4> Inputs
;                          PORTB<3:0> Outputs
;PORTB<7:6> have external pull-ups and are
;not connected to other circuitry
;
;                          PORT latch  PORT pins
;                          -----  -----
BCF PORTB, 7      ; 01pp pppp      11pp pppp
BCF PORTB, 6      ; 10pp pppp      11pp pppp
BSF STATUS, RP0  ;
BCF TRISB, 7      ; 10pp pppp      11pp pppp
BCF TRISB, 6      ; 10pp pppp      10pp pppp
;
;Note that the user may have expected the
;pin values to be 00pp ppp. The 2nd BCF
;caused RB7 to be latched as the pin value
;(high).
    
```

A pin actively outputting a Low or High should not be driven from external devices at the same time in order to change the level on this pin (“wired-or”, “wired-and”). The resulting high output currents may damage the chip.

### 5.6.2 SUCCESSIVE OPERATIONS ON I/O PORTS

The actual write to an I/O port happens at the end of an instruction cycle, whereas for reading, the data must be valid at the beginning of the instruction cycle (Figure 5-10). Therefore, care must be exercised if a write followed by a read operation is carried out on the same I/O port. The sequence of instructions should be such to allow the pin voltage to stabilize (load dependent) before the next instruction which causes that file to be read into the CPU is executed. Otherwise, the previous state of that pin may be read into the CPU rather than the new state. When in doubt, it is better to separate these instructions with a `NOP` or another instruction not accessing this I/O port.

**FIGURE 5-10: SUCCESSIVE I/O OPERATION**



# PIC16C7X

## 5.7 Parallel Slave Port

Applicable Devices			
72	73	73A	74
74A	76	77	

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 the external world through RD control input pin RE0/RD/AN5 and WR control input pin RE1/WR/AN6.

It 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/AN5 to be the RD input, RE1/WR/AN6 to be the WR input and RE2/CS/AN7 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 (set) and the A/D port configuration bits PCFG2:PCFG0 (ADCON1<2:0>) must be set, which will configure pins RE2:RE0 as digital I/O.

There are actually two 8-bit latches, one for data-out (from the PIC16/17) and one for data input. The user writes 8-bit data to PORTD data latch and reads data from the port pin latch (note that they have the same address). In this mode, the TRISD register is ignored, since the microprocessor is controlling the direction of data flow.

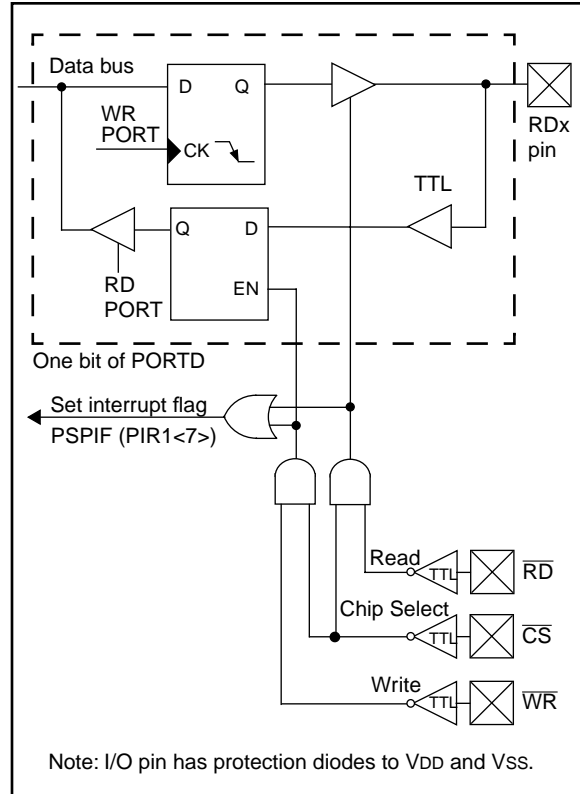
A write to the PSP occurs when both the CS and WR lines are first detected low. When either the CS or WR lines become high (level triggered), then the Input Buffer Full status flag bit IBF (TRISE<7>) is set on the Q4 clock cycle, following the next Q2 cycle, to signal the write is complete (Figure 5-12). The interrupt flag bit PSPIF (PIR1<7>) is also set on the same Q4 clock cycle. IBF can only be cleared by reading the PORTD input latch. The input Buffer Overflow status flag bit IBOV (TRISE<5>) is set if a second write to the Parallel Slave Port is attempted when the previous byte has not been read out of the buffer.

A read from the PSP occurs when both the CS and RD lines are first detected low. The Output Buffer Full status flag bit OBF (TRISE<6>) is cleared immediately (Figure 5-13) indicating that the PORTD latch is waiting to be read by the external bus. When either the CS or RD pin becomes high (level triggered), 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 data is written to PORTD by the user firmware.

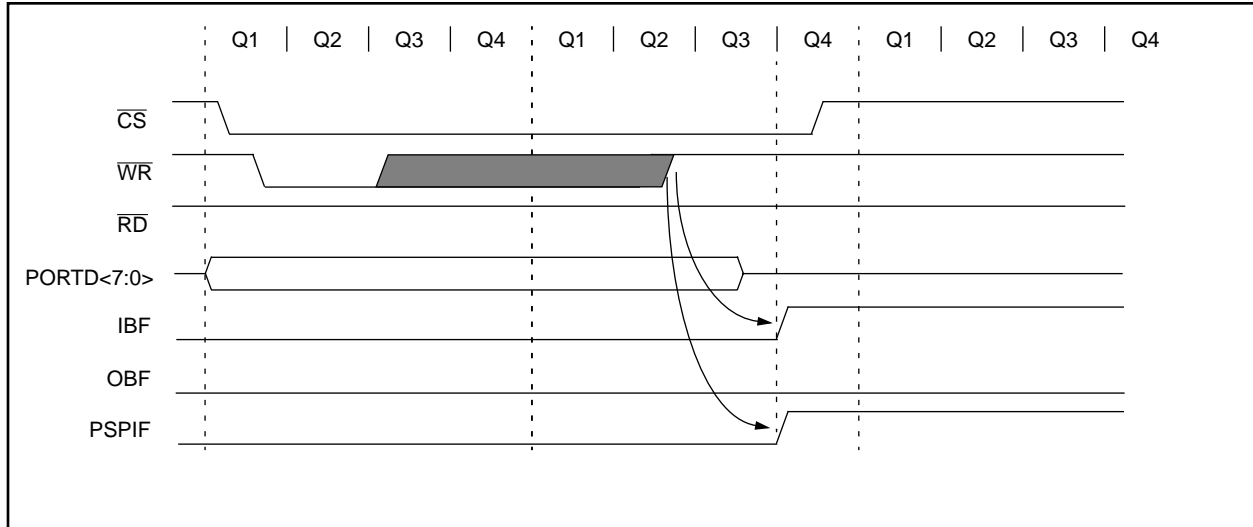
When not in Parallel Slave Port mode, the IBF and OBF bits are held clear. However, if flag bit IBOV was previously set, it must be cleared in firmware.

An interrupt is generated and latched into flag bit PSPIF when a read or write operation is completed. PSPIF must be cleared by the user in firmware and the interrupt can be disabled by clearing the interrupt enable bit PSPIE (PIE1<7>).

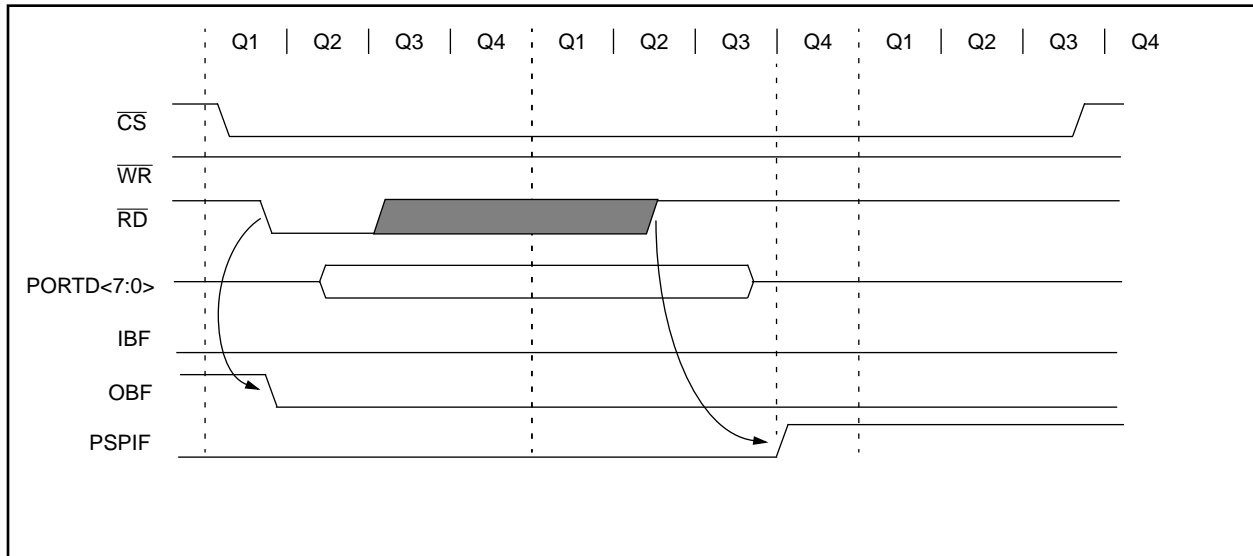
FIGURE 5-11: PORTD AND PORTE BLOCK DIAGRAM (PARALLEL SLAVE PORT)



**FIGURE 5-12: PARALLEL SLAVE PORT WRITE WAVEFORMS**



**FIGURE 5-13: PARALLEL SLAVE PORT READ WAVEFORMS**



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

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other resets
08h	PORTD	Port data latch when written: Port pins when read								xxxx xxxx	uuuu uuuu
09h	PORTE	—	—	—	—	—	RE2	RE1	RE0	---- -xxx	---- -uuu
89h	TRISE	IBF	OBF	IBOV	PSPMODE	—	PORTE Data Direction Bits			0000 -111	0000 -111
0Ch	PIR1	PSPIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
8Ch	PIE1	PSPIE	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.

# PIC16C7X

---

NOTES:



## 6.0 OVERVIEW OF TIMER MODULES

Applicable Devices						
72	73	73A	74	74A	76	77

The PIC16C72, PIC16C73/73A, PIC16C74/74A, PIC16C76/77 each have three timer modules.

Each module can generate an interrupt to indicate that an event has occurred (i.e. timer overflow). Each of these modules is explained in full detail in the following sections. The timer modules are:

- Timer0 Module (Section 7.0)
- Timer1 Module (Section 8.0)
- Timer2 Module (Section 9.0)

### 6.1 Timer0 Overview

Applicable Devices						
72	73	73A	74	74A	76	77

The Timer0 module is a simple 8-bit overflow counter. The clock source can be either the internal system clock ( $F_{osc}/4$ ) or an external clock. When the clock source is an external clock, the Timer0 module can be selected to increment on either the rising or falling edge.

The Timer0 module also has a programmable prescaler option. This prescaler can be assigned to either the Timer0 module or the Watchdog Timer. Bit PSA (OPTION<3>) assigns the prescaler, and bits PS2:PS0 (OPTION<2:0>) determine the prescaler value. Timer0 can increment at the following rates: 1:1 (when prescaler assigned to Watchdog timer), 1:2, 1:4, 1:8, 1:16, 1:32, 1:64, 1:128, and 1:256 (Timer0 only).

Synchronization of the external clock occurs after the prescaler. When the prescaler is used, the external clock frequency may be higher than the device's frequency. The maximum frequency is 50 MHz, given the high and low time requirements of the clock.

### 6.2 Timer1 Overview

Applicable Devices						
72	73	73A	74	74A	76	77

Timer1 is a 16-bit timer/counter. The clock source can be either the internal system clock ( $F_{osc}/4$ ), an external clock, or an external crystal. Timer1 can operate as either a timer or a counter. When operating as a counter (external clock source), the counter can either operate synchronized to the device or asynchronously to the device. Asynchronous operation allows Timer1 to operate during sleep, which is useful for applications that require a real-time clock as well as the power savings of SLEEP mode.

Timer1 also has a prescaler option which allows Timer1 to increment at the following rates: 1:1, 1:2, 1:4, and 1:8. Timer1 can be used in conjunction with the Capture/Compare/PWM module. When used with a

CCP module, Timer1 is the time-base for 16-bit Capture or the 16-bit Compare and must be synchronized to the device.

### 6.3 Timer2 Overview

Applicable Devices						
72	73	73A	74	74A	76	77

Timer2 is an 8-bit timer with a programmable prescaler and postscaler, as well as an 8-bit period register (PR2). Timer2 can be used with the CCP1 module (in PWM mode) as well as the Baud Rate Generator for the Synchronous Serial Port (SSP). The prescaler option allows Timer2 to increment at the following rates: 1:1, 1:4, 1:16.

The postscaler allows the TMR2 register to match the period register (PR2) a programmable number of times before generating an interrupt. The postscaler can be programmed from 1:1 to 1:16 (inclusive).

### 6.4 CCP Overview

Applicable Devices						
72	73	73A	74	74A	76	77

The CCP module(s) can operate in one of these three modes: 16-bit capture, 16-bit compare, or up to 10-bit Pulse Width Modulation (PWM).

Capture mode captures the 16-bit value of TMR1 into the CCPRxH:CCPRxL register pair. The capture event can be programmed for either the falling edge, rising edge, fourth rising edge, or the sixteenth rising edge of the CCPx pin.

Compare mode compares the TMR1H:TMR1L register pair to the CCPRxH:CCPRxL register pair. When a match occurs an interrupt can be generated, and the output pin CCPx can be forced to given state (High or Low), TMR1 can be reset (CCP1), or TMR1 reset and start A/D conversion (CCP2). This depends on the control bits CCPxM3:CCPxM0.

PWM mode compares the TMR2 register to a 10-bit duty cycle register (CCPRxH:CCPRxL<5:4>) as well as to an 8-bit period register (PR2). When the TMR2 register = Duty Cycle register, the CCPx pin will be forced low. When TMR2 = PR2, TMR2 is cleared to 00h, an interrupt can be generated, and the CCPx pin (if an output) will be forced high.

# PIC16C7X

---

NOTES:

## 7.0 TIMER0 MODULE

Applicable Devices					
72	73	73A	74	74A	76/77

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

Figure 7-1 is a simplified block diagram of the Timer0 module.

Timer mode is selected by clearing bit T0CS (OPTION<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 (Figure 7-2 and Figure 7-3). The user can work around this by writing an adjusted value to the TMR0 register.

Counter mode is selected by setting bit T0CS (OPTION<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<4>). Clearing bit T0SE selects the rising edge. Restrictions on the external clock input are discussed in detail in Section 7.2.

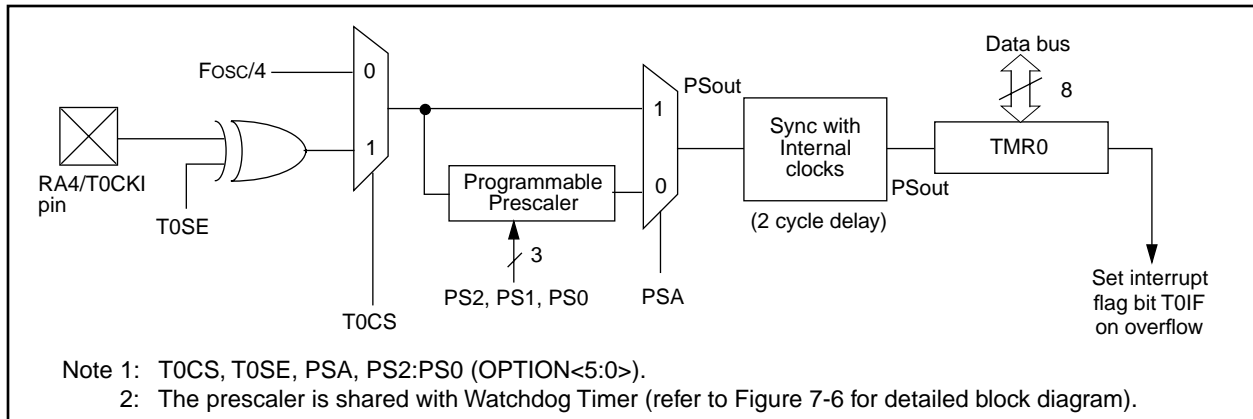
The prescaler is mutually exclusively shared between the Timer0 module and the Watchdog Timer. The prescaler assignment is controlled in software by control bit PSA (OPTION<3>). Clearing bit PSA will assign the prescaler to the Timer0 module. The prescaler is not readable or writable. When the prescaler is assigned to the Timer0 module, prescale values of 1:2, 1:4, ..., 1:256 are selectable. Section 7.3 details the operation of the prescaler.

### 7.1 Timer0 Interrupt

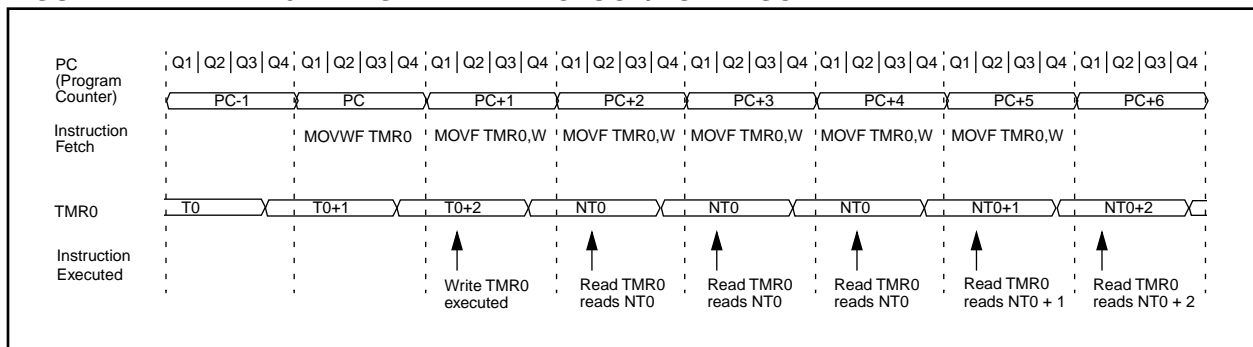
Applicable Devices					
72	73	73A	74	74A	76/77

The TMR0 interrupt is generated when the TMR0 register overflows from FFh to 00h. This overflow sets bit T0IF (INTCON<2>). The interrupt can be masked by clearing bit T0IE (INTCON<5>). Bit T0IF must be cleared in software by the Timer0 module interrupt service routine before re-enabling this interrupt. The TMR0 interrupt cannot awaken the processor from SLEEP since the timer is shut off during SLEEP. See Figure 7-4 for Timer0 interrupt timing.

**FIGURE 7-1: TIMER0 BLOCK DIAGRAM**

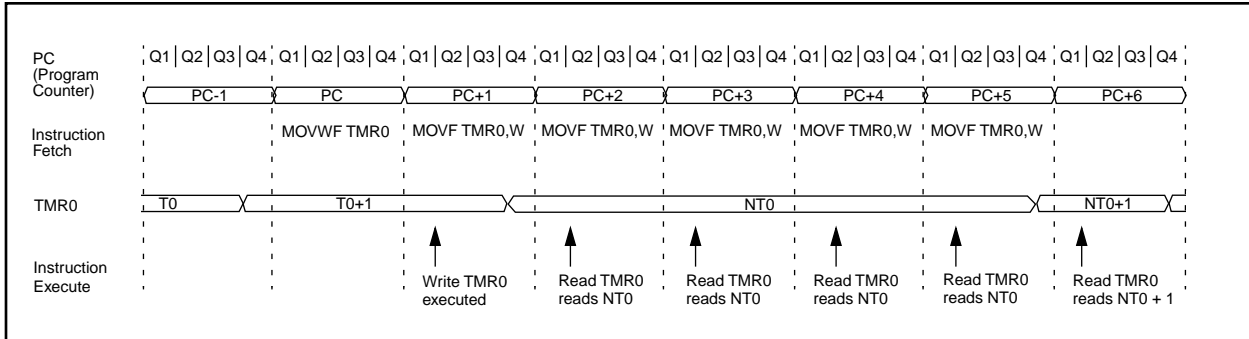


**FIGURE 7-2: TIMER0 TIMING: INTERNAL CLOCK/NO PRESCALE**

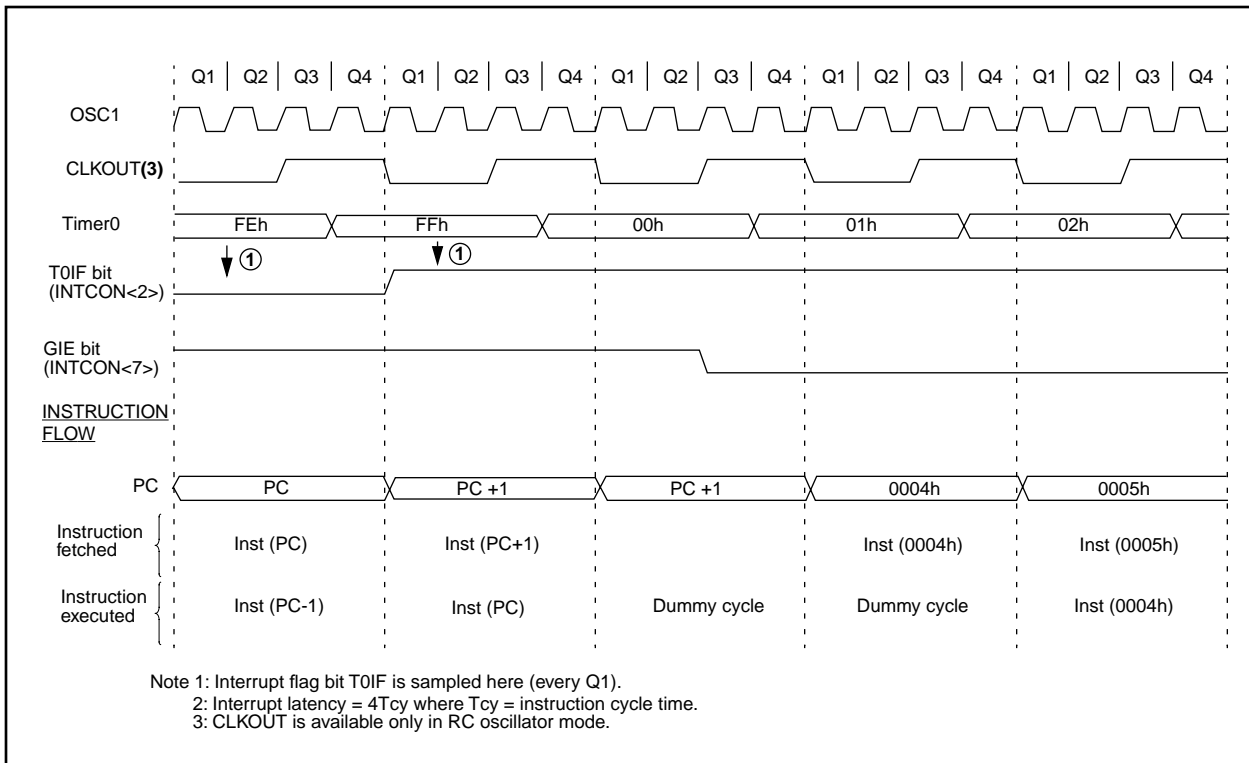


# PIC16C7X

**FIGURE 7-3: TIMER0 TIMING: INTERNAL CLOCK/PRESCALE 1:2**



**FIGURE 7-4: TIMER0 INTERRUPT TIMING**



## 7.2 Using Timer0 with an External Clock

Applicable Devices					
72	73	73A	74	74A	76/77

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

### 7.2.1 EXTERNAL CLOCK SYNCHRONIZATION

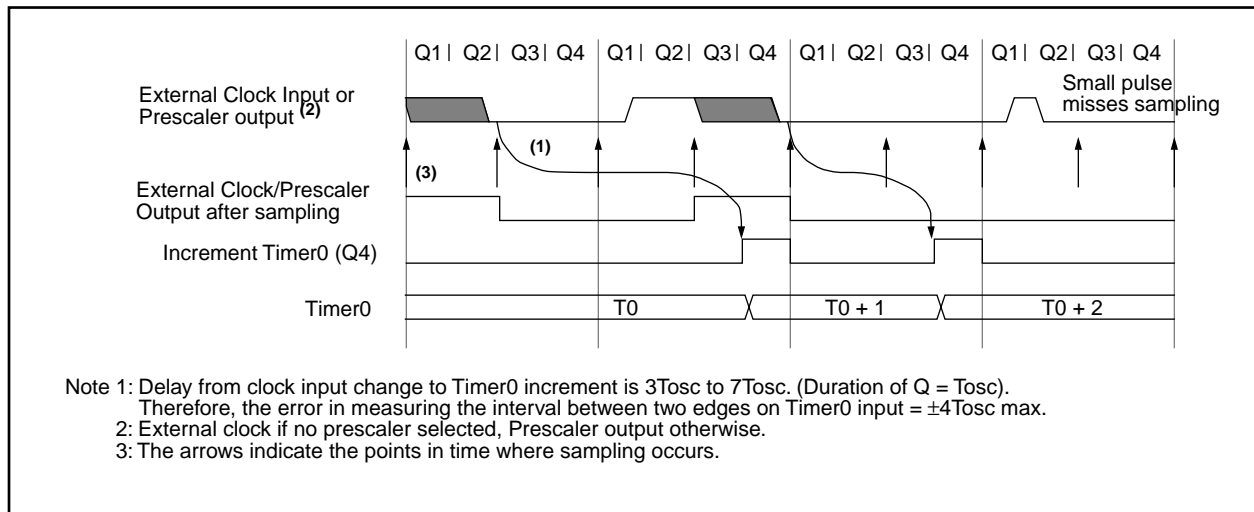
When no prescaler is used, the external clock input is the same as the prescaler output. The synchronization of T0CKI with the internal phase clocks is accomplished by sampling the prescaler output on the Q2 and Q4 cycles of the internal phase clocks (Figure 7-5). Therefore, it is necessary for T0CKI to be high for at least  $2T_{OSC}$  (and a small RC delay of 20 ns) and low for at least  $2T_{OSC}$  (and a small RC delay of 20 ns). Refer to the electrical specification of the desired device.

When a prescaler is used, the external clock input is divided by the asynchronous ripple-counter type prescaler so that the prescaler output is symmetrical. For the external clock to meet the sampling requirement, the ripple-counter must be taken into account. Therefore, it is necessary for T0CKI to have a period of at least  $4T_{OSC}$  (and a small RC delay of 40 ns) divided by the prescaler value. The only requirement on T0CKI high and low time is that they do not violate the minimum pulse width requirement of 10 ns. Refer to parameters 40, 41 and 42 in the electrical specification of the desired device.

### 7.2.2 TMR0 INCREMENT DELAY

Since the prescaler output is synchronized with the internal clocks, there is a small delay from the time the external clock edge occurs to the time the Timer0 module is actually incremented. Figure 7-5 shows the delay from the external clock edge to the timer incrementing.

**FIGURE 7-5: TIMER0 TIMING WITH EXTERNAL CLOCK**



# PIC16C7X

## 7.3 Prescaler

Applicable Devices					
72	73	73A	74	74A	76/77

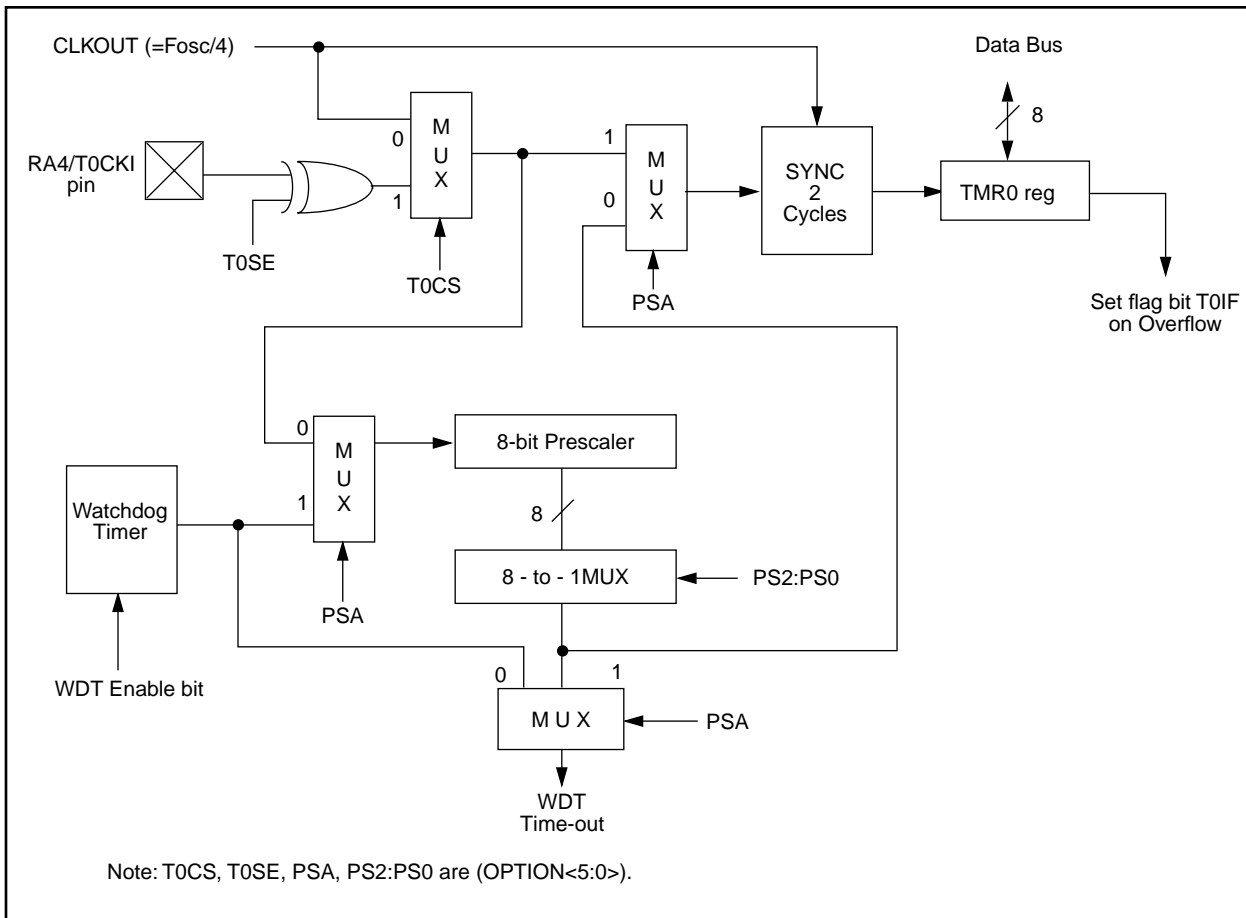
An 8-bit counter is available as a prescaler for the Timer0 module, or as a postscaler for the Watchdog Timer, respectively (Figure 7-6). For simplicity, this counter is being referred to as “prescaler” throughout this data sheet. Note that there is only one prescaler available which is mutually exclusively shared between the Timer0 module and the Watchdog Timer. Thus, a prescaler assignment for the Timer0 module means that there is no prescaler for the Watchdog Timer, and vice-versa.

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

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

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

**FIGURE 7-6: BLOCK DIAGRAM OF THE TIMER0/WDT PRESCALER**



## 7.3.1 SWITCHING PRESCALER ASSIGNMENT

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

**Note:** To avoid an unintended device RESET, the following instruction sequence (shown in Example 7-1) must be executed when changing the prescaler assignment from Timer0 to the WDT. This sequence must be followed even if the WDT is disabled.

### EXAMPLE 7-1: CHANGING PRESCALER (TIMER0→WDT)

```

1) BSF    STATUS, RP0    ;Bank 1
2) MOVLW  b'xx0x0xxx'   ;Select clock source and prescale value of
3) MOVWF  OPTION_REG    ;other than 1:1
4) BCF    STATUS, RP0    ;Bank 0
5) CLRF   TMR0          ;Clear TMR0 and prescaler
6) BSF    STATUS, RP1    ;Bank 1
7) MOVLW  b'xxxx1xxx'   ;Select WDT, do not change prescale value
8) MOVWF  OPTION_REG    ;
9) CLRWD  TMR0          ;Clears WDT and prescaler
10) MOVLW b'xxxx1xxx'   ;Select new prescale value and WDT
11) MOVWF  OPTION_REG    ;
12) BCF    STATUS, RP0    ;Bank 0

```

Lines 2 and 3 do NOT have to be included if the final desired prescale value is other than 1:1. If 1:1 is final desired value, then a temporary prescale value is set in lines 2 and 3 and the final prescale value will be set in lines 10 and 11.

To change prescaler from the WDT to the Timer0 module use the sequence shown in Example 7-2.

### EXAMPLE 7-2: CHANGING PRESCALER (WDT→TIMER0)

```

CLRWD  TMR0          ;Clear WDT and prescaler
BSF    STATUS, RP0    ;Bank 1
MOVLW  b'xxx0xxx'   ;Select TMR0, new prescale value and
MOVWF  OPTION_REG    ;clock source
BCF    STATUS, RP0    ;Bank 0

```

**TABLE 7-1: 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's register								xxxx xxxx	uuuu uuuu
0Bh,8Bh, 10Bh,18Bh	INTCON	GIE	PEIE	TOIE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
81h,181h	OPTION	RBPU	INTEDG	TOCS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
85h	TRISA	—	—	PORTA Data Direction Register						--11 1111	--11 1111

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

# PIC16C7X

---

NOTES:



## 8.0 TIMER1 MODULE

Applicable Devices					
72	73	73A	74	74A	76/77

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.

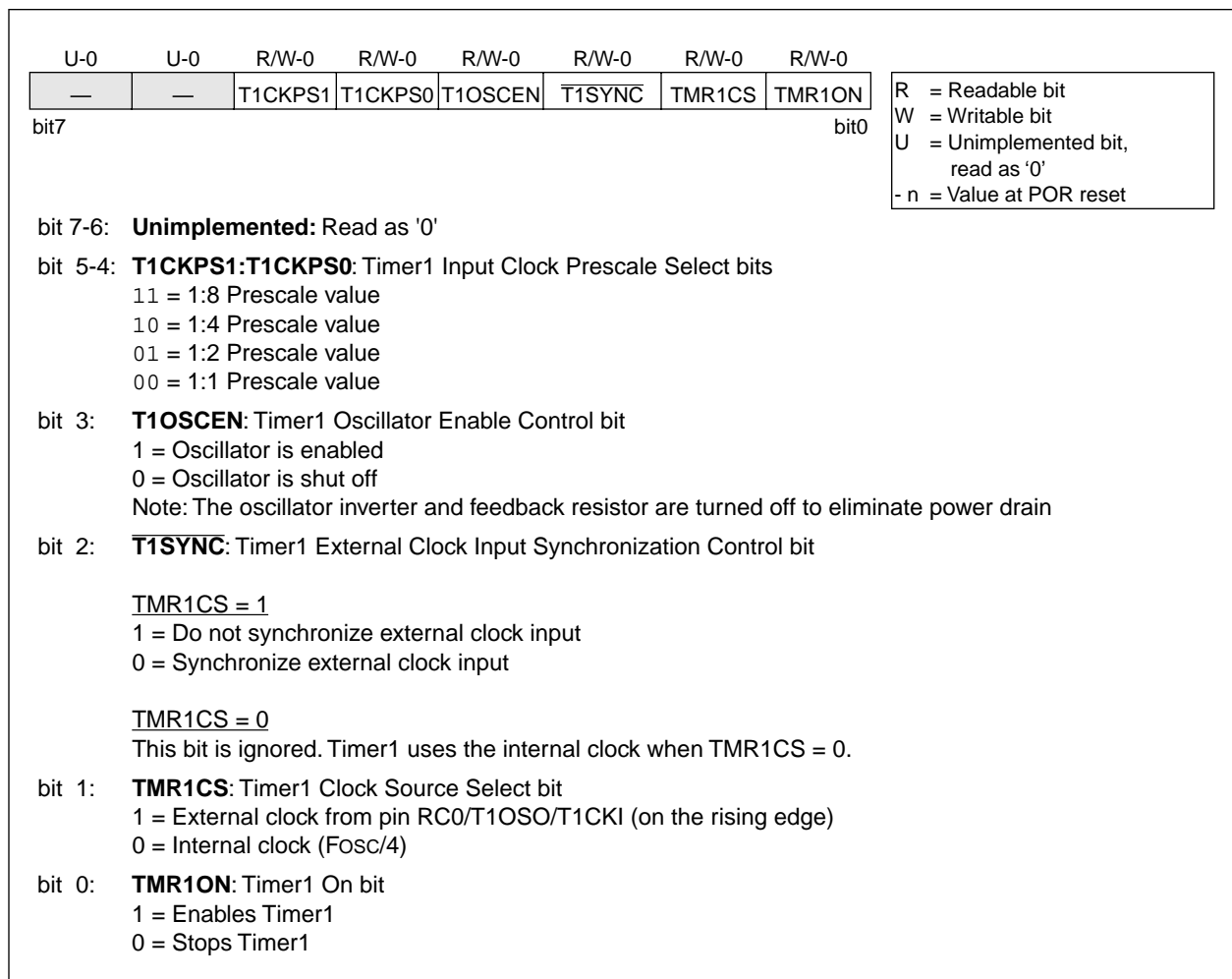
Timer1 can be enabled/disabled by setting/clearing control bit TMR1ON (T1CON<0>).

Timer1 also has an internal "reset input". This reset can be generated by either of the two CCP modules (Section 10.0). Figure 8-1 shows the Timer1 control register.

For the PIC16C72/73A/74A/76/77, 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.

For the PIC16C73/74, when the Timer1 oscillator is enabled (T1OSCEN is set), RC1/T1OSI/CCP2 pin becomes an input, however the RC0/T1OSO/T1CKI pin will have to be configured as an input by setting the TRISC<0> bit.

**FIGURE 8-1: T1CON:TIMER1 CONTROL REGISTER (ADDRESS 10h)**



# PIC16C7X

## 8.1 Timer1 Operation in Timer Mode

Applicable Devices					
72	73	73A	74	74A	76/77

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

## 8.2 Timer1 Operation in Synchronized Counter Mode

Applicable Devices					
72	73	73A	74	74A	76/77

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 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.

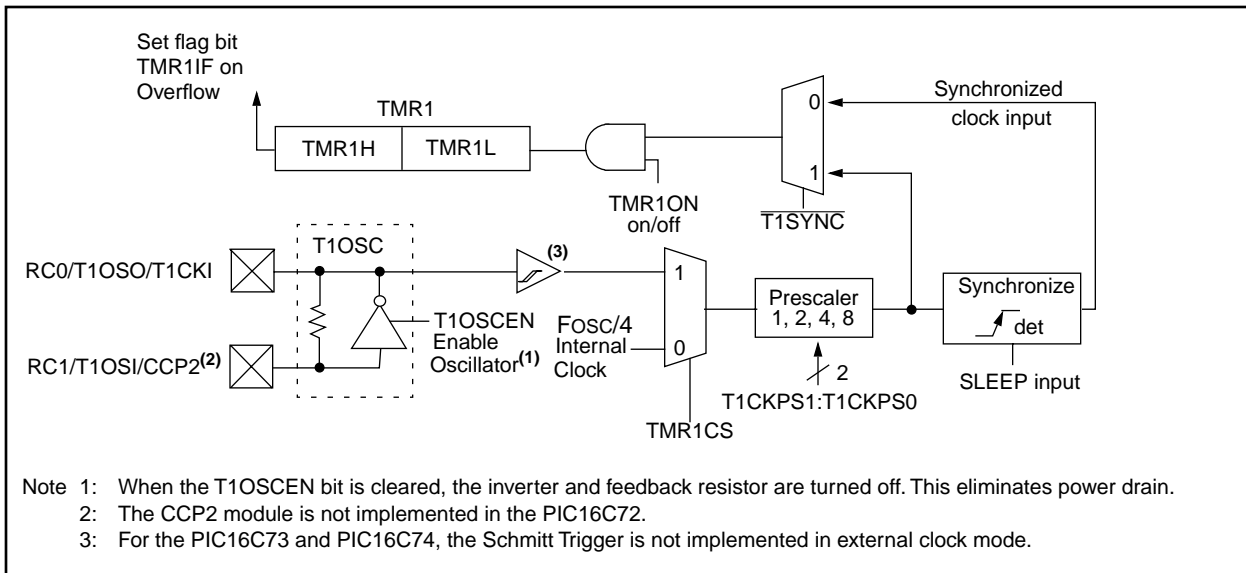
## 8.2.1 EXTERNAL CLOCK INPUT TIMING FOR SYNCHRONIZED COUNTER MODE

When an external clock input is used for Timer1 in synchronized counter mode, it must meet certain requirements. The external clock requirement is due to internal phase clock (Tosc) synchronization. Also, there is a delay in the actual incrementing of TMR1 after synchronization.

When the prescaler is 1:1, the external clock input is the same as the prescaler output. The synchronization of T1CKI 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 T1CKI to be high for at least  $2T_{osc}$  (and a small RC delay of 20 ns) and low for at least  $2T_{osc}$  (and a small RC delay of 20 ns). Refer to the appropriate electrical specifications, parameters 45, 46, and 47.

When a prescaler other than 1:1 is used, the external clock input is divided by the asynchronous ripple-counter type prescaler so that the prescaler output is symmetrical. In order for the external clock to meet the sampling requirement, the ripple-counter must be taken into account. Therefore, it is necessary for T1CKI to have a period of at least  $4T_{osc}$  (and a small RC delay of 40 ns) divided by the prescaler value. The only requirement on T1CKI high and low time is that they do not violate the minimum pulse width requirements of 10 ns). Refer to the appropriate electrical specifications, parameters 40, 42, 45, 46, and 47.

FIGURE 8-2: TIMER1 BLOCK DIAGRAM



## 8.3 Timer1 Operation in Asynchronous Counter Mode

Applicable Devices			
72	73	73A	74
74A	76	77	

If control bit  $\overline{T1SYNC}$  (T1CON<2>) is set, the external clock input is not synchronized. The timer continues to increment asynchronous to the internal phase clocks. The timer will continue to run during SLEEP and can generate an interrupt on overflow which will wake-up the processor. However, special precautions in software are needed to read/write the timer (Section 8.3.2).

In asynchronous counter mode, Timer1 can not be used as a time-base for capture or compare operations.

### 8.3.1 EXTERNAL CLOCK INPUT TIMING WITH UNSYNCHRONIZED CLOCK

If control bit  $\overline{T1SYNC}$  is set, the timer will increment completely asynchronously. The input clock must meet certain minimum high time and low time requirements. Refer to the appropriate Electrical Specifications Section, timing parameters 45, 46, and 47.

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

Reading TMR1H or TMR1L while the timer is running, from an external asynchronous clock, will guarantee 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. Example 8-1 is an example routine to read the 16-bit timer value. This is useful if the timer cannot be stopped.

## EXAMPLE 8-1: READING A 16-BIT FREE-RUNNING TIMER

```

; All interrupts are disabled
MOVVF TMR1H, W ;Read high byte
MOVWF TMPH ;
MOVVF TMR1L, W ;Read low byte
MOVWF TMPL ;
MOVVF 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.
;
MOVVF TMR1H, W ;Read high byte
MOVWF TMPH ;
MOVVF TMR1L, W ;Read low byte
MOVWF TMPL ;
; Re-enable the Interrupt (if required)
CONTINUE ;Continue with your code

```

## 8.4 Timer1 Oscillator

Applicable Devices			
72	73	73A	74
74A	76	77	

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 a 32 kHz crystal. Table 8-1 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.

**TABLE 8-1: CAPACITOR SELECTION FOR THE TIMER1 OSCILLATOR**

Osc Type	Freq	C1	C2
LP	32 kHz	33 pF	33 pF
	100 kHz	15 pF	15 pF
	200 kHz	15 pF	15 pF

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

#### Crystals Tested:

32.768 kHz	Epson C-001R32.768K-A	± 20 PPM
100 kHz	Epson C-2 100.00 KC-P	± 20 PPM
200 kHz	STD XTL 200.000 kHz	± 20 PPM

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.

# PIC16C7X

## 8.5 Resetting Timer1 using a CCP Trigger Output

Applicable Devices						
72	73	73A	74	74A	76	77

The CCP2 module is not implemented on the PIC16C72 device.

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 registers pair effectively becomes the period register for Timer1.

## 8.6 Resetting of Timer1 Register Pair (TMR1H, TMR1L)

Applicable Devices						
72	73	73A	74	74A	76	77

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.

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.

## 8.7 Timer1 Prescaler

Applicable Devices						
72	73	73A	74	74A	76	77

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

**TABLE 8-2: 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	TOIE	INTE	RBIE	TOIF	INTF	RBFIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF <sup>(1,2)</sup>	ADIF	RCIF <sup>(2)</sup>	TXIF <sup>(2)</sup>	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
8Ch	PIE1	PSPIE <sup>(1,2)</sup>	ADIE	RCIE <sup>(2)</sup>	TXIE <sup>(2)</sup>	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 PIC16C73/73A/76, always maintain these bits clear.

Note 2: The PIC16C72 does not have a Parallel Slave Port or a USART, these bits are unimplemented, read as '0'.

## 9.0 TIMER2 MODULE

Applicable Devices					
72	73	73A	74	74A	76/77

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

Figure 9-2 shows the Timer2 control register.

## 9.1 Timer2 Prescaler and Postscaler

Applicable Devices					
72	73	73A	74	74A	76/77

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

- a write to the TMR2 register
- a write to the T2CON register
- any device reset (Power-on Reset,  $\overline{MCLR}$  reset, Watchdog Timer reset, or Brown-out Reset)

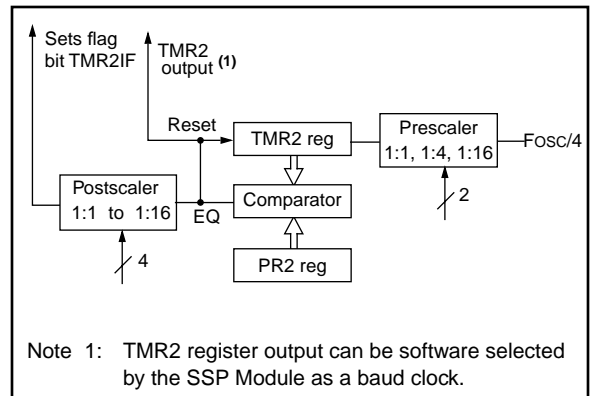
TMR2 is not cleared when T2CON is written.

## 9.2 Output of TMR2

Applicable Devices					
72	73	73A	74	74A	76/77

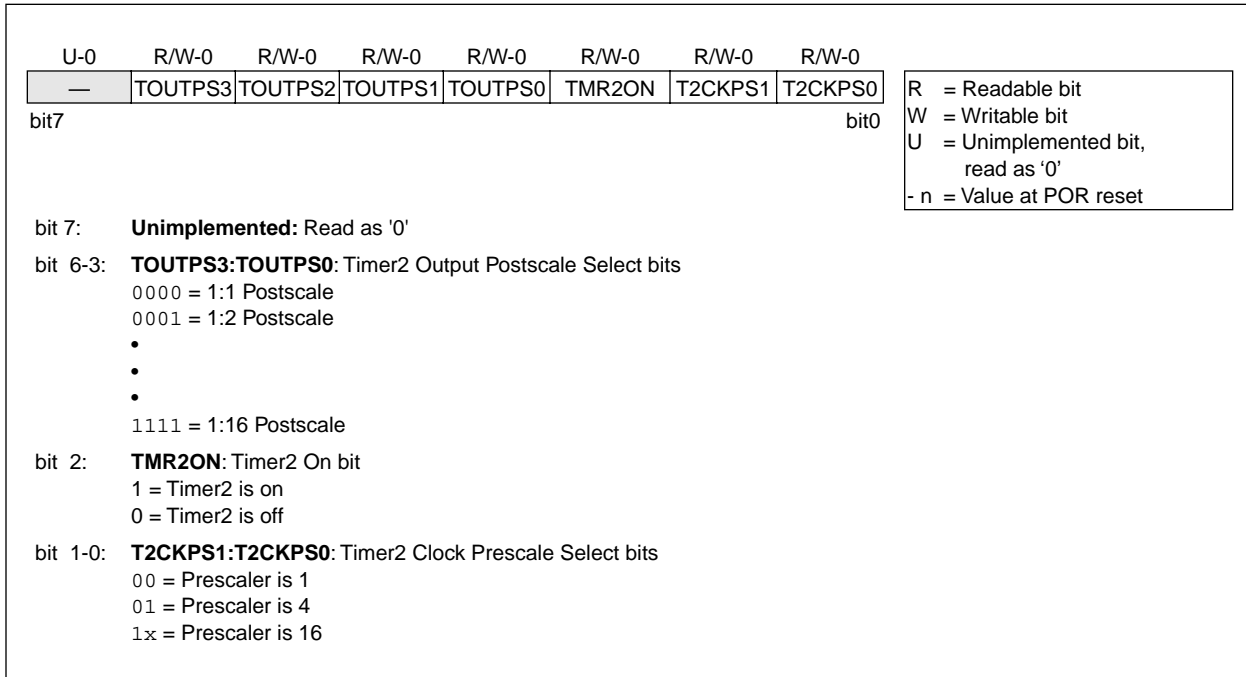
The output of TMR2 (before the postscaler) is fed to the Synchronous Serial Port module which optionally uses it to generate shift clock.

**FIGURE 9-1: TIMER2 BLOCK DIAGRAM**



# PIC16C7X

**FIGURE 9-2: T2CON: TIMER2 CONTROL REGISTER (ADDRESS 12h)**



**TABLE 9-1: 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	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF <sup>(1,2)</sup>	ADIF	RCIF <sup>(2)</sup>	TXIF <sup>(2)</sup>	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
8Ch	PIE1	PSPIE <sup>(1,2)</sup>	ADIE	RCIE <sup>(2)</sup>	TXIE <sup>(2)</sup>	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
11h	TMR2	Timer2 module's 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 PIC16C73/73A/76, always maintain these bits clear.

Note 2: The PIC16C72 does not have a Parallel Slave Port or a USART, these bits are unimplemented, read as '0'.

## 10.0 CAPTURE/COMPARE/PWM MODULE(S)

Applicable Devices							
72	73	73A	74	74A	76	77	CCP1
72	73	73A	74	74A	76	77	CCP2

Each CCP (Capture/Compare/PWM) module contains a 16-bit register which can operate as a 16-bit capture register, as a 16-bit compare register or as a PWM master/slave Duty Cycle register. Both the CCP1 and CCP2 modules are identical in operation, with the exception of the operation of the special event trigger. Table 10-1 and Table 10-2 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.

### 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. All are readable and writable.

### CCP2 module:

Capture/Compare/PWM Register2 (CCPR2) is comprised of two 8-bit registers: CCPR2L (low byte) and CCPR2H (high byte). The CCP2CON register controls the operation of CCP2. All are readable and writable.

For use of the CCP modules, refer to the Embedded Control Handbook, "Using the CCP Modules" (AN594).

**TABLE 10-1: CCP MODE - TIMER RESOURCE**

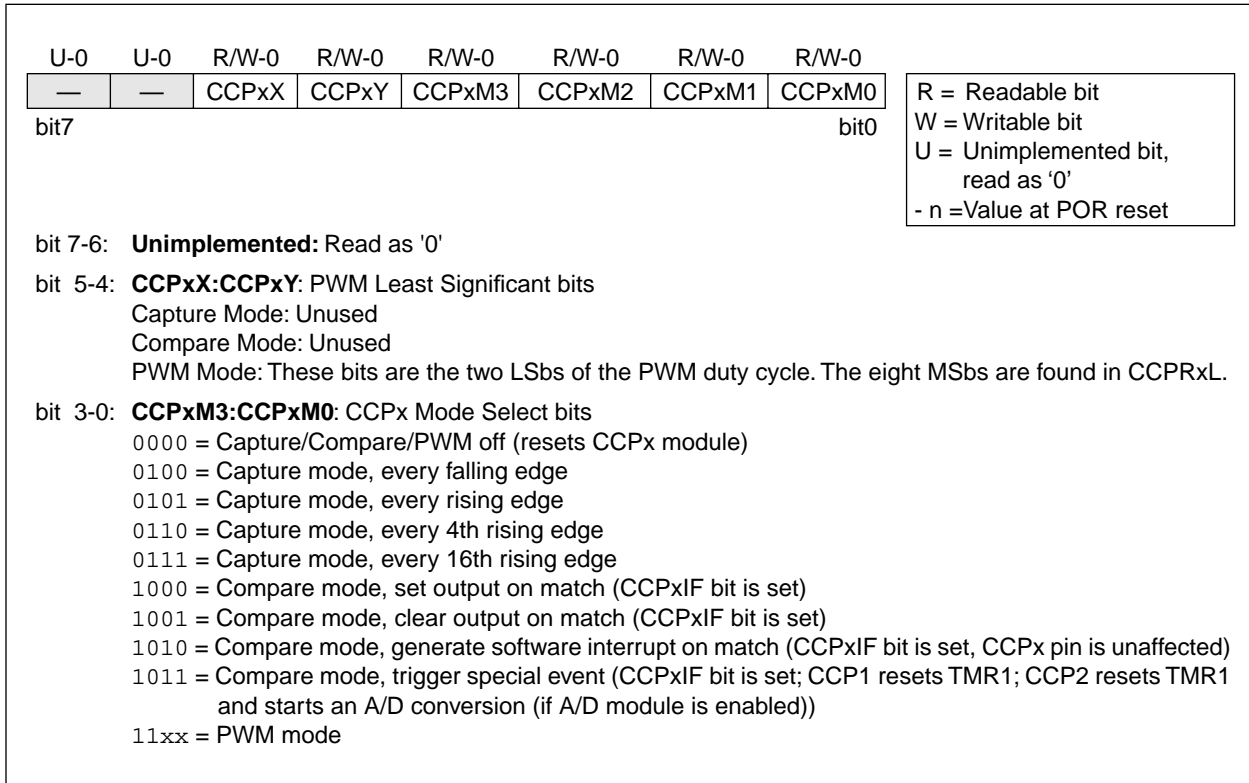
CCP Mode	Timer Resource
Capture	Timer1
Compare	Timer1
PWM	Timer2

**TABLE 10-2: INTERACTION OF TWO CCP MODULES**

CCPx Mode	CCPy Mode	Interaction
Capture	Capture	Same TMR1 time-base.
Capture	Compare	The compare should be configured for the special event trigger, which clears TMR1.
Compare	Compare	The compare(s) should be configured for the special event trigger, which clears TMR1.
PWM	PWM	The PWMs will have the same frequency, and update rate (TMR2 interrupt).
PWM	Capture	None
PWM	Compare	None

# PIC16C7X

**FIGURE 10-1: CCP1CON REGISTER (ADDRESS 17h)/CCP2CON REGISTER (ADDRESS 1Dh)**



## 10.1 Capture Mode

Applicable Devices			
72	73	73A	74
74A	76	77	

In Capture mode, CCPR1H:CCPR1L captures the 16-bit value of the TMR1 register when an event occurs on pin RC2/CCP1. An event is defined as:

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

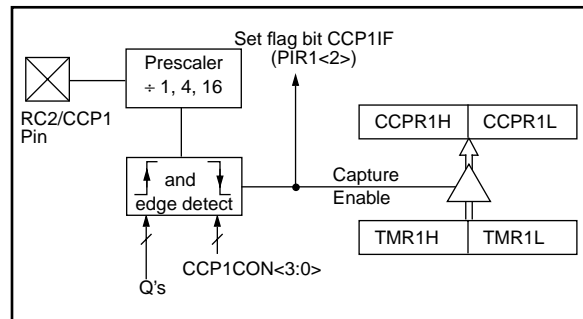
An event is selected by control bits CCP1M3:CCP1M0 (CCP1CON<3:0>). When a capture is made, the interrupt request flag bit CCP1IF (PIR1<2>) is set. It must be cleared in software. If another capture occurs before the value in register CCPR1 is read, the old captured value will be lost.

### 10.1.1 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 is configured as an output, a write to the port can cause a capture condition.

**FIGURE 10-2: CAPTURE MODE OPERATION BLOCK DIAGRAM**



### 10.1.2 TIMER1 MODE SELECTION

Timer1 must be running in timer mode or synchronized counter mode for the CCP module to use the capture feature. In asynchronous counter mode, the capture operation may not work.

### 10.1.3 SOFTWARE INTERRUPT

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



## 10.1.4 CCP PRESCALER

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

Switching from one capture prescaler to another may generate an interrupt. Also, the prescaler counter will not be cleared, therefore the first capture may be from a non-zero prescaler. Example 10-1 shows the recommended method for switching between capture prescalers. This example also clears the prescaler counter and will not generate the “false” interrupt.

### EXAMPLE 10-1: CHANGING BETWEEN CAPTURE PRESCALERS

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

## 10.2 Compare Mode

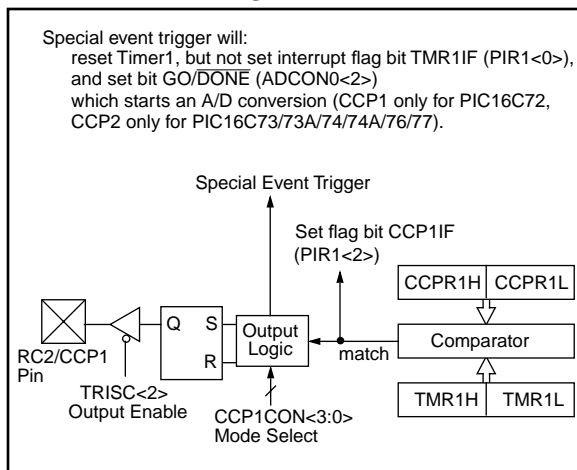
Applicable Devices					
72	73	73A	74	74A	76/77

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 10-3: COMPARE MODE OPERATION BLOCK DIAGRAM



## 10.2.1 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 data latch.

## 10.2.2 TIMER1 MODE SELECTION

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

## 10.2.3 SOFTWARE INTERRUPT MODE

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

## 10.2.4 SPECIAL EVENT TRIGGER

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

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

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

For the PIC16C72 only, the special event trigger output of CCP1 resets the TMR1 register pair, and starts an A/D conversion (if the A/D module is enabled).

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

# PIC16C7X

## 10.3 PWM Mode

Applicable Devices					
72	73	73A	74	74A	76/77

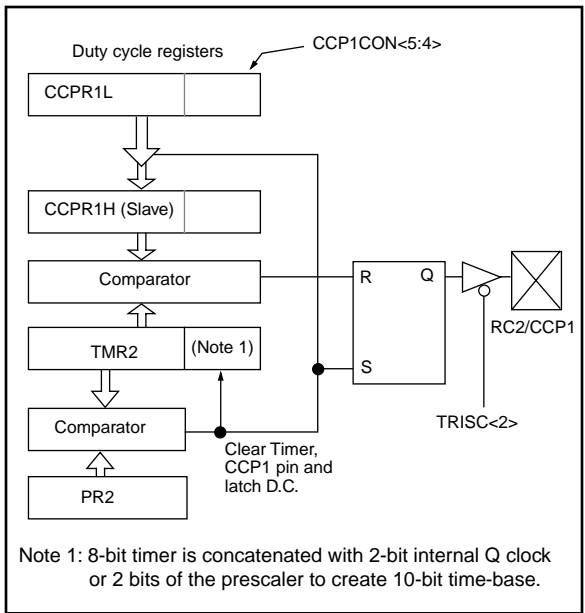
In Pulse Width Modulation (PWM) mode, the CCPx pin produces up to a 10-bit resolution PWM output. Since the CCP1 pin is multiplexed with the PORTC data latch, the TRISC<2> bit must be cleared to make the CCP1 pin an output.

**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 10-4 shows a simplified block diagram of the CCP module in PWM mode.

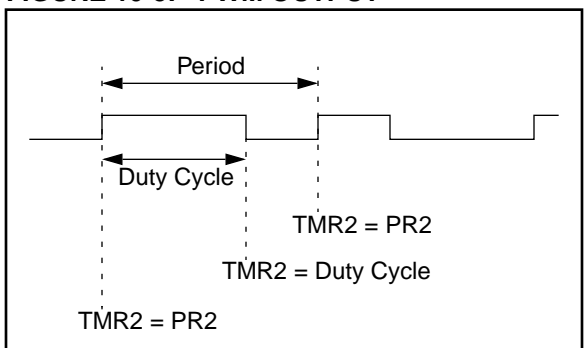
For a step by step procedure on how to set up the CCP module for PWM operation, see Section 10.3.3.

**FIGURE 10-4: SIMPLIFIED PWM BLOCK DIAGRAM**



A PWM output (Figure 10-5) 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 10-5: PWM OUTPUT**



### 10.3.1 PWM PERIOD

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

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

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 9.1) 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.

### 10.3.2 PWM DUTY CYCLE

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

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

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

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

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

Maximum PWM resolution (bits) for a given PWM frequency:

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

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

# PIC16C7X

## EXAMPLE 10-2: PWM PERIOD AND DUTY CYCLE CALCULATION

Desired PWM frequency is 78.125 kHz,  
 $F_{osc} = 20 \text{ MHz}$   
 TMR2 prescale = 1

$$1/78.125 \text{ kHz} = [(PR2) + 1] \cdot 4 \cdot 1/20 \text{ MHz} \cdot 1$$

$$12.8 \mu\text{s} = [(PR2) + 1] \cdot 4 \cdot 50 \text{ ns} \cdot 1$$

$$PR2 = 63$$

Find the maximum resolution of the duty cycle that can be used with a 78.125 kHz frequency and 20 MHz oscillator:

$$1/78.125 \text{ kHz} = 2^{\text{PWM RESOLUTION}} \cdot 1/20 \text{ MHz} \cdot 1$$

$$12.8 \mu\text{s} = 2^{\text{PWM RESOLUTION}} \cdot 50 \text{ ns} \cdot 1$$

$$256 = 2^{\text{PWM RESOLUTION}}$$

$$\log(256) = (\text{PWM Resolution}) \cdot \log(2)$$

$$8.0 = \text{PWM Resolution}$$

At most, an 8-bit resolution duty cycle can be obtained from a 78.125 kHz frequency and a 20 MHz oscillator, i.e.,  $0 \leq \text{CCPR1L:CCP1CON}\langle 5:4 \rangle \leq 255$ . Any value greater than 255 will result in a 100% duty cycle.

In order to achieve higher resolution, the PWM frequency must be decreased. In order to achieve higher PWM frequency, the resolution must be decreased.

Table 10-3 lists example PWM frequencies and resolutions for  $F_{osc} = 20 \text{ MHz}$ . The TMR2 prescaler and PR2 values are also shown.

### 10.3.3 SET-UP 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 10-3: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS AT 20 MHz**

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

**TABLE 10-4: 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	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF <sup>(1,2)</sup>	ADIF	RCIF <sup>(2)</sup>	TXIF <sup>(2)</sup>	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
0Dh <sup>(2)</sup>	PIR2	—	—	—	—	—	—	—	CCP2IF	---- --0	---- --0
8Ch	PIE1	PSPIE <sup>(1,2)</sup>	ADIE	RCIE <sup>(2)</sup>	TXIE <sup>(2)</sup>	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
8Dh <sup>(2)</sup>	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	T1SYNCR	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 <sup>(2)</sup>	CCPR2L	Capture/Compare/PWM register2 (LSB)								xxxx xxxx	uuuu uuuu
1Ch <sup>(2)</sup>	CCPR2H	Capture/Compare/PWM register2 (MSB)								xxxx xxxx	uuuu uuuu
1Dh <sup>(2)</sup>	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: Bits PSPIE and PSPIF are reserved on the PIC16C73/73A/76, always maintain these bits clear.

Note 2: The PIC16C72 does not have a Parallel Slave Port, USART or CCP2 module, these bits are unimplemented, read as '0'.

# PIC16C7X

**TABLE 10-5: 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	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF <sup>(1,2)</sup>	ADIF	RCIF <sup>(2)</sup>	TXIF <sup>(2)</sup>	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
0Dh <sup>(2)</sup>	PIR2	—	—	—	—	—	—	—	CCP2IF	---- --0	---- --0
8Ch	PIE1	PSPIE <sup>(1,2)</sup>	ADIE	RCIE <sup>(2)</sup>	TXIE <sup>(2)</sup>	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
8Dh <sup>(2)</sup>	PIE2	—	—	—	—	—	—	—	CCP2IE	---- --0	---- --0
87h	TRISC	PORTC Data Direction Register								1111 1111	1111 1111
11h	TMR2	Timer2 module's register								0000 0000	0000 0000
92h	PR2	Timer2 module's 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
1Bh <sup>(2)</sup>	CCPR2L	Capture/Compare/PWM register2 (LSB)								xxxx xxxx	uuuu uuuu
1Ch <sup>(2)</sup>	CCPR2H	Capture/Compare/PWM register2 (MSB)								xxxx xxxx	uuuu uuuu
1Dh <sup>(2)</sup>	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 PIC16C73/73A/76, always maintain these bits clear.

2: The PIC16C72 does not have a Parallel Slave Port, USART or CCP2 module, these bits are unimplemented, read as '0'.

Applicable Devices					
72	73	73A	74	74A	76/77

# PIC16C7X

## 11.0 SYNCHRONOUS SERIAL PORT (SSP) MODULE

### 11.1 SSP Module Overview

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

- Serial Peripheral Interface (SPI)
- Inter-Integrated Circuit (I<sup>2</sup>C)

The SSP module in I<sup>2</sup>C mode works the same in all PIC16C7X devices that have an SSP module. However the SSP Module in SPI mode has differences between the PIC16C76/77 and the other PIC16C7X devices.

The register definitions and operational description of SPI mode has been split into two sections because of the differences between the PIC16C76/77 and the other PIC16C7X devices. The default reset values of both the SPI modules is the same regardless of the device:

11.2 SPI Mode for PIC16C72/73/73A/74/74A .....	78
11.3 SPI Mode for PIC16C76/77 .....	83
11.4 I <sup>2</sup> C™ Overview .....	89
11.5 SSP I <sup>2</sup> C Operation.....	93

Refer to Application Note AN578, *“Use of the SSP Module in the I<sup>2</sup>C Multi-Master Environment.”*

# PIC16C7X

Applicable Devices						
72	73	73A	74	74A	76	77

## 11.2 SPI Mode for PIC16C72/73/73A/74/74A

This section contains register definitions and operational characteristics of the SPI module for the PIC16C72, PIC16C73, PIC16C73A, PIC16C74, PIC16C74A.

**FIGURE 11-1: SSPSTAT: SYNC SERIAL PORT STATUS REGISTER (ADDRESS 94h)**

U-0	U-0	R-0	R-0	R-0	R-0	R-0	R-0	
—	—	D/ $\bar{A}$	P	S	R/ $\bar{W}$	UA	BF	
bit7								bit0

R = Readable bit  
W = Writable bit  
U = Unimplemented bit, read as '0'  
- n = Value at POR reset

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

bit 5: **D/ $\bar{A}$ :** Data/Address bit (I<sup>2</sup>C 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 (I<sup>2</sup>C mode only). This bit is cleared when the SSP module is disabled, 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 (I<sup>2</sup>C mode only). This bit is cleared when the SSP module is disabled, 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/ $\bar{W}$ :** Read/Write bit information (I<sup>2</sup>C mode only)  
This bit holds the R/W bit information following the last address match. This bit is valid from the address match to the next start bit, stop bit, or  $\bar{ACK}$  bit.  
1 = Read  
0 = Write

bit 1: **UA:** Update Address (10-bit I<sup>2</sup>C mode only)  
1 = Indicates that the user needs to update the address in the SSPADD register  
0 = Address does not need to be updated

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

Receive (SPI and I<sup>2</sup>C modes)  
1 = Receive complete, SSPBUF is full  
0 = Receive not complete, SSPBUF is empty

Transmit (I<sup>2</sup>C mode only)  
1 = Transmit in progress, SSPBUF is full  
0 = Transmit complete, SSPBUF is empty

Applicable Devices							
72	73	73A	74	74A	76	77	

# PIC16C7X

**FIGURE 11-2: 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	
bit7								bit0

R = Readable bit  
W = Writable bit  
U = Unimplemented bit, read as '0'  
- n = Value at POR reset

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 Detect 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 register is lost. Overflow can only occur in slave mode. The user must read the SSPBUF, even if only transmitting data, to avoid setting overflow. In master mode the overflow bit is not set since each new reception (and transmission) is initiated by writing to the SSPBUF register.  
0 = No overflow  
In I<sup>2</sup>C mode  
1 = A byte is received while the SSPBUF register is still holding the previous byte. SSPOV is a "don't care" in transmit mode. SSPOV must be cleared in software in either mode.  
0 = No overflow

bit 5: **SSPEN**: Synchronous Serial Port Enable bit  
In SPI mode  
1 = Enables serial port and configures SCK, SDO, and SDI as serial port pins  
0 = Disables serial port and configures these pins as I/O port pins  
In I<sup>2</sup>C mode  
1 = Enables the serial port and configures the SDA and SCL pins as serial port pins  
0 = Disables serial port and configures these pins as I/O port pins  
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. Transmit happens on falling edge, receive on rising edge.  
0 = Idle state for clock is a low level. Transmit happens on rising edge, receive on falling edge.  
In I<sup>2</sup>C 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 = I<sup>2</sup>C slave mode, 7-bit address  
0111 = I<sup>2</sup>C slave mode, 10-bit address  
1011 = I<sup>2</sup>C firmware controlled Master Mode (slave idle)  
1110 = I<sup>2</sup>C slave mode, 7-bit address with start and stop bit interrupts enabled  
1111 = I<sup>2</sup>C slave mode, 10-bit address with start and stop bit interrupts enabled

# PIC16C7X

Applicable Devices						
72	73	73A	74	74A	76	77

## 11.2.1 OPERATION OF SSP MODULE IN SPI MODE

Applicable Devices						
72	73	73A	74	74A	76	77

The 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)
- Serial Data In (SDI)
- Serial Clock (SCK)

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

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

When initializing the SPI, several options need to be specified. This is done by programming the appropriate control bits in the SSPCON register (SSPCON<5:0>). 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 (Output/Input data on the Rising/Falling edge of SCK)
- Clock Rate (Master mode only)
- Slave Select Mode (Slave mode only)

The SSP consists of a transmit/receive Shift Register (SSPSR) and a Buffer register (SSPBUF). The SSPSR shifts the data in and out of the device, MSb first. The SSPBUF holds the data that was written to the SSPSR, until the received data is ready. Once the 8-bits of data have been received, that byte is moved to the SSPBUF register. Then the Buffer Full bit, BF (SSPSTAT<0>) and flag bit SSPIF are set. This double buffering of the received data (SSPBUF) allows the next byte to start reception before reading the data that was just received. Any write to the SSPBUF register during transmission/reception of data will be ignored, and the write collision detect bit, WCOL (SSPCON<7>) will be set. User software must clear bit WCOL so that it can be determined if the following write(s) to the SSPBUF completed successfully. When the application software is expecting to receive valid data, the SSPBUF register should be read before the next byte of data to transfer is written to the SSPBUF register. The Buffer Full bit BF (SSPSTAT<0>) indicates when the SSPBUF register has been loaded with the received data (transmission is complete). When the SSPBUF is read, bit BF is cleared. This data may be irrelevant if the SPI is only a transmitter. Generally the SSP Interrupt is used to determine when the transmission/reception has completed. The SSPBUF register must be read and/or written. If the interrupt method is not going to be used, then software polling can be done to ensure that a write collision does not occur. Example 11-1 shows the loading of the SSPBUF (SSPSR) register for data transmission. The shaded instruction is only required if the received data is meaningful.

## EXAMPLE 11-1: LOADING THE SSPBUF (SSPSR) REGISTER

```

BSF STATUS, RP0 ;Specify Bank 1
LOOP BTFSS SSPSTAT, BF ;Has data been
;received
;(transmit
;complete)?

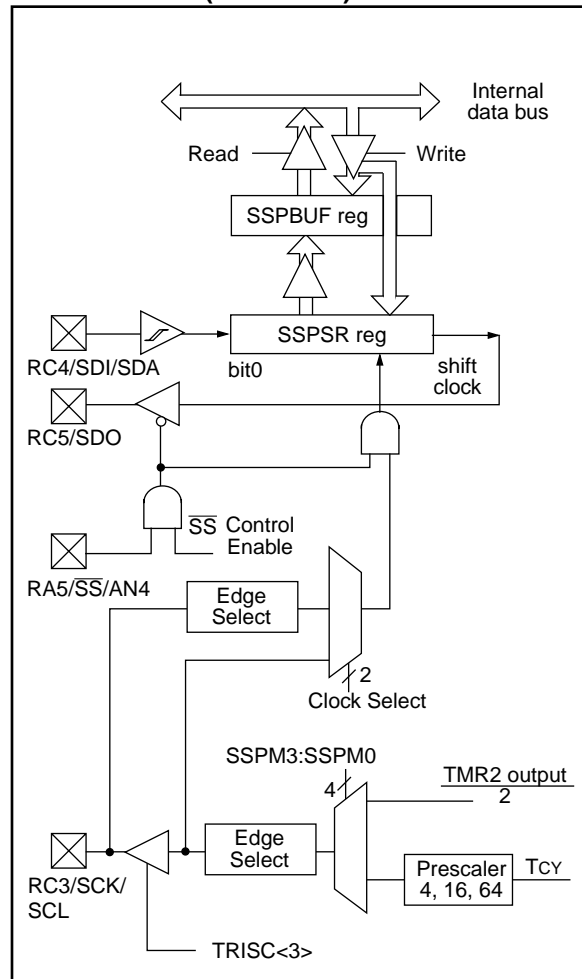
GOTO LOOP ;No
BCF STATUS, RP0 ;Specify Bank 0
MOVWF SSPBUF, W ;W reg = contents
;of SSPBUF

MOVWF RXDATA ;Save in user RAM
MOVF TXDATA, W ;W reg = contents
; of TXDATA

MOVWF SSPBUF ;New data to xmit
    
```

The block diagram of the SSP module, when in SPI mode (Figure 11-3), shows that the SSPSR register is not directly readable or writable, and can only be accessed from addressing the SSPBUF register. Additionally, the SSP status register (SSPSTAT) indicates the various status conditions.

**FIGURE 11-3: SSP BLOCK DIAGRAM (SPI MODE)**





Applicable Devices						
72	73	73A	74	74A	76	77

# PIC16C7X

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

- 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 (if implemented)

Any serial port function that is not desired may be overridden by programming the corresponding data direction (TRIS) register to the opposite value. An example would be in master mode where you are only sending data (to a display driver), then both SDI and  $\overline{SS}$  could be used as general purpose outputs by clearing their corresponding TRIS register bits.

Figure 11-4 shows a typical connection between two microcontrollers. The master controller (Processor 1) initiates the data transfer by sending the SCK signal. Data is shifted out of both shift registers on their programmed clock edge, and latched on the opposite edge of the clock. Both processors should be programmed to the same Clock Polarity (CKP), then both controllers would send and receive data at the same time. Whether the data is meaningful (or dummy data) depends on the application software. This leads to three scenarios for data transmission:

- Master sends data — Slave sends dummy data
- Master sends data — Slave sends data
- Master sends dummy data — Slave sends data

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

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

In slave mode, the data is transmitted and received as the external clock pulses appear on SCK. When the last bit is latched interrupt flag bit SSPIF (PIR1<3>) is set.

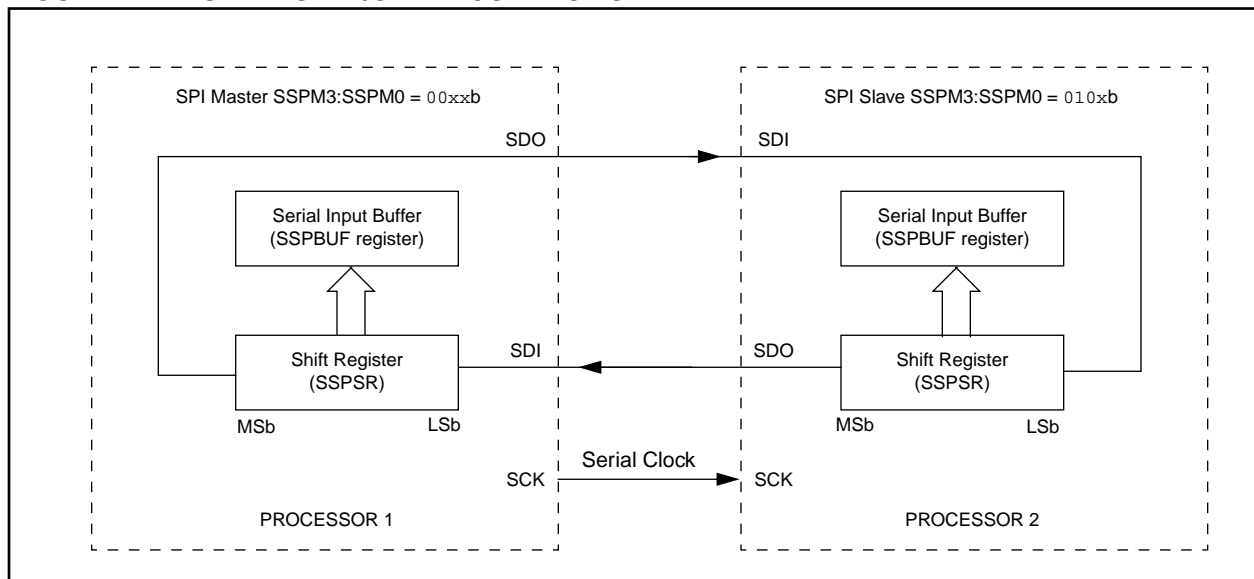
The clock polarity is selected by appropriately programming bit CKP (SSPCON<4>). This then would give waveforms for SPI communication as shown in Figure 11-5 and Figure 11-6 where the MSB is transmitted first. In master mode, the SPI clock rate (bit rate) is user programmable to be one of the following:

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

This allows a maximum bit clock frequency (at 20 MHz) of 5 MHz. When in slave mode the external clock must meet the minimum high and low times.

In sleep mode, the slave can transmit and receive data and wake the device from sleep.

**FIGURE 11-4: SPI MASTER/SLAVE CONNECTION**



# PIC16C7X

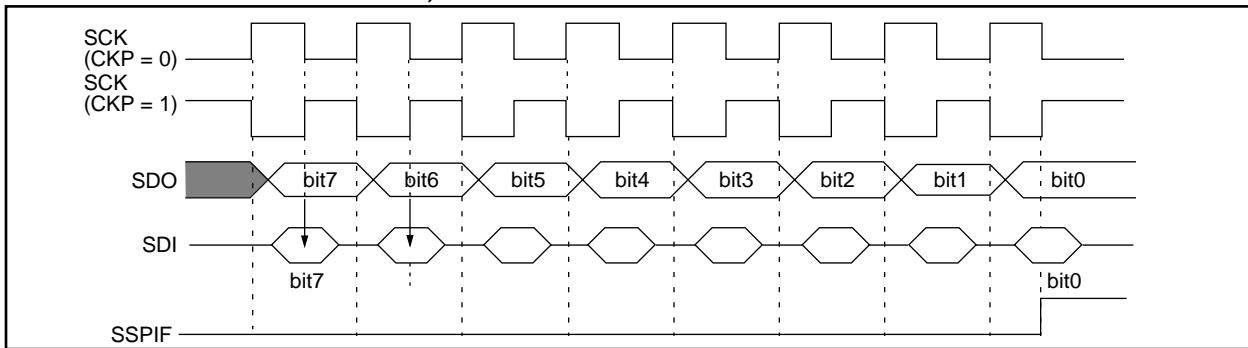
Applicable Devices					
72	73	73A	74	74A	76/77

The  $\overline{SS}$  pin allows a synchronous slave mode. The SPI must be in slave mode (SSPCON<3:0> = 04h) and the TRISA<5> bit must be set for synchronous slave mode to be enabled. When the  $\overline{SS}$  pin is low, transmission and reception are enabled and the SDO pin is driven. When the  $\overline{SS}$  pin goes high, the SDO pin is no longer driven, even if in the middle of a transmitted byte, and becomes a floating output. If the  $\overline{SS}$  pin is taken low without resetting SPI mode, the transmission will continue from the

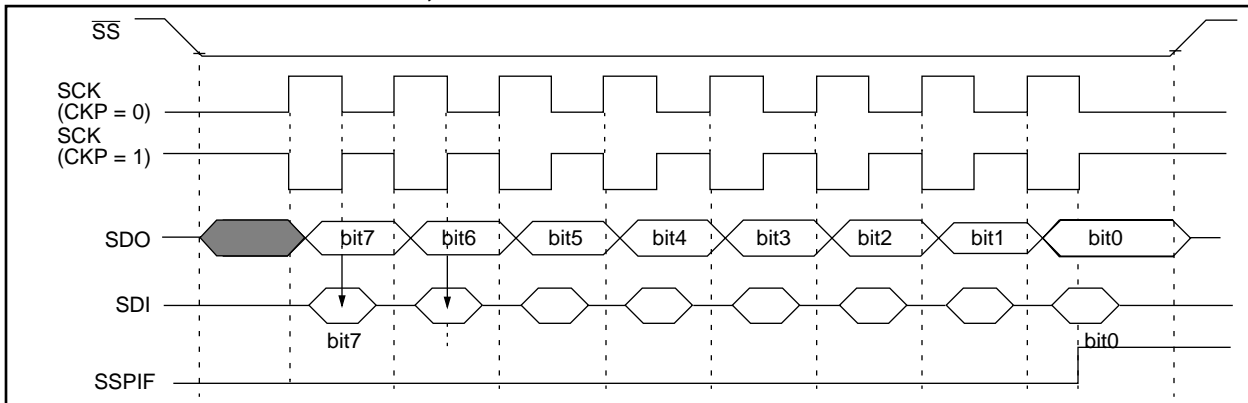
point at which it was taken high. External pull-up/pull-down resistors may be desirable, depending on the application.

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

**FIGURE 11-5: SPI MODE TIMING, MASTER MODE OR SLAVE MODE W/O  $\overline{SS}$  CONTROL**



**FIGURE 11-6: SPI MODE TIMING, SLAVE MODE WITH  $\overline{SS}$  CONTROL**



**TABLE 11-1: 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	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF <sup>(1,2)</sup>	ADIF	RCIF <sup>(2)</sup>	TXIF <sup>(2)</sup>	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
8Ch	PIE1	PSPIE <sup>(1,2)</sup>	ADIE	RCIE <sup>(2)</sup>	TXIE <sup>(2)</sup>	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	—	—	D/Ā	P	S	R/Ā	UA	BF	--00 0000	--00 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 PIC16C73/73A, always maintain these bits clear.

Note 2: The PIC16C72 does not have a Parallel Slave Port or USART, these bits are unimplemented, read as '0'.

Applicable Devices					
72	73	73A	74	74A	76/77

# PIC16C7X

## 11.3 SPI Mode for PIC16C76/77

This section contains register definitions and operational characteristics of the SPI module on the PIC16C76 and PIC16C77 only.

**FIGURE 11-7: SSPSTAT: SYNC SERIAL PORT STATUS REGISTER (ADDRESS 94h)(PIC16C76/77)**

R/W-0	R/W-0	R-0	R-0	R-0	R-0	R-0	R-0
SMP	CKE	D/A	P	S	R/W	UA	BF
bit7							bit0

R = Readable bit  
W = Writable bit  
U = Unimplemented bit, read as '0'  
- n = Value at POR reset

bit 7: **SMP**: SPI data input sample phase  
SPI Master Mode  
1 = Input data sampled at end of data output time  
0 = Input data sampled at middle of data output time  
SPI Slave Mode  
SMP must be cleared when SPI is used in slave mode

bit 6: **CKE**: SPI Clock Edge Select (Figure 11-11, Figure 11-12, and Figure 11-13)  
CKP = 0  
1 = Data transmitted on rising edge of SCK  
0 = Data transmitted on falling edge of SCK  
CKP = 1  
1 = Data transmitted on falling edge of SCK  
0 = Data transmitted on rising edge of SCK

bit 5: **D/A**: Data/Address bit (I<sup>2</sup>C 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 (I<sup>2</sup>C 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 (I<sup>2</sup>C 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/W**: Read/Write bit information (I<sup>2</sup>C 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  $\overline{ACK}$  bit.  
1 = Read  
0 = Write

bit 1: **UA**: Update Address (10-bit I<sup>2</sup>C mode only)  
1 = Indicates that the user needs to update the address in the SSPADD register  
0 = Address does not need to be updated

bit 0: **BF**: Buffer Full Status bit  
Receive (SPI and I<sup>2</sup>C modes)  
1 = Receive complete, SSPBUF is full  
0 = Receive not complete, SSPBUF is empty  
Transmit (I<sup>2</sup>C mode only)  
1 = Transmit in progress, SSPBUF is full  
0 = Transmit complete, SSPBUF is empty

# PIC16C7X

Applicable Devices					
72	73	73A	74	74A	76/77

**FIGURE 11-8: SSPCON: SYNC SERIAL PORT CONTROL REGISTER (ADDRESS 14h)(PIC16C76/77)**

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	
bit7								bit0

R = Readable bit  
W = Writable bit  
U = Unimplemented bit, read as '0'  
- n = Value at POR reset

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 I<sup>2</sup>C mode  
1 = A byte is received while the SSPBUF register is still holding the previous byte. SSPOV is a "don't care" in transmit mode. SSPOV must be cleared in software in either mode.  
0 = No overflow

bit 5: **SSPEN**: Synchronous Serial Port Enable bit  
In SPI mode  
1 = Enables serial port and configures SCK, SDO, and SDI as serial port pins  
0 = Disables serial port and configures these pins as I/O port pins  
In I<sup>2</sup>C mode  
1 = Enables the serial port and configures the SDA and SCL pins as serial port pins  
0 = Disables serial port and configures these pins as I/O port pins  
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  
0 = Idle state for clock is a low level  
In I<sup>2</sup>C 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 = I<sup>2</sup>C slave mode, 7-bit address  
0111 = I<sup>2</sup>C slave mode, 10-bit address  
1011 = I<sup>2</sup>C firmware controlled master mode (slave idle)  
1110 = I<sup>2</sup>C slave mode, 7-bit address with start and stop bit interrupts enabled  
1111 = I<sup>2</sup>C slave mode, 10-bit address with start and stop bit interrupts enabled

Applicable Devices					
72	73	73A	74	74A	76/77

# PIC16C7X

## 11.3.1 SPI MODE FOR PIC16C76/77

The 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)

The SSP consists of a transmit/receive Shift Register (SSPSR) and a buffer register (SSPBUF). The SSPSR shifts the data in and out of the device, MSb first. The SSPBUF holds the data that was written to the SSPSR until the received data is ready. Once the 8-bits of data have been received, that byte is moved to the SSPBUF register. Then the buffer full detect bit BF (SSPSTAT<0>) and interrupt flag bit SSIIF (PIR1<3>) are set. This double buffering of the received data (SSPBUF) allows the next byte to start reception before reading the data that was just received. Any write to the SSPBUF register during transmission/reception of data will be ignored, and the write collision detect bit WCOL (SSPCON<7>) will be set. User software must clear the WCOL bit so that it can be determined if the following write(s) to the SSPBUF register completed successfully. When the application software is expecting to receive valid data, the SSPBUF should be read before the next byte of data to transfer is written to the SSPBUF. Buffer full bit BF (SSPSTAT<0>) indicates when SSPBUF has been loaded with the received data (transmission is complete). When the SSPBUF is read, bit BF is cleared. This data may be irrelevant if the SPI is only a transmitter. Generally the SSP Interrupt is used to determine when the transmission/reception has completed. The SSPBUF must be read and/or written. If the interrupt method is not going to be used, then software polling can be done to ensure that a write collision does not occur. Example 11-2 shows the loading of the SSPBUF (SSPSR) for data transmission. The shaded instruction is only required if the received data is meaningful.

## EXAMPLE 11-2: LOADING THE SSPBUF (SSPSR) REGISTER (PIC16C76/77)

```

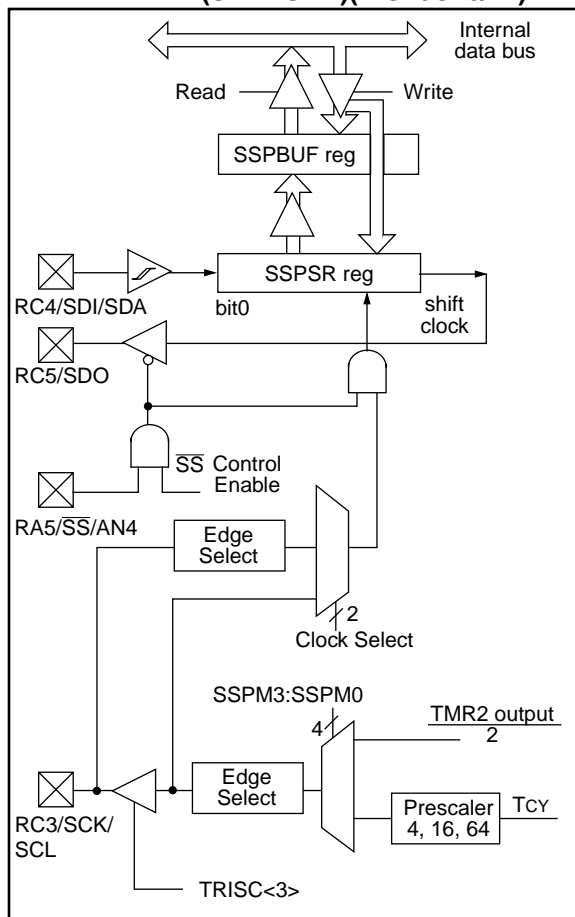
BCF    STATUS, RP1    ;Specify Bank 1
BSF    STATUS, RP0    ;
LOOP  BTFSS SSPSTAT, BF ;Has data been
                                ;received
                                ;(transmit
                                ;(transmit
                                ;complete)?

GOTO   LOOP           ;No
BCF    STATUS, RP0    ;Specify Bank 0
MOVWF  SSPBUF, W      ;W reg = contents
                                ; of SSPBUF
MOVWF  RXDATA         ;Save in user RAM
MOVWF  TXDATA, W      ;W reg = contents
                                ; of TXDATA
MOVWF  SSPBUF         ;New data to xmit

```

The block diagram of the SSP module, when in SPI mode (Figure 11-9), shows that the SSPSR is not directly readable or writable, and can only be accessed from addressing the SSPBUF register. Additionally, the SSP status register (SSPSTAT) indicates the various status conditions.

**FIGURE 11-9: SSP BLOCK DIAGRAM (SPI MODE)(PIC16C76/77)**



# PIC16C7X

Applicable Devices					
72	73	73A	74	74A	76/77

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

- 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

Any serial port function that is not desired may be overridden by programming the corresponding data direction (TRIS) register to the opposite value. An example would be in master mode where you are only sending data (to a display driver), then both SDI and  $\overline{SS}$  could be used as general purpose outputs by clearing their corresponding TRIS register bits.

Figure 11-10 shows a typical connection between two microcontrollers. The master controller (Processor 1) initiates the data transfer by sending the SCK signal. Data is shifted out of both shift registers on their programmed clock edge, and latched on the opposite edge of the clock. Both processors should be programmed to same Clock Polarity (CKP), then both controllers would send and receive data at the same time. Whether the data is meaningful (or dummy data) depends on the application firmware. This leads to three scenarios for data transmission:

- Master sends data — Slave sends dummy data
- Master sends data — Slave sends data
- Master sends dummy data — Slave sends data

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

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

In slave mode, the data is transmitted and received as the external clock pulses appear on SCK. When the last bit is latched the interrupt flag bit SSPIF (PIR1<3>) is set.

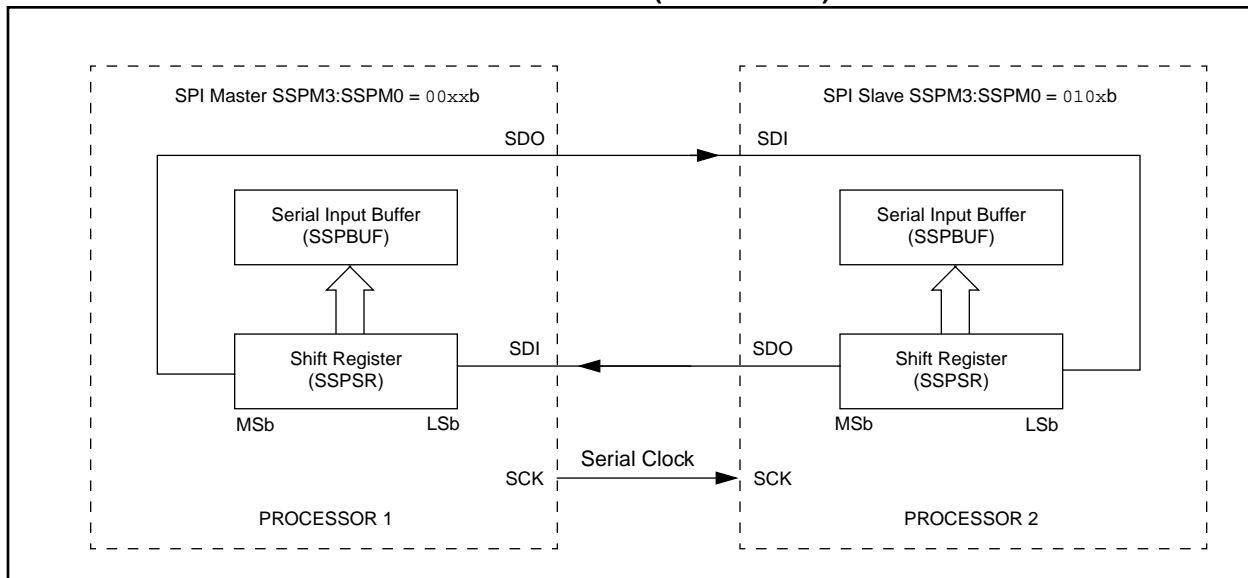
The clock polarity is selected by appropriately programming bit CKP (SSPCON<4>). This then would give waveforms for SPI communication as shown in Figure 11-11, Figure 11-12, and Figure 11-13 where the MSB is transmitted first. In master mode, the SPI clock rate (bit rate) is user programmable to be one of the following:

- FOSC/4 (or T<sub>CY</sub>)
- FOSC/16 (or 4 • T<sub>CY</sub>)
- FOSC/64 (or 16 • T<sub>CY</sub>)
- Timer2 output/2

This allows a maximum bit clock frequency (at 20 MHz) of 5 MHz. When in slave mode the external clock must meet the minimum high and low times.

In sleep mode, the slave can transmit and receive data and wake the device from sleep.

**FIGURE 11-10: SPI MASTER/SLAVE CONNECTION (PIC16C76/77)**



Applicable Devices					
72	73	73A	74	74A	76/77

# PIC16C7X

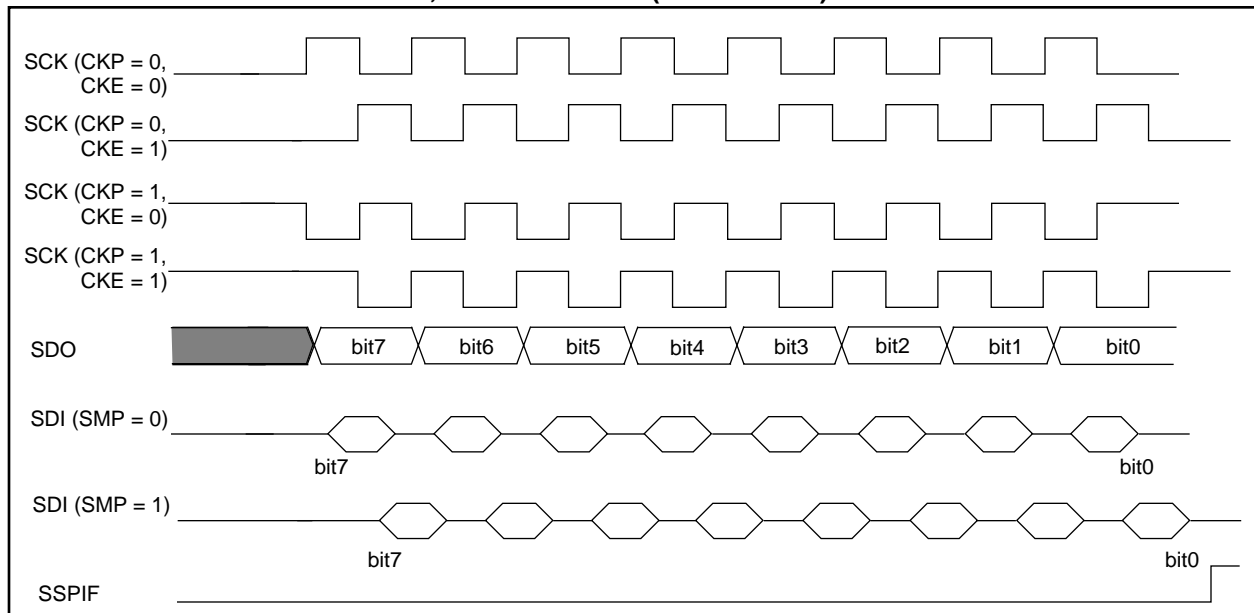
The  $\overline{SS}$  pin allows a synchronous slave mode. The SPI must be in slave mode ( $SSPCON\langle 3:0 \rangle = 04h$ ) and the  $TRISA\langle 5 \rangle$  bit must be set for the synchronous slave mode to be enabled. When the  $\overline{SS}$  pin is low, transmission and reception are enabled and the SDO pin is driven. When the  $\overline{SS}$  pin goes high, the SDO pin is no longer driven, even if in the middle of a transmitted byte, and becomes a floating output. If the  $\overline{SS}$  pin is taken low without resetting SPI mode, the transmission will continue from the point at which it was taken high. External pull-up/pull-down resistors may be desirable, depending on the application.

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

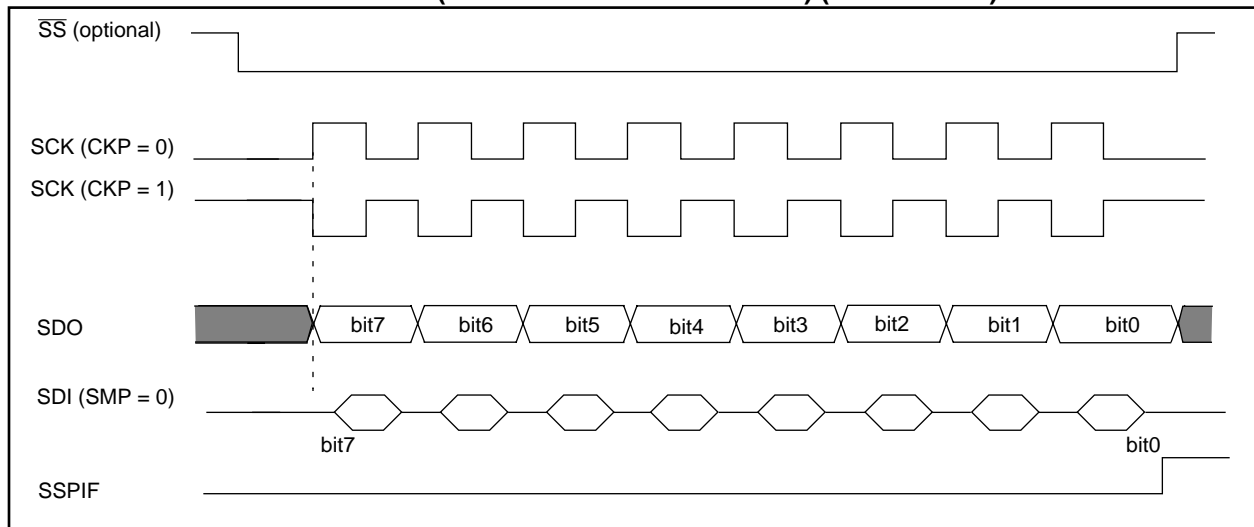
**Note:** If the SPI is used in Slave Mode with  $CKE = '1'$ , then the  $\overline{SS}$  pin control must be enabled.

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

**FIGURE 11-11: SPI MODE TIMING, MASTER MODE (PIC16C76/77)**



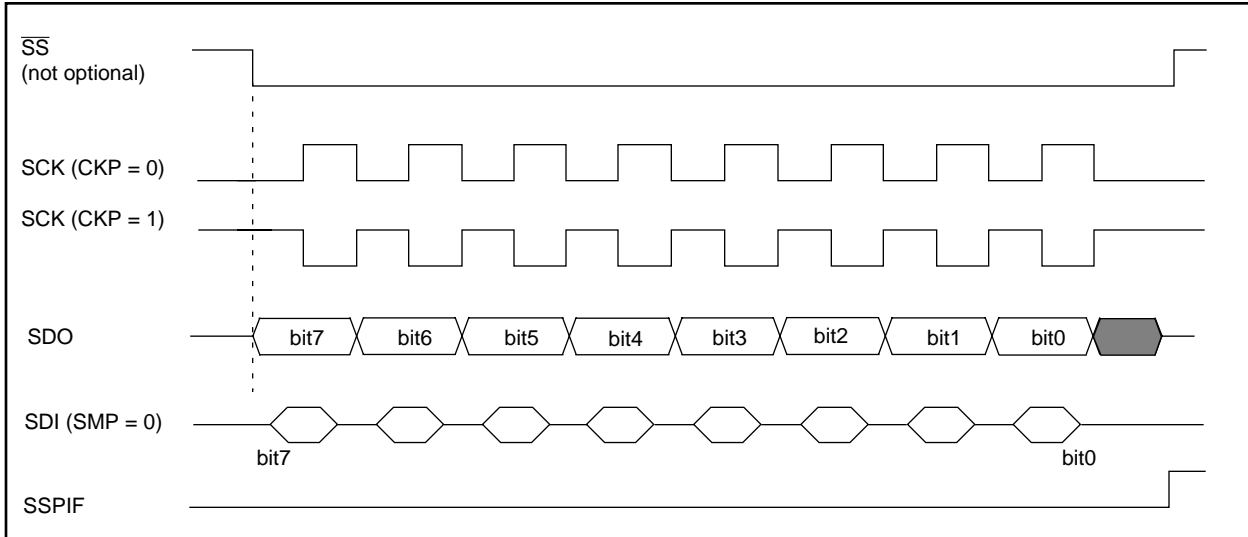
**FIGURE 11-12: SPI MODE TIMING (SLAVE MODE WITH CKE = 0) (PIC16C76/77)**



# PIC16C7X

Applicable Devices					
72	73	73A	74	74A	76/77

**FIGURE 11-13: SPI MODE TIMING (SLAVE MODE WITH CKE = 1) (PIC16C76/77)**



**TABLE 11-2: REGISTERS ASSOCIATED WITH SPI OPERATION (PIC16C76/77)**

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	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF <sup>(1)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
8Ch	PIE1	PSPIE <sup>(1)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
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/A	P	S	R/W	UA	BF	0000 0000	0000 0000

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

Note 1: Bits PSPIE and PSPIF are reserved on the PIC16C76, always maintain these bits clear.



Applicable Devices					
72	73	73A	74	74A	76/77

# PIC16C7X

## 11.4 I<sup>2</sup>C™ Overview

This section provides an overview of the Inter-Integrated Circuit (I<sup>2</sup>C) bus, with Section 11.5 discussing the operation of the SSP module in I<sup>2</sup>C mode.

The I<sup>2</sup>C bus is a two-wire serial interface developed by the Philips Corporation. The original specification, or standard mode, was for data transfers of up to 100 Kbps. The enhanced specification (fast mode) is also supported. This device will communicate with both standard and fast mode devices if attached to the same bus. The clock will determine the data rate.

The I<sup>2</sup>C interface employs a comprehensive protocol to ensure reliable transmission and reception of data. When transmitting data, one device is the “master” which initiates transfer on the bus and generates the clock signals to permit that transfer, while the other device(s) acts as the “slave.” All portions of the slave protocol are implemented in the SSP module’s hardware, except general call support, while portions of the master protocol need to be addressed in the PIC16CXX software. Table 11-3 defines some of the I<sup>2</sup>C bus terminology. For additional information on the I<sup>2</sup>C interface specification, refer to the Philips document “*The I<sup>2</sup>C bus and how to use it.*” #939839340011, which can be obtained from the Philips Corporation.

In the I<sup>2</sup>C interface protocol each device has an address. When a master wishes to initiate a data transfer, it first transmits the address of the device that it wishes to “talk” to. All devices “listen” to see if this is their address. Within this address, a bit specifies if the master wishes to read-from/write-to the slave device. The master and slave are always in opposite modes (transmitter/receiver) of operation during a data transfer. That is they can be thought of as operating in either of these two relations:

- Master-transmitter and Slave-receiver
- Slave-transmitter and Master-receiver

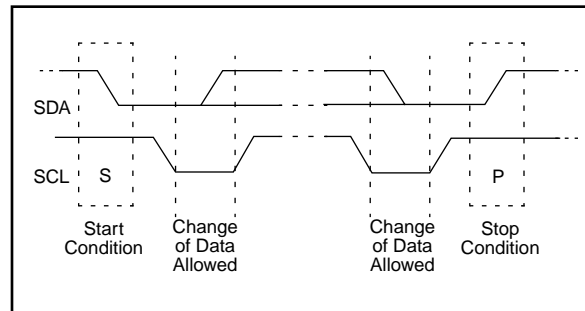
In both cases the master generates the clock signal.

The output stages of the clock (SCL) and data (SDA) lines must have an open-drain or open-collector in order to perform the wired-AND function of the bus. External pull-up resistors are used to ensure a high level when no device is pulling the line down. The number of devices that may be attached to the I<sup>2</sup>C bus is limited only by the maximum bus loading specification of 400 pF.

### 11.4.1 INITIATING AND TERMINATING DATA TRANSFER

During times of no data transfer (idle time), both the clock line (SCL) and the data line (SDA) are pulled high through the external pull-up resistors. The START and STOP conditions determine the start and stop of data transmission. The START condition is defined as a high to low transition of the SDA when the SCL is high. The STOP condition is defined as a low to high transition of the SDA when the SCL is high. Figure 11-14 shows the START and STOP conditions. The master generates these conditions for starting and terminating data transfer. Due to the definition of the START and STOP conditions, when data is being transmitted, the SDA line can only change state when the SCL line is low.

**FIGURE 11-14: START AND STOP CONDITIONS**



**TABLE 11-3: I<sup>2</sup>C BUS TERMINOLOGY**

Term	Description
Transmitter	The device that sends the data to the bus.
Receiver	The device that receives the data from the bus.
Master	The device which initiates the transfer, generates the clock and terminates the transfer.
Slave	The device addressed by a master.
Multi-master	More than one master device in a system. These masters can attempt to control the bus at the same time without corrupting the message.
Arbitration	Procedure that ensures that only one of the master devices will control the bus. This ensure that the transfer data does not get corrupted.
Synchronization	Procedure where the clock signals of two or more devices are synchronized.

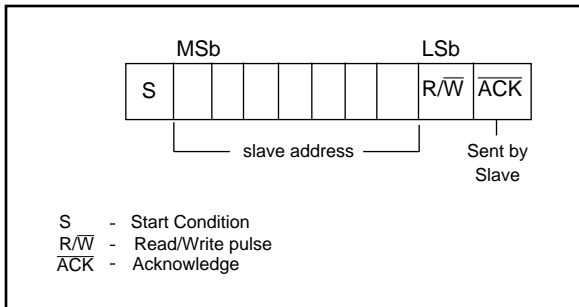
# PIC16C7X

Applicable Devices					
72	73	73A	74	74A	76/77

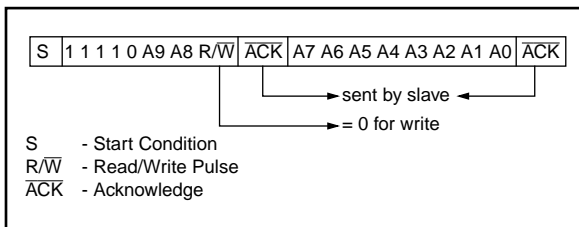
## 11.4.2 ADDRESSING I<sup>2</sup>C DEVICES

There are two address formats. The simplest is the 7-bit address format with a R/W bit (Figure 11-15). The more complex is the 10-bit address with a R/W bit (Figure 11-16). For 10-bit address format, two bytes must be transmitted with the first five bits specifying this to be a 10-bit address.

**FIGURE 11-15: 7-BIT ADDRESS FORMAT**



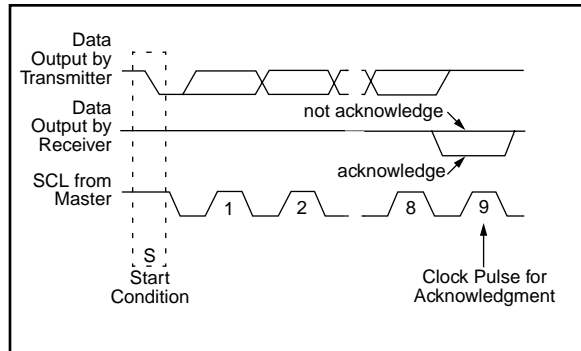
**FIGURE 11-16: I<sup>2</sup>C 10-BIT ADDRESS FORMAT**



## 11.4.3 TRANSFER ACKNOWLEDGE

All data must be transmitted per byte, with no limit to the number of bytes transmitted per data transfer. After each byte, the slave-receiver generates an acknowledge bit (ACK) (Figure 11-17). When a slave-receiver doesn't acknowledge the slave address or received data, the master must abort the transfer. The slave must leave SDA high so that the master can generate the STOP condition (Figure 11-14).

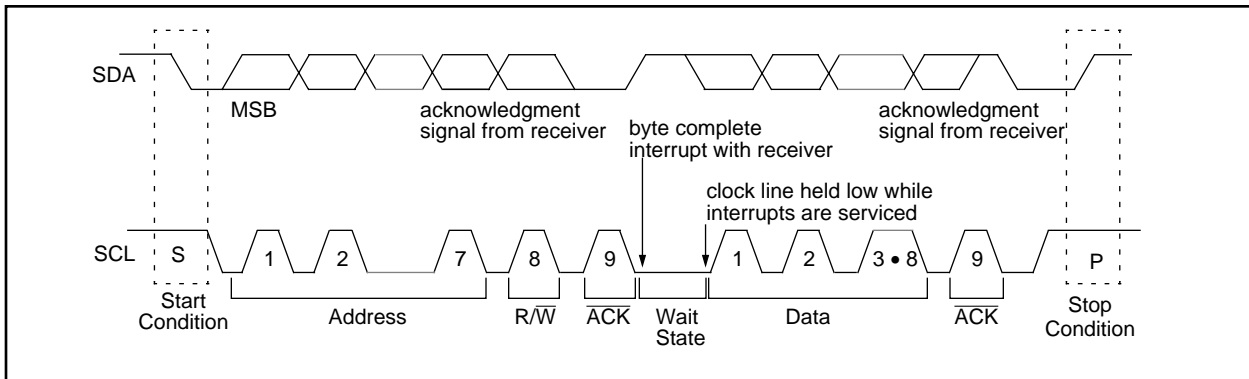
**FIGURE 11-17: SLAVE-RECEIVER ACKNOWLEDGE**



If the master is receiving the data (master-receiver), it generates an acknowledge signal for each received byte of data, except for the last byte. To signal the end of data to the slave-transmitter, the master does not generate an acknowledge (not acknowledge). The slave then releases the SDA line so the master can generate the STOP condition. The master can also generate the STOP condition during the acknowledge pulse for valid termination of data transfer.

If the slave needs to delay the transmission of the next byte, holding the SCL line low will force the master into a wait state. Data transfer continues when the slave releases the SCL line. This allows the slave to move the received data or fetch the data it needs to transfer before allowing the clock to start. This wait state technique can also be implemented at the bit level, Figure 11-18. The slave will inherently stretch the clock, when it is a transmitter, but will not when it is a receiver. The slave will have to clear the SSPCON<4> bit to enable clock stretching when it is a receiver.

**FIGURE 11-18: DATA TRANSFER WAIT STATE**



Applicable Devices						
72	73	73A	74	74A	76	77

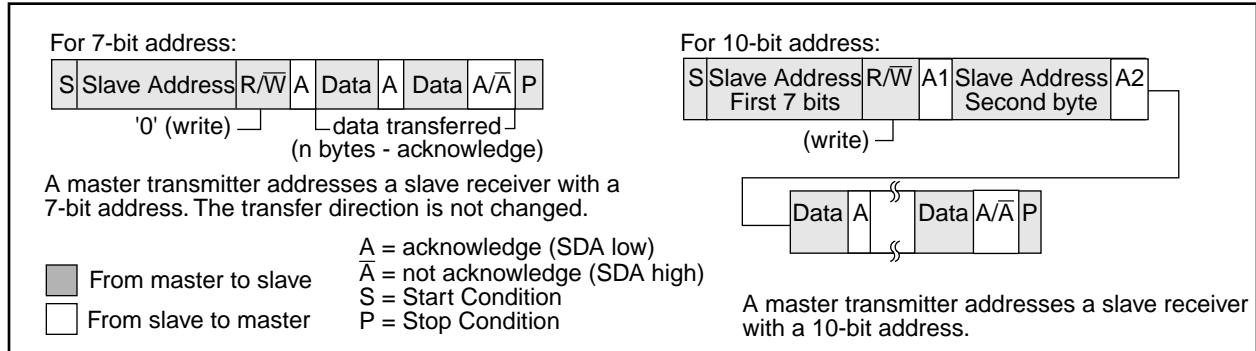
# PIC16C7X

Figure 11-19 and Figure 11-20 show Master-transmitter and Master-receiver data transfer sequences.

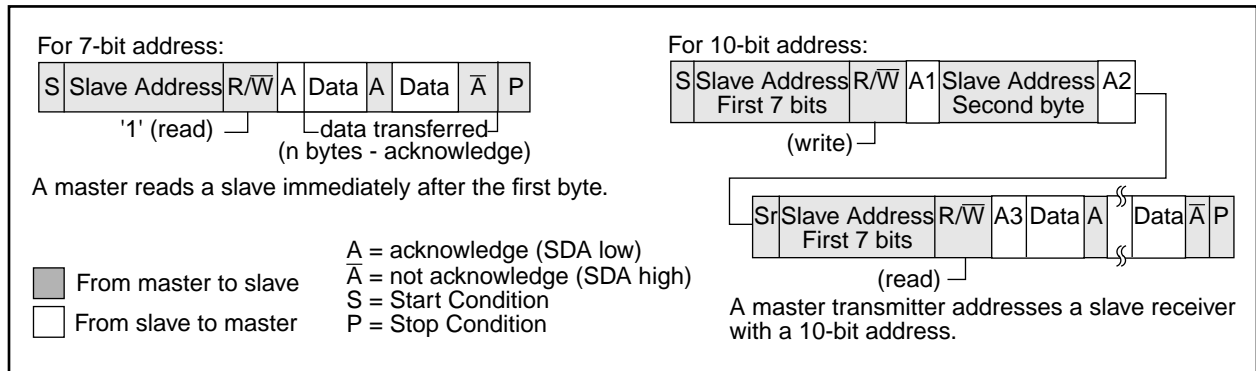
When a master does not wish to relinquish the bus (by generating a STOP condition), a repeated START condition (Sr) must be generated. This condition is identical to the start condition (S) (SDA goes high-to-low while

SCL is high), but occurs after a data transfer acknowledge pulse (not the bus-free state). This allows a master to send "commands" to the slave and then receive the requested information or to address a different slave device. This sequence is shown in Figure 11-21.

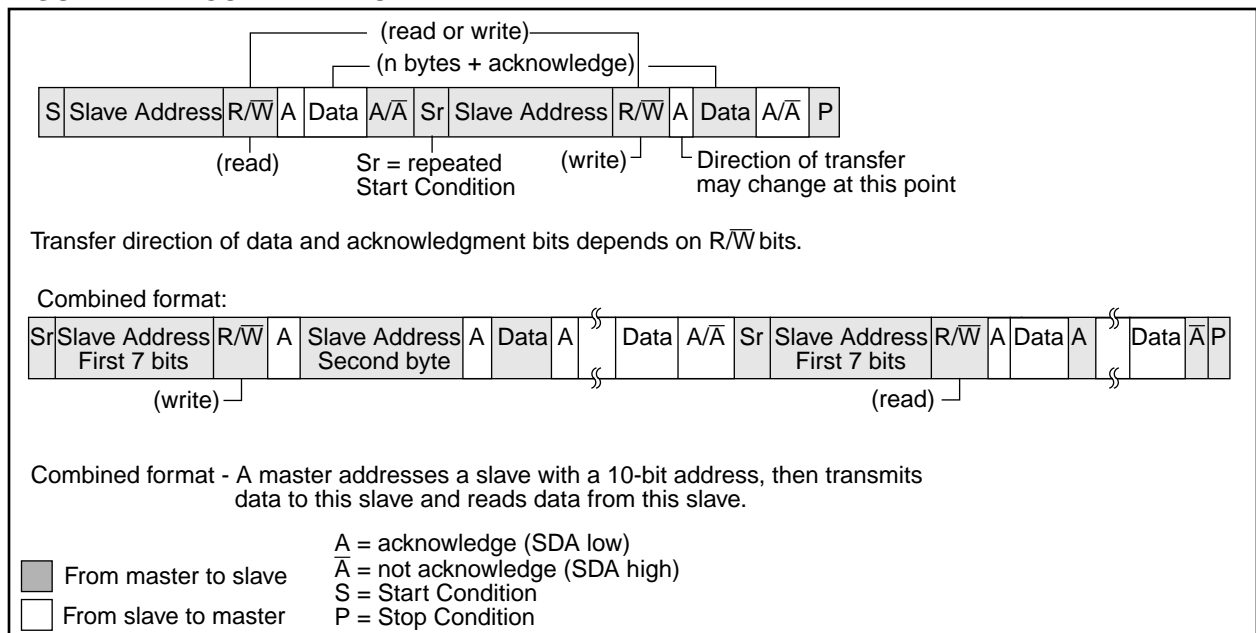
**FIGURE 11-19: MASTER-TRANSMITTER SEQUENCE**



**FIGURE 11-20: MASTER-RECEIVER SEQUENCE**



**FIGURE 11-21: COMBINED FORMAT**



# PIC16C7X

Applicable Devices					
72	73	73A	74	74A	76/77

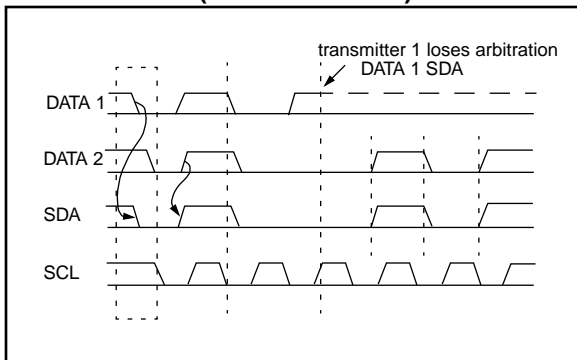
## 11.4.4 MULTI-MASTER

The I<sup>2</sup>C protocol allows a system to have more than one master. This is called multi-master. When two or more masters try to transfer data at the same time, arbitration and synchronization occur.

### 11.4.4.1 ARBITRATION

Arbitration takes place on the SDA line, while the SCL line is high. The master which transmits a high when the other master transmits a low loses arbitration (Figure 11-22), and turns off its data output stage. A master which lost arbitration can generate clock pulses until the end of the data byte where it lost arbitration. When the master devices are addressing the same device, arbitration continues into the data.

**FIGURE 11-22: MULTI-MASTER ARBITRATION (TWO MASTERS)**



Masters that also incorporate the slave function, and have lost arbitration must immediately switch over to slave-receiver mode. This is because the winning master-transmitter may be addressing it.

Arbitration is not allowed between:

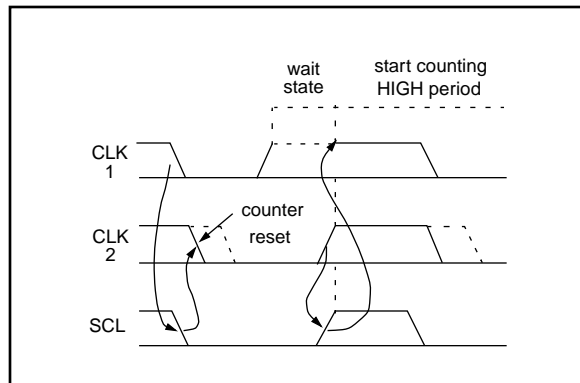
- A repeated START condition
- A STOP condition and a data bit
- A repeated START condition and a STOP condition

Care needs to be taken to ensure that these conditions do not occur.

### 11.2.4.2 Clock Synchronization

Clock synchronization occurs after the devices have started arbitration. This is performed using a wired-AND connection to the SCL line. A high to low transition on the SCL line causes the concerned devices to start counting off their low period. Once a device clock has gone low, it will hold the SCL line low until its SCL high state is reached. The low to high transition of this clock may not change the state of the SCL line, if another device clock is still within its low period. The SCL line is held low by the device with the longest low period. Devices with shorter low periods enter a high wait-state, until the SCL line comes high. When the SCL line comes high, all devices start counting off their high periods. The first device to complete its high period will pull the SCL line low. The SCL line high time is determined by the device with the shortest high period, Figure 11-23.

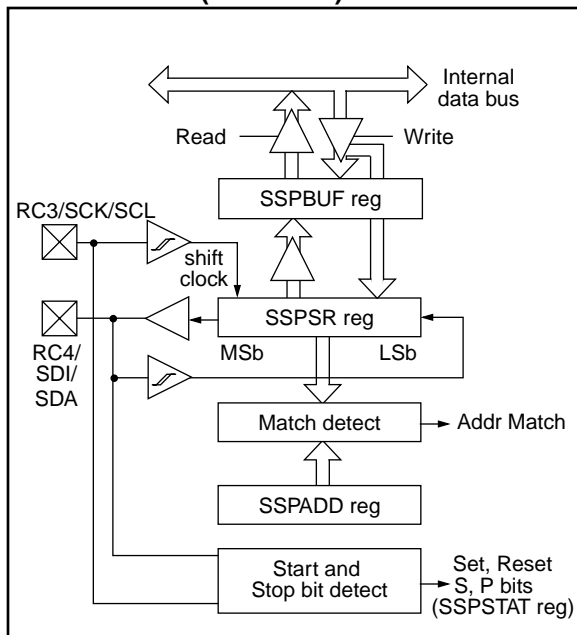
**FIGURE 11-23: CLOCK SYNCHRONIZATION**



## 11.5 SSP I<sup>2</sup>C Operation

The SSP module in I<sup>2</sup>C mode fully implements all slave functions, except general call support, and provides interrupts on start and stop bits in hardware to facilitate firmware implementations of the master functions. The SSP module implements the standard mode specifications as well as 7-bit and 10-bit addressing. Two pins are used for data transfer. These are the 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 11-24: SSP BLOCK DIAGRAM (I<sup>2</sup>C MODE)**



The SSP module has five registers for I<sup>2</sup>C 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 I<sup>2</sup>C operation. Four mode selection bits (SSPCON<3:0>) allow one of the following I<sup>2</sup>C modes to be selected:

- I<sup>2</sup>C Slave mode (7-bit address)
- I<sup>2</sup>C Slave mode (10-bit address)
- I<sup>2</sup>C Slave mode (7-bit address), with start and stop bit interrupts enabled
- I<sup>2</sup>C Slave mode (10-bit address), with start and stop bit interrupts enabled
- I<sup>2</sup>C Firmware controlled Master Mode, slave is idle

Selection of any I<sup>2</sup>C mode, with the SSPEN bit set, forces the SCL and SDA pins to be open drain, provided these pins are programmed to inputs by setting the appropriate TRISC bits.

The SSPSTAT register gives the status of the data transfer. This information includes detection of a START or STOP bit, specifies if the received byte was data or address if the next byte is the completion of 10-bit address, and if this will be a read or write data transfer. The SSPSTAT register is read only.

The SSPBUF is the register to which transfer data is written to or read from. The SSPSR register shifts the data in or out of the device. In receive operations, the SSPBUF and SSPSR create a doubled buffered receiver. This allows reception of the next byte to begin before reading the last byte of received data. When the complete byte is received, it is transferred to the SSPBUF register and flag bit SSIIF is set. If another complete byte is received before the SSPBUF register is read, a receiver overflow has occurred and bit SSPOV (SSPCON<6>) is set and the byte in the SSPSR is lost.

The SSPADD register holds the slave address. In 10-bit mode, the user first needs to write the high byte of the address (1111 0 A9 A8 0). Following the high byte address match, the low byte of the address needs to be loaded (A7:A0).

# PIC16C7X

Applicable Devices					
72	73	73A	74	74A	76/77

## 11.5.1 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. These are if 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 11-4 shows what happens when a data transfer byte is received, given the status of bits BF and SSPOV. The shaded cells show the condition where user software did not properly clear the overflow condition. Flag bit BF is cleared by reading the SSPBUF register while bit SSPOV is cleared through software.

The SCL clock input must have a minimum high and low for proper operation. The high and low times of the I<sup>2</sup>C specification as well as the requirement of the SSP module is shown in timing parameter #100 and parameter #101.

### 11.5.1.1 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  $\overline{ACK}$  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 11-16). The five Most Significant bits (MSbs) of the first address byte specify if this is a 10-bit address. Bit  $R/\overline{W}$  (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 11-4: DATA TRANSFER RECEIVED BYTE ACTIONS**

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

Applicable Devices						
72	73	73A	74	74A	76	77

# PIC16C7X

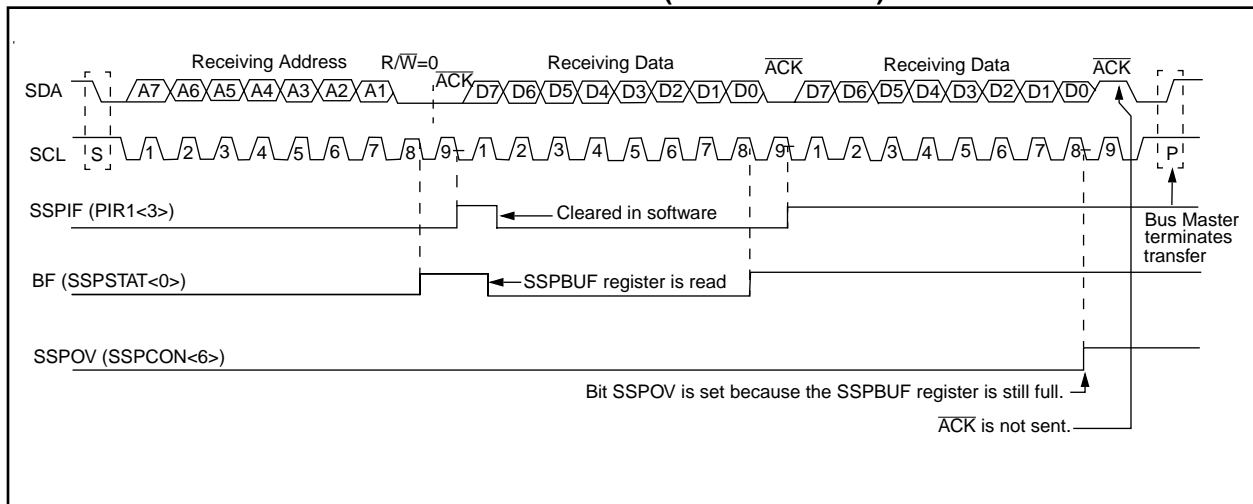
## 11.5.1.2 RECEPTION

When the  $R/\bar{W}$  bit of the address byte is clear and an address match occurs, the  $R/\bar{W}$  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 ( $\bar{ACK}$ ) pulse is given. An overflow condition is defined as either bit BF (SSPSTAT<0>) is set or bit SSPOV (SSPCON<6>) is set.

An 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 11-25: I<sup>2</sup>C WAVEFORMS FOR RECEPTION (7-BIT ADDRESS)**



# PIC16C7X

Applicable Devices						
72	73	73A	74	74A	76	77

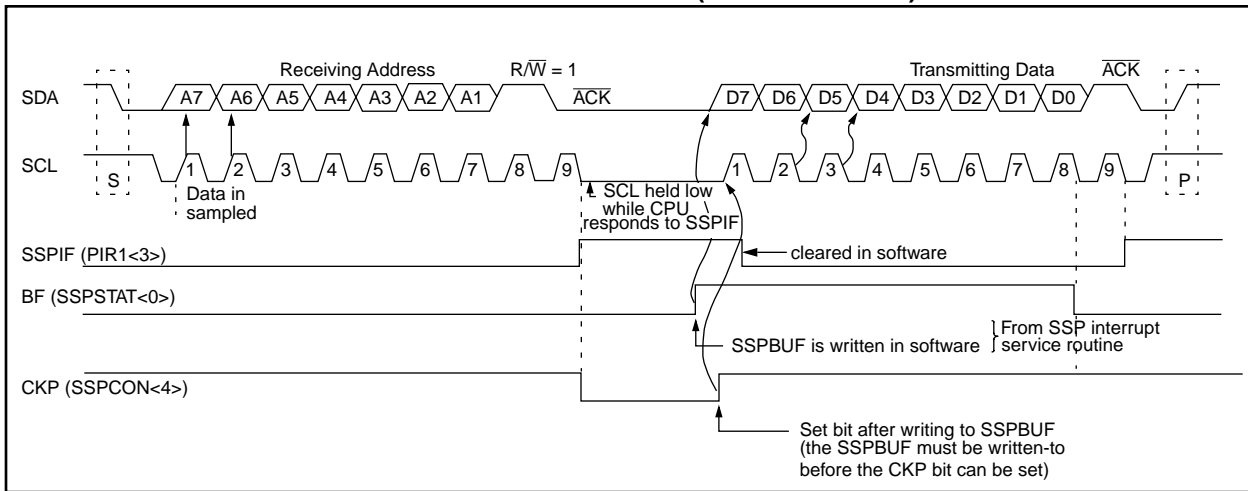
## 11.5.1.3 TRANSMISSION

When the  $R/\bar{W}$  bit of the incoming address byte is set and an address match occurs, the  $R/\bar{W}$  bit of the SSPSTAT register is set. The received address is loaded into the SSPBUF register. The  $\bar{ACK}$  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 11-26).

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  $\bar{ACK}$  pulse from the master-receiver is latched on the rising edge of the ninth SCL input pulse. If the SDA line was high (not  $\bar{ACK}$ ), then the data transfer is complete. When the  $\bar{ACK}$  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 ( $\bar{ACK}$ ), 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 11-26: I<sup>2</sup>C WAVEFORMS FOR TRANSMISSION (7-BIT ADDRESS)**





Applicable Devices					
72	73	73A	74	74A	76/77

# PIC16C7X

## 11.5.2 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 I<sup>2</sup>C bus may be taken when the P bit is set, or the bus is idle and both the S and P bits are clear.

In master mode 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.

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

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

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.

## 11.5.3 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 I<sup>2</sup>C bus may be taken when bit P (SSPSTAT<4>) is set, or the bus is idle and both the S and P bits clear. When the bus is busy, enabling the SSP Interrupt will generate the interrupt when the STOP condition occurs.

In multi-master 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  $\overline{ACK}$  pulse will be generated. If arbitration was lost during the data transfer stage, the device will need to re-transfer the data at a later time.

**TABLE 11-5: REGISTERS ASSOCIATED WITH I<sup>2</sup>C OPERATION**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other resets
0Bh, 8Bh, 10Bh, 18Bh	INTCON	GIE	PEIE	TOIE	INTE	RBIE	T0IF	INTF	RBF	0000 000x	0000 000u
0Ch	PIR1	PSPIF <sup>(1)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
8Ch	PIE1	PSPIE <sup>(1)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
13h	SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register								xxxxx xxxxx	uuuu uuuu
93h	SSPADD	Synchronous Serial Port (I <sup>2</sup> C mode) Address Register								0000 0000	0000 0000
14h	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	0000 0000
94h	SSPSTAT	SMP <sup>(2)</sup>	CKE <sup>(2)</sup>	D/ $\overline{A}$	P	S	R/ $\overline{W}$	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 SPI mode.

Note 1: PSPIF and PSPIE are reserved on the PIC16C73/73A/76, always maintain these bits clear.

Note 2: The SMP and CKE bits are implemented on the PIC16C76/77 only. All other PIC16C7X devices have these two bits unimplemented, read as '0'.

# PIC16C7X

Applicable Devices						
72	73	73A	74	74A	76	77

FIGURE 11-27: OPERATION OF THE I<sup>2</sup>C MODULE IN IDLE\_MODE, RCV\_MODE OR XMIT\_MODE

<pre> <b>IDLE_MODE (7-bit):</b> if (Addr_match)          {      Set interrupt;                            if (R/W = 1)  {      Send <math>\overline{ACK}</math> = 0;  set XMIT_MODE;  }                            else if (R/W = 0) set RCV_MODE;                            } </pre>
<pre> <b>RCV_MODE:</b> if ((SSPBUF=Full) OR (SSPOV = 1))     {      Set SSPOV;           Do not acknowledge;     } else     {      transfer SSPSR → SSPBUF;           send <math>\overline{ACK}</math> = 0;     } Receive 8-bits in SSPSR; Set interrupt; </pre>
<pre> <b>XMIT_MODE:</b> While ((SSPBUF = Empty) AND (CKP=0)) Hold SCL Low; Send byte; Set interrupt; if ( <math>\overline{ACK}</math> Received = 1)      {      End of transmission;                               Go back to IDLE_MODE;                               } else if ( <math>\overline{ACK}</math> Received = 0) Go back to XMIT_MODE; </pre>
<pre> <b>IDLE_MODE (10-Bit):</b> If (High_byte_addr_match AND (R/W = 0))     {      PRIOR_ADDR_MATCH = FALSE;           Set interrupt;           if ((SSPBUF = Full) OR ((SSPOV = 1))               {      Set SSPOV;                     Do not acknowledge;               }           else     {      Set UA = 1;                     Send <math>\overline{ACK}</math> = 0;                     While (SSPADD not updated) Hold SCL low;                     Clear UA = 0;                     Receive Low_addr_byte;                     Set interrupt;                     Set UA = 1;                     If (Low_byte_addr_match)                         {      PRIOR_ADDR_MATCH = TRUE;                               Send <math>\overline{ACK}</math> = 0;                               while (SSPADD not updated) Hold SCL low;                               Clear UA = 0;                               Set RCV_MODE;                               }                         }           } else if (High_byte_addr_match AND (R/W = 1))     {      if (PRIOR_ADDR_MATCH)           {      send <math>\overline{ACK}</math> = 0;                 set XMIT_MODE;           }           else PRIOR_ADDR_MATCH = FALSE;     } </pre>

## 12.0 UNIVERSAL SYNCHRONOUS ASYNCHRONOUS RECEIVER TRANSMITTER (USART)

Applicable Devices					
72	73	73A	74	74A	76/77

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.

**FIGURE 12-1: 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
bit7							bit0

R = Readable bit  
W = Writable bit  
U = Unimplemented bit, read as '0'  
- n = Value at POR reset

bit 7: **CSRC:** Clock Source Select bit

Asynchronous mode  
Don't care

Synchronous mode  
1 = Master mode (Clock generated internally from BRG)  
0 = Slave mode (Clock from external source)

bit 6: **TX9:** 9-bit Transmit Enable bit  
1 = Selects 9-bit transmission  
0 = Selects 8-bit transmission

bit 5: **TXEN:** Transmit Enable bit  
1 = Transmit enabled  
0 = Transmit disabled  
Note: SREN/CREN overrides TXEN in SYNC mode.

bit 4: **SYNC:** USART Mode Select bit  
1 = Synchronous mode  
0 = Asynchronous mode

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

bit 2: **BRGH:** High Baud Rate Select bit

Asynchronous mode  
1 = High speed

**Note:** For the PIC16C73/73A/74/74A, the asynchronous high speed mode (BRGH = 1) may experience a high rate of receive errors. It is recommended that BRGH = 0. If you desire a higher baud rate than BRGH = 0 can support, refer to the device errata for additional information, or use the PIC16C76/77.

0 = Low speed

Synchronous mode  
Unused in this mode

bit 1: **TRMT:** Transmit Shift Register Status bit  
1 = TSR empty  
0 = TSR full

bit 0: **TX9D:** 9th bit of transmit data. Can be parity bit.

# PIC16C7X

**FIGURE 12-2: 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
							bit0
bit7							

R = Readable bit  
W = Writable bit  
U = Unimplemented bit, read as '0'  
- n = Value at POR reset

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  
Unused in this mode

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)

## 12.1 USART Baud Rate Generator (BRG)

Applicable Devices				
72	73	73A	74	74A
76	77			

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 12-1 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 12-1. From this, the error in baud rate can be determined.

Example 12-1 shows the calculation of the baud rate error for the following conditions:

Fosc = 16 MHz  
 Desired Baud Rate = 9600  
 BRGH = 0  
 SYNC = 0

### EXAMPLE 12-1: CALCULATING BAUD RATE ERROR

$$\begin{aligned} \text{Desired Baud rate} &= F_{osc} / (64 (X + 1)) \\ 9600 &= 16000000 / (64 (X + 1)) \\ X &= \lfloor 25.042 \rfloor = 25 \\ \text{Calculated Baud Rate} &= 16000000 / (64 (25 + 1)) \\ &= 9615 \\ \text{Error} &= \frac{(\text{Calculated Baud Rate} - \text{Desired Baud Rate})}{\text{Desired Baud Rate}} \\ &= (9615 - 9600) / 9600 \\ &= 0.16\% \end{aligned}$$

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.

**Note:** For the PIC16C73/73A/74/74A, the asynchronous high speed mode (BRGH = 1) may experience a high rate of receive errors. It is recommended that BRGH = 0. If you desire a higher baud rate than BRGH = 0 can support, refer to the device errata for additional information, or use the PIC16C76/77.

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.

**TABLE 12-1: 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 = $F_{osc}/(4(X+1))$	NA

X = value in SPBRG (0 to 255)

**TABLE 12-2: 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.

# PIC16C7X

**TABLE 12-3: BAUD RATES FOR SYNCHRONOUS MODE**

BAUD RATE (K)	FOSC = 20 MHz			16 MHz			10 MHz			7.15909 MHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	NA	-	-	NA	-	-	NA	-	-	NA	-	-
1.2	NA	-	-	NA	-	-	NA	-	-	NA	-	-
2.4	NA	-	-	NA	-	-	NA	-	-	NA	-	-
9.6	NA	-	-	NA	-	-	9.766	+1.73	255	9.622	+0.23	185
19.2	19.53	+1.73	255	19.23	+0.16	207	19.23	+0.16	129	19.24	+0.23	92
76.8	76.92	+0.16	64	76.92	+0.16	51	75.76	-1.36	32	77.82	+1.32	22
96	96.15	+0.16	51	95.24	-0.79	41	96.15	+0.16	25	94.20	-1.88	18
300	294.1	-1.96	16	307.69	+2.56	12	312.5	+4.17	7	298.3	-0.57	5
500	500	0	9	500	0	7	500	0	4	NA	-	-
HIGH	5000	-	0	4000	-	0	2500	-	0	1789.8	-	0
LOW	19.53	-	255	15.625	-	255	9.766	-	255	6.991	-	255

BAUD RATE (K)	FOSC = 5.0688 MHz			4 MHz			3.579545 MHz			1 MHz			32.768 kHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	NA	-	-	NA	-	-	NA	-	-	NA	-	-	0.303	+1.14	26
1.2	NA	-	-	NA	-	-	NA	-	-	1.202	+0.16	207	1.170	-2.48	6
2.4	NA	-	-	NA	-	-	NA	-	-	2.404	+0.16	103	NA	-	-
9.6	9.6	0	131	9.615	+0.16	103	9.622	+0.23	92	9.615	+0.16	25	NA	-	-
19.2	19.2	0	65	19.231	+0.16	51	19.04	-0.83	46	19.24	+0.16	12	NA	-	-
76.8	79.2	+3.13	15	76.923	+0.16	12	74.57	-2.90	11	83.34	+8.51	2	NA	-	-
96	97.48	+1.54	12	1000	+4.17	9	99.43	+3.57	8	NA	-	-	NA	-	-
300	316.8	+5.60	3	NA	-	-	298.3	-0.57	2	NA	-	-	NA	-	-
500	NA	-	-	NA	-	-	NA	-	-	NA	-	-	NA	-	-
HIGH	1267	-	0	100	-	0	894.9	-	0	250	-	0	8.192	-	0
LOW	4.950	-	255	3.906	-	255	3.496	-	255	0.9766	-	255	0.032	-	255

**TABLE 12-4: BAUD RATES FOR ASYNCHRONOUS MODE (BRGH = 0)**

BAUD RATE (K)	FOSC = 20 MHz			16 MHz			10 MHz			7.15909 MHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	NA	-	-	NA	-	-	NA	-	-	NA	-	-
1.2	1.221	+1.73	255	1.202	+0.16	207	1.202	+0.16	129	1.203	+0.23	92
2.4	2.404	+0.16	129	2.404	+0.16	103	2.404	+0.16	64	2.380	-0.83	46
9.6	9.469	-1.36	32	9.615	+0.16	25	9.766	+1.73	15	9.322	-2.90	11
19.2	19.53	+1.73	15	19.23	+0.16	12	19.53	+1.73	7	18.64	-2.90	5
76.8	78.13	+1.73	3	83.33	+8.51	2	78.13	+1.73	1	NA	-	-
96	104.2	+8.51	2	NA	-	-	NA	-	-	NA	-	-
300	312.5	+4.17	0	NA	-	-	NA	-	-	NA	-	-
500	NA	-	-	NA	-	-	NA	-	-	NA	-	-
HIGH	312.5	-	0	250	-	0	156.3	-	0	111.9	-	0
LOW	1.221	-	255	0.977	-	255	0.6104	-	255	0.437	-	255

BAUD RATE (K)	FOSC = 5.0688 MHz			4 MHz			3.579545 MHz			1 MHz			32.768 kHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	0.31	+3.13	255	0.3005	-0.17	207	0.301	+0.23	185	0.300	+0.16	51	0.256	-14.67	1
1.2	1.2	0	65	1.202	+1.67	51	1.190	-0.83	46	1.202	+0.16	12	NA	-	-
2.4	2.4	0	32	2.404	+1.67	25	2.432	+1.32	22	2.232	-6.99	6	NA	-	-
9.6	9.9	+3.13	7	NA	-	-	9.322	-2.90	5	NA	-	-	NA	-	-
19.2	19.8	+3.13	3	NA	-	-	18.64	-2.90	2	NA	-	-	NA	-	-
76.8	79.2	+3.13	0	NA	-	-	NA	-	-	NA	-	-	NA	-	-
96	NA	-	-	NA	-	-	NA	-	-	NA	-	-	NA	-	-
300	NA	-	-	NA	-	-	NA	-	-	NA	-	-	NA	-	-
500	NA	-	-	NA	-	-	NA	-	-	NA	-	-	NA	-	-
HIGH	79.2	-	0	62.500	-	0	55.93	-	0	15.63	-	0	0.512	-	0
LOW	0.3094	-	255	3.906	-	255	0.2185	-	255	0.0610	-	255	0.0020	-	255

# PIC16C7X

**TABLE 12-5: BAUD RATES FOR ASYNCHRONOUS MODE (BRGH = 1)**

BAUD RATE (K)	FOSC = 20 MHz			16 MHz			10 MHz			7.16 MHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
9.6	9.615	+0.16	129	9.615	+0.16	103	9.615	+0.16	64	9.520	-0.83	46
19.2	19.230	+0.16	64	19.230	+0.16	51	18.939	-1.36	32	19.454	+1.32	22
38.4	37.878	-1.36	32	38.461	+0.16	25	39.062	+1.7	15	37.286	-2.90	11
57.6	56.818	-1.36	21	58.823	+2.12	16	56.818	-1.36	10	55.930	-2.90	7
115.2	113.636	-1.36	10	111.111	-3.55	8	125	+8.51	4	111.860	-2.90	3
250	250	0	4	250	0	3	NA	-	-	NA	-	-
625	625	0	1	NA	-	-	625	0	0	NA	-	-
1250	1250	0	0	NA	-	-	NA	-	-	NA	-	-

BAUD RATE (K)	FOSC = 5.068 MHz			4 MHz			3.579 MHz			1 MHz			32.768 kHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
9.6	9.6	0	32	NA	-	-	9.727	+1.32	22	8.928	-6.99	6	NA	-	-
19.2	18.645	-2.94	16	1.202	+0.17	207	18.643	-2.90	11	20.833	+8.51	2	NA	-	-
38.4	39.6	+3.12	7	2.403	+0.13	103	37.286	-2.90	5	31.25	-18.61	1	NA	-	-
57.6	52.8	-8.33	5	9.615	+0.16	25	55.930	-2.90	3	62.5	+8.51	0	NA	-	-
115.2	105.6	-8.33	2	19.231	+0.16	12	111.860	-2.90	1	NA	-	-	NA	-	-
250	NA	-	-	NA	-	-	223.721	-10.51	0	NA	-	-	NA	-	-
625	NA	-	-	NA	-	-	NA	-	-	NA	-	-	NA	-	-
1250	NA	-	-	NA	-	-	NA	-	-	NA	-	-	NA	-	-

**Note:** For the PIC16C73/73A/74/74A, the asynchronous high speed mode (BRGH = 1) may experience a high rate of receive errors. It is recommended that BRGH = 0. If you desire a higher baud rate than BRGH = 0 can support, refer to the device errata for additional information, or use the PIC16C76/77.

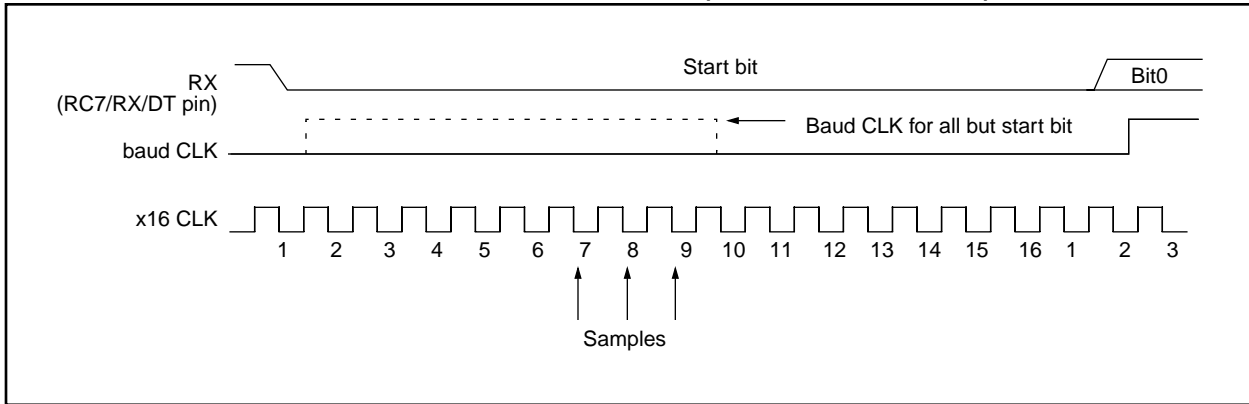
# PIC16C7X

## 12.1.1 SAMPLING

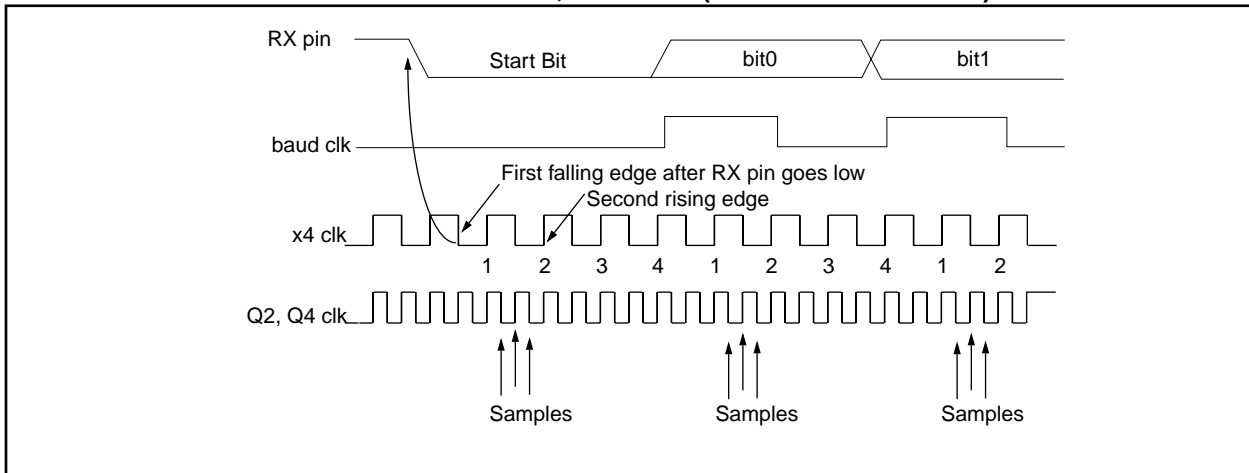
The data on the RC7/RX/DT pin is sampled three times by a majority detect circuit to determine if a high or a low level is present at the RX pin. If bit BRGH (TXSTA<2>) is clear (i.e., at the low baud rates), the sampling is done on the seventh, eighth and ninth falling edges of a x16 clock (Figure 12-3). If bit BRGH is

set (i.e., at the high baud rates), the sampling is done on the 3 clock edges preceding the second rising edge after the first falling edge of a x4 clock (Figure 12-4 and Figure 12-5).

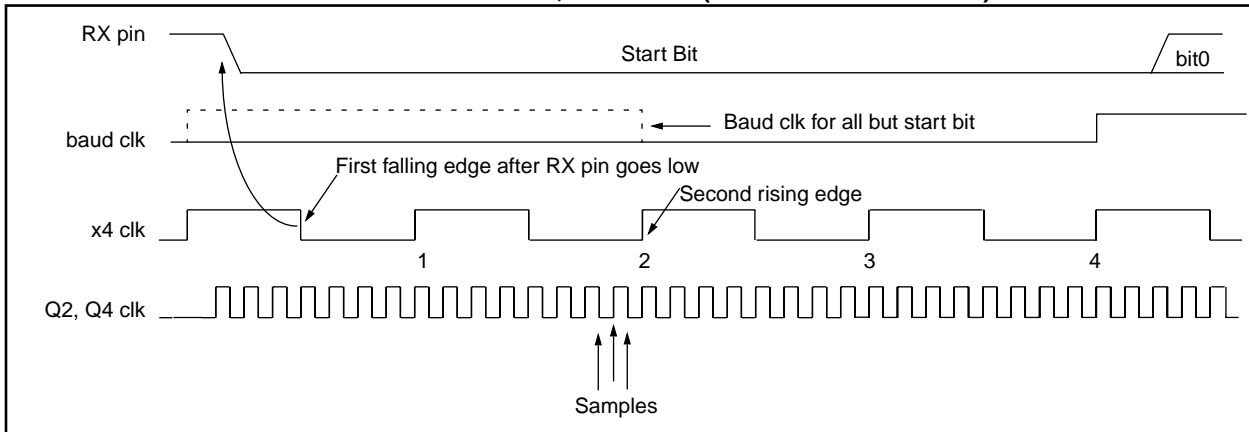
**FIGURE 12-3: RX PIN SAMPLING SCHEME. BRGH = 0 (PIC16C73/73A/74/74A)**



**FIGURE 12-4: RX PIN SAMPLING SCHEME, BRGH = 1 (PIC16C73/73A/74/74A)**



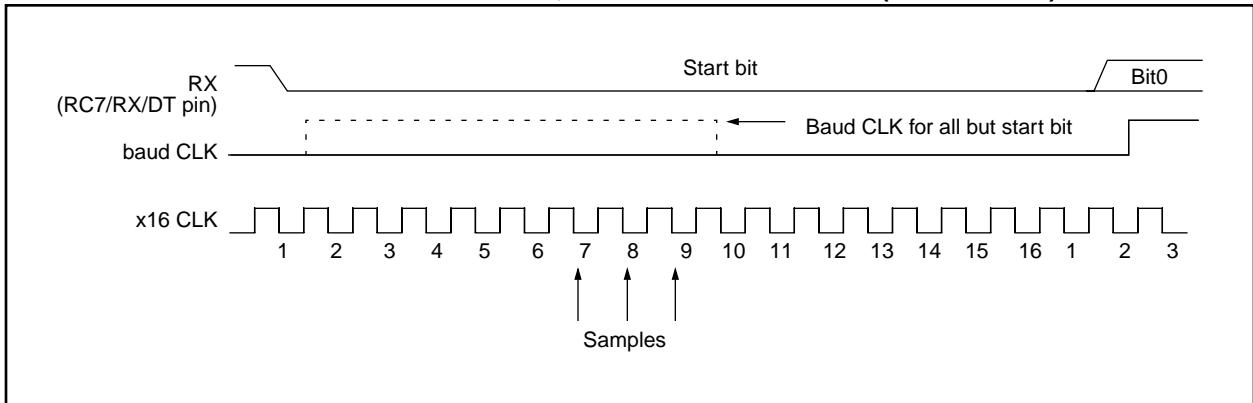
**FIGURE 12-5: RX PIN SAMPLING SCHEME, BRGH = 1 (PIC16C73/73A/74/74A)**





# PIC16C7X

**FIGURE 12-6: RX PIN SAMPLING SCHEME, BRGH = 0 OR BRGH = 1 (PIC16C76/77)**



# PIC16C7X

## 12.2 USART Asynchronous Mode

Applicable Devices				
72	73	73A	74	74A
76	77			

In this mode, the USART uses standard nonreturn-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

### 12.2.1 USART ASYNCHRONOUS TRANSMITTER

The USART transmitter block diagram is shown in Figure 12-7. 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 in software. The TSR register is not loaded until the STOP bit has been transmitted from the previous load. As soon as the STOP bit is transmitted, the TSR is loaded with new data from the TXREG register (if available). Once the TXREG register transfers the data to the TSR register (occurs in one T<sub>cy</sub>), the TXREG register is empty and

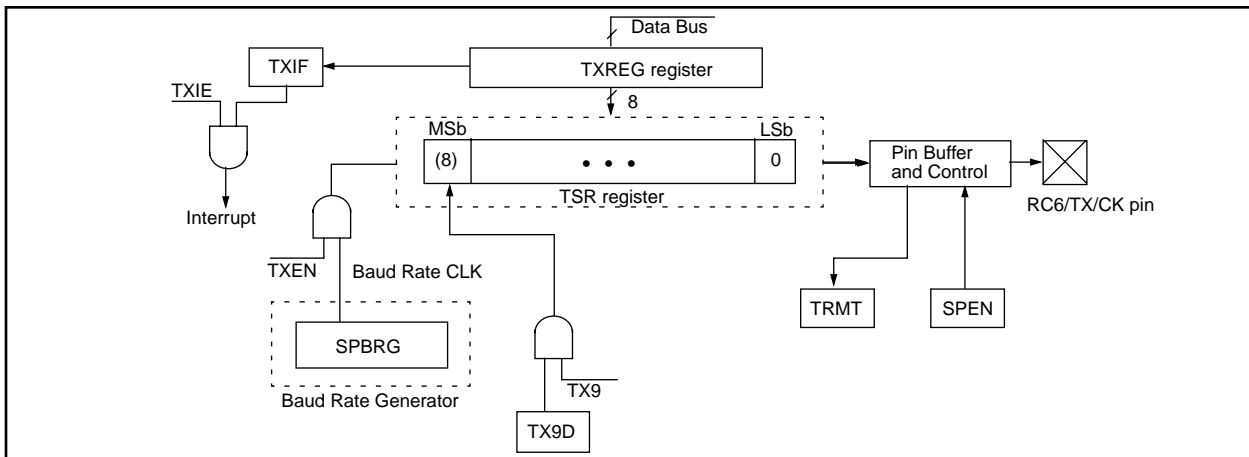
flag bit TXIF (PIR1<4>) is set. This interrupt can be enabled/disabled by setting/clearing enable bit TXIE (PIE1<4>). Flag bit TXIF will be set regardless of the state of enable bit TXIE and cannot be cleared in software. It will reset only when new data is loaded into the TXREG register. While flag bit TXIF indicated the status of the TXREG register, another bit TRMT (TXSTA<1>) shows the status of the TSR register. Status bit TRMT is a read only bit which is set when the TSR register is empty. No interrupt logic is tied to this bit, so the user has to poll this bit in order to determine if the TSR register is empty.

**Note 1:** The TSR register is not mapped in data memory so it is not available to the user.  
**Note 2:** Flag bit TXIF is set when enable bit TXEN is set.

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 12-7). 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, so a 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 12-9). 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 maybe loaded in the TSR register.

**FIGURE 12-7: USART TRANSMIT BLOCK DIAGRAM**

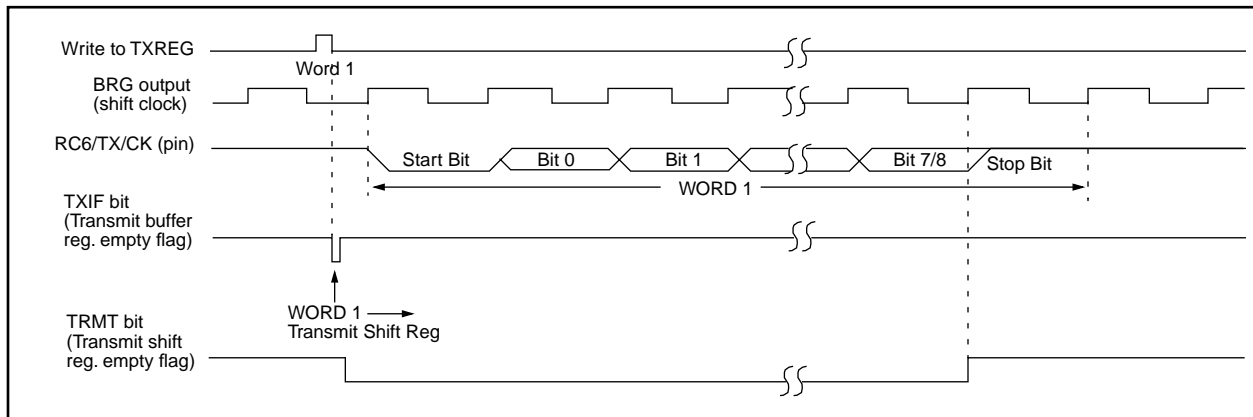


# PIC16C7X

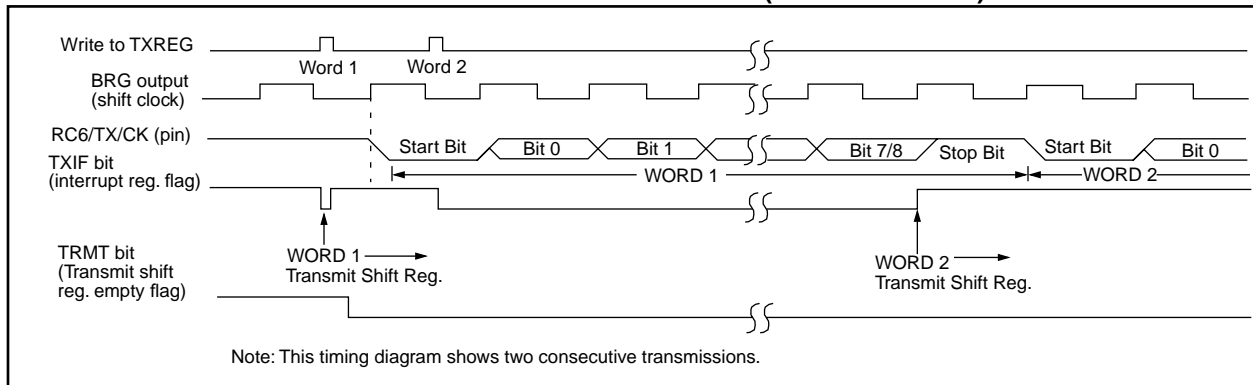
Steps to follow when setting up an Asynchronous Transmission:

1. Initialize the SPBRG register for the appropriate baud rate. If a high speed baud rate is desired, set bit BRGH. (Section 12.1)
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).

**FIGURE 12-8: ASYNCHRONOUS MASTER TRANSMISSION**



**FIGURE 12-9: ASYNCHRONOUS MASTER TRANSMISSION (BACK TO BACK)**



**TABLE 12-6: 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
0Ch	PIR1	PSPIF <sup>(1)</sup>	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 <sup>(1)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

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

Note 1: Bits PSPIE and PSPIF are reserved on the PIC16C73/73A/76, always maintain these bits clear.

# PIC16C7X

## 12.2.2 USART ASYNCHRONOUS RECEIVER

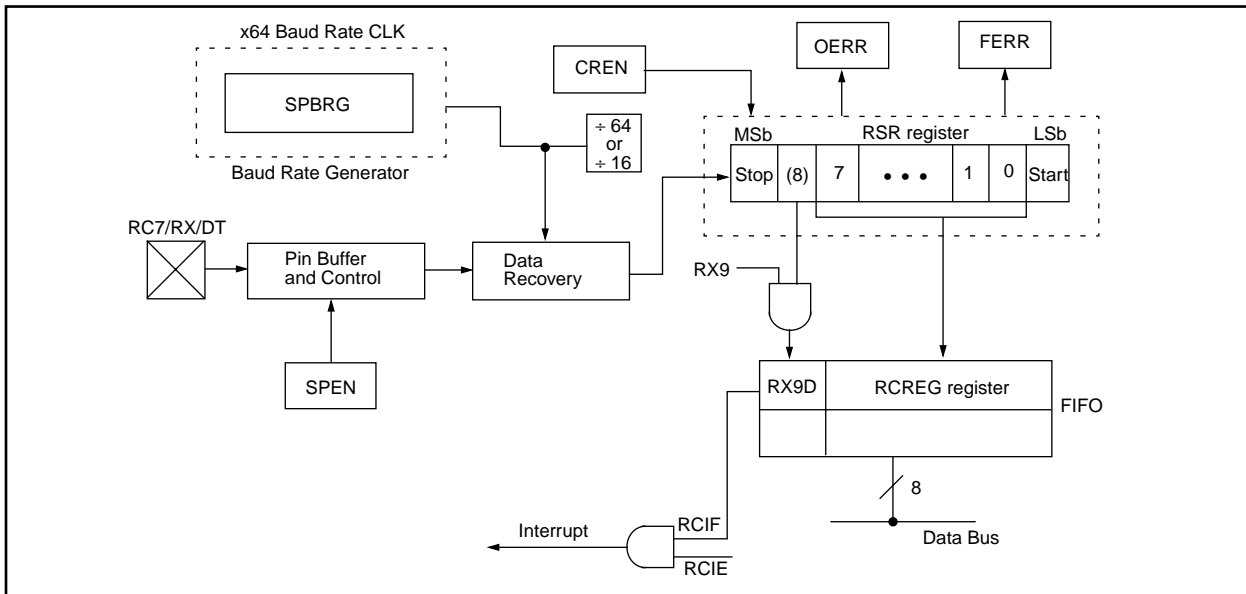
The receiver block diagram is shown in Figure 12-10. The data is received on the RC7/RX/DT pin and drives the data recovery block. The data recovery block is actually a high speed shifter operating at x16 times the baud rate, whereas the main receive serial shifter operates at the bit rate or at FOSC.

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

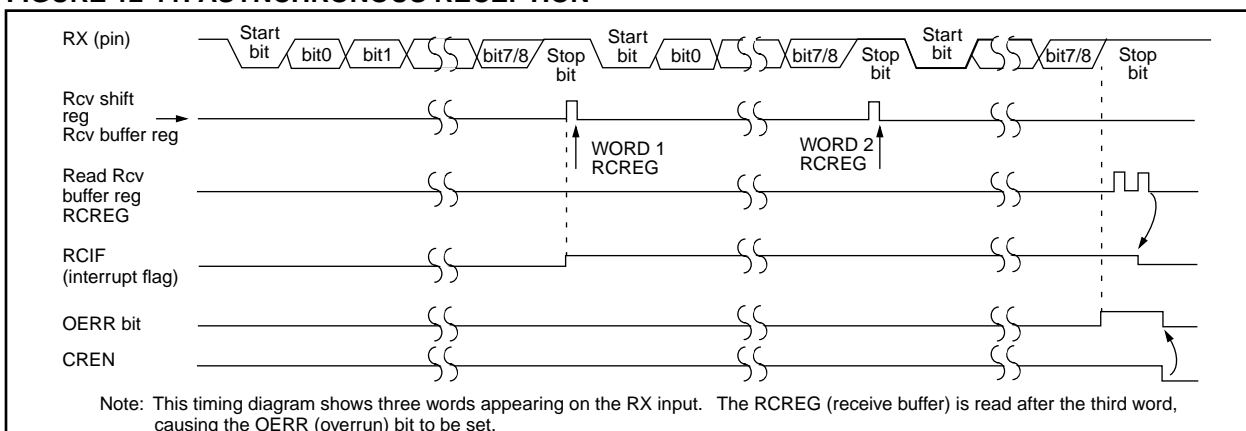
The heart of the receiver is the receive (serial) shift register (RSR). After sampling the STOP bit, the received data in the RSR is transferred to the RCREG register (if it is empty). If the transfer is complete, flag bit RCIF (PIR1<5>) is set. The actual interrupt can be enabled/disabled by setting/clearing 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 begin shifting to the RSR register. On the detection of the STOP bit of the third byte, if the RCREG register is still full then 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, so 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 12-10: USART RECEIVE BLOCK DIAGRAM**



**FIGURE 12-11: ASYNCHRONOUS RECEPTION**



# PIC16C7X

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 12.1).
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 was 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.

**TABLE 12-7: 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
0Ch	PIR1	PSPIF <sup>(1)</sup>	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 <sup>(1)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

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

Note 1: Bits PSPIE and PSPIF are reserved on the PIC16C73/73A/76, always maintain these bits clear.

# PIC16C7X

## 12.3 USART Synchronous Master Mode

Applicable Devices					
72	73	73A	74	74A	76/77

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>).

### 12.3.1 USART SYNCHRONOUS MASTER TRANSMISSION

The USART transmitter block diagram is shown in Figure 12-7. 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 12-12). The transmission can also be started by first loading the TXREG register and then setting bit TXEN (Figure 12-13). 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 12.1).
2. Enable the synchronous master serial port by setting bits SYNC, SPEN, and CSRC.
3. If interrupts are desired, then set enable bit TXIE.
4. If 9-bit transmission is desired, then 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.

# PIC16C7X

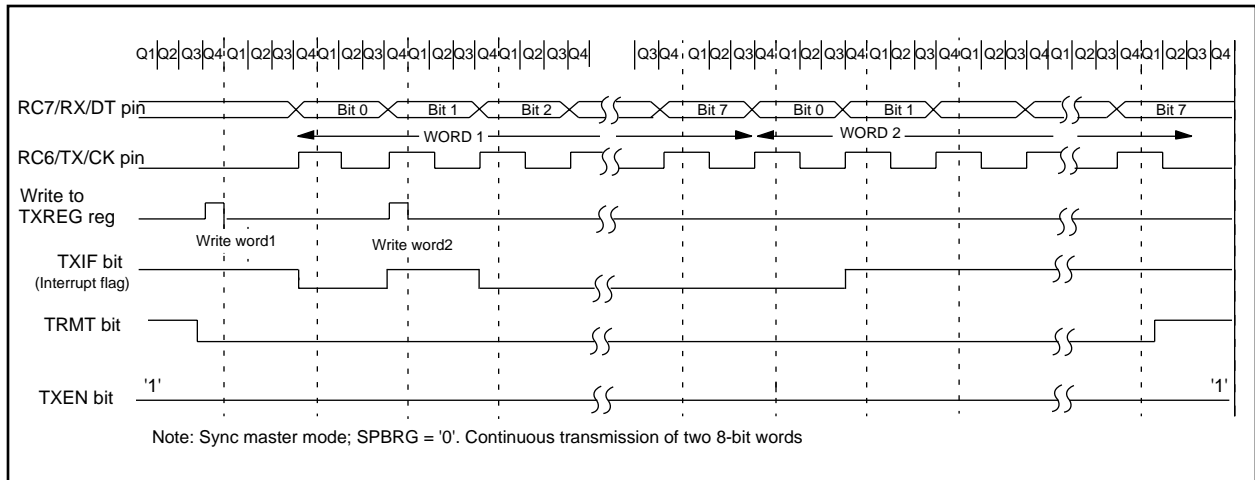
**TABLE 12-8: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER TRANSMISSION**

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
0Ch	PIR1	PSPIF <sup>(1)</sup>	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 <sup>(1)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

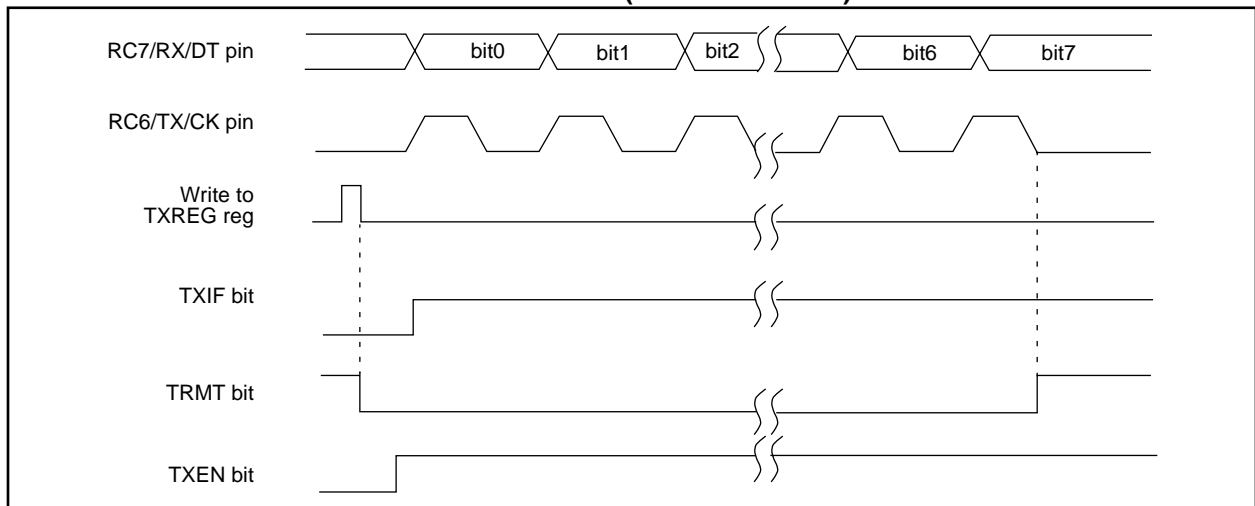
Legend: x = unknown, - = unimplemented, read as '0'. Shaded cells are not used for Synchronous Master Transmission.

Note 1: Bits PSPIE and PSPIF are reserved on the PIC16C73/73A/76, always maintain these bits clear.

**FIGURE 12-12: SYNCHRONOUS TRANSMISSION**



**FIGURE 12-13: SYNCHRONOUS TRANSMISSION (THROUGH TXEN)**



# PIC16C7X

## 12.3.2 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 then 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 9th 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 12.1)
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.

**TABLE 12-9: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER RECEPTION**

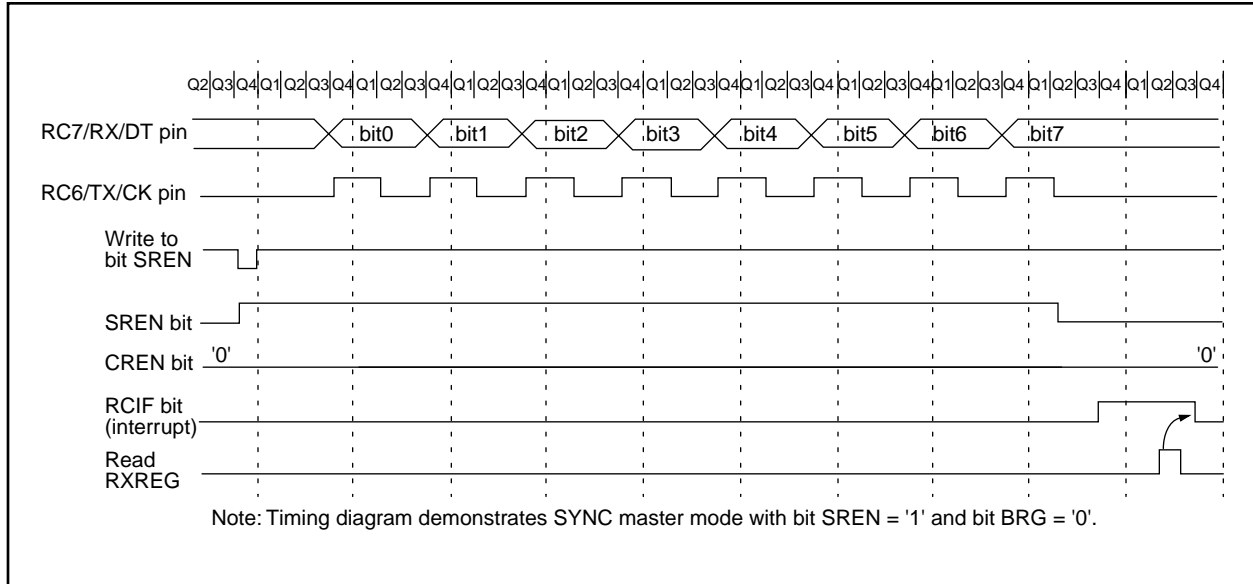
Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other Resets
0Ch	PIR1	PSPIF <sup>(1)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
18h	RCSTA	SPEN	RX9	SREN	CREN	—	FERR	OERR	RX9D	0000 -00x	0000 -00x
1Ah	RCREG	USART Receive Register								0000 0000	0000 0000
8Ch	PIE1	PSPIE <sup>(1)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

Legend: x = unknown, - = unimplemented read as '0'. Shaded cells are not used for Synchronous Master Reception.

Note 1: Bits PSPIE and PSPIF are reserved on the PIC16C73/73A/76, always maintain these bits clear.



**FIGURE 12-14: SYNCHRONOUS RECEPTION (MASTER MODE, SREN)**



# PIC16C7X

---

## 12.4 USART Synchronous Slave Mode

Applicable Devices					
72	73	73A	74	74A	76/77

Synchronous slave mode differs from the Master mode in the fact that the shift clock is supplied externally at the RC6/TX/CK pin (instead of being supplied internally in master mode). This allows the device to transfer or receive data while in SLEEP mode. Slave mode is entered by clearing bit CSRC (TXSTA<7>).

### 12.4.1 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.
- 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).

Steps to follow 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.

### 12.4.2 USART SYNCHRONOUS SLAVE RECEPTION

The operation of the synchronous master and slave modes is identical except in the case of the SLEEP mode. Also, 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).

Steps to follow when setting up a Synchronous Slave Reception:

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

# PIC16C7X

**TABLE 12-10: 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
0Ch	PIR1	PSPIF <sup>(1)</sup>	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 <sup>(1)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

Legend: x = unknown, - = unimplemented read as '0'. Shaded cells are not used for Synchronous Slave Transmission.

Note 1: Bits PSPIE and PSPIF are reserved on the PIC16C73/73A/76, always maintain these bits clear.

**TABLE 12-11: 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
0Ch	PIR1	PSPIF <sup>(1)</sup>	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 <sup>(1)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

Legend: x = unknown, - = unimplemented read as '0'. Shaded cells are not used for Synchronous Slave Reception.

Note 1: Bits PSPIE and PSPIF are reserved on the PIC16C73/73A/76, always maintain these bits clear.

# PIC16C7X

---

NOTES:

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

Applicable Devices						
72	73	73A	74	74A	76	77

The analog-to-digital (A/D) converter module has five inputs for the PIC16C72/73/73A/76, and eight for the PIC16C74/74A/77.

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

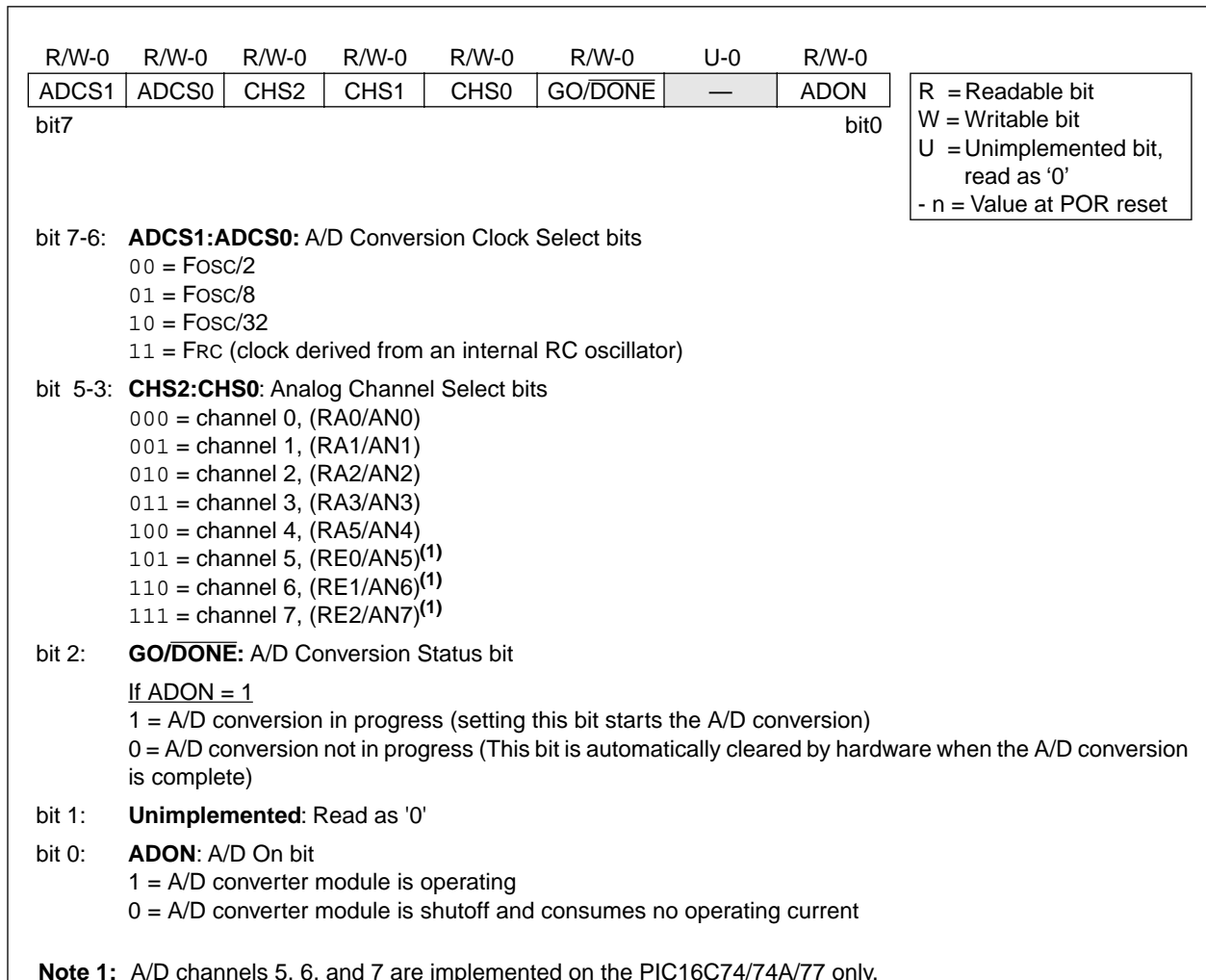
The A/D converter has a unique feature of being able to operate while the device is in SLEEP mode. To operate in sleep, the A/D conversion clock must be derived from the A/D's internal RC oscillator.

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)

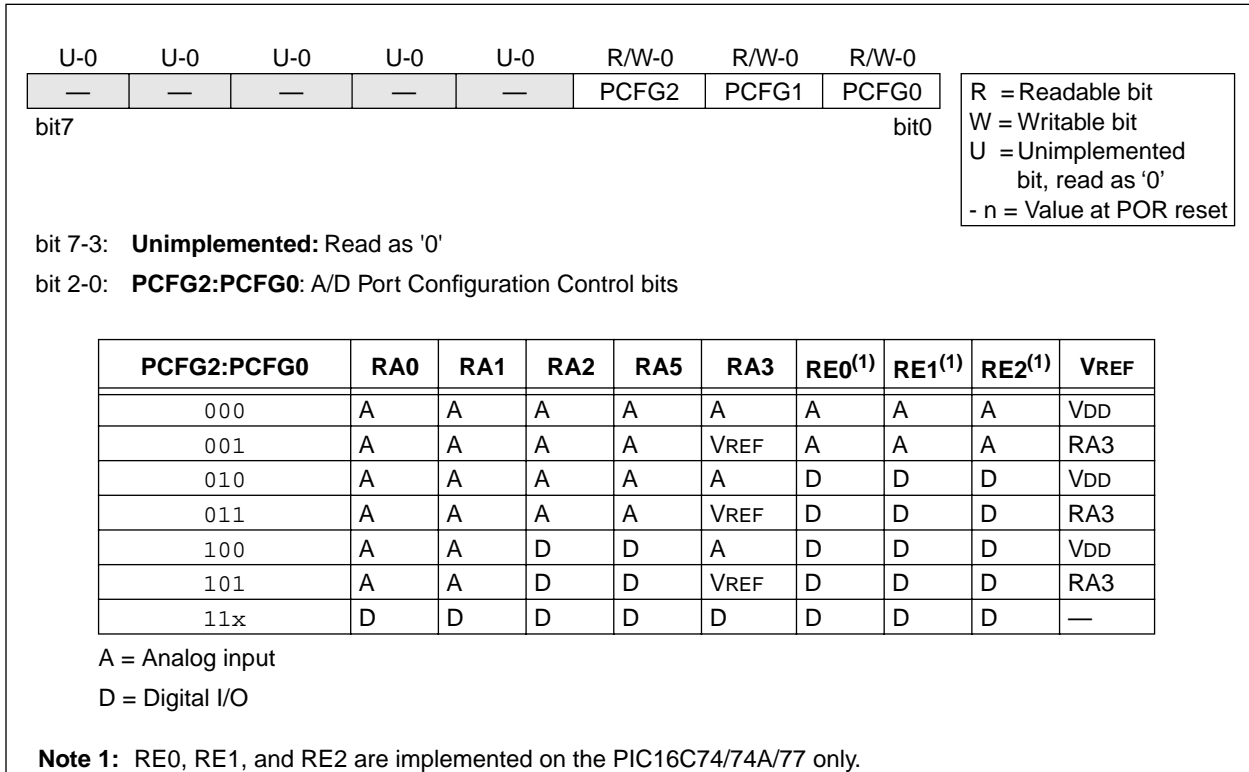
The ADCON0 register, shown in Figure 13-1, controls the operation of the A/D module. The ADCON1 register, shown in Figure 13-2, configures the functions of the port pins. The port pins can be configured as analog inputs (RA3 can also be a voltage reference) or as digital I/O.

**FIGURE 13-1: ADCON0 REGISTER (ADDRESS 1Fh)**



# PIC16C7X

**FIGURE 13-2: ADCON1 REGISTER (ADDRESS 9Fh)**



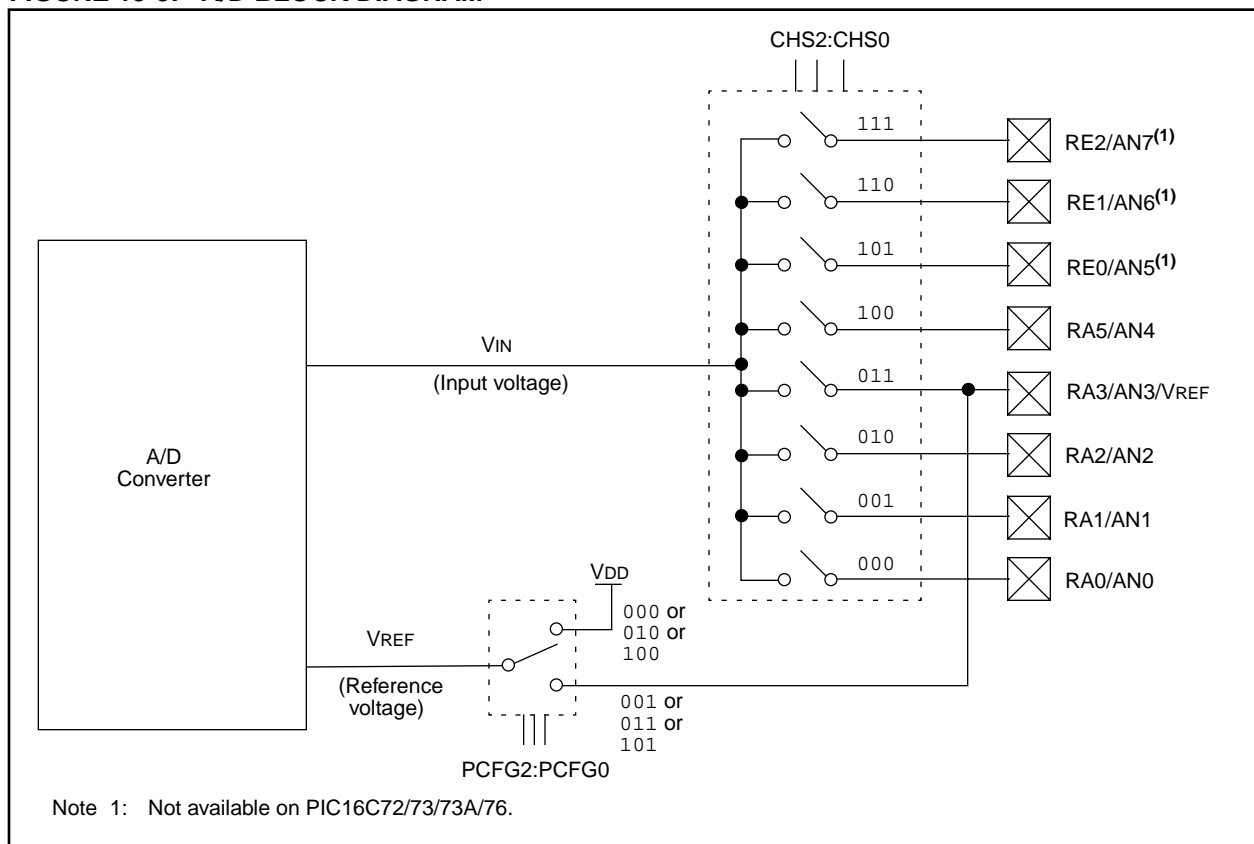
# PIC16C7X

The ADRES register contains the result of the A/D conversion. When the A/D conversion is complete, the result is loaded into the ADRES register, the GO/DONE bit (ADCON0<2>) is cleared, and A/D interrupt flag bit ADIF is set. The block diagrams of the A/D module are shown in Figure 13-3.

After the A/D module has been configured as desired, the selected channel must be acquired before the conversion is started. The analog input channels must have their corresponding TRIS bits selected as an input. To determine acquisition time, see Section 13.1. After this acquisition time has elapsed the A/D conversion can be started. The following steps should be followed for doing an A/D conversion:

1. Configure the A/D module:
  - Configure analog pins / voltage reference / and digital I/O (ADCON1)
  - Select A/D input channel (ADCON0)
  - Select A/D conversion clock (ADCON0)
  - Turn on A/D module (ADCON0)
2. Configure A/D interrupt (if desired):
  - Clear ADIF bit
  - Set ADIE bit
  - Set GIE bit
3. Wait the required acquisition time.
4. Start conversion:
  - Set GO/DONE bit (ADCON0)
5. Wait for A/D conversion to complete, by either:
  - Polling for the GO/DONE bit to be cleared
 OR
  - Waiting for the A/D interrupt
6. Read A/D Result register (ADRES), clear bit ADIF if required.
7. For next conversion, go to step 1 or step 2 as required. The A/D conversion time per bit is defined as TAD. A minimum wait of 2TAD is required before next acquisition starts.

**FIGURE 13-3: A/D BLOCK DIAGRAM**



# PIC16C7X

## 13.1 A/D Acquisition Requirements

Applicable Devices				
72	73	73A	74	74A
76	77			

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 13-4. The source impedance ( $R_s$ ) and the internal sampling switch ( $R_{ss}$ ) impedance directly affect the time required to charge the capacitor CHOLD. The sampling switch ( $R_{ss}$ ) impedance varies over the device voltage ( $V_{DD}$ ), Figure 13-4. 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 k $\Omega$ .** After the analog input channel is selected (changed) this acquisition must be done before the conversion can be started.

To calculate the minimum acquisition time, Equation 13-1 may be used. This equation calculates the acquisition time to within 1/2 LSb error is used (512 steps for the A/D). The 1/2 LSb error is the maximum error allowed for the A/D to meet its specified accuracy.

### EQUATION 13-1: A/D MINIMUM CHARGING TIME

$$V_{HOLD} = (V_{REF} - (V_{REF}/512)) \cdot (1 - e^{-(T_{CAP}/CHOLD)(R_{IC} + R_{SS} + R_s)})$$

Given:  $V_{HOLD} = (V_{REF}/512)$ , for 1/2 LSb resolution

The above equation reduces to:

$$T_{CAP} = -(51.2 \text{ pF})(1 \text{ k}\Omega + R_{SS} + R_s) \ln(1/511)$$

Example 13-1 shows the calculation of the minimum required acquisition time  $T_{ACQ}$ . This calculation is based on the following system assumptions.

$$CHOLD = 51.2 \text{ pF}$$

$$R_s = 10 \text{ k}\Omega$$

1/2 LSb error

$$V_{DD} = 5V \rightarrow R_{ss} = 7 \text{ k}\Omega$$

Temp (application system max.) = 50°C

$$V_{HOLD} = 0 @ t = 0$$

**Note 1:** The reference voltage ( $V_{REF}$ ) has no effect on the equation, since it cancels itself out.

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

**Note 3:** The maximum recommended impedance for analog sources is 10 k $\Omega$ . This is required to meet the pin leakage specification.

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

### EXAMPLE 13-1: CALCULATING THE MINIMUM REQUIRED ACQUISITION TIME

$T_{ACQ} =$  Amplifier Settling Time +  
Holding Capacitor Charging Time +  
Temperature Coefficient

$$T_{ACQ} = 5 \mu\text{s} + T_{CAP} + [(Temp - 25^\circ\text{C})(0.05 \mu\text{s}/^\circ\text{C})]$$

$$T_{CAP} = -CHOLD (R_{IC} + R_{SS} + R_s) \ln(1/511)$$

$$= -51.2 \text{ pF} (1 \text{ k}\Omega + 7 \text{ k}\Omega + 10 \text{ k}\Omega) \ln(0.0020)$$

$$= -51.2 \text{ pF} (18 \text{ k}\Omega) \ln(0.0020)$$

$$= -0.921 \mu\text{s} (-6.2364)$$

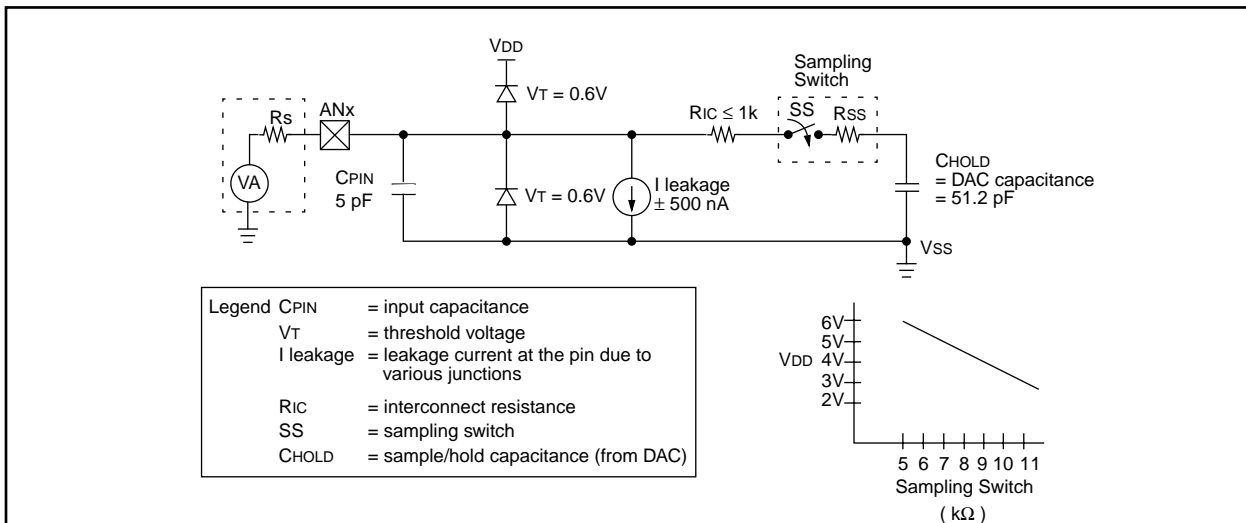
$$5.747 \mu\text{s}$$

$$T_{ACQ} = 5 \mu\text{s} + 5.747 \mu\text{s} + [(50^\circ\text{C} - 25^\circ\text{C})(0.05 \mu\text{s}/^\circ\text{C})]$$

$$10.747 \mu\text{s} + 1.25 \mu\text{s}$$

$$11.997 \mu\text{s}$$

FIGURE 13-4: ANALOG INPUT MODEL





## 13.2 Selecting the A/D Conversion Clock

Applicable Devices					
72	73	73A	74	74A	76/77

The A/D conversion time per bit is defined as TAD. The A/D conversion requires 9.5TAD per 8-bit conversion. The source of the A/D conversion clock is software selectable. The four possible options for TAD are:

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

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

Table 13-1 shows the resultant TAD times derived from the device operating frequencies and the A/D clock source selected.

## 13.3 Configuring Analog Port Pins

Applicable Devices					
72	73	73A	74	74A	76/77

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 (VOH or VOL) will be converted.

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

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

**Note 2:** Analog levels on any pin that is defined as a digital input (including the AN7:AN0 pins), may cause the input buffer to consume current that is out of the devices specification.

**TABLE 13-1: TAD vs. DEVICE OPERATING FREQUENCIES**

AD Clock Source (TAD)		Device Frequency			
Operation	ADCS1:ADCS0	20 MHz	5 MHz	1.25 MHz	333.33 kHz
2Tosc	00	100 ns <sup>(2)</sup>	400 ns <sup>(2)</sup>	1.6 $\mu$ s	6 $\mu$ s
8Tosc	01	400 ns <sup>(2)</sup>	1.6 $\mu$ s	6.4 $\mu$ s	24 $\mu$ s <sup>(3)</sup>
32Tosc	10	1.6 $\mu$ s	6.4 $\mu$ s	25.6 $\mu$ s <sup>(3)</sup>	96 $\mu$ s <sup>(3)</sup>
RC <sup>(5)</sup>	11	2 - 6 $\mu$ s <sup>(1,4)</sup>	2 - 6 $\mu$ s <sup>(1,4)</sup>	2 - 6 $\mu$ s <sup>(1,4)</sup>	2 - 6 $\mu$ s <sup>(1)</sup>

Legend: Shaded cells are outside of recommended range.

Note 1: The RC source has a typical TAD time of 4  $\mu$ s.

2: These values violate the minimum required TAD time.

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

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

5: For extended voltage devices (LC), please refer to Electrical Specifications section.

# PIC16C7X

## 13.4 A/D Conversions

Applicable Devices					
72	73	73A	74	74A	76/77

Example 13-2 shows how to perform an A/D conversion. The RA pins are configured as analog inputs. The analog reference (VREF) is the device VDD. The A/D interrupt is enabled, and the A/D conversion clock is FRC. The conversion is performed on the RA0 pin (channel 0).

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

Clearing the GO/DONE bit during a conversion will abort the current conversion. The ADRES register will NOT be updated with the partially completed A/D conversion sample. That is, the ADRES register will continue to contain the value of the last completed conversion (or the last value written to the ADRES register). After the A/D conversion is aborted, a 2TAD wait is required before the next acquisition is started. After this 2TAD wait, an acquisition is automatically started on the selected channel.

### EXAMPLE 13-2: A/D CONVERSION

```
BSF     STATUS, RP0           ; Select Bank 1
BCF     STATUS, RP1           ; PIC16C76/77 only
CLRF    ADCON1                ; Configure A/D inputs
BSF     PIE1, ADIE            ; Enable A/D interrupts
BCF     STATUS, RP0           ; Select Bank 0
MOVLW   0xC1                  ; RC Clock, A/D is on, Channel 0 is selected
MOVWF   ADCON0                ;
BCF     PIR1, ADIF            ; Clear A/D interrupt flag bit
BSF     INTCON, PEIE          ; Enable peripheral interrupts
BSF     INTCON, GIE           ; Enable all interrupts
;
; Ensure that the required sampling time for the selected input channel has elapsed.
; Then the conversion may be started.
;
BSF     ADCON0, GO             ; Start A/D Conversion
:       ; The ADIF bit will be set and the GO/DONE bit
:       ; is cleared upon completion of the A/D Conversion.
```

## 13.4.1 FASTER CONVERSION - LOWER RESOLUTION TRADE-OFF

Not all applications require a result with 8-bits of resolution, but may instead require a faster conversion time. The A/D module allows users to make the trade-off of conversion speed to resolution. Regardless of the resolution required, the acquisition time is the same. To speed up the conversion, the clock source of the A/D module may be switched so that the TAD time violates the minimum specified time (see the applicable electrical specification). Once the TAD time violates the minimum specified time, all the following A/D result bits are not valid (see A/D Conversion Timing in the Electrical Specifications section.) The clock sources may only be switched between the three oscillator versions (cannot be switched from/to RC). The equation to determine the time before the oscillator can be switched is as follows:

$$\text{Conversion time} = 2T_{AD} + N \cdot T_{AD} + (8 - N)(2T_{OSC})$$

Where: N = number of bits of resolution required.

Since the TAD is based from the device oscillator, the user must use some method (a timer, software loop, etc.) to determine when the A/D oscillator may be changed. Example 13-3 shows a comparison of time required for a conversion with 4-bits of resolution, versus the 8-bit resolution conversion. The example is for devices operating at 20 MHz and 16 MHz (The A/D clock is programmed for 32TOSC), and assumes that immediately after 6TAD, the A/D clock is programmed for 2TOSC.

The 2TOSC violates the minimum TAD time since the last 4-bits will not be converted to correct values.

### EXAMPLE 13-3: 4-BIT vs. 8-BIT CONVERSION TIMES

	Freq. (MHz) <sup>(1)</sup>	Resolution	
		4-bit	8-bit
TAD	20	1.6 μs	1.6 μs
	16	2.0 μs	2.0 μs
TOSC	20	50 ns	50 ns
	16	62.5 ns	62.5 ns
2TAD + N • TAD + (8 - N)(2TOSC)	20	10 μs	16 μs
	16	12.5 μs	20 μs

Note 1: PIC16C7X devices have a minimum TAD time of 1.6 μs.

# PIC16C7X

## 13.5 A/D Operation During Sleep

Applicable Devices					
72	73	73A	74	74A	76/77

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

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

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

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

## 13.6 A/D Accuracy/Error

Applicable Devices					
72	73	73A	74	74A	76/77

The absolute accuracy specified for the A/D converter includes the sum of all contributions for quantization error, integral error, differential error, full scale error, offset error, and monotonicity. It is defined as the maximum deviation from an actual transition versus an ideal transition for any code. The absolute error of the A/D converter is specified at  $\pm 1$  LSb for  $V_{DD} = V_{REF}$  (over the device's specified operating range). However, the accuracy of the A/D converter will degrade as  $V_{DD}$  diverges from  $V_{REF}$ .

For a given range of analog inputs, the output digital code will be the same. This is due to the quantization of the analog input to a digital code. Quantization error is typically  $\pm 1/2$  LSb and is inherent in the analog to digital conversion process. The only way to reduce quantization error is to increase the resolution of the A/D converter.

Offset error measures the first actual transition of a code versus the first ideal transition of a code. Offset error shifts the entire transfer function. Offset error can be calibrated out of a system or introduced into a system through the interaction of the total leakage current and source impedance at the analog input.

Gain error measures the maximum deviation of the last actual transition and the last ideal transition adjusted for offset error. This error appears as a change in slope of the transfer function. The difference in gain error to full scale error is that full scale does not take offset error into account. Gain error can be calibrated out in software.

Linearity error refers to the uniformity of the code changes. Linearity errors cannot be calibrated out of the system. Integral non-linearity error measures the actual code transition versus the ideal code transition adjusted by the gain error for each code.

Differential non-linearity measures the maximum actual code width versus the ideal code width. This measure is unadjusted.

The maximum pin leakage current is  $\pm 1 \mu A$ .

In systems where the device frequency is low, use of the A/D RC clock is preferred. At moderate to high frequencies, TAD should be derived from the device oscillator. TAD must not violate the minimum and should be  $\leq 8 \mu s$  for preferred operation. This is because TAD, when derived from TOSC, is kept away from on-chip phase clock transitions. This reduces, to a large extent, the effects of digital switching noise. This is not possible with the RC derived clock. The loss of accuracy due to digital switching noise can be significant if many I/O pins are active.

In systems where the device will enter SLEEP mode after the start of the A/D conversion, the RC clock source selection is required. In this mode, the digital noise from the modules in SLEEP are stopped. This method gives high accuracy.

## 13.7 Effects of a RESET

Applicable Devices					
72	73	73A	74	74A	76/77

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

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

## 13.8 Use of the CCP Trigger

Applicable Devices
72 73 73A 74 74A 76 77

**Note:** In the PIC16C72, the "special event trigger" is implemented in the CCP1 module.

An A/D conversion can be started by the "special event trigger" of the CCP2 module (CCP1 on the PIC16C72 only). 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 the minimum acquisition done before the "special event trigger" sets the GO/DONE bit (starts a conversion).

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

## 13.9 Connection Considerations

Applicable Devices
72 73 73A 74 74A 76 77

If the input voltage exceeds the rail values (VSS or VDD) by greater than 0.2V, then the accuracy of the conversion is out of specification.

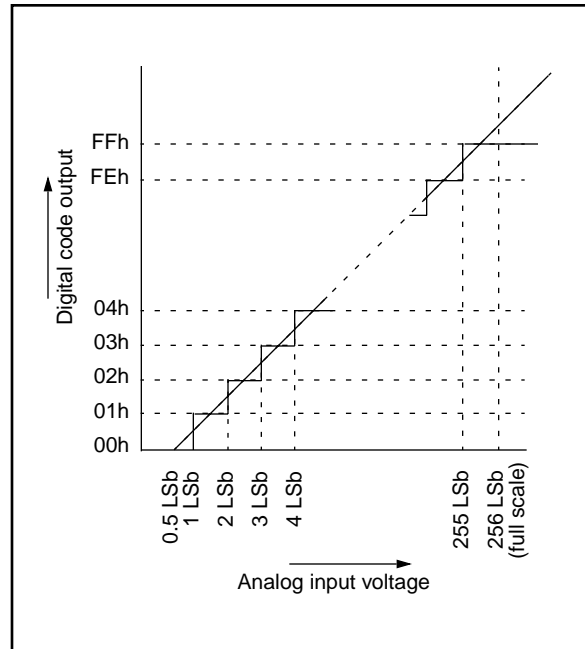
An external RC filter is sometimes added for anti-aliasing of the input signal. The R component should be selected to ensure that the total source impedance is kept under the 10 kΩ recommended specification. Any external components connected (via hi-impedance) to an analog input pin (capacitor, zener diode, etc.) should have very little leakage current at the pin.

## 13.10 Transfer Function

Applicable Devices
72 73 73A 74 74A 76 77

The ideal transfer function of the A/D converter is as follows: the first transition occurs when the analog input voltage (VAIN) is Analog VREF/256 (Figure 13-5).

FIGURE 13-5: A/D TRANSFER FUNCTION

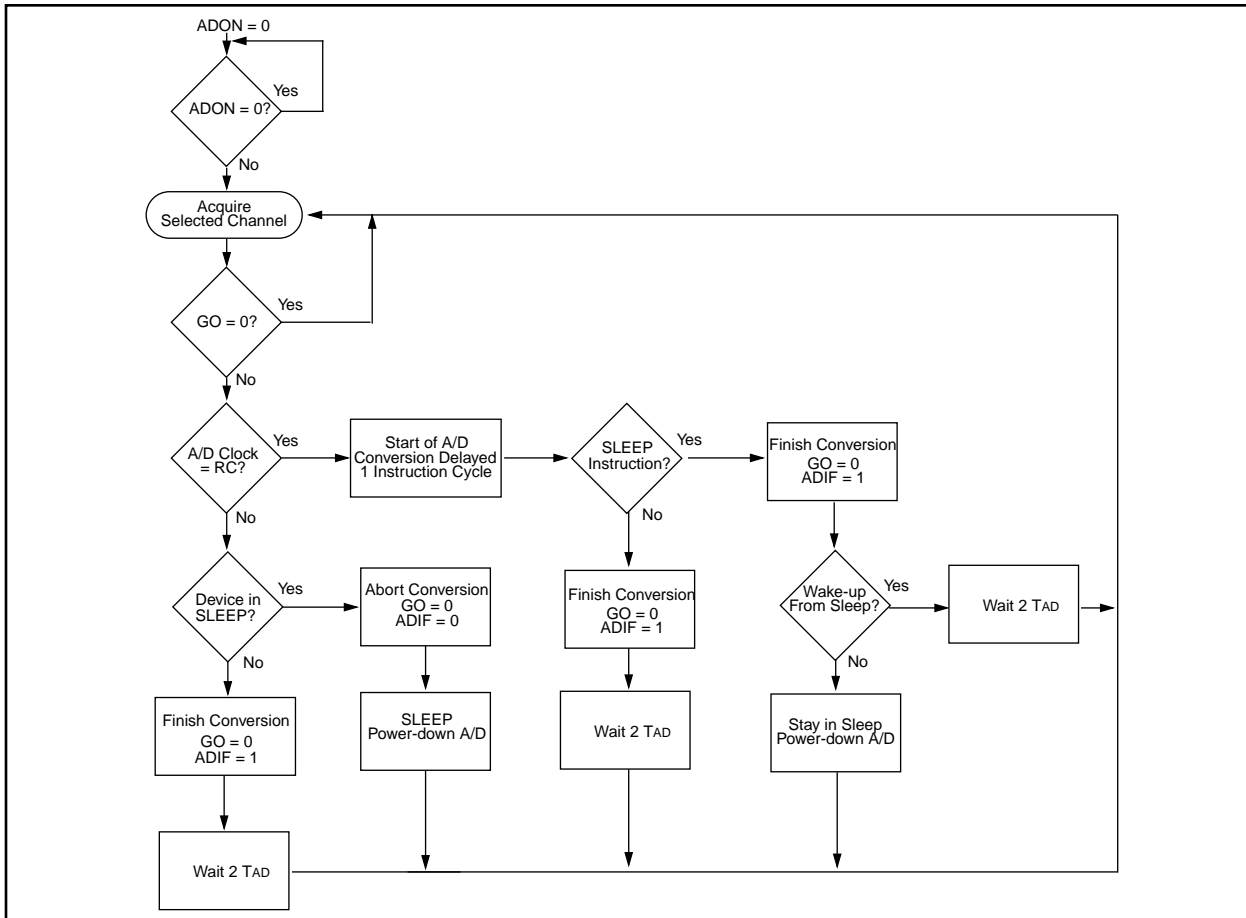


## 13.11 References

A very good reference for understanding A/D converters is the "Analog-Digital Conversion Handbook" third edition, published by Prentice Hall (ISBN 0-13-03-2848-0).

# PIC16C7X

**FIGURE 13-6: FLOWCHART OF A/D OPERATION**



**TABLE 13-2: REGISTERS/BITS ASSOCIATED WITH A/D, PIC16C72**

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	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	—	ADIF	—	—	SSPIF	CCP1IF	TMR2IF	TMR1IF	-0-- 0000	-0-- 0000
8Ch	PIE1	—	ADIE	—	—	SSPIE	CCP1IE	TMR2IE	TMR1IE	-0-- 0000	-0-- 0000
1Eh	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

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

# PIC16C7X

**TABLE 13-3: SUMMARY OF A/D REGISTERS, PIC16C73/73A/74/74A/76/77**

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	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u	
0Ch	PIR1	PSPIF <sup>(1)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000	
8Ch	PIE1	PSPIE <sup>(1)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000	
0Dh	PIR2	—	—	—	—	—	—	—	CCP2IF	---- -0	---- -0	
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	—	—	—	—	—	RE2	RE1	RE0	---- -xxx	---- -uuu	
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 for A/D conversion.

Note 1: Bits PSPIE and PSPIF are reserved on the PIC6C73/73A/76, always maintain these bits clear.

# PIC16C7X

---

NOTES:



## 14.0 SPECIAL FEATURES OF THE CPU

Applicable Devices						
72	73	73A	74	74A	76	77

What sets a microcontroller apart from other processors are special circuits to deal with the needs of real-time applications. The PIC16CXX family has a host of such 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

The PIC16CXX has a Watchdog Timer which can be shut off only through configuration bits. It runs off its own RC oscillator for added reliability. There are two timers that offer necessary delays on power-up. One is the Oscillator Start-up Timer (OST), intended to keep

the chip in reset until the crystal oscillator is stable. The other is the Power-up Timer (PWRT), which provides a fixed delay of 72 ms (nominal) on power-up only, designed to keep the part in reset while the power supply stabilizes. With these two timers on-chip, most applications need no external reset circuitry.

SLEEP mode is designed to offer a very low current power-down mode. The user can wake-up from SLEEP through external reset, Watchdog Timer Wake-up, or through an interrupt. Several oscillator options are also made available to allow the part to fit the application. The RC oscillator option saves system cost while the LP crystal option saves power. A set of configuration bits are used to select various options.

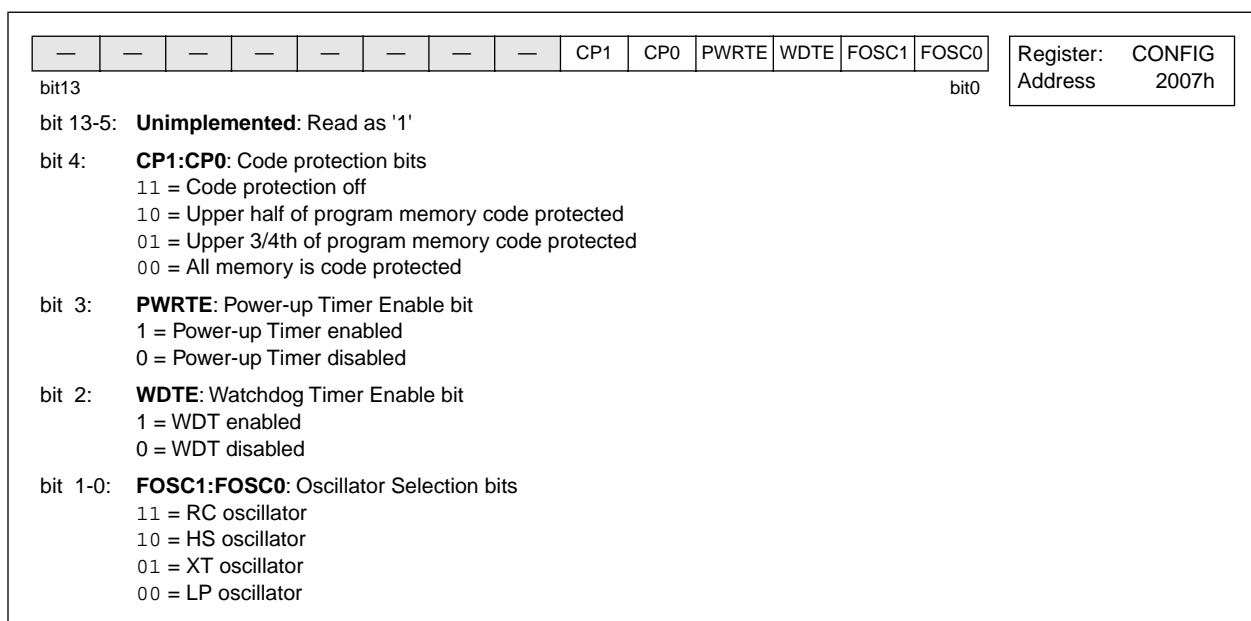
### 14.1 Configuration Bits

Applicable Devices						
72	73	73A	74	74A	76	77

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

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

**FIGURE 14-1: CONFIGURATION WORD FOR PIC16C73/74**



# PIC16C7X

**FIGURE 14-2: CONFIGURATION WORD FOR PIC16C72/73A/74A/76/77**

CP1	CP0	CP1	CP0	CP1	CP0	—	BODEN	CP1	CP0	PWRTE	WDTE	FOSC1	FOSC0	Register: CONFIG
													Address: 2007h	
bit13													bit0	
<p>bit 13-8: <b>CP1:CP0</b>: Code Protection bits <sup>(2)</sup></p> <p>5-4: 11 = Code protection off          10 = Upper half of program memory code protected          01 = Upper 3/4th of program memory code protected          00 = All memory is code protected</p> <p>bit 7: <b>Unimplemented</b>: Read as '1'</p> <p>bit 6: <b>BODEN</b>: Brown-out Reset Enable bit <sup>(1)</sup>          1 = BOR enabled          0 = BOR disabled</p> <p>bit 3: <b>PWRTE</b>: Power-up Timer Enable bit <sup>(1)</sup>          1 = PWRT disabled          0 = PWRT enabled</p> <p>bit 2: <b>WDTE</b>: Watchdog Timer Enable bit          1 = WDT enabled          0 = WDT disabled</p> <p>bit 1-0: <b>FOSC1:FOSC0</b>: Oscillator Selection bits          11 = RC oscillator          10 = HS oscillator          01 = XT oscillator          00 = LP oscillator</p> <p>Note 1: Enabling Brown-out Reset automatically enables Power-up Timer (PWRT) regardless of the value of bit <b>PWRTE</b>. Ensure the Power-up Timer is enabled anytime Brown-out Reset is enabled.</p> <p>Note 2: All of the CP1:CP0 pairs have to be given the same value to enable the code protection scheme listed.</p>														

# PIC16C7X

## 14.2 Oscillator Configurations

Applicable Devices
72 73 73A 74 74A 76 77

### 14.2.1 OSCILLATOR TYPES

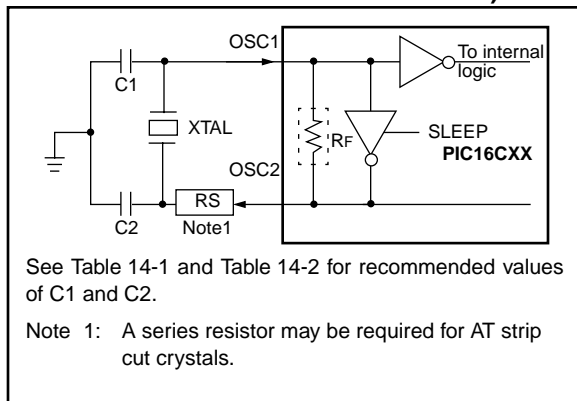
The PIC16CXX can be operated in four different oscillator modes. The user can program two configuration bits (FOSC1 and FOSC0) to select one of these four modes:

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

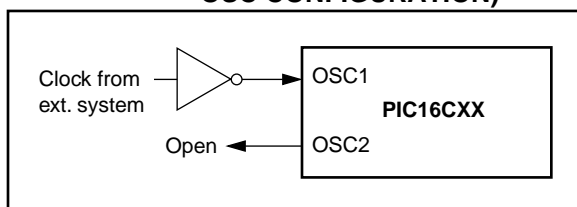
### 14.2.2 CRYSTAL OSCILLATOR/CERAMIC RESONATORS

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 14-3). The PIC16CXX Oscillator design requires the use of a parallel cut crystal. Use of a series cut crystal may give a frequency out of the crystal manufacturers specifications. When in XT, LP or HS modes, the device can have an external clock source to drive the OSC1/CLKIN pin (Figure 14-4).

**FIGURE 14-3: CRYSTAL/CERAMIC RESONATOR OPERATION (HS, XT OR LP OSC CONFIGURATION)**



**FIGURE 14-4: EXTERNAL CLOCK INPUT OPERATION (HS, XT OR LP OSC CONFIGURATION)**



**TABLE 14-1: CERAMIC RESONATORS**

Ranges Tested:			
Mode	Freq	OSC1	OSC2
XT	455 kHz	68 - 100 pF	68 - 100 pF
	2.0 MHz	15 - 68 pF	15 - 68 pF
	4.0 MHz	15 - 68 pF	15 - 68 pF
HS	8.0 MHz	10 - 68 pF	10 - 68 pF
	16.0 MHz	10 - 22 pF	10 - 22 pF
These values are for design guidance only. See notes at bottom of page.			
Resonators Used:			
455 kHz	Panasonic EFO-A455K04B	± 0.3%	
2.0 MHz	Murata Erie CSA2.00MG	± 0.5%	
4.0 MHz	Murata Erie CSA4.00MG	± 0.5%	
8.0 MHz	Murata Erie CSA8.00MT	± 0.5%	
16.0 MHz	Murata Erie CSA16.00MX	± 0.5%	
All resonators used did not have built-in capacitors.			

**TABLE 14-2: CAPACITOR SELECTION FOR CRYSTAL OSCILLATOR**

Osc Type	Crystal Freq	Cap. Range C1	Cap. Range C2
LP	32 kHz	33 pF	33 pF
	200 kHz	15 pF	15 pF
XT	200 kHz	47-68 pF	47-68 pF
	1 MHz	15 pF	15 pF
	4 MHz	15 pF	15 pF
HS	4 MHz	15 pF	15 pF
	8 MHz	15-33 pF	15-33 pF
	20 MHz	15-33 pF	15-33 pF
These values are for design guidance only. See notes at bottom of page.			
Crystals Used			
32 kHz	Epson C-001R32.768K-A	± 20 PPM	
200 kHz	STD XTL 200.000KHz	± 20 PPM	
1 MHz	ECS ECS-10-13-1	± 50 PPM	
4 MHz	ECS ECS-40-20-1	± 50 PPM	
8 MHz	EPSON CA-301 8.000M-C	± 30 PPM	
20 MHz	EPSON CA-301 20.000M-C	± 30 PPM	

- Note 1: Recommended values of C1 and C2 are identical to the ranges tested (Table 14-1).
- 2: Higher capacitance increases the stability of oscillator but also increases the start-up time.
- 3: Since each resonator/crystal has its own characteristics, the user should consult the resonator/crystal manufacturer for appropriate values of external components.
- 4: Rs may be required in HS mode as well as XT mode to avoid overdriving crystals with low drive level specification.

# PIC16C7X

## 14.2.3 EXTERNAL CRYSTAL OSCILLATOR CIRCUIT

Either a prepackaged oscillator can be used or a simple oscillator circuit with TTL gates can be built. Prepackaged oscillators provide a wide operating range and better stability. A well-designed crystal oscillator will provide good performance with TTL gates. Two types of crystal oscillator circuits can be used; one with series resonance, or one with parallel resonance.

Figure 14-5 shows implementation of a parallel resonant oscillator circuit. The circuit is designed to use the fundamental frequency of the crystal. The 74AS04 inverter performs the 180-degree phase shift that a parallel oscillator requires. The 4.7 kΩ resistor provides the negative feedback for stability. The 10 kΩ potentiometer biases the 74AS04 in the linear region. This could be used for external oscillator designs.

**FIGURE 14-5: EXTERNAL PARALLEL RESONANT CRYSTAL OSCILLATOR CIRCUIT**

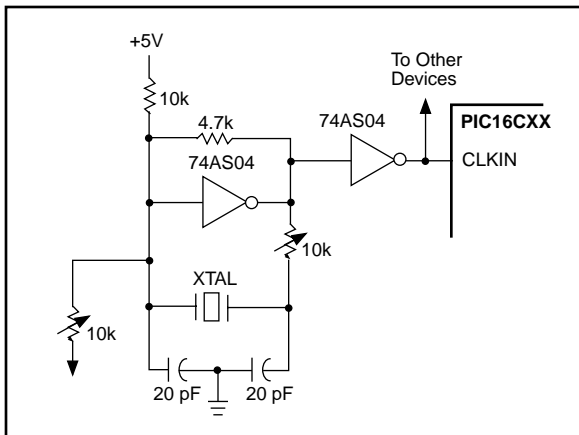
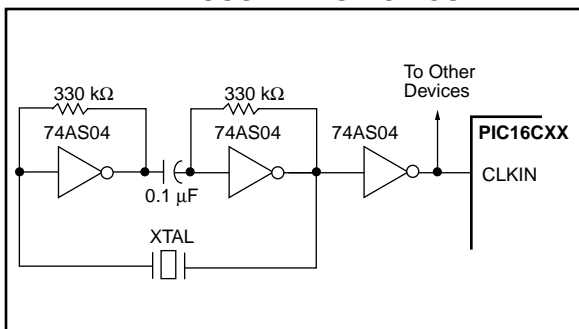


Figure 14-6 shows a series resonant oscillator circuit. This circuit is also designed to use the fundamental frequency of the crystal. The inverter performs a 180-degree phase shift in a series resonant oscillator circuit. The 330 kΩ resistors provide the negative feedback to bias the inverters in their linear region.

**FIGURE 14-6: EXTERNAL SERIES RESONANT CRYSTAL OSCILLATOR CIRCUIT**



## 14.2.4 RC OSCILLATOR

For timing insensitive applications the “RC” device option offers additional cost savings. The RC oscillator frequency is a function of the supply voltage, the resistor ( $R_{ext}$ ) and capacitor ( $C_{ext}$ ) values, and the operating temperature. In addition to this, the oscillator frequency will vary from unit to unit due to normal process parameter variation. Furthermore, the difference in lead frame capacitance between package types will also affect the oscillation frequency, especially for low  $C_{ext}$  values. The user also needs to take into account variation due to tolerance of external R and C components used. Figure 14-7 shows how the R/C combination is connected to the PIC16CXX. For  $R_{ext}$  values below 2.2 kΩ, the oscillator operation may become unstable, or stop completely. For very high  $R_{ext}$  values (e.g. 1 MΩ), the oscillator becomes sensitive to noise, humidity and leakage. Thus, we recommend to keep  $R_{ext}$  between 3 kΩ and 100 kΩ.

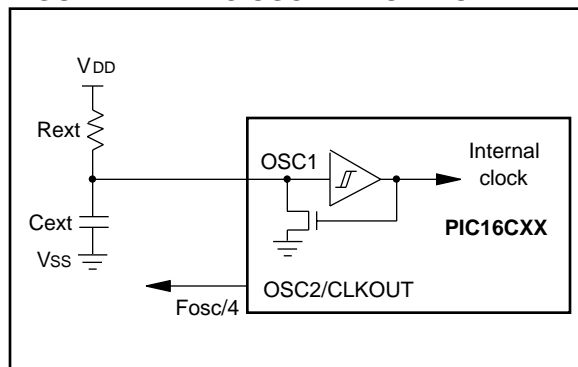
Although the oscillator will operate with no external capacitor ( $C_{ext} = 0$  pF), we recommend using values above 20 pF for noise and stability reasons. With no or small external capacitance, the oscillation frequency can vary dramatically due to changes in external capacitances, such as PCB trace capacitance or package lead frame capacitance.

See characterization data for desired device for RC frequency variation from part to part due to normal process variation. The variation is larger for larger R (since leakage current variation will affect RC frequency more for large R) and for smaller C (since variation of input capacitance will affect RC frequency more).

See characterization data for desired device for variation of oscillator frequency due to  $V_{DD}$  for given  $R_{ext}/C_{ext}$  values as well as frequency variation due to operating temperature for given R, C, and  $V_{DD}$  values.

The oscillator frequency, divided by 4, is available on the OSC2/CLKOUT pin, and can be used for test purposes or to synchronize other logic (see Figure 3-4 for waveform).

**FIGURE 14-7: RC OSCILLATOR MODE**



# PIC16C7X

## 14.3 Reset

Applicable Devices					
72	73	73A	74	74A	76/77

The PIC16CXX 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 (normal operation)
- Brown-out Reset (BOR) (PIC16C72/73A/74A/76/77)

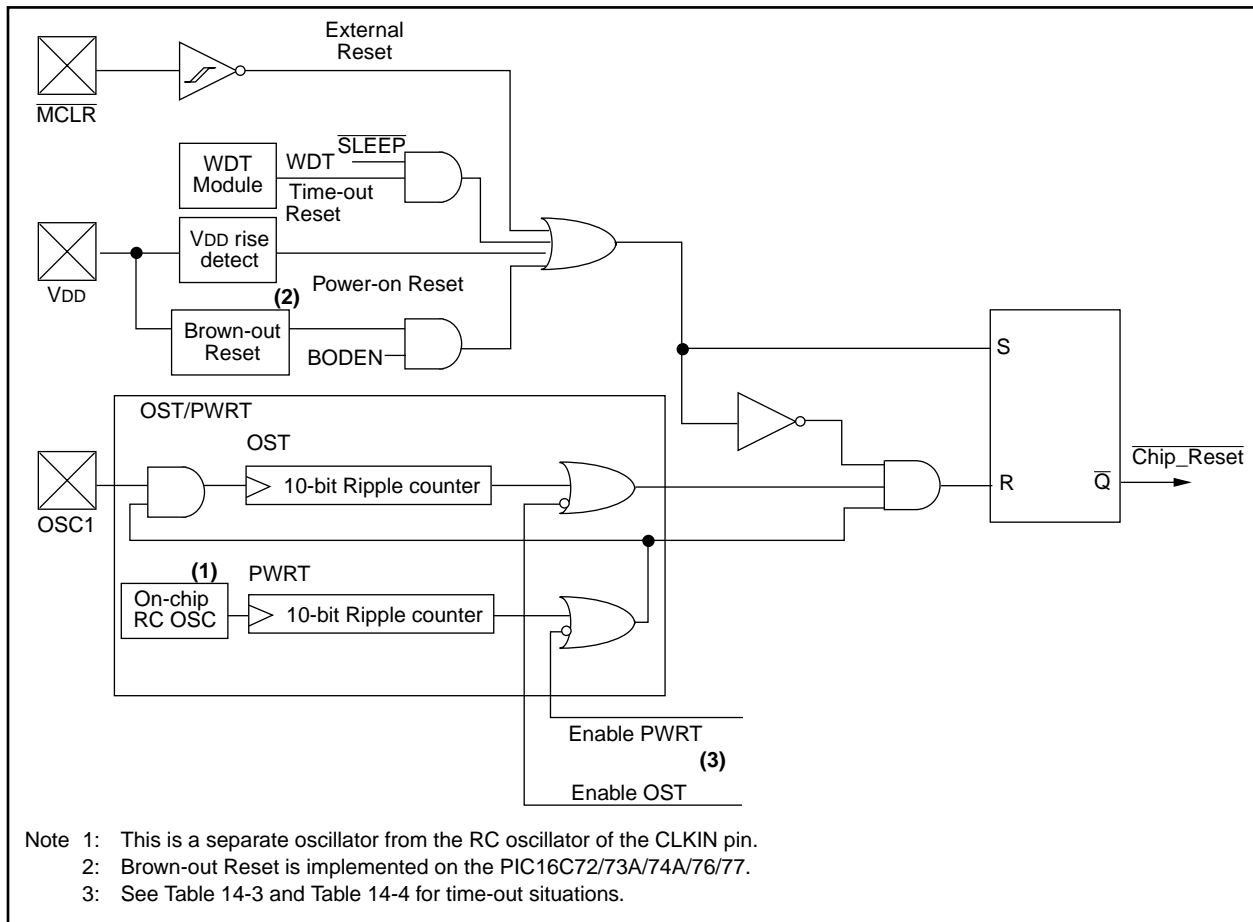
Some registers are not affected in any reset condition; their status is unknown on POR and unchanged in any other reset. Most other registers are reset to a "reset state" on Power-on Reset (POR), on the  $\overline{\text{MCLR}}$  and WDT Reset, on  $\overline{\text{MCLR}}$  reset during SLEEP, and Brown-out 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 14-5 and Table 14-6. These bits are used in software to determine the nature of the reset. See Table 14-8 for a full description of reset states of all registers.

A simplified block diagram of the on-chip reset circuit is shown in Figure 14-8.

The PIC16C72/73A/74A/76/77 have a  $\overline{\text{MCLR}}$  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.

**FIGURE 14-8: SIMPLIFIED BLOCK DIAGRAM OF ON-CHIP RESET CIRCUIT**



# PIC16C7X

## 14.4 Power-on Reset (POR), Power-up Timer (PWRT) and Oscillator Start-up Timer (OST), and Brown-out Reset (BOR)

Applicable Devices					
72	73	73A	74	74A	76/77

### 14.4.1 POWER-ON RESET (POR)

A Power-on Reset pulse is generated on-chip when VDD rise is detected (in the range of 1.5V - 2.1V). To take advantage of the POR, just tie the MCLR pin directly (or through a resistor) to VDD. This will eliminate external RC components usually needed to create a Power-on Reset. A maximum rise time for VDD is specified. See 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. Brown-out Reset may be used to meet the startup conditions.

For additional information, refer to Application Note AN607, "Power-up Trouble Shooting."

### 14.4.2 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 VDD 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 VDD, temperature, and process variation. See DC parameters for details.

### 14.4.3 OSCILLATOR START-UP TIMER (OST)

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

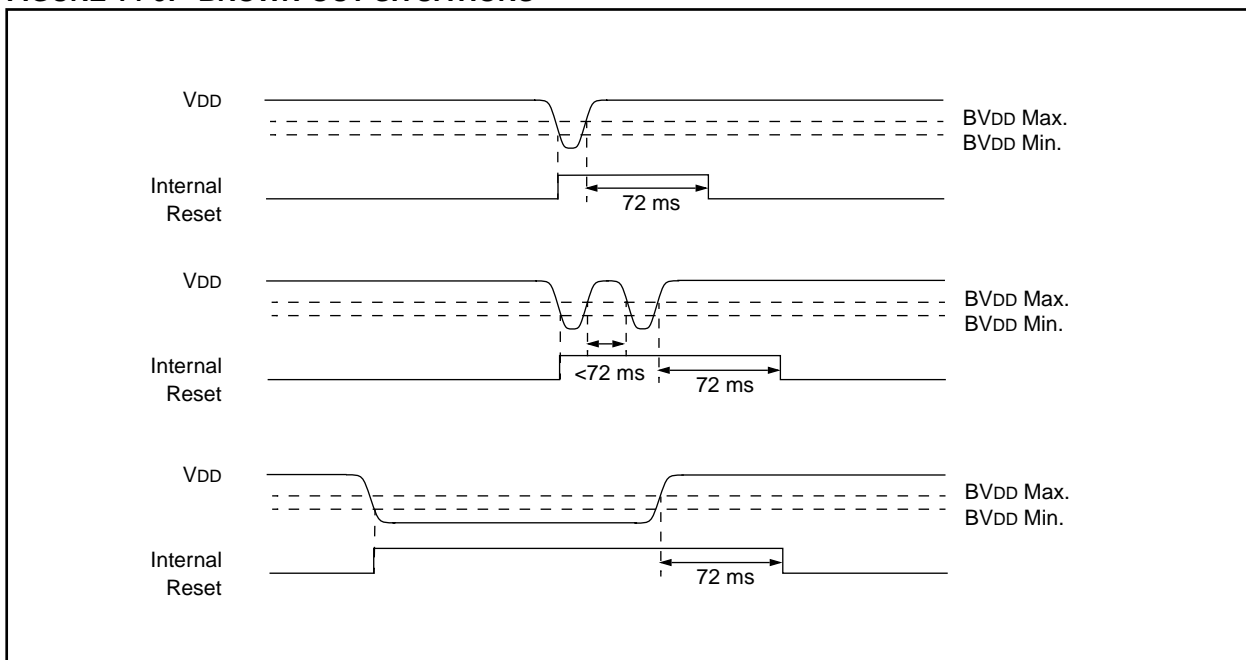
The OST time-out is invoked only for XT, LP and HS modes and only on Power-on Reset or wake-up from SLEEP.

### 14.4.4 BROWN-OUT RESET (BOR)

Applicable Devices					
72	73	73A	74	74A	76/77

A configuration bit, BODEN, can disable (if clear/programmed) or enable (if set) the Brown-out Reset circuitry. If VDD falls below 4.0V (3.8V - 4.2V range) for greater than parameter #35, the brown-out situation will reset the chip. A reset may not occur if VDD falls below 4.0V for less than parameter #35. The chip will remain in Brown-out Reset until VDD rises above BVDD. The Power-up Timer will now be invoked and will keep the chip in RESET an additional 72 ms. If VDD drops below BVDD while the Power-up Timer is running, the chip will go back into a Brown-out Reset and the Power-up Timer will be initialized. Once VDD rises above BVDD, the Power-up Timer will execute a 72 ms time delay. The Power-up Timer should always be enabled when Brown-out Reset is enabled. Figure 14-9 shows typical brown-out situations.

**FIGURE 14-9: BROWN-OUT SITUATIONS**



# PIC16C7X

## 14.4.5 TIME-OUT SEQUENCE

On power-up the time-out sequence is as follows: First PWRT time-out is invoked after the POR time delay has expired. Then OST is activated. The total time-out will vary based on oscillator configuration and the status of the PWRT. For example, in RC mode with the PWRT disabled, there will be no time-out at all. Figure 14-10, Figure 14-11, and Figure 14-12 depict time-out sequences on power-up.

Since the time-outs occur from the POR pulse, if  $\overline{\text{MCLR}}$  is kept low long enough, the time-outs will expire. Then bringing  $\overline{\text{MCLR}}$  high will begin execution immediately (Figure 14-11). This is useful for testing purposes or to synchronize more than one PIC16CXX device operating in parallel.

Table 14-7 shows the reset conditions for some special function registers, while Table 14-8 shows the reset conditions for all the registers.

## 14.4.6 POWER CONTROL/STATUS REGISTER (PCON)

Applicable Devices					
72	73	73A	74	74A	76/77

The Power Control/Status Register, PCON has up to two bits, depending upon the device. Bit0 is not implemented on the PIC16C73 or PIC16C74.

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 BOR occurred. The  $\overline{\text{BOR}}$  bit is a "Don't Care" bit and is not necessarily predictable if the Brown-out Reset circuitry is disabled (by clearing bit BODEN in the Configuration Word).

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 14-3: TIME-OUT IN VARIOUS SITUATIONS, PIC16C73/74**

Oscillator Configuration	Power-up		Wake-up from SLEEP
	PWRTE = 1	PWRTE = 0	
XT, HS, LP	72 ms + 1024Tosc	1024Tosc	1024 Tosc
RC	72 ms	—	—

**TABLE 14-4: TIME-OUT IN VARIOUS SITUATIONS, PIC16C72/73A/74A/76/77**

Oscillator Configuration	Power-up		Brown-out	Wake-up from SLEEP
	PWRTE = 0	PWRTE = 1		
XT, HS, LP	72 ms + 1024Tosc	1024Tosc	72 ms + 1024Tosc	1024Tosc
RC	72 ms	—	72 ms	—

**TABLE 14-5: STATUS BITS AND THEIR SIGNIFICANCE, PIC16C73/74**

POR	$\overline{\text{TO}}$	$\overline{\text{PD}}$	
0	1	1	Power-on Reset
0	0	x	Illegal, $\overline{\text{TO}}$ is set on $\overline{\text{POR}}$
0	x	0	Illegal, $\overline{\text{PD}}$ is set on $\overline{\text{POR}}$
1	0	1	WDT Reset
1	0	0	WDT Wake-up
1	u	u	$\overline{\text{MCLR}}$ Reset during normal operation
1	1	0	$\overline{\text{MCLR}}$ Reset during SLEEP or interrupt wake-up from SLEEP

Legend: u = unchanged, x = unknown

# PIC16C7X

**TABLE 14-6: STATUS BITS AND THEIR SIGNIFICANCE, PIC16C72/73A/74A/76/77**

POR	BOR	TO	PD	
0	x	1	1	Power-on Reset
0	x	0	x	Illegal, $\overline{TO}$ is set on $\overline{POR}$
0	x	x	0	Illegal, $\overline{PD}$ is set on $\overline{POR}$
1	0	x	x	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 14-7: RESET CONDITION FOR SPECIAL REGISTERS**

Condition	Program Counter	STATUS Register	PCON Register PIC16C73/74	PCON Register PIC16C72/73A/74A/76/77
Power-on Reset	000h	0001 1xxx	---- --0-	---- --0x
MCLR Reset during normal operation	000h	000u uuuu	---- --u-	---- --uu
MCLR Reset during SLEEP	000h	0001 0uuu	---- --u-	---- --uu
WDT Reset	000h	0000 1uuu	---- --u-	---- --uu
WDT Wake-up	PC + 1	uuu0 0uuu	---- --u-	---- --uu
Brown-out Reset	000h	0001 1uuu	N/A	---- --u0
Interrupt wake-up from SLEEP	PC + 1 <sup>(1)</sup>	uuu1 0uuu	---- --u-	---- --uu

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

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

**TABLE 14-8: INITIALIZATION CONDITIONS FOR ALL REGISTERS**

Register	Applicable Devices							Power-on Reset, Brown-out Reset	MCLR Resets WDT Reset	Wake-up via WDT or Interrupt
W	72	73	73A	74	74A	76	77	xxxx xxxx	uuuu uuuu	uuuu uuuu
INDF	72	73	73A	74	74A	76	77	N/A	N/A	N/A
TMR0	72	73	73A	74	74A	76	77	xxxx xxxx	uuuu uuuu	uuuu uuuu
PCL	72	73	73A	74	74A	76	77	0000h	0000h	PC + 1 <sup>(2)</sup>
STATUS	72	73	73A	74	74A	76	77	0001 1xxx	000q quuu <sup>(3)</sup>	uuuq quuu <sup>(3)</sup>
FSR	72	73	73A	74	74A	76	77	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTA	72	73	73A	74	74A	76	77	--0x 0000	--0u 0000	--uu uuuu
PORTB	72	73	73A	74	74A	76	77	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTC	72	73	73A	74	74A	76	77	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTD	72	73	73A	74	74A	76	77	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTE	72	73	73A	74	74A	76	77	---- -xxx	---- -uuu	---- -uuu
PCLATH	72	73	73A	74	74A	76	77	---0 0000	---0 0000	---u uuuu

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

Note 1: One or more bits in INTCON, PIR1 and/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 14-7 for reset value for specific condition.



# PIC16C7X

**TABLE 14-8: INITIALIZATION CONDITIONS FOR ALL REGISTERS (Cont'd)**

Register	Applicable Devices							Power-on Reset, Brown-out Reset	MCLR Resets WDT Reset	Wake-up via WDT or Interrupt
INTCON	72	73	73A	74	74A	76	77	0000 000x	0000 000u	uuuu uuuu <sup>(1)</sup>
PIR1	72	73	73A	74	74A	76	77	-0-- 0000	-0-- 0000	-u-- uuuu <sup>(1)</sup>
	72	73	73A	74	74A	76	77	-000 0000	-000 0000	-uuu uuuu <sup>(1)</sup>
	72	73	73A	74	74A	76	77	0000 0000	0000 0000	uuuu uuuu <sup>(1)</sup>
PIR2	72	73	73A	74	74A	76	77	---- --0	---- --0	---- --u <sup>(1)</sup>
TMR1L	72	73	73A	74	74A	76	77	xxxx xxxx	uuuu uuuu	uuuu uuuu
TMR1H	72	73	73A	74	74A	76	77	xxxx xxxx	uuuu uuuu	uuuu uuuu
T1CON	72	73	73A	74	74A	76	77	--00 0000	--uu uuuu	--uu uuuu
TMR2	72	73	73A	74	74A	76	77	0000 0000	0000 0000	uuuu uuuu
T2CON	72	73	73A	74	74A	76	77	-000 0000	-000 0000	-uuu uuuu
SSPBUF	72	73	73A	74	74A	76	77	xxxx xxxx	uuuu uuuu	uuuu uuuu
SSPCON	72	73	73A	74	74A	76	77	0000 0000	0000 0000	uuuu uuuu
CCPR1L	72	73	73A	74	74A	76	77	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCPR1H	72	73	73A	74	74A	76	77	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCP1CON	72	73	73A	74	74A	76	77	--00 0000	--00 0000	--uu uuuu
RCSTA	72	73	73A	74	74A	76	77	0000 -00x	0000 -00x	uuuu -uuu
TXREG	72	73	73A	74	74A	76	77	0000 0000	0000 0000	uuuu uuuu
RCREG	72	73	73A	74	74A	76	77	0000 0000	0000 0000	uuuu uuuu
CCPR2L	72	73	73A	74	74A	76	77	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCPR2H	72	73	73A	74	74A	76	77	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCP2CON	72	73	73A	74	74A	76	77	0000 0000	0000 0000	uuuu uuuu
ADRES	72	73	73A	74	74A	76	77	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADCON0	72	73	73A	74	74A	76	77	0000 00-0	0000 00-0	uuuu uu-u
OPTION	72	73	73A	74	74A	76	77	1111 1111	1111 1111	uuuu uuuu
TRISA	72	73	73A	74	74A	76	77	--11 1111	--11 1111	--uu uuuu
TRISB	72	73	73A	74	74A	76	77	1111 1111	1111 1111	uuuu uuuu
TRISC	72	73	73A	74	74A	76	77	1111 1111	1111 1111	uuuu uuuu
TRISD	72	73	73A	74	74A	76	77	1111 1111	1111 1111	uuuu uuuu
TRISE	72	73	73A	74	74A	76	77	0000 -111	0000 -111	uuuu -uuu
PIE1	72	73	73A	74	74A	76	77	-0-- 0000	-0-- 0000	-u-- uuuu
	72	73	73A	74	74A	76	77	-000 0000	-000 0000	-uuu uuuu
	72	73	73A	74	74A	76	77	0000 0000	0000 0000	uuuu uuuu
PIE2	72	73	73A	74	74A	76	77	---- --0	---- --0	---- --u
PCON	72	73	73A	74	74A	76	77	---- --0-	---- --u-	---- --u-
	72	73	73A	74	74A	76	77	---- --0u	---- --uu	---- --uu
PR2	72	73	73A	74	74A	76	77	1111 1111	1111 1111	1111 1111

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

Note 1: One or more bits in INTCON, PIR1 and/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 14-7 for reset value for specific condition.

# PIC16C7X

**TABLE 14-8: INITIALIZATION CONDITIONS FOR ALL REGISTERS (Cont.'d)**

Register	Applicable Devices							Power-on Reset, Brown-out Reset	MCLR Resets WDT Reset	Wake-up via WDT or Interrupt
SSPADD	72	73	73A	74	74A	76	77	0000 0000	0000 0000	uuuu uuuu
SSPSTAT	72	73	73A	74	74A	76	77	--00 0000	--00 0000	--uu uuuu
TXSTA	72	73	73A	74	74A	76	77	0000 -010	0000 -010	uuuu -uuu
SPBRG	72	73	73A	74	74A	76	77	0000 0000	0000 0000	uuuu uuuu
ADCON1	72	73	73A	74	74A	76	77	---- -000	---- -000	---- -uuu

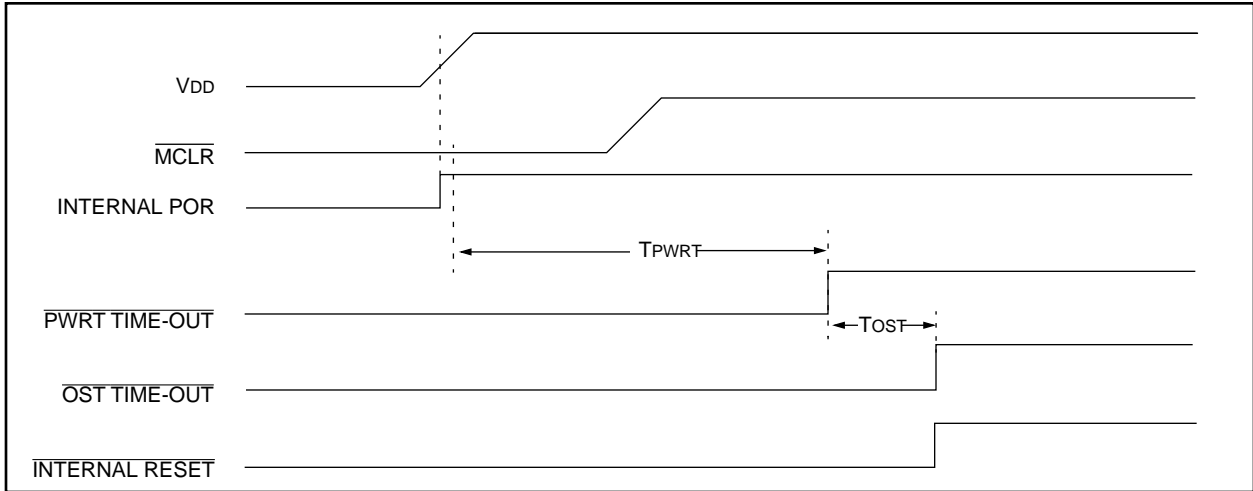
Legend: u = unchanged, x = unknown, - = unimplemented bit, read as '0', q = value depends on condition

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

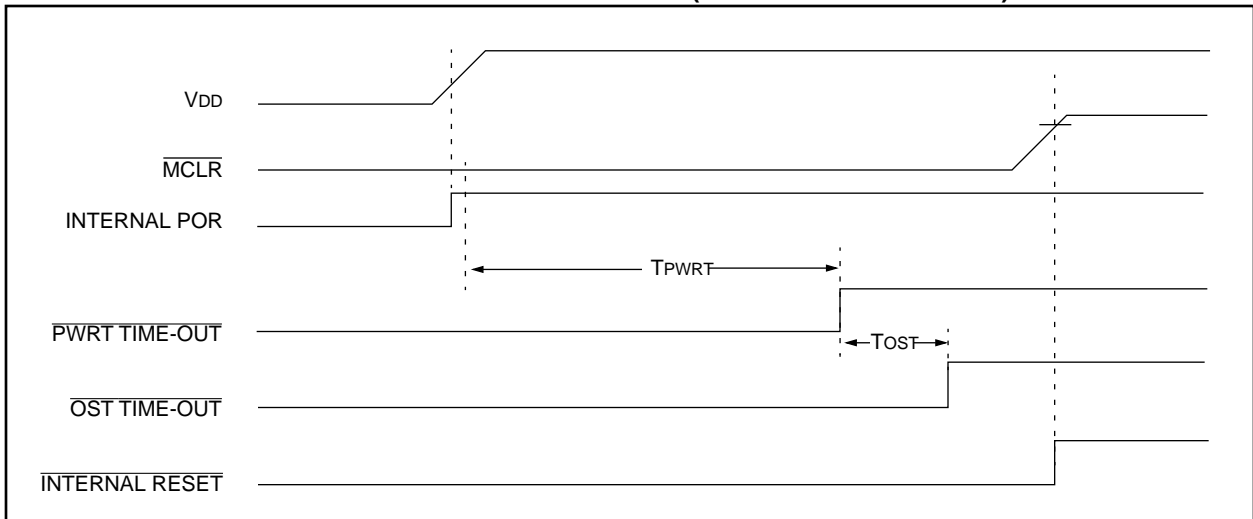
2: When the wake-up is due to an interrupt and the GIE bit is set, the PC is loaded with the interrupt vector (0004h).

3: See Table 14-7 for reset value for specific condition.

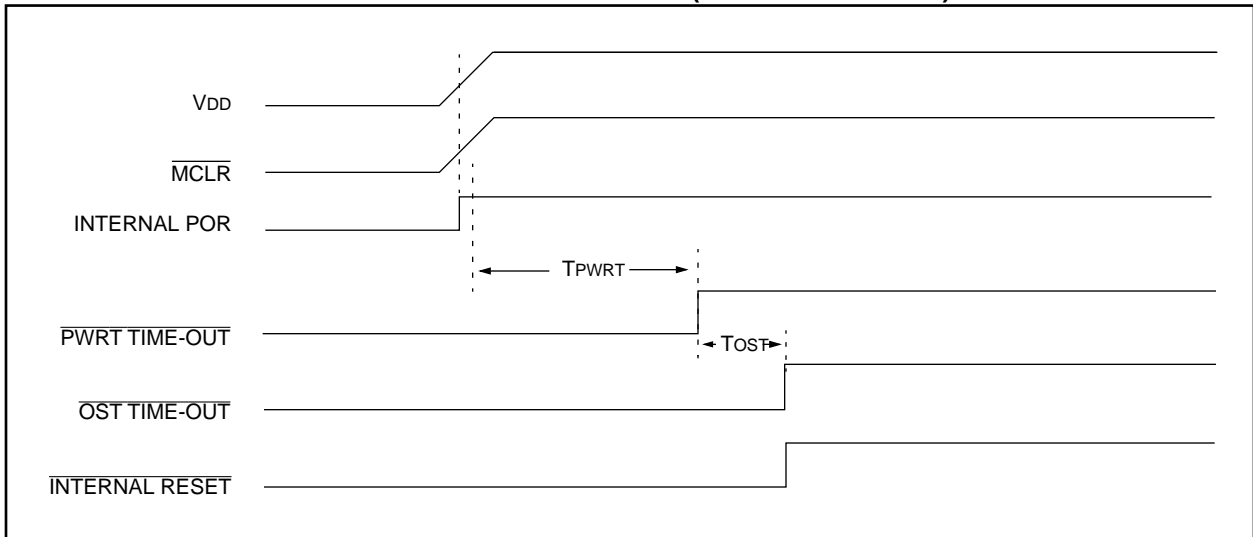
**FIGURE 14-10: TIME-OUT SEQUENCE ON POWER-UP ( $\overline{\text{MCLR}}$  NOT TIED TO  $V_{\text{DD}}$ ): CASE 1**



**FIGURE 14-11: TIME-OUT SEQUENCE ON POWER-UP ( $\overline{\text{MCLR}}$  NOT TIED TO  $V_{\text{DD}}$ ): CASE 2**

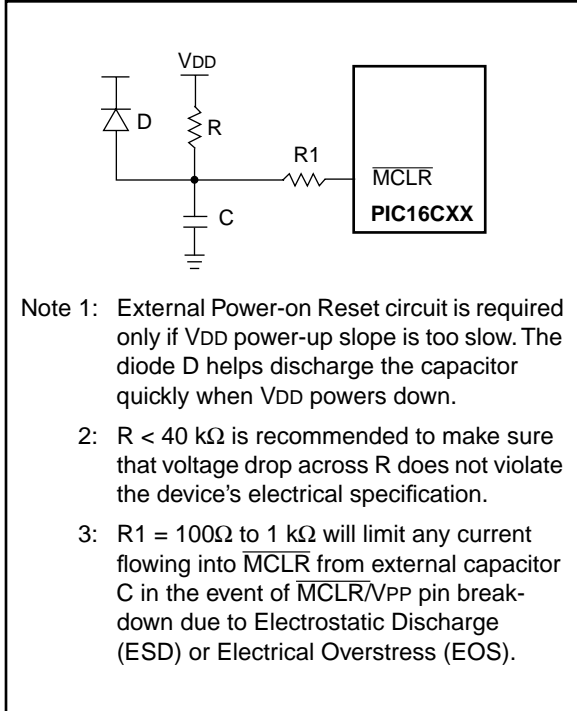


**FIGURE 14-12: TIME-OUT SEQUENCE ON POWER-UP ( $\overline{\text{MCLR}}$  TIED TO  $V_{\text{DD}}$ )**

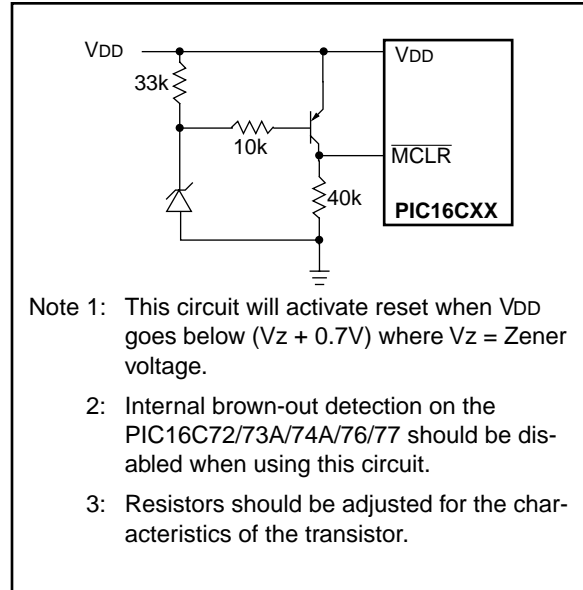


# PIC16C7X

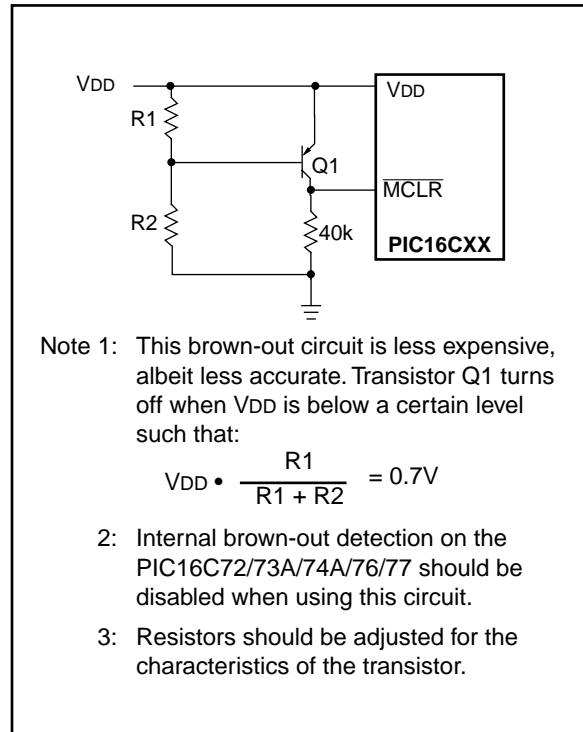
**FIGURE 14-13: EXTERNAL POWER-ON RESET CIRCUIT (FOR SLOW V<sub>DD</sub> POWER-UP)**



**FIGURE 14-14: EXTERNAL BROWN-OUT PROTECTION CIRCUIT 1**



**FIGURE 14-15: EXTERNAL BROWN-OUT PROTECTION CIRCUIT 2**



## 14.5 Interrupts

Applicable Devices					
72	73	73A	74	74A	76/77

The PIC16C7X 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 un-masked 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 (Figure 14-17). The latency is the same for one or two cycle

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

**Note:** For the PIC16C73/74, if an interrupt occurs while the Global Interrupt Enable (GIE) bit is being cleared, the GIE bit may unintentionally be re-enabled by the user's Interrupt Service Routine (the `RETFIE` instruction). The events that would cause this to occur are:

1. An instruction clears the GIE bit while an interrupt is acknowledged.
2. The program branches to the Interrupt vector and executes the Interrupt Service Routine.
3. The Interrupt Service Routine completes with the execution of the `RETFIE` instruction. This causes the GIE bit to be set (enables interrupts), and the program returns to the instruction after the one which was meant to disable interrupts.

Perform the following to ensure that interrupts are globally disabled:

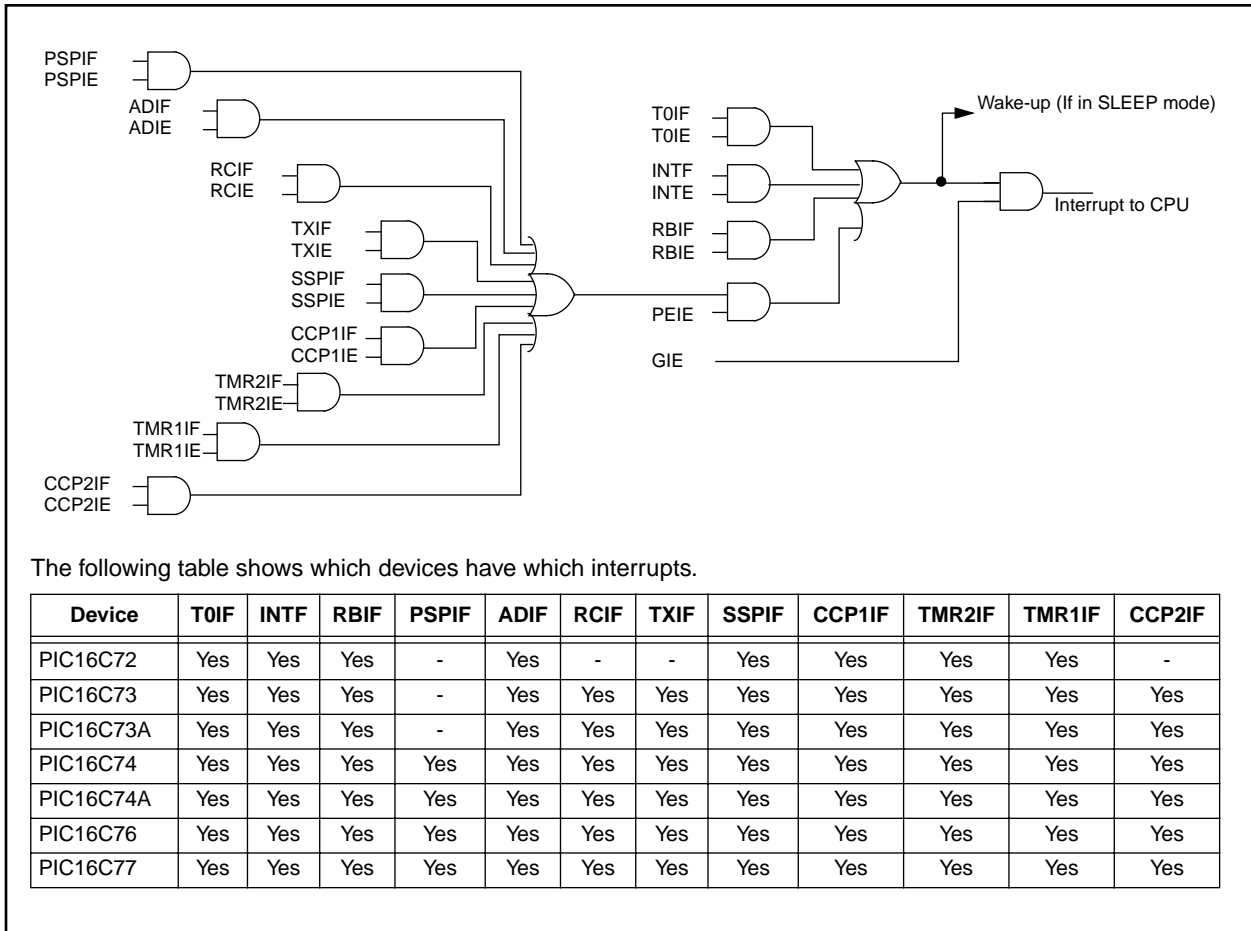
```

LOOP BCF    INTCON, GIE    ; Disable global
                                ; interrupt bit
      BTFSC INTCON, GIE    ; Global interrupt
                                ; disabled?
      GOTO  LOOP           ; NO, try again
      :                   ; Yes, continue
                                ; with program
                                ; flow

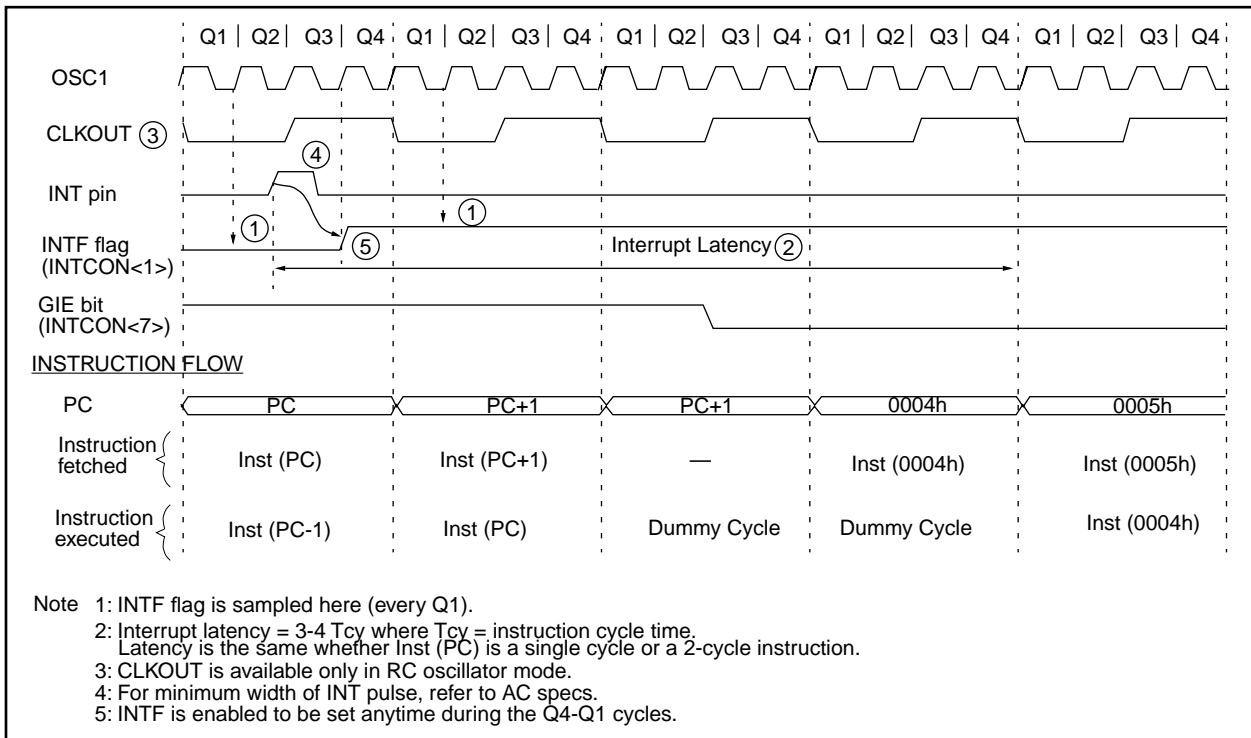
```

# PIC16C7X

**FIGURE 14-16: INTERRUPT LOGIC**



**FIGURE 14-17: INT PIN INTERRUPT TIMING**



## 14.5.1 INT INTERRUPT

External interrupt on RB0/INT pin is edge triggered: either rising if bit INTEDG (OPTION<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 14.8 for details on SLEEP mode.

## 14.5.2 TMR0 INTERRUPT

An overflow (FFh → 00h) in the TMR0 register will set flag bit T0IF (INTCON<2>). The interrupt can be enabled/disabled by setting/clearing enable bit T0IE (INTCON<5>). (Section 7.0)

## 14.5.3 PORTB INTCON CHANGE

An input change on PORTB<7:4> sets flag bit RBIF (INTCON<0>). The interrupt can be enabled/disabled by setting/clearing enable bit RBIE (INTCON<4>). (Section 5.2)

**Note:** For the PIC16C73/74, if a change on the I/O pin should occur when the read operation is being executed (start of the Q2 cycle), then the RBIF interrupt flag may not get set.

## 14.6 Context Saving During Interrupts

### Applicable Devices

72	73	73A	74	74A	76	77
----	----	-----	----	-----	----	----

During an interrupt, only the return PC value is saved on the stack. Typically, users may wish to save key registers during an interrupt i.e., W register and STATUS register. This will have to be implemented in software.

Example 14-1 stores and restores the STATUS, W, and PCLATH registers. The register, W\_TEMP, must be defined in each bank and must be defined at the same offset from the bank base address (i.e., if W\_TEMP is defined at 0x20 in bank 0, it must also be defined at 0xA0 in bank 1).

The example:

- a) Stores the W register.
- b) Stores the STATUS register in bank 0.
- c) Stores the PCLATH register.
- d) Executes the ISR code.
- e) Restores the STATUS register (and bank select bit).
- f) Restores the W and PCLATH registers.

### EXAMPLE 14-1: SAVING STATUS, W, AND PCLATH REGISTERS IN RAM

```

MOVWF    W_TEMP                ;Copy W to TEMP register, could be bank one or zero
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
BCF     STATUS, IRP            ;Return to Bank 0
MOVF    FSR, W                 ;Copy FSR to W
MOVWF   FSR_TEMP              ;Copy FSR from W to FSR_TEMP
:
:(ISR)
:
MOVF    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

```

# PIC16C7X

## 14.7 Watchdog Timer (WDT)

Applicable Devices				
72	73	73A	74	74A
76	77			

The Watchdog Timer is as 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 WDT can be permanently disabled by clearing configuration bit WDTE (Section 14.1).

### 14.7.1 WDT PERIOD

The WDT has a nominal time-out period of 18 ms, (with no prescaler). The time-out periods vary with temperature, VDD and process variations from part to part (see DC specs). If longer time-out periods are desired, a

prescaler with a division ratio of up to 1:128 can be assigned to the WDT under software control by writing to the OPTION register. Thus, time-out periods up to 2.3 seconds can be realized.

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.

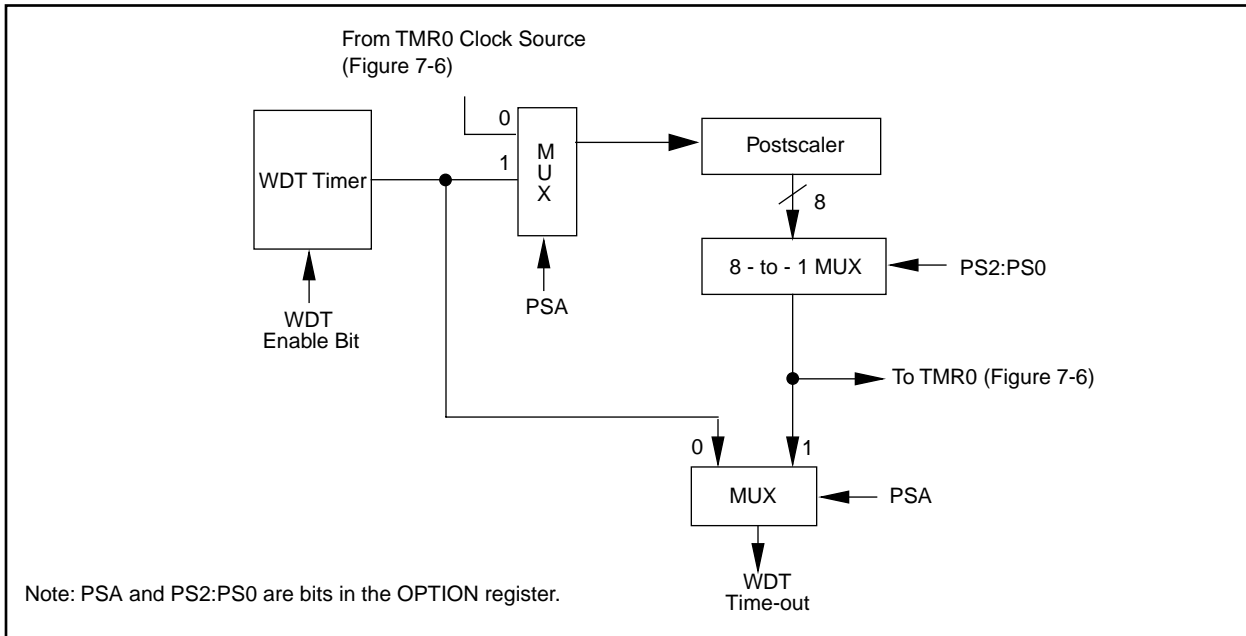
The  $\overline{TO}$  bit in the STATUS register will be cleared upon a Watchdog Timer time-out.

### 14.7.2 WDT PROGRAMMING CONSIDERATIONS

It should also be taken into account that under worst case conditions (VDD = Min., Temperature = Max., and max. WDT prescaler) it may take several seconds before a WDT time-out occurs.

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

**FIGURE 14-18: WATCHDOG TIMER BLOCK DIAGRAM**



**FIGURE 14-19: 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 <sup>(1)</sup>	CP1	CP0	PWRTE <sup>(1)</sup>	WDTE	FOSC1	FOSC0
81h,181h	OPTION	RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0

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

Note 1: See Figure 14-1, and Figure 14-2 for operation of these bits.



## 14.8 Power-down Mode (SLEEP)

Applicable Devices					
72	73	73A	74	74A	76/77

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 VDD, or VSS, ensure no external circuitry is drawing current from the I/O pin, power-down the A/D, disable external clocks. Pull all I/O pins, that are hi-impedance inputs, high or low externally to avoid switching currents caused by floating inputs. The T0CKI input should also be at VDD or VSS for lowest current consumption. The contribution from on-chip pull-ups on PORTB should be considered.

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

### 14.8.1 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 some Peripheral Interrupts.

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. TMR1 interrupt. Timer1 must be operating as an asynchronous counter.
2. SSP (Start/Stop) bit detect interrupt.
3. SSP transmit or receive in slave mode (SPI/I<sup>2</sup>C).
4. CCP capture mode interrupt.
5. Parallel Slave Port read or write.
6. A/D conversion (when A/D clock source is RC).
7. Special event trigger (Timer1 in asynchronous mode using an external clock).
8. USART TX or RX (synchronous slave mode).

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

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

### 14.8.2 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:

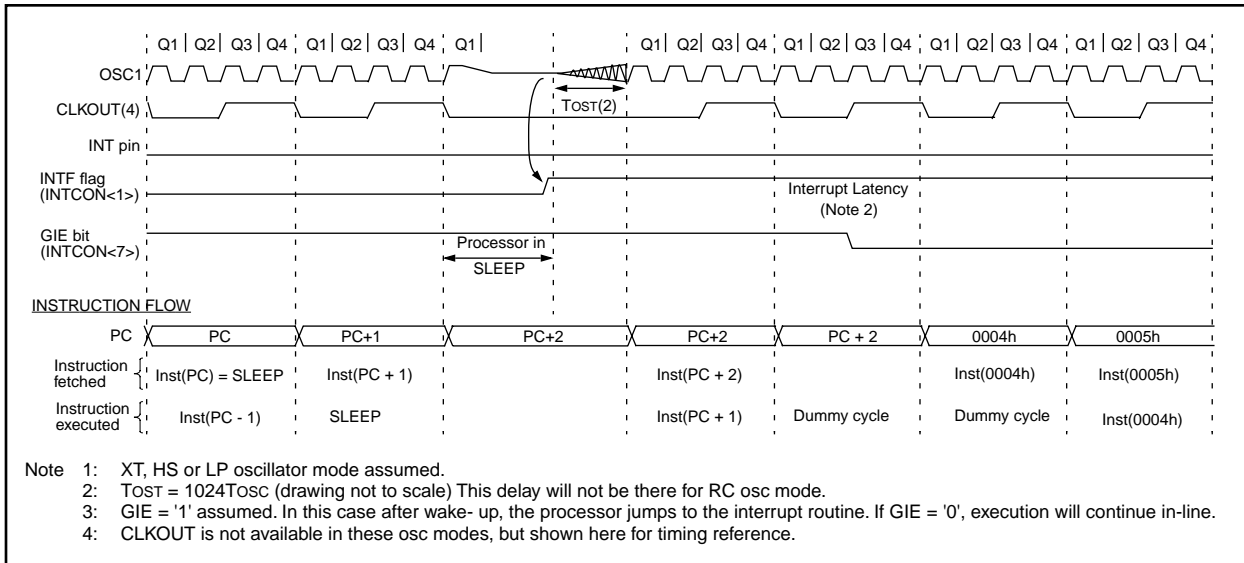
- If the interrupt occurs **before** the execution of a SLEEP instruction, the SLEEP instruction will complete as a NOP. Therefore, the WDT and WDT postscaler will not be cleared, the  $\overline{TO}$  bit will not be set and  $\overline{PD}$  bits will not be cleared.
- If the interrupt occurs **during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

# PIC16C7X

**FIGURE 14-20: WAKE-UP FROM SLEEP THROUGH INTERRUPT**



## 14.9 Program Verification/Code Protection

Applicable Devices
72 73 73A 74 74A 76 77

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

**Note:** Microchip does not recommend code protecting windowed devices.

## 14.10 ID Locations

Applicable Devices
72 73 73A 74 74A 76 77

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.

## 14.11 In-Circuit Serial Programming

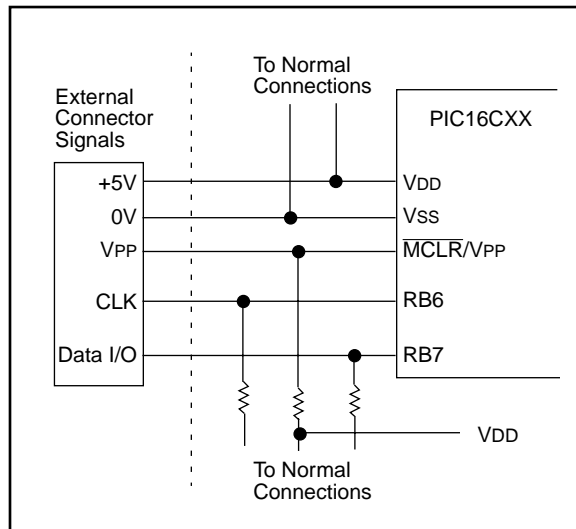
Applicable Devices
72 73 73A 74 74A 76 77

PIC16CXX microcontrollers can be serially programmed while in the end application circuit. This is simply done with two lines for clock and data, and three other lines for power, ground, and the programming voltage. This allows customers to manufacture boards with unprogrammed devices, and then program the microcontroller just before shipping the product. This also allows the most recent firmware or a custom firmware to be programmed.

The device is placed into a program/verify mode by holding the RB6 and RB7 pins low while raising the MCLR (VPP) pin from  $V_{IL}$  to  $V_{IH}$  (see programming specification). RB6 becomes the programming clock and RB7 becomes the programming data. Both RB6 and RB7 are Schmitt Trigger inputs in this mode.

After reset, to place the device into programming/verify mode, the program counter (PC) is at location 00h. A 6-bit command is then supplied to the device. Depending on the command, 14-bits of program data are then supplied to or from the device, depending if the command was a load or a read. For complete details of serial programming, please refer to the PIC16C6X/7X Programming Specifications (Literature #DS30228).

**FIGURE 14-21: TYPICAL IN-CIRCUIT SERIAL PROGRAMMING CONNECTION**



## 15.0 INSTRUCTION SET SUMMARY

Each PIC16CXX 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 PIC16CXX instruction set summary in Table 15-2 lists **byte-oriented**, **bit-oriented**, and **literal and control** operations. Table 15-1 shows the opcode field descriptions.

For **byte-oriented** instructions, 'f' represents a file register designator and 'd' represents a destination designator. The file register designator specifies which file register is to be used by the instruction.

The destination designator specifies where the result of the operation is to be placed. If 'd' is zero, the result is placed in the W register. If 'd' is one, the result is placed in the file register specified in the instruction.

For **bit-oriented** instructions, 'b' represents a bit field designator which selects the number of the bit affected by the operation, while 'f' represents the number of the file in which the bit is located.

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

**TABLE 15-1: 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
label	Label name
TOS	Top of Stack
PC	Program Counter
PCLATH	Program Counter High Latch
GIE	Global Interrupt Enable bit
WDT	Watchdog Timer/Counter
TO	Time-out bit
PD	Power-down bit
dest	Destination either the W register or the specified register file location
[ ]	Options
( )	Contents
→	Assigned to
< >	Register bit field
∈	In the set of
<i>italics</i>	User defined term (font is courier)

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

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

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

Table 15-2 lists the instructions recognized by the MPASM assembler.

Figure 15-1 shows the general formats that the instructions can have.

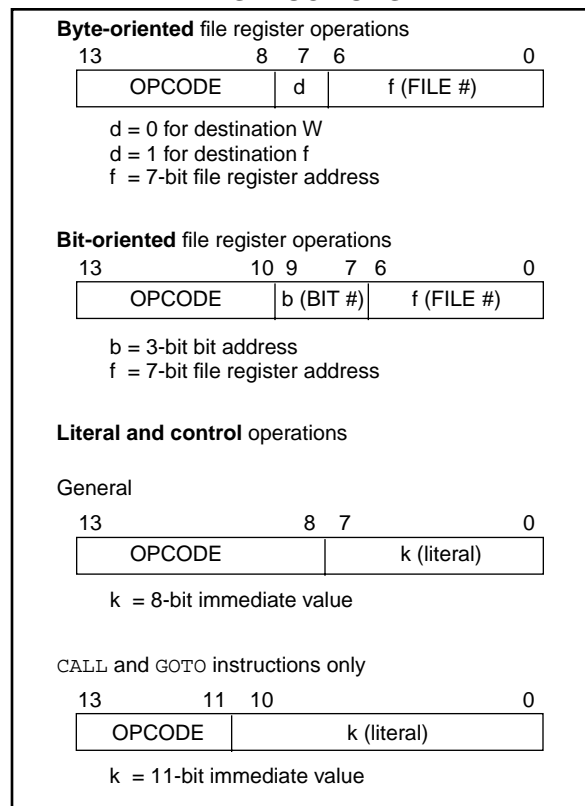
**Note:** To maintain upward compatibility with future PIC16CXX products, do not use the `OPTION` and `TRIS` instructions.

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

0xhh

where h signifies a hexadecimal digit.

**FIGURE 15-1: GENERAL FORMAT FOR INSTRUCTIONS**



# PIC16C7X

TABLE 15-2: PIC16CXX INSTRUCTION SET

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

Note 1: When an I/O register is modified as a function of itself ( e.g., `MOVF PORTB, 1`), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as input and is driven low by an external device, the data will be written back with a '0'.

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.

# PIC16C7X

## 15.1 Instruction Descriptions

**ADDLW Add Literal and W**

Syntax: `[label] ADDLW k`

Operands:  $0 \leq k \leq 255$

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

Status Affected: C, DC, Z

Encoding: 

11	111x	kkkk	kkkk
----	------	------	------

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

Words: 1

Cycles: 1

Q Cycle Activity: 

Q1	Q2	Q3	Q4
Decode	Read literal 'k'	Process data	Write to W

Example: `ADDLW 0x15`

Before Instruction  
W = 0x10

After Instruction  
W = 0x25

**ANDLW AND Literal with W**

Syntax: `[label] ANDLW k`

Operands:  $0 \leq k \leq 255$

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

Status Affected: Z

Encoding: 

11	1001	kkkk	kkkk
----	------	------	------

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

Words: 1

Cycles: 1

Q Cycle Activity: 

Q1	Q2	Q3	Q4
Decode	Read literal "k"	Process data	Write to W

Example: `ANDLW 0x5F`

Before Instruction  
W = 0xA3

After Instruction  
W = 0x03

**ADDWF Add W and f**

Syntax: `[label] ADDWF f,d`

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

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

Status Affected: C, DC, Z

Encoding: 

00	0111	dfff	ffff
----	------	------	------

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'.

Words: 1

Cycles: 1

Q Cycle Activity: 

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process data	Write to destination

Example: `ADDWF FSR, 0`

Before Instruction  
W = 0x17  
FSR = 0xC2

After Instruction  
W = 0xD9  
FSR = 0xC2

**ANDWF AND W with f**

Syntax: `[label] ANDWF f,d`

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

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

Status Affected: Z

Encoding: 

00	0101	dfff	ffff
----	------	------	------

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'.

Words: 1

Cycles: 1

Q Cycle Activity: 

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process data	Write to destination

Example: `ANDWF FSR, 1`

Before Instruction  
W = 0x17  
FSR = 0xC2

After Instruction  
W = 0x17  
FSR = 0x02

# PIC16C7X

BCF	Bit Clear f								
Syntax:	<code>[label] BCF f,b</code>								
Operands:	$0 \leq f \leq 127$ $0 \leq b \leq 7$								
Operation:	$0 \rightarrow (f<b>)$								
Status Affected:	None								
Encoding:	<table border="1"> <tr> <td>01</td> <td>00bb</td> <td>bfff</td> <td>ffff</td> </tr> </table>	01	00bb	bfff	ffff				
01	00bb	bfff	ffff						
Description:	Bit 'b' in register 'f' is cleared.								
Words:	1								
Cycles:	1								
Q Cycle Activity:	<table border="1"> <thead> <tr> <th>Q1</th> <th>Q2</th> <th>Q3</th> <th>Q4</th> </tr> </thead> <tbody> <tr> <td>Decode</td> <td>Read register 'f'</td> <td>Process data</td> <td>Write register 'f'</td> </tr> </tbody> </table>	Q1	Q2	Q3	Q4	Decode	Read register 'f'	Process data	Write register 'f'
Q1	Q2	Q3	Q4						
Decode	Read register 'f'	Process data	Write register 'f'						

Example

```

BCF    FLAG_REG, 7
Before Instruction
      FLAG_REG = 0xC7
After Instruction
      FLAG_REG = 0x47
  
```

BTFSF	Bit Test, Skip if Clear																
Syntax:	<code>[label] BTFSF f,b</code>																
Operands:	$0 \leq f \leq 127$ $0 \leq b \leq 7$																
Operation:	skip if $(f<b>) = 0$																
Status Affected:	None																
Encoding:	<table border="1"> <tr> <td>01</td> <td>10bb</td> <td>bfff</td> <td>ffff</td> </tr> </table>	01	10bb	bfff	ffff												
01	10bb	bfff	ffff														
Description:	If bit 'b' in register 'f' is '1' then the next instruction is executed. If bit 'b', in register 'f', is '0' then the next instruction is discarded, and a NOP is executed instead, making this a 2TCY instruction.																
Words:	1																
Cycles:	1(2)																
Q Cycle Activity:	<table border="1"> <thead> <tr> <th>Q1</th> <th>Q2</th> <th>Q3</th> <th>Q4</th> </tr> </thead> <tbody> <tr> <td>Decode</td> <td>Read register 'f'</td> <td>Process data</td> <td>No-Operation</td> </tr> </tbody> </table> <p>If Skip: (2nd Cycle)</p> <table border="1"> <thead> <tr> <th>Q1</th> <th>Q2</th> <th>Q3</th> <th>Q4</th> </tr> </thead> <tbody> <tr> <td>No-Operation</td> <td>No-Operation</td> <td>No-Operation</td> <td>No-Operation</td> </tr> </tbody> </table>	Q1	Q2	Q3	Q4	Decode	Read register 'f'	Process data	No-Operation	Q1	Q2	Q3	Q4	No-Operation	No-Operation	No-Operation	No-Operation
Q1	Q2	Q3	Q4														
Decode	Read register 'f'	Process data	No-Operation														
Q1	Q2	Q3	Q4														
No-Operation	No-Operation	No-Operation	No-Operation														

Example

```

HERE   BTFSF  FLAG, 1
FALSE  GOTO  PROCESS_CODE
TRUE   :
       :
       :
Before Instruction
      PC = address HERE
After Instruction
      if FLAG<1> = 0,
      PC = address TRUE
      if FLAG<1> = 1,
      PC = address FALSE
  
```

BSF	Bit Set f								
Syntax:	<code>[label] BSF f,b</code>								
Operands:	$0 \leq f \leq 127$ $0 \leq b \leq 7$								
Operation:	$1 \rightarrow (f<b>)$								
Status Affected:	None								
Encoding:	<table border="1"> <tr> <td>01</td> <td>01bb</td> <td>bfff</td> <td>ffff</td> </tr> </table>	01	01bb	bfff	ffff				
01	01bb	bfff	ffff						
Description:	Bit 'b' in register 'f' is set.								
Words:	1								
Cycles:	1								
Q Cycle Activity:	<table border="1"> <thead> <tr> <th>Q1</th> <th>Q2</th> <th>Q3</th> <th>Q4</th> </tr> </thead> <tbody> <tr> <td>Decode</td> <td>Read register 'f'</td> <td>Process data</td> <td>Write register 'f'</td> </tr> </tbody> </table>	Q1	Q2	Q3	Q4	Decode	Read register 'f'	Process data	Write register 'f'
Q1	Q2	Q3	Q4						
Decode	Read register 'f'	Process data	Write register 'f'						

Example

```

BSF    FLAG_REG, 7
Before Instruction
      FLAG_REG = 0x0A
After Instruction
      FLAG_REG = 0x8A
  
```

# PIC16C7X

## BTFSS Bit Test f, Skip if Set

Syntax: `[label] BTFSS f,b`

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

Operation: skip if  $(f < b) = 1$

Status Affected: None

Encoding: 

01	11bb	bfff	ffff
----	------	------	------

Description: If bit 'b' in register 'f' is '0' then 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.

Words: 1

Cycles: 1(2)

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process data	No-Operation

If Skip: (2nd Cycle)

Q1	Q2	Q3	Q4
No-Operation	No-Operation	No-Operation	No-Operation

**Example**

```

HERE   BTFSC  FLAG, 1
FALSE  GOTO  PROCESS_CODE
TRUE   •
        •
        •

```

Before Instruction  
PC = address HERE

After Instruction  
if FLAG<1> = 0,  
PC = address FALSE  
if FLAG<1> = 1,  
PC = address TRUE

## CALL Call Subroutine

Syntax: `[label] CALL k`

Operands:  $0 \leq k \leq 2047$

Operation:  $(PC)+1 \rightarrow TOS$ ,  
 $k \rightarrow PC < 10:0 >$ ,  
 $(PCLATH < 4:3 >) \rightarrow PC < 12:11 >$

Status Affected: None

Encoding: 

10	0kkk	kkkk	kkkk
----	------	------	------

Description: Call Subroutine. First, return address (PC+1) is pushed onto the stack. The eleven bit immediate address is loaded into PC bits <10:0>. The upper bits of the PC are loaded from PCLATH. CALL is a two cycle instruction.

Words: 1

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'k', Push PC to Stack	Process data	Write to PC
No-Operation	No-Operation	No-Operation	No-Operation

**Example**

```

HERE   CALL  THERE

```

Before Instruction  
PC = Address HERE

After Instruction  
PC = Address THERE  
TOS = Address HERE+1

# PIC16C7X

## CLRF Clear f

Syntax: `[label] CLRF f`

Operands:  $0 \leq f \leq 127$

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

Status Affected: Z

Encoding: 

00	0001	1fff	ffff
----	------	------	------

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

Words: 1

Cycles: 1

Q Cycle Activity: 

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process data	Write register 'f'

Example

```

CLRF    FLAG_REG
Before Instruction
FLAG_REG = 0x5A
After Instruction
FLAG_REG = 0x00
Z        = 1
    
```

## CLRW Clear W

Syntax: `[label] CLRW`

Operands: None

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

Status Affected: Z

Encoding: 

00	0001	0xxx	xxxx
----	------	------	------

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

Words: 1

Cycles: 1

Q Cycle Activity: 

Q1	Q2	Q3	Q4
Decode	No-Operation	Process data	Write to W

Example

```

CLRW
Before Instruction
W = 0x5A
After Instruction
W = 0x00
Z = 1
    
```

## CLRWDT Clear Watchdog Timer

Syntax: `[label] CLRWDT`

Operands: None

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

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

Encoding: 

00	0000	0110	0100
----	------	------	------

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

Words: 1

Cycles: 1

Q Cycle Activity: 

Q1	Q2	Q3	Q4
Decode	No-Operation	Process data	Clear WDT Counter

Example

```

CLRWDT
Before Instruction
WDT counter = ?
After Instruction
WDT counter = 0x00
WDT prescaler = 0
 $\overline{TO}$  = 1
 $\overline{PD}$  = 1
    
```



# PIC16C7X

**COMF**                      **Complement f**

---

Syntax:                      [label] COMF f,d

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

Operation:                   (f̄) → (destination)

Status Affected:           Z

Encoding:                   

00	1001	dfff	ffff
----	------	------	------

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

Words:                        1

Cycles:                        1

Q Cycle Activity:            Q1        Q2        Q3        Q4

Decode	Read register 'f'	Process data	Write to destination
--------	-------------------	--------------	----------------------

**Example**

```

COMF    REG1, 0
Before Instruction
    REG1 = 0x13
After Instruction
    REG1 = 0x13
    W    = 0xEC
  
```

**DECF**                      **Decrement f**

---

Syntax:                      [label] DECF f,d

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

Operation:                   (f - 1) → (destination)

Status Affected:           Z

Encoding:                   

00	0011	dfff	ffff
----	------	------	------

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'.

Words:                        1

Cycles:                        1

Q Cycle Activity:            Q1        Q2        Q3        Q4

Decode	Read register 'f'	Process data	Write to destination
--------	-------------------	--------------	----------------------

**Example**

```

DECF    CNT, 1
Before Instruction
    CNT = 0x01
    Z   = 0
After Instruction
    CNT = 0x00
    Z   = 1
  
```

**DECFSZ**                    **Decrement f, Skip if 0**

---

Syntax:                      [label] DECFSZ f,d

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

Operation:                   (f - 1) → (destination);  
                                  skip if result = 0

Status Affected:           None

Encoding:                   

00	1011	dfff	ffff
----	------	------	------

Description:                The contents of register 'f' are decremented. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is placed back in register 'f'.  
If the result is 1, the next instruction, is executed. If the result is 0, then a NOP is executed instead making it a 2Tcy instruction.

Words:                        1

Cycles:                        1(2)

Q Cycle Activity:            Q1        Q2        Q3        Q4

Decode	Read register 'f'	Process data	Write to destination
--------	-------------------	--------------	----------------------

If Skip: (2nd Cycle)

Q1	Q2	Q3	Q4
No-Operation	No-Operation	No-Operation	No-Operation

**Example**

```

HERE    DECFSZ  CNT, 1
        GOTO    LOOP
CONTINUE •
        •
        •
Before Instruction
    PC = address HERE
After Instruction
    CNT = CNT - 1
    if CNT = 0,
    PC = address CONTINUE
    if CNT ≠ 0,
    PC = address HERE+1
  
```

# PIC16C7X

**GOTO**                      **Unconditional Branch**

---

Syntax:                      [*label*] GOTO *k*

Operands:                     $0 \leq k \leq 2047$

Operation:                    $k \rightarrow PC<10:0>$   
 $PCLATH<4:3> \rightarrow PC<12:11>$

Status Affected:          None

Encoding:                    

10	1kkk	kkkk	kkkk
----	------	------	------

Description:                GOTO is an unconditional branch. The eleven bit immediate value is loaded into PC bits <10:0>. The upper bits of PC are loaded from PCLATH<4:3>. GOTO is a two cycle instruction.

Words:                        1

Cycles:                        2

Q Cycle Activity:            

	Q1	Q2	Q3	Q4
1st Cycle	Decode	Read literal 'k'	Process data	Write to PC
2nd Cycle	No-Operation	No-Operation	No-Operation	No-Operation

Example                      GOTO THERE

                                  After Instruction

                                  PC = Address THERE

**INCF**                        **Increment f**

---

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

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

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

Status Affected:          Z

Encoding:                    

00	1010	dfff	ffff
----	------	------	------

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

Words:                        1

Cycles:                        1

Q Cycle Activity:            

	Q1	Q2	Q3	Q4
	Decode	Read register 'f'	Process data	Write to destination

Example                      INCF CNT, 1

                                  Before Instruction

                                  CNT = 0xFF

                                  Z = 0

                                  After Instruction

                                  CNT = 0x00

                                  Z = 1

# PIC16C7X

**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

Encoding:    

00	1111	dfff	ffff
----	------	------	------

Description: The contents of register 'f' are incremented. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is placed back in register 'f'. If the result is 1, the next instruction is executed. If the result is 0, a NOP is executed instead making it a 2TCY instruction.

Words:      1

Cycles:      1(2)

Q Cycle Activity:      Q1      Q2      Q3      Q4

Decode	Read register 'f'	Process data	Write to destination
--------	-------------------	--------------	----------------------

If Skip: (2nd Cycle)

No-Operation	No-Operation	No-Operation	No-Operation
--------------	--------------	--------------	--------------

Example

```

HERE      INCFSZ      CNT, 1
            GOTO      LOOP
CONTINUE •
            •
            •
  
```

Before Instruction  
 PC = address HERE

After Instruction  
 CNT = CNT + 1  
 if CNT= 0,  
 PC = address CONTINUE  
 if CNT≠ 0,  
 PC = address HERE +1

**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

Encoding:    

11	1000	kkkk	kkkk
----	------	------	------

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

Words:      1

Cycles:      1

Q Cycle Activity:      Q1      Q2      Q3      Q4

Decode	Read literal 'k'	Process data	Write to W
--------	------------------	--------------	------------

Example

```

IORLW      0x35
  
```

Before Instruction  
 W = 0x9A

After Instruction  
 W = 0xBF  
 Z = 1

# PIC16C7X

**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

Encoding:     

00	0100	dfff	ffff
----	------	------	------

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

Words:        1

Cycles:        1

Q Cycle Activity:    

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process data	Write to destination

Example        `IORWF        RESULT, 0`

Before Instruction

RESULT = 0x13  
W = 0x91

After Instruction

RESULT = 0x13  
W = 0x93  
Z = 1

**MOVF**        **Move f**

---

Syntax:        `[label] MOVF f,d`

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

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

Status Affected: Z

Encoding:     

00	1000	dfff	ffff
----	------	------	------

Description:   The contents of register f is 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.

Words:        1

Cycles:        1

Q Cycle Activity:    

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process data	Write to destination

Example        `MOVF        FSR, 0`

After Instruction

W = value in FSR register  
Z = 1

**MOVLW**      **Move Literal to W**

---

Syntax:        `[label] MOVLW k`

Operands:      $0 \leq k \leq 255$

Operation:     $k \rightarrow (W)$

Status Affected: None

Encoding:     

11	00xx	kkkk	kkkk
----	------	------	------

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

Words:        1

Cycles:        1

Q Cycle Activity:    

Q1	Q2	Q3	Q4
Decode	Read literal 'k'	Process data	Write to W

Example        `MOVLW    0x5A`

After Instruction

W = 0x5A

**MOVWF**      **Move W to f**

---

Syntax:        `[label] MOVWF f`

Operands:      $0 \leq f \leq 127$

Operation:     $(W) \rightarrow (f)$

Status Affected: None

Encoding:     

00	0000	1fff	ffff
----	------	------	------

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

Words:        1

Cycles:        1

Q Cycle Activity:    

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process data	Write register 'f'

Example        `MOVWF    OPTION_REG`

Before Instruction

OPTION = 0xFF  
W = 0x4F

After Instruction

OPTION = 0x4F  
W = 0x4F

# PIC16C7X

<b>NOP</b>	<b>No Operation</b>			
Syntax:	[ <i>label</i> ] NOP			
Operands:	None			
Operation:	No operation			
Status Affected:	None			
Encoding:	00	0000	0xx0	0000
Description:	No operation.			
Words:	1			
Cycles:	1			
Q Cycle Activity:	Q1	Q2	Q3	Q4
	Decode	No-Operation	No-Operation	No-Operation

Example            NOP

<b>RETFIE</b>	<b>Return from Interrupt</b>			
Syntax:	[ <i>label</i> ] RETFIE			
Operands:	None			
Operation:	TOS → PC, 1 → GIE			
Status Affected:	None			
Encoding:	00	0000	0000	1001
Description:	Return from Interrupt. Stack is POPed and Top of Stack (TOS) is loaded in the PC. Interrupts are enabled by setting Global Interrupt Enable bit, GIE (INTCON<7>). This is a two cycle instruction.			
Words:	1			
Cycles:	2			
Q Cycle Activity:	Q1	Q2	Q3	Q4
1st Cycle	Decode	No-Operation	Set the GIE bit	Pop from the Stack
2nd Cycle	No-Operation	No-Operation	No-Operation	No-Operation

Example            RETFIE  
 After Interrupt  
                   PC = TOS  
                   GIE = 1

<b>OPTION</b>	<b>Load Option Register</b>			
Syntax:	[ <i>label</i> ] OPTION			
Operands:	None			
Operation:	(W) → OPTION			
Status Affected:	None			
Encoding:	00	0000	0110	0010
Description:	The contents of the W register are loaded in the OPTION register. This instruction is supported for code compatibility with PIC16C5X products. Since OPTION is a readable/writable register, the user can directly address it.			
Words:	1			
Cycles:	1			
Example	<b>To maintain upward compatibility with future PIC16CXX products, do not use this instruction.</b>			

# PIC16C7X

## 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

Encoding:        

11	01xx	kkkk	kkkk
----	------	------	------

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.

Words:            1

Cycles:            2

Q Cycle Activity:    

Q1	Q2	Q3	Q4
----	----	----	----

	1st Cycle	Decode	Read literal 'k'	No-Operation	Write to W, Pop from the Stack
	2nd Cycle	No-Operation	No-Operation	No-Operation	No-Operation

Example

```
CALL TABLE ;W contains table
           ;offset value
           ;W now has table value
•
•
•
TABLE ADDWF PC ;W = offset
      RETLW k1 ;Begin table
      RETLW k2 ;
      •
      •
      RETLW kn ; End of table
```

Before Instruction  
W = 0x07

After Instruction  
W = value of k8

## RETURN      Return from Subroutine

Syntax:            [ *label* ] RETURN

Operands:        None

Operation:         $TOS \rightarrow PC$

Status Affected:    None

Encoding:        

00	0000	0000	1000
----	------	------	------

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.

Words:            1

Cycles:            2

Q Cycle Activity:    

Q1	Q2	Q3	Q4
----	----	----	----

	1st Cycle	Decode	No-Operation	No-Operation	Pop from the Stack
	2nd Cycle	No-Operation	No-Operation	No-Operation	No-Operation

Example

```
RETURN
After Interrupt
PC = TOS
```

# PIC16C7X

**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

Encoding: 

00	1101	dfff	ffff
----	------	------	------

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'.

Words: 1

Cycles: 1

Q Cycle Activity:

	Q1	Q2	Q3	Q4
Decode				
Read register 'f'				
Process data				
Write to destination				

**Example**

```

RLF      REG1,0

Before Instruction
REG1    = 1110 0110
C       = 0

After Instruction
REG1    = 1110 0110
W       = 1100 1100
C       = 1
  
```

**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

Encoding: 

00	1100	dfff	ffff
----	------	------	------

Description: The contents of register 'f' are rotated one bit to the right through the Carry Flag. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is placed back in register 'f'.

Words: 1

Cycles: 1

Q Cycle Activity:

	Q1	Q2	Q3	Q4
Decode				
Read register 'f'				
Process data				
Write to destination				

**Example**

```

RRF      REG1,0

Before Instruction
REG1    = 1110 0110
C       = 0

After Instruction
REG1    = 1110 0110
W       = 0111 0011
C       = 0
  
```

# PIC16C7X

## SLEEP

Syntax: [ *label* ] SLEEP

Operands: None

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

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

Encoding: 

00	0000	0110	0011
----	------	------	------

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. See Section 14.8 for more details.

Words: 1

Cycles: 1

Q Cycle Activity: 

	Q1	Q2	Q3	Q4
Decode		No-Operation	No-Operation	Go to Sleep

Example: SLEEP

## 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

Encoding: 

11	110x	kkkk	kkkk
----	------	------	------

Description: The W register is subtracted (2's complement method) from the eight bit literal 'k'. The result is placed in the W register.

Words: 1

Cycles: 1

Q Cycle Activity: 

	Q1	Q2	Q3	Q4
Decode		Read literal 'k'	Process data	Write to W

Example 1: SUBLW 0x02

Before Instruction

W = 1  
C = ?  
Z = ?

After Instruction

W = 1  
C = 1; result is positive  
Z = 0

Example 2: Before Instruction

W = 2  
C = ?  
Z = ?

After Instruction

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

Example 3: Before Instruction

W = 3  
C = ?  
Z = ?

After Instruction

W = 0xFF  
C = 0; result is negative  
Z = 0



# PIC16C7X

## **SUBWF**      **Subtract W from f**

Syntax:            [ *label* ]    SUBWF f,d

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

Operation:        (f) - (W) → (destination)

Status Affected: C, DC, Z

Encoding:        

00	0010	dfff	ffff
----	------	------	------

Description:     Subtract (2's complement method) W register from register 'f'. If 'd' is 0 the result is stored in the W register. If 'd' is 1 the result is stored back in register 'f'.

Words:            1

Cycles:            1

Q Cycle Activity:    

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process data	Write to destination

Example 1:        SUBWF    REG1, 1

Before Instruction

REG1    =    3  
W        =    2  
C        =    ?  
Z        =    ?

After Instruction

REG1    =    1  
W        =    2  
C        =    1; result is positive  
Z        =    0

Example 2:        Before Instruction

REG1    =    2  
W        =    2  
C        =    ?  
Z        =    ?

After Instruction

REG1    =    0  
W        =    2  
C        =    1; result is zero  
Z        =    1

Example 3:        Before Instruction

REG1    =    1  
W        =    2  
C        =    ?  
Z        =    ?

After Instruction

REG1    =    0xFF  
W        =    2  
C        =    0; result is negative  
Z        =    0

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

Syntax:            [ *label* ]    SWAPF f,d

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

Operation:        (f<3:0>) → (destination<7:4>),  
(f<7:4>) → (destination<3:0>)

Status Affected: None

Encoding:        

00	1110	dfff	ffff
----	------	------	------

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

Words:            1

Cycles:            1

Q Cycle Activity:    

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process data	Write to destination

Example            SWAPF REG, 0

Before Instruction

REG1    =    0xA5

After Instruction

REG1    =    0xA5  
W        =    0x5A

## **TRIS**      **Load TRIS Register**

Syntax:            [ *label* ]    TRIS f

Operands:         $5 \leq f \leq 7$

Operation:        (W) → TRIS register f;

Status Affected: None

Encoding:        

00	0000	0110	0fff
----	------	------	------

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

Words:            1

Cycles:            1

Example

**To maintain upward compatibility with future PIC16CXX products, do not use this instruction.**

# PIC16C7X

---

## **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

Encoding: 

11	1010	kkkk	kkkk
----	------	------	------

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

Words: 1

Cycles: 1

Q Cycle Activity: 

Q1	Q2	Q3	Q4
Decode	Read literal 'k'	Process data	Write to W

Example: `XORLW 0xAF`

Before Instruction

W = 0xB5

After Instruction

W = 0x1A

## **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

Encoding: 

00	0110	dfff	ffff
----	------	------	------

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'.

Words: 1

Cycles: 1

Q Cycle Activity: 

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process data	Write to destination

Example `XORWF REG 1`

Before Instruction

REG = 0xAF

W = 0xB5

After Instruction

REG = 0x1A

W = 0xB5

## 16.0 DEVELOPMENT SUPPORT

### 16.1 Development Tools

The PIC16/17 microcontrollers are supported with a full range of hardware and software development tools:

- PICMASTER/PICMASTER CE Real-Time In-Circuit Emulator
- ICEPIC Low-Cost PIC16C5X and PIC16CXXX In-Circuit Emulator
- PRO MATE<sup>®</sup> II Universal Programmer
- PICSTART<sup>®</sup> Plus Entry-Level Prototype Programmer
- PICDEM-1 Low-Cost Demonstration Board
- PICDEM-2 Low-Cost Demonstration Board
- PICDEM-3 Low-Cost Demonstration Board
- MPASM Assembler
- MPLAB-SIM Software Simulator
- MPLAB-C (C Compiler)
- Fuzzy logic development system (*fuzzyTECH<sup>®</sup>-MP*)

### 16.2 PICMASTER: High Performance Universal In-Circuit Emulator with MPLAB IDE

The PICMASTER Universal In-Circuit Emulator is intended to provide the product development engineer with a complete microcontroller design tool set for all microcontrollers in the PIC12C5XX, PIC14C000, PIC16C5X, PIC16CXXX and PIC17CXX families. PICMASTER is supplied with the MPLAB<sup>™</sup> Integrated Development Environment (IDE), which allows editing, "make" and download, and source debugging from a single environment.

Interchangeable target probes allow the system to be easily reconfigured for emulation of different processors. The universal architecture of the PICMASTER allows expansion to support all new Microchip microcontrollers.

The PICMASTER 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 compatible 386 (and higher) machine platform and Microsoft Windows<sup>®</sup> 3.x environment were chosen to best make these features available to you, the end user.

A CE compliant version of PICMASTER is available for European Union (EU) countries.

### 16.3 ICEPIC: Low-cost PIC16CXXX In-Circuit Emulator

ICEPIC is a low-cost in-circuit emulator solution for the Microchip PIC16C5X and PIC16CXXX families of 8-bit OTP microcontrollers.

ICEPIC is designed to operate on PC-compatible machines ranging from 286-AT<sup>®</sup> through Pentium<sup>™</sup> based machines under Windows 3.x environment. ICEPIC features real time, non-intrusive emulation.

### 16.4 PRO MATE II: Universal Programmer

The PRO MATE II Universal Programmer is a full-featured programmer capable of operating in stand-alone mode as well as PC-hosted mode.

The PRO MATE II has programmable VDD and VPP supplies which allows it to verify programmed memory at VDD min and VDD max for maximum reliability. It has an LCD display for displaying 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 can read, verify or program PIC16C5X, PIC16CXXX, PIC17CXX and PIC14000 devices. It can also set configuration and code-protect bits in this mode.

### 16.5 PICSTART Plus Entry Level Development System

The PICSTART programmer is an easy-to-use, low-cost prototype programmer. It connects to the PC via one of the COM (RS-232) ports. MPLAB Integrated Development Environment software makes using the programmer simple and efficient. PICSTART Plus is not recommended for production programming.

PICSTART Plus supports all PIC12C5XX, PIC14000, PIC16C5X, PIC16CXXX and PIC17CXX devices with up to 40 pins. Larger pin count devices such as the PIC16C923 and PIC16C924 may be supported with an adapter socket.

# PIC16C7X

---

## 16.6 PICDEM-1 Low-Cost PIC16/17 Demonstration Board

The PICDEM-1 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 users can program the sample microcontrollers provided with the PICDEM-1 board, on a PRO MATE II or PICSTART-16B programmer, and easily test firmware. The user can also connect the PICDEM-1 board to the PICMASTER emulator and download the firmware to the emulator for testing. Additional 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.

## 16.7 PICDEM-2 Low-Cost PIC16CXX Demonstration Board

The PICDEM-2 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 board, on a PRO MATE II programmer or PICSTART-16C, and easily test firmware. The PICMASTER emulator may also be used with the PICDEM-2 board to test firmware. Additional 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 I<sup>2</sup>C bus and separate headers for connection to an LCD module and a keypad.

## 16.8 PICDEM-3 Low-Cost PIC16CXXX Demonstration Board

The PICDEM-3 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 a 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 board, on a PRO MATE II programmer or PICSTART Plus with an adapter socket, and easily test firmware. The PICMASTER emulator may also be used with the PICDEM-3 board to test firmware. Additional prototype area has been provided to the user for adding hardware and connecting it to the microcontroller socket(s). Some of the features include

an 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 board is an LCD panel, with 4 commons and 12 segments, that is capable of displaying time, temperature and day of the week. The PICDEM-3 provides an additional RS-232 interface and Windows 3.1 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.

## 16.9 MPLAB Integrated Development Environment Software

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

- A full featured editor
- Three operating modes
  - editor
  - emulator
  - simulator
- A project manager
- Customizable tool bar and key mapping
- A status bar with project information
- Extensive on-line help

MPLAB allows you to:

- Edit your source files (either assembly or 'C')
- One touch assemble (or compile) and download to PIC16/17 tools (automatically updates all project information)
- Debug using:
  - source files
  - absolute listing file
- Transfer data dynamically via DDE (soon to be replaced by OLE)
- Run up to four emulators on the same PC

The ability to use MPLAB with Microchip's simulator allows a consistent platform and the ability to easily switch from the low cost simulator to the full featured emulator with minimal retraining due to development tools.

## 16.10 Assembler (MPASM)

The MPASM Universal Macro Assembler is a PC-hosted symbolic assembler. It supports all microcontroller series including the PIC12C5XX, PIC14000, PIC16C5X, PIC16CXXX, and PIC17CXX families.

MPASM offers full featured Macro capabilities, conditional assembly, and several source and listing formats. It generates various object code formats to support Microchip's development tools as well as third party programmers.

MPASM allows full symbolic debugging from PICMASTER, Microchip's Universal Emulator System.

---

MPASM has the following features to assist in developing software for specific use applications.

- Provides translation of Assembler source code to object code for all Microchip microcontrollers.
- Macro assembly capability.
- Produces all the files (Object, Listing, Symbol, and special) required for symbolic debug with Microchip's emulator systems.
- Supports Hex (default), Decimal and Octal source and listing formats.

MPASM provides a rich directive language to support programming of the PIC16/17. Directives are helpful in making the development of your assemble source code shorter and more maintainable.

## **16.11 Software Simulator (MPLAB-SIM)**

The MPLAB-SIM Software Simulator allows code development in a PC host environment. It allows the user to simulate the PIC16/17 series microcontrollers on an instruction level. On any given instruction, the user may examine or modify any of the data areas or provide external stimulus to any of the pins. The input/output radix can be set by the user and the execution can be performed in; single step, execute until break, or in a trace mode.

MPLAB-SIM fully supports symbolic debugging using MPLAB-C and MPASM. The Software Simulator offers the low cost flexibility to develop and debug code outside of the laboratory environment making it an excellent multi-project software development tool.

## **16.12 C Compiler (MPLAB-C)**

The MPLAB-C Code Development System is a complete 'C' compiler and integrated development environment for Microchip's PIC16/17 family of microcontrollers. The compiler provides powerful integration capabilities and ease of use not found with other compilers.

For easier source level debugging, the compiler provides symbol information that is compatible with the MPLAB IDE memory display (PICMASTER emulator software versions 1.13 and later).

## **16.13 Fuzzy Logic Development System (fuzzyTECH-MP)**

*fuzzyTECH-MP* fuzzy logic development tool is available in two versions - a low cost introductory version, MP Explorer, for designers to gain a comprehensive working knowledge of fuzzy logic system design; and a full-featured version, *fuzzyTECH-MP*, edition for implementing more complex systems.

Both versions include Microchip's *fuzzyLAB™* demonstration board for hands-on experience with fuzzy logic systems implementation.

## **16.14 MP-DriveWay™ – Application Code Generator**

MP-DriveWay is an easy-to-use Windows-based Application Code Generator. With MP-DriveWay you can visually configure all the peripherals in a PIC16/17 device and, with a click of the mouse, generate all the initialization and many functional code modules in C language. The output is fully compatible with Microchip's MPLAB-C C compiler. The code produced is highly modular and allows easy integration of your own code. MP-DriveWay is intelligent enough to maintain your code through subsequent code generation.

## **16.15 SEEVAL® Evaluation and Programming System**

The SEEVAL SEEPROM Designer's Kit supports all Microchip 2-wire and 3-wire Serial EEPROMs. The kit includes everything necessary to read, write, erase or program special features of any Microchip SEEPROM product including Smart Serials™ and secure serials. The Total Endurance™ Disk is included to aid in trade-off analysis and reliability calculations. The total kit can significantly reduce time-to-market and result in an optimized system.

## **16.16 TrueGauge® Intelligent Battery Management**

The TrueGauge development tool supports system development with the MTA11200B TrueGauge Intelligent Battery Management IC. System design verification can be accomplished before hardware prototypes are built. User interface is graphically-oriented and measured data can be saved in a file for exporting to Microsoft Excel.

## **16.17 KEELOQ® Evaluation and Programming Tools**

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



# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

## 17.0 ELECTRICAL CHARACTERISTICS FOR PIC16C72

### 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 Vss (except VDD, $\overline{\text{MCLR}}$ , and RA4) .....	-0.3V to (VDD + 0.3V)
Voltage on VDD with respect to Vss .....	-0.3 to +7.5V
Voltage on $\overline{\text{MCLR}}$ with respect to Vss (Note 2) .....	0 to +14V
Voltage on RA4 with respect to Vss .....	0 to +14V
Total power dissipation (Note 1).....	1.0W
Maximum current out of Vss pin .....	300 mA
Maximum current into VDD pin .....	250 mA
Input clamp current, I <sub>IK</sub> (V <sub>I</sub> < 0 or V <sub>I</sub> > VDD) .....	±20 mA
Output clamp current, I <sub>OK</sub> (V <sub>O</sub> < 0 or V <sub>O</sub> > VDD).....	±20 mA
Maximum output current sunk by any I/O pin.....	25 mA
Maximum output current sourced by any I/O pin .....	25 mA
Maximum current sunk by PORTA and PORTB (combined).....	200 mA
Maximum current sourced by PORTA and PORTB (combined).....	200 mA
Maximum current sunk by PORTC .....	200 mA
Maximum current sourced by PORTC .....	200 mA

**Note 1:** Power dissipation is calculated as follows:  $P_{dis} = V_{DD} \times \{I_{DD} - \sum I_{OH}\} + \sum \{(V_{DD} - V_{OH}) \times I_{OH}\} + \sum (V_{OL} \times I_{OL})$

**Note 2:** Voltage spikes below Vss at the  $\overline{\text{MCLR}}$  pin, inducing currents greater than 80 mA, may cause latch-up. Thus, a series resistor of 50-100Ω should be used when applying a “low” level to the  $\overline{\text{MCLR}}$  pin rather than pulling this pin directly to Vss.

† NOTICE: Stresses above those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

**TABLE 17-1: CROSS REFERENCE OF DEVICE SPECS FOR OSCILLATOR CONFIGURATIONS AND FREQUENCIES OF OPERATION (COMMERCIAL DEVICES)**

OSC	PIC16C72-04	PIC16C72-10	PIC16C72-20	PIC16LC72-04	JW Devices
RC	VDD: 4.0V to 6.0V IDD: 5 mA max. at 5.5V IPD: 16 μA max. at 4V Freq: 4 MHz max.	VDD: 4.5V to 5.5V IDD: 2.7 mA typ. at 5.5V IPD: 1.5 μA typ. at 4V Freq: 4 MHz max.	VDD: 4.5V to 5.5V IDD: 2.7 mA typ. at 5.5V IPD: 1.5 μA typ. at 4V Freq: 4 MHz max.	VDD: 2.5V to 6.0V IDD: 3.8 mA max. at 3.0V IPD: 5.0 μA max. at 3V Freq: 4 MHz max.	VDD: 4.0V to 6.0V IDD: 5 mA max. at 5.5V IPD: 16 μA max. at 4V Freq: 4 MHz max.
XT	VDD: 4.0V to 6.0V IDD: 5 mA max. at 5.5V IPD: 16 μA max. at 4V Freq: 4 MHz max.	VDD: 4.5V to 5.5V IDD: 2.7 mA typ. at 5.5V IPD: 1.5 μA typ. at 4V Freq: 4 MHz max.	VDD: 4.5V to 5.5V IDD: 2.7 mA typ. at 5.5V IPD: 1.5 μA typ. at 4V Freq: 4 MHz max.	VDD: 2.5V to 6.0V IDD: 3.8 mA max. at 3.0V IPD: 5.0 μA max. at 3V Freq: 4 MHz max.	VDD: 4.0V to 6.0V IDD: 5 mA max. at 5.5V IPD: 16 μA max. at 4V Freq: 4 MHz max.
HS	VDD: 4.5V to 5.5V IDD: 13.5 mA typ. at 5.5V IPD: 1.5 μA typ. at 4.5V Freq: 4 MHz max.	VDD: 4.5V to 5.5V IDD: 10 mA max. at 5.5V IPD: 1.5 μA typ. at 4.5V Freq: 10 MHz max.	VDD: 4.5V to 5.5V IDD: 20 mA max. at 5.5V IPD: 1.5 μA typ. at 4.5V Freq: 20 MHz max.	Not recommended for use in HS mode	VDD: 4.5V to 5.5V IDD: 20 mA max. at 5.5V IPD: 1.5 μA typ. at 4.5V Freq: 20 MHz max.
LP	VDD: 4.0V to 6.0V IDD: 52.5 μA typ. at 32 kHz, 4.0V IPD: 0.9 μA typ. at 4.0V Freq: 200 kHz max.	Not recommended for use in LP mode	Not recommended for use in LP mode	VDD: 2.5V to 6.0V IDD: 48 μA max. at 32 kHz, 3.0V IPD: 5.0 μA max. at 3.0V Freq: 200 kHz max.	VDD: 2.5V to 6.0V IDD: 48 μA max. at 32 kHz, 3.0V IPD: 5.0 μA max. at 3.0V Freq: 200 kHz max.

The shaded sections indicate oscillator selections which are tested for functionality, but not for MIN/MAX specifications. It is recommended that the user select the device type that ensures the specifications required.

# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

**17.1 DC Characteristics: PIC16C72-04 (Commercial, Industrial, Extended)  
PIC16C72-10 (Commercial, Industrial, Extended)  
PIC16C72-20 (Commercial, Industrial, Extended)**

Standard Operating Conditions (unless otherwise stated)							
DC CHARACTERISTICS							
		Operating temperature -40°C ≤ TA ≤ +125°C for extended, -40°C ≤ TA ≤ +85°C for industrial and 0°C ≤ TA ≤ +70°C for commercial					
Param No.	Characteristic	Sym	Min	Typ†	Max	Units	Conditions
D001 D001A	Supply Voltage	VDD	4.0 4.5	- -	6.0 5.5	V V	XT, RC and LP osc configuration HS osc configuration
D002*	RAM Data Retention Voltage (Note 1)	VDR	-	1.5	-	V	
D003	VDD start voltage to ensure internal Power-on Reset Signal	VPOR	-	VSS	-	V	See section on Power-on Reset for details
D004*	VDD rise rate to ensure internal Power-on Reset Signal	SVDD	0.05	-	-	V/ms	See section on Power-on Reset for details
D005	Brown-out Reset Voltage	BVDD	3.7 3.7	4.0 4.0	4.3 4.4	V V	BODEN bit in configuration word enabled Extended Only
D010 D013 D015	Supply Current (Note 2,5)  Brown-out Reset Current (Note 6)	IDD  ΔIBOR	-  -	2.7 10 350	5.0 20 425	mA mA μA	XT, RC osc configuration FOSC = 4 MHz, VDD = 5.5V (Note 4)  HS osc configuration FOSC = 20 MHz, VDD = 5.5V  BOR enabled VDD = 5.0V
D020 D021 D021A D021B D023	Power-down Current (Note 3,5)   Brown-out Reset Current (Note 6)	IPD   ΔIBOR	- - - - -	10.5 1.5 1.5 2.5 350	42 16 19 19 425	μA μA μA μA μA	VDD = 4.0V, WDT enabled, -40°C to +85°C VDD = 4.0V, WDT disabled, -0°C to +70°C VDD = 4.0V, WDT disabled, -40°C to +85°C VDD = 4.0V, WDT disabled, -40°C to +125°C BOR enabled VDD = 5.0V

\* 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 tristated, 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.



# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

## 17.2 DC Characteristics: PIC16LC72-04 (Commercial, 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 and $0^{\circ}\text{C} \leq T_A \leq +70^{\circ}\text{C}$ for commercial					
Param No.	Characteristic	Sym	Min	Typ†	Max	Units	Conditions
D001	Supply Voltage	VDD	2.5	-	6.0	V	LP, XT, RC osc configuration (DC - 4 MHz)
D002*	RAM Data Retention Voltage (Note 1)	VDR	-	1.5	-	V	
D003	VDD start voltage to ensure internal Power-on Reset signal	VPOR	-	VSS	-	V	See section on Power-on Reset for details
D004*	VDD rise rate to ensure internal Power-on Reset signal	SVDD	0.05	-	-	V/ms	See section on Power-on Reset for details
D005	Brown-out Reset Voltage	BVDD	3.7	4.0	4.3	V	BODEN bit in configuration word enabled
D010	Supply Current (Note 2,5)	IDD	-	2.0	3.8	mA	XT, RC osc configuration FOSC = 4 MHz, VDD = 3.0V (Note 4)
D010A			-	22.5	48	$\mu\text{A}$	LP osc configuration FOSC = 32 kHz, VDD = 3.0V, WDT disabled
D015*	Brown-out Reset Current (Note 6)	$\Delta\text{IBOR}$	-	350	425	$\mu\text{A}$	BOR enabled VDD = 5.0V
D020	Power-down Current (Note 3,5)	IPD	-	7.5	30	$\mu\text{A}$	VDD = 3.0V, WDT enabled, $-40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$
D021			-	0.9	5	$\mu\text{A}$	VDD = 3.0V, WDT disabled, $0^{\circ}\text{C}$ to $+70^{\circ}\text{C}$
D021A			-	0.9	5	$\mu\text{A}$	VDD = 3.0V, WDT disabled, $-40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$
D023*	Brown-out Reset Current (Note 6)	$\Delta\text{IBOR}$	-	350	425	$\mu\text{A}$	BOR enabled VDD = 5.0V

\* 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 tristated, 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  $\mu\text{A}$  to the specification. This value is from characterization and is for design guidance only. This is not tested.

6: The  $\Delta$  current is the additional current consumed when this peripheral is enabled. This current should be added to the base IDD or IPD measurement.

# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

**17.3 DC Characteristics:** **PIC16C72-04 (Commercial, Industrial, Extended)**  
**PIC16C72-10 (Commercial, Industrial, Extended)**  
**PIC16C72-20 (Commercial, Industrial, Extended)**  
**PIC16LC72-04 (Commercial, Industrial)**

Standard Operating Conditions (unless otherwise stated)							
DC CHARACTERISTICS							
Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ for extended, $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial and $0^{\circ}\text{C} \leq T_A \leq +70^{\circ}\text{C}$ for commercial							
Operating voltage $V_{DD}$ range as described in DC spec Section 17.1 and Section 17.2.							
Param No.	Characteristic	Sym	Min	Typ †	Max	Units	Conditions
D030	<b>Input Low Voltage</b> I/O ports with TTL buffer with Schmitt Trigger buffer MCLR, OSC1 (in RC mode) OSC1 (in XT, HS and LP)	$V_{IL}$	$V_{SS}$	-	$0.15V_{DD}$	V	For entire $V_{DD}$ range $4.5 \leq V_{DD} \leq 5.5\text{V}$ Note1
D030A			$V_{SS}$	-	0.8V	V	
D031			$V_{SS}$	-	$0.2V_{DD}$	V	
D032			$V_{SS}$	-	$0.2V_{DD}$	V	
D033			$V_{SS}$	-	$0.3V_{DD}$	V	
D040	<b>Input High Voltage</b> I/O ports with TTL buffer with Schmitt Trigger buffer MCLR OSC1 (XT, HS and LP) OSC1 (in RC mode)	$V_{IH}$		-			For entire $V_{DD}$ range Note1
D040A			2.0	-	$V_{DD}$	V	
			$0.25V_{DD} + 0.8\text{V}$	-	$V_{DD}$	V	
D041			$0.8V_{DD}$	-	$V_{DD}$	V	
D042			$0.8V_{DD}$	-	$V_{DD}$	V	
D042A			$0.7V_{DD}$	-	$V_{DD}$	V	
D043			$0.9V_{DD}$	-	$V_{DD}$	V	
D070	PORTB weak pull-up current	$I_{PURB}$	50	250	†400	$\mu\text{A}$	$V_{DD} = 5\text{V}, V_{PIN} = V_{SS}$
D060	<b>Input Leakage Current</b> (Notes 2, 3) I/O ports MCLR, RA4/T0CKI OSC1	$I_{IL}$	-	-	$\pm 1$	$\mu\text{A}$	$V_{SS} \leq V_{PIN} \leq V_{DD}$ , Pin at hi-impedance
D061			-	-	$\pm 5$	$\mu\text{A}$	$V_{SS} \leq V_{PIN} \leq V_{DD}$
D063			-	-	$\pm 5$	$\mu\text{A}$	$V_{SS} \leq V_{PIN} \leq V_{DD}$ , XT, HS and LP osc configuration
D080	<b>Output Low Voltage</b> I/O ports OSC2/CLKOUT (RC osc config)	$V_{OL}$	-	-	0.6	V	$I_{OL} = 8.5\text{ mA}, V_{DD} = 4.5\text{V}, -40^{\circ}\text{C to } +85^{\circ}\text{C}$
D080A			-	-	0.6	V	$I_{OL} = 7.0\text{ mA}, V_{DD} = 4.5\text{V}, -40^{\circ}\text{C to } +125^{\circ}\text{C}$
D083			-	-	0.6	V	$I_{OL} = 1.6\text{ mA}, V_{DD} = 4.5\text{V}, -40^{\circ}\text{C to } +85^{\circ}\text{C}$
D083A			-	-	0.6	V	$I_{OL} = 1.2\text{ mA}, V_{DD} = 4.5\text{V}, -40^{\circ}\text{C to } +125^{\circ}\text{C}$

\* These parameters are characterized but not tested.

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

- Note 1: In RC oscillator configuration, the OSC1/CLKIN pin is a Schmitt trigger input. It is not recommended that the PIC16C7X be driven with external clock in RC mode.
- 2: The leakage current on the MCLR/VPP pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.
- 3: Negative current is defined as current sourced by the pin.

# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

DC CHARACTERISTICS							
Standard Operating Conditions (unless otherwise stated)							
Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ for extended, $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial and $0^{\circ}\text{C} \leq T_A \leq +70^{\circ}\text{C}$ for commercial							
Operating voltage $V_{DD}$ range as described in DC spec Section 17.1 and Section 17.2.							
Param No.	Characteristic	Sym	Min	Typ †	Max	Units	Conditions
D090	<b>Output High Voltage</b> I/O ports (Note 3)	VOH	VDD - 0.7	-	-	V	IOH = -3.0 mA, VDD = 4.5V, -40°C to +85°C
D090A			VDD - 0.7	-	-	V	
D092	OSC2/CLKOUT (RC osc config)		VDD - 0.7	-	-	V	IOH = -1.3 mA, VDD = 4.5V, -40°C to +85°C
D092A			VDD - 0.7	-	-	V	
D150*	<b>Open-Drain High Voltage</b>	VOD	-	-	14	V	RA4 pin
<b>Capacitive Loading Specs on Output Pins</b>							
D100	OSC2 pin	COSC2	-	-	15	pF	In XT, HS and LP modes when external clock is used to drive OSC1.
D101	All I/O pins and OSC2 (in RC mode)	CIO	-	-	50	pF	
D102	SCL, SDA in I <sup>2</sup> C mode	CB	-	-	400	pF	

\* 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 PIC16C7X be driven with external clock in RC mode.
- 2: The leakage current on the MCLR/VPP pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.
- 3: Negative current is defined as current sourced by the pin.

# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

## 17.4 Timing Parameter Symbology

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

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

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

Lowercase letters (pp) and their meanings:

<b>pp</b>	
cc      CCP1	osc      OSC1
ck      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 $\overline{WR}$

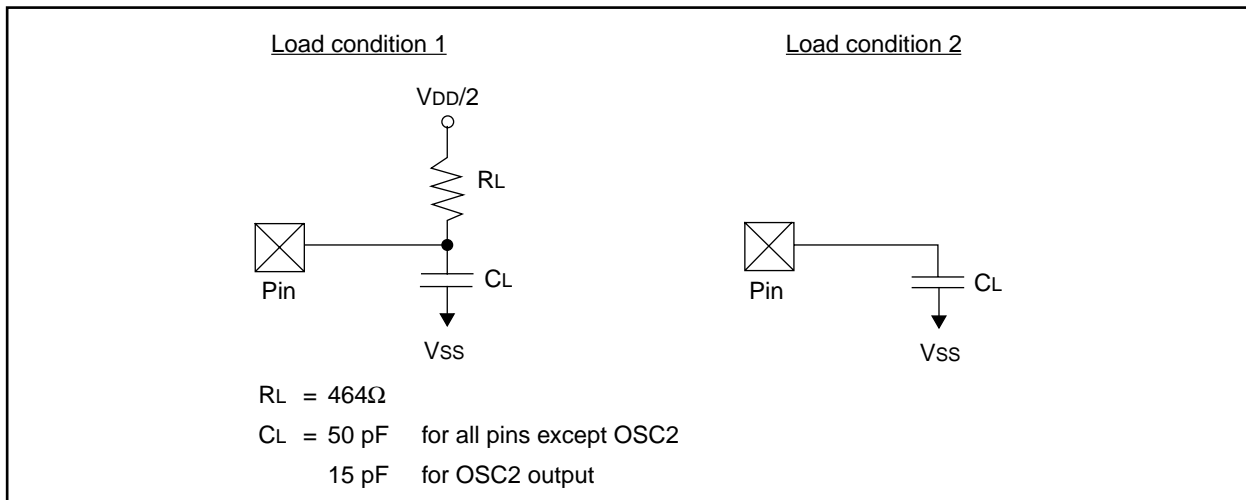
Uppercase letters and their meanings:

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

Tcc:st (I<sup>2</sup>C specifications only)

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

**FIGURE 17-1: LOAD CONDITIONS**



## 17.5 Timing Diagrams and Specifications

FIGURE 17-2: EXTERNAL CLOCK TIMING

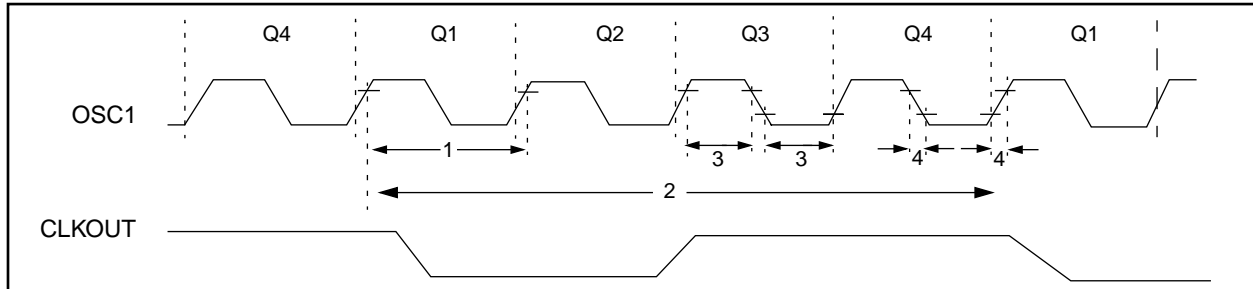


TABLE 17-2: EXTERNAL CLOCK TIMING REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
	Fosc	<b>External CLKIN Frequency (Note 1)</b>	DC	—	4	MHz	XT and RC osc mode
			DC	—	4	MHz	HS osc mode (-04)
			DC	—	10	MHz	HS osc mode (-10)
			DC	—	20	MHz	HS osc mode (-20)
			DC	—	200	kHz	LP osc mode
		<b>Oscillator Frequency (Note 1)</b>	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	<b>External CLKIN Period (Note 1)</b>	250	—	—	ns	XT and RC osc mode
			250	—	—	ns	HS osc mode (-04)
			100	—	—	ns	HS osc mode (-10)
			50	—	—	ns	HS osc mode (-20)
			5	—	—	µs	LP osc mode
		<b>Oscillator Period (Note 1)</b>	250	—	—	ns	RC osc mode
			250	—	10,000	ns	XT osc mode
			250	—	250	ns	HS osc mode (-04)
			100	—	250	ns	HS osc mode (-10)
			50	—	250	ns	HS osc mode (-20)
			5	—	—	µs	LP osc mode
2	Tcy	<b>Instruction Cycle Time (Note 1)</b>	200	—	DC	ns	Tcy = 4/Fosc
3	TosL, TosH	<b>External Clock in (OSC1) High or Low Time</b>	100	—	—	ns	XT oscillator
			2.5	—	—	µs	LP oscillator
			15	—	—	ns	HS oscillator
4	TosR, TosF	<b>External Clock in (OSC1) Rise or Fall Time</b>	—	—	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.

# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

FIGURE 17-3: CLKOUT AND I/O TIMING

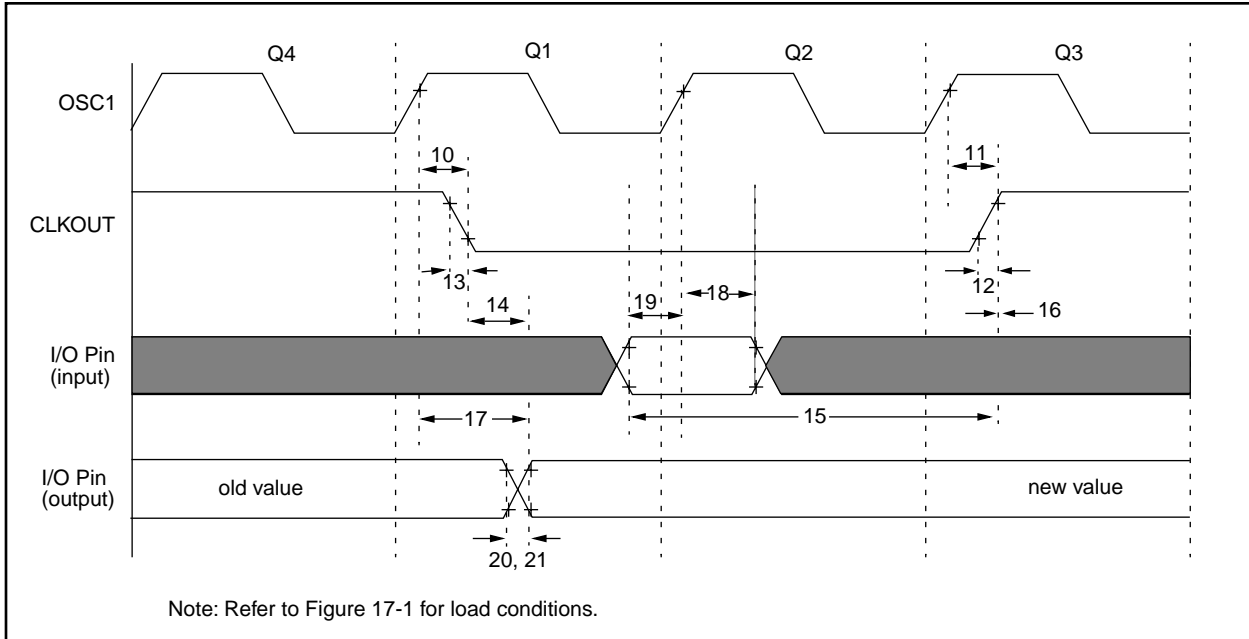


TABLE 17-3: CLKOUT AND I/O TIMING REQUIREMENTS

Parameter No.	Sym	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	—	—	0.5TCY + 20	ns	Note 1
15*	TioV2ckH	Port in valid before CLKOUT ↑	Tosc + 200	—	—	ns	Note 1
16*	TckH2iol	Port in hold after CLKOUT ↑	0	—	—	ns	Note 1
17*	TosH2ioV	OSC1↑ (Q1 cycle) to Port out valid	—	50	150	ns	
18*	TosH2iol	OSC1↑ (Q2 cycle) to Port input invalid (I/O in hold time)	PIC16C72	100	—	—	ns
			PIC16LC72	200	—	—	ns
19*	TioV2osH	Port input valid to OSC1↑ (I/O in setup time)	0	—	—	ns	
20*	TioR	Port output rise time	PIC16C72	—	10	40	ns
			PIC16LC72	—	—	80	ns
21*	TioF	Port output fall time	PIC16C72	—	10	40	ns
			PIC16LC72	—	—	80	ns
22††*	Tinp	INT pin high or low time	TCY	—	—	ns	
23††*	Trbp	RB7:RB4 change INT high or low time	TCY	—	—	ns	

\* These parameters are characterized but not tested.

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

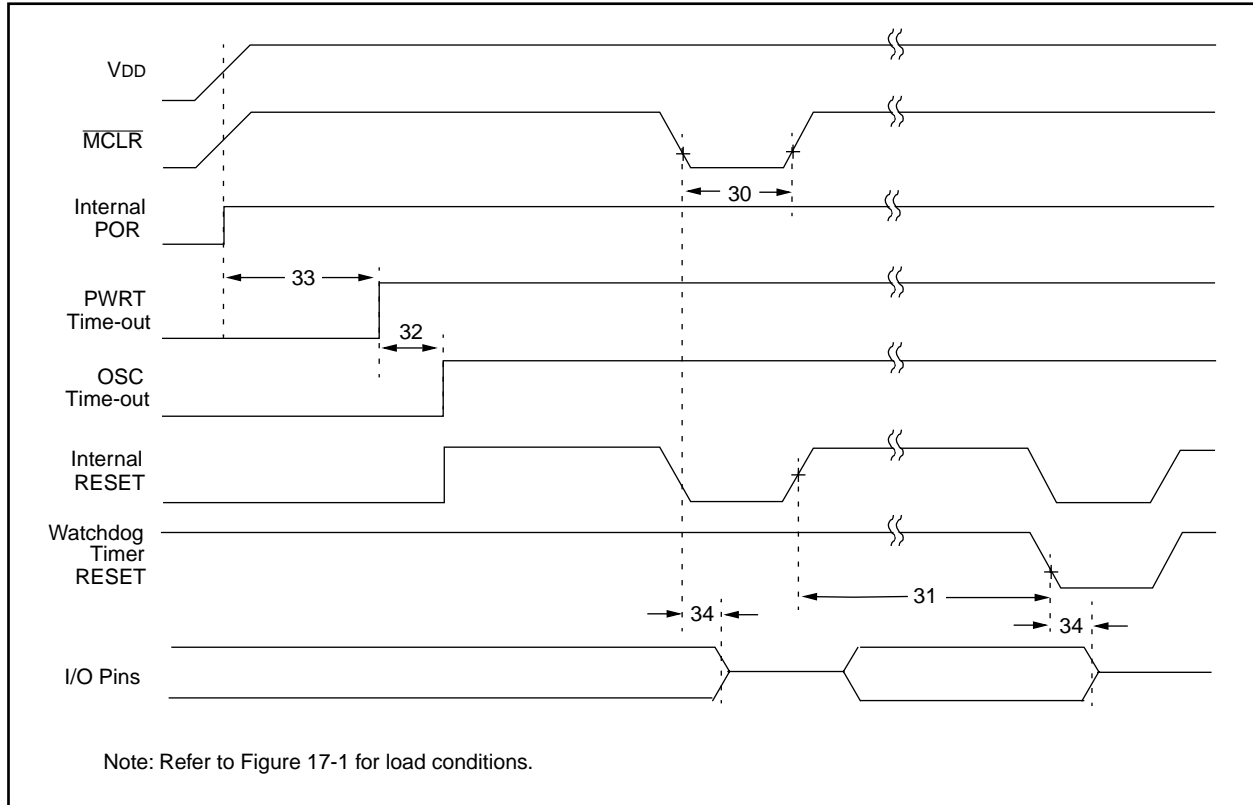
†† These parameters are asynchronous events not related to any internal clock edges.

Note 1: Measurements are taken in RC Mode where CLKOUT output is 4 x TOSC.

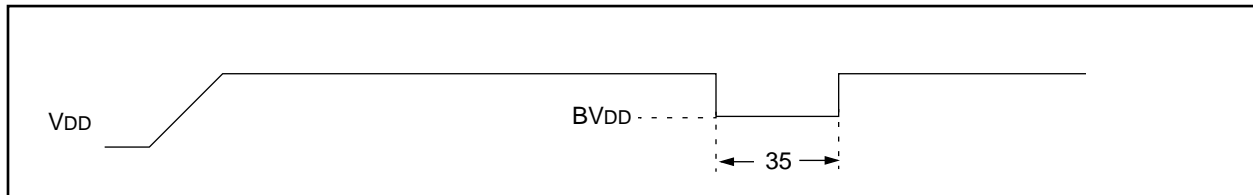
# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

**FIGURE 17-4: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING**



**FIGURE 17-5: BROWN-OUT RESET TIMING**



**TABLE 17-4: 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 +125°C
31*	Twdt	Watchdog Timer Time-out Period (No Prescaler)	7	18	33	ms	VDD = 5V, -40°C to +125°C
32	Tost	Oscillation Start-up Timer Period	—	1024Tosc	—	—	Tosc = OSC1 period
33*	Tpwrt	Power-up Timer Period	28	72	132	ms	VDD = 5V, -40°C to +125°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 ≤ BVDD (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.

# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

FIGURE 17-6: TIMER0 AND TIMER1 EXTERNAL CLOCK TIMINGS

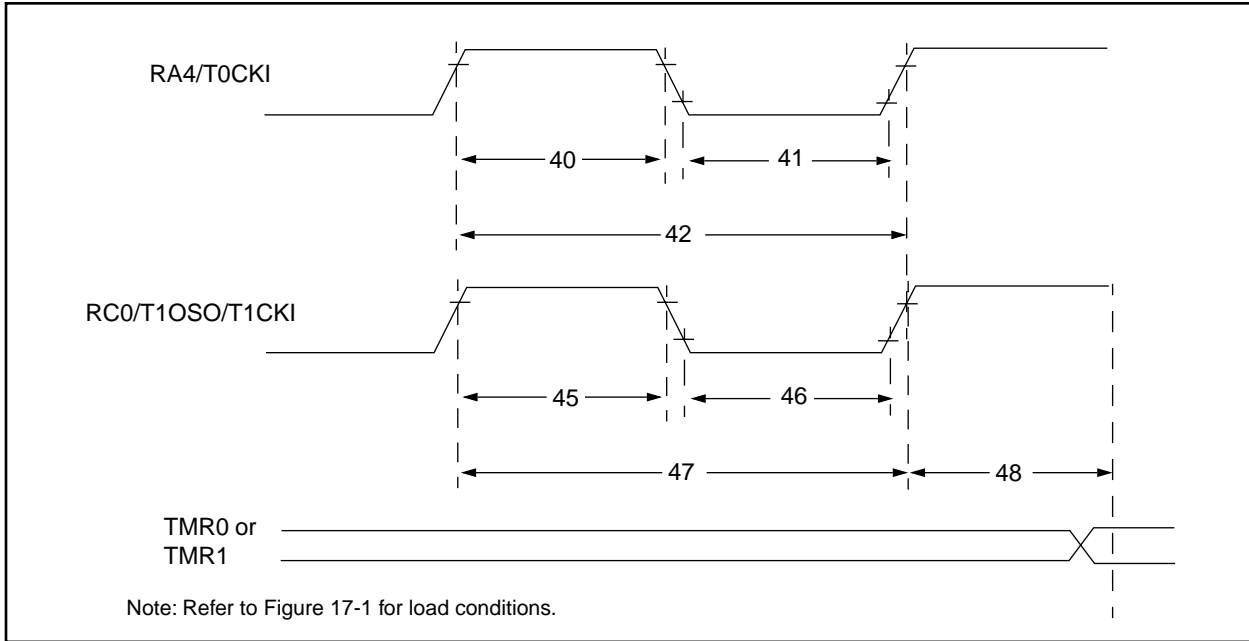


TABLE 17-5: TIMER0 AND TIMER1 EXTERNAL CLOCK REQUIREMENTS

Param No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions	
40*	Tt0H	T0CKI High Pulse Width	No Prescaler	$0.5T_{CY} + 20$	—	—	ns	Must also meet parameter 42	
			With Prescaler	10	—	—	ns		
41*	Tt0L	T0CKI Low Pulse Width	No Prescaler	$0.5T_{CY} + 20$	—	—	ns	Must also meet parameter 42	
			With Prescaler	10	—	—	ns		
42*	Tt0P	T0CKI Period	No Prescaler	$T_{CY} + 40$	—	—	ns	N = prescale value (2, 4, ..., 256)	
			With Prescaler	Greater of: $20$ or $\frac{T_{CY} + 40}{N}$	—	—	ns		
45*	Tt1H	T1CKI High Time	Synchronous, Prescaler = 1	$0.5T_{CY} + 20$	—	—	ns	Must also meet parameter 47	
			Synchronous, Prescaler = 2,4,8	PIC16C7X	15	—	—		ns
				PIC16LC7X	25	—	—		ns
			Asynchronous	PIC16C7X	30	—	—		ns
			PIC16LC7X	50	—	—	ns		
46*	Tt1L	T1CKI Low Time	Synchronous, Prescaler = 1	$0.5T_{CY} + 20$	—	—	ns	Must also meet parameter 47	
			Synchronous, Prescaler = 2,4,8	PIC16C7X	15	—	—		ns
				PIC16LC7X	25	—	—		ns
			Asynchronous	PIC16C7X	30	—	—		ns
			PIC16LC7X	50	—	—	ns		
47*	Tt1P	T1CKI input period	Synchronous	PIC16C7X	Greater of: $30$ or $\frac{T_{CY} + 40}{N}$	—	—	ns	N = prescale value (1, 2, 4, 8)
				PIC16LC7X	Greater of: $50$ or $\frac{T_{CY} + 40}{N}$	—	—	ns	N = prescale value (1, 2, 4, 8)
			Asynchronous	PIC16C7X	60	—	—	ns	
				PIC16LC7X	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		$2T_{osc}$	—	$7T_{osc}$	—		

\* These parameters are characterized but not tested.

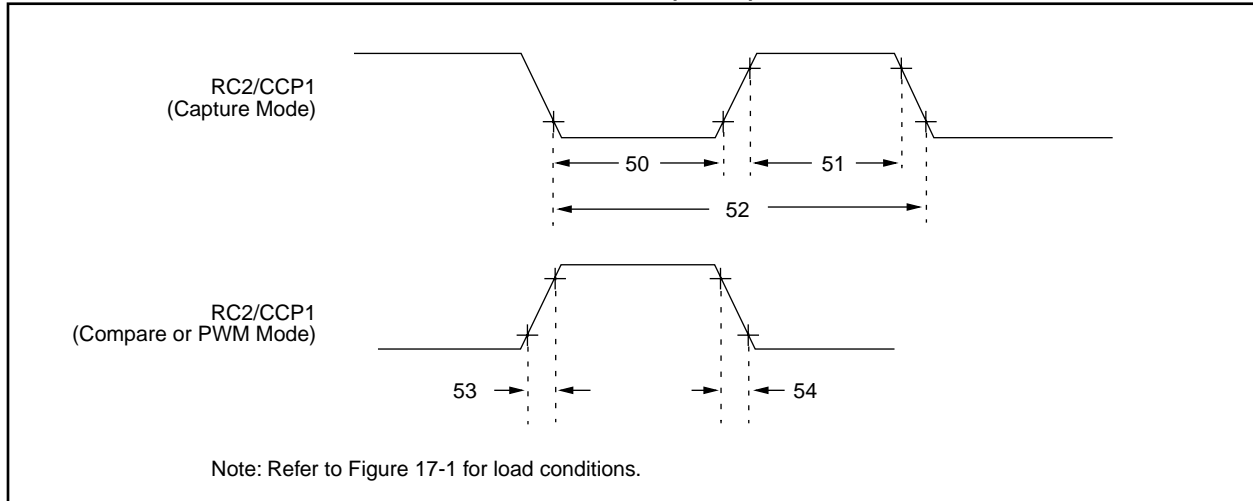
† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.



# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

**FIGURE 17-7: CAPTURE/COMPARE/PWM TIMINGS (CCP1)**



**TABLE 17-6: CAPTURE/COMPARE/PWM REQUIREMENTS (CCP1)**

Param No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions	
50*	TccL	CCP1 input low time	No Prescaler	$0.5T_{CY} + 20$	—	—	ns		
			With Prescaler	PIC16C72	10	—	—		ns
				PIC16LC72	20	—	—		ns
51*	TccH	CCP1 input high time	No Prescaler	$0.5T_{CY} + 20$	—	—	ns		
			With Prescaler	PIC16C72	10	—	—		ns
				PIC16LC72	20	—	—		ns
52*	TccP	CCP1 input period		$\frac{3T_{CY} + 40}{N}$	—	—	ns	N = prescale value (1,4 or 16)	
53*	TccR	CCP1 output rise time	PIC16C72	—	10	25	ns		
			PIC16LC72	—	25	45	ns		
54*	TccF	CCP1 output fall time	PIC16C72	—	10	25	ns		
			PIC16LC72	—	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.

# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

FIGURE 17-8: SPI MODE TIMING

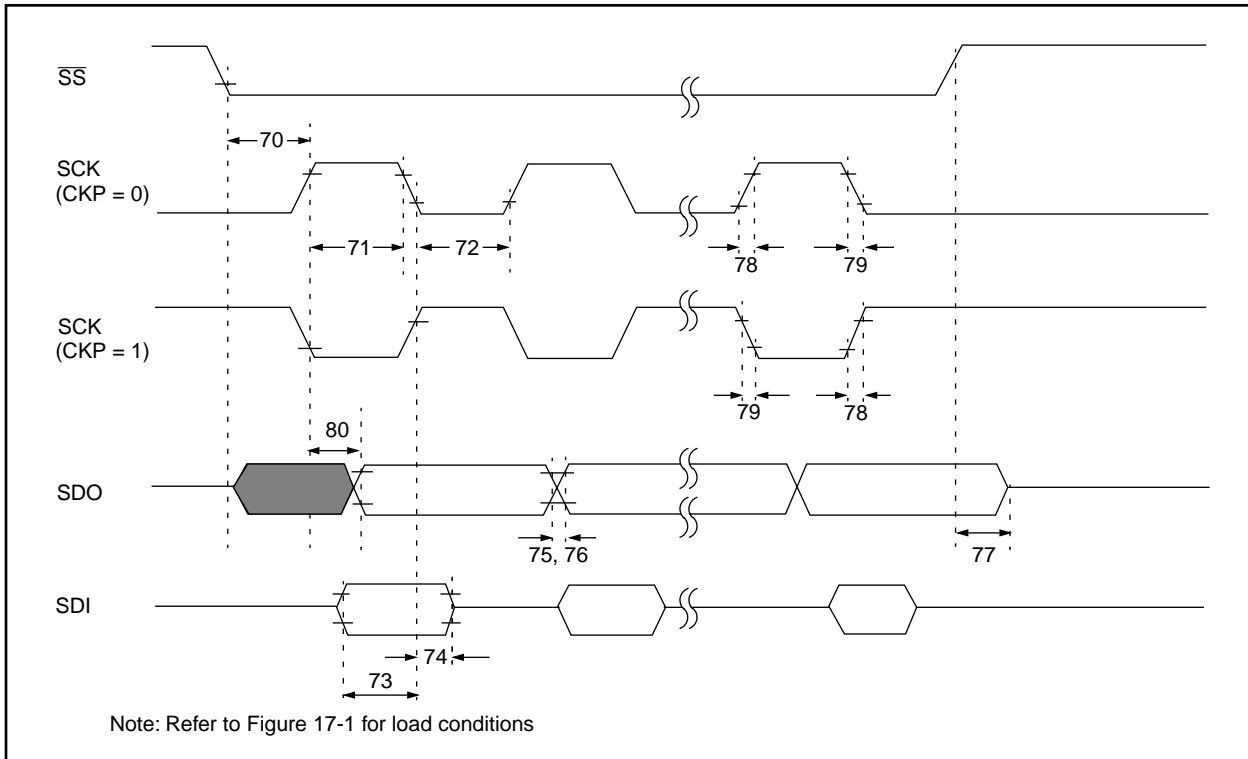


TABLE 17-7: SPI MODE REQUIREMENTS

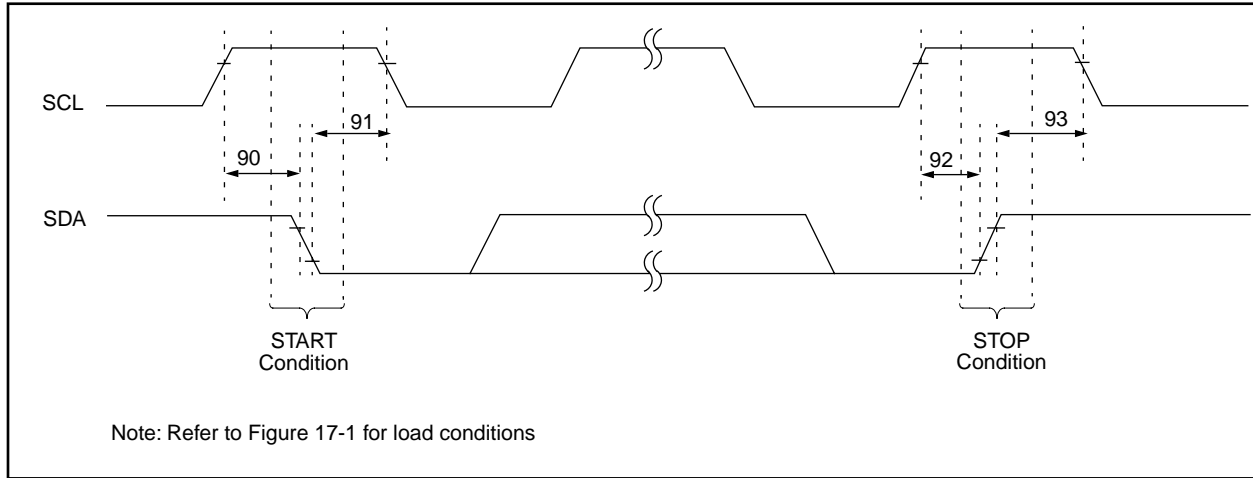
Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
70	TssL2scH, TssL2scL	SS↓ to SCK↓ or SCK↑ input	T <sub>cy</sub>	—	—	ns	
71	TscH	SCK input high time (slave mode)	T <sub>cy</sub> + 20	—	—	ns	
72	TscL	SCK input low time (slave mode)	T <sub>cy</sub> + 20	—	—	ns	
73	TdiV2scH, TdiV2scL	Setup time of SDI data input to SCK edge	50	—	—	ns	
74	Tsch2diL, TscL2diL	Hold time of SDI data input to SCK edge	50	—	—	ns	
75	TdoR	SDO data output rise time	—	10	25	ns	
76	TdoF	SDO data output fall time	—	10	25	ns	
77	TssH2doZ	SS↑ to SDO output hi-impedance	10	—	50	ns	
78	TscR	SCK output rise time (master mode)	—	10	25	ns	
79	TscF	SCK output fall time (master mode)	—	10	25	ns	
80	Tsch2doV, TscL2doV	SDO data output valid after SCK edge	—	—	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.

# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

**FIGURE 17-9: I<sup>2</sup>C BUS START/STOP BITS TIMING**



**TABLE 17-8: I<sup>2</sup>C BUS START/STOP BITS REQUIREMENTS**

Parameter No.	Sym	Characteristic	Min	Typ	Max	Units	Conditions	
90	T <sub>SU:STA</sub>	START condition Setup time	100 kHz mode	4700	—	—	ns	Only relevant for repeated START condition
			400 kHz mode	600	—	—		
91	T <sub>HD:STA</sub>	START condition Hold time	100 kHz mode	4000	—	—	ns	After this period the first clock pulse is generated
			400 kHz mode	600	—	—		
92	T <sub>SU:STO</sub>	STOP condition Setup time	100 kHz mode	4700	—	—	ns	
			400 kHz mode	600	—	—		
93	T <sub>HD:STO</sub>	STOP condition Hold time	100 kHz mode	4000	—	—	ns	
			400 kHz mode	600	—	—		

# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

FIGURE 17-10: I<sup>2</sup>C BUS DATA TIMING

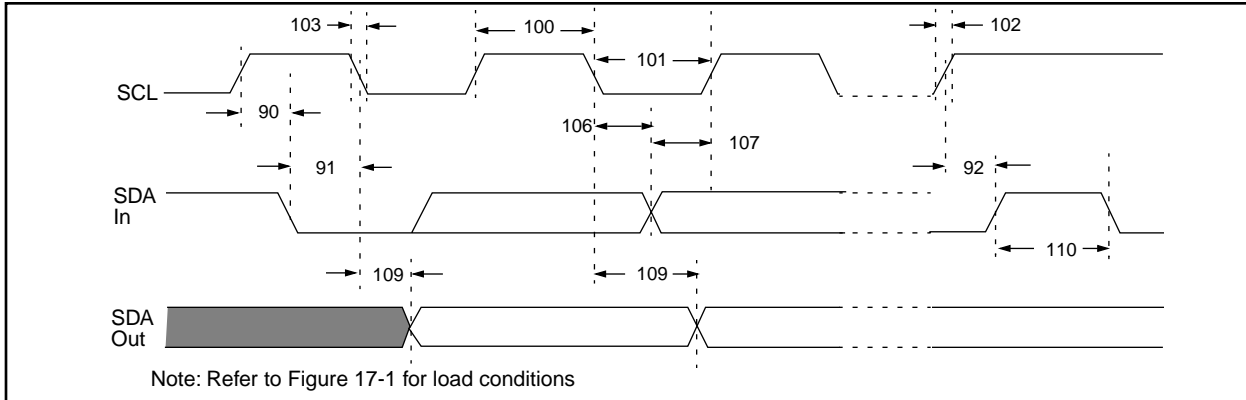


TABLE 17-9: I<sup>2</sup>C BUS DATA REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Max	Units	Conditions	
100	T <sub>HIGH</sub>	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	1.5T <sub>CY</sub>	—		
101	T <sub>LOW</sub>	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	1.5T <sub>CY</sub>	—		
102	T <sub>R</sub>	SDA and SCL rise time	100 kHz mode	—	1000	ns	
			400 kHz mode	20 + 0.1C <sub>b</sub>	300	ns	C <sub>b</sub> is specified to be from 10 to 400 pF
103	T <sub>F</sub>	SDA and SCL fall time	100 kHz mode	—	300	ns	
			400 kHz mode	20 + 0.1C <sub>b</sub>	300	ns	C <sub>b</sub> is specified to be from 10 to 400 pF
90	T <sub>SU:STA</sub>	START condition setup time	100 kHz mode	4.7	—	μs	Only relevant for repeated START condition
			400 kHz mode	0.6	—	μs	
91	T <sub>HD:STA</sub>	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	T <sub>HD:DAT</sub>	Data input hold time	100 kHz mode	0	—	ns	
			400 kHz mode	0	0.9	μs	
107	T <sub>SU:DAT</sub>	Data input setup time	100 kHz mode	250	—	ns	Note 2
			400 kHz mode	100	—	ns	
92	T <sub>SU:STO</sub>	STOP condition setup time	100 kHz mode	4.7	—	μs	
			400 kHz mode	0.6	—	μs	
109	T <sub>AA</sub>	Output valid from clock	100 kHz mode	—	3500	ns	Note 1
			400 kHz mode	—	—	ns	
110	T <sub>BUF</sub>	Bus free time	100 kHz mode	4.7	—	μs	Time the bus must be free before a new transmission can start
			400 kHz mode	1.3	—	μs	
	C <sub>b</sub>	Bus capacitive loading	—	400	pF		

Note 1: As a transmitter, the device must provide this internal minimum delay time to bridge the undefined region (min. 300 ns) of the falling edge of SCL to avoid unintended generation of START or STOP conditions.

2: A fast-mode (400 kHz) I<sup>2</sup>C-bus device can be used in a standard-mode (100 kHz) I<sup>2</sup>C-bus system, but the requirement t<sub>SU:DAT</sub> ≥ 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<sub>R</sub> max. + t<sub>SU:DAT</sub> = 1000 + 250 = 1250 ns (according to the standard-mode I<sup>2</sup>C bus specification) before the SCL line is released.

# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

**TABLE 17-10: A/D CONVERTER CHARACTERISTICS:**  
**PIC16C72-04 (Commercial, Industrial, Extended)**  
**PIC16C72-10 (Commercial, Industrial, Extended)**  
**PIC16C72-20 (Commercial, Industrial, Extended)**  
**PIC16LC72-04 (Commercial, Industrial)**

Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions	
A01	NR	Resolution	—	—	8-bits	bit	VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF	
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	E0FF	Offset error	—	—	< ± 1	LSb	VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF	
A10	—	Monotonicity	—	guaranteed	—	—	VSS ≤ VAIN ≤ VREF	
A20	VREF	Reference voltage	3.0V	—	VDD + 0.3	V		
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)	PIC16C72	—	180	—	μA	Average current consumption when A/D is on. (Note 1)
			PIC16LC72	—	90	—	μA	
A50	IREF	VREF input current (Note 2)	—	—	1000	μA	During VAIN acquisition. Based on differential of VHOLD to VAIN to charge CHOLD, see Section 13.1.	
			10	—	10	μA	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 RA3 pin or VDD pin, whichever is selected as reference input.

# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

FIGURE 17-11: A/D CONVERSION TIMING

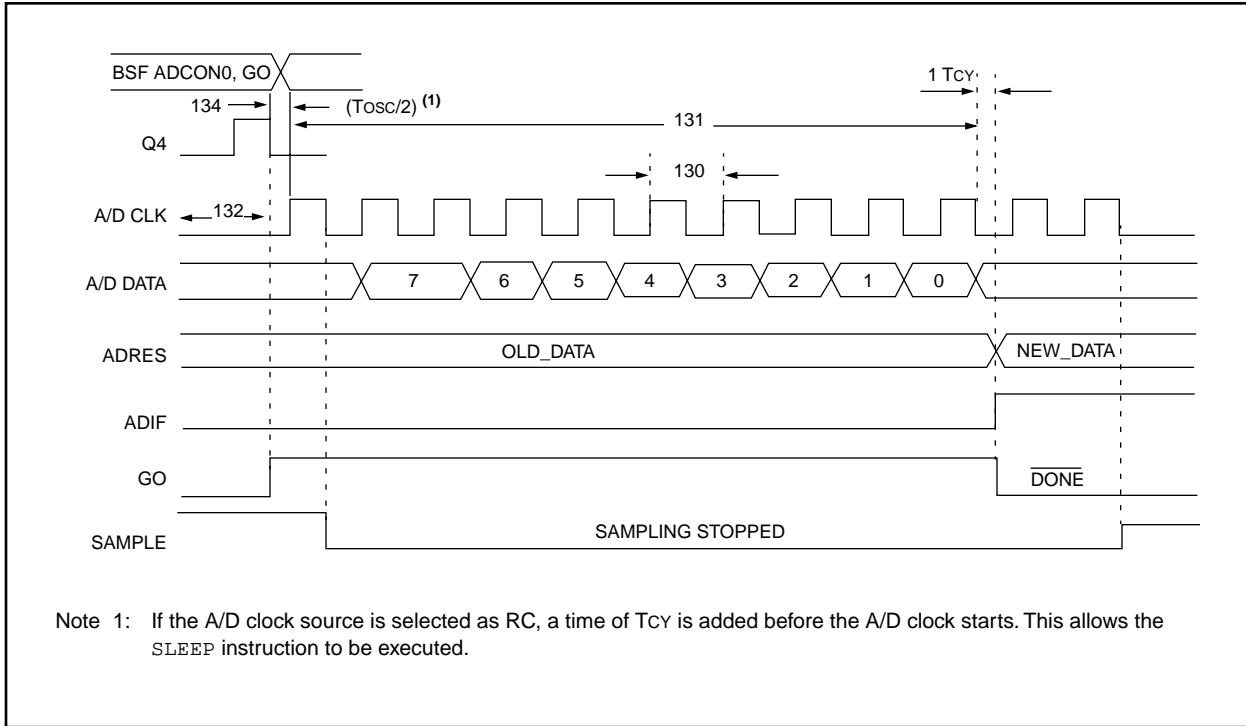


TABLE 17-11: A/D CONVERSION REQUIREMENTS

Param No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions
130	TAD	A/D clock period	PIC16C72	1.6	—	—	$\mu\text{s}$	TOSC based, $V_{REF} \geq 3.0V$
			PIC16LC72	2.0	—	—	$\mu\text{s}$	TOSC based, $V_{REF}$ full range
			PIC16C72	2.0	4.0	6.0	$\mu\text{s}$	A/D RC Mode
			PIC16LC72	3.0	6.0	9.0	$\mu\text{s}$	A/D RC Mode
131	T <sub>CNV</sub>	Conversion time (not including S/H time) (Note 1)	—	9.5	—	TAD		
132	TACQ	Acquisition time	Note 2	20	—	$\mu\text{s}$	The minimum time is the amplifier settling time. This may be used if the "new" input voltage has not changed by more than 1 LSb (i.e., 20.0 mV @ 5.12V) from the last sampled voltage (as stated on CHOLD).	
			5*	—	—	$\mu\text{s}$		
134	T <sub>GO</sub>	Q4 to A/D clock start	—	$T_{osc}/2$ §	—	—	If the A/D clock source is selected as RC, a time of $T_{CY}$ is added before the A/D clock starts. This allows the SLEEP instruction to be executed.	
135	T <sub>SWC</sub>	Switching from convert → sample time	1.5 §	—	—	TAD		

\* These parameters are characterized but not tested.

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

§ This specification ensured by design.

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

Note 2: See Section 13.1 for min conditions.

# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

## 18.0 ELECTRICAL CHARACTERISTICS FOR PIC16C73/74

### 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 Vss (except VDD, $\overline{\text{MCLR}}$ , and RA4).....	-0.3V to (VDD + 0.3V)
Voltage on VDD with respect to Vss .....	-0.3 to +7.5V
Voltage on $\overline{\text{MCLR}}$ with respect to Vss (Note 2) .....	0 to +14V
Voltage on RA4 with respect to Vss .....	0 to +14V
Total power dissipation (Note 1).....	1.0W
Maximum current out of Vss pin .....	300 mA
Maximum current into VDD pin .....	250 mA
Input clamp current, I <sub>IK</sub> (V <sub>I</sub> < 0 or V <sub>I</sub> > VDD) .....	±20 mA
Output clamp current, I <sub>OK</sub> (V <sub>O</sub> < 0 or V <sub>O</sub> > VDD).....	±20 mA
Maximum output current sunk by any I/O pin.....	25 mA
Maximum output current sourced by any I/O pin .....	25 mA
Maximum current sunk by PORTA, 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:  $P_{dis} = V_{DD} \times \{I_{DD} - \sum IOH\} + \sum \{(V_{DD} - V_{OH}) \times IOH\} + \sum (V_{OL} \times I_{OL})$

**Note 2:** Voltage spikes below Vss at the  $\overline{\text{MCLR}}$  pin, inducing currents greater than 80 mA, may cause latch-up. Thus, a series resistor of 50-100Ω should be used when applying a “low” level to the  $\overline{\text{MCLR}}$  pin rather than pulling this pin directly to Vss.

**Note 3:** PORTD and PORTE are not implemented on the PIC16C73.

† 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.

**TABLE 18-1: CROSS REFERENCE OF DEVICE SPECS FOR OSCILLATOR CONFIGURATIONS AND FREQUENCIES OF OPERATION (COMMERCIAL DEVICES)**

OSC	PIC16C73-04 PIC16C74-04	PIC16C73-10 PIC16C74-10	PIC16C73-20 PIC16C74-20	PIC16LC73-04 PIC16LC74-04	JW Devices
RC	VDD: 4.0V to 6.0V IDD: 5 mA max. at 5.5V IPD: 21 μA max. at 4V Freq: 4 MHz max.	VDD: 4.5V to 5.5V IDD: 2.7 mA typ. at 5.5V IPD: 1.5 μA typ. at 4V Freq: 4 MHz max.	VDD: 4.5V to 5.5V IDD: 2.7 mA typ. at 5.5V IPD: 1.5 μA typ. at 4V Freq: 4 MHz max.	VDD: 3.0V to 6.0V IDD: 3.8 mA max. at 3.0V IPD: 13.5 μA max. at 3V Freq: 4 MHz max.	VDD: 4.0V to 6.0V IDD: 5 mA max. at 5.5V IPD: 21 μA max. at 4V Freq: 4 MHz max.
XT	VDD: 4.0V to 6.0V IDD: 5 mA max. at 5.5V IPD: 21 μA max. at 4V Freq: 4 MHz max.	VDD: 4.5V to 5.5V IDD: 2.7 mA typ. at 5.5V IPD: 1.5 μA typ. at 4V Freq: 4 MHz max.	VDD: 4.5V to 5.5V IDD: 2.7 mA typ. at 5.5V IPD: 1.5 μA typ. at 4V Freq: 4 MHz max.	VDD: 3.0V to 6.0V IDD: 3.8 mA max. at 3.0V IPD: 13.5 μA max. at 3V Freq: 4 MHz max.	VDD: 4.0V to 6.0V IDD: 5 mA max. at 5.5V IPD: 21 μA max. at 4V Freq: 4 MHz max.
HS	VDD: 4.5V to 5.5V IDD: 13.5 mA typ. at 5.5V IPD: 1.5 μA typ. at 4.5V Freq: 4 MHz max.	VDD: 4.5V to 5.5V IDD: 15 mA max. at 5.5V IPD: 1.5 μA typ. at 4.5V Freq: 10 MHz max.	VDD: 4.5V to 5.5V IDD: 30 mA max. at 5.5V IPD: 1.5 μA typ. at 4.5V Freq: 20 MHz max.	Not recommended for use in HS mode	VDD: 4.5V to 5.5V IDD: 30 mA max. at 5.5V IPD: 1.5 μA typ. at 4.5V Freq: 20 MHz max.
LP	VDD: 4.0V to 6.0V IDD: 52.5 μA typ. at 32 kHz, 4.0V IPD: 0.9 μA typ. at 4.0V Freq: 200 kHz max.	Not recommended for use in LP mode	Not recommended for use in LP mode	VDD: 3.0V to 6.0V IDD: 48 μA max. at 32 kHz, 3.0V IPD: 13.5 μA max. at 3.0V Freq: 200 kHz max.	VDD: 3.0V to 6.0V IDD: 48 μA max. at 32 kHz, 3.0V IPD: 13.5 μA max. at 3.0V Freq: 200 kHz max.

The shaded sections indicate oscillator selections which are tested for functionality, but not for MIN/MAX specifications. It is recommended that the user select the device type that ensures the specifications required.

# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

## 18.1 DC Characteristics: PIC16C73/74-04 (Commercial, Industrial) PIC16C73/74-10 (Commercial, Industrial) PIC16C73/74-20 (Commercial, 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 and $0^{\circ}\text{C} \leq T_A \leq +70^{\circ}\text{C}$ for commercial					
Param No.	Characteristic	Sym	Min	Typ†	Max	Units	Conditions
D001 D001A	Supply Voltage	VDD	4.0 4.5	- -	6.0 5.5	V V	XT, RC and LP osc configuration HS osc configuration
D002*	RAM Data Retention Voltage (Note 1)	VDR	-	1.5	-	V	
D003	VDD start voltage to ensure internal Power-on Reset signal	VPOR	-	VSS	-	V	See section on Power-on Reset for details
D004*	VDD rise rate to ensure internal Power-on Reset signal	SVDD	0.05	-	-	V/ms	See section on Power-on Reset for details
D010  D013	Supply Current (Note 2,5)	IDD	- -	2.7 13.5	5 30	mA mA	XT, RC osc configuration FOSC = 4 MHz, VDD = 5.5V (Note 4)  HS osc configuration FOSC = 20 MHz, VDD = 5.5V
D020 D021 D021A	Power-down Current (Note 3,5)	IPD	- - -	10.5 1.5 1.5	42 21 24	$\mu\text{A}$ $\mu\text{A}$ $\mu\text{A}$	VDD = 4.0V, WDT enabled, $-40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$ VDD = 4.0V, WDT disabled, $-0^{\circ}\text{C}$ to $+70^{\circ}\text{C}$ VDD = 4.0V, WDT disabled, $-40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$

\* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V,  $25^{\circ}\text{C}$  unless otherwise stated. These parameters are for design guidance only and are not tested.

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

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 tristated, 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  $20\mu\text{A}$  to the specification. This value is from characterization and is for design guidance only. This is not tested.



# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

## 18.2 DC Characteristics: PIC16LC73/74-04 (Commercial, 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 and $0^{\circ}\text{C} \leq T_A \leq +70^{\circ}\text{C}$ for commercial					
Param No.	Characteristic	Sym	Min	Typ†	Max	Units	Conditions
D001	Supply Voltage	VDD	3.0	-	6.0	V	LP, XT, RC osc configuration (DC - 4 MHz)
D002*	RAM Data Retention Voltage (Note 1)	VDR	-	1.5	-	V	
D003	VDD start voltage to ensure internal Power-on Reset signal	VPOR	-	VSS	-	V	See section on Power-on Reset for details
D004*	VDD rise rate to ensure internal Power-on Reset signal	SVDD	0.05	-	-	V/ms	See section on Power-on Reset for details
D010	Supply Current (Note 2,5)	IDD	-	2.0	3.8	mA	XT, RC osc configuration FOSC = 4 MHz, VDD = 3.0V (Note 4)
D010A			-	22.5	48	$\mu\text{A}$	LP osc configuration FOSC = 32 kHz, VDD = 3.0V, WDT disabled
D020	Power-down Current (Note 3,5)	IPD	-	7.5	30	$\mu\text{A}$	VDD = 3.0V, WDT enabled, $-40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$
D021			-	0.9	13.5	$\mu\text{A}$	VDD = 3.0V, WDT disabled, $0^{\circ}\text{C}$ to $+70^{\circ}\text{C}$
D021A			-	0.9	18	$\mu\text{A}$	VDD = 3.0V, WDT disabled, $-40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$

\* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V,  $25^{\circ}\text{C}$  unless otherwise stated. These parameters are for design guidance only and are not tested.

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

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 tristated, 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  $20\mu\text{A}$  to the specification. This value is from characterization and is for design guidance only. This is not tested.

# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

**18.3 DC Characteristics:** PIC16C73/74-04 (Commercial, Industrial)  
 PIC16C73/74-10 (Commercial, Industrial)  
 PIC16C73/74-20 (Commercial, Industrial)  
 PIC16LC73/74-04 (Commercial, 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 and $0^{\circ}\text{C} \leq T_A \leq +70^{\circ}\text{C}$ for commercial Operating voltage VDD range as described in DC spec Section 18.1 and Section 18.2.					
Param No.	Characteristic	Sym	Min	Typ †	Max	Units	Conditions
D030 D030A D031 D032 D033	<b>Input Low Voltage</b> I/O ports with TTL buffer with Schmitt Trigger buffer MCLR, OSC1 (in RC mode) OSC1 (in XT, HS and LP)	VIL	VSS VSS VSS VSS	- - - -	0.15VDD 0.8V 0.2VDD 0.2VDD 0.3VDD	V V V V V	For entire VDD range $4.5\text{V} \leq V_{DD} \leq 5.5\text{V}$ Note1
D040 D040A D041 D042 D042A D043	<b>Input High Voltage</b> I/O ports with TTL buffer with Schmitt Trigger buffer MCLR OSC1 (XT, HS and LP) OSC1 (in RC mode)	VIH	2.0 0.25VDD + 0.8V 0.8VDD 0.8VDD 0.7VDD 0.9VDD	- - - - - -	VDD VDD VDD VDD VDD VDD	V V V V V V	$4.5\text{V} \leq V_{DD} \leq 5.5\text{V}$ For entire VDD range For entire VDD range Note1
D070	PORTB weak pull-up current	IPURB	50	250	400	μA	VDD = 5V, VPIN = VSS
D060 D061 D063	<b>Input Leakage Current</b> (Notes 2, 3) I/O ports MCLR, RA4/T0CKI OSC1	IIL	- - -	- - -	±1 ±5 ±5	μA μA μA	VSS ≤ VPIN ≤ VDD, Pin at hi-impedance VSS ≤ VPIN ≤ VDD VSS ≤ VPIN ≤ VDD, XT, HS and LP osc configuration
D080 D083	<b>Output Low Voltage</b> I/O ports OSC2/CLKOUT (RC osc config)	VOL	- -	- -	0.6 0.6	V V	IOI = 8.5 mA, VDD = 4.5V, -40°C to +85°C IOI = 1.6 mA, VDD = 4.5V, -40°C to +85°C
D090 D092	<b>Output High Voltage</b> I/O ports (Note 3) OSC2/CLKOUT (RC osc config)	VOH	VDD - 0.7 VDD - 0.7	- -	- -	V V	IOH = -3.0 mA, VDD = 4.5V, -40°C to +85°C IOH = -1.3 mA, VDD = 4.5V, -40°C to +85°C
D150*	<b>Open-Drain High Voltage</b>	VOD	-	-	14	V	RA4 pin

\* 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 PIC16C7X 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.

# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

DC CHARACTERISTICS		Standard Operating Conditions (unless otherwise stated)					
		Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial and $0^{\circ}\text{C} \leq T_A \leq +70^{\circ}\text{C}$ for commercial					
		Operating voltage $V_{DD}$ range as described in DC spec Section 18.1 and Section 18.2.					
Param No.	Characteristic	Sym	Min	Typ †	Max	Units	Conditions
	<b>Capacitive Loading Specs on Output Pins</b>						
D100	OSC2 pin	COSC2	-	-	15	pF	In XT, HS and LP modes when external clock is used to drive OSC1.
D101	All I/O pins and OSC2 (in RC mode) SCL, SDA in I <sup>2</sup> C mode	CIO	-	-	50	pF	
D102		CB	-	-	400	pF	

\* 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 PIC16C7X 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.

# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

## 18.4 Timing Parameter Symbology

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

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

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

Lowercase letters (pp) and their meanings:

<b>pp</b>	
cc      CCP1	osc      OSC1
ck      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 $\overline{WR}$

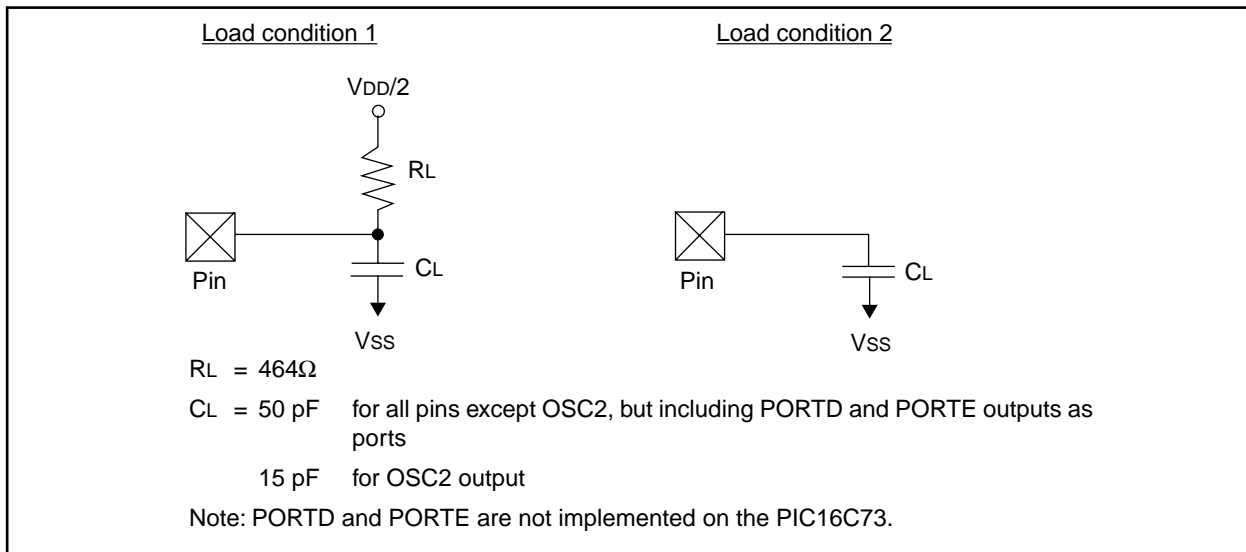
Uppercase letters and their meanings:

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

Tcc:st (I<sup>2</sup>C specifications only)

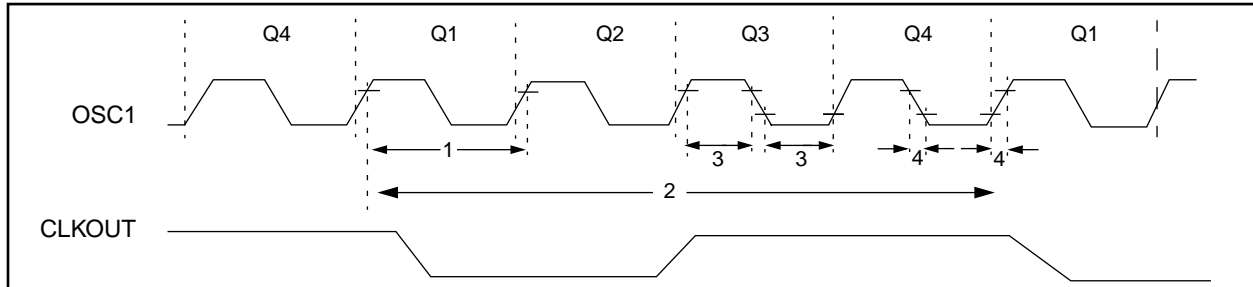
<b>CC</b>	
HD      Hold	SU      Setup
<b>ST</b>	
DAT      DATA input hold	STO      STOP condition
STA      START condition	

**FIGURE 18-1: LOAD CONDITIONS**



## 18.5 Timing Diagrams and Specifications

**FIGURE 18-2: EXTERNAL CLOCK TIMING**



**TABLE 18-2: EXTERNAL CLOCK TIMING REQUIREMENTS**

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions	
	Fosc	<b>External CLKIN Frequency (Note 1)</b>	DC	—	4	MHz	XT and RC osc mode	
			DC	—	4	MHz	HS osc mode (-04)	
			DC	—	10	MHz	HS osc mode (-10)	
			DC	—	20	MHz	HS osc mode (-20)	
			DC	—	200	kHz	LP osc mode	
			<b>Oscillator Frequency (Note 1)</b>	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	<b>External CLKIN Period (Note 1)</b>	250	—
250	—	—	ns	HS osc mode (-04)				
100	—	—	ns	HS osc mode (-10)				
50	—	—	ns	HS osc mode (-20)				
5	—	—	μs	LP osc mode				
		<b>Oscillator Period (Note 1)</b>	250	—		—	ns	RC osc mode
			250	—		10,000	ns	XT osc mode
			250	—		250	ns	HS osc mode (-04)
			100	—		250	ns	HS osc mode (-10)
			50	—		250	ns	HS osc mode (-20)
2	Tcy	<b>Instruction Cycle Time (Note 1)</b>	200	—	DC	ns	Tcy = 4/Fosc	
3	TosL, TosH	<b>External Clock in (OSC1) High or Low Time</b>	50	—	—	ns	XT oscillator	
			2.5	—	—	μs	LP oscillator	
			15	—	—	ns	HS oscillator	
4	TosR, TosF	<b>External Clock in (OSC1) Rise or Fall Time</b>	—	—	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.

# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

FIGURE 18-3: CLKOUT AND I/O TIMING

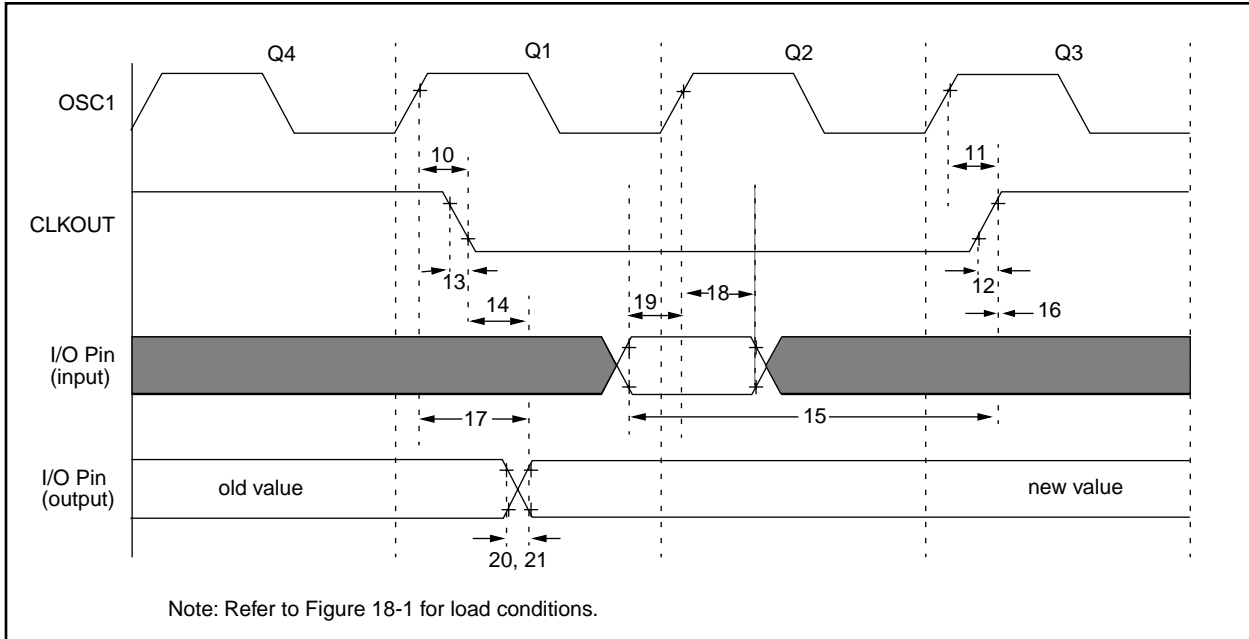


TABLE 18-3: CLKOUT AND I/O TIMING REQUIREMENTS

Parameter No.	Sym	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	—	—	0.5TCY + 20	ns	Note 1	
15*	TioV2ckH	Port in valid before CLKOUT ↑	0.25TCY + 25	—	—	ns	Note 1	
16*	TckH2iol	Port in hold after CLKOUT ↑	0	—	—	ns	Note 1	
17*	TosH2ioV	OSC1↑ (Q1 cycle) to Port out valid	—	50	150	ns		
18*	TosH2iol	OSC1↑ (Q2 cycle) to Port input invalid (I/O in hold time)	PIC16C73/74	100	—	—	ns	
			PIC16LC73/74	200	—	—	ns	
19*	TioV2osH	Port input valid to OSC1↑ (I/O in setup time)	0	—	—	ns		
20*	TioR	Port output rise time	PIC16C73/74	—	10	25	ns	
			PIC16LC73/74	—	—	60	ns	
21*	TioF	Port output fall time	PIC16C73/74	—	10	25	ns	
			PIC16LC73/74	—	—	60	ns	
22††*	Tinp	INT pin high or low time	TCY	—	—	ns		
23††*	Trbp	RB7:RB4 change INT high or low time	TCY	—	—	ns		

\* These parameters are characterized but not tested.

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

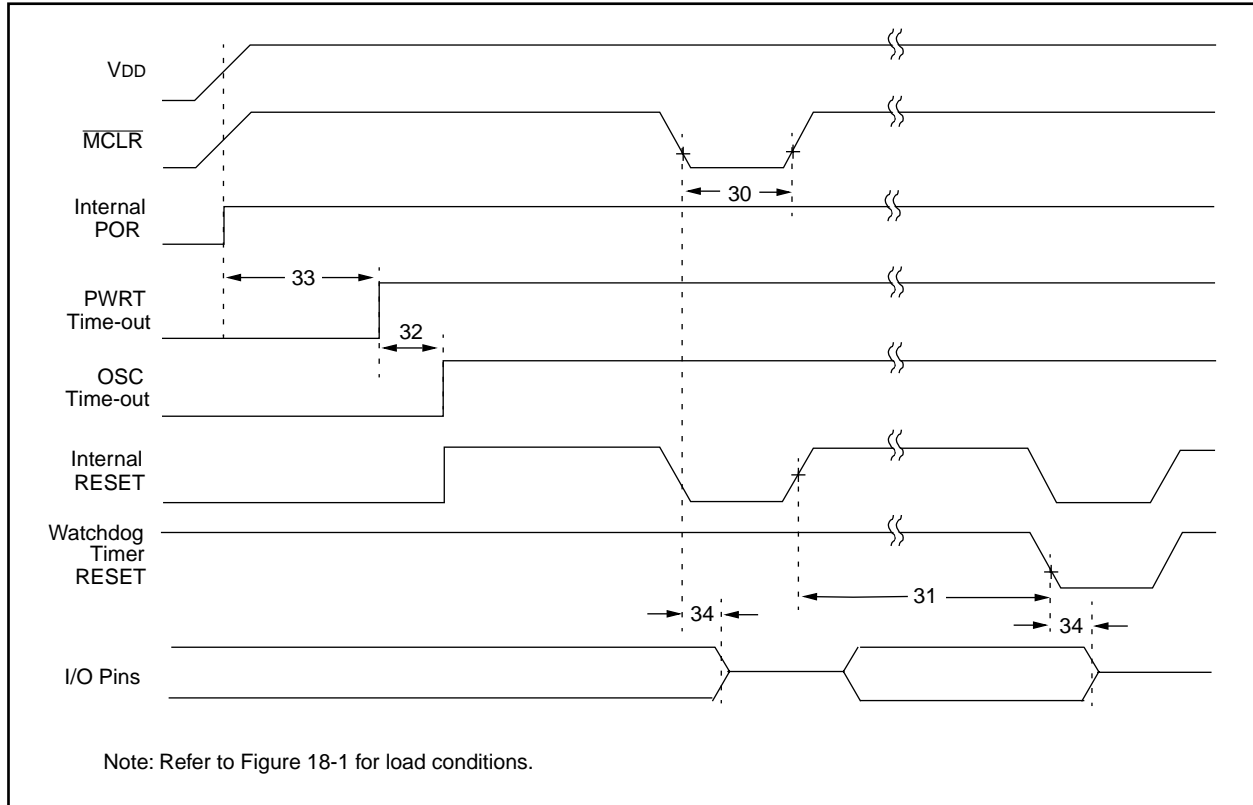
†† These parameters are asynchronous events not related to any internal clock edges.

Note 1: Measurements are taken in RC Mode where CLKOUT output is 4 x TOSC.

# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

**FIGURE 18-4: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING**



**TABLE 18-4: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER REQUIREMENTS**

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
30	TmCL	MCLR Pulse Width (low)	100	—	—	ns	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	—	1024TOSC	—	—	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	—	—	100	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.

# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

FIGURE 18-5: TIMER0 AND TIMER1 EXTERNAL CLOCK TIMINGS

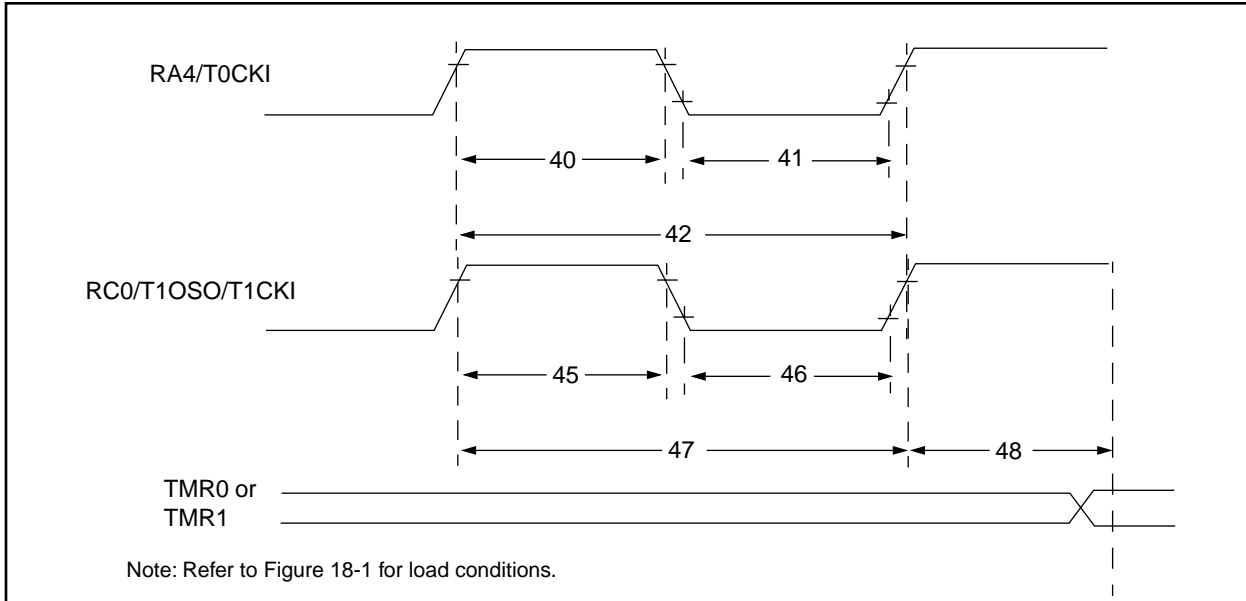


TABLE 18-5: TIMER0 AND TIMER1 EXTERNAL CLOCK REQUIREMENTS

Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions	
40*	Tt0H	T0CKI High Pulse Width	No Prescaler $0.5T_{CY} + 20$ With Prescaler 10	—	—	ns	Must also meet parameter 42	
41*	Tt0L	T0CKI Low Pulse Width	No Prescaler $0.5T_{CY} + 20$ With Prescaler 10	—	—	ns		
42*	Tt0P	T0CKI Period	No Prescaler $T_{CY} + 40$ With Prescaler Greater of: $20$ or $T_{CY} + 40$ N	—	—	ns	N = prescale value (2, 4, ..., 256)	
45*	Tt1H	T1CKI High Time	Synchronous, Prescaler = 1 $0.5T_{CY} + 20$ Synchronous, Prescaler = 2,4,8 PIC16C7X 15 PIC16LC7X 25 Asynchronous PIC16C7X 30 PIC16LC7X 50	—	—	ns	Must also meet parameter 47	
46*	Tt1L	T1CKI Low Time	Synchronous, Prescaler = 1 $0.5T_{CY} + 20$ Synchronous, Prescaler = 2,4,8 PIC16C7X 15 PIC16LC7X 25 Asynchronous PIC16C7X 30 PIC16LC7X 50	—	—	ns		
47*	Tt1P	T1CKI input period	Synchronous PIC16C7X Greater of: $30$ or $T_{CY} + 40$ N PIC16LC7X Greater of: $50$ or $T_{CY} + 40$ N Asynchronous PIC16C7X 60 PIC16LC7X 100	—	—	ns		N = prescale value (1, 2, 4, 8)
	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	$2T_{osc}$	—	$7T_{osc}$	—		

\* These parameters are characterized but not tested.

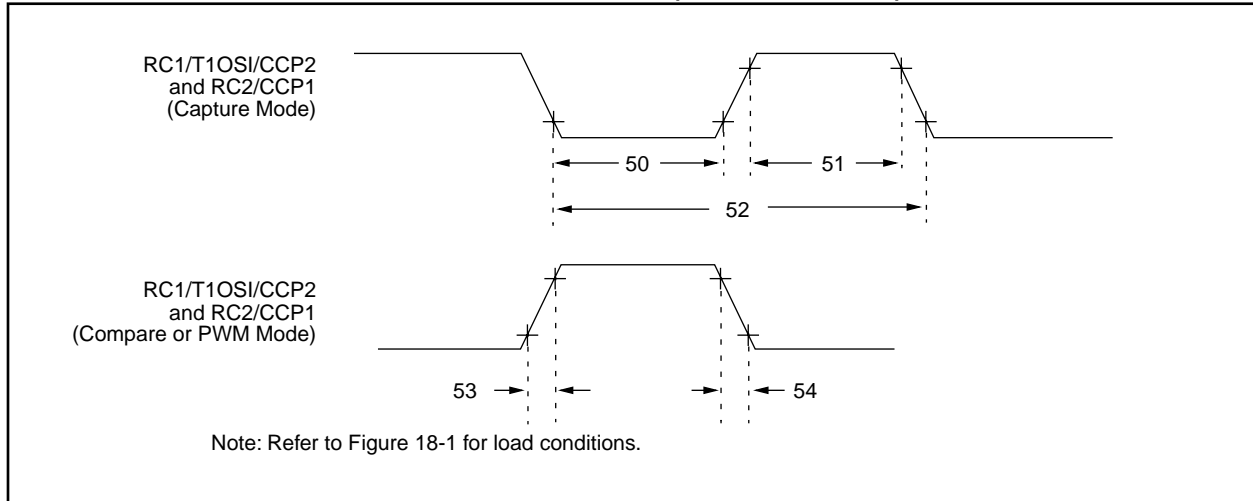
† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.



# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

**FIGURE 18-6: CAPTURE/COMPARE/PWM TIMINGS (CCP1 AND CCP2)**



**TABLE 18-6: CAPTURE/COMPARE/PWM REQUIREMENTS (CCP1 AND CCP2)**

Parameter No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions	
50*	TccL	CCP1 and CCP2 input low time	No Prescaler	$0.5T_{CY} + 20$	—	—	ns		
			With Prescaler	PIC16C73/74	10	—	—		ns
				PIC16LC73/74	20	—	—		ns
51*	TccH	CCP1 and CCP2 input high time	No Prescaler	$0.5T_{CY} + 20$	—	—	ns		
			With Prescaler	PIC16C73/74	10	—	—		ns
				PIC16LC73/74	20	—	—		ns
52*	TccP	CCP1 and CCP2 input period		$\frac{3T_{CY} + 40}{N}$	—	—	ns	N = prescale value (1,4 or 16)	
53*	TccR	CCP1 and CCP2 output fall time	PIC16C73/74	—	10	25	ns		
			PIC16LC73/74	—	25	45	ns		
54*	TccF	CCP1 and CCP2 output fall time	PIC16C73/74	—	10	25	ns		
			PIC16LC73/74	—	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.

# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

FIGURE 18-7: PARALLEL SLAVE PORT TIMING (PIC16C74)

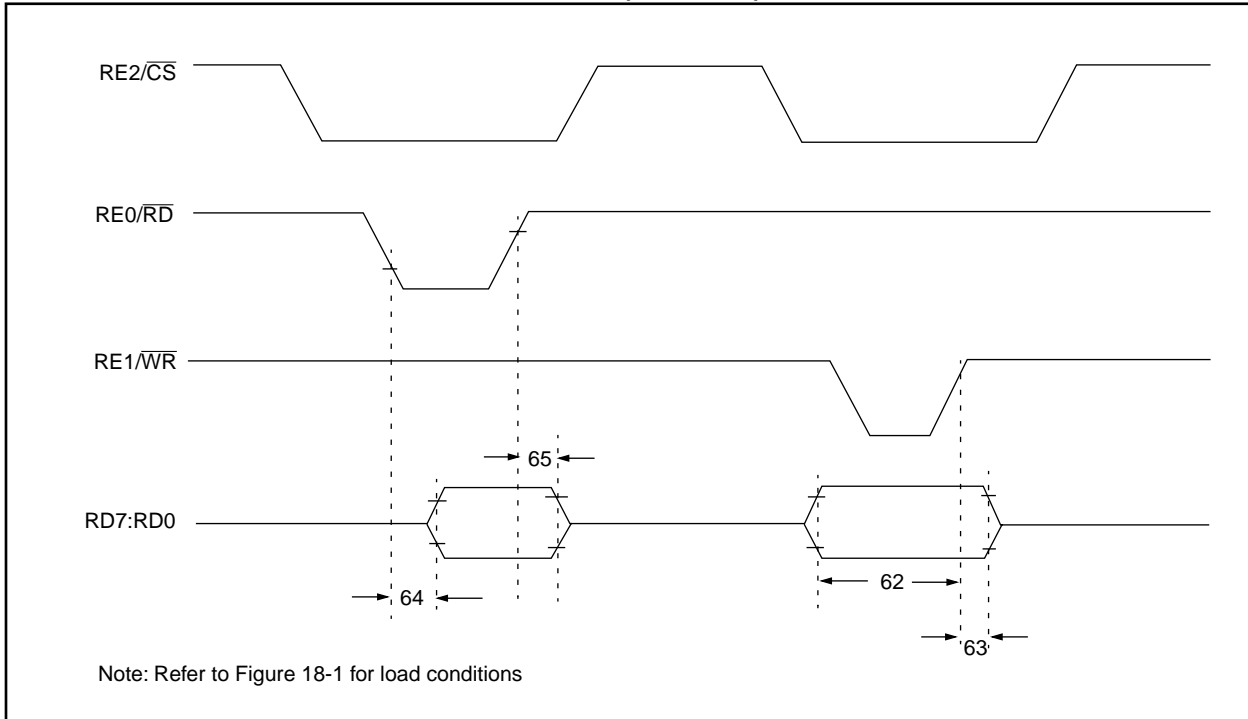


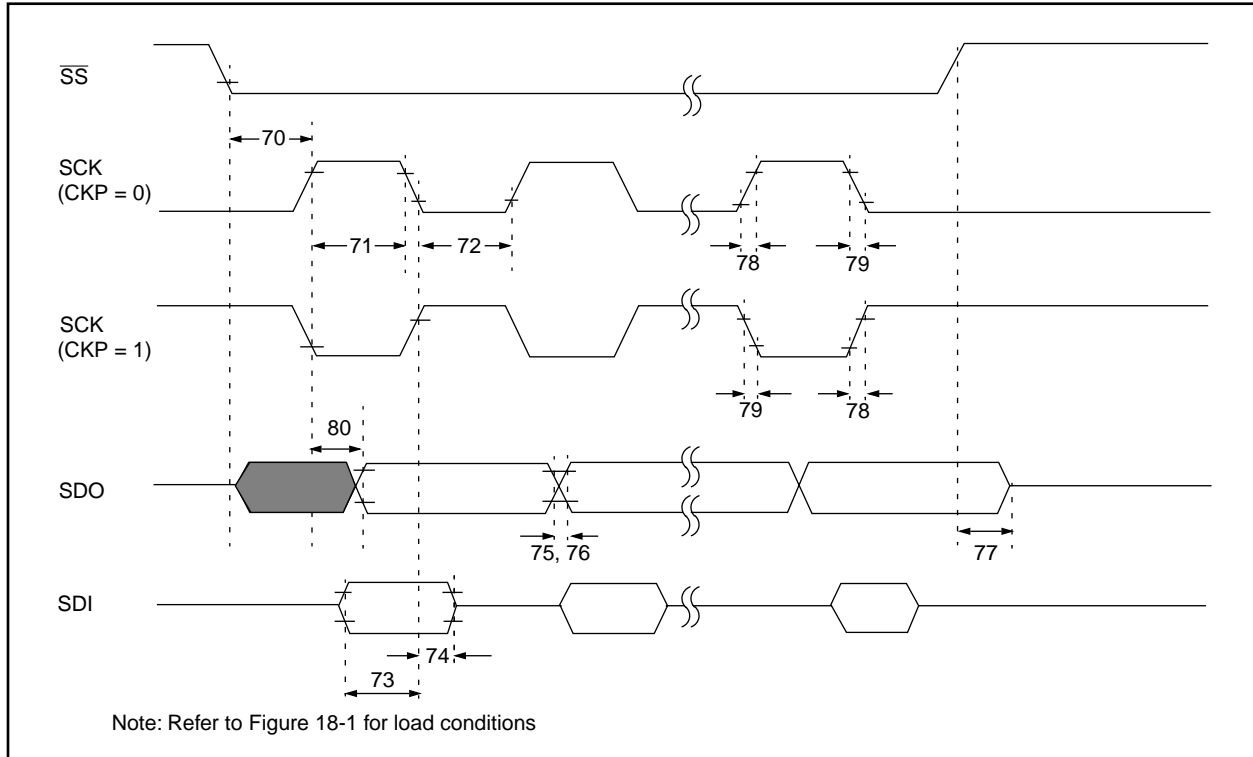
TABLE 18-7: PARALLEL SLAVE PORT REQUIREMENTS (PIC16C74)

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions	
62	TdtV2wrH	Data in valid before $\overline{WR}\uparrow$ or $\overline{CS}\uparrow$ (setup time)	20	—	—	ns		
63*	TwrH2dtI	$\overline{WR}\uparrow$ or $\overline{CS}\uparrow$ to data-in invalid (hold time)	PIC16C74	20	—	—	ns	
			PIC16LC74	35	—	—	ns	
64	TrdL2dtV	$\overline{RD}\downarrow$ and $\overline{CS}\downarrow$ to data-out valid	—	—	80	ns		
65	TrdH2dtI	$\overline{RD}\uparrow$ or $\overline{CS}\downarrow$ to data-out invalid	10	—	30	ns		

\* These parameters are characterized but not tested.

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

**FIGURE 18-8: SPI MODE TIMING**



**TABLE 18-8: SPI MODE REQUIREMENTS**

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
70	TssL2scl, TssL2sclL	$\overline{SS}\downarrow$ to SCK $\downarrow$ or SCK $\uparrow$ input	T <sub>cy</sub>	—	—	ns	
71	Tsch	SCK input high time (slave mode)	T <sub>cy</sub> + 20	—	—	ns	
72	Tscl	SCK input low time (slave mode)	T <sub>cy</sub> + 20	—	—	ns	
73	TdiV2scl, TdiV2sclL	Setup time of SDI data input to SCK edge	50	—	—	ns	
74	Tsch2diL, TscL2diL	Hold time of SDI data input to SCK edge	50	—	—	ns	
75	TdoR	SDO data output rise time	—	10	25	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)	—	10	25	ns	
79	TscF	SCK output fall time (master mode)	—	10	25	ns	
80	Tsch2doV, TscL2doV	SDO data output valid after SCK edge	—	—	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.

# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

FIGURE 18-9: I<sup>2</sup>C BUS START/STOP BITS TIMING

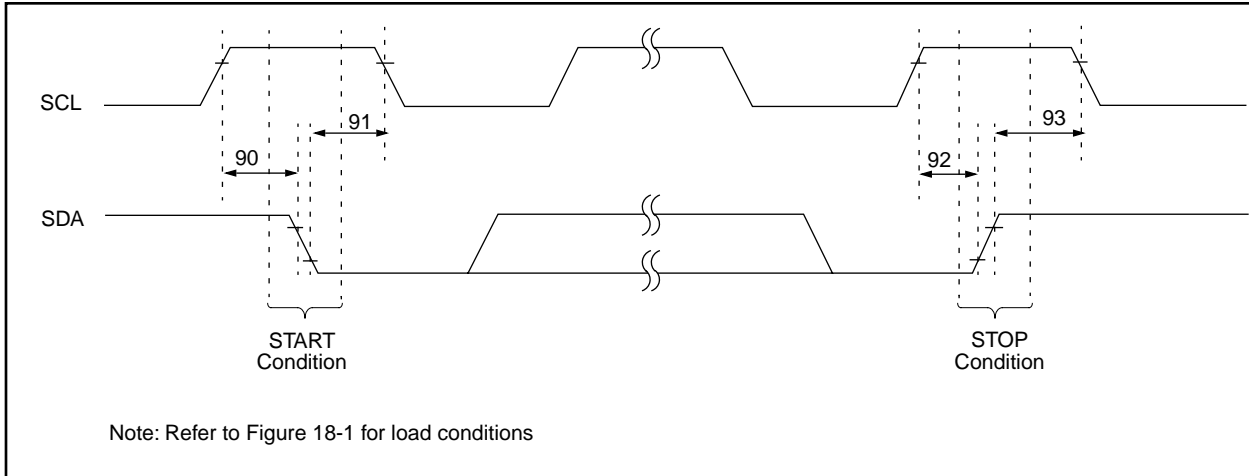


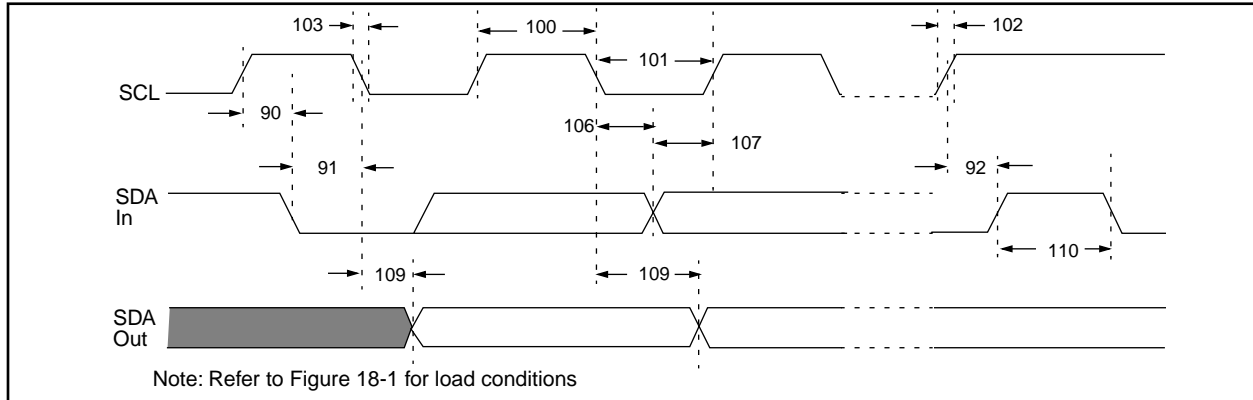
TABLE 18-9: I<sup>2</sup>C BUS START/STOP BITS REQUIREMENTS

Parameter No.	Sym	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	—	—			

# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

**FIGURE 18-10: I<sup>2</sup>C BUS DATA TIMING**



**TABLE 18-10: I<sup>2</sup>C BUS DATA REQUIREMENTS**

Parameter No.	Sym	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 to 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 to 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		

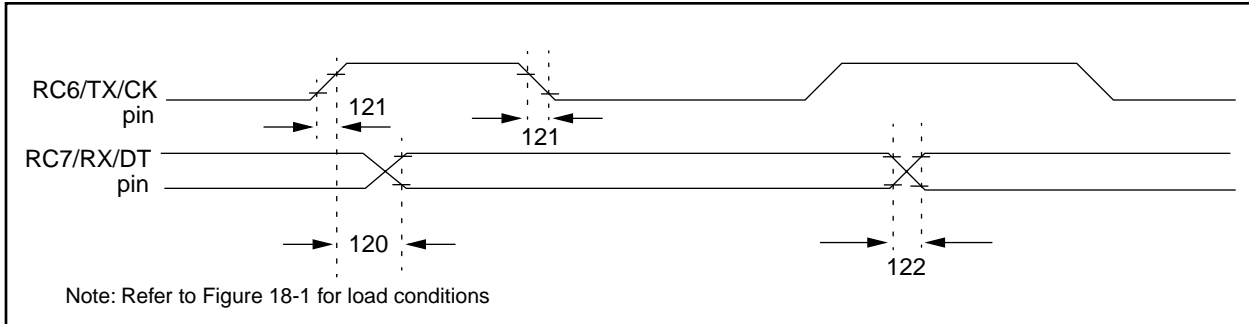
Note 1: As a transmitter, the device must provide this internal minimum delay time to bridge the undefined region (min. 300 ns) of the falling edge of SCL to avoid unintended generation of START or STOP conditions.

2: A fast-mode (400 kHz) I<sup>2</sup>C-bus device can be used in a standard-mode (100 kHz) I<sup>2</sup>C-bus system, but the requirement tsu;DAT ≥ 250 ns must then be met. This will automatically be the case if the device does not stretch the LOW period of the SCL signal. If such a device does stretch the LOW period of the SCL signal, it must output the next data bit to the SDA line Tr max.+tsu;DAT = 1000 + 250 = 1250 ns (according to the standard-mode I<sup>2</sup>C bus specification) before the SCL line is released.

# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

**FIGURE 18-11: USART SYNCHRONOUS TRANSMISSION (MASTER/SLAVE) TIMING**

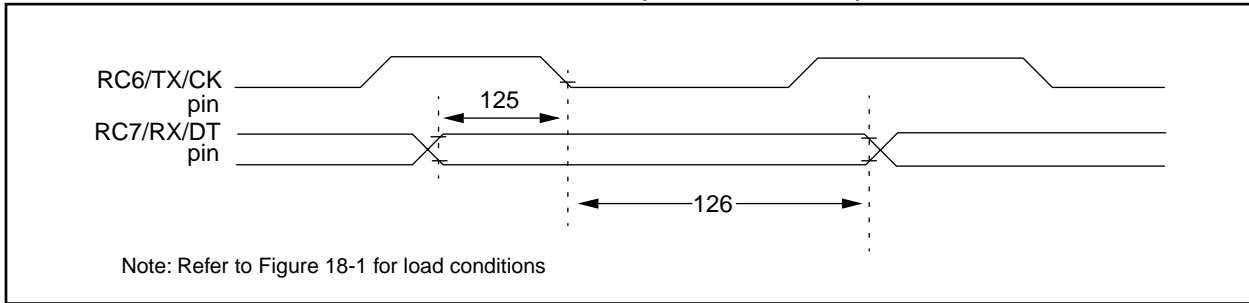


**TABLE 18-11: USART SYNCHRONOUS TRANSMISSION REQUIREMENTS**

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
120	TckH2dtV	SYNC XMIT (MASTER & SLAVE) Clock high to data out valid	PIC16C73/74	—	—	80	ns
			PIC16LC73/74	—	—	100	ns
121	Tckrf	Clock out rise time and fall time (Master Mode)	PIC16C73/74	—	—	45	ns
			PIC16LC73/74	—	—	50	ns
122	Tdtrf	Data out rise time and fall time	PIC16C73/74	—	—	45	ns
			PIC16LC73/74	—	—	50	ns

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

**FIGURE 18-12: USART SYNCHRONOUS RECEIVE (MASTER/SLAVE) TIMING**



**TABLE 18-12: USART SYNCHRONOUS RECEIVE REQUIREMENTS**

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
125	TdtV2ckL	SYNC RCV (MASTER & SLAVE) Data setup before CK ↓ (DT setup time)	15	—	—	ns	
126	TckL2dtl	Data hold after CK ↓ (DT hold time)	15	—	—	ns	

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

# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

**TABLE 18-13: A/D CONVERTER CHARACTERISTICS:**

**PIC16C73/74-04 (Commercial, Industrial)**  
**PIC16C73/74-10 (Commercial, Industrial)**  
**PIC16C73/74-20 (Commercial, Industrial)**  
**PIC16LC73/74-04 (Commercial, Industrial)**

Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions	
A01	NR	Resolution	—	—	8-bits	bit	VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF	
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	E0FF	Offset error	—	—	< ± 1	LSb	VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF	
A10	—	Monotonicity	—	guaranteed	—	—	VSS ≤ VAIN ≤ VREF	
A20	VREF	Reference voltage	3.0V	—	VDD + 0.3	V		
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)	PIC16C73/74	—	180	—	μA	Average current consumption when A/D is on. (Note 1)
			PIC16LC73/74	—	90	—	μA	
A50	IREF	VREF input current (Note 2)	10	—	1000	μA	During VAIN acquisition. Based on differential of VHOLD to VAIN to charge CHOLD, see Section 13.1.	
			—	—	10	μA	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 RA3 pin or VDD pin, whichever is selected as reference input.

# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

FIGURE 18-13: A/D CONVERSION TIMING

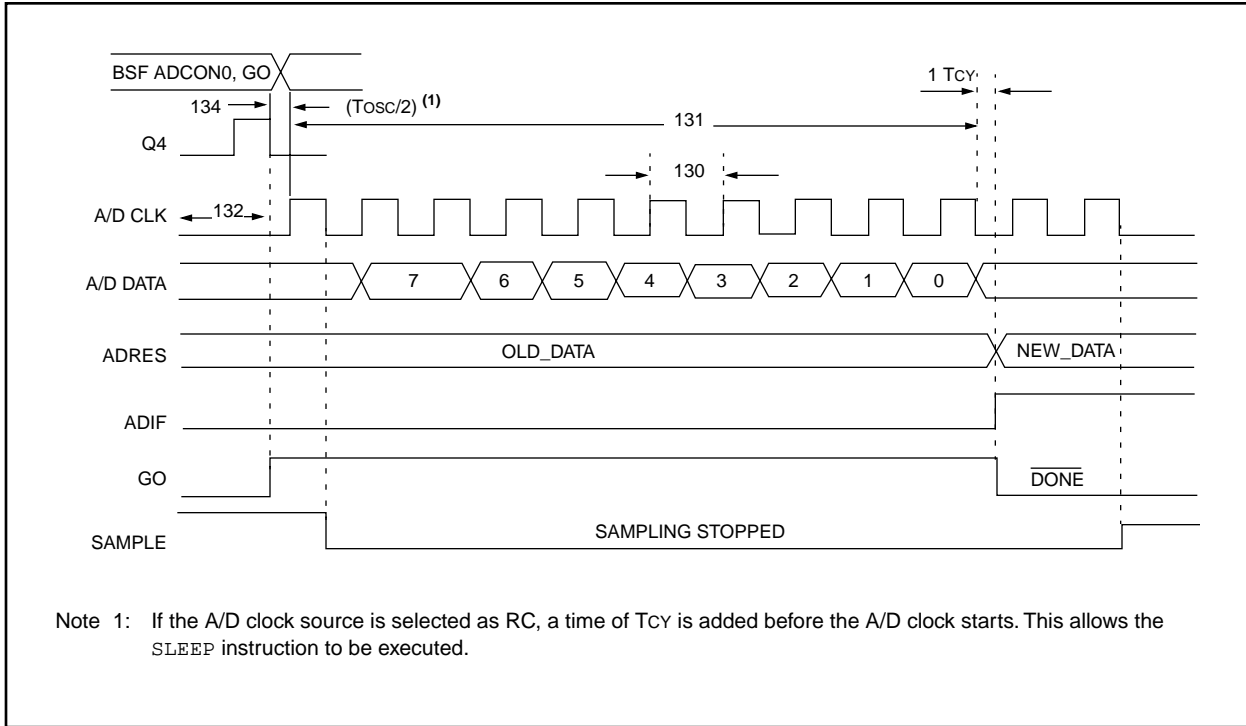


TABLE 18-14: A/D CONVERSION REQUIREMENTS

Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions	
130	TAD	A/D clock period	PIC16C73/74	1.6	—	—	$\mu\text{s}$	TOSC based, $V_{REF} \geq 3.0\text{V}$
			PIC16LC73/74	2.0	—	—	$\mu\text{s}$	TOSC based, $V_{REF}$ full range
			PIC16C73/74	2.0	4.0	6.0	$\mu\text{s}$	A/D RC Mode
			PIC16LC73/74	3.0	6.0	9.0	$\mu\text{s}$	A/D RC Mode
131	T <sub>CVN</sub>	Conversion time (not including S/H time) (Note 1)	—	9.5	—	TAD		
132	TACQ	Acquisition time	Note 2	20	—	$\mu\text{s}$	The minimum time is the amplifier settling time. This may be used if the "new" input voltage has not changed by more than 1 LSB (i.e., 20 mV @ 5.12V) from the last sampled voltage (as stated on CHOLD).	
			5*	—	—	$\mu\text{s}$		
134	T <sub>GO</sub>	Q4 to A/D clock start	—	$T_{osc}/2$ §	—	—	If the A/D clock source is selected as RC, a time of $T_{cy}$ is added before the A/D clock starts. This allows the SLEEP instruction to be executed.	
135	T <sub>SWC</sub>	Switching from convert → sample time	1.5 §	—	—	TAD		

\* These parameters are characterized but not tested.

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

§ This specification ensured by design.

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

Note 2: See Section 13.1 for min conditions.



# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

## 19.0 ELECTRICAL CHARACTERISTICS FOR PIC16C73A/74A

### 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 Vss (except VDD, $\overline{\text{MCLR}}$ , and RA4).....	-0.3V to (VDD + 0.3V)
Voltage on VDD with respect to Vss .....	-0.3 to +7.5V
Voltage on $\overline{\text{MCLR}}$ with respect to Vss (Note 2) .....	0 to +14V
Voltage on RA4 with respect to Vss .....	0 to +14V
Total power dissipation (Note 1).....	1.0W
Maximum current out of Vss pin .....	300 mA
Maximum current into VDD pin .....	250 mA
Input clamp current, I <sub>IK</sub> (V <sub>I</sub> < 0 or V <sub>I</sub> > VDD) .....	±20 mA
Output clamp current, I <sub>OK</sub> (V <sub>O</sub> < 0 or V <sub>O</sub> > VDD).....	±20 mA
Maximum output current sunk by any I/O pin.....	25 mA
Maximum output current sourced by any I/O pin .....	25 mA
Maximum current sunk by PORTA, 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:  $P_{dis} = V_{DD} \times \{I_{DD} - \sum IOH\} + \sum \{(V_{DD} - V_{OH}) \times IOH\} + \sum (V_{OL} \times I_{OL})$

**Note 2:** Voltage spikes below Vss at the  $\overline{\text{MCLR}}$  pin, inducing currents greater than 80 mA, may cause latch-up. Thus, a series resistor of 50-100Ω should be used when applying a “low” level to the  $\overline{\text{MCLR}}$  pin rather than pulling this pin directly to Vss.

**Note 3:** PORTD and PORTE are not implemented on the PIC16C73A.

† 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.

**TABLE 19-1: CROSS REFERENCE OF DEVICE SPECS FOR OSCILLATOR CONFIGURATIONS AND FREQUENCIES OF OPERATION (COMMERCIAL DEVICES)**

OSC	PIC16C73A-04 PIC16C74A-04	PIC16C73A-10 PIC16C74A-10	PIC16C73A-20 PIC16C74A-20	PIC16LC73A-04 PIC16LC74A-04	JW Devices
RC	VDD: 4.0V to 6.0V IDD: 5 mA max. at 5.5V IPD: 16 μA max. at 4V Freq: 4 MHz max.	VDD: 4.5V to 5.5V IDD: 2.7 mA typ. at 5.5V IPD: 1.5 μA typ. at 4V Freq: 4 MHz max.	VDD: 4.5V to 5.5V IDD: 2.7 mA typ. at 5.5V IPD: 1.5 μA typ. at 4V Freq: 4 MHz max.	VDD: 2.5V to 6.0V IDD: 3.8 mA max. at 3.0V IPD: 5 μA max. at 3V Freq: 4 MHz max.	VDD: 4.0V to 6.0V IDD: 5 mA max. at 5.5V IPD: 16 μA max. at 4V Freq: 4 MHz max.
XT	VDD: 4.0V to 6.0V IDD: 5 mA max. at 5.5V IPD: 16 μA max. at 4V Freq: 4 MHz max.	VDD: 4.5V to 5.5V IDD: 2.7 mA typ. at 5.5V IPD: 1.5 μA typ. at 4V Freq: 4 MHz max.	VDD: 4.5V to 5.5V IDD: 2.7 mA typ. at 5.5V IPD: 1.5 μA typ. at 4V Freq: 4 MHz max.	VDD: 2.5V to 6.0V IDD: 3.8 mA max. at 3.0V IPD: 5 μA max. at 3V Freq: 4 MHz max.	VDD: 4.0V to 6.0V IDD: 5 mA max. at 5.5V IPD: 16 μA max. at 4V Freq: 4 MHz max.
HS	VDD: 4.5V to 5.5V IDD: 13.5 mA typ. at 5.5V IPD: 1.5 μA typ. at 4.5V Freq: 4 MHz max.	VDD: 4.5V to 5.5V IDD: 10 mA max. at 5.5V IPD: 1.5 μA typ. at 4.5V Freq: 10 MHz max.	VDD: 4.5V to 5.5V IDD: 20 mA max. at 5.5V IPD: 1.5 μA typ. at 4.5V Freq: 20 MHz max.	Not recommended for use in HS mode	VDD: 4.5V to 5.5V IDD: 20 mA max. at 5.5V IPD: 1.5 μA typ. at 4.5V Freq: 20 MHz max.
LP	VDD: 4.0V to 6.0V IDD: 52.5 μA typ. at 32 kHz, 4.0V IPD: 0.9 μA typ. at 4.0V Freq: 200 kHz max.	Not recommended for use in LP mode	Not recommended for use in LP mode	VDD: 2.5V to 6.0V IDD: 48 μA max. at 32 kHz, 3.0V IPD: 5.0 μA max. at 3.0V Freq: 200 kHz max.	VDD: 2.5V to 6.0V IDD: 48 μA max. at 32 kHz, 3.0V IPD: 5.0 μA max. at 3.0V Freq: 200 kHz max.

The shaded sections indicate oscillator selections which are tested for functionality, but not for MIN/MAX specifications. It is recommended that the user select the device type that ensures the specifications required.

# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

**19.1 DC Characteristics: PIC16C73A/74A-04 (Commercial, Industrial, Extended)**  
**PIC16C73A/74A-10 (Commercial, Industrial, Extended)**  
**PIC16C73A/74A-20 (Commercial, Industrial, Extended)**

DC CHARACTERISTICS		Standard Operating Conditions (unless otherwise stated)					
		Operating temperature -40°C ≤ TA ≤ +125°C for extended, -40°C ≤ TA ≤ +85°C for industrial and 0°C ≤ TA ≤ +70°C for commercial					
Param No.	Characteristic	Sym	Min	Typ†	Max	Units	Conditions
D001 D001A	Supply Voltage	VDD	4.0 4.5	- -	6.0 5.5	V V	XT, RC and LP osc configuration HS osc configuration
D002*	RAM Data Retention Voltage (Note 1)	VDR	-	1.5	-	V	
D003	VDD start voltage to ensure internal Power-on Reset signal	VPOR	-	VSS	-	V	See section on Power-on Reset for details
D004*	VDD rise rate to ensure internal Power-on Reset signal	SVDD	0.05	-	-	V/ms	See section on Power-on Reset for details
D005	Brown-out Reset Voltage	BVDD	3.7 3.7	4.0 4.0	4.3 4.4	V V	BODEN bit in configuration word enabled Extended Range Only
D010  D013	Supply Current (Note 2,5)	IDD	- -	2.7 10	5 20	mA mA	XT, RC osc configuration FOSC = 4 MHz, VDD = 5.5V (Note 4)  HS osc configuration FOSC = 20 MHz, VDD = 5.5V
D015*	Brown-out Reset Current (Note 6)	ΔIBOR	-	350	425	μA	BOR enabled VDD = 5.0V
D020 D021 D021A D021B	Power-down Current (Note 3,5)	IPD	- - - -	10.5 1.5 1.5 2.5	42 16 19 19	μA μA μA μA	VDD = 4.0V, WDT enabled, -40°C to +85°C VDD = 4.0V, WDT disabled, -0°C to +70°C VDD = 4.0V, WDT disabled, -40°C to +85°C VDD = 4.0V, WDT disabled, -40°C to +125°C
D023*	Brown-out Reset Current (Note 6)	ΔIBOR	-	350	425	μA	BOR enabled VDD = 5.0V

\* 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 tristated, 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.

# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

## 19.2 DC Characteristics: PIC16LC73A/74A-04 (Commercial, Industrial)

DC CHARACTERISTICS		Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial and 0°C ≤ TA ≤ +70°C for commercial					
Param No.	Characteristic	Sym	Min	Typ†	Max	Units	Conditions
D001	Supply Voltage	VDD	2.5	-	6.0	V	LP, XT, RC osc configuration (DC - 4 MHz)
D002*	RAM Data Retention Voltage (Note 1)	VDR	-	1.5	-	V	
D003	VDD start voltage to ensure internal Power-on Reset signal	VPOR	-	VSS	-	V	See section on Power-on Reset for details
D004*	VDD rise rate to ensure internal Power-on Reset signal	SVDD	0.05	-	-	V/ms	See section on Power-on Reset for details
D005	Brown-out Reset Voltage	BVDD	3.7	4.0	4.3	V	BODEN bit in configuration word enabled
D010	Supply Current (Note 2,5)	IDD	-	2.0	3.8	mA	XT, RC osc configuration FOSC = 4 MHz, VDD = 3.0V (Note 4)
D010A				-	22.5	48	μA
D015*	Brown-out Reset Current (Note 6)	ΔIBOR	-	350	425	μA	BOR enabled VDD = 5.0V
D020	Power-down Current (Note 3,5)	IPD	-	7.5	30	μA	VDD = 3.0V, WDT enabled, -40°C to +85°C
D021				0.9	5	μA	VDD = 3.0V, WDT disabled, 0°C to +70°C
D021A				0.9	5	μA	VDD = 3.0V, WDT disabled, -40°C to +85°C
D023*	Brown-out Reset Current (Note 6)	ΔIBOR	-	350	425	μA	BOR enabled VDD = 5.0V

\* 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 tristated, 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.

# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

19.3 DC Characteristics: **PIC16C73A/74A-04 (Commercial, Industrial, Extended)**  
**PIC16C73A/74A-10 (Commercial, Industrial, Extended)**  
**PIC16C73A/74A-20 (Commercial, Industrial, Extended)**  
**PIC16LC73A/74A-04 (Commercial, Industrial)**

Standard Operating Conditions (unless otherwise stated)							
DC CHARACTERISTICS							
		Operating temperature		-40°C	≤ T <sub>A</sub> ≤ +125°C for extended, -40°C ≤ T <sub>A</sub> ≤ +85°C for industrial and 0°C ≤ T <sub>A</sub> ≤ +70°C for commercial		
		Operating voltage V <sub>DD</sub> range as described in DC spec Section 19.1 and Section 19.2.					
Param No.	Characteristic	Sym	Min	Typ †	Max	Units	Conditions
D030 D030A D031 D032 D033	<b>Input Low Voltage</b> I/O ports with TTL buffer with Schmitt Trigger buffer MCLR, OSC1 (in RC mode) OSC1 (in XT, HS and LP)	V <sub>IL</sub>	V <sub>SS</sub>	-	0.15V <sub>DD</sub> 0.8V 0.2V <sub>DD</sub> 0.2V <sub>DD</sub> 0.3V <sub>DD</sub>	V	For entire V <sub>DD</sub> range 4.5V ≤ V <sub>DD</sub> ≤ 5.5V Note1
D040 D040A D041 D042 D042A D043	<b>Input High Voltage</b> I/O ports with TTL buffer with Schmitt Trigger buffer MCLR OSC1 (XT, HS and LP) OSC1 (in RC mode)	V <sub>IH</sub>	2.0 0.25V <sub>DD</sub> + 0.8V 0.8V <sub>DD</sub> 0.8V <sub>DD</sub> 0.7V <sub>DD</sub> 0.9V <sub>DD</sub>	-	V <sub>DD</sub> V <sub>DD</sub> V <sub>DD</sub> V <sub>DD</sub> V <sub>DD</sub> V <sub>DD</sub> V <sub>DD</sub>	V	4.5V ≤ V <sub>DD</sub> ≤ 5.5V For entire V <sub>DD</sub> range For entire V <sub>DD</sub> range Note1
D070	PORTB weak pull-up current	I <sub>PURB</sub>	50	250	400	μA	V <sub>DD</sub> = 5V, V <sub>PIN</sub> = V <sub>SS</sub>
D060 D061 D063	<b>Input Leakage Current</b> (Notes 2, 3) I/O ports MCLR, RA4/T0CKI OSC1	I <sub>IL</sub>	-	-	±1 ±5 ±5	μA	V <sub>SS</sub> ≤ V <sub>PIN</sub> ≤ V <sub>DD</sub> , Pin at hi-impedance V <sub>SS</sub> ≤ V <sub>PIN</sub> ≤ V <sub>DD</sub> V <sub>SS</sub> ≤ V <sub>PIN</sub> ≤ V <sub>DD</sub> , XT, HS and LP osc configuration
D080 D080A D083 D083A	<b>Output Low Voltage</b> I/O ports OSC2/CLKOUT (RC osc config)	V <sub>OL</sub>	-	-	0.6 0.6 0.6 0.6	V	I <sub>OL</sub> = 8.5 mA, V <sub>DD</sub> = 4.5V, -40°C to +85°C I <sub>OL</sub> = 7.0 mA, V <sub>DD</sub> = 4.5V, -40°C to +125°C I <sub>OL</sub> = 1.6 mA, V <sub>DD</sub> = 4.5V, -40°C to +85°C I <sub>OL</sub> = 1.2 mA, V <sub>DD</sub> = 4.5V, -40°C to +125°C

\* These parameters are characterized but not tested.

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

- Note 1: In RC oscillator configuration, the OSC1/CLKIN pin is a Schmitt Trigger input. It is not recommended that the PIC16C7X 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.

# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

Standard Operating Conditions (unless otherwise stated)							
DC CHARACTERISTICS		Operating temperature					
		-40°C ≤ TA ≤ +125°C for extended,					
		-40°C ≤ TA ≤ +85°C for industrial and 0°C ≤ TA ≤ +70°C for commercial					
		Operating voltage VDD range as described in DC spec Section 19.1 and Section 19.2.					
Param No.	Characteristic	Sym	Min	Typ †	Max	Units	Conditions
	<b>Output High Voltage</b>						
D090	I/O ports (Note 3)	VOH	VDD - 0.7	-	-	V	IOH = -3.0 mA, VDD = 4.5V, -40°C to +85°C IOH = -2.5 mA, VDD = 4.5V, -40°C to +125°C IOH = -1.3 mA, VDD = 4.5V, -40°C to +85°C IOH = -1.0 mA, VDD = 4.5V, -40°C to +125°C
D090A			VDD - 0.7	-	-	V	
D092	OSC2/CLKOUT (RC osc config)		VDD - 0.7	-	-	V	
D092A			VDD - 0.7	-	-	V	
D150*	<b>Open-Drain High Voltage</b>	VOD	-	-	14	V	RA4 pin
	<b>Capacitive Loading Specs on Output Pins</b>						
D100	OSC2 pin	COsc2	-	-	15	pF	In XT, HS and LP modes when external clock is used to drive OSC1.
D101	All I/O pins and OSC2 (in RC mode)	CI/O	-	-	50	pF	
D102	SCL, SDA in I <sup>2</sup> C mode	CB	-	-	400	pF	

\* 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 PIC16C7X 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.

# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

## 19.4 Timing Parameter Symbology

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

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

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

Lowercase letters (pp) and their meanings:

<b>pp</b>	
cc      CCP1	osc      OSC1
ck      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 $\overline{WR}$

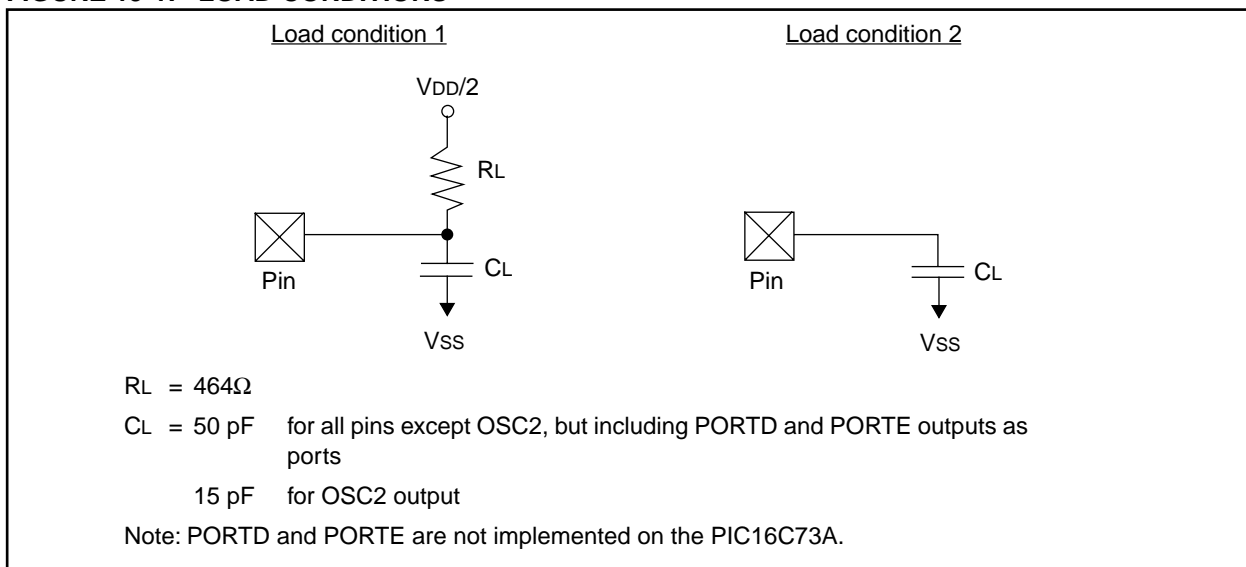
Uppercase letters and their meanings:

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

Tcc:st (I<sup>2</sup>C specifications only)

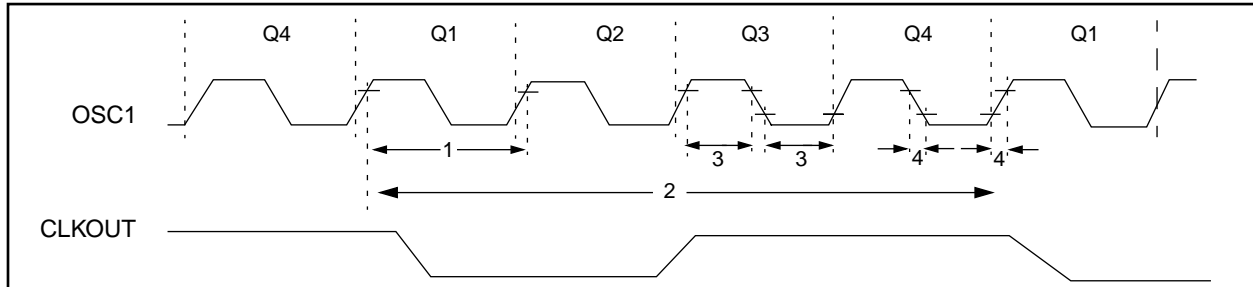
<b>CC</b>	
HD      Hold	SU      Setup
<b>ST</b>	
DAT      DATA input hold	STO      STOP condition
STA      START condition	

**FIGURE 19-1: LOAD CONDITIONS**



## 19.5 Timing Diagrams and Specifications

**FIGURE 19-2: EXTERNAL CLOCK TIMING**



**TABLE 19-2: EXTERNAL CLOCK TIMING REQUIREMENTS**

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
	Fosc	<b>External CLKIN Frequency (Note 1)</b>	DC	—	4	MHz	XT and RC osc mode
			DC	—	4	MHz	HS osc mode (-04)
			DC	—	10	MHz	HS osc mode (-10)
			DC	—	20	MHz	HS osc mode (-20)
			DC	—	200	kHz	LP osc mode
		<b>Oscillator Frequency (Note 1)</b>	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	<b>External CLKIN Period (Note 1)</b>	250	—	—	ns	XT and RC osc mode
			250	—	—	ns	HS osc mode (-04)
			100	—	—	ns	HS osc mode (-10)
			50	—	—	ns	HS osc mode (-20)
			5	—	—	μs	LP osc mode
		<b>Oscillator Period (Note 1)</b>	250	—	—	ns	RC osc mode
			250	—	10,000	ns	XT osc mode
			250	—	250	ns	HS osc mode (-04)
			100	—	250	ns	HS osc mode (-10)
			50	—	250	ns	HS osc mode (-20)
			5	—	—	μs	LP osc mode
2	Tcy	<b>Instruction Cycle Time (Note 1)</b>	200	Tcy	DC	ns	Tcy = 4/Fosc
3	TosL, TosH	<b>External Clock in (OSC1) High or Low Time</b>	100	—	—	ns	XT oscillator
			2.5	—	—	μs	LP oscillator
			15	—	—	ns	HS oscillator
4	TosR, TosF	<b>External Clock in (OSC1) Rise or Fall Time</b>	—	—	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.

# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

FIGURE 19-3: CLKOUT AND I/O TIMING

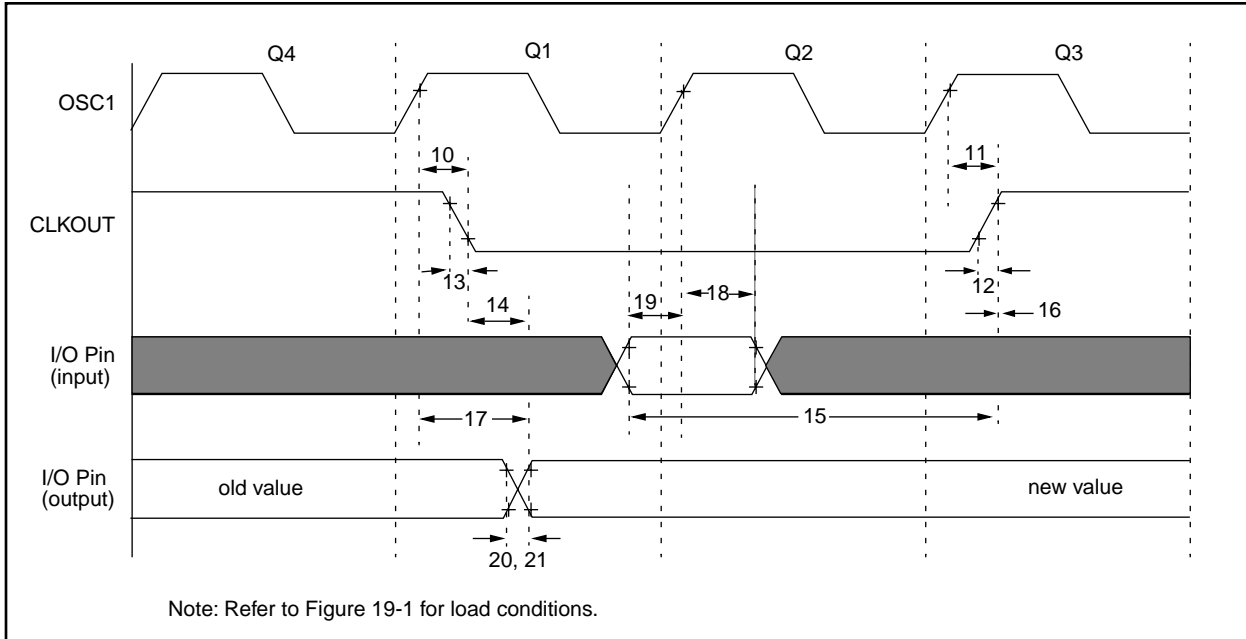


TABLE 19-3: CLKOUT AND I/O TIMING REQUIREMENTS

Param No.	Sym	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	—	—	0.5TCY + 20	ns	Note 1	
15*	TioV2ckH	Port in valid before CLKOUT ↑	Tosc + 200	—	—	ns	Note 1	
16*	TckH2iol	Port in hold after CLKOUT ↑	0	—	—	ns	Note 1	
17*	TosH2ioV	OSC1↑ (Q1 cycle) to Port out valid	—	50	150	ns		
18*	TosH2iol	OSC1↑ (Q2 cycle) to Port input invalid (I/O in hold time)	PIC16C73A/74A	100	—	—	ns	
			PIC16LC73A/74A	200	—	—	ns	
19*	TioV2osH	Port input valid to OSC1↑ (I/O in setup time)	0	—	—	ns		
20*	TioR	Port output rise time	PIC16C73A/74A	—	10	40	ns	
			PIC16LC73A/74A	—	—	80	ns	
21*	TioF	Port output fall time	PIC16C73A/74A	—	10	40	ns	
			PIC16LC73A/74A	—	—	80	ns	
22††*	Tinp	INT pin high or low time	TCY	—	—	ns		
23††*	Trbp	RB7:RB4 change INT high or low time	TCY	—	—	ns		

\* These parameters are characterized but not tested.

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

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

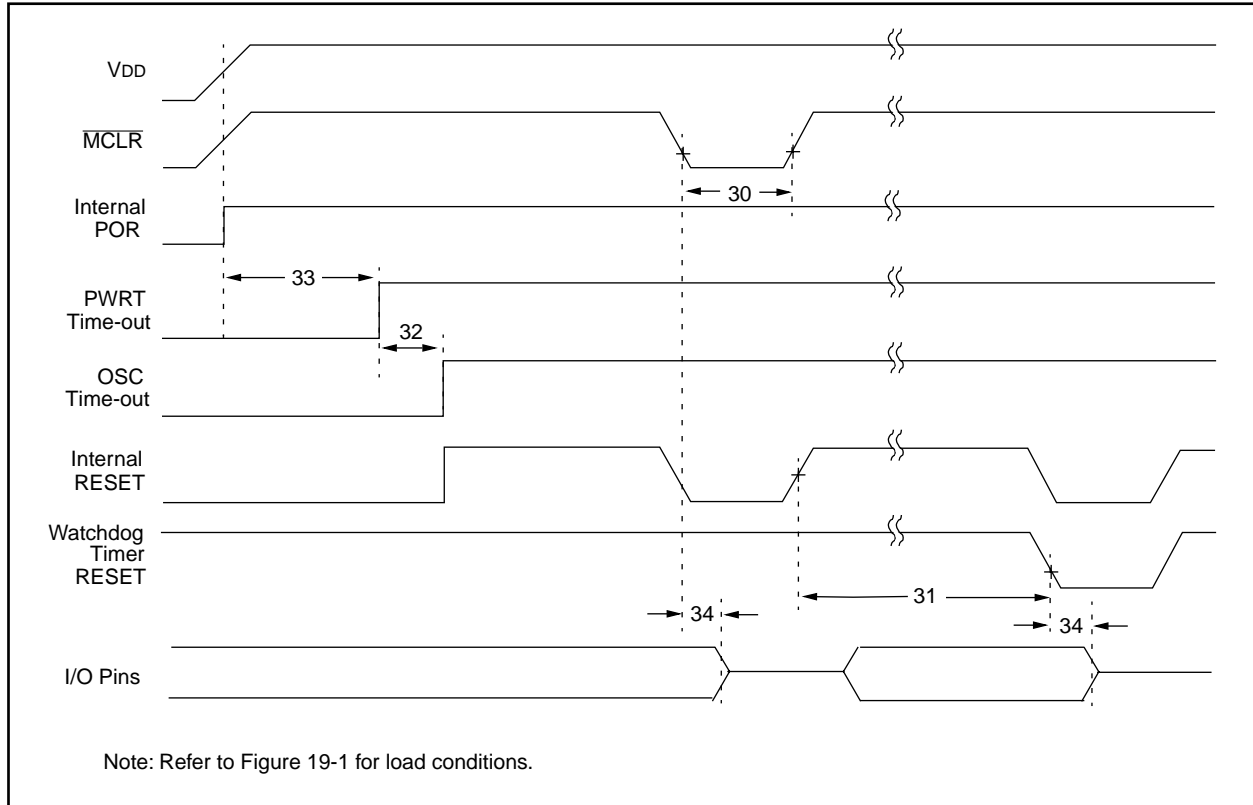
Note 1: Measurements are taken in RC Mode where CLKOUT output is 4 x Tosc.



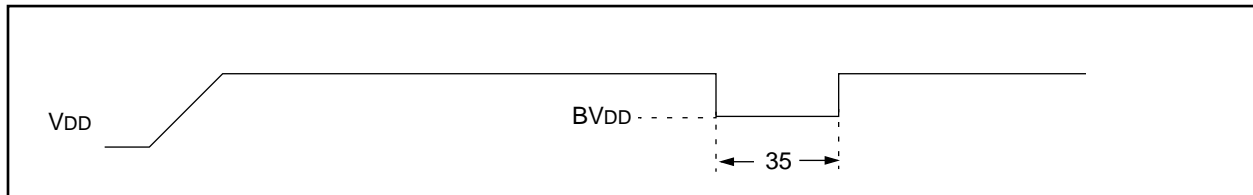
# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

**FIGURE 19-4: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING**



**FIGURE 19-5: BROWN-OUT RESET TIMING**



**TABLE 19-4: 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 +125°C
31*	Twdt	Watchdog Timer Time-out Period (No Prescaler)	7	18	33	ms	VDD = 5V, -40°C to +125°C
32	Tost	Oscillation Start-up Timer Period	—	1024Tosc	—	—	Tosc = OSC1 period
33*	Tpwrt	Power up Timer Period	28	72	132	ms	VDD = 5V, -40°C to +125°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 ≤ BVDD (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.

# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

FIGURE 19-6: TIMER0 AND TIMER1 EXTERNAL CLOCK TIMINGS

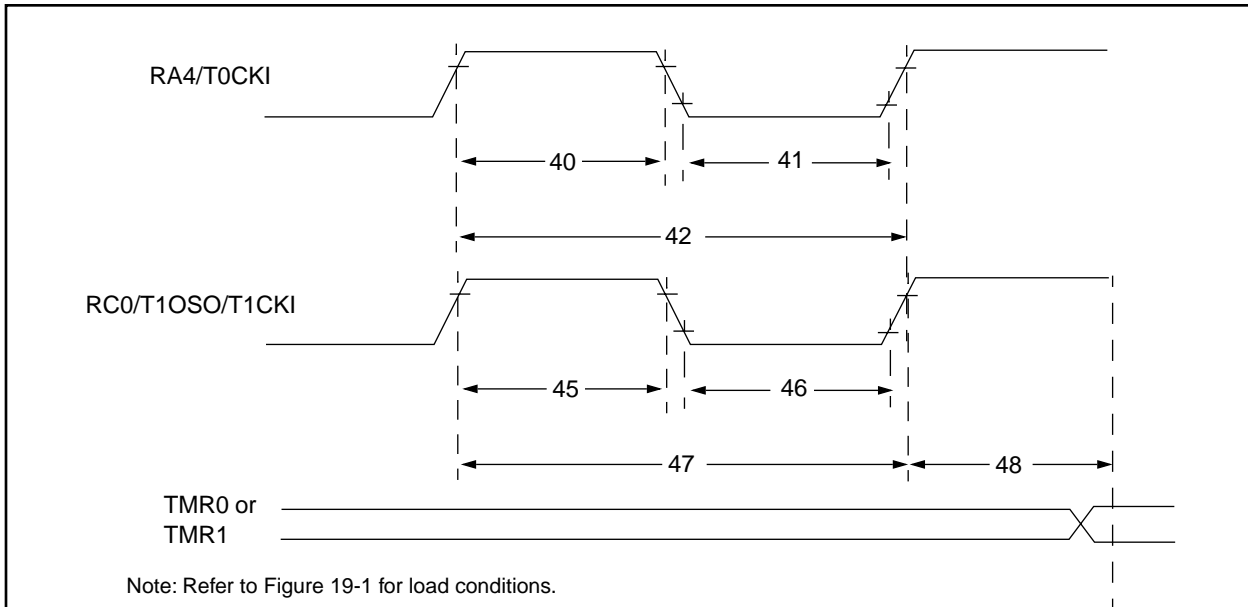


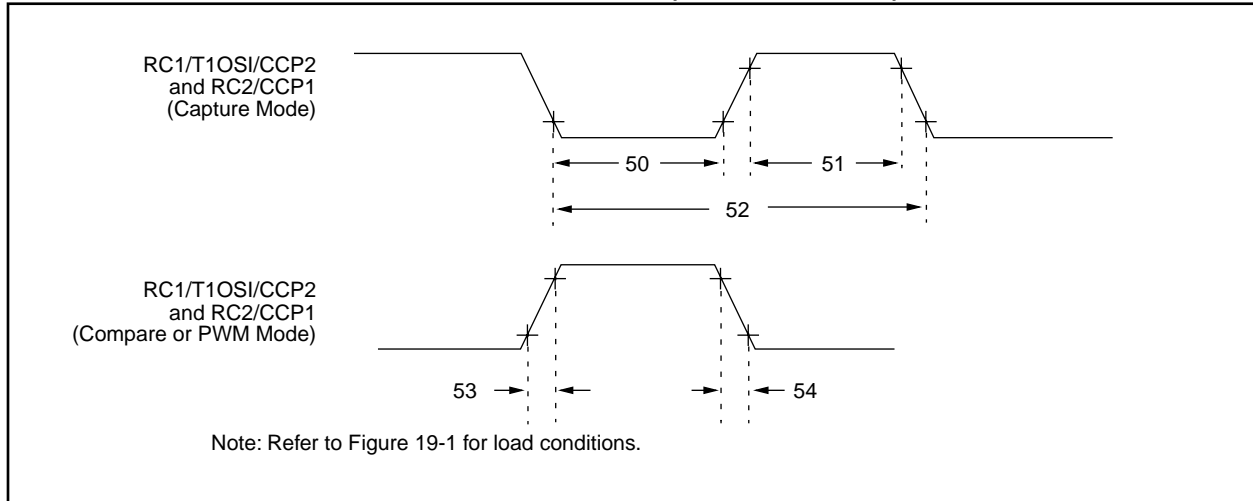
TABLE 19-5: TIMER0 AND TIMER1 EXTERNAL CLOCK REQUIREMENTS

Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
40*	Tt0H	TOCKI High Pulse Width	No Prescaler $0.5T_{CY} + 20$ With Prescaler 10	—	—	ns	Must also meet parameter 42
41*	Tt0L	TOCKI Low Pulse Width	No Prescaler $0.5T_{CY} + 20$ With Prescaler 10	—	—	ns	
42*	Tt0P	TOCKI Period	No Prescaler $T_{CY} + 40$ With Prescaler Greater of: $20$ or $\frac{T_{CY} + 40}{N}$	—	—	ns	N = prescale value (2, 4, ..., 256)
45*	Tt1H	T1CKI High Time	Synchronous, Prescaler = 1 PIC16C7X $0.5T_{CY} + 20$	—	—	ns	Must also meet parameter 47
			Synchronous, Prescaler = 2,4,8 PIC16LC7X 15	—	—	ns	
			Asynchronous PIC16C7X 30 PIC16LC7X 50	—	—	ns	
46*	Tt1L	T1CKI Low Time	Synchronous, Prescaler = 1 PIC16C7X $0.5T_{CY} + 20$	—	—	ns	Must also meet parameter 47
			Synchronous, Prescaler = 2,4,8 PIC16LC7X 15	—	—	ns	
			Asynchronous PIC16C7X 30 PIC16LC7X 50	—	—	ns	
47*	Tt1P	T1CKI input period	Synchronous PIC16C7X Greater of: $30$ OR $\frac{T_{CY} + 40}{N}$	—	—	ns	N = prescale value (1, 2, 4, 8)
			PIC16LC7X Greater of: $50$ OR $\frac{T_{CY} + 40}{N}$	—	—	ns	N = prescale value (1, 2, 4, 8)
			Asynchronous PIC16C7X 60 PIC16LC7X 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	$2T_{osc}$	—	$7T_{osc}$	—	

\* These parameters are characterized but not tested.

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

**FIGURE 19-7: CAPTURE/COMPARE/PWM TIMINGS (CCP1 AND CCP2)**



**TABLE 19-6: CAPTURE/COMPARE/PWM REQUIREMENTS (CCP1 AND CCP2)**

Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions					
50*	TccL	CCP1 and CCP2 input low time	No Prescaler	$0.5T_{CY} + 20$	—	—	ns					
			With Prescaler	PIC16C73A/74A	10	—	—	ns				
							PIC16LC73A/74A	20	—	—	ns	
51*	TccH	CCP1 and CCP2 input high time	No Prescaler	$0.5T_{CY} + 20$	—	—	ns					
			With Prescaler	PIC16C73A/74A	10	—	—	ns				
							PIC16LC73A/74A	20	—	—	ns	
52*	TccP	CCP1 and CCP2 input period	$\frac{3T_{CY} + 40}{N}$	—	—	ns	N = prescale value (1,4 or 16)					
53*	TccR	CCP1 and CCP2 output rise time	PIC16C73A/74A	—	10	25	ns					
			PIC16LC73A/74A	—	25	45	ns					
54*	TccF	CCP1 and CCP2 output fall time	PIC16C73A/74A	—	10	25	ns					
			PIC16LC73A/74A	—	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.

# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

FIGURE 19-8: PARALLEL SLAVE PORT TIMING (PIC16C74A)

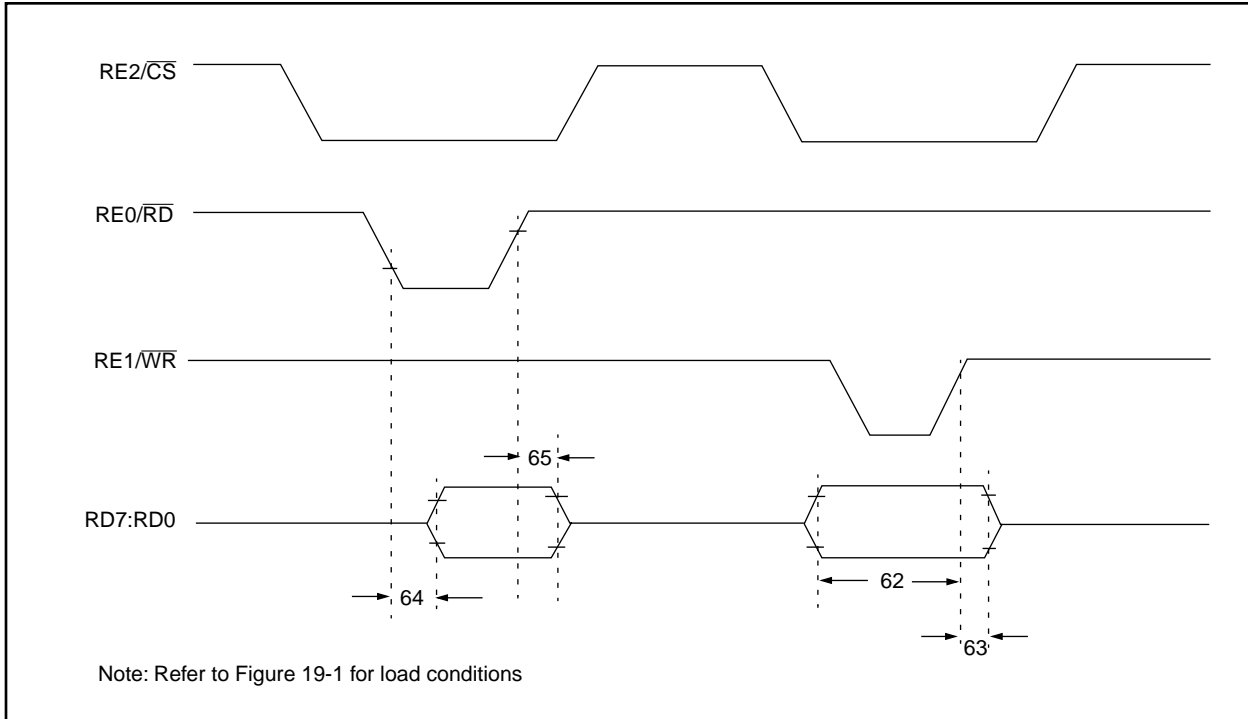


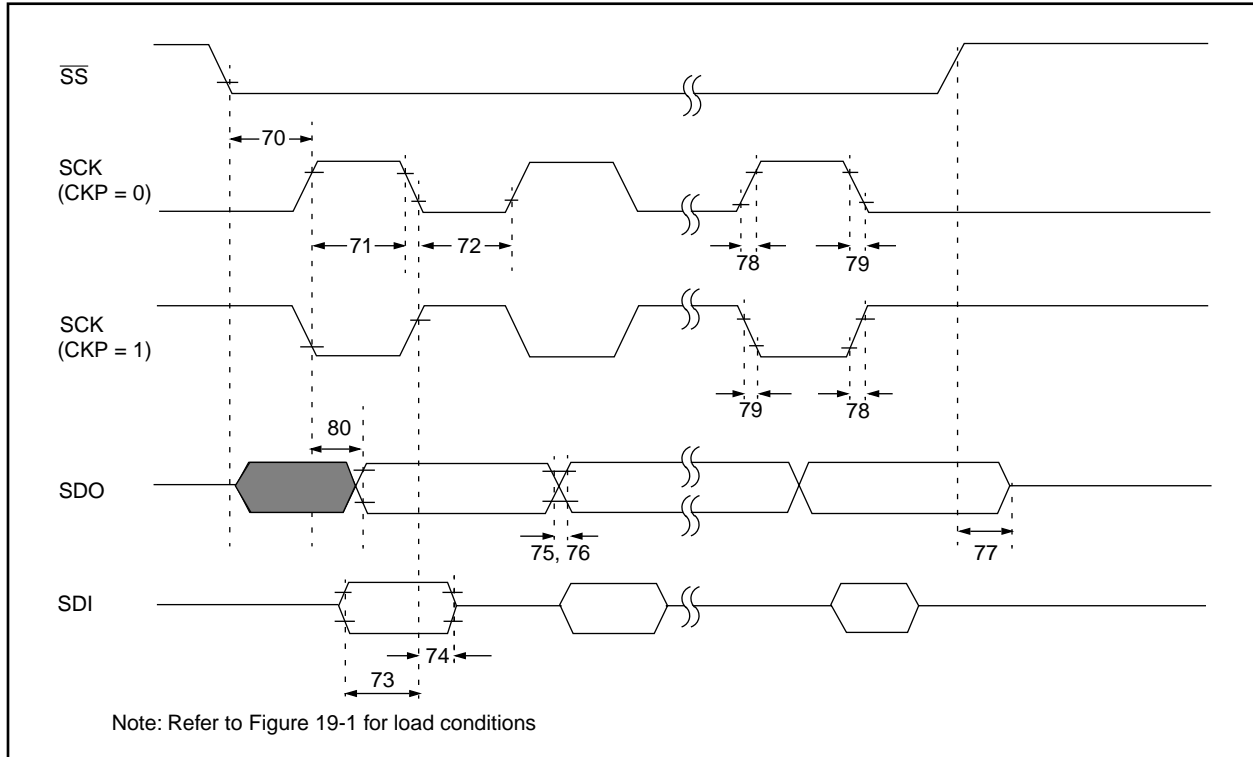
TABLE 19-7: PARALLEL SLAVE PORT REQUIREMENTS (PIC16C74A)

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
62	TdtV2wrH	Data in valid before WR↑ or CS↑ (setup time)	20 25	— —	— —	ns ns	Extended Range Only
63*	TwrH2dtI	WR↑ or CS↑ to data-in invalid (hold time)	20 35	— —	— —	ns ns	
64	TrdL2dtV	RD↓ and CS↓ to data-out valid	— —	— —	80 90	ns ns	Extended Range Only
65	TrdH2dtI	RD↑ or CS↓ to data-out invalid	10	—	30	ns	

\* These parameters are characterized but not tested.

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

**FIGURE 19-9: SPI MODE TIMING**



**TABLE 19-8: SPI MODE REQUIREMENTS**

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
70	TssL2scl, TssL2sclL	$\overline{SS}\downarrow$ to SCK $\downarrow$ or SCK $\uparrow$ input	T <sub>cy</sub>	—	—	ns	
71	Tsch	SCK input high time (slave mode)	T <sub>cy</sub> + 20	—	—	ns	
72	Tscl	SCK input low time (slave mode)	T <sub>cy</sub> + 20	—	—	ns	
73	TdiV2scl, TdiV2sclL	Setup time of SDI data input to SCK edge	100	—	—	ns	
74	Tsch2diL, TscL2diL	Hold time of SDI data input to SCK edge	100	—	—	ns	
75	TdoR	SDO data output rise time	—	10	25	ns	
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)	—	10	25	ns	
79	TscF	SCK output fall time (master mode)	—	10	25	ns	
80	Tsch2doV, TscL2doV	SDO data output valid after SCK edge	—	—	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.

# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

FIGURE 19-10: I<sup>2</sup>C BUS START/STOP BITS TIMING

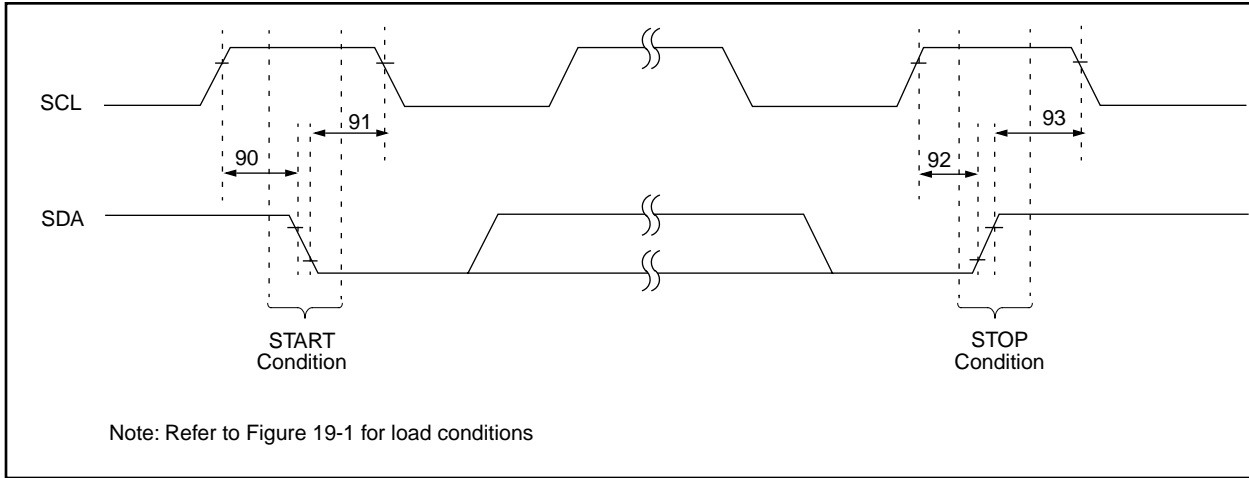


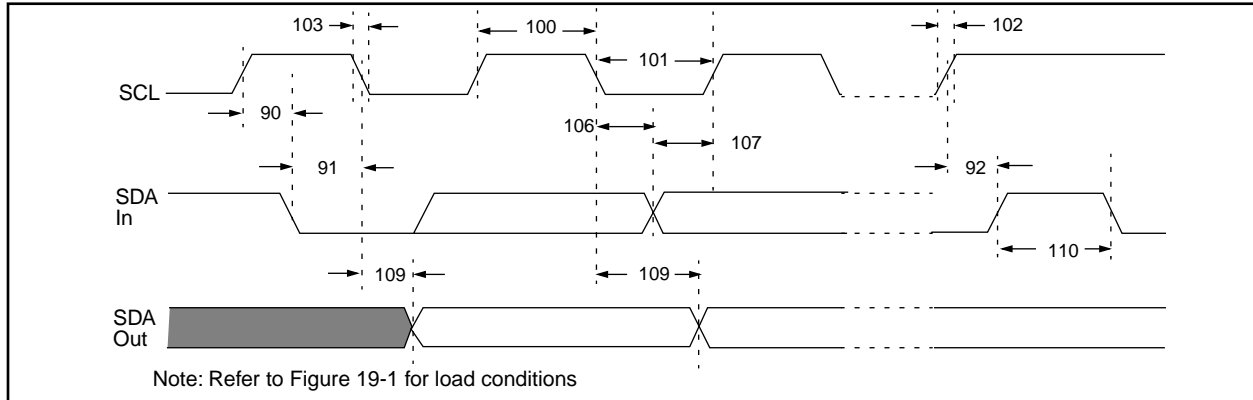
TABLE 19-9: I<sup>2</sup>C BUS START/STOP BITS REQUIREMENTS

Parameter No.	Sym	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	—	—			

# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

**FIGURE 19-11: I<sup>2</sup>C BUS DATA TIMING**



**TABLE 19-10: I<sup>2</sup>C BUS DATA REQUIREMENTS**

Parameter No.	Sym	Characteristic	Min	Max	Units	Conditions	
100	T <sub>HIGH</sub>	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	1.5T <sub>CY</sub>	—		
101	T <sub>LOW</sub>	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	1.5T <sub>CY</sub>	—		
102	T <sub>R</sub>	SDA and SCL rise time	100 kHz mode	—	1000	ns	
			400 kHz mode	20 + 0.1C <sub>b</sub>	300	ns	C <sub>b</sub> is specified to be from 10 to 400 pF
103	T <sub>F</sub>	SDA and SCL fall time	100 kHz mode	—	300	ns	
			400 kHz mode	20 + 0.1C <sub>b</sub>	300	ns	C <sub>b</sub> is specified to be from 10 to 400 pF
90	T <sub>SU:STA</sub>	START condition setup time	100 kHz mode	4.7	—	μs	Only relevant for repeated START condition
			400 kHz mode	0.6	—	μs	
91	T <sub>HD:STA</sub>	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	T <sub>HD:DAT</sub>	Data input hold time	100 kHz mode	0	—	ns	
			400 kHz mode	0	0.9	μs	
107	T <sub>SU:DAT</sub>	Data input setup time	100 kHz mode	250	—	ns	Note 2
			400 kHz mode	100	—	ns	
92	T <sub>SU:STO</sub>	STOP condition setup time	100 kHz mode	4.7	—	μs	
			400 kHz mode	0.6	—	μs	
109	T <sub>AA</sub>	Output valid from clock	100 kHz mode	—	3500	ns	Note 1
			400 kHz mode	—	—	ns	
110	T <sub>BUF</sub>	Bus free time	100 kHz mode	4.7	—	μs	Time the bus must be free before a new transmission can start
			400 kHz mode	1.3	—	μs	
	C <sub>b</sub>	Bus capacitive loading	—	400	pF		

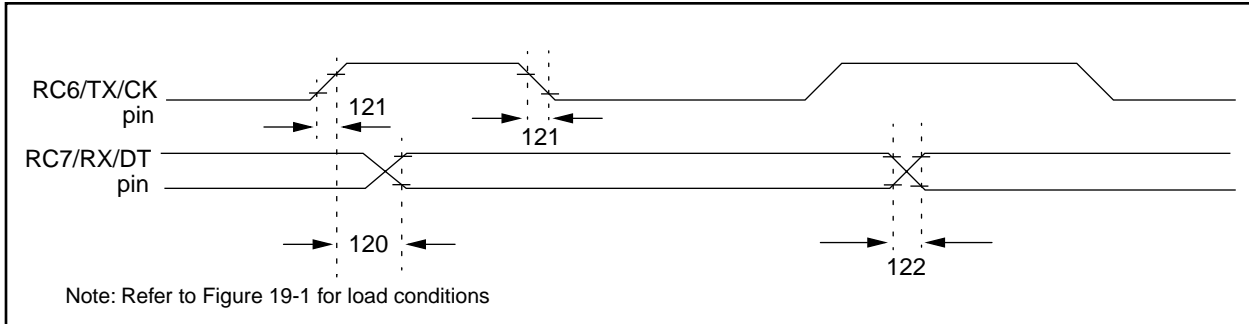
Note 1: As a transmitter, the device must provide this internal minimum delay time to bridge the undefined region (min. 300 ns) of the falling edge of SCL to avoid unintended generation of START or STOP conditions.

Note 2: A fast-mode (400 kHz) I<sup>2</sup>C-bus device can be used in a standard-mode (100 kHz) I<sup>2</sup>C-bus system, but the requirement t<sub>SU:DAT</sub> ≥ 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<sub>R</sub> max.+t<sub>SU:DAT</sub> = 1000 + 250 = 1250 ns (according to the standard-mode I<sup>2</sup>C bus specification) before the SCL line is released.

# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

**FIGURE 19-12: USART SYNCHRONOUS TRANSMISSION (MASTER/SLAVE) TIMING**

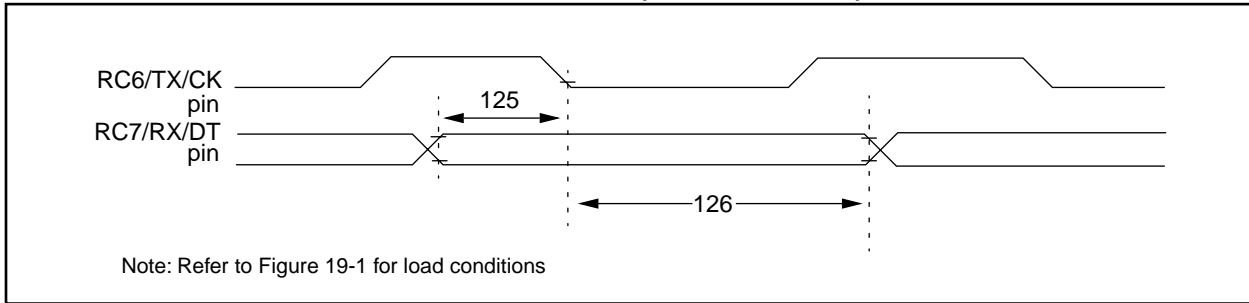


**TABLE 19-11: USART SYNCHRONOUS TRANSMISSION REQUIREMENTS**

Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
120	TckH2dtV	SYNC XMIT (MASTER & SLAVE)					
		Clock high to data out valid					
					80	ns	PIC16C73A/74A
					100	ns	PIC16LC73A/74A
121	Tckrf	Clock out rise time and fall time (Master Mode)			45	ns	PIC16C73A/74A
					50	ns	PIC16LC73A/74A
122	Tdtrf	Data out rise time and fall time			45	ns	PIC16C73A/74A
					50	ns	PIC16LC73A/74A

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

**FIGURE 19-13: USART SYNCHRONOUS RECEIVE (MASTER/SLAVE) TIMING**



**TABLE 19-12: USART SYNCHRONOUS RECEIVE REQUIREMENTS**

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
125	TdtV2ckL	SYNC RCV (MASTER & SLAVE) Data setup before CK ↓ (DT setup time)	15	—	—	ns	
126	TckL2dtL	Data hold after CK ↓ (DT hold time)	15	—	—	ns	

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



# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

**TABLE 19-13: A/D CONVERTER CHARACTERISTICS:**

**PIC16C73A/74A-04 (Commercial, Industrial, Extended)**  
**PIC16C73A/74A-10 (Commercial, Industrial, Extended)**  
**PIC16C73A/74A-20 (Commercial, Industrial, Extended)**  
**PIC16LC73A/74A-04 (Commercial, Industrial)**

Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions	
A01	NR	Resolution	—	—	8-bits	bit	$V_{REF} = V_{DD} = 5.12V$ , $V_{SS} \leq V_{AIN} \leq V_{REF}$	
A02	EABS	Total Absolute error	—	—	$< \pm 1$	LSb	$V_{REF} = V_{DD} = 5.12V$ , $V_{SS} \leq V_{AIN} \leq V_{REF}$	
A03	EIL	Integral linearity error	—	—	$< \pm 1$	LSb	$V_{REF} = V_{DD} = 5.12V$ , $V_{SS} \leq V_{AIN} \leq V_{REF}$	
A04	EDL	Differential linearity error	—	—	$< \pm 1$	LSb	$V_{REF} = V_{DD} = 5.12V$ , $V_{SS} \leq V_{AIN} \leq V_{REF}$	
A05	EFS	Full scale error	—	—	$< \pm 1$	LSb	$V_{REF} = V_{DD} = 5.12V$ , $V_{SS} \leq V_{AIN} \leq V_{REF}$	
A06	EOFF	Offset error	—	—	$< \pm 1$	LSb	$V_{REF} = V_{DD} = 5.12V$ , $V_{SS} \leq V_{AIN} \leq V_{REF}$	
A10	—	Monotonicity	—	guaranteed	—	—	$V_{SS} \leq V_{AIN} \leq V_{REF}$	
A20	VREF	Reference voltage	3.0V	—	$V_{DD} + 0.3$	V		
A25	VAIN	Analog input voltage	$V_{SS} - 0.3$	—	$V_{REF} + 0.3$	V		
A30	ZAIN	Recommended impedance of analog voltage source	—	—	10.0	k $\Omega$		
A40	IAD	A/D conversion current (VDD)	PIC16C73A/74A	—	180	—	$\mu A$	Average current consumption when A/D is on. (Note 1)
			PIC16LC73A/74A	—	90	—	$\mu A$	
A50	IREF	VREF input current (Note 2)	10	—	1000	$\mu A$	During VAIN acquisition. Based on differential of VHOLD to VAIN to charge CHOLD, see Section 13.1.	
			—	—	10	$\mu A$	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 RA3 pin or VDD pin, whichever is selected as reference input.

# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

FIGURE 19-14: A/D CONVERSION TIMING

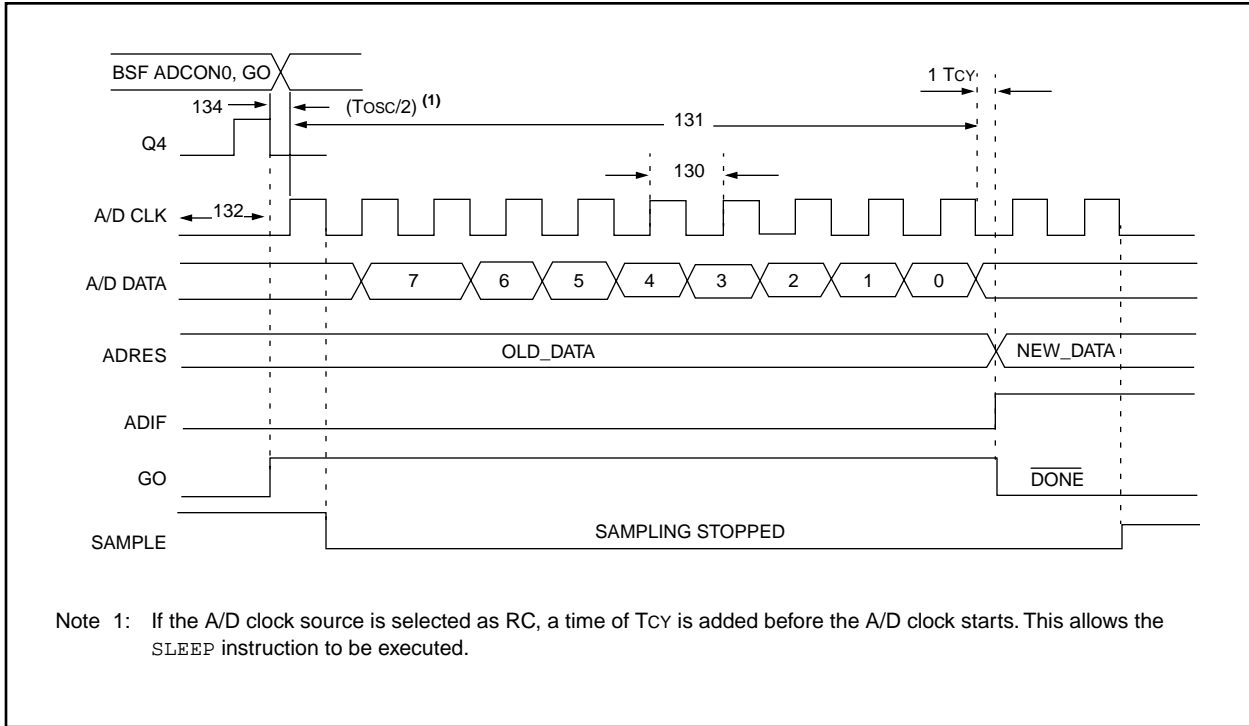


TABLE 19-14: A/D CONVERSION REQUIREMENTS

Param No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions
130	TAD	A/D clock period	PIC16C73A/74A	1.6	—	—	μs	TOSC based, VREF ≥ 3.0V
			PIC16LC73A/74A	2.0	—	—	μs	TOSC based, VREF full range
			PIC16C73A/74A	2.0	4.0	6.0	μs	A/D RC Mode
			PIC16LC73A/74A	3.0	6.0	9.0	μs	A/D RC Mode
131	TcNV	Conversion time (not including S/H time) (Note 1)	—	9.5	—	TAD		
132	TACQ	Acquisition time	Note 2	20	—	—	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).
			5*	—	—	μs		
134	Tgo	Q4 to A/D clock start	—	Tosc/2 §	—	—	If the A/D clock source is selected as RC, a time of Tcy is added before the A/D clock starts. This allows the SLEEP instruction to be executed.	
135	Tswc	Switching from convert → sample time	1.5 §	—	—	—	TAD	

\* These parameters are characterized but not tested.

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

§ This specification ensured by design.

Note 1: ADRES register may be read on the following Tcy cycle.

Note 2: See Section 13.1 for min conditions.

# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

## 20.0 ELECTRICAL CHARACTERISTICS FOR PIC16C76/77

### 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 Vss (except VDD, $\overline{\text{MCLR}}$ , and RA4).....	-0.3V to (VDD + 0.3V)
Voltage on VDD with respect to Vss .....	-0.3 to +7.5V
Voltage on $\overline{\text{MCLR}}$ with respect to Vss (Note 2) .....	0 to +14V
Voltage on RA4 with respect to Vss .....	0 to +14V
Total power dissipation (Note 1).....	1.0W
Maximum current out of Vss pin .....	300 mA
Maximum current into VDD pin .....	250 mA
Input clamp current, I <sub>IK</sub> (V <sub>I</sub> < 0 or V <sub>I</sub> > VDD) .....	±20 mA
Output clamp current, I <sub>OK</sub> (V <sub>O</sub> < 0 or V <sub>O</sub> > VDD).....	±20 mA
Maximum output current sunk by any I/O pin.....	25 mA
Maximum output current sourced by any I/O pin .....	25 mA
Maximum current sunk by PORTA, 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:  $P_{dis} = V_{DD} \times \{I_{DD} - \sum I_{OH}\} + \sum \{(V_{DD} - V_{OH}) \times I_{OH}\} + \sum (V_{OL} \times I_{OL})$

**Note 2:** Voltage spikes below Vss at the  $\overline{\text{MCLR}}$  pin, inducing currents greater than 80 mA, may cause latch-up. Thus, a series resistor of 50-100Ω should be used when applying a “low” level to the  $\overline{\text{MCLR}}$  pin rather than pulling this pin directly to Vss.

**Note 3:** PORTD and PORTE are not implemented on the PIC16C76.

† 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.

# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

**TABLE 20-1: CROSS REFERENCE OF DEVICE SPECS FOR OSCILLATOR CONFIGURATIONS AND FREQUENCIES OF OPERATION (COMMERCIAL DEVICES)**

OSC	PIC16C76-04 PIC16C77-04	PIC16C76-10 PIC16C77-10	PIC16C76-20 PIC16C77-20	PIC16LC76-04 PIC16LC77-04	JW Devices
RC	VDD: 4.0V to 6.0V IDD: 5 mA max. at 5.5V IPD: 16 $\mu$ A max. at 4V Freq: 4 MHz max.	VDD: 4.5V to 5.5V IDD: 2.7 mA typ. at 5.5V IPD: 1.5 $\mu$ A typ. at 4V Freq: 4 MHz max.	VDD: 4.5V to 5.5V IDD: 2.7 mA typ. at 5.5V IPD: 1.5 $\mu$ A typ. at 4V Freq: 4 MHz max.	VDD: 2.5V to 6.0V IDD: 3.8 mA max. at 3.0V IPD: 5 $\mu$ A max. at 3V Freq: 4 MHz max.	VDD: 4.0V to 6.0V IDD: 5 mA max. at 5.5V IPD: 16 $\mu$ A max. at 4V Freq: 4 MHz max.
XT	VDD: 4.0V to 6.0V IDD: 5 mA max. at 5.5V IPD: 16 $\mu$ A max. at 4V Freq: 4 MHz max.	VDD: 4.5V to 5.5V IDD: 2.7 mA typ. at 5.5V IPD: 1.5 $\mu$ A typ. at 4V Freq: 4 MHz max.	VDD: 4.5V to 5.5V IDD: 2.7 mA typ. at 5.5V IPD: 1.5 $\mu$ A typ. at 4V Freq: 4 MHz max.	VDD: 2.5V to 6.0V IDD: 3.8 mA max. at 3.0V IPD: 5 $\mu$ A max. at 3V Freq: 4 MHz max.	VDD: 4.0V to 6.0V IDD: 5 mA max. at 5.5V IPD: 16 $\mu$ A max. at 4V Freq: 4 MHz max.
HS	VDD: 4.5V to 5.5V IDD: 13.5 mA typ. at 5.5V IPD: 1.5 $\mu$ A typ. at 4.5V Freq: 4 MHz max.	VDD: 4.5V to 5.5V IDD: 10 mA max. at 5.5V IPD: 1.5 $\mu$ A typ. at 4.5V Freq: 10 MHz max.	VDD: 4.5V to 5.5V IDD: 20 mA max. at 5.5V IPD: 1.5 $\mu$ A typ. at 4.5V Freq: 20 MHz max.	Not recommended for use in HS mode	VDD: 4.5V to 5.5V IDD: 20 mA max. at 5.5V IPD: 1.5 $\mu$ A typ. at 4.5V Freq: 20 MHz max.
LP	VDD: 4.0V to 6.0V IDD: 52.5 $\mu$ A typ. at 32 kHz, 4.0V IPD: 0.9 $\mu$ A typ. at 4.0V Freq: 200 kHz max.	Not recommended for use in LP mode	Not recommended for use in LP mode	VDD: 2.5V to 6.0V IDD: 48 $\mu$ A max. at 32 kHz, 3.0V IPD: 5.0 $\mu$ A max. at 3.0V Freq: 200 kHz max.	VDD: 2.5V to 6.0V IDD: 48 $\mu$ A max. at 32 kHz, 3.0V IPD: 5.0 $\mu$ A max. at 3.0V Freq: 200 kHz max.

The shaded sections indicate oscillator selections which are tested for functionality, but not for MIN/MAX specifications. It is recommended that the user select the device type that ensures the specifications required.

# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

**20.1 DC Characteristics: PIC16C76/77-04 (Commercial, Industrial, Extended)  
PIC16C76/77-10 (Commercial, Industrial, Extended)  
PIC16C76/77-20 (Commercial, Industrial, Extended)**

Standard Operating Conditions (unless otherwise stated)							
DC CHARACTERISTICS							
		Operating temperature -40°C ≤ TA ≤ +125°C for extended, -40°C ≤ TA ≤ +85°C for industrial and 0°C ≤ TA ≤ +70°C for commercial					
Param No.	Characteristic	Sym	Min	Typ†	Max	Units	Conditions
D001 D001A	Supply Voltage	VDD	4.0 4.5	- -	6.0 5.5	V V	XT, RC and LP osc configuration HS osc configuration
D002*	RAM Data Retention Voltage (Note 1)	VDR	-	1.5	-	V	
D003	VDD start voltage to ensure internal Power-on Reset signal	VPOR	-	VSS	-	V	See section on Power-on Reset for details
D004*	VDD rise rate to ensure internal Power-on Reset signal	SVDD	0.05	-	-	V/ms	See section on Power-on Reset for details
D005	Brown-out Reset Voltage	BVDD	3.7 3.7	4.0 4.0	4.3 4.4	V V	BODEN bit in configuration word enabled Extended Range Only
D010 D013	Supply Current (Note 2,5)	IDD	- -	2.7 10	5 20	mA mA	XT, RC osc configuration FOSC = 4 MHz, VDD = 5.5V (Note 4) HS osc configuration FOSC = 20 MHz, VDD = 5.5V
D015*	Brown-out Reset Current (Note 6)	ΔIBOR	-	350	425	μA	BOR enabled VDD = 5.0V
D020 D021 D021A D021B	Power-down Current (Note 3,5)	IPD	- - - -	10.5 1.5 1.5 2.5	42 16 19 19	μA μA μA μA	VDD = 4.0V, WDT enabled, -40°C to +85°C VDD = 4.0V, WDT disabled, -0°C to +70°C VDD = 4.0V, WDT disabled, -40°C to +85°C VDD = 4.0V, WDT disabled, -40°C to +125°C
D023*	Brown-out Reset Current (Note 6)	ΔIBOR	-	350	425	μA	BOR enabled VDD = 5.0V

\* 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 tristated, 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.

# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

## 20.2 DC Characteristics: PIC16LC76/77-04 (Commercial, 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 and $0^{\circ}\text{C} \leq T_A \leq +70^{\circ}\text{C}$ for commercial					
Param No.	Characteristic	Sym	Min	Typ†	Max	Units	Conditions
D001	Supply Voltage	VDD	2.5	-	6.0	V	LP, XT, RC osc configuration (DC - 4 MHz)
D002*	RAM Data Retention Voltage (Note 1)	VDR	-	1.5	-	V	
D003	VDD start voltage to ensure internal Power-on Reset signal	VPOR	-	VSS	-	V	See section on Power-on Reset for details
D004*	VDD rise rate to ensure internal Power-on Reset signal	SVDD	0.05	-	-	V/ms	See section on Power-on Reset for details
D005	Brown-out Reset Voltage	BVDD	3.7	4.0	4.3	V	BODEN bit in configuration word enabled
D010	Supply Current (Note 2,5)	IDD	-	2.0	3.8	mA	XT, RC osc configuration FOSC = 4 MHz, VDD = 3.0V (Note 4)
D010A			-	22.5	48	$\mu\text{A}$	LP osc configuration FOSC = 32 kHz, VDD = 3.0V, WDT disabled
D015*	Brown-out Reset Current (Note 6)	$\Delta\text{IBOR}$	-	350	425	$\mu\text{A}$	BOR enabled VDD = 5.0V
D020	Power-down Current (Note 3,5)	IPD	-	7.5	30	$\mu\text{A}$	VDD = 3.0V, WDT enabled, $-40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$
D021			-	0.9	5	$\mu\text{A}$	VDD = 3.0V, WDT disabled, $0^{\circ}\text{C}$ to $+70^{\circ}\text{C}$
D021A			-	0.9	5	$\mu\text{A}$	VDD = 3.0V, WDT disabled, $-40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$
D023*	Brown-out Reset Current (Note 6)	$\Delta\text{IBOR}$	-	350	425	$\mu\text{A}$	BOR enabled VDD = 5.0V

\* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V,  $25^{\circ}\text{C}$  unless otherwise stated. These parameters are for design guidance only and are not tested.

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

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 tristated, 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  $20\mu\text{A}$  to the specification. This value is from characterization and is for design guidance only. This is not tested.

6: The  $\Delta$  current is the additional current consumed when this peripheral is enabled. This current should be added to the base IDD or IPD measurement.

# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

**20.3 DC Characteristics:** PIC16C76/77-04 (Commercial, Industrial, Extended)  
 PIC16C76/77-10 (Commercial, Industrial, Extended)  
 PIC16C76/77-20 (Commercial, Industrial, Extended)  
 PIC16LC76/77-04 (Commercial, Industrial)

Standard Operating Conditions (unless otherwise stated)							
DC CHARACTERISTICS							
Operating temperature -40°C ≤ TA ≤ +125°C for extended, -40°C ≤ TA ≤ +85°C for industrial and 0°C ≤ TA ≤ +70°C for commercial							
Operating voltage VDD range as described in DC spec Section 20.1 and Section 20.2.							
Param No.	Characteristic	Sym	Min	Typ †	Max	Units	Conditions
D030 D030A D031 D032 D033	<b>Input Low Voltage</b> I/O ports with TTL buffer with Schmitt Trigger buffer MCLR, OSC1 (in RC mode) OSC1 (in XT, HS and LP)	VIL	VSS	-	0.15VDD 0.8V 0.2VDD 0.2VDD 0.3VDD	V	For entire VDD range 4.5V ≤ VDD ≤ 5.5V Note1
D040 D040A D041 D042 D042A D043 D070	<b>Input High Voltage</b> I/O ports with TTL buffer with Schmitt Trigger buffer MCLR OSC1 (XT, HS and LP) OSC1 (in RC mode) PORTB weak pull-up current	VIH	2.0 0.25VDD + 0.8V 0.8VDD 0.8VDD 0.7VDD 0.9VDD 50	- - - - - - - 250	VDD VDD VDD VDD VDD VDD VDD 400	V	4.5V ≤ VDD ≤ 5.5V For entire VDD range For entire VDD range Note1 VDD = 5V, VPIN = VSS
D060 D061 D063	<b>Input Leakage Current</b> (Notes 2, 3) I/O ports MCLR, RA4/T0CKI OSC1	IIL	-	-	±1 ±5 ±5	µA	VSS ≤ VPIN ≤ VDD, Pin at hi-impedance VSS ≤ VPIN ≤ VDD VSS ≤ VPIN ≤ VDD, XT, HS and LP osc configuration
D080 D080A D083 D083A	<b>Output Low Voltage</b> I/O ports OSC2/CLKOUT (RC osc config)	VOL	-	-	0.6 0.6 0.6 0.6	V	IOL = 8.5 mA, VDD = 4.5V, -40°C to +85°C IOL = 7.0 mA, VDD = 4.5V, -40°C to +125°C IOL = 1.6 mA, VDD = 4.5V, -40°C to +85°C IOL = 1.2 mA, VDD = 4.5V, -40°C to +125°C

\* These parameters are characterized but not tested.

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

- Note 1: In RC oscillator configuration, the OSC1/CLKIN pin is a Schmitt Trigger input. It is not recommended that the PIC16C7X 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.

# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

Standard Operating Conditions (unless otherwise stated)							
Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ for extended, $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial and $0^{\circ}\text{C} \leq T_A \leq +70^{\circ}\text{C}$ for commercial Operating voltage $V_{DD}$ range as described in DC spec Section 20.1 and Section 20.2.							
Param No.	Characteristic	Sym	Min	Typ †	Max	Units	Conditions
D090	<b>Output High Voltage</b> I/O ports (Note 3)	$V_{OH}$	$V_{DD} - 0.7$	-	-	V	$I_{OH} = -3.0\text{ mA}$ , $V_{DD} = 4.5\text{V}$ , $-40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$
D090A			$V_{DD} - 0.7$	-	-	V	$I_{OH} = -2.5\text{ mA}$ , $V_{DD} = 4.5\text{V}$ , $-40^{\circ}\text{C}$ to $+125^{\circ}\text{C}$
D092	OSC2/CLKOUT (RC osc config)		$V_{DD} - 0.7$	-	-	V	$I_{OH} = -1.3\text{ mA}$ , $V_{DD} = 4.5\text{V}$ , $-40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$
D092A			$V_{DD} - 0.7$	-	-	V	$I_{OH} = -1.0\text{ mA}$ , $V_{DD} = 4.5\text{V}$ , $-40^{\circ}\text{C}$ to $+125^{\circ}\text{C}$
D150*	<b>Open-Drain High Voltage</b>	$V_{OD}$	-	-	14	V	RA4 pin
<b>Capacitive Loading Specs on Output Pins</b>							
D100	OSC2 pin	$C_{OSC2}$	-	-	15	pF	In XT, HS and LP modes when external clock is used to drive OSC1.
D101	All I/O pins and OSC2 (in RC mode) SCL, SDA in I <sup>2</sup> C mode	$C_{IO}$	-	-	50	pF	
D102		$C_B$	-	-	400	pF	

\* 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 PIC16C7X 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.



# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

## 20.4 Timing Parameter Symbology

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

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

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

Lowercase letters (pp) and their meanings:

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

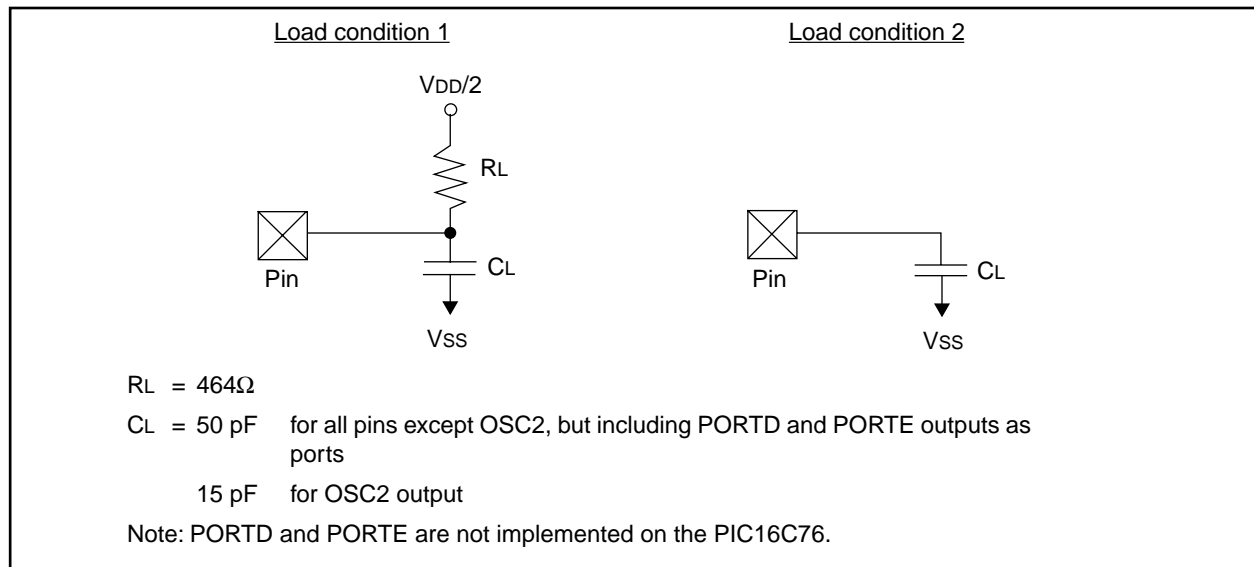
Uppercase letters and their meanings:

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

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

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

**FIGURE 20-1: LOAD CONDITIONS**



# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

## 20.5 Timing Diagrams and Specifications

FIGURE 20-2: EXTERNAL CLOCK TIMING

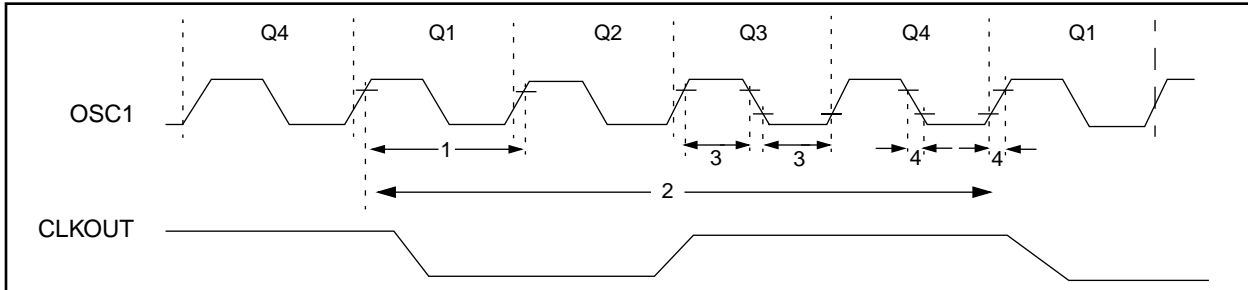


TABLE 20-2: EXTERNAL CLOCK TIMING REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions	
	Fosc	<b>External CLKIN Frequency (Note 1)</b>	DC	—	4	MHz	XT and RC osc mode	
			DC	—	4	MHz	HS osc mode (-04)	
			DC	—	10	MHz	HS osc mode (-10)	
			DC	—	20	MHz	HS osc mode (-20)	
			DC	—	200	kHz	LP osc mode	
			<b>Oscillator Frequency (Note 1)</b>	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	<b>External CLKIN Period (Note 1)</b>	250	—	—	ns	XT and RC osc mode	
			250	—	—	ns	HS osc mode (-04)	
			100	—	—	ns	HS osc mode (-10)	
			50	—	—	ns	HS osc mode (-20)	
			5	—	—	μs	LP osc mode	
			<b>Oscillator Period (Note 1)</b>	250	—	—	ns	RC osc mode
				250	—	10,000	ns	XT osc mode
				250	—	250	ns	HS osc mode (-04)
				100	—	250	ns	HS osc mode (-10)
				50	—	250	ns	HS osc mode (-20)
2	Tcy	<b>Instruction Cycle Time (Note 1)</b>	200	Tcy	DC	ns	Tcy = 4/Fosc	
3	TosL, TosH	<b>External Clock in (OSC1) High or Low Time</b>	100	—	—	ns	XT oscillator	
			2.5	—	—	μs	LP oscillator	
			15	—	—	ns	HS oscillator	
4	TosR, TosF	<b>External Clock in (OSC1) Rise or Fall Time</b>	—	—	25	ns	XT oscillator	
			—	—	50	ns	LP oscillator	
			—	—	15	ns	HS oscillator	

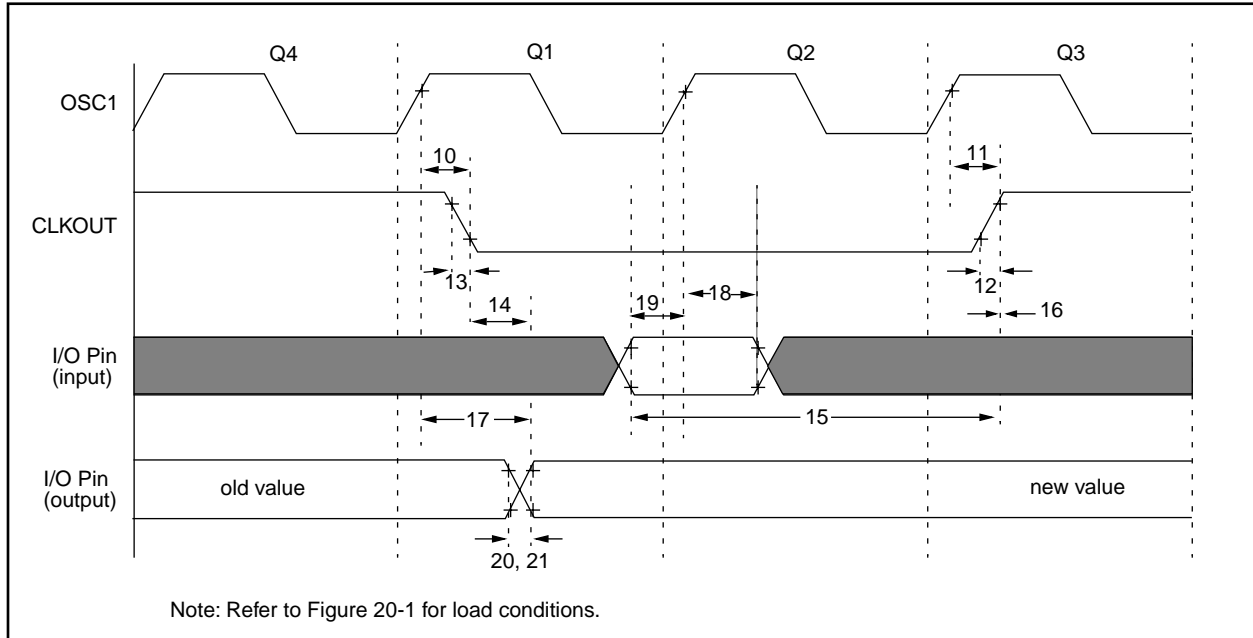
† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

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.

# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

**FIGURE 20-3: CLKOUT AND I/O TIMING**



**TABLE 20-3: CLKOUT AND I/O TIMING REQUIREMENTS**

Param No.	Sym	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	—	—	0.5TCY + 20	ns	Note 1	
15*	TioV2ckH	Port in valid before CLKOUT ↑	Tosc + 200	—	—	ns	Note 1	
16*	TckH2iol	Port in hold after CLKOUT ↑	0	—	—	ns	Note 1	
17*	TosH2ioV	OSC1↑ (Q1 cycle) to Port out valid	—	50	150	ns		
18*	TosH2iol	OSC1↑ (Q2 cycle) to Port input invalid (I/O in hold time)	PIC16C76/77	100	—	—	ns	
			PIC16LC76/77	200	—	—	ns	
19*	TioV2osH	Port input valid to OSC1↑ (I/O in setup time)	0	—	—	ns		
20*	TioR	Port output rise time	PIC16C76/77	—	10	40	ns	
			PIC16LC76/77	—	—	80	ns	
21*	TioF	Port output fall time	PIC16C76/77	—	10	40	ns	
			PIC16LC76/77	—	—	80	ns	
22††*	Tinp	INT pin high or low time	TCY	—	—	ns		
23††*	Trbp	RB7:RB4 change INT high or low time	TCY	—	—	ns		

\* These parameters are characterized but not tested.

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

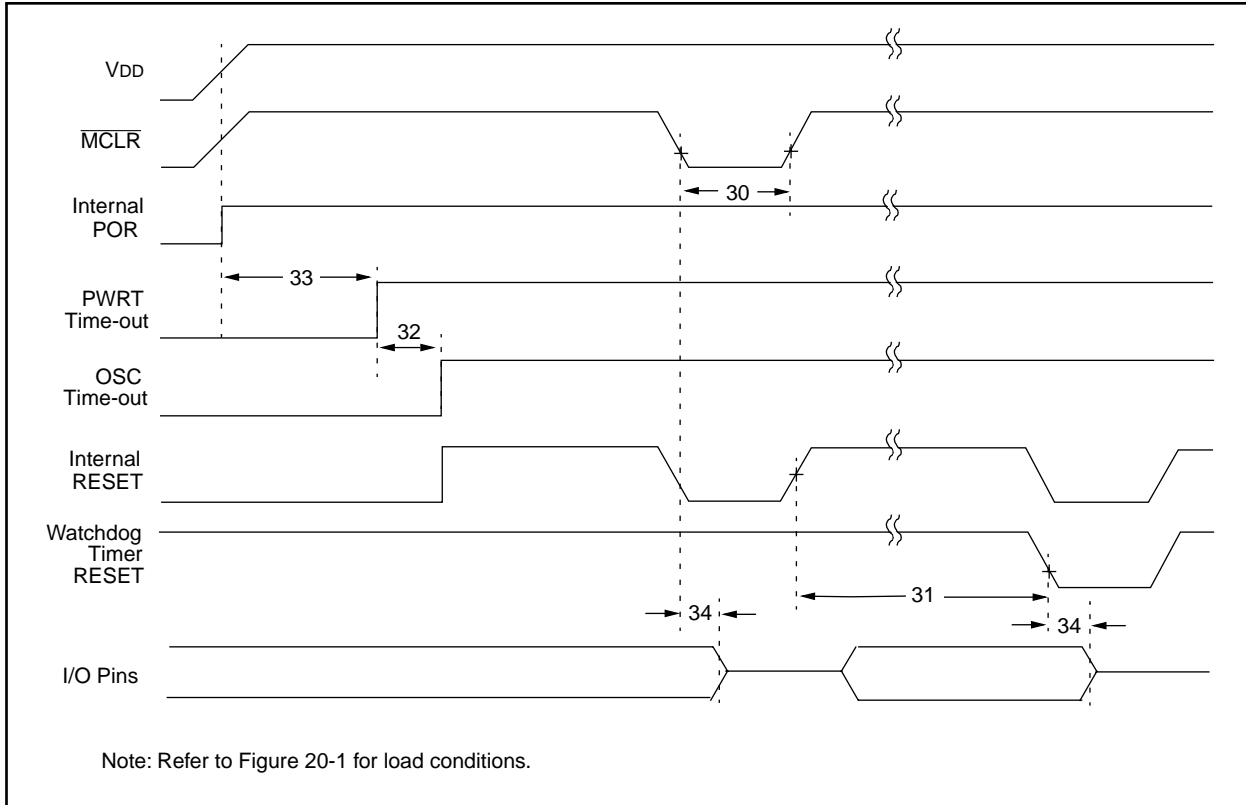
†† These parameters are asynchronous events not related to any internal clock edges.

Note 1: Measurements are taken in RC Mode where CLKOUT output is 4 x Tosc.

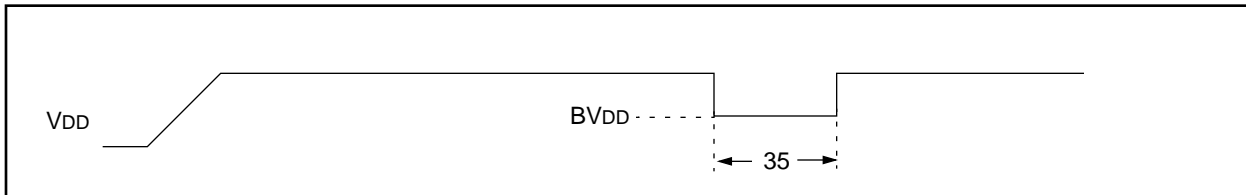
# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

**FIGURE 20-4: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING**



**FIGURE 20-5: BROWN-OUT RESET TIMING**



**TABLE 20-4: 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 +125°C
31*	Twdt	Watchdog Timer Time-out Period (No Prescaler)	7	18	33	ms	VDD = 5V, -40°C to +125°C
32	Tost	Oscillation Start-up Timer Period	—	1024TOSC	—	—	TOSC = OSC1 period
33*	Tpwrt	Power up Timer Period	28	72	132	ms	VDD = 5V, -40°C to +125°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 ≤ BVDD (D005)

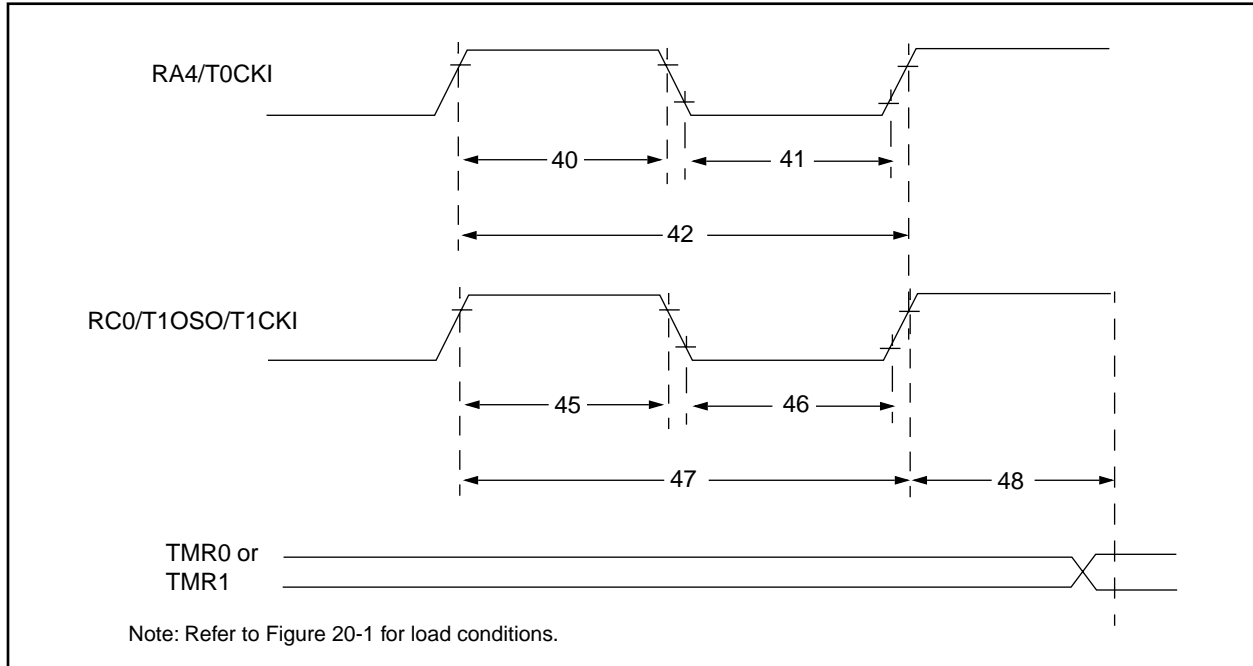
\* These parameters are characterized but not tested.

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

# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

**FIGURE 20-6: TIMER0 AND TIMER1 EXTERNAL CLOCK TIMINGS**



**TABLE 20-5: TIMER0 AND TIMER1 EXTERNAL CLOCK REQUIREMENTS**

Param No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions	
40*	Tt0H	T0CKI High Pulse Width	No Prescaler	$0.5T_{CY} + 20$	—	—	ns	Must also meet parameter 42	
			With Prescaler	10	—	—	ns		
41*	Tt0L	T0CKI Low Pulse Width	No Prescaler	$0.5T_{CY} + 20$	—	—	ns	Must also meet parameter 42	
			With Prescaler	10	—	—	ns		
42*	Tt0P	T0CKI Period	No Prescaler	$T_{CY} + 40$	—	—	ns	N = prescale value (2, 4, ..., 256)	
			With Prescaler	Greater of: $20$ or $\frac{T_{CY} + 40}{N}$	—	—	ns		
45*	Tt1H	T1CKI High Time	Synchronous, Prescaler = 1	$0.5T_{CY} + 20$	—	—	ns	Must also meet parameter 47	
			Synchronous, Prescaler = 2,4,8	PIC16C7X	15	—	—		ns
				PIC16LC7X	25	—	—		ns
			Asynchronous	PIC16C7X	30	—	—		ns
			PIC16LC7X	50	—	—	ns		
46*	Tt1L	T1CKI Low Time	Synchronous, Prescaler = 1	$0.5T_{CY} + 20$	—	—	ns	Must also meet parameter 47	
			Synchronous, Prescaler = 2,4,8	PIC16C7X	15	—	—		ns
				PIC16LC7X	25	—	—		ns
			Asynchronous	PIC16C7X	30	—	—		ns
			PIC16LC7X	50	—	—	ns		
47*	Tt1P	T1CKI input period	Synchronous	PIC16C7X	Greater of: $30$ or $\frac{T_{CY} + 40}{N}$	—	—	ns	N = prescale value (1, 2, 4, 8)
				PIC16LC7X	Greater of: $50$ or $\frac{T_{CY} + 40}{N}$				N = prescale value (1, 2, 4, 8)
			Asynchronous	PIC16C7X	60	—	—	ns	
				PIC16LC7X	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		$2T_{osc}$	—	$7T_{osc}$	—		

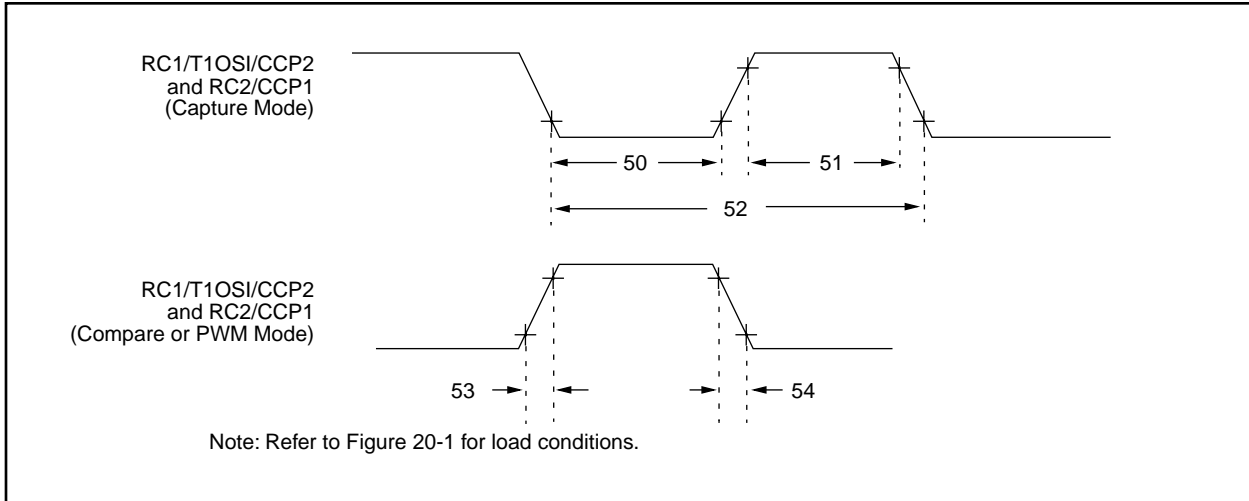
\* These parameters are characterized but not tested.

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

# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

**FIGURE 20-7: CAPTURE/COMPARE/PWM TIMINGS (CCP1 AND CCP2)**



**TABLE 20-6: CAPTURE/COMPARE/PWM REQUIREMENTS (CCP1 AND CCP2)**

Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions					
50*	TccL	CCP1 and CCP2 input low time	No Prescaler	$0.5T_{CY} + 20$	—	—	ns					
			With Prescaler	PIC16C76/77	10	—	—	ns				
							PIC16LC76/77	20	—	—	ns	
51*	TccH	CCP1 and CCP2 input high time	No Prescaler	$0.5T_{CY} + 20$	—	—	ns					
			With Prescaler	PIC16C76/77	10	—	—	ns				
							PIC16LC76/77	20	—	—	ns	
52*	TccP	CCP1 and CCP2 input period	$\frac{3T_{CY} + 40}{N}$	—	—	ns	N = prescale value (1, 4 or 16)					
53*	TccR	CCP1 and CCP2 output rise time	PIC16C76/77	—	10	25	ns					
			PIC16LC76/77	—	25	45	ns					
54*	TccF	CCP1 and CCP2 output fall time	PIC16C76/77	—	10	25	ns					
			PIC16LC76/77	—	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.

# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

FIGURE 20-8: PARALLEL SLAVE PORT TIMING (PIC16C77)

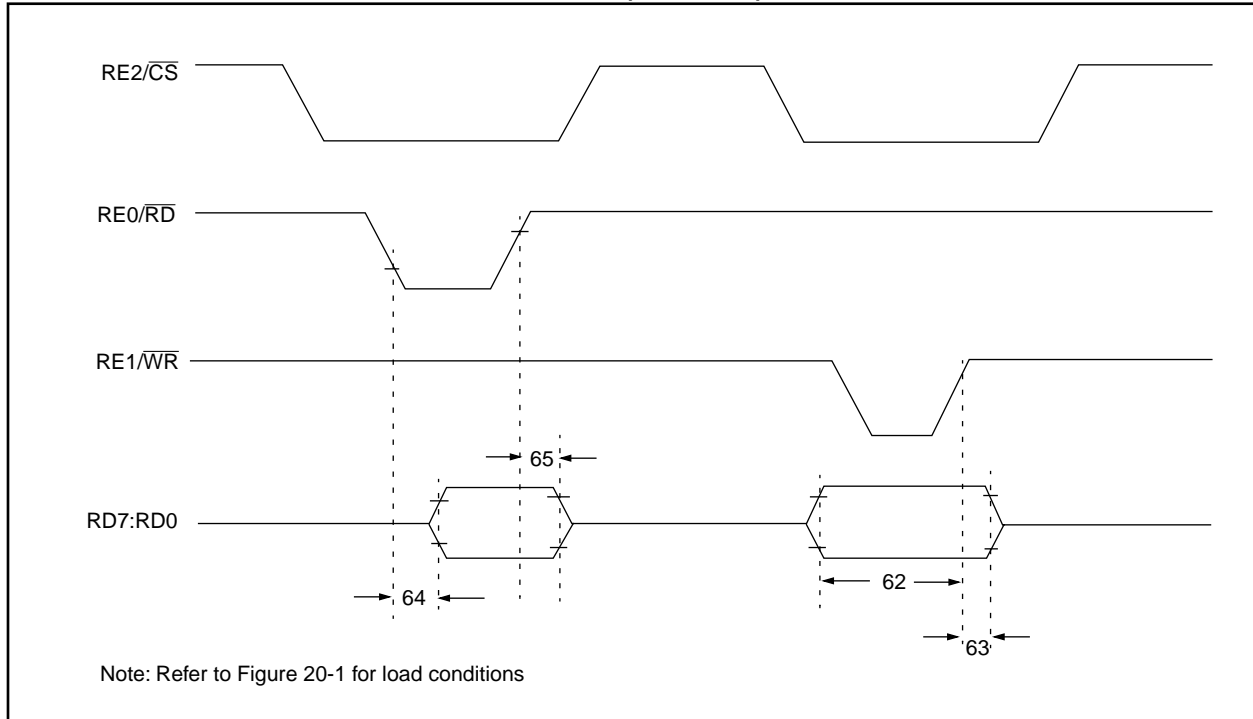


TABLE 20-7: PARALLEL SLAVE PORT REQUIREMENTS (PIC16C77)

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions	
62	TdtV2wrH	Data in valid before $\overline{WR}\uparrow$ or $\overline{CS}\uparrow$ (setup time)	20	—	—	ns	Extended Range Only	
			25	—	—	ns		
63*	TwrH2dtI	$\overline{WR}\uparrow$ or $\overline{CS}\uparrow$ to data-in invalid (hold time)	PIC16C77	20	—	—	ns	
			PIC16LC77	35	—	—	ns	
64	TrdL2dtV	$\overline{RD}\downarrow$ and $\overline{CS}\downarrow$ to data-out valid	—	—	80	ns	Extended Range Only	
			—	—	90	ns		
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.

# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

FIGURE 20-9: SPI MASTER MODE TIMING (CKE = 0)

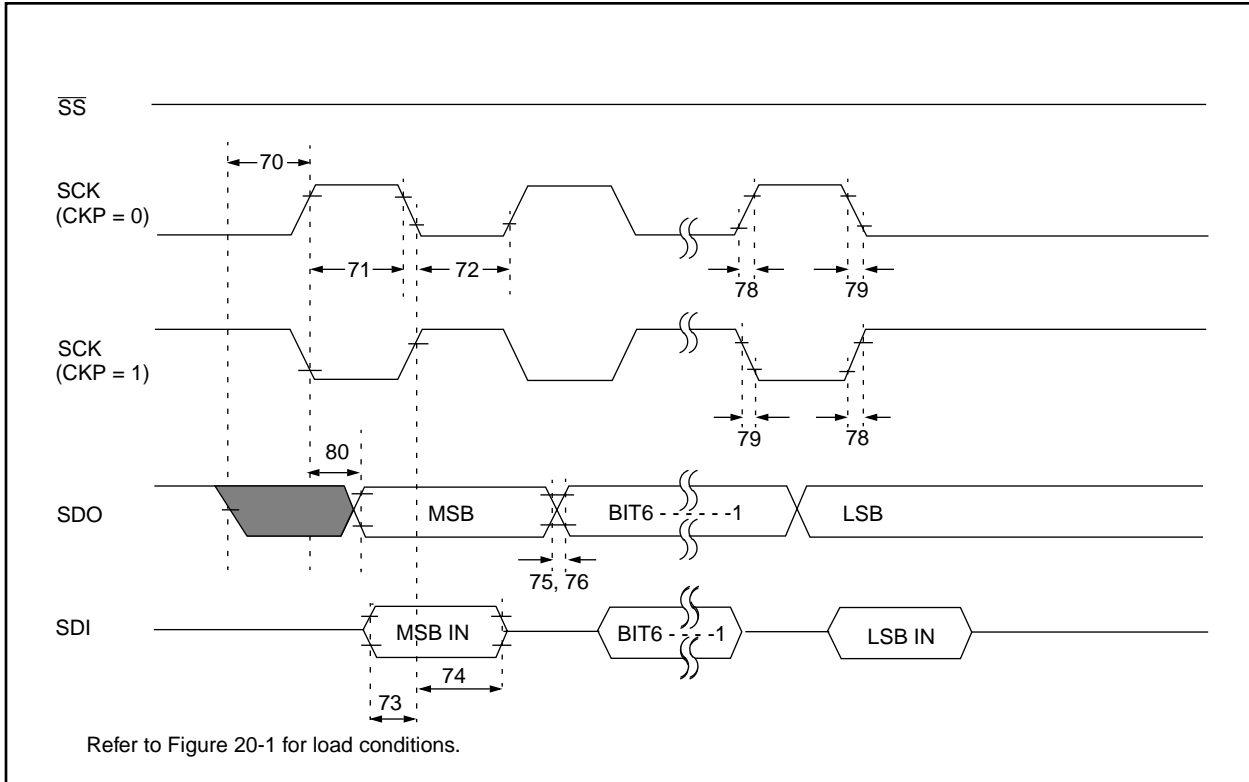
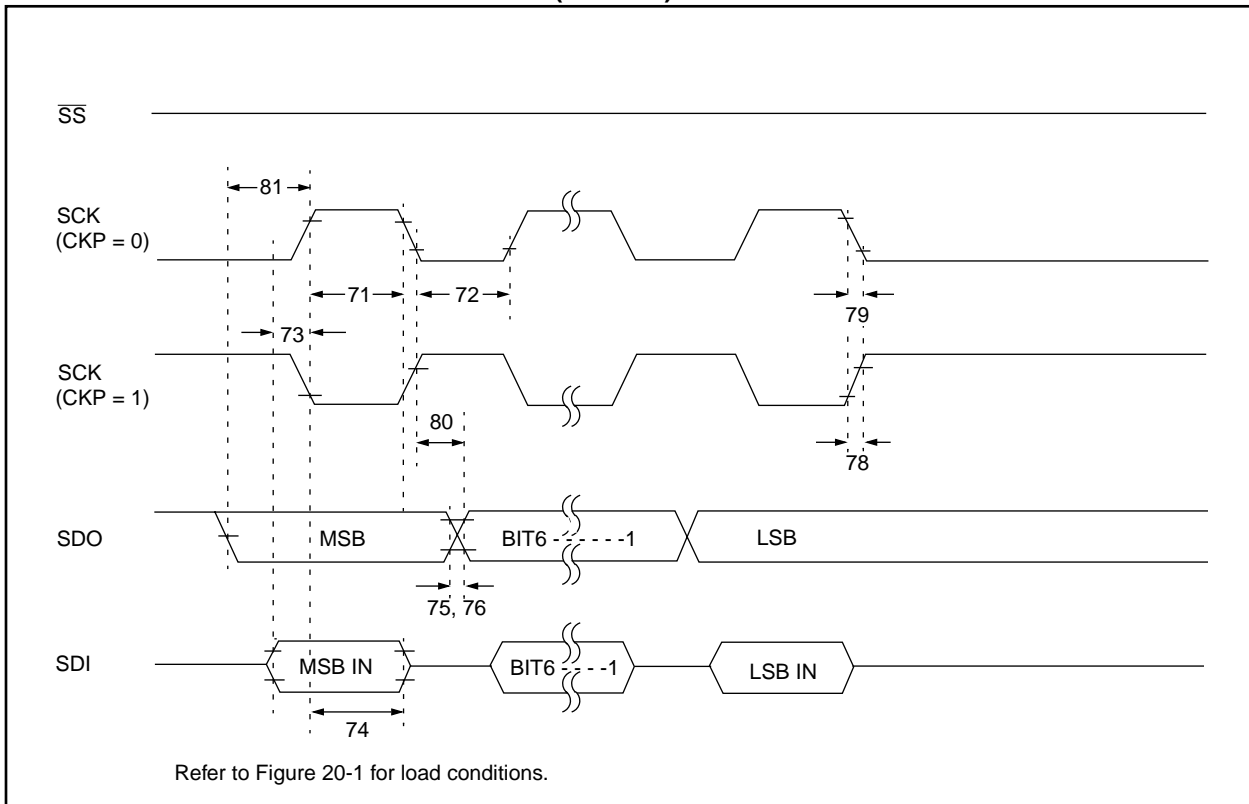


FIGURE 20-10: SPI MASTER MODE TIMING (CKE = 1)

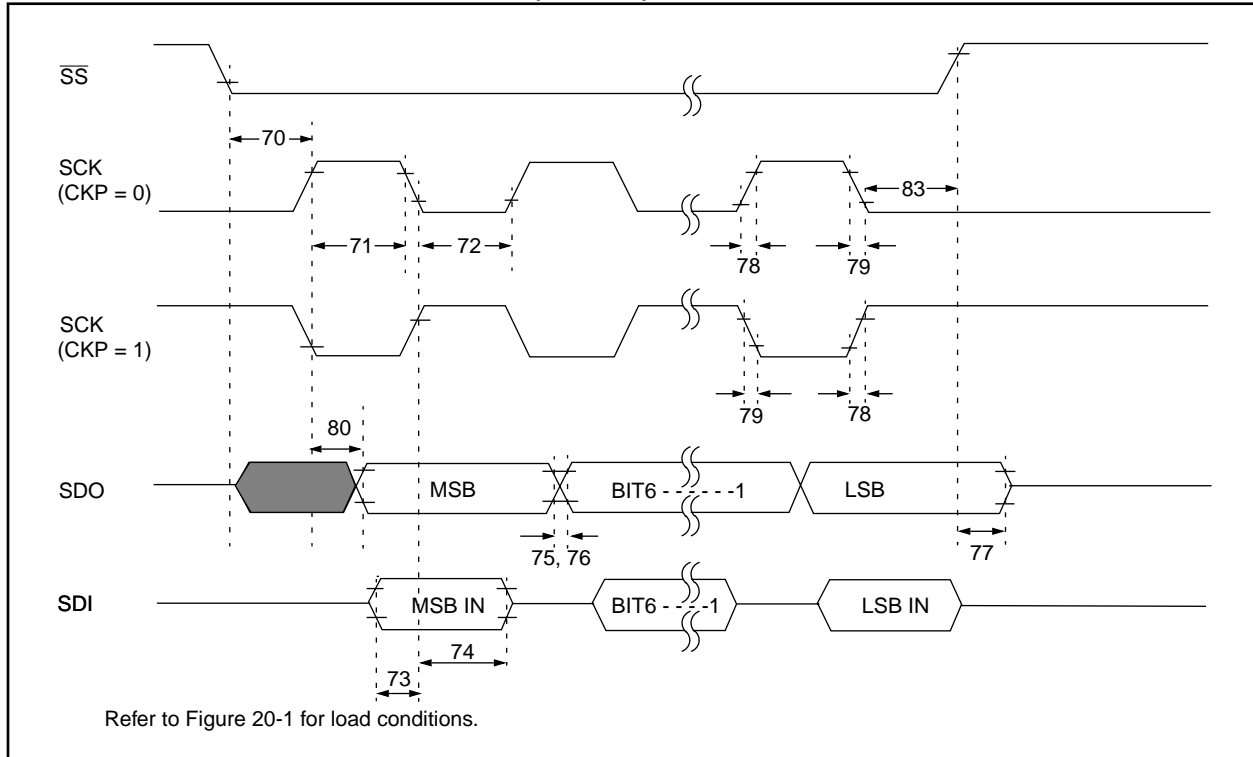




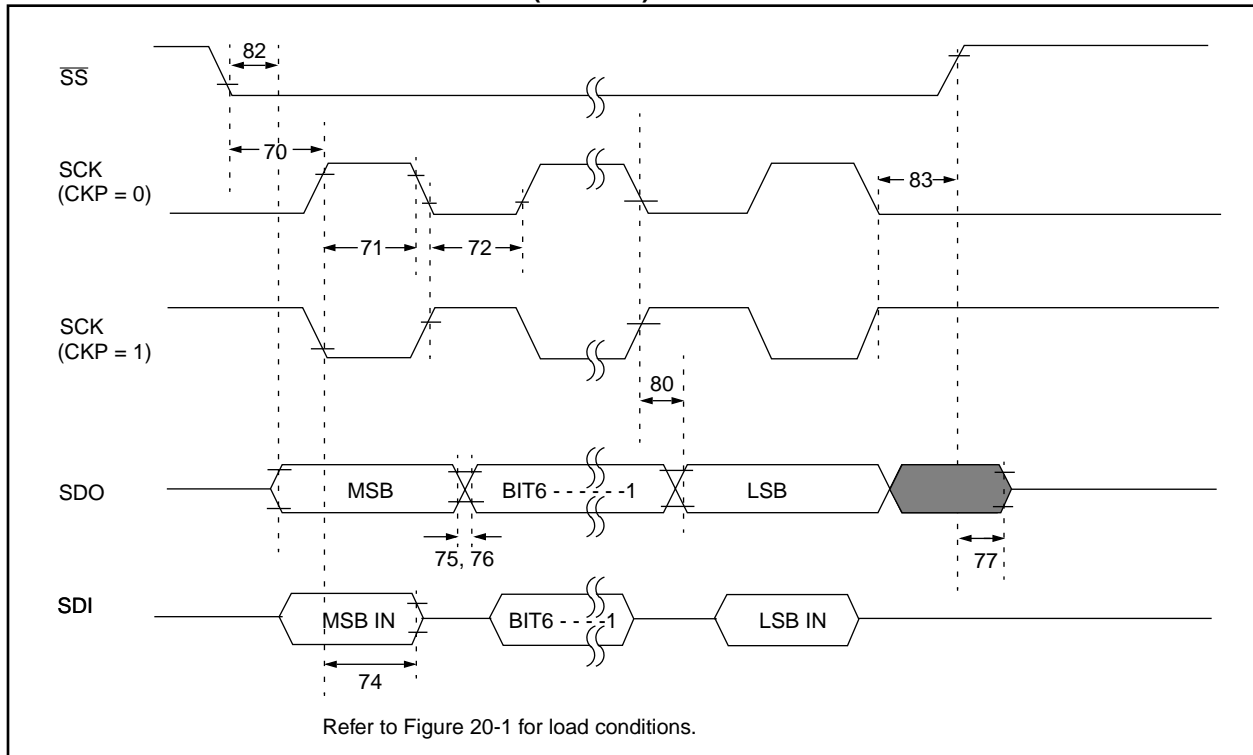
# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

**FIGURE 20-11: SPI SLAVE MODE TIMING (CKE = 0)**



**FIGURE 20-12: SPI SLAVE MODE TIMING (CKE = 1)**



# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

**TABLE 20-8: SPI MODE REQUIREMENTS**

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
70*	TssL2scH, TssL2scL	$\overline{SS}\downarrow$ to SCK $\downarrow$ or SCK $\uparrow$ input	T <sub>CY</sub>	—	—	ns	
71*	TscH	SCK input high time (slave mode)	T <sub>CY</sub> + 20	—	—	ns	
72*	TscL	SCK input low time (slave mode)	T <sub>CY</sub> + 20	—	—	ns	
73*	TdiV2scH, TdiV2scL	Setup time of SDI data input to SCK edge	100	—	—	ns	
74*	Tsch2diL, TscL2diL	Hold time of SDI data input to SCK edge	100	—	—	ns	
75*	TdoR	SDO data output rise time	—	10	25	ns	
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)	—	10	25	ns	
79*	TscF	SCK output fall time (master mode)	—	10	25	ns	
80*	Tsch2doV, TscL2doV	SDO data output valid after SCK edge	—	—	50	ns	
81*	TdoV2scH, TdoV2scL	SDO data output setup to SCK edge	T <sub>CY</sub>	—	—	ns	
82*	TssL2doV	SDO data output valid after $\overline{SS}\downarrow$ edge	—	—	50	ns	
83*	Tsch2ssH, TscL2ssH	$\overline{SS}\uparrow$ after SCK edge	1.5T <sub>CY</sub> + 40	—	—	ns	

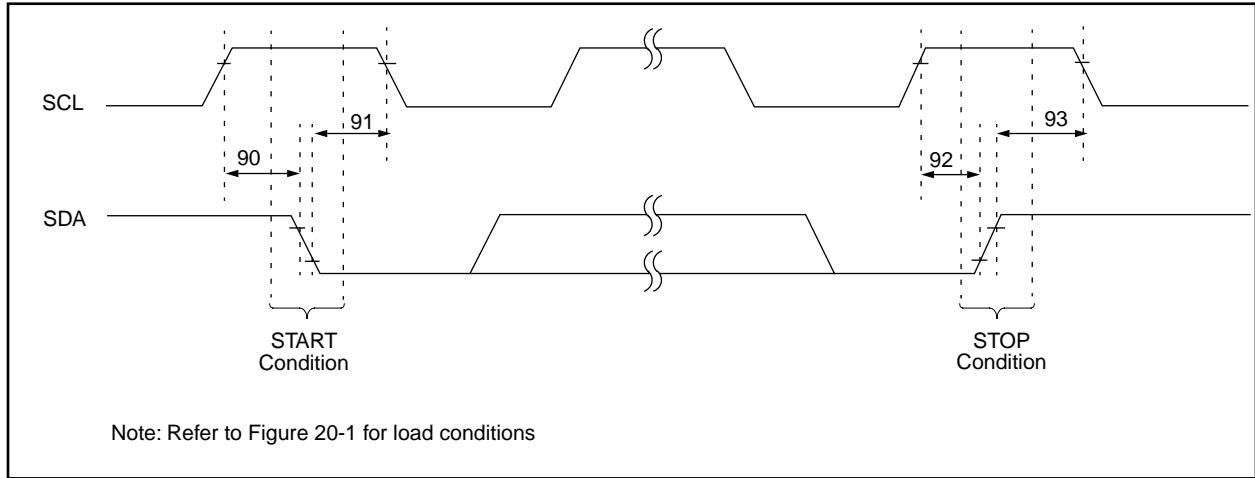
\* These parameters are characterized but not tested.

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

# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

**FIGURE 20-13: I<sup>2</sup>C BUS START/STOP BITS TIMING**



**TABLE 20-9: I<sup>2</sup>C BUS START/STOP BITS REQUIREMENTS**

Parameter No.	Sym	Characteristic	Min	Typ	Max	Units	Conditions	
90	T <sub>SU:STA</sub>	START condition Setup time	100 kHz mode	4700	—	—	ns	Only relevant for repeated START condition
			400 kHz mode	600	—	—		
91	T <sub>HD:STA</sub>	START condition Hold time	100 kHz mode	4000	—	—	ns	After this period the first clock pulse is generated
			400 kHz mode	600	—	—		
92	T <sub>SU:STO</sub>	STOP condition Setup time	100 kHz mode	4700	—	—	ns	
			400 kHz mode	600	—	—		
93	T <sub>HD:STO</sub>	STOP condition Hold time	100 kHz mode	4000	—	—	ns	
			400 kHz mode	600	—	—		

# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

FIGURE 20-14: I<sup>2</sup>C BUS DATA TIMING

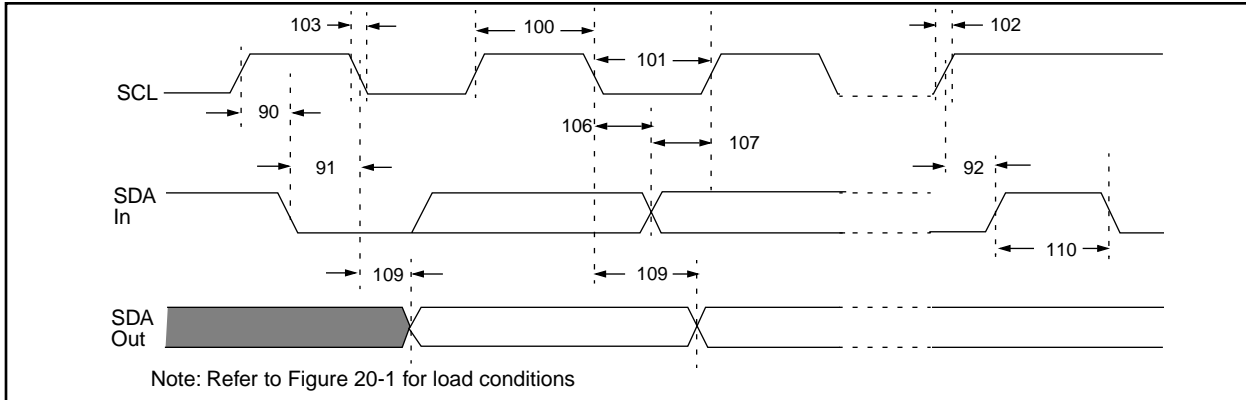


TABLE 20-10: I<sup>2</sup>C BUS DATA REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Max	Units	Conditions	
100	T <sub>HIGH</sub>	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	1.5T <sub>CY</sub>	—		
101	T <sub>LOW</sub>	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	1.5T <sub>CY</sub>	—		
102	T <sub>R</sub>	SDA and SCL rise time	100 kHz mode	—	1000	ns	
			400 kHz mode	20 + 0.1C <sub>b</sub>	300	ns	C <sub>b</sub> is specified to be from 10 to 400 pF
103	T <sub>F</sub>	SDA and SCL fall time	100 kHz mode	—	300	ns	
			400 kHz mode	20 + 0.1C <sub>b</sub>	300	ns	C <sub>b</sub> is specified to be from 10 to 400 pF
90	T <sub>SU:STA</sub>	START condition setup time	100 kHz mode	4.7	—	μs	Only relevant for repeated START condition
			400 kHz mode	0.6	—	μs	
91	T <sub>HD:STA</sub>	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	T <sub>HD:DAT</sub>	Data input hold time	100 kHz mode	0	—	ns	
			400 kHz mode	0	0.9	μs	
107	T <sub>SU:DAT</sub>	Data input setup time	100 kHz mode	250	—	ns	Note 2
			400 kHz mode	100	—	ns	
92	T <sub>SU:STO</sub>	STOP condition setup time	100 kHz mode	4.7	—	μs	
			400 kHz mode	0.6	—	μs	
109	T <sub>A</sub>	Output valid from clock	100 kHz mode	—	3500	ns	Note 1
			400 kHz mode	—	—	ns	
110	T <sub>BUF</sub>	Bus free time	100 kHz mode	4.7	—	μs	Time the bus must be free before a new transmission can start
			400 kHz mode	1.3	—	μs	
	C <sub>b</sub>	Bus capacitive loading	—	400	pF		

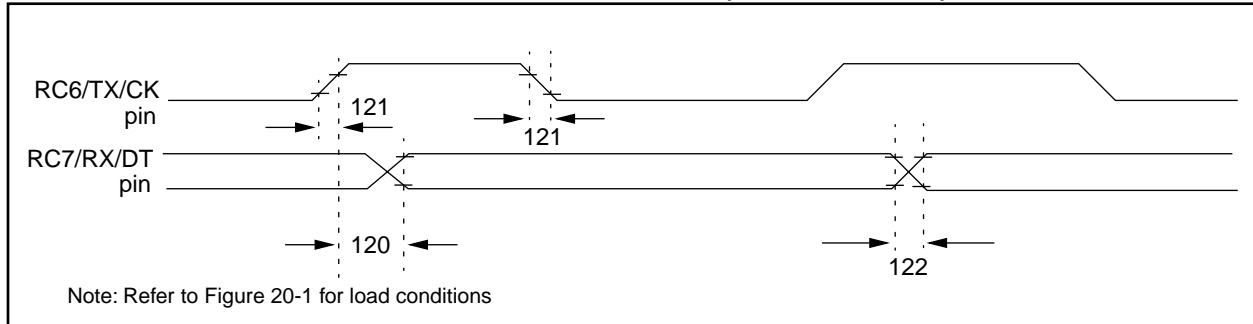
Note 1: As a transmitter, the device must provide this internal minimum delay time to bridge the undefined region (min. 300 ns) of the falling edge of SCL to avoid unintended generation of START or STOP conditions.

Note 2: A fast-mode (400 kHz) I<sup>2</sup>C-bus device can be used in a standard-mode (100 kHz) I<sup>2</sup>C-bus system, but the requirement t<sub>SU:DAT</sub> ≥ 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<sub>R</sub> max. + t<sub>SU:DAT</sub> = 1000 + 250 = 1250 ns (according to the standard-mode I<sup>2</sup>C bus specification) before the SCL line is released.

# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

**FIGURE 20-15: USART SYNCHRONOUS TRANSMISSION (MASTER/SLAVE) TIMING**

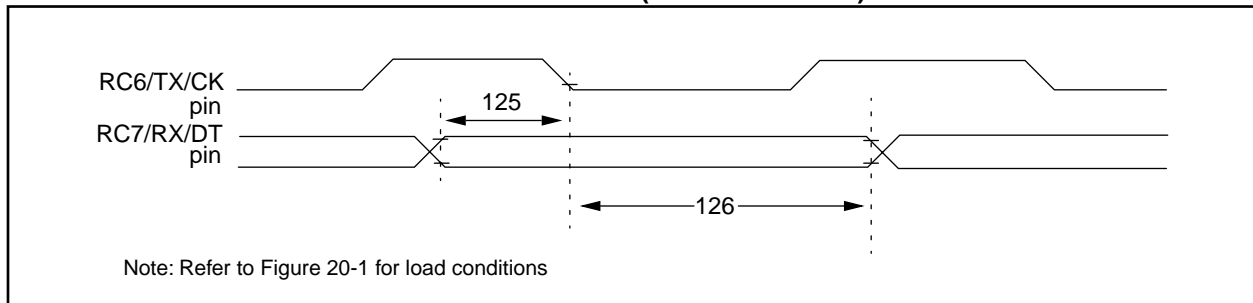


**TABLE 20-11: USART SYNCHRONOUS TRANSMISSION REQUIREMENTS**

Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
120	TckH2dtV	SYNC XMIT (MASTER & SLAVE)					
		Clock high to data out valid			80	ns	
121	Tckrf	Clock out rise time and fall time (Master Mode)	PIC16C76/77		45	ns	
			PIC16LC76/77		50	ns	
122	Tdtrf	Data out rise time and fall time	PIC16C76/77		45	ns	
			PIC16LC76/77		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 20-16: USART SYNCHRONOUS RECEIVE (MASTER/SLAVE) TIMING**



**TABLE 20-12: USART SYNCHRONOUS RECEIVE REQUIREMENTS**

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
125	TdtV2ckL	SYNC RCV (MASTER & SLAVE) Data setup before CK ↓ (DT setup time)	15			ns	
126	TckL2dtl	Data hold after CK ↓ (DT hold time)	15			ns	

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

# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

TABLE 20-13: A/D CONVERTER CHARACTERISTICS:

PIC16C76/77-04 (Commercial, Industrial, Extended)  
 PIC16C76/77-10 (Commercial, Industrial, Extended)  
 PIC16C76/77-20 (Commercial, Industrial, Extended)  
 PIC16LC76/77-04 (Commercial, Industrial)

Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions	
A01	NR	Resolution	—	—	8-bits	bit	VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF	
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	E0FF	Offset error	—	—	< ± 1	LSb	VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF	
A10	—	Monotonicity	—	guaranteed	—	—	VSS ≤ VAIN ≤ VREF	
A20	VREF	Reference voltage	3.0V	—	VDD + 0.3	V		
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)	PIC16C76/77	—	180	—	μA	Average current consumption when A/D is on. (Note 1)
			PIC16LC76/77	—	90	—	μA	
A50	IREF	VREF input current (Note 2)	10	—	1000	μA	During VAIN acquisition. Based on differential of VHOLD to VAIN to charge CHOLD, see Section 13.1.	
			—	—	10	μA	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.

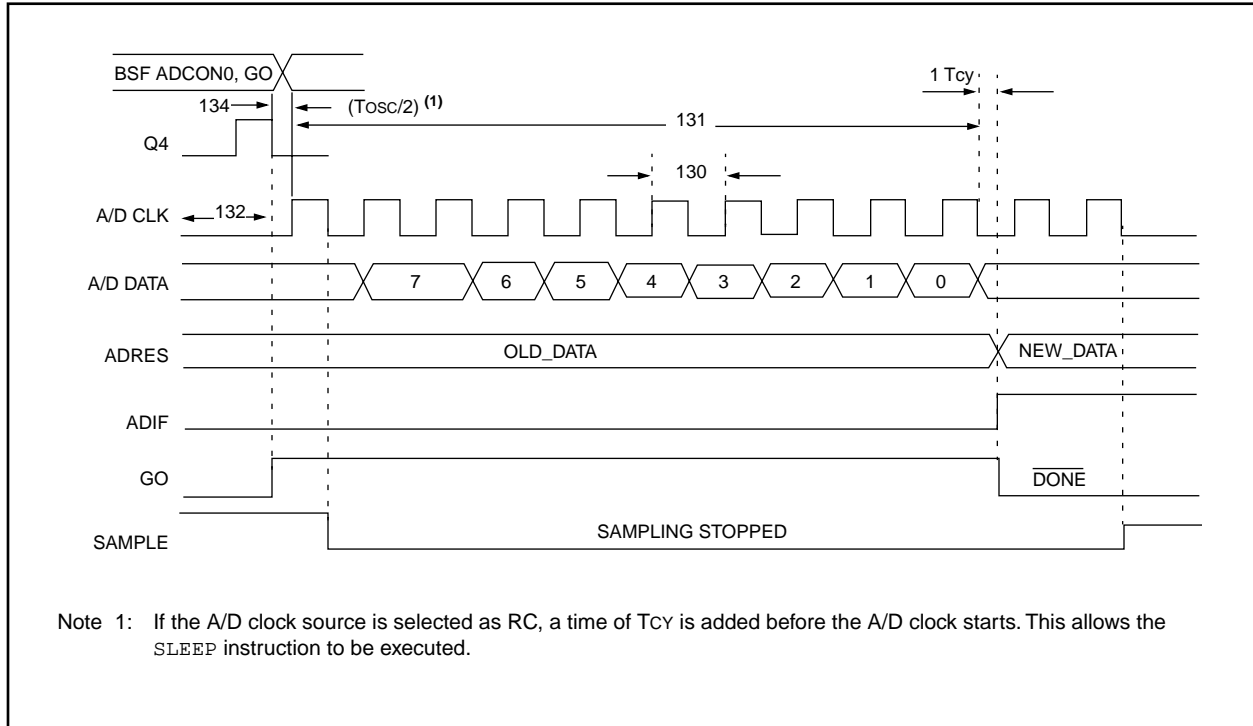
The power-down current spec includes any such leakage from the A/D module.

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

# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

**FIGURE 20-17: A/D CONVERSION TIMING**



**TABLE 20-14: A/D CONVERSION REQUIREMENTS**

Param No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions
130	TAD	A/D clock period	PIC16C76/77	1.6	—	—	μs	TOSC based, VREF ≥ 3.0V
			PIC16LC76/77	2.0	—	—	μs	TOSC based, VREF full range
			PIC16C76/77	2.0	4.0	6.0	μs	A/D RC Mode
			PIC16LC76/77	3.0	6.0	9.0	μs	A/D RC Mode
131	TcNV	Conversion time (not including S/H time) (Note 1)	—	9.5	—	TAD		
132	TACQ	Acquisition time		Note 2	20	—	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).
				5*	—	—	μs	
134	TGo	Q4 to A/D clock start	—	Tosc/2 §	—	—		If the A/D clock source is selected as RC, a time of $T_{CY}$ is added before the A/D clock starts. This allows the SLEEP instruction to be executed.
135	Tswc	Switching from convert → sample time	1.5 §	—	—	—	TAD	

\* These parameters are characterized but not tested.

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

§ This specification ensured by design.

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

Note 2: See Section 13.1 for min conditions.

# PIC16C7X

---

Applicable Devices	72	73	73A	74	74A	76	77
--------------------	----	----	-----	----	-----	----	----

NOTES:



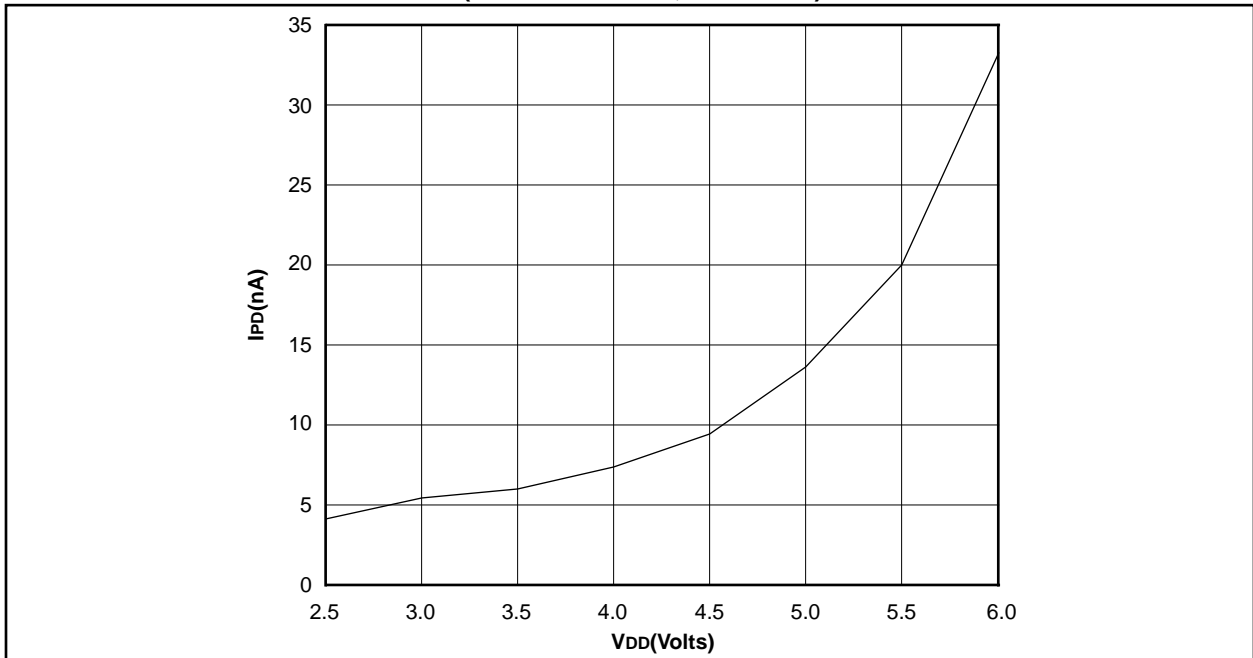
## 21.0 DC AND AC CHARACTERISTICS GRAPHS AND TABLES

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

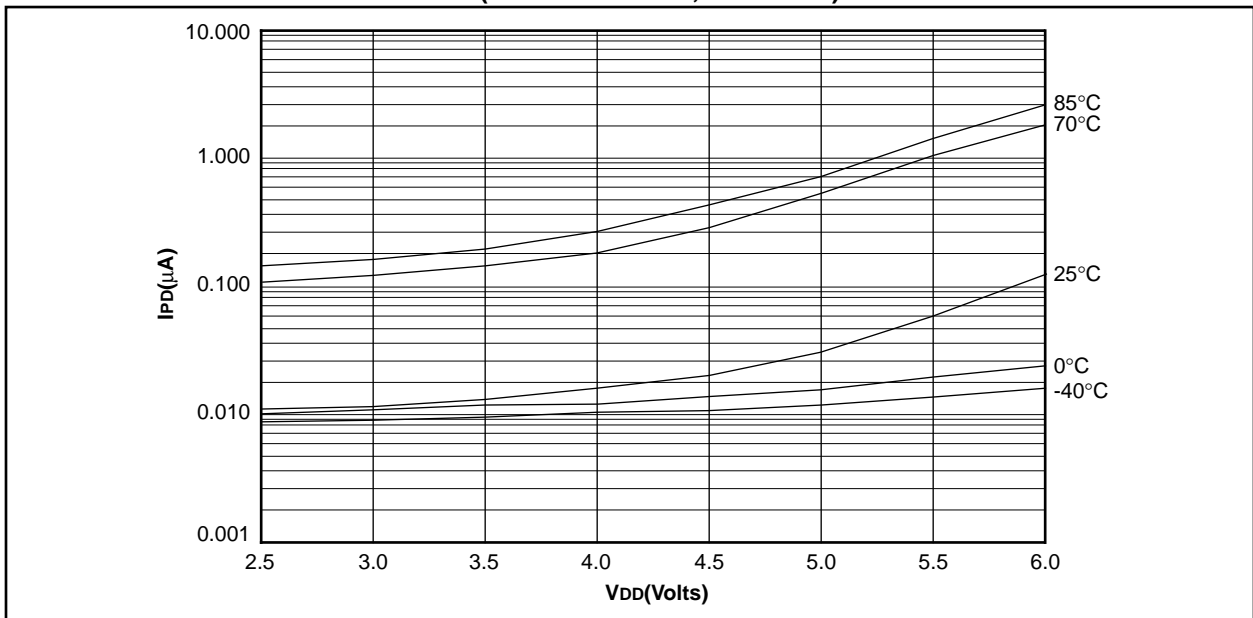
In some graphs or tables the data presented are outside specified operating range (i.e., outside specified  $V_{DD}$  range). This is for information only and devices are guaranteed to operate properly only within the specified range.

**Note:** The data presented in this section is a statistical summary of data collected on units from different lots over a period of time and matrix samples. 'Typical' represents the mean of the distribution at, 25°C, while 'max' or 'min' represents (mean +3 $\sigma$ ) and (mean -3 $\sigma$ ) respectively where  $\sigma$  is standard deviation.

**FIGURE 21-1: TYPICAL  $I_{PD}$  vs.  $V_{DD}$  (WDT DISABLED, RC MODE)**



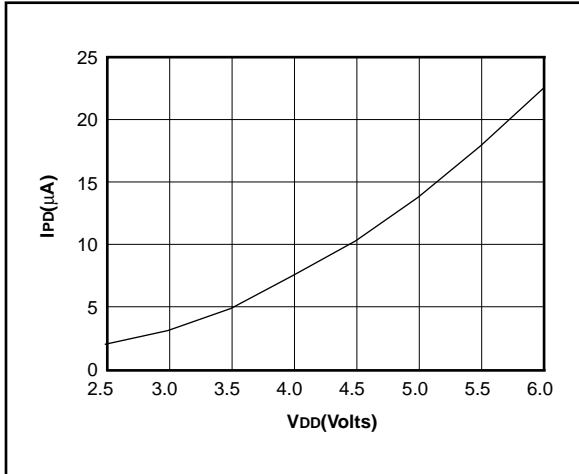
**FIGURE 21-2: MAXIMUM  $I_{PD}$  vs.  $V_{DD}$  (WDT DISABLED, RC MODE)**



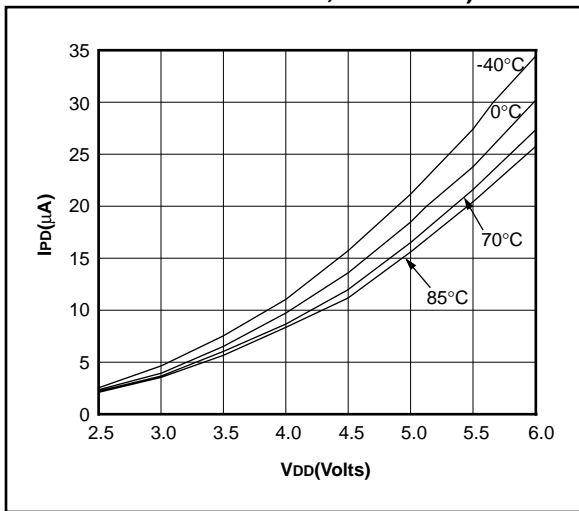
# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

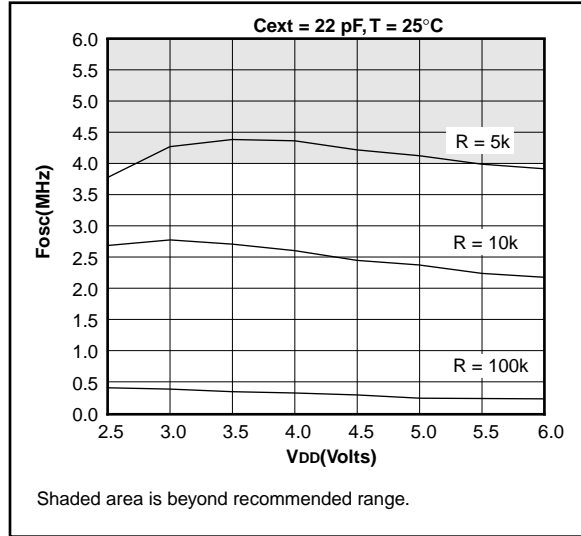
**FIGURE 21-3: TYPICAL IPD vs. VDD @ 25°C (WDT ENABLED, RC MODE)**



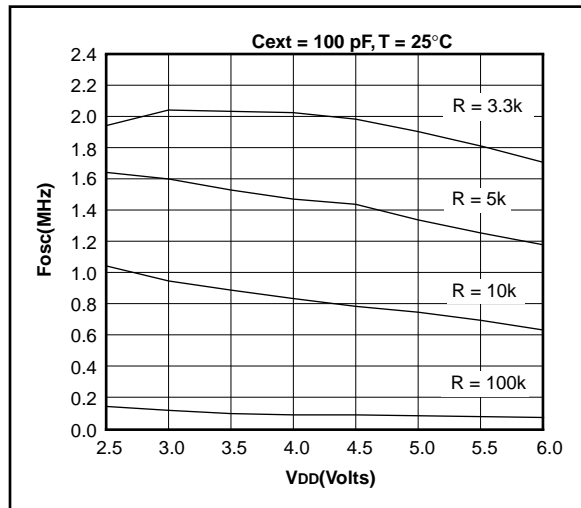
**FIGURE 21-4: MAXIMUM IPD vs. VDD (WDT ENABLED, RC MODE)**



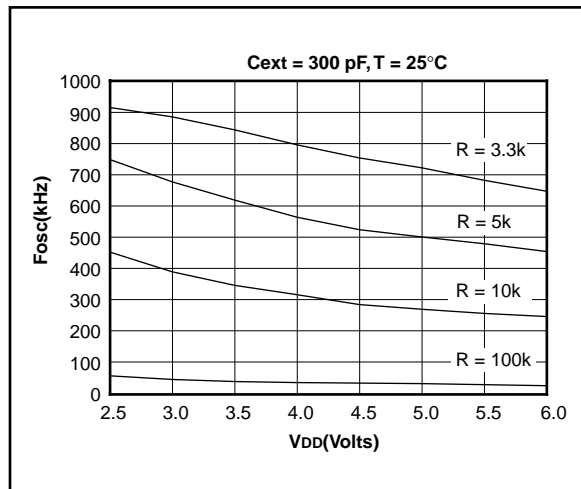
**FIGURE 21-5: TYPICAL RC OSCILLATOR FREQUENCY vs. VDD**



**FIGURE 21-6: TYPICAL RC OSCILLATOR FREQUENCY vs. VDD**



**FIGURE 21-7: TYPICAL RC OSCILLATOR FREQUENCY vs. VDD**

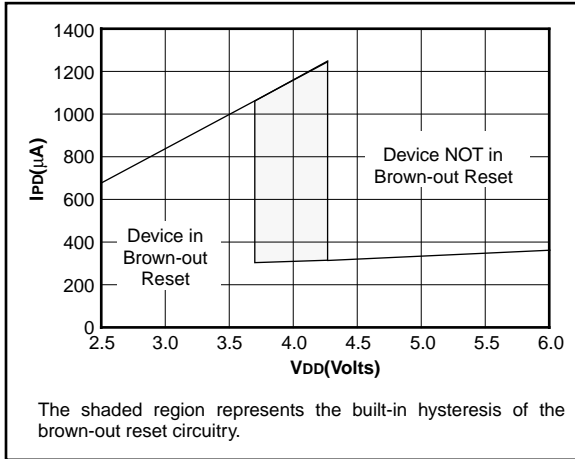


# PIC16C7X

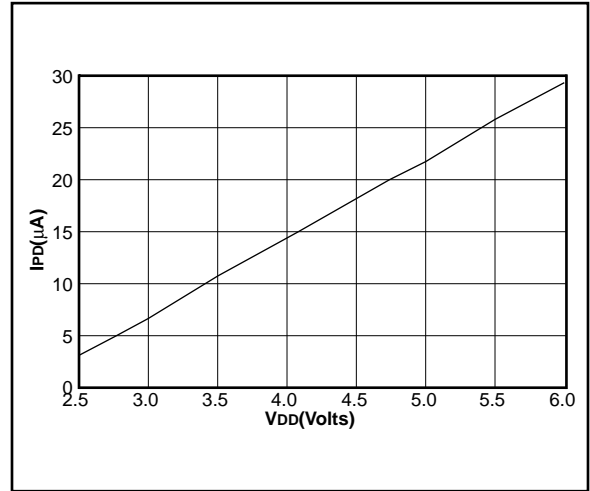
Applicable Devices 

72	73	73A	74	74A	76	77
----	----	-----	----	-----	----	----

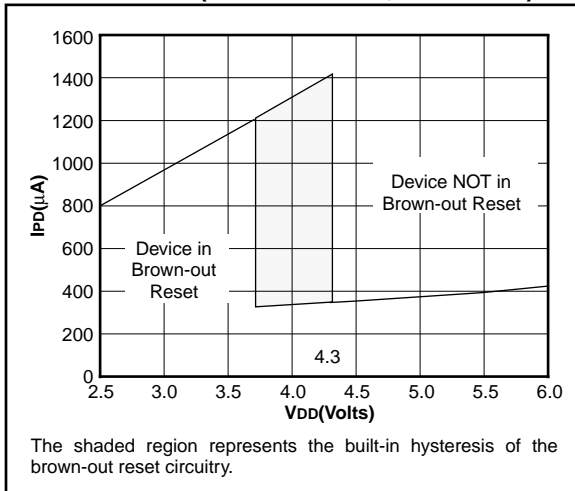
**FIGURE 21-8: TYPICAL  $I_{PD}$  vs.  $V_{DD}$  BROWN-OUT DETECT ENABLED (RC MODE)**



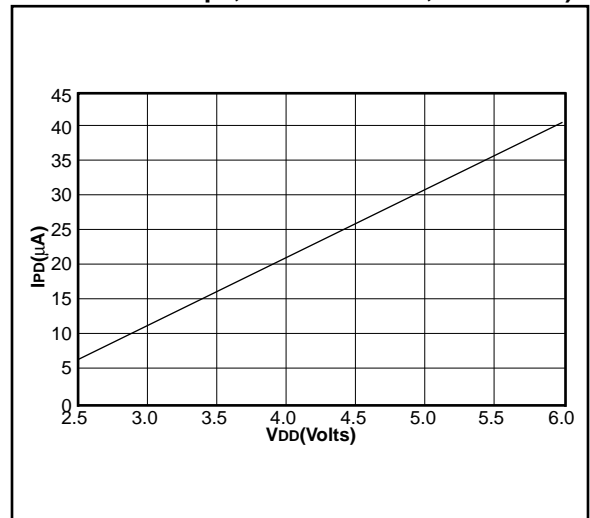
**FIGURE 21-10: TYPICAL  $I_{PD}$  vs. TIMER1 ENABLED (32 kHz, RC0/RC1 = 33 pF/33 pF, RC MODE)**



**FIGURE 21-9: MAXIMUM  $I_{PD}$  vs.  $V_{DD}$  BROWN-OUT DETECT ENABLED (85°C TO -40°C, RC MODE)**



**FIGURE 21-11: MAXIMUM  $I_{PD}$  vs. TIMER1 ENABLED (32 kHz, RC0/RC1 = 33 pF/33 pF, 85°C TO -40°C, RC MODE)**



# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

FIGURE 21-12: TYPICAL  $I_{DD}$  vs. FREQUENCY (RC MODE @ 22 pF, 25°C)

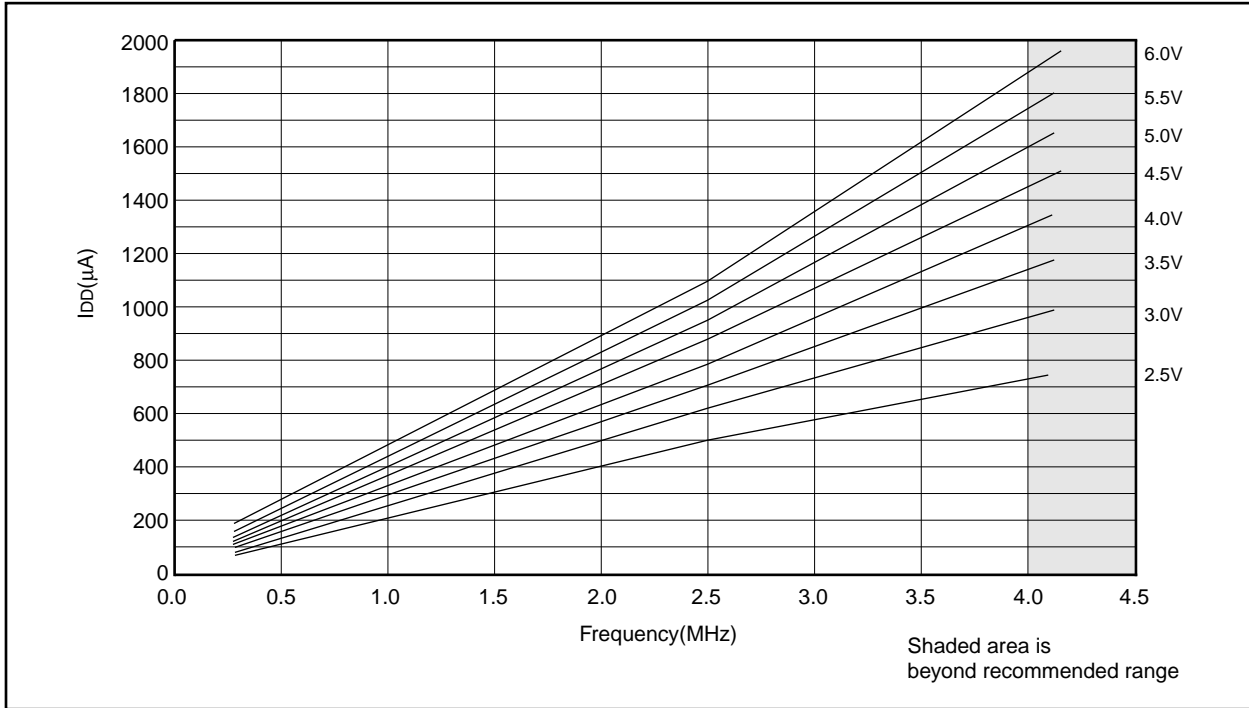
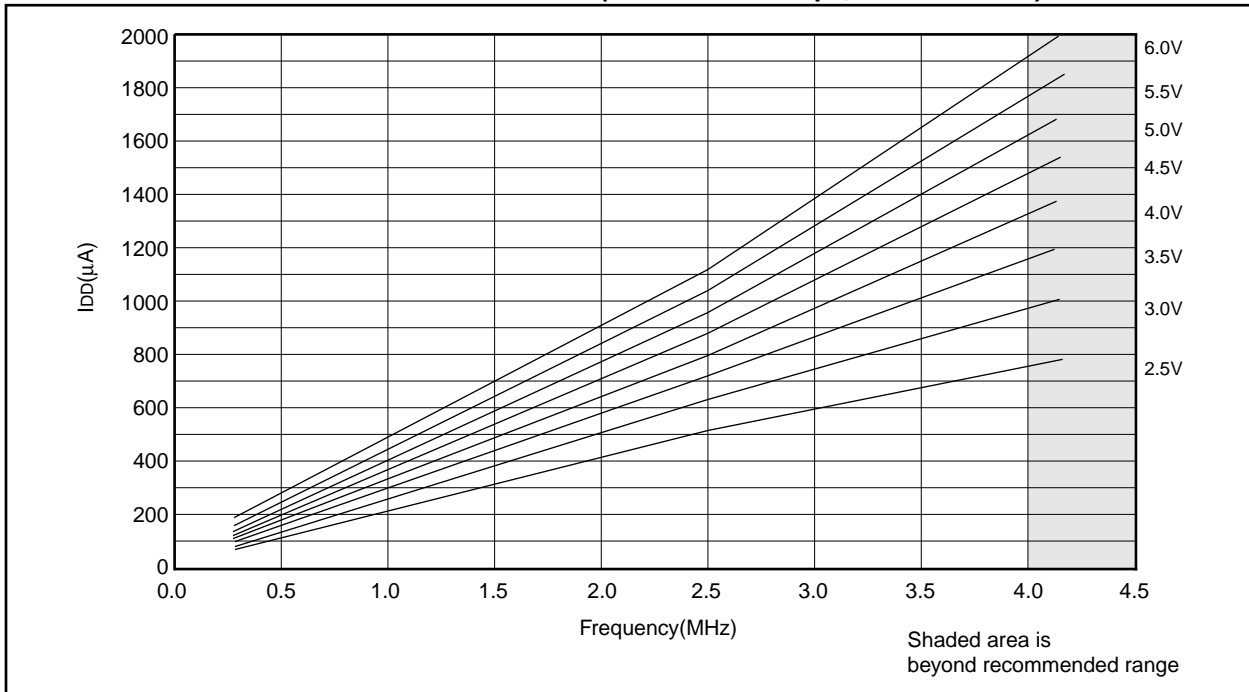


FIGURE 21-13: MAXIMUM  $I_{DD}$  vs. FREQUENCY (RC MODE @ 22 pF, -40°C TO 85°C)



# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

FIGURE 21-14: TYPICAL  $I_{DD}$  vs. FREQUENCY (RC MODE @ 100 pF, 25°C)

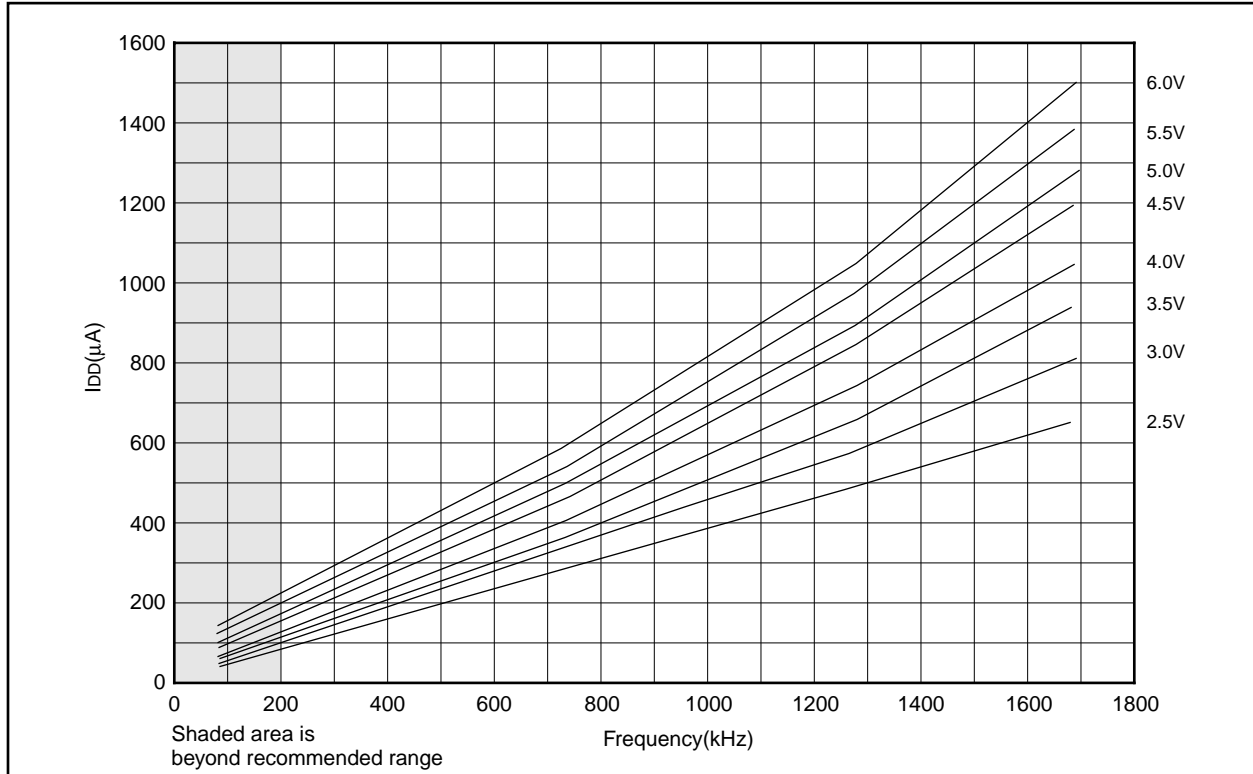
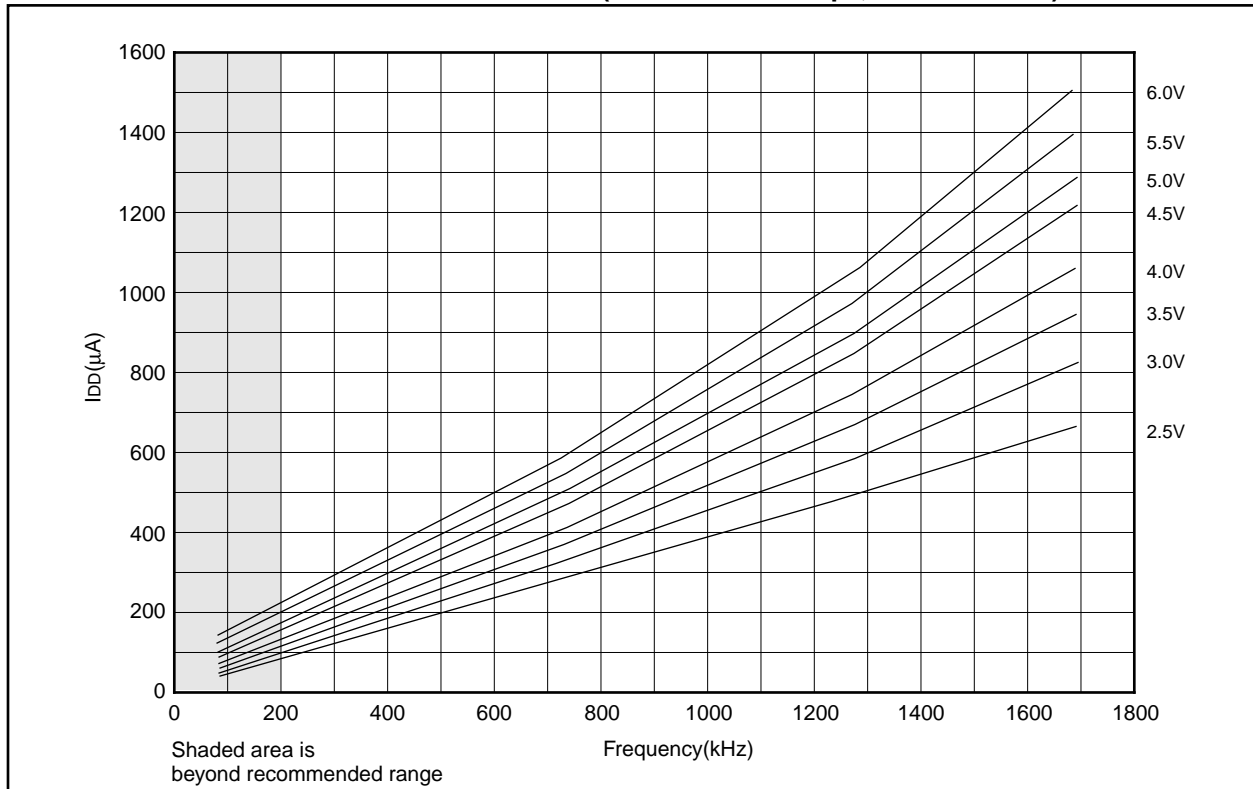


FIGURE 21-15: MAXIMUM  $I_{DD}$  vs. FREQUENCY (RC MODE @ 100 pF, -40°C TO 85°C)



# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

FIGURE 21-16: TYPICAL  $I_{DD}$  vs. FREQUENCY (RC MODE @ 300 pF, 25°C)

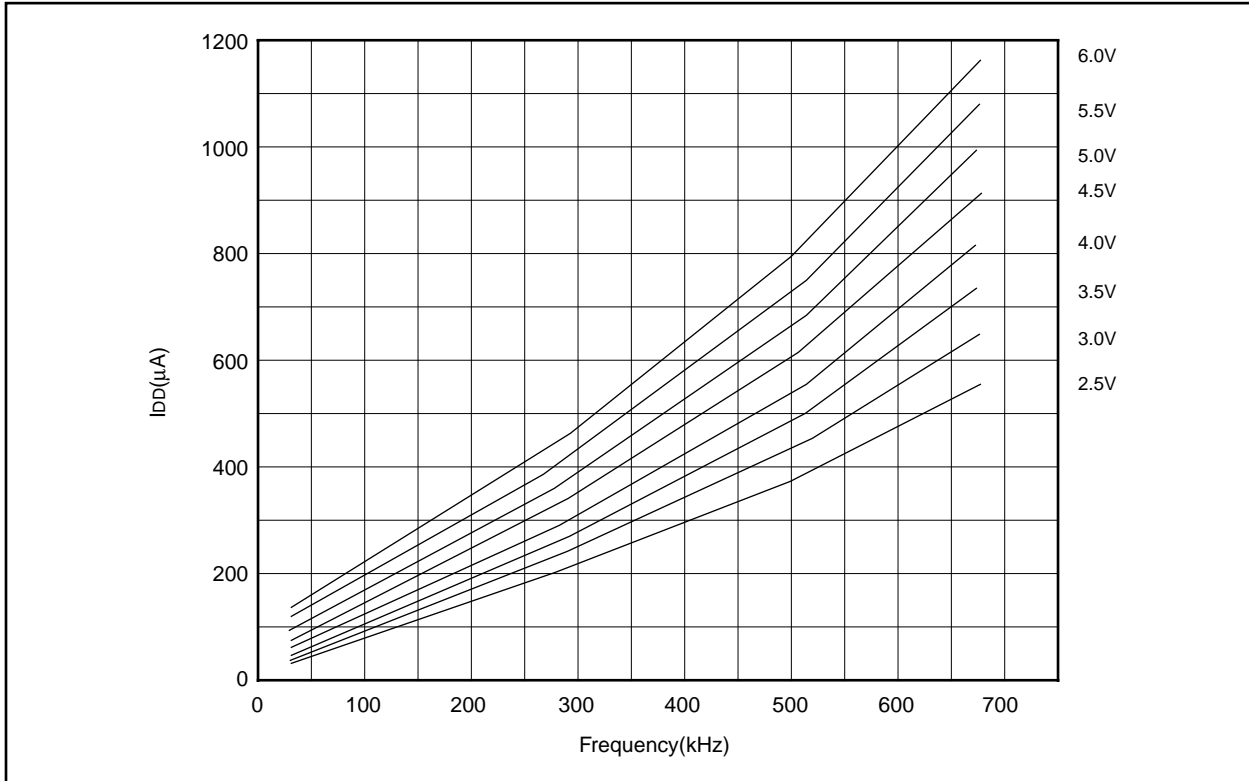
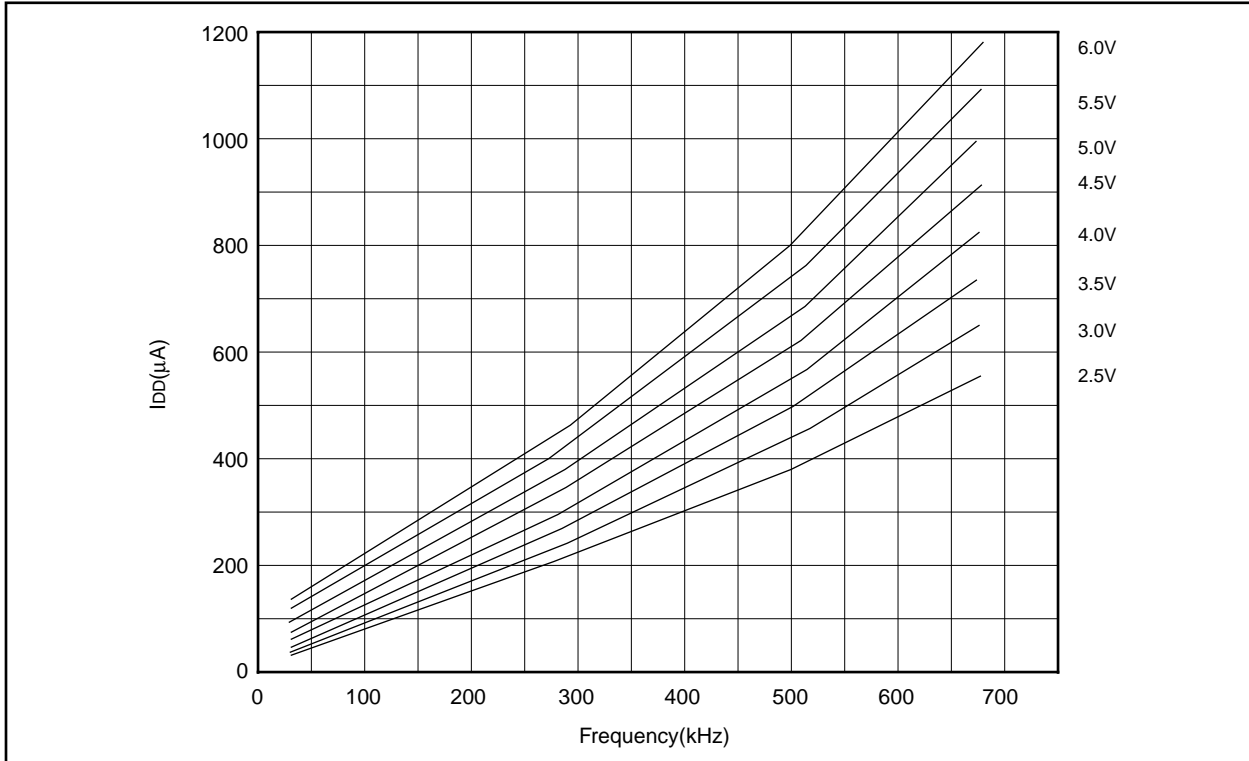


FIGURE 21-17: MAXIMUM  $I_{DD}$  vs. FREQUENCY (RC MODE @ 300 pF, -40°C TO 85°C)

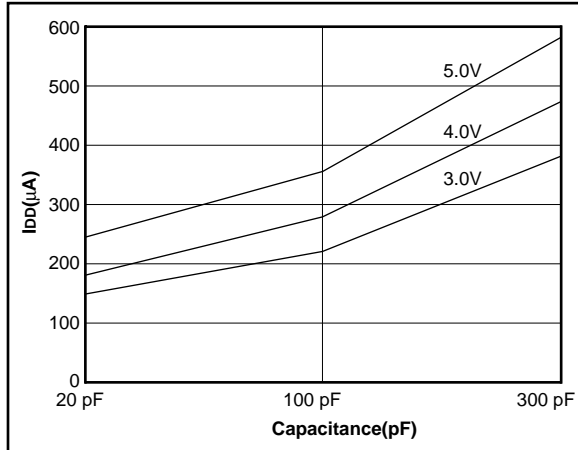


Data based on matrix samples. See first page of this section for details.

# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

**FIGURE 21-18: TYPICAL I<sub>DD</sub> vs. CAPACITANCE @ 500 kHz (RC MODE)**

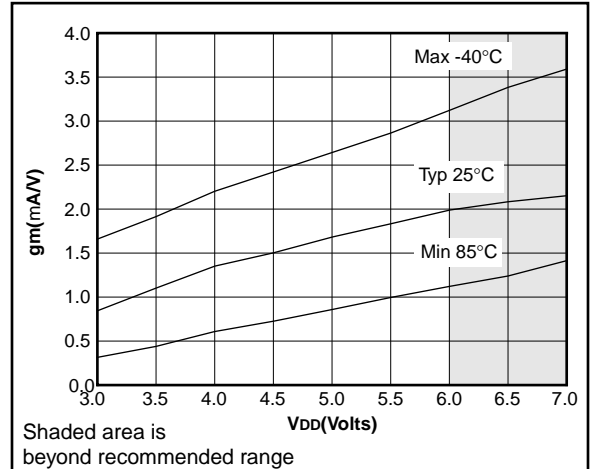


**TABLE 21-1: RC OSCILLATOR FREQUENCIES**

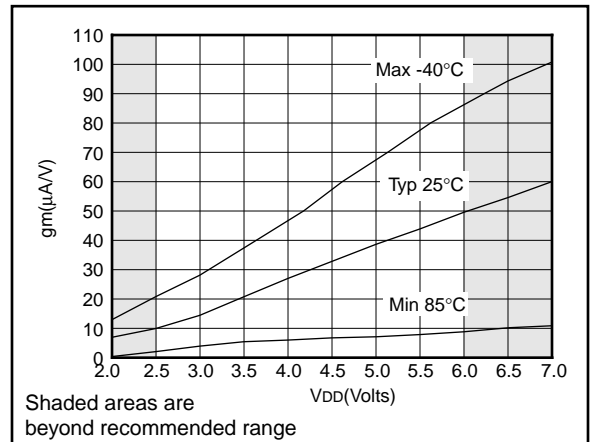
Cext	Rext	Average	
		Fosc @ 5V, 25°C	
22 pF	5k	4.12 MHz	± 1.4%
	10k	2.35 MHz	± 1.4%
	100k	268 kHz	± 1.1%
100 pF	3.3k	1.80 MHz	± 1.0%
	5k	1.27 MHz	± 1.0%
	10k	688 kHz	± 1.2%
300 pF	100k	77.2 kHz	± 1.0%
	3.3k	707 kHz	± 1.4%
	5k	501 kHz	± 1.2%
	10k	269 kHz	± 1.6%
	100k	28.3 kHz	± 1.1%

The percentage variation indicated here is part to part variation due to normal process distribution. The variation indicated is ±3 standard deviation from average value for V<sub>DD</sub> = 5V.

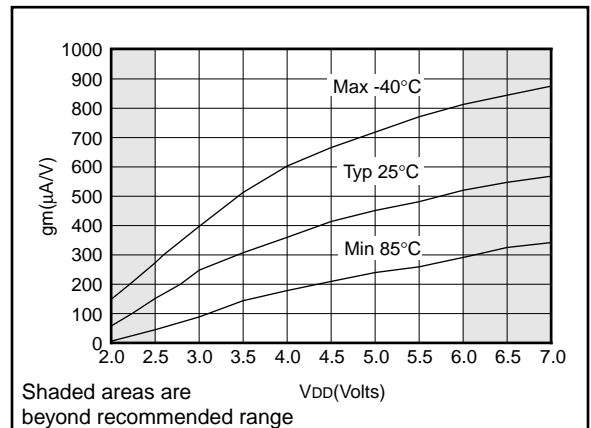
**FIGURE 21-19: TRANSCONDUCTANCE(g<sub>m</sub>) OF HS OSCILLATOR vs. V<sub>DD</sub>**



**FIGURE 21-20: TRANSCONDUCTANCE(g<sub>m</sub>) OF LP OSCILLATOR vs. V<sub>DD</sub>**



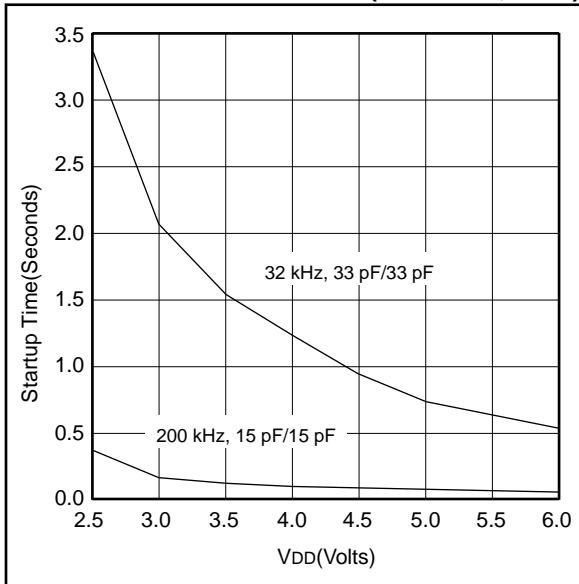
**FIGURE 21-21: TRANSCONDUCTANCE(g<sub>m</sub>) OF XT OSCILLATOR vs. V<sub>DD</sub>**



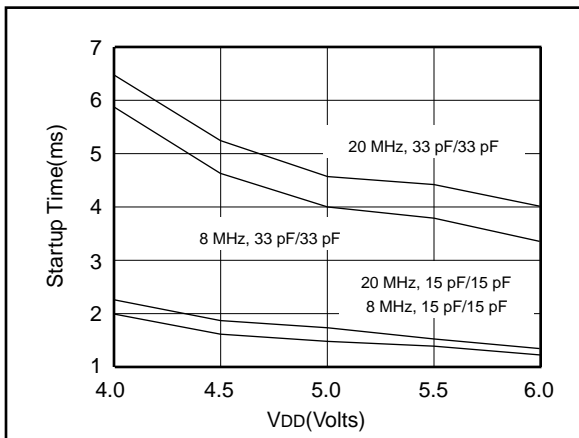
# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

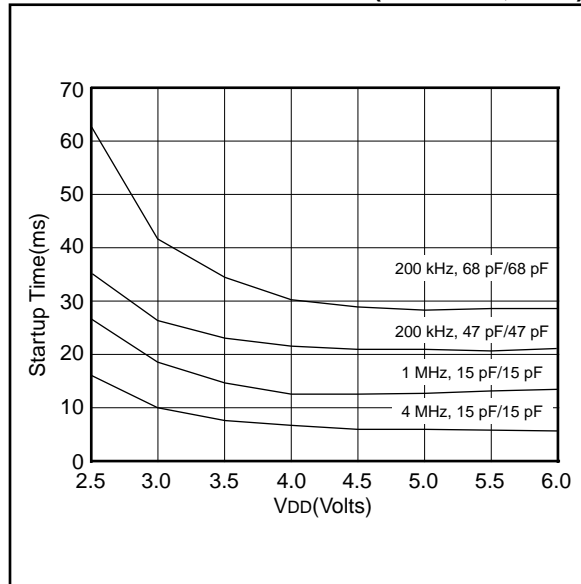
**FIGURE 21-22: TYPICAL XTAL STARTUP TIME vs. VDD (LP MODE, 25°C)**



**FIGURE 21-23: TYPICAL XTAL STARTUP TIME vs. VDD (HS MODE, 25°C)**



**FIGURE 21-24: TYPICAL XTAL STARTUP TIME vs. VDD (XT MODE, 25°C)**



**TABLE 21-2: CAPACITOR SELECTION FOR CRYSTAL OSCILLATORS**

Osc Type	Crystal Freq	Cap. Range C1	Cap. Range C2
LP	32 kHz	33 pF	33 pF
	200 kHz	15 pF	15 pF
XT	200 kHz	47-68 pF	47-68 pF
	1 MHz	15 pF	15 pF
	4 MHz	15 pF	15 pF
HS	4 MHz	15 pF	15 pF
	8 MHz	15-33 pF	15-33 pF
	20 MHz	15-33 pF	15-33 pF
<b>Crystals Used</b>			
32 kHz	Epson C-001R32.768K-A		± 20 PPM
200 kHz	STD XTL 200.000KHz		± 20 PPM
1 MHz	ECS ECS-10-13-1		± 50 PPM
4 MHz	ECS ECS-40-20-1		± 50 PPM
8 MHz	EPSON CA-301 8.000M-C		± 30 PPM
20 MHz	EPSON CA-301 20.000M-C		± 30 PPM

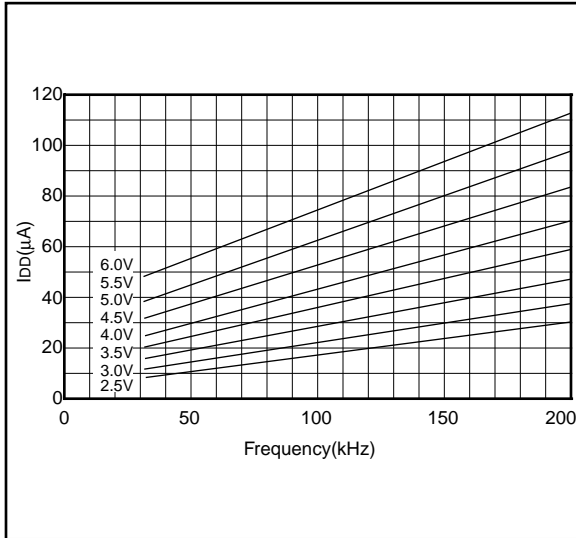


# PIC16C7X

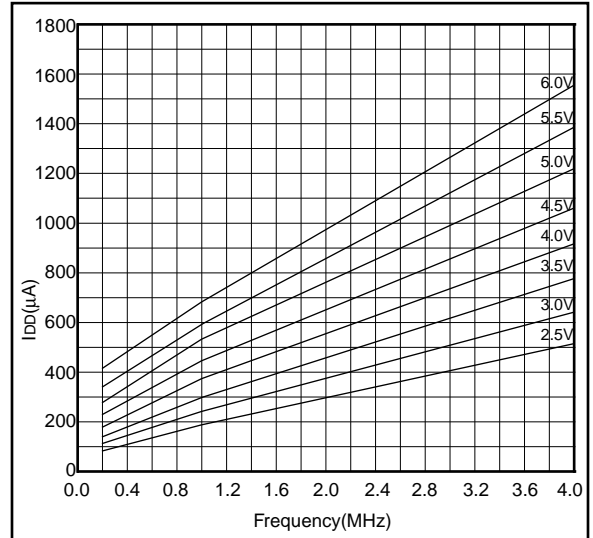
Applicable Devices 

72	73	73A	74	74A	76	77
----	----	-----	----	-----	----	----

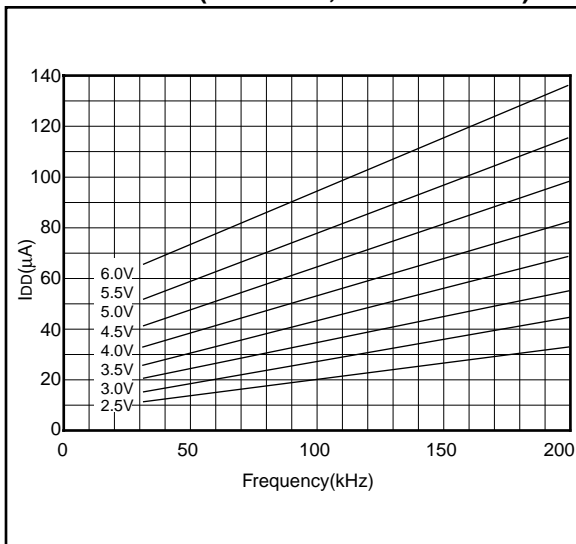
**FIGURE 21-25: TYPICAL I<sub>DD</sub> vs. FREQUENCY  
(LP MODE, 25°C)**



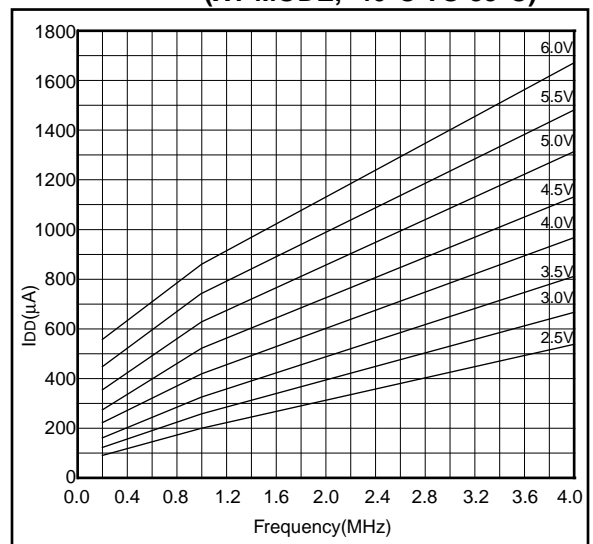
**FIGURE 21-27: TYPICAL I<sub>DD</sub> vs. FREQUENCY  
(XT MODE, 25°C)**



**FIGURE 21-26: MAXIMUM I<sub>DD</sub> vs.  
FREQUENCY  
(LP MODE, 85°C TO -40°C)**



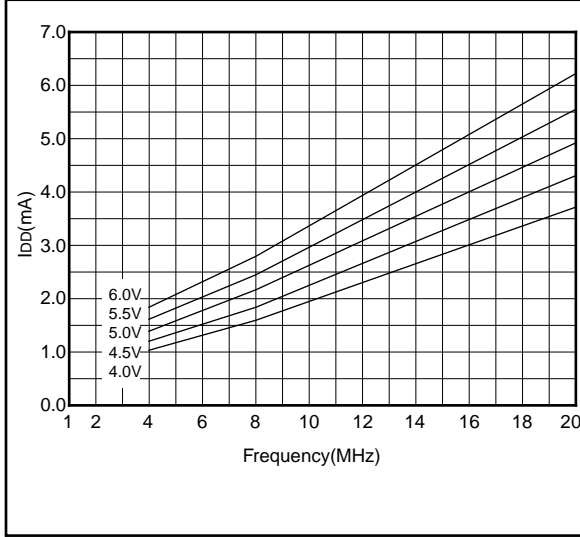
**FIGURE 21-28: MAXIMUM I<sub>DD</sub> vs.  
FREQUENCY  
(XT MODE, -40°C TO 85°C)**



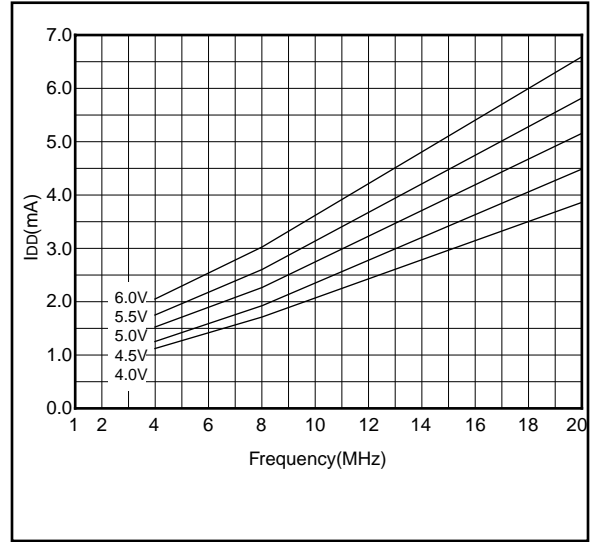
# PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

**FIGURE 21-29: TYPICAL  $I_{DD}$  vs. FREQUENCY  
(HS MODE, 25°C)**

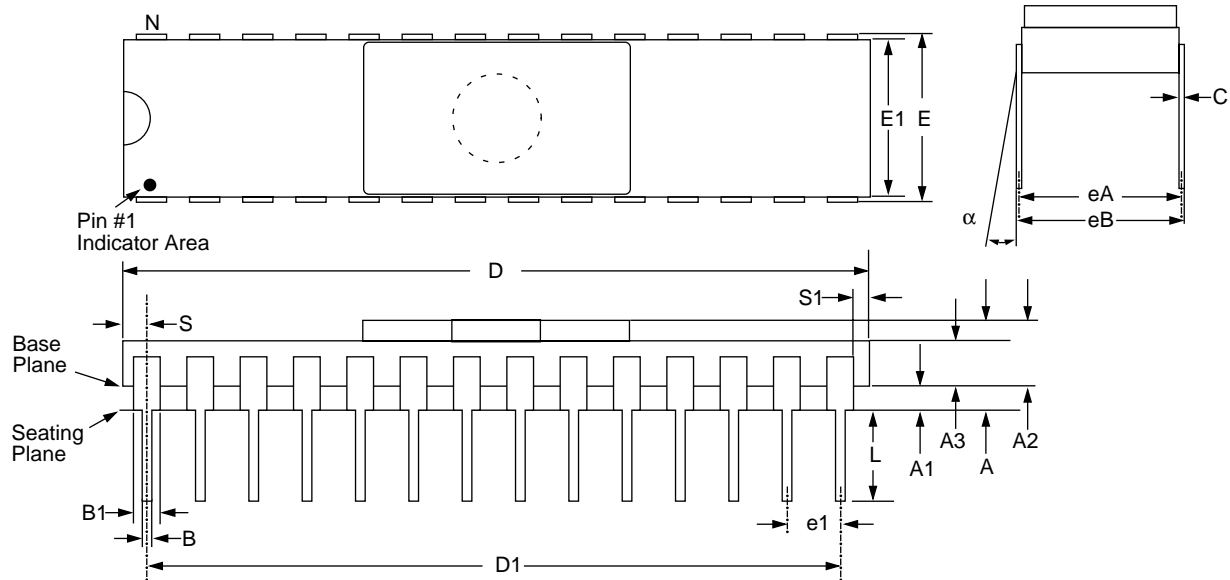


**FIGURE 21-30: MAXIMUM  $I_{DD}$  vs.  
FREQUENCY  
(HS MODE, -40°C TO 85°C)**



## 22.0 PACKAGING INFORMATION

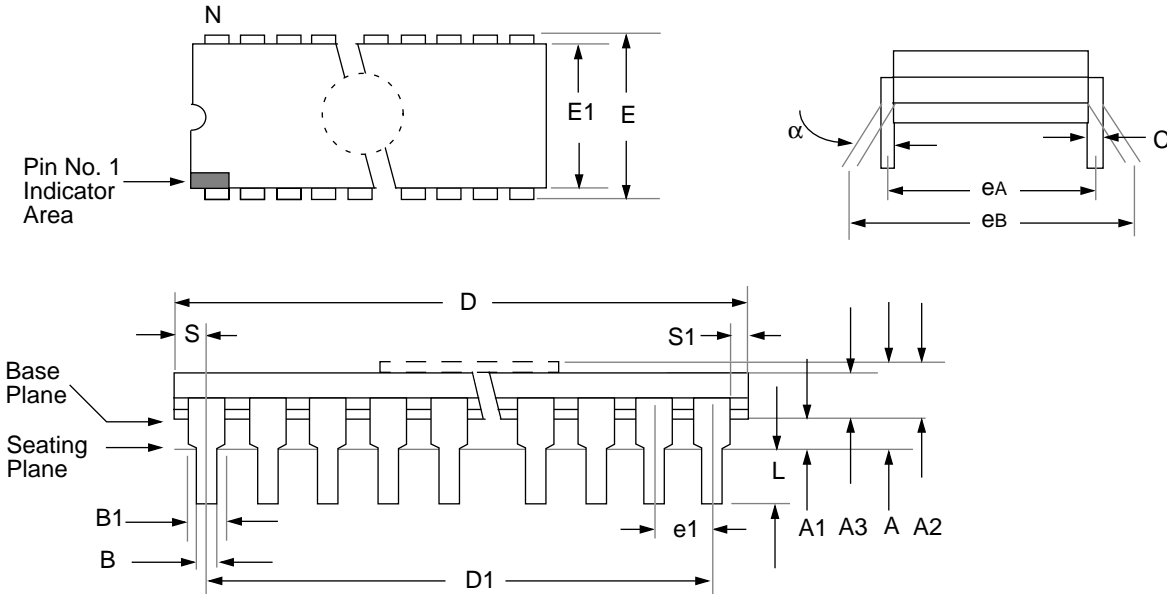
### 22.1 28-Lead Ceramic Side Brazed Dual In-Line with Window (300 mil)(JW)



Package Group: Ceramic Side Brazed Dual In-Line (CER)						
Symbol	Millimeters			Inches		
	Min	Max	Notes	Min	Max	Notes
$\alpha$	0°	10°		0°	10°	
A	3.937	5.030		0.155	0.198	
A1	1.016	1.524		0.040	0.060	
A2	2.921	3.506		0.115	0.138	
A3	1.930	2.388		0.076	0.094	
B	0.406	0.508		0.016	0.020	
B1	1.219	1.321	Typical	0.048	0.052	
C	0.228	0.305	Typical	0.009	0.012	
D	35.204	35.916		1.386	1.414	
D1	32.893	33.147	Reference	1.295	1.305	
E	7.620	8.128		0.300	0.320	
E1	7.366	7.620		0.290	0.300	
e1	2.413	2.667	Typical	0.095	0.105	
eA	7.366	7.874	Reference	0.290	0.310	
eB	7.594	8.179		0.299	0.322	
L	3.302	4.064		0.130	0.160	
N	28	28		28	28	
S	1.143	1.397		0.045	0.055	
S1	0.533	0.737		0.021	0.029	

# PIC16C7X

## 22.2 40-Lead Ceramic Cerdip Dual In-line with Window (600 mil) (JW)

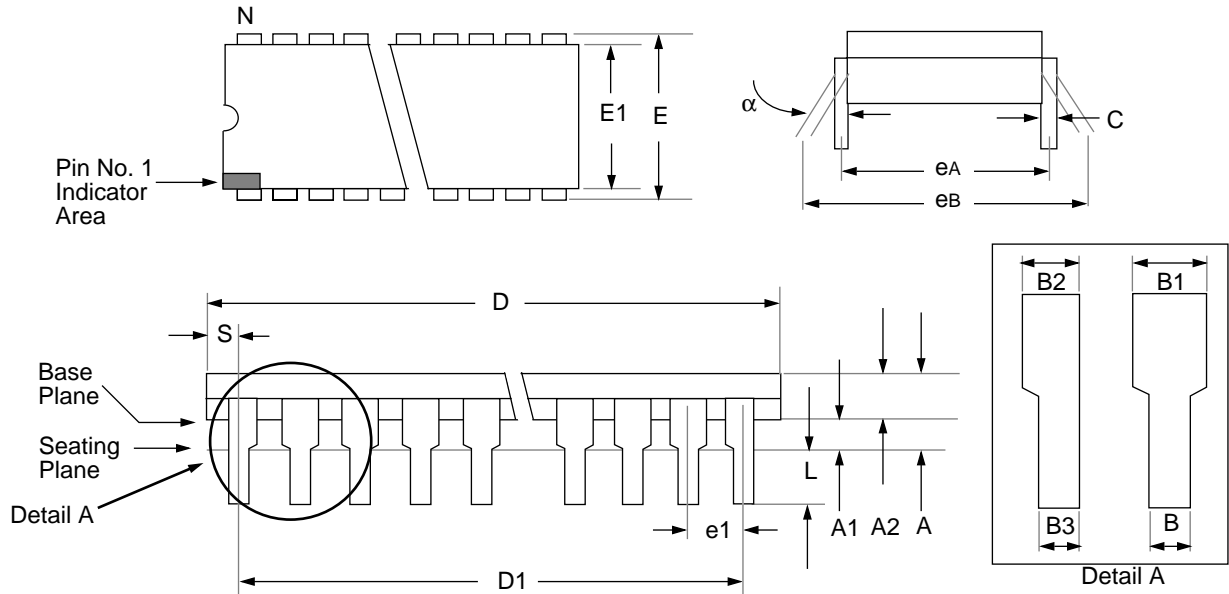


Package Group: Ceramic Cerdip Dual In-Line (CDP)

Symbol	Millimeters			Inches		
	Min	Max	Notes	Min	Max	Notes
$\alpha$	0°	10°		0°	10°	
A	4.318	5.715		0.170	0.225	
A1	0.381	1.778		0.015	0.070	
A2	3.810	4.699		0.150	0.185	
A3	3.810	4.445		0.150	0.175	
B	0.355	0.585		0.014	0.023	
B1	1.270	1.651	Typical	0.050	0.065	Typical
C	0.203	0.381	Typical	0.008	0.015	Typical
D	51.435	52.705		2.025	2.075	
D1	48.260	48.260	Reference	1.900	1.900	Reference
E	15.240	15.875		0.600	0.625	
E1	12.954	15.240		0.510	0.600	
e1	2.540	2.540	Reference	0.100	0.100	Reference
eA	14.986	16.002	Typical	0.590	0.630	Typical
eB	15.240	18.034		0.600	0.710	
L	3.175	3.810		0.125	0.150	
N	40	40		40	40	
S	1.016	2.286		0.040	0.090	
S1	0.381	1.778		0.015	0.070	

# PIC16C7X

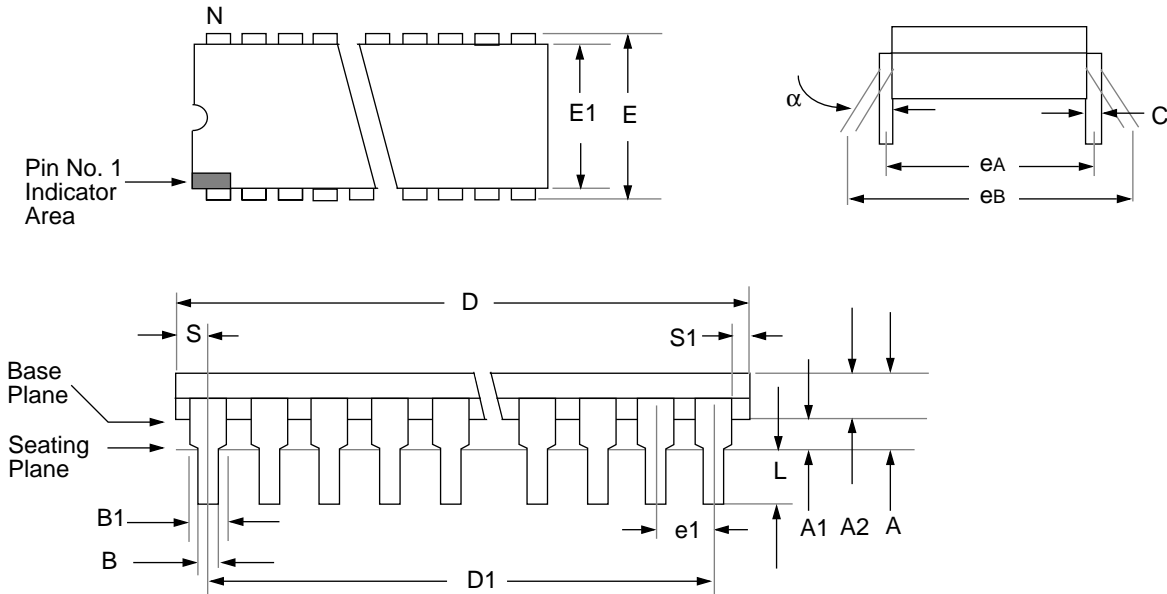
## 22.3 28-Lead Plastic Dual In-line (300 mil) (SP)



Package Group: Plastic Dual In-Line (PLA)						
Symbol	Millimeters			Inches		
	Min	Max	Notes	Min	Max	Notes
$\alpha$	0°	10°		0°	10°	
A	3.632	4.572		0.143	0.180	
A1	0.381	–		0.015	–	
A2	3.175	3.556		0.125	0.140	
B	0.406	0.559		0.016	0.022	
B1	1.016	1.651	<b>Typical</b>	0.040	0.065	<b>Typical</b>
B2	0.762	1.016	<b>4 places</b>	0.030	0.040	<b>4 places</b>
B3	0.203	0.508	<b>4 places</b>	0.008	0.020	<b>4 places</b>
C	0.203	0.331	<b>Typical</b>	0.008	0.013	<b>Typical</b>
D	34.163	35.179		1.385	1.395	
D1	33.020	33.020	<b>Reference</b>	1.300	1.300	<b>Reference</b>
E	7.874	8.382		0.310	0.330	
E1	7.112	7.493		0.280	0.295	
e1	2.540	2.540	<b>Typical</b>	0.100	0.100	<b>Typical</b>
eA	7.874	7.874	<b>Reference</b>	0.310	0.310	<b>Reference</b>
eB	8.128	9.652		0.320	0.380	
L	3.175	3.683		0.125	0.145	
N	28	–		28	–	
S	0.584	1.220		0.023	0.048	

# PIC16C7X

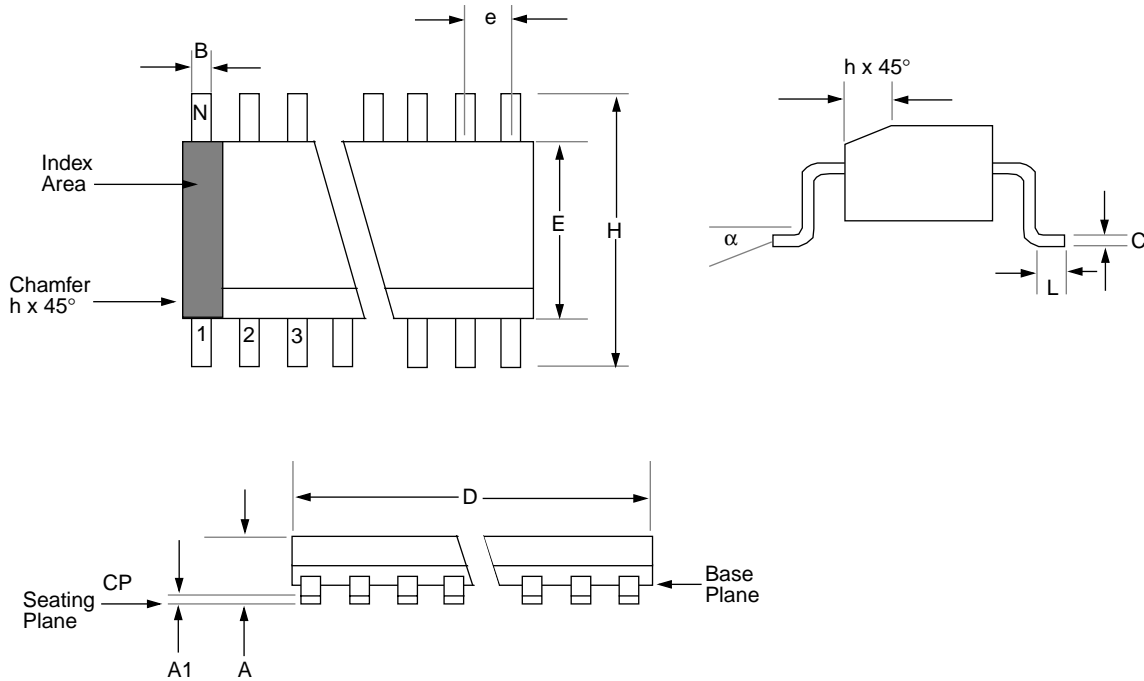
## 22.4 40-Lead Plastic Dual In-line (600 mil) (P)



Package Group: Plastic Dual In-Line (PLA)						
Symbol	Millimeters			Inches		
	Min	Max	Notes	Min	Max	Notes
$\alpha$	0°	10°		0°	10°	
A	–	5.080		–	0.200	
A1	0.381	–		0.015	–	
A2	3.175	4.064		0.125	0.160	
B	0.355	0.559		0.014	0.022	
B1	1.270	1.778	<b>Typical</b>	0.050	0.070	<b>Typical</b>
C	0.203	0.381	<b>Typical</b>	0.008	0.015	<b>Typical</b>
D	51.181	52.197		2.015	2.055	
D1	48.260	48.260	<b>Reference</b>	1.900	1.900	<b>Reference</b>
E	15.240	15.875		0.600	0.625	
E1	13.462	13.970		0.530	0.550	
e1	2.489	2.591	<b>Typical</b>	0.098	0.102	<b>Typical</b>
eA	15.240	15.240	<b>Reference</b>	0.600	0.600	<b>Reference</b>
eB	15.240	17.272		0.600	0.680	
L	2.921	3.683		0.115	0.145	
N	40	40		40	40	
S	1.270	–		0.050	–	
S1	0.508	–		0.020	–	

# PIC16C7X

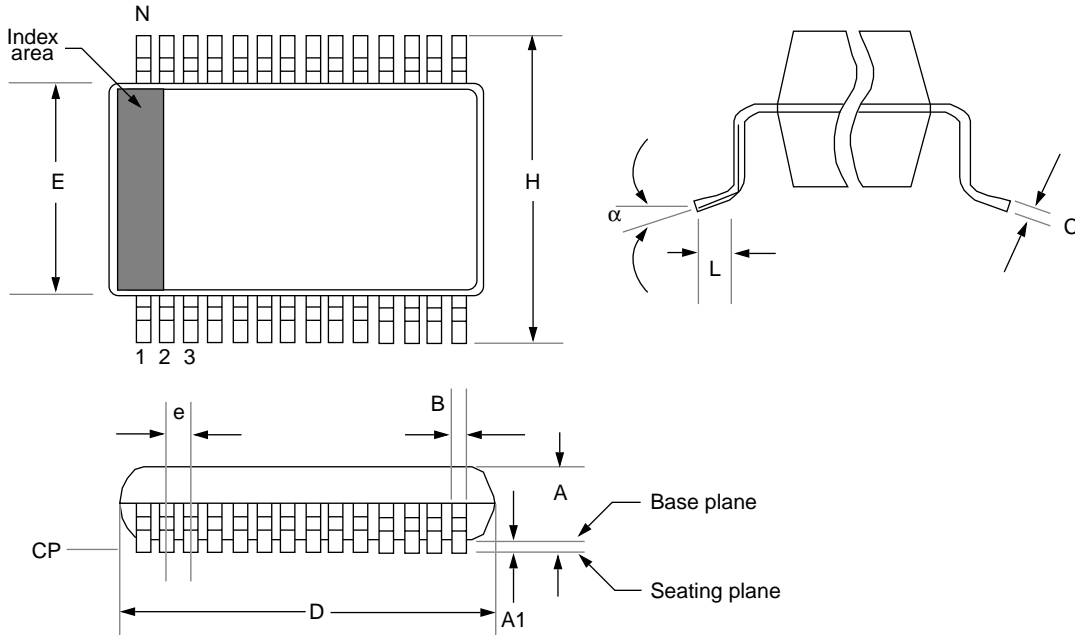
## 22.5 28-Lead Plastic Surface Mount (SOIC - Wide, 300 mil Body) (SO)



Package Group: Plastic SOIC (SO)						
Symbol	Millimeters			Inches		
	Min	Max	Notes	Min	Max	Notes
$\alpha$	0°	8°		0°	8°	
A	2.362	2.642		0.093	0.104	
A1	0.101	0.300		0.004	0.012	
B	0.355	0.483		0.014	0.019	
C	0.241	0.318		0.009	0.013	
D	17.703	18.085		0.697	0.712	
E	7.416	7.595		0.292	0.299	
e	1.270	1.270	Typical	0.050	0.050	Typical
H	10.007	10.643		0.394	0.419	
h	0.381	0.762		0.015	0.030	
L	0.406	1.143		0.016	0.045	
N	28	28		28	28	
CP	–	0.102		–	0.004	

# PIC16C7X

## 22.6 28-Lead Plastic Surface Mount (SSOP - 209 mil Body 5.30 mm) (SS)

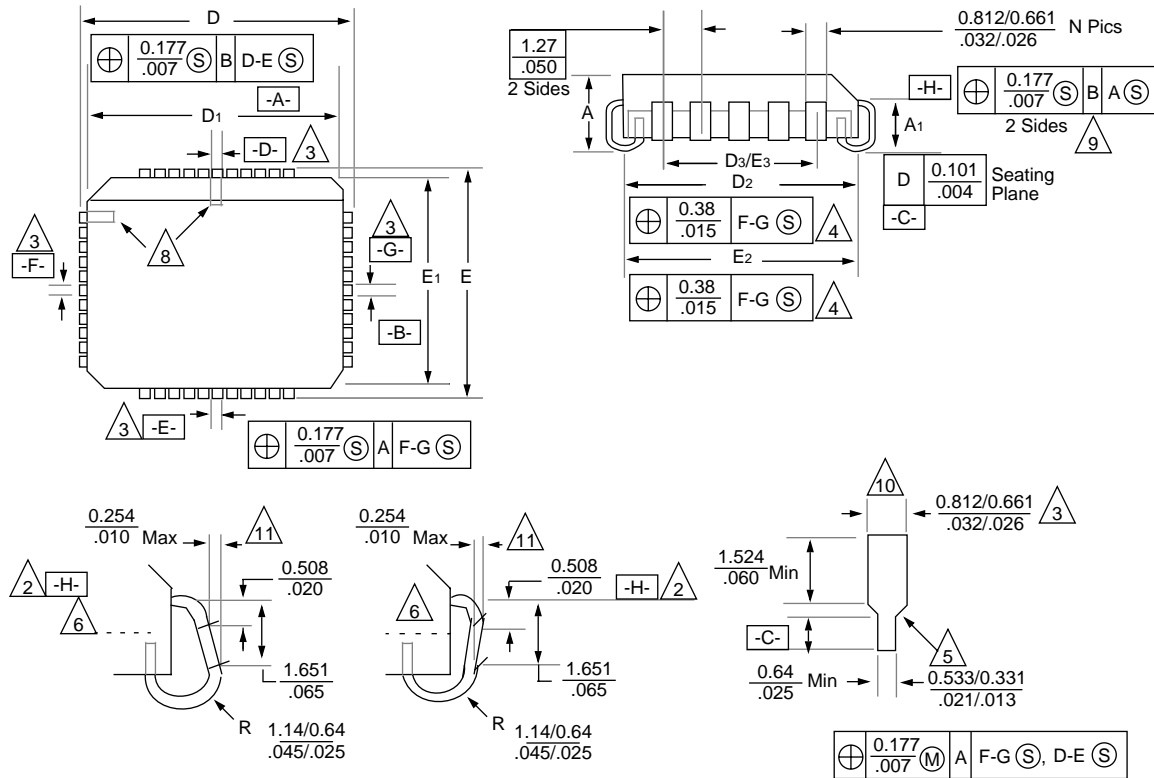


Package Group: Plastic SSOP						
Symbol	Millimeters			Inches		
	Min	Max	Notes	Min	Max	Notes
$\alpha$	0°	8°		0°	8°	
A	1.730	1.990		0.068	0.078	
A1	0.050	0.210		0.002	0.008	
B	0.250	0.380		0.010	0.015	
C	0.130	0.220		0.005	0.009	
D	10.070	10.330		0.396	0.407	
E	5.200	5.380		0.205	0.212	
e	0.650	0.650	Reference	0.026	0.026	Reference
H	7.650	7.900		0.301	0.311	
L	0.550	0.950		0.022	0.037	
N	28	28		28	28	
CP	-	0.102		-	0.004	



# PIC16C7X

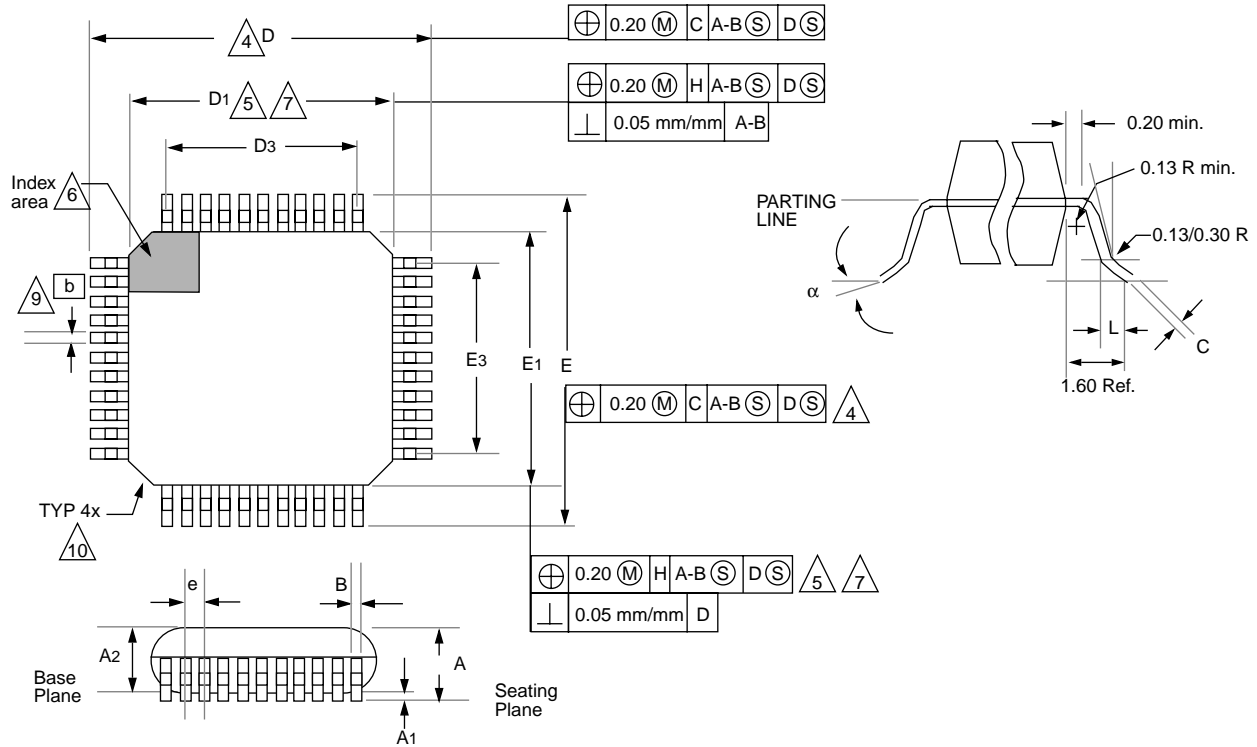
## 22.7 44-Lead Plastic Leaded Chip Carrier (Square)(PLCC)



Package Group: Plastic Leaded Chip Carrier (PLCC)						
Symbol	Millimeters			Inches		
	Min	Max	Notes	Min	Max	Notes
A	4.191	4.572		0.165	0.180	
A1	2.413	2.921		0.095	0.115	
D	17.399	17.653		0.685	0.695	
D1	16.510	16.663		0.650	0.656	
D2	15.494	16.002		0.610	0.630	
D3	12.700	12.700	Reference	0.500	0.500	Reference
E	17.399	17.653		0.685	0.695	
E1	16.510	16.663		0.650	0.656	
E2	15.494	16.002		0.610	0.630	
E3	12.700	12.700	Reference	0.500	0.500	Reference
N	44	44		44	44	
CP	-	0.102		-	0.004	
LT	0.203	0.381		0.008	0.015	

# PIC16C7X

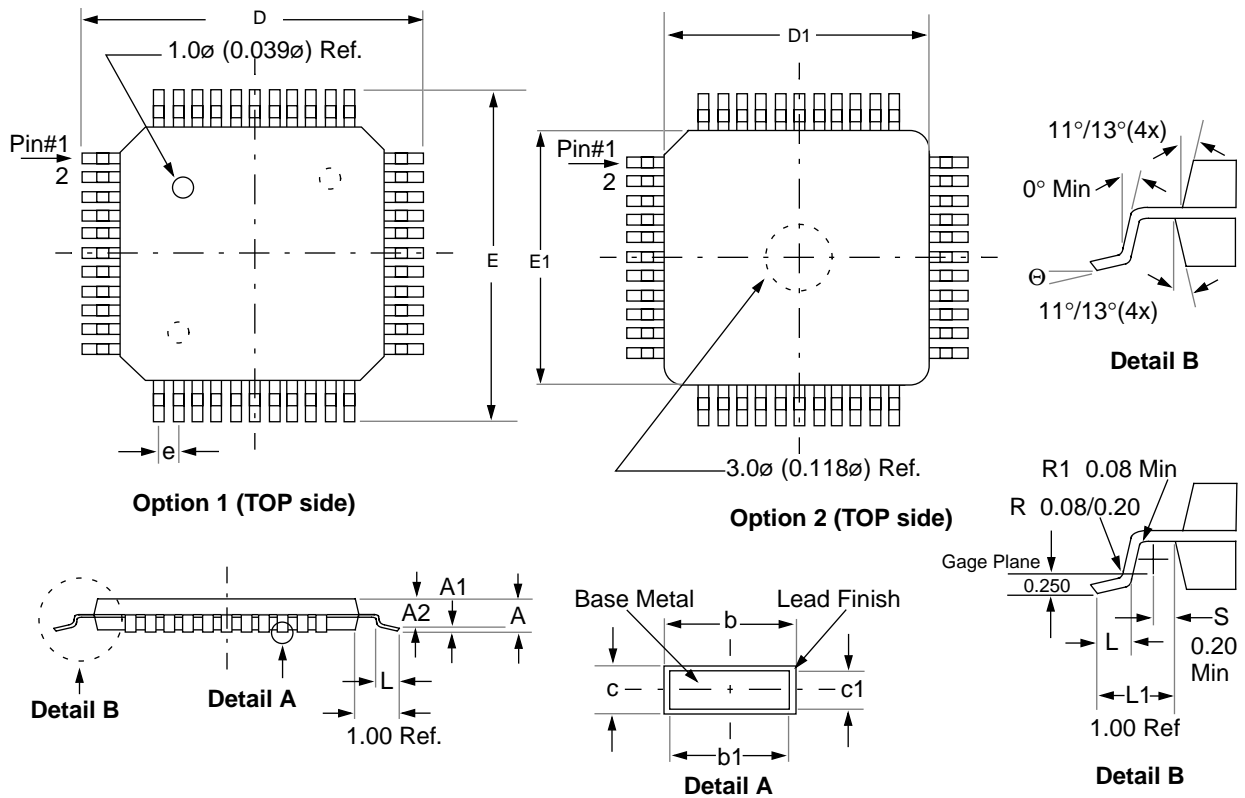
## 22.8 44-Lead Plastic Surface Mount (MQFP 10x10 mm Body 1.6/0.15 mm Lead Form) (PQ)



Package Group: Plastic MQFP						
Symbol	Millimeters			Inches		
	Min	Max	Notes	Min	Max	Notes
$\alpha$	0°	7°		0°	7°	
A	2.000	2.350		0.078	0.093	
A1	0.050	0.250		0.002	0.010	
A2	1.950	2.100		0.768	0.083	
b	0.300	0.450	Typical	0.011	0.018	Typical
C	0.150	0.180		0.006	0.007	
D	12.950	13.450		0.510	0.530	
D1	9.900	10.100		0.390	0.398	
D3	8.000	8.000	Reference	0.315	0.315	Reference
E	12.950	13.450		0.510	0.530	
E1	9.900	10.100		0.390	0.398	
E3	8.000	8.000	Reference	0.315	0.315	Reference
e	0.800	0.800		0.031	0.032	
L	0.730	1.030		0.028	0.041	
N	44	44		44	44	
CP	0.102	-		0.004	-	

# PIC16C7X

## 22.9 44-Lead Plastic Surface Mount (TQFP 10x10 mm Body 1.0/0.10 mm Lead Form) (TQ)



Package Group: Plastic TQFP						
Symbol	Millimeters			Inches		
	Min	Max	Notes	Min	Max	Notes
A	1.00	1.20		0.039	0.047	
A1	0.05	0.15		0.002	0.006	
A2	0.95	1.05		0.037	0.041	
D	11.75	12.25		0.463	0.482	
D1	9.90	10.10		0.390	0.398	
E	11.75	12.25		0.463	0.482	
E1	9.90	10.10		0.390	0.398	
L	0.45	0.75		0.018	0.030	
e	0.80 BSC			0.031 BSC		
b	0.30	0.45		0.012	0.018	
b1	0.30	0.40		0.012	0.016	
c	0.09	0.20		0.004	0.008	
c1	0.09	0.16		0.004	0.006	
N	44	44		44	44	
θ	0°	7°		0°	7°	

Note 1: Dimensions D1 and E1 do not include mold protrusion. Allowable mold protrusion is 0.25m/m (0.010") per side. D1 and E1 dimensions including mold mismatch.

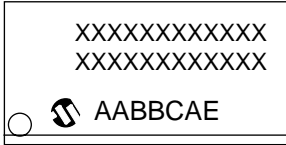
2: Dimension "b" does not include Dambar protrusion, allowable Dambar protrusion shall be 0.08m/m (0.003")max.

3: This outline conforms to JEDEC MS-026.

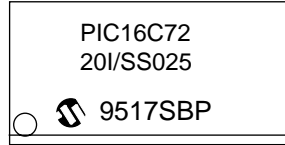
# PIC16C7X

## 22.10 Package Marking Information

### 28-Lead SSOP



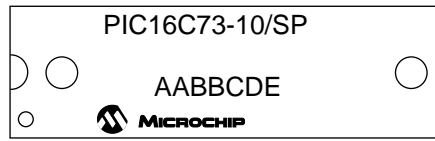
### Example



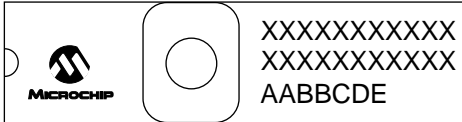
### 28-Lead PDIP (Skinny DIP)



### Example



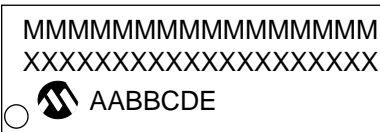
### 28-Lead Side Brazed Skinny Windowed



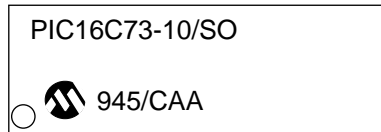
### Example



### 28-Lead SOIC



### Example



<b>Legend:</b>	MM...M	Microchip part number information
	XX...X	Customer specific information*
	AA	Year code (last 2 digits of calendar year)
	BB	Week code (week of January 1 is week '01')
	C	Facility code of the plant at which wafer is manufactured. C = Chandler, Arizona, U.S.A. S = Tempe, Arizona, U.S.A.
	D <sub>1</sub>	Mask revision number for microcontroller
	E	Assembly code of the plant or country of origin in which part was assembled.
<b>Note:</b>	In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line thus limiting the number of available characters for customer specific information.	

\* Standard OTP marking consists of Microchip part number, year code, week code, facility code, mask revision number, and assembly code. For OTP 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.

# PIC16C7X

## Package Marking Information (Cont'd)

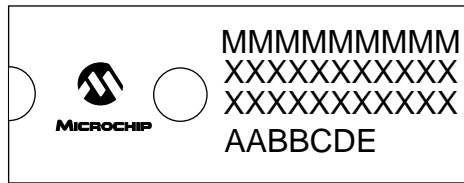
### 40-Lead PDIP



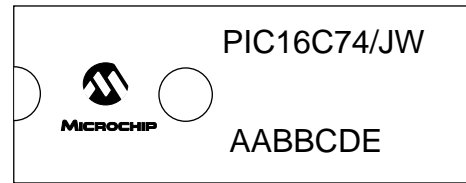
### Example



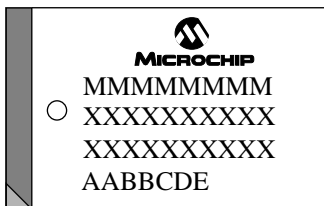
### 40-Lead CERDIP Windowed



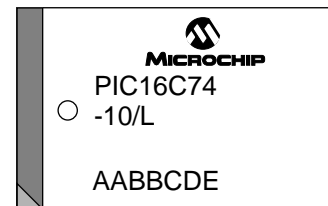
### Example



### 44-Lead PLCC



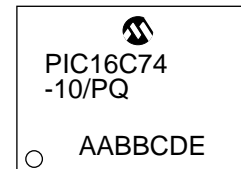
### Example



### 44-Lead MQFP



### Example



<b>Legend:</b>	MM...M	Microchip part number information
	XX...X	Customer specific information*
	AA	Year code (last 2 digits of calendar year)
	BB	Week code (week of January 1 is week '01')
	C	Facility code of the plant at which wafer is manufactured. C = Chandler, Arizona, U.S.A. S = Tempe, Arizona, U.S.A.
	D <sub>1</sub>	Mask revision number for microcontroller
	E	Assembly code of the plant or country of origin in which part was assembled.

**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.

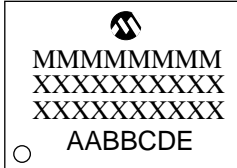
\* Standard OTP marking consists of Microchip part number, year code, week code, facility code, mask revision number, and assembly code. For OTP marking beyond this, certain price adders apply. Please check with your Microchip Sales Office. For QTP devices, any special marking adders are included in QTP price.

# PIC16C7X

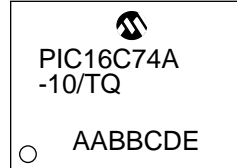
---

## Package Marking Information (Cont'd)

44-Lead TQFP



Example



<b>Legend:</b>	MM...M	Microchip part number information
	XX...X	Customer specific information*
	AA	Year code (last 2 digits of calendar year)
	BB	Week code (week of January 1 is week '01')
	C	Facility code of the plant at which wafer is manufactured. C = Chandler, Arizona, U.S.A. S = Tempe, Arizona, U.S.A.
	D <sub>1</sub>	Mask revision number for microcontroller
	E	Assembly code of the plant or country of origin in which part was assembled.
<b>Note:</b>	In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line thus limiting the number of available characters for customer specific information.	

\* Standard OTP marking consists of Microchip part number, year code, week code, facility code, mask revision number, and assembly code. For OTP 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.

## APPENDIX A:

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

1. Instruction word length is increased to 14-bits. This allows larger page sizes both in program memory (2K now as opposed to 512 before) and register file (128 bytes now versus 32 bytes before).
2. A PC high latch register (PCLATH) is added to handle program memory paging. Bits PA2, PA1, PA0 are removed from STATUS register.
3. Data memory paging is redefined slightly. STATUS register is modified.
4. Four new instructions have been added: RETURN, RETFIE, ADDLW, and SUBLW. Two instructions TRIS and OPTION are being phased out although they are kept for compatibility with PIC16C5X.
5. OPTION and TRIS registers are made addressable.
6. Interrupt capability is added. Interrupt vector is at 0004h.
7. Stack size is increased to 8 deep.
8. Reset vector is changed to 0000h.
9. Reset of all registers is revisited. Five different reset (and wake-up) types are recognized. Registers are reset differently.
10. Wake up from SLEEP through interrupt is added.
11. Two separate timers, Oscillator Start-up Timer (OST) and Power-up Timer (PWRT) are included for more reliable power-up. These timers are invoked selectively to avoid unnecessary delays on power-up and wake-up.
12. PORTB has weak pull-ups and interrupt on change feature.
13. T0CKI pin is also a port pin (RA4) now.
14. FSR is made a full eight bit register.
15. "In-circuit serial programming" is made possible. The user can program PIC16CXX devices using only five pins: VDD, VSS, MCLR/VPP, RB6 (clock) and RB7 (data in/out).
16. PCON status register is added with a Power-on Reset status bit (POR).
17. Code protection scheme is enhanced such that portions of the program memory can be protected, while the remainder is unprotected.
18. Brown-out protection circuitry has been added. Controlled by configuration word bit BODEN. Brown-out reset ensures the device is placed in a reset condition if VDD dips below a fixed set-point.

## APPENDIX B: COMPATIBILITY

To convert code written for PIC16C5X to PIC16CXX, the user should take the following steps:

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

# PIC16C7X

---

## APPENDIX C: WHAT'S NEW

Added the following devices:

- PIC16C76
- PIC16C77

Removed the PIC16C710, PIC16C71, PIC16C711 from this datasheet.

Added PIC16C76 and PIC16C77 devices. The PIC16C76/77 devices have 368 bytes of data memory distributed in 4 banks and 8K of program memory in 4 pages. These two devices have an enhanced SPI that supports both clock phase and polarity. The USART has been enhanced.

When upgrading to the PIC16C76/77 please note that the upper 16 bytes of data memory in banks 1,2, and 3 are mapped into bank 0. This may require relocation of data memory usage in the user application code.

Added Q-cycle definitions to the Instruction Set Summary section.

## APPENDIX D: WHAT'S CHANGED

Minor changes, spelling and grammatical changes.

Added the following note to the USART section. This note applies to all devices except the PIC16C76 and PIC16C77.

For the PIC16C73/73A/74/74A the asynchronous high speed mode (BRGH = 1) may experience a high rate of receive errors. It is recommended that BRGH = 0. If you desire a higher baud rate than BRGH = 0 can support, refer to the device errata for additional information or use the PIC16C76/77.

Divided SPI section into SPI for the PIC16C76/77 and SPI for all other devices.



## APPENDIX E: PIC16/17 MICROCONTROLLERS

### E.1 PIC12CXXX Family of Devices

		PIC12C508	PIC12C509	PIC12C671	PIC12C672
<b>Clock</b>	Maximum Frequency of Operation (MHz)	4	4	4	4
	<b>Memory</b>	EPROM Program Memory	512 x 12	1024 x 12	1024 x 14
<b>Peripherals</b>	Data Memory (bytes)	25	41	128	128
	Timer Module(s)	TMRO	TMRO	TMRO	TMRO
<b>Features</b>	A/D Converter (8-bit) Channels	—	—	4	4
	Wake-up from SLEEP on pin change	Yes	Yes	Yes	Yes
	I/O Pins	5	5	5	5
	Input Pins	1	1	1	1
	Internal Pull-ups	Yes	Yes	Yes	Yes
	Voltage Range (Volts)	2.5-5.5	2.5-5.5	2.5-5.5	2.5-5.5
	In-Circuit Serial Programming	Yes	Yes	Yes	Yes
	Number of Instructions	33	33	35	35
Packages	8-pin DIP, SOIC	8-pin DIP, SOIC	8-pin DIP, SOIC	8-pin DIP, SOIC	

All PIC12C5XX devices have Power-on Reset, selectable Watchdog Timer, selectable code protect and high I/O current capability. All PIC12C5XX devices use serial programming with data pin GP1 and clock pin GP0.

### E.2 PIC14C000 Family of Devices

		PIC14C000
<b>Clock</b>	Maximum Frequency of Operation (MHz)	20
<b>Memory</b>	EPROM Program Memory (x14 words)	4K
	Data Memory (bytes)	192
	Timer Module(s)	TMRO ADTMR
<b>Peripherals</b>	Serial Port(s) (SPI/I <sup>2</sup> C, USART)	I <sup>2</sup> C with SMBus Support
<b>Features</b>	Slope A/D Converter Channels	8 External; 6 Internal
	Interrupt Sources	11
	I/O Pins	22
	Voltage Range (Volts)	2.7-6.0
	In-Circuit Serial Programming	Yes
	Additional On-chip Features	Internal 4MHz Oscillator, Bandgap Reference, Temperature Sensor, Calibration Factors, Low Voltage Detector, SLEEP, HIBERNATE, Comparators with Programmable References (2)
Packages	28-pin DIP (.300 mil), SOIC, SSOP	

# PIC16C7X

## E.3 PIC16C15X Family of Devices

		PIC16C154	PIC16CR154	PIC16C156	PIC16CR156	PIC16C158	PIC16CR158
<b>Clock</b>	Maximum Frequency of Operation (MHz)	20	20	20	20	20	20
<b>Memory</b>	EPROM Program Memory (x12 words)	512	—	1K	—	2K	—
	ROM Program Memory (x12 words)	—	512	—	1K	—	2K
	RAM Data Memory (bytes)	25	25	25	25	73	73
<b>Peripherals</b>	Timer Module(s)	TMR0	TMR0	TMR0	TMR0	TMR0	TMR0
<b>Features</b>	I/O Pins	12	12	12	12	12	12
	Voltage Range (Volts)	3.0-5.5	2.5-5.5	3.0-5.5	2.5-5.5	3.0-5.5	2.5-5.5
	Number of Instructions	33	33	33	33	33	33
	Packages	18-pin DIP, SOIC; 20-pin SSOP	18-pin DIP, SOIC; 20-pin SSOP	18-pin DIP, SOIC; 20-pin SSOP	18-pin DIP, SOIC; 20-pin SSOP	18-pin DIP, SOIC; 20-pin SSOP	18-pin DIP, SOIC; 20-pin SSOP

All PIC16/17 Family devices have Power-on Reset, selectable Watchdog Timer, selectable code protect and high I/O current capability.

## E.4 PIC16C5X Family of Devices

		PIC16C52	PIC16C54	PIC16C54A	PIC16CR54A	PIC16C55	PIC16C56
<b>Clock</b>	Maximum Frequency of Operation (MHz)	4	20	20	20	20	20
<b>Memory</b>	EPROM Program Memory (x12 words)	384	512	512	—	512	1K
	ROM Program Memory (x12 words)	—	—	—	512	—	—
	RAM Data Memory (bytes)	25	25	25	25	24	25
<b>Peripherals</b>	Timer Module(s)	TMR0	TMR0	TMR0	TMR0	TMR0	TMR0
<b>Features</b>	I/O Pins	12	12	12	12	20	12
	Voltage Range (Volts)	2.5-6.25	2.5-6.25	2.0-6.25	2.0-6.25	2.5-6.25	2.5-6.25
	Number of Instructions	33	33	33	33	33	33
	Packages	18-pin DIP, SOIC	18-pin DIP, SOIC; 20-pin SSOP	18-pin DIP, SOIC; 20-pin SSOP	18-pin DIP, SOIC; 20-pin SSOP	28-pin DIP, SOIC, SSOP	18-pin DIP, SOIC; 20-pin SSOP

		PIC16C57	PIC16CR57B	PIC16C58A	PIC16CR58A
<b>Clock</b>	Maximum Frequency of Operation (MHz)	20	20	20	20
<b>Memory</b>	EPROM Program Memory (x12 words)	2K	—	2K	—
	ROM Program Memory (x12 words)	—	2K	—	2K
	RAM Data Memory (bytes)	72	72	73	73
<b>Peripherals</b>	Timer Module(s)	TMR0	TMR0	TMR0	TMR0
<b>Features</b>	I/O Pins	20	20	12	12
	Voltage Range (Volts)	2.5-6.25	2.5-6.25	2.0-6.25	2.5-6.25
	Number of Instructions	33	33	33	33
	Packages	28-pin DIP, SOIC, SSOP	28-pin DIP, SOIC, SSOP	18-pin DIP, SOIC; 20-pin SSOP	18-pin DIP, SOIC; 20-pin SSOP

All PIC16/17 Family devices have Power-on Reset, selectable Watchdog Timer (except PIC16C52), selectable code protect and high I/O current capability.

# PIC16C7X

## E.5 PIC16C55X Family of Devices

		PIC16C554	PIC16C556 <sup>(1)</sup>	PIC16C558
<b>Clock</b>	Maximum Frequency of Operation (MHz)	20	20	20
<b>Memory</b>	EPROM Program Memory (x14 words)	512	1K	2K
	Data Memory (bytes)	80	80	128
<b>Peripherals</b>	Timer Module(s)	TMR0	TMR0	TMR0
	Comparators(s)	—	—	—
	Internal Reference Voltage	—	—	—
<b>Features</b>	Interrupt Sources	3	3	3
	I/O Pins	13	13	13
	Voltage Range (Volts)	2.5-6.0	2.5-6.0	2.5-6.0
	Brown-out Reset	—	—	—
	Packages	18-pin DIP, SOIC; 20-pin SSOP	18-pin DIP, SOIC; 20-pin SSOP	18-pin DIP, SOIC; 20-pin SSOP

All PIC16/17 Family devices have Power-on Reset, selectable Watchdog Timer, selectable code protect and high I/O current capability. All PIC16C5XX Family devices use serial programming with clock pin RB6 and data pin RB7.  
 Note 1: Please contact your local Microchip sales office for availability of these devices.

## E.6 PIC16C62X and PIC16C64X Family of Devices

		PIC16C620	PIC16C621	PIC16C622	PIC16C642	PIC16C662
<b>Clock</b>	Maximum Frequency of Operation (MHz)	20	20	20	20	20
<b>Memory</b>	EPROM Program Memory (x14 words)	512	1K	2K	4K	4K
	Data Memory (bytes)	80	80	128	176	176
<b>Peripherals</b>	Timer Module(s)	TMR0	TMR0	TMR0	TMR0	TMR0
	Comparators(s)	2	2	2	2	2
	Internal Reference Voltage	Yes	Yes	Yes	Yes	Yes
<b>Features</b>	Interrupt Sources	4	4	4	4	5
	I/O Pins	13	13	13	22	33
	Voltage Range (Volts)	2.5-6.0	2.5-6.0	2.5-6.0	3.0-6.0	3.0-6.0
	Brown-out Reset	Yes	Yes	Yes	Yes	Yes
	Packages	18-pin DIP, SOIC; 20-pin SSOP	18-pin DIP, SOIC; 20-pin SSOP	18-pin DIP, SOIC; 20-pin SSOP	28-pin PDIP, SOIC, Windowed CDIP	40-pin PDIP, Windowed CDIP; 44-pin PLCC, MQFP

All PIC16/17 Family devices have Power-on Reset, selectable Watchdog Timer, selectable code protect and high I/O current capability. All PIC16C62X and PIC16C64X Family devices use serial programming with clock pin RB6 and data pin RB7.

# PIC16C7X

## E.7 PIC16C6X Family of Devices

		PIC16C61	PIC16C62A	PIC16CR62	PIC16C63	PIC16CR63
<b>Clock</b>	Maximum Frequency of Operation (MHz)	20	20	20	20	20
	<b>Memory</b>					
	EPROM Program Memory (x14 words)	1K	2K	—	4K	—
	ROM Program Memory (x14 words)	—	—	2K	—	4K
	Data Memory (bytes)	36	128	128	192	192
<b>Peripherals</b>	Timer Module(s)	TMR0	TMR0, TMR1, TMR2	TMR0, TMR1, TMR2	TMR0, TMR1, TMR2	TMR0, TMR1, TMR2
	Capture/Compare/PWM Module(s)	—	1	1	2	2
	Serial Port(s) (SPI/I <sup>2</sup> C, USART)	—	SPI/I <sup>2</sup> C	SPI/I <sup>2</sup> C	SPI/I <sup>2</sup> C, USART	SPI/I <sup>2</sup> C, USART
	Parallel Slave Port	—	—	—	—	—
<b>Features</b>	Interrupt Sources	3	7	7	10	10
	I/O Pins	13	22	22	22	22
	Voltage Range (Volts)	3.0-6.0	2.5-6.0	2.5-6.0	2.5-6.0	2.5-6.0
	In-Circuit Serial Programming	Yes	Yes	Yes	Yes	Yes
	Brown-out Reset	—	Yes	Yes	Yes	Yes
	Packages	18-pin DIP, SO	28-pin SDIP, SOIC, SSOP	28-pin SDIP, SOIC, SSOP	28-pin SDIP, SOIC	28-pin SDIP, SOIC

		PIC16C64A	PIC16CR64	PIC16C65A	PIC16CR65	PIC16C66	PIC16C67
<b>Clock</b>	Maximum Frequency of Operation (MHz)	20	20	20	20	20	20
	<b>Memory</b>						
	EPROM Program Memory (x14 words)	2K	—	4K	—	8K	8K
	ROM Program Memory (x14 words)	—	2K	—	4K	—	—
	Data Memory (bytes)	128	128	192	192	368	368
<b>Peripherals</b>	Timer Module(s)	TMR0, TMR1, TMR2	TMR0, TMR1, TMR2	TMR0, TMR1, TMR2	TMR0, TMR1, TMR2	TMR0, TMR1, TMR2	TMR0, TMR1, TMR2
	Capture/Compare/PWM Module(s)	1	1	2	2	2	2
	Serial Port(s) (SPI/I <sup>2</sup> C, USART)	SPI/I <sup>2</sup> C	SPI/I <sup>2</sup> C	SPI/I <sup>2</sup> C, USART	SPI/I <sup>2</sup> C, USART	SPI/I <sup>2</sup> C, USART	SPI/I <sup>2</sup> C, USART
	Parallel Slave Port	Yes	Yes	Yes	Yes	—	Yes
<b>Features</b>	Interrupt Sources	8	8	11	11	10	11
	I/O Pins	33	33	33	33	22	33
	Voltage Range (Volts)	2.5-6.0	2.5-6.0	2.5-6.0	2.5-6.0	2.5-6.0	2.5-6.0
	In-Circuit Serial Programming	Yes	Yes	Yes	Yes	Yes	Yes
	Brown-out Reset	Yes	Yes	Yes	Yes	Yes	Yes
	Packages	40-pin DIP; 44-pin PLCC, MQFP, TQFP	40-pin DIP; 44-pin PLCC, MQFP, TQFP	40-pin DIP; 44-pin PLCC, MQFP, TQFP	40-pin DIP; 44-pin PLCC, MQFP, TQFP	28-pin SDIP, SOIC	40-pin DIP; 44-pin PLCC, MQFP, TQFP

All PIC16/17 Family devices have Power-on Reset, selectable Watchdog Timer, selectable code protect and high I/O current capability. All PIC16C6X Family devices use serial programming with clock pin RB6 and data pin RB7.

# PIC16C7X

## E.8 PIC16C8X Family of Devices

		PIC16F83	PIC16CR83	PIC16F84	PIC16CR84
<b>Clock</b>	Maximum Frequency of Operation (MHz)	10	10	10	10
	<b>Memory</b>				
	Flash Program Memory	512	—	1K	—
	EEPROM Program Memory	—	—	—	—
	ROM Program Memory	—	512	—	1K
	Data Memory (bytes)	36	36	68	68
	Data EEPROM (bytes)	64	64	64	64
<b>Peripherals</b>	Timer Module(s)	TMR0	TMR0	TMR0	TMR0
	<b>Features</b>				
	Interrupt Sources	4	4	4	4
	I/O Pins	13	13	13	13
	Voltage Range (Volts)	2.0-6.0	2.0-6.0	2.0-6.0	2.0-6.0
	Packages	18-pin DIP, SOIC	18-pin DIP, SOIC	18-pin DIP, SOIC	18-pin DIP, SOIC

All PIC16/17 Family devices have Power-on Reset, selectable Watchdog Timer, selectable code protect and high I/O current capability. All PIC16C8X Family devices use serial programming with clock pin RB6 and data pin RB7.

## E.9 PIC16C9XX Family Of Devices

		PIC16C923	PIC16C924
<b>Clock</b>	Maximum Frequency of Operation (MHz)	8	8
	<b>Memory</b>		
	EPROM Program Memory	4K	4K
	Data Memory (bytes)	176	176
<b>Peripherals</b>	Timer Module(s)	TMR0, TMR1, TMR2	TMR0, TMR1, TMR2
	Capture/Compare/PWM Module(s)	1	1
	Serial Port(s) (SPI/I <sup>2</sup> C, USART)	SPI/I <sup>2</sup> C	SPI/I <sup>2</sup> C
	Parallel Slave Port	—	—
	A/D Converter (8-bit) Channels	—	5
	LCD Module	4 Com, 32 Seg	4 Com, 32 Seg
<b>Features</b>	Interrupt Sources	8	9
	I/O Pins	25	25
	Input Pins	27	27
	Voltage Range (Volts)	3.0-6.0	3.0-6.0
	In-Circuit Serial Programming	Yes	Yes
	Brown-out Reset	—	—
	Packages	64-pin SDIP <sup>(1)</sup> , TQFP; 68-pin PLCC, Die	64-pin SDIP <sup>(1)</sup> , TQFP; 68-pin PLCC, Die

All PIC16/17 Family devices have Power-on Reset, selectable Watchdog Timer, selectable code protect and high I/O current capability. All PIC16C9XX Family devices use serial programming with clock pin RB6 and data pin RB7.

# PIC16C7X

## E.10 PIC17CXXX Family of Devices

		PIC17C42A	PIC17CR42	PIC17C43	PIC17CR43	PIC17C44
<b>Clock</b>	Maximum Frequency of Operation (MHz)	33	33	33	33	33
	<b>Memory</b>					
	EPROM Program Memory (words)	2K	—	4K	—	8K
	ROM Program Memory (words)	—	2K	—	4K	—
	RAM Data Memory (bytes)	232	232	454	454	454
<b>Peripherals</b>	Timer Module(s)	TMR0, TMR1, TMR2, TMR3	TMR0, TMR1, TMR2, TMR3	TMR0, TMR1, TMR2, TMR3	TMR0, TMR1, TMR2, TMR3	TMR0, TMR1, TMR2, TMR3
	Captures/PWM Module(s)	2	2	2	2	2
	Serial Port(s) (USART)	Yes	Yes	Yes	Yes	Yes
<b>Features</b>	Hardware Multiply	Yes	Yes	Yes	Yes	Yes
	External Interrupts	Yes	Yes	Yes	Yes	Yes
	Interrupt Sources	11	11	11	11	11
	I/O Pins	33	33	33	33	33
	Voltage Range (Volts)	2.5-6.0	2.5-6.0	2.5-6.0	2.5-6.0	2.5-6.0
	Number of Instructions	58	58	58	58	58
	Packages	40-pin DIP; 44-pin PLCC, MQFP, TQFP	40-pin DIP; 44-pin PLCC, MQFP, TQFP	40-pin DIP; 44-pin PLCC, MQFP, TQFP	40-pin DIP; 44-pin PLCC, MQFP, TQFP	40-pin DIP; 44-pin PLCC, MQFP, TQFP

		PIC17C752	PIC17C756
<b>Clock</b>	Maximum Frequency of Operation (MHz)	33	33
	<b>Memory</b>		
	EPROM Program Memory (words)	8K	16K
	ROM Program Memory (words)	—	—
	RAM Data Memory (bytes)	454	902
<b>Peripherals</b>	Timer Module(s)	TMR0, TMR1, TMR2, TMR3	TMR0, TMR1, TMR2, TMR3
	Captures/PWM Module(s)	4/3	4/3
	Serial Port(s) (USART)	2	2
<b>Features</b>	Hardware Multiply	Yes	Yes
	External Interrupts	Yes	Yes
	Interrupt Sources	18	18
	I/O Pins	50	50
	Voltage Range (Volts)	3.0-6.0	3.0-6.0
	Number of Instructions	58	58
	Packages	64-pin DIP; 68-pin LCC, 68-pin TQFP	64-pin DIP; 68-pin LCC, 68-pin TQFP

All PIC16/17 Family devices have Power-on Reset, selectable Watchdog Timer, selectable code protect and high I/O current capability.

## PIN COMPATIBILITY

Devices that have the same package type and VDD, VSS and MCLR pin locations are said to be pin compatible. This allows these different devices to operate in the same socket. Compatible devices may only require minor software modification to allow proper operation in the application socket (ex., PIC16C56 and PIC16C61 devices). Not all devices in the same package size are pin compatible; for example, the PIC16C62 is compatible with the PIC16C63, but not the PIC16C55.

Pin compatibility does not mean that the devices offer the same features. As an example, the PIC16C54 is pin compatible with the PIC16C71, but does not have an A/D converter, weak pull-ups on PORTB, or interrupts.

**TABLE E-1: PIN COMPATIBLE DEVICES**

Pin Compatible Devices	Package
PIC12C508, PIC12C509, PIC12C671, PIC12C672	8-pin
PIC16C154, PIC16CR154, PIC16C156, PIC16CR156, PIC16C158, PIC16CR158, PIC16C52, PIC16C54, PIC16C54A, PIC16CR54A, PIC16C56, PIC16C58A, PIC16CR58A, PIC16C61, PIC16C554, PIC16C556, PIC16C558 PIC16C620, PIC16C621, PIC16C622 PIC16C641, PIC16C642, PIC16C661, PIC16C662 PIC16C710, PIC16C711, PIC16C715 PIC16F83, PIC16CR83, PIC16F84A, PIC16CR84	18-pin, 20-pin
PIC16C55, PIC16C57, PIC16CR57B	28-pin
PIC16CR62, PIC16C62A, PIC16C63, PIC16CR63, PIC16C66, PIC16C72, PIC16C73A, PIC16C76	28-pin
PIC16CR64, PIC16C64A, PIC16C65A, PIC16CR65, PIC16C67, PIC16C74A, PIC16C77	40-pin
PIC17CR42, PIC17C42A, PIC17C43, PIC17CR43, PIC17C44	40-pin
PIC16C923, PIC16C924	64/68-pin
PIC17C756, PIC17C752	64/68-pin

# PIC16C7X

---

---

NOTES:



# PIC16C7X

## INDEX

### A

#### A/D

Accuracy/Error .....	124
ADCON0 Register .....	117
ADCON1 Register .....	118
ADIF bit .....	119
Analog Input Model Block Diagram .....	120
Analog-to-Digital Converter .....	117
Block Diagram .....	119
Configuring Analog Port Pins .....	121
Configuring the Interrupt .....	119
Configuring the Module .....	119
Connection Considerations .....	125
Conversion Clock .....	121
Conversion Time .....	123
Conversions .....	122
Converter Characteristics .....	181, 199, 217, 238
Delays .....	120
Effects of a Reset .....	124
Equations .....	120
Faster Conversion - Lower Resolution Tradeoff .....	123
Flowchart of A/D Operation .....	126
GO/DONE bit .....	119
Internal Sampling Switch (R <sub>ss</sub> ) Impedance .....	120
Operation During Sleep .....	124
Sampling Requirements .....	120
Sampling Time .....	120
Source Impedance .....	120
Time Delays .....	120
Transfer Function .....	125
Using the CCP Trigger .....	125
Absolute Maximum Ratings .....	167, 183, 201, 219
ACK .....	90, 94, 95
ADIE bit .....	33
ADIF bit .....	35
ADRES Register .....	23, 25, 27, 117, 119
ALU .....	9
Application Notes	
AN546 (Using the Analog-to-Digital Converter) .....	117
AN552 (Implementing Wake-up on Key Strokes Using PIC16CXXX) .....	45
AN556 (Table Reading Using PIC16CXX) .....	40
AN578 (Use of the SSP Module in the I <sup>2</sup> C Multi-Master Environment) .....	77
AN594 (Using the CCP Modules) .....	71
AN607, Power-up Trouble Shooting .....	134
Architecture	
Harvard .....	9
Overview .....	9
von Neumann .....	9
Assembler	
MPASM Assembler .....	164

### B

Baud Rate Error .....	101
Baud Rate Formula .....	101
Baud Rates	
Asynchronous Mode .....	102
Synchronous Mode .....	102
BF .....	78, 83, 94
Block Diagrams	
A/D .....	119
Analog Input Model .....	120
Capture .....	72

Compare .....	73
I <sup>2</sup> C Mode .....	93
On-Chip Reset Circuit .....	133
PIC16C72 .....	10
PIC16C73 .....	11
PIC16C73A .....	11
PIC16C74 .....	12
PIC16C74A .....	12
PIC16C76 .....	11
PIC16C77 .....	12
PORTC .....	48
PORTD (In I/O Port Mode) .....	50
PORTD and PORTE as a Parallel Slave Port .....	54
PORTE (In I/O Port Mode) .....	51
PWM .....	74
RA3:RA0 and RA5 Port Pins .....	43
RA4/T0CKI Pin .....	43
RB3:RB0 Port Pins .....	45
RB7:RB4 Port Pins .....	46
SPI Master/Slave Connection .....	81
SSP in I <sup>2</sup> C Mode .....	93
SSP in SPI Mode .....	80, 85
Timer0 .....	59
Timer0/WDT Prescaler .....	62
Timer1 .....	66
Timer2 .....	69
USART Receive .....	108
USART Transmit .....	106
Watchdog Timer .....	144
BOR bit .....	39, 135
BRGH bit .....	101
Buffer Full Status bit, BF .....	78, 83
<b>C</b>	
C bit .....	30
C Compiler .....	165
Capture/Compare/PWM	
Capture	
Block Diagram .....	72
CCP1CON Register .....	72
CCP1IF .....	72
CCPR1 .....	72
CCPR1H:CCPR1L .....	72
Mode .....	72
Prescaler .....	73
CCP Timer Resources .....	71
Compare	
Block Diagram .....	73
Mode .....	73
Software Interrupt Mode .....	73
Special Event Trigger .....	73
Special Trigger Output of CCP1 .....	73
Special Trigger Output of CCP2 .....	73
Interaction of Two CCP Modules .....	71
Section .....	71
Special Event Trigger and A/D Conversions .....	73
Capture/Compare/PWM (CCP)	
PWM Block Diagram .....	74
PWM Mode .....	74
PWM, Example Frequencies/Resolutions .....	75
Carry bit .....	9
CCP1CON .....	29
CCP1IE bit .....	33
CCP1IF bit .....	35, 36
CCP2CON .....	29
CCP2IE bit .....	37

# PIC16C7X

CCP2IF bit .....	38
CCPR1H Register .....	25, 27, 29, 71
CCPR1L Register .....	29, 71
CCPR2H Register .....	25, 27, 29, 71
CCPR2L Register .....	25, 27, 29, 71
CCPxM0 bit .....	72
CCPxM1 bit .....	72
CCPxM2 bit .....	72
CCPxM3 bit .....	72
CCPxX bit .....	72
CCPxY bit .....	72
CKE .....	83
CKP .....	79, 84
Clock Polarity Select bit, CKP .....	79, 84
Clock Polarity, SPI Mode .....	81
Clocking Scheme .....	17
Code Examples	
Call of a Subroutine in Page 1 from Page 0 .....	41
Changing Between Capture Prescalers .....	73
Changing Prescaler (Timer0 to WDT) .....	63
Changing Prescaler (WDT to Timer0) .....	63
I/O Programming .....	53
Indirect Addressing .....	41
Initializing PORTA .....	43
Initializing PORTB .....	45
Initializing PORTC .....	48
Loading the SSPBUF Register .....	80, 85
Code Protection .....	129, 146
Computed GOTO .....	40
Configuration Bits .....	129
Configuration Word .....	129
Connecting Two Microcontrollers .....	81
CREN bit .....	100
$\overline{CS}$ pin .....	54
<b>D</b>	
D/ $\overline{A}$ .....	78, 83
Data/Address bit, D/ $\overline{A}$ .....	78, 83
DC bit .....	30
DC Characteristics	
PIC16C72 .....	168
PIC16C73 .....	184
PIC16C73A .....	202
PIC16C74 .....	184
PIC16C74A .....	202
PIC16C76 .....	221
PIC16C77 .....	221
Development Support .....	5, 163
Development Tools .....	163
Digit Carry bit .....	9
Direct Addressing .....	41
<b>E</b>	
Electrical Characteristics	
PIC16C72 .....	167
PIC16C73 .....	183
PIC16C73A .....	201
PIC16C74 .....	183
PIC16C74A .....	201
PIC16C76 .....	219
PIC16C77 .....	219
External Brown-out Protection Circuit .....	140
External Power-on Reset Circuit .....	140

## F

Family of Devices	
PIC12CXXX .....	265
PIC14C000 .....	265
PIC16C15X .....	266
PIC16C55X .....	267
PIC16C5X .....	266
PIC16C62X and PIC16C64X .....	267
PIC16C6X .....	268
PIC16C7XX .....	6
PIC16C8X .....	269
PIC16C9XX .....	269
PIC17CXX .....	270
FERR bit .....	100
FSR Register .....	23, 24, 25, 26, 27, 28, 29, 41
Fuzzy Logic Dev. System ( <i>fuzzyTECH</i> <sup>®</sup> -MP) .....	163, 165

## G

General Description .....	5
GIE bit .....	141

## I

I/O Ports	
PORTA .....	43
PORTB .....	45
PORTC .....	48
PORTD .....	50, 54
PORTE .....	51
Section .....	43
I/O Programming Considerations .....	53
I <sup>2</sup> C	
Addressing .....	94
Addressing I <sup>2</sup> C Devices .....	90
Arbitration .....	92
Block Diagram .....	93
Clock Synchronization .....	92
Combined Format .....	91
I <sup>2</sup> C Operation .....	93
I <sup>2</sup> C Overview .....	89
Initiating and Terminating Data Transfer .....	89
Master Mode .....	97
Master-Receiver Sequence .....	91
Master-Transmitter Sequence .....	91
Mode .....	93
Mode Selection .....	93
Multi-master .....	92
Multi-Master Mode .....	97
Reception .....	95
Reception Timing Diagram .....	95
SCL and SDA pins .....	94
Slave Mode .....	94
START .....	89
STOP .....	89, 90
Transfer Acknowledge .....	90
Transmission .....	96
IDLE_MODE .....	98
In-Circuit Serial Programming .....	129, 146
INDF .....	29
INDF Register .....	24, 25, 26, 27, 28, 41
Indirect Addressing .....	41
Initialization Condition for all Register .....	136
Instruction Cycle .....	17
Instruction Flow/Pipelining .....	17
Instruction Format .....	147

# PIC16C7X

## Instruction Set

ADDLW .....	149
ADDWF .....	149
ANDLW .....	149
ANDWF .....	149
BCF .....	150
BSF .....	150
BTFSC .....	150
BTFSS .....	151
CALL .....	151
CLRF .....	152
CLRWF .....	152
CLRWDW .....	152
COMF .....	153
DECF .....	153
DECFSZ .....	153
GOTO .....	154
INCF .....	154
INCFSZ .....	155
IORLW .....	155
IORWF .....	156
MOVF .....	156
MOVLW .....	156
MOVWF .....	156
NOP .....	157
OPTION .....	157
RETFIE .....	157
RETLW .....	158
RETURN .....	158
RLF .....	159
RRF .....	159
SLEEP .....	160
SUBLW .....	160
SUBWF .....	161
SWAPF .....	161
TRIS .....	161
XORLW .....	162
XORWF .....	162
Section .....	147
Summary Table .....	148
INT Interrupt .....	143
INTCON .....	29
INTCON Register .....	32
INTEDG bit .....	31, 143
Internal Sampling Switch (Rss) Impedance .....	120
Interrupts .....	129
PortB Change .....	143
RB7:RB4 Port Change .....	45
Section .....	141
TMR0 .....	143
IRP bit .....	30

## L

Loading of PC .....	40
---------------------	----

## M

MCLR .....	133, 136
Memory .....	
Data Memory .....	20
Program Memory .....	19
Program Memory Maps .....	
PIC16C72 .....	19
PIC16C73 .....	19
PIC16C73A .....	19
PIC16C74 .....	19
PIC16C74A .....	19
PIC16C76 .....	20
PIC16C77 .....	20
Register File Maps .....	
PIC16C72 .....	21
PIC16C73 .....	21
PIC16C73A .....	21
PIC16C74 .....	21
PIC16C74A .....	21
PIC16C76 .....	21
PIC16C77 .....	21
MPASM Assembler .....	163
MPLAB-C .....	165
MPSIM Software Simulator .....	163, 165

## O

OERR bit .....	100
OPCODE .....	147
OPTION .....	29
OPTION Register .....	31
Orthogonal .....	9
OSC selection .....	129
Oscillator .....	
HS .....	131, 135
LP .....	131, 135
RC .....	131
XT .....	131, 135
Oscillator Configurations .....	131
Output of TMR2 .....	69

## P

P .....	78, 83
Packaging .....	
28-Lead Ceramic w/Window .....	251
28-Lead PDIP .....	253
28-Lead SOIC .....	255
28-Lead SSOP .....	256
40-Lead CERDIP w/Window .....	252
40-Lead PDIP .....	254
44-Lead MQFP .....	258
44-Lead PLCC .....	257
44-Lead TQFP .....	259
Paging, Program Memory .....	40
Parallel Slave Port .....	50, 54
PCFG0 bit .....	118
PCFG1 bit .....	118
PCFG2 bit .....	118
PCL Register .....	23, 24, 25, 26, 27, 28, 29, 40
PCLATH .....	136
PCLATH Register .....	23, 24, 25, 26, 27, 28, 29, 40
PCON Register .....	29, 39, 135
$\overline{PD}$ bit .....	30, 133, 135
PICDEM-1 Low-Cost PIC16/17 Demo Board .....	163, 164
PICDEM-2 Low-Cost PIC16CXX Demo Board .....	163, 164
PICDEM-3 Low-Cost PIC16C9XXX Demo Board .....	164
PICMASTER In-Circuit Emulator .....	163

# PIC16C7X

PICSTART Low-Cost Development System .....	163	POR .....	134, 135
PIE1 Register .....	29, 33	Oscillator Start-up Timer (OST) .....	129, 134
PIE2 Register .....	29, 37	Power Control Register (PCON) .....	135
Pin Compatible Devices .....	271	Power-on Reset (POR) .....	129, 134, 136
Pin Functions		Power-up Timer (PWRT) .....	129, 134
MCLR/VPP .....	13, 14, 15	Power-Up-Timer (PWRT) .....	134
OSC1/CLKIN .....	13, 14, 15	Time-out Sequence .....	135
OSC2/CLKOUT .....	13, 14, 15	Time-out Sequence on Power-up .....	139
RA0/AN0 .....	13, 14, 15	TO .....	133, 135
RA1/AN1 .....	13, 14, 15	POR bit .....	39, 135
RA2/AN2 .....	13, 14, 15	Port RB Interrupt .....	143
RA3/AN3/VREF .....	13, 14, 15	PORTA .....	29, 136
RA4/T0CKI .....	13, 14, 15	PORTA Register .....	23, 25, 27, 43
RA5/AN4/SS .....	13, 14, 15	PORTB .....	29, 136
RB0/INT .....	13, 14, 15	PORTB Register .....	23, 25, 27, 45
RB1 .....	13, 14, 15	PORTC .....	29, 136
RB2 .....	13, 14, 15	PORTC Register .....	23, 25, 27, 48
RB3 .....	13, 14, 15	PORTD .....	29, 136
RB4 .....	13, 14, 15	PORTD Register .....	25, 27, 50
RB5 .....	13, 14, 15	PORTE .....	29, 136
RB6 .....	13, 14, 15	PORTE Register .....	25, 27, 51
RB7 .....	13, 14, 15	Power-down Mode (SLEEP) .....	145
RC0/T1OSO/T1CKI .....	13, 14, 16	PR2 .....	29
RC1/T1OSI .....	13	PR2 Register .....	26, 28, 69
RC1/T1OSI/CCP2 .....	14, 16	Prescaler, Switching Between Timer0 and WDT .....	63
RC2/CCP1 .....	13, 14, 16	PRO MATE Universal Programmer .....	163
RC3/SCK/SCL .....	13, 14, 16	Program Branches .....	9
RC4/SDI/SDA .....	13, 14, 16	Program Memory	
RC5/SDO .....	13, 14, 16	Paging .....	40
RC6 .....	13	Program Memory Maps	
RC6/TX/CK .....	14, 16, 99–114	PIC16C72 .....	19
RC7 .....	13	PIC16C73 .....	19
RC7/RX/DT .....	14, 16, 99–114	PIC16C73A .....	19
RD0/PSP0 .....	16	PIC16C74 .....	19
RD1/PSP1 .....	16	PIC16C74A .....	19
RD2/PSP2 .....	16	Program Verification .....	146
RD3/PSP3 .....	16	PS0 bit .....	31
RD4/PSP4 .....	16	PS1 bit .....	31
RD5/PSP5 .....	16	PS2 bit .....	31
RD6/PSP6 .....	16	PSA bit .....	31
RD7/PSP7 .....	16	PSPIE bit .....	34
RE0/RD/AN5 .....	16	PSPIF bit .....	36
RE1/WR/AN6 .....	16	PSPMODE bit .....	50, 51, 54
RE2/CS/AN7 .....	16	PUSH .....	40
SCK .....	80–82	<b>R</b>	
SDI .....	80–82	R/W .....	78, 83
SDO .....	80–82	R/W bit .....	90, 94, 95, 96
SS .....	80–82	RBIF bit .....	45, 143
VDD .....	13, 14, 16	RBPU bit .....	31
VSS .....	13, 14, 16	RC Oscillator .....	132, 135
Pinout Descriptions		RCIE bit .....	34
PIC16C72 .....	13	RCIF bit .....	36
PIC16C73 .....	14	RCREG .....	29
PIC16C73A .....	14	RCSTA Register .....	29, 100
PIC16C74 .....	15	RCV_MODE .....	98
PIC16C74A .....	15	RD pin .....	54
PIC16C76 .....	14	Read/Write bit Information, R/W .....	78, 83
PIC16C77 .....	15	Read-Modify-Write .....	53
PIR1 Register .....	35	Receive Overflow Detect bit, SSPOV .....	79
PIR2 Register .....	38	Receive Overflow Indicator bit, SSPOV .....	84
POP .....	40	Register File .....	20

# PIC16C7X

Registers	
FSR	
Summary	29
INDF	
Summary	29
Initialization Conditions	136
INTCON	
Summary	29
Maps	
PIC16C72	21
PIC16C73	21
PIC16C73A	21
PIC16C74	21
PIC16C74A	21
PIC16C76	22
PIC16C77	22
OPTION	
Summary	29
PCL	
Summary	29
PCLATH	
Summary	29
PORTB	
Summary	29
Reset Conditions	136
SSPBUF	
Section	80
SSPCON	
Diagram	79
SSPSR	
Section	80
SSPSTAT	83
Diagram	78
Section	78
STATUS	
Summary	29
Summary	25, 27
TMR0	
Summary	29
TRISB	
Summary	29
Reset	129, 133
Reset Conditions for Special Registers	136
RP0 bit	20, 30
RP1 bit	30
RX9 bit	100
RX9D bit	100
<b>S</b>	
S	78, 83
SCK	80
SCL	94
SDI	80
SDO	80
Serial Communication Interface (SCI) Module, See USART	
Services	
One-Time-Programmable (OTP)	7
Quick-Turnaround-Production (QTP)	7
Serialized Quick-Turnaround Production (SQTP)	7
Slave Mode	
SCL	94
SDA	94
SLEEP	129, 133
SMP	83
Software Simulator (MPSIM)	165
SPBRG	
Register	26, 28
Special Event Trigger	125
Special Features of the CPU	129
Special Function Registers	
PIC16C72	23
PIC16C73	25, 27
PIC16C73A	25, 27
PIC16C74	25, 27
PIC16C74A	25, 27
PIC16C76	27
PIC16C77	27
Special Function Registers, Section	23
SPEN bit	100
SPI	
Block Diagram	80, 85
Master Mode	86
Master Mode Timing	87
Mode	80
Serial Clock	85
Serial Data In	85
Serial Data Out	85
Slave Mode Timing	88
Slave Mode Timing Diagram	87
Slave Select	85
SPI clock	86
SPI Mode	85
SSPCON	84
SSPSTAT	83
SPI Clock Edge Select bit, CKE	83
SPI Data Input Sample Phase Select bit, SMP	83
SPI Mode	80
SREN bit	100
SS	80
SSP	
Module Overview	77
Section	77
SSPBUF	86
SSPCON	84
SSPSR	86
SSPSTAT	83
SSP in I <sup>2</sup> C Mode - See I <sup>2</sup> C	
SSPADD	93
SSPADD Register	24, 26, 28, 29
SSPBUF	29, 93
SSPBUF Register	25, 27
SSPCON	79, 84
SSPCON Register	25, 27
SSPEN	79, 84
SSPIE bit	33
SSPIF bit	35, 36
SSPM3:SSPM0	79, 84
SSPOV	79, 84, 94
SSPSTAT	78, 93
SSPSTAT Register	24, 26, 28, 29, 83
Stack	40
Overflows	40
Underflow	40
Start bit, S	78, 83
STATUS Register	29, 30
Stop bit, P	78, 83
Synchronous Serial Port (SSP)	
Block Diagram, SPI Mode	80
SPI Master/Slave Diagram	81
SPI Mode	80
Synchronous Serial Port Enable bit, SSPEN	79, 84

# PIC16C7X

Synchronous Serial Port Mode Select bits, SSPM3:SSPM0 .....	79, 84
Synchronous Serial Port Module .....	77
Synchronous Serial Port Status Register .....	83
<b>T</b>	
T0CS bit .....	31
T1CKPS0 bit .....	65
T1CKPS1 bit .....	65
T1CON .....	29
T1CON Register .....	29, 65
T1OSCEN bit .....	65
T1SYNC bit .....	65
T2CKPS0 bit .....	70
T2CKPS1 bit .....	70
T2CON Register .....	29, 70
TAD .....	121
Timer Modules, Overview .....	57
Timer0	
RTCC .....	136
Timers	
Timer0	
Block Diagram .....	59
External Clock .....	61
External Clock Timing .....	61
Increment Delay .....	61
Interrupt .....	59
Interrupt Timing .....	60
Overview .....	57
Prescaler .....	62
Prescaler Block Diagram .....	62
Section .....	59
Switching Prescaler Assignment .....	63
Synchronization .....	61
T0CKI .....	61
T0IF .....	143
Timing .....	59
TMR0 Interrupt .....	143
Timer1	
Asynchronous Counter Mode .....	67
Block Diagram .....	66
Capacitor Selection .....	67
External Clock Input .....	66
External Clock Input Timing .....	67
Operation in Timer Mode .....	66
Oscillator .....	67
Overview .....	57
Prescaler .....	66, 68
Resetting of Timer1 Registers .....	68
Resetting Timer1 using a CCP Trigger Output ..	68
Synchronized Counter Mode .....	66
T1CON .....	65
TMR1H .....	67
TMR1L .....	67
Timer2	
Block Diagram .....	69
Module .....	69
Overview .....	57
Postscaler .....	69
Prescaler .....	69
T2CON .....	70
Timing Diagrams	
A/D Conversion .....	182, 200, 218, 239
Brown-out Reset .....	134, 175, 209, 228
Capture/Compare/PWM .....	177, 193, 211, 230
CLKOUT and I/O .....	174, 190, 208, 227
External Clock Timing .....	173, 189, 207, 226
I <sup>2</sup> C Bus Data .....	180, 197, 215, 236
I <sup>2</sup> C Bus Start/Stop bits .....	179, 196, 214, 235
I <sup>2</sup> C Clock Synchronization .....	92
I <sup>2</sup> C Data Transfer Wait State .....	90
I <sup>2</sup> C Multi-Master Arbitration .....	92
I <sup>2</sup> C Reception (7-bit Address) .....	95
Parallel Slave Port .....	194
Power-up Timer .....	175, 191, 209, 228
Reset .....	175, 191, 209, 228
SPI Master Mode .....	87
SPI Mode .....	178, 195, 213
SPI Mode, Master/Slave Mode, No $\overline{SS}$ Control .....	82
SPI Mode, Slave Mode With $\overline{SS}$ Control .....	82
SPI Slave Mode (CKE = 1) .....	88
SPI Slave Mode Timing (CKE = 0) .....	87
Start-up Timer .....	175, 191, 209, 228
Time-out Sequence .....	139
Timer0 .....	59, 176, 192, 210, 229
Timer0 Interrupt Timing .....	60
Timer0 with External Clock .....	61
Timer1 .....	176, 192, 210, 229
USART Asynchronous Master Transmission .....	107
USART Asynchronous Reception .....	108
USART RX Pin Sampling .....	104, 105
USART Synchronous Receive .....	198, 216, 237
USART Synchronous Reception .....	113
USART Synchronous Transmission 111, 198, 216, 237	
Wake-up from Sleep via Interrupt .....	146
Watchdog Timer .....	175, 191, 209, 228
TMR0 .....	29
TMR0 Register .....	25, 27
TMR1CS bit .....	65
TMR1H .....	29
TMR1H Register .....	23, 25, 27
TMR1IE bit .....	33
TMR1IF bit .....	35, 36
TMR1L .....	29
TMR1L Register .....	23, 25, 27
TMR1ON bit .....	65
TMR2 .....	29
TMR2 Register .....	23, 25, 27
TMR2IE bit .....	33
TMR2IF bit .....	35, 36
TMR2ON bit .....	70
$\overline{TO}$ bit .....	30
TOUTPS0 bit .....	70
TOUTPS1 bit .....	70
TOUTPS2 bit .....	70
TOUTPS3 bit .....	70
TRISA .....	29
TRISA Register .....	24, 26, 28, 43
TRISB .....	29
TRISB Register .....	24, 26, 28, 45
TRISC .....	29
TRISC Register .....	24, 26, 28, 48
TRISD .....	29
TRISD Register .....	26, 28, 50
TRISE .....	29
TRISE Register .....	26, 28, 51
Two's Complement .....	9
TXIE bit .....	34
TXIF bit .....	36
TXREG .....	29
TXSTA .....	29
TXSTA Register .....	99

## U

UA .....	78, 83
Universal Synchronous Asynchronous Receiver Transmitter (USART) .....	99
Update Address bit, UA .....	78, 83
USART	
Asynchronous Mode .....	106
Asynchronous Receiver .....	108
Asynchronous Reception .....	109
Asynchronous Transmission .....	107
Asynchronous Transmitter .....	106
Baud Rate Generator (BRG) .....	101
Receive Block Diagram .....	108
Sampling .....	104
Synchronous Master Mode .....	110
Synchronous Master Reception .....	112
Synchronous Master Transmission .....	110
Synchronous Slave Mode .....	114
Synchronous Slave Reception .....	114
Synchronous Slave Transmit .....	114
Transmit Block Diagram .....	106
UV Erasable Devices .....	7

## W

W Register	
ALU .....	9
Wake-up from SLEEP .....	145
Watchdog Timer (WDT) .....	129, 133, 136, 144
WCOL .....	79, 84
WDT	
Block Diagram .....	144
Period .....	144
Programming Considerations .....	144
Timeout .....	136
Word .....	129
WR pin .....	54
Write Collision Detect bit, WCOL .....	79, 84

## X

XMIT_MODE .....	98
-----------------	----

## Z

Z bit .....	30
Zero bit .....	9

## LIST OF EXAMPLES

Example 3-1: Instruction Pipeline Flow.....	17
Example 4-1: Call of a Subroutine in Page 1 from Page 0 .....	41
Example 4-2: Indirect Addressing.....	41
Example 5-1: Initializing PORTA.....	43
Example 5-2: Initializing PORTB.....	45
Example 5-3: Initializing PORTC .....	48
Example 5-4: Read-Modify-Write Instructions on an I/O Port .....	53
Example 7-1: Changing Prescaler (Timer0→WDT).....	63
Example 7-2: Changing Prescaler (WDT→Timer0).....	63
Example 8-1: Reading a 16-bit Free-Running Timer .....	67
Example 10-1: Changing Between Capture Prescalers.....	73
Example 10-2: PWM Period and Duty Cycle Calculation.....	75
Example 11-1: Loading the SSPBUF (SSPSR) Register .....	80
Example 11-2: Loading the SSPBUF (SSPSR) Register (PIC16C76/77) .....	85
Example 12-1: Calculating Baud Rate Error.....	101
Equation 13-1: A/D Minimum Charging Time.....	120
Example 13-1: Calculating the Minimum Required Acquisition Time .....	120
Example 13-2: A/D Conversion.....	122
Example 13-3: 4-bit vs. 8-bit Conversion Times .....	123
Example 14-1: Saving STATUS, W, and PCLATH Registers in RAM.....	143

# PIC16C7X

## LIST OF FIGURES

Figure 3-1:	PIC16C72 Block Diagram .....	10	Figure 8-1:	T1CON: Timer1 Control Register (Address 10h) .....	65
Figure 3-2:	PIC16C73/73A/76 Block Diagram .....	11	Figure 8-2:	Timer1 Block Diagram .....	66
Figure 3-3:	PIC16C74/74A/77 Block Diagram .....	12	Figure 9-1:	Timer2 Block Diagram .....	69
Figure 3-4:	Clock/Instruction Cycle.....	17	Figure 9-2:	T2CON: Timer2 Control Register (Address 12h) .....	70
Figure 4-1:	PIC16C72 Program Memory Map and Stack .....	19	Figure 10-1:	CCP1CON Register (Address 17h)/ CCP2CON Register (Address 1Dh).....	72
Figure 4-2:	PIC16C73/73A/74/74A Program Memory Map and Stack .....	19	Figure 10-2:	Capture Mode Operation Block Diagram .....	72
Figure 4-3:	PIC16C76/77 Program Memory Map and Stack .....	20	Figure 10-3:	Compare Mode Operation Block Diagram .....	73
Figure 4-4:	PIC16C72 Register File Map .....	21	Figure 10-4:	Simplified PWM Block Diagram .....	74
Figure 4-5:	PIC16C73/73A/74/74A Register File Map .....	21	Figure 10-5:	PWM Output .....	74
Figure 4-6:	PIC16C76/77 Register File Map .....	22	Figure 11-1:	SSPSTAT: Sync Serial Port Status Register (Address 94h).....	78
Figure 4-7:	Status Register (Address 03h, 83h, 103h, 183h).....	30	Figure 11-2:	SSPCON: Sync Serial Port Control Register (Address 14h).....	79
Figure 4-8:	OPTION Register (Address 81h, 181h).....	31	Figure 11-3:	SSP Block Diagram (SPI Mode).....	80
Figure 4-9:	INTCON Register (Address 0Bh, 8Bh, 10bh, 18bh).....	32	Figure 11-4:	SPI Master/Slave Connection.....	81
Figure 4-10:	PIE1 Register PIC16C72 (Address 8Ch).....	33	Figure 11-5:	SPI Mode Timing, Master Mode or Slave Mode w/o SS Control.....	82
Figure 4-11:	PIE1 Register PIC16C73/73A/ 74/74A/76/77 (Address 8Ch).....	34	Figure 11-6:	SPI Mode Timing, Slave Mode with SS Control .....	82
Figure 4-12:	PIR1 Register PIC16C72 (Address 0Ch) .....	35	Figure 11-7:	SSPSTAT: Sync Serial Port Status Register (Address 94h)(PIC16C76/77)....	83
Figure 4-13:	PIR1 Register PIC16C73/73A/ 74/74A/76/77 (Address 0Ch).....	36	Figure 11-8:	SSPCON: Sync Serial Port Control Register (Address 14h)(PIC16C76/77)....	84
Figure 4-14:	PIE2 Register (Address 8Dh).....	37	Figure 11-9:	SSP Block Diagram (SPI Mode) (PIC16C76/77).....	85
Figure 4-15:	PIR2 Register (Address 0Dh).....	38	Figure 11-10:	SPI Master/Slave Connection PIC16C76/77) .....	86
Figure 4-16:	PCON Register (Address 8Eh) .....	39	Figure 11-11:	SPI Mode Timing, Master Mode (PIC16C76/77).....	87
Figure 4-17:	Loading of PC In Different Situations .....	40	Figure 11-12:	SPI Mode Timing (Slave Mode With CKE = 0) (PIC16C76/77).....	87
Figure 4-18:	Direct/Indirect Addressing .....	41	Figure 11-13:	SPI Mode Timing (Slave Mode With CKE = 1) (PIC16C76/77).....	88
Figure 5-1:	Block Diagram of RA3:RA0 and RA5 Pins .....	43	Figure 11-14:	Start and Stop Conditions.....	89
Figure 5-2:	Block Diagram of RA4/T0CKI Pin .....	43	Figure 11-15:	7-bit Address Format .....	90
Figure 5-3:	Block Diagram of RB3:RB0 Pins.....	45	Figure 11-16:	I <sup>2</sup> C 10-bit Address Format.....	90
Figure 5-4:	Block Diagram of RB7:RB4 Pins (PIC16C73/74) .....	46	Figure 11-17:	Slave-receiver Acknowledge .....	90
Figure 5-5:	Block Diagram of RB7:RB4 Pins (PIC16C72/73A/ 74A/76/77).....	46	Figure 11-18:	Data Transfer Wait State .....	90
Figure 5-6:	PORTC Block Diagram (Peripheral Output Override).....	48	Figure 11-19:	Master-transmitter Sequence .....	91
Figure 5-7:	PORTD Block Diagram (in I/O Port Mode).....	50	Figure 11-20:	Master-receiver Sequence.....	91
Figure 5-8:	PORTE Block Diagram (in I/O Port Mode).....	51	Figure 11-21:	Combined Format .....	91
Figure 5-9:	TRISE Register (Address 89h).....	51	Figure 11-22:	Multi-master Arbitration (Two Masters).....	92
Figure 5-10:	Successive I/O Operation .....	53	Figure 11-23:	Clock Synchronization .....	92
Figure 5-11:	PORTD and PORTE Block Diagram (Parallel Slave Port) .....	54	Figure 11-24:	SSP Block Diagram (I <sup>2</sup> C Mode) .....	93
Figure 5-12:	Parallel Slave Port Write Waveforms .....	55	Figure 11-25:	I <sup>2</sup> C Waveforms for Reception (7-bit Address) .....	95
Figure 5-13:	Parallel Slave Port Read Waveforms.....	55	Figure 11-26:	I <sup>2</sup> C Waveforms for Transmission (7-bit Address) .....	96
Figure 7-1:	Timer0 Block Diagram.....	59	Figure 11-27:	Operation of the I <sup>2</sup> C Module in IDLE_MODE, RCV_MODE or XMIT_MODE .....	98
Figure 7-2:	Timer0 Timing: Internal Clock/No Prescale .....	59	Figure 12-1:	TXSTA: Transmit Status and Control Register (Address 98h) .....	99
Figure 7-3:	Timer0 Timing: Internal Clock/Prescale 1:2 .....	60	Figure 12-2:	RCSTA: Receive Status and Control Register (Address 18h) .....	100
Figure 7-4:	Timer0 Interrupt Timing.....	60	Figure 12-3:	RX Pin Sampling Scheme. BRGH = 0 (PIC16C73/73A/74/74A) .....	104
Figure 7-5:	Timer0 Timing with External Clock.....	61	Figure 12-4:	RX Pin Sampling Scheme, BRGH = 1 (PIC16C73/73A/74/74A) .....	104
Figure 7-6:	Block Diagram of the Timer0/WDT Prescaler .....	62			



# PIC16C7X

Figure 12-5:	RX Pin Sampling Scheme, BRGH = 1 (PIC16C73/73A/74/74A) .....	104	Figure 17-4:	Reset, Watchdog Timer, Oscillator Start-up Timer and Power-up Timer Timing .....	175
Figure 12-6:	RX Pin Sampling Scheme, BRGH = 0 OR BRGH = 1 (PIC16C76/77) .....	105	Figure 17-5:	Brown-out Reset Timing .....	175
Figure 12-7:	USART Transmit Block Diagram.....	106	Figure 17-6:	Timer0 and Timer1 External Clock Timings .....	176
Figure 12-8:	Asynchronous Master Transmission .....	107	Figure 17-7:	Capture/Compare/PWM Timings (CCP1) .....	177
Figure 12-9:	Asynchronous Master Transmission (Back to Back) .....	107	Figure 17-8:	SPI Mode Timing .....	178
Figure 12-10:	USART Receive Block Diagram.....	108	Figure 17-9:	I <sup>2</sup> C Bus Start/Stop Bits Timing.....	179
Figure 12-11:	Asynchronous Reception .....	108	Figure 17-10:	I <sup>2</sup> C Bus Data Timing.....	180
Figure 12-12:	Synchronous Transmission.....	111	Figure 17-11:	A/D Conversion Timing.....	182
Figure 12-13:	Synchronous Transmission (Through TXEN).....	111	Figure 18-1:	Load Conditions.....	188
Figure 12-14:	Synchronous Reception (Master Mode, SREN).....	113	Figure 18-2:	External Clock Timing.....	189
Figure 13-1:	ADCON0 Register (Address 1Fh) .....	117	Figure 18-3:	CLKOUT and I/O Timing.....	190
Figure 13-2:	ADCON1 Register (Address 9Fh) .....	118	Figure 18-4:	Reset, Watchdog Timer, Oscillator Start-up Timer and Power-up Timer Timing.....	191
Figure 13-3:	A/D Block Diagram .....	119	Figure 18-5:	Timer0 and Timer1 External Clock Timings .....	192
Figure 13-4:	Analog Input Model .....	120	Figure 18-6:	Capture/Compare/PWM Timings (CCP1 and CCP2) .....	193
Figure 13-5:	A/D Transfer Function .....	125	Figure 18-7:	Parallel Slave Port Timing (PIC16C74).....	194
Figure 13-6:	Flowchart of A/D Operation.....	126	Figure 18-8:	SPI Mode Timing .....	195
Figure 14-1:	Configuration Word for PIC16C73/74.....	129	Figure 18-9:	I <sup>2</sup> C Bus Start/Stop Bits Timing.....	196
Figure 14-2:	Configuration Word for PIC16C72/73A/74A/76/77 .....	130	Figure 18-10:	I <sup>2</sup> C Bus Data Timing .....	197
Figure 14-3:	Crystal/Ceramic Resonator Operation (HS, XT or LP OSC Configuration).....	131	Figure 18-11:	USART Synchronous Transmission (Master/Slave) Timing.....	198
Figure 14-4:	External Clock Input Operation (HS, XT or LP OSC Configuration) .....	131	Figure 18-12:	USART Synchronous Receive (Master/Slave) Timing.....	198
Figure 14-5:	External Parallel Resonant Crystal Oscillator Circuit .....	132	Figure 18-13:	A/D Conversion Timing.....	200
Figure 14-6:	External Series Resonant Crystal Oscillator Circuit .....	132	Figure 19-1:	Load Conditions.....	206
Figure 14-7:	RC Oscillator Mode .....	132	Figure 19-2:	External Clock Timing.....	207
Figure 14-8:	Simplified Block Diagram of On-chip Reset Circuit.....	133	Figure 19-3:	CLKOUT and I/O Timing.....	208
Figure 14-9:	Brown-out Situations .....	134	Figure 19-4:	Reset, Watchdog Timer, Oscillator Start-up Timer and Power-up Timer Timing .....	209
Figure 14-10:	Time-out Sequence on Power-up (MCLR not Tied to VDD): Case 1.....	139	Figure 19-5:	Brown-out Reset Timing .....	209
Figure 14-11:	Time-out Sequence on Power-up (MCLR Not Tied To VDD): Case 2 .....	139	Figure 19-6:	Timer0 and Timer1 External Clock Timings .....	210
Figure 14-12:	Time-out Sequence on Power-up (MCLR Tied to VDD) .....	139	Figure 19-7:	Capture/Compare/PWM Timings (CCP1 and CCP2) .....	211
Figure 14-13:	External Power-on Reset Circuit (for Slow VDD Power-up).....	140	Figure 19-8:	Parallel Slave Port Timing (PIC16C74A) .....	212
Figure 14-14:	External Brown-out Protection Circuit 1 .....	140	Figure 19-9:	SPI Mode Timing .....	213
Figure 14-15:	External Brown-out Protection Circuit 2 .....	140	Figure 19-10:	I <sup>2</sup> C Bus Start/Stop Bits Timing.....	214
Figure 14-16:	Interrupt Logic .....	142	Figure 19-11:	I <sup>2</sup> C Bus Data Timing .....	215
Figure 14-17:	INT Pin Interrupt Timing .....	142	Figure 19-12:	USART Synchronous Transmission (Master/Slave) Timing.....	216
Figure 14-18:	Watchdog Timer Block Diagram .....	144	Figure 19-13:	USART Synchronous Receive (Master/Slave) Timing.....	216
Figure 14-19:	Summary of Watchdog Timer Registers.....	144	Figure 19-14:	A/D Conversion Timing.....	218
Figure 14-20:	Wake-up from Sleep Through Interrupt.....	146	Figure 20-1:	Load Conditions.....	225
Figure 14-21:	Typical In-Circuit Serial Programming Connection .....	146	Figure 20-2:	External Clock Timing.....	226
Figure 15-1:	General Format for Instructions .....	147	Figure 20-3:	CLKOUT and I/O Timing.....	227
Figure 17-1:	Load Conditions .....	172	Figure 20-4:	Reset, Watchdog Timer, Oscillator Start-up Timer and Power-up Timer Timing .....	228
Figure 17-2:	External Clock Timing .....	173	Figure 20-5:	Brown-out Reset Timing .....	228
Figure 17-3:	CLKOUT and I/O Timing.....	174	Figure 20-6:	Timer0 and Timer1 External Clock Timings .....	229
			Figure 20-7:	Capture/Compare/PWM Timings (CCP1 and CCP2) .....	230
			Figure 20-8:	Parallel Slave Port Timing (PIC16C77).....	231

# PIC16C7X

---

Figure 20-9:	SPI Master Mode Timing (CKE = 0).....	232	Figure 21-27:	Typical IDD vs. Frequency (XT Mode, 25°C).....	249
Figure 20-10:	SPI Master Mode Timing (CKE = 1).....	232	Figure 21-28:	Maximum IDD vs. Frequency (XT Mode, -40°C to 85°C).....	249
Figure 20-11:	SPI Slave Mode Timing (CKE = 0).....	233	Figure 21-29:	Typical IDD vs. Frequency (HS Mode, 25°C).....	250
Figure 20-12:	SPI Slave Mode Timing (CKE = 1).....	233	Figure 21-30:	Maximum IDD vs. Frequency (HS Mode, -40°C to 85°C).....	250
Figure 20-13:	I <sup>2</sup> C Bus Start/Stop Bits Timing.....	235			
Figure 20-14:	I <sup>2</sup> C Bus Data Timing.....	236			
Figure 20-15:	USART Synchronous Transmission (Master/Slave) Timing.....	237			
Figure 20-16:	USART Synchronous Receive (Master/Slave) Timing.....	237			
Figure 20-17:	A/D Conversion Timing.....	239			
Figure 21-1:	Typical IPD vs. VDD (WDT Disabled, RC Mode).....	241			
Figure 21-2:	Maximum IPD vs. VDD (WDT Disabled, RC Mode).....	241			
Figure 21-3:	Typical IPD vs. VDD @ 25°C (WDT Enabled, RC Mode).....	242			
Figure 21-4:	Maximum IPD vs. VDD (WDT Enabled, RC Mode).....	242			
Figure 21-5:	Typical RC Oscillator Frequency vs. VDD.....	242			
Figure 21-6:	Typical RC Oscillator Frequency vs. VDD.....	242			
Figure 21-7:	Typical RC Oscillator Frequency vs. VDD.....	242			
Figure 21-8:	Typical IPD vs. VDD Brown-out Detect Enabled (RC Mode).....	243			
Figure 21-9:	Maximum IPD vs. VDD Brown-out Detect Enabled (85°C to -40°C, RC Mode).....	243			
Figure 21-10:	Typical IPD vs. Timer1 Enabled (32 kHz, RC0/RC1= 33 pF/33 pF, RC Mode).....	243			
Figure 21-11:	Maximum IPD vs. Timer1 Enabled (32 kHz, RC0/RC1 = 33 pF/33 pF, 85°C to -40°C, RC Mode).....	243			
Figure 21-12:	Typical IDD vs. Frequency (RC Mode @ 22 pF, 25°C).....	244			
Figure 21-13:	Maximum IDD vs. Frequency (RC Mode @ 22 pF, -40°C to 85°C).....	244			
Figure 21-14:	Typical IDD vs. Frequency (RC Mode @ 100 pF, 25°C).....	245			
Figure 21-15:	Maximum IDD vs. Frequency ( RC Mode @ 100 pF, -40°C to 85°C).....	245			
Figure 21-16:	Typical IDD vs. Frequency (RC Mode @ 300 pF, 25°C).....	246			
Figure 21-17:	Maximum IDD vs. Frequency (RC Mode @ 300 pF, -40°C to 85°C).....	246			
Figure 21-18:	Typical IDD vs. Capacitance @ 500 kHz (RC Mode).....	247			
Figure 21-19:	Transconductance(gm) of HS Oscillator vs. VDD.....	247			
Figure 21-20:	Transconductance(gm) of LP Oscillator vs. VDD.....	247			
Figure 21-21:	Transconductance(gm) of XT Oscillator vs. VDD.....	247			
Figure 21-22:	Typical XTAL Startup Time vs. VDD (LP Mode, 25°C).....	248			
Figure 21-23:	Typical XTAL Startup Time vs. VDD (HS Mode, 25°C).....	248			
Figure 21-24:	Typical XTAL Startup Time vs. VDD (XT Mode, 25°C).....	248			
Figure 21-25:	Typical Idd vs. Frequency (LP Mode, 25°C).....	249			
Figure 21-26:	Maximum IDD vs. Frequency (LP Mode, 85°C to -40°C).....	249			

# PIC16C7X

## LIST OF TABLES

Table 1-1:	PIC16C7XX Family of Devces .....	6	Table 12-8:	Registers Associated with Synchronous Master Transmission .....	111
Table 3-1:	PIC16C72 Pinout Description .....	13	Table 12-9:	Registers Associated with Synchronous Master Reception .....	112
Table 3-2:	PIC16C73/73A/76 Pinout Description .....	14	Table 12-10:	Registers Associated with Synchronous Slave Transmission .....	115
Table 3-3:	PIC16C74/74A/77 Pinout Description .....	15	Table 12-11:	Registers Associated with Synchronous Slave Reception.....	115
Table 4-1:	PIC16C72 Special Function Register Summary.....	23	Table 13-1:	TAD vs. Device Operating Frequencies.....	121
Table 4-2:	PIC16C73/73A/74/74A Special Function Register Summary.....	25	Table 13-2:	Registers/Bits Associated with A/D, PIC16C72 .....	126
Table 4-3:	PIC16C76/77 Special Function Register Summary .....	27	Table 13-3:	Summary of A/D Registers, PIC16C73/73A/74/74A/76/77 .....	127
Table 5-1:	PORTA Functions .....	44	Table 14-1:	Ceramic Resonators .....	131
Table 5-2:	Summary of Registers Associated with PORTA .....	44	Table 14-2:	Capacitor Selection for Crystal Oscillator.....	131
Table 5-3:	PORTB Functions .....	46	Table 14-3:	Time-out in Various Situations, PIC16C73/74 .....	135
Table 5-4:	Summary of Registers Associated with PORTB .....	47	Table 14-4:	Time-out in Various Situations, PIC16C72/73A/74A/76/77 .....	135
Table 5-5:	PORTC Functions .....	48	Table 14-5:	Status Bits and Their Significance, PIC16C73/74 .....	135
Table 5-6:	Summary of Registers Associated with PORTC .....	49	Table 14-6:	Status Bits and Their Significance, PIC16C72/73A/74A/76/77 .....	136
Table 5-7:	PORTD Functions .....	50	Table 14-7:	Reset Condition for Special Registers.....	136
Table 5-8:	Summary of Registers Associated with PORTD .....	50	Table 14-8:	Initialization Conditions for all Registers.....	136
Table 5-9:	PORTE Functions .....	52	Table 15-1:	Opcode Field Descriptions.....	147
Table 5-10:	Summary of Registers Associated with PORTE .....	52	Table 15-2:	PIC16CXX Instruction Set .....	148
Table 5-11:	Registers Associated with Parallel Slave Port.....	55	Table 16-1:	Development Tools from Microchip .....	166
Table 7-1:	Registers Associated with Timer0.....	63	Table 17-1:	Cross Reference of Device Specs for Oscillator Configurations and Frequencies of Operation (Commercial Devices) .....	167
Table 8-1:	Capacitor Selection for the Timer1 Oscillator .....	67	Table 17-2:	External Clock Timing Requirements .....	173
Table 8-2:	Registers Associated with Timer1 as a Timer/Counter .....	68	Table 17-3:	CLKOUT and I/O Timing Requirements .....	174
Table 9-1:	Registers Associated with Timer2 as a Timer/Counter .....	70	Table 17-4:	Reset, Watchdog Timer, Oscillator Start-up Timer, Power-up Timer, and brown-out Reset Requirements .....	175
Table 10-1:	CCP Mode - Timer Resource.....	71	Table 17-5:	Timer0 and Timer1 External Clock Requirements .....	176
Table 10-2:	Interaction of Two CCP Modules .....	71	Table 17-6:	Capture/Compare/PWM Requirements (CCP1) .....	177
Table 10-3:	Example PWM Frequencies and Resolutions at 20 MHz .....	75	Table 17-7:	SPI Mode Requirements.....	178
Table 10-4:	Registers Associated with Capture, Compare, and Timer1 .....	75	Table 17-8:	I <sup>2</sup> C Bus Start/Stop Bits Requirements .....	179
Table 10-5:	Registers Associated with PWM and Timer2.....	76	Table 17-9:	I <sup>2</sup> C Bus Data Requirements .....	180
Table 11-1:	Registers Associated with SPI Operation .....	82	Table 17-10:	A/D Converter Characteristics: PIC16C72-04 (Commercial, Industrial, Extended) PIC16C72-10 (Commercial, Industrial, Extended) PIC16C72-20 (Commercial, Industrial, Extended) PIC16LC72-04 (Commercial, Industrial).....	181
Table 11-2:	Registers Associated with SPI Operation (PIC16C76/77) .....	88	Table 17-11:	A/D Conversion Requirements .....	182
Table 11-3:	I <sup>2</sup> C Bus Terminology .....	89	Table 18-1:	Cross Reference of Device Specs for Oscillator Configurations and Frequencies of Operation (Commercial Devices) .....	183
Table 11-4:	Data Transfer Received Byte Actions .....	94			
Table 11-5:	Registers Associated with I <sup>2</sup> C Operation .....	97			
Table 12-1:	Baud Rate Formula.....	101			
Table 12-2:	Registers Associated with Baud Rate Generator .....	101			
Table 12-3:	Baud Rates for Synchronous Mode .....	102			
Table 12-4:	Baud Rates for Asynchronous Mode (BRGH = 0) .....	102			
Table 12-5:	Baud Rates for Asynchronous Mode (BRGH = 1) .....	103			
Table 12-6:	Registers Associated with Asynchronous Transmission .....	107			
Table 12-7:	Registers Associated with Asynchronous Reception .....	109			

# PIC16C7X

Table 18-2:	external Clock Timing Requirements.....	189	Table 20-1:	Cross Reference of Device Specs for Oscillator Configurations and Frequencies of Operation (Commercial Devices) .....	220
Table 18-3:	CLKOUT and I/O Timing Requirements.....	190	Table 20-2:	External Clock Timing Requirements .....	226
Table 18-4:	Reset, Watchdog Timer, Oscillator Start-up Timer and Power-up Timer Requirements.....	191	Table 20-3:	CLKOUT and I/O Timing Requirements .....	227
Table 18-5:	Timer0 and Timer1 External Clock Requirements.....	192	Table 20-4:	Reset, Watchdog Timer, Oscillator Start-up Timer, Power-up Timer, and brown-out reset Requirements .....	228
Table 18-6:	Capture/Compare/PWM Requirements (CCP1 and CCP2) .....	193	Table 20-5:	Timer0 and Timer1 External Clock Requirements .....	229
Table 18-7:	Parallel Slave Port Requirements (PIC16C74) .....	194	Table 20-6:	Capture/Compare/PWM Requirements (CCP1 and CCP2) .....	230
Table 18-8:	SPI Mode Requirements .....	195	Table 20-7:	Parallel Slave Port Requirements (PIC16C77) .....	231
Table 18-9:	I <sup>2</sup> C Bus Start/Stop Bits Requirements.....	196	Table 20-8:	SPI Mode requirements .....	234
Table 18-10:	I <sup>2</sup> C Bus Data Requirements.....	197	Table 20-9:	I <sup>2</sup> C Bus Start/Stop Bits Requirements ....	235
Table 18-11:	USART Synchronous Transmission Requirements.....	198	Table 20-10:	I <sup>2</sup> C Bus Data Requirements .....	236
Table 18-12:	usart Synchronous Receive Requirements.....	198	Table 20-11:	USART Synchronous Transmission Requirements .....	237
Table 18-13:	A/D Converter Characteristics:.....	199	Table 20-12:	USART Synchronous Receive Requirements .....	237
	PIC16C73/74-04 (Commercial, Industrial)		Table 20-13:	A/D Converter Characteristics: .....	238
	PIC16C73/74-10 (Commercial, Industrial)			PIC16C76/77-04 (Commercial, Industrial, Extended)	
	PIC16C73/74-20 (Commercial, Industrial)			PIC16C76/77-10 (Commercial, Industrial, Extended)	
	PIC16LC73/74-04 (Commercial, Industrial) .....	199		PIC16C76/77-20 (Commercial, Industrial, Extended)	
Table 18-14:	A/D Conversion Requirements.....	200		PIC16LC76/77-04 (Commercial, Industrial).....	238
Table 19-1:	Cross Reference of Device Specs for Oscillator Configurations and Frequencies of Operation (Commercial Devices) .....	201	Table 20-14:	A/D Conversion Requirements .....	239
Table 19-2:	External Clock Timing Requirements.....	207	Table 21-1:	RC Oscillator Frequencies .....	247
Table 19-3:	CLKOUT and I/O Timing Requirements.....	208	Table 21-2:	Capacitor Selection for Crystal Oscillators .....	248
Table 19-4:	Reset, Watchdog Timer, Oscillator Start-up Timer, Power-up Timer, and brown-out reset Requirements.....	209	Table E-1:	Pin Compatible Devices.....	271
Table 19-5:	Timer0 and Timer1 External Clock Requirements.....	210			
Table 19-6:	Capture/Compare/PWM Requirements (CCP1 and CCP2) .....	211			
Table 19-7:	Parallel Slave Port Requirements (PIC16C74A) .....	212			
Table 19-8:	SPI Mode Requirements .....	213			
Table 19-9:	I <sup>2</sup> C Bus Start/Stop Bits Requirements ....	214			
Table 19-10:	I <sup>2</sup> C Bus Data Requirements.....	215			
Table 19-11:	USART Synchronous Transmission Requirements.....	216			
Table 19-12:	USART Synchronous Receive Requirements.....	216			
Table 19-13:	A/D Converter Characteristics:.....	217			
	PIC16C73A/74A-04 (Commercial, Industrial, Extended)				
	PIC16C73A/74A-10 (Commercial, Industrial, Extended)				
	PIC16C73A/74A-20 (Commercial, Industrial, Extended)				
	PIC16LC73A/74A-04 (Commercial, Industrial) .....	217			
Table 19-14:	A/D Conversion Requirements.....	218			

## ON-LINE SUPPORT

Microchip provides two methods of on-line support. These are the Microchip BBS and the Microchip World Wide Web (WWW) site.

Use Microchip's Bulletin Board Service (BBS) to get current information and help about Microchip products. Microchip provides the BBS communication channel for you to use in extending your technical staff with microcontroller and memory experts.

To provide you with the most responsive service possible, the Microchip systems team monitors the BBS, posts the latest component data and software tool updates, provides technical help and embedded systems insights, and discusses how Microchip products provide project solutions.

The web site, like the BBS, 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](http://www.microchip.com)**

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

**<ftp://ftp.futureone.com/pub/microchip>**

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

### Connecting to the Microchip BBS

Connect worldwide to the Microchip BBS using either the Internet or the CompuServe® communications network.

#### Internet:

You can telnet or ftp to the Microchip BBS at the address: **[mchipbbs.microchip.com](http://mchipbbs.microchip.com)**

#### **CompuServe Communications Network:**

When using the BBS via the Compuserve Network, in most cases, a local call is your only expense. The Microchip BBS connection does not use CompuServe membership services, therefore you do not need CompuServe membership to join Microchip's BBS. There is no charge for connecting to the Microchip BBS.

The procedure to connect will vary slightly from country to country. Please check with your local CompuServe agent for details if you have a problem. CompuServe service allow multiple users various baud rates depending on the local point of access.

The following connect procedure applies in most locations.

1. Set your modem to 8-bit, No parity, and One stop (8N1). This is not the normal CompuServe setting which is 7E1.
2. Dial your local CompuServe access number.
3. Depress the **<Enter>** key and a garbage string will appear because CompuServe is expecting a 7E1 setting.
4. Type +, depress the **<Enter>** key and "Host Name:" will appear.
5. Type MCHIPBBS, depress the **<Enter>** key and you will be connected to the Microchip BBS.

In the United States, to find the CompuServe phone number closest to you, set your modem to 7E1 and dial (800) 848-4480 for 300-2400 baud or (800) 331-7166 for 9600-14400 baud connection. After the system responds with "Host Name:", type NETWORK, depress the **<Enter>** key and follow CompuServe's directions.

For voice information (or calling from overseas), you may call (614) 723-1550 for your local CompuServe number.

Microchip regularly uses the Microchip BBS to distribute technical information, application notes, source code, errata sheets, bug reports, and interim patches for Microchip systems software products. For each SIG, a moderator monitors, scans, and approves or disapproves files submitted to the SIG. No executable files are accepted from the user community in general to limit the spread of computer viruses.

### 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

1-602-786-7302 for the rest of the world.

970301

**Trademarks:** The Microchip name, logo, PIC, PICSTART, PICMASTER and PRO MATE are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries. *FlexROM*, *MPLAB* and *fuzzyLAB*, are trademarks and SQTP is a service mark of Microchip in the U.S.A.

*fuzzyTECH* is a registered trademark of Inform Software Corporation. IBM, IBM PC-AT are registered trademarks of International Business Machines Corp. Pentium is a trademark of Intel Corporation. Windows is a trademark and MS-DOS, Microsoft Windows are registered trademarks of Microsoft Corporation. CompuServe is a registered trademark of CompuServe Incorporated.

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

# PIC16C6X

---

---

## 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 (602) 786-7578.

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

To: Technical Publications Manager Total Pages Sent  
RE: Reader Response  
From: Name \_\_\_\_\_  
Company \_\_\_\_\_  
Address \_\_\_\_\_  
City / State / ZIP / Country \_\_\_\_\_  
Telephone: (\_\_\_\_) \_\_\_\_\_ - \_\_\_\_\_ FAX: (\_\_\_\_) \_\_\_\_\_ - \_\_\_\_\_

Application (optional):

Would you like a reply? \_\_\_Y \_\_\_N

Device: **PIC16C6X** Literature Number: **DS30390E**

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?

---

---

# PIC16C7X

## PIC16C7X 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.	-XX	X	/XX	XXX		Examples	
					<b>Pattern:</b>	QTP, SQTP, Code or Special Requirements	a) PIC16C72 - 04/P 301 Commercial Temp., PDIP Package, 4 MHz, normal V <sub>DD</sub> limits, QTP pattern #301
					<b>Package:</b>	JW = Windowed CERDIP PQ = MQFP (Metric PQFP) TQ = TQFP (Thin Quad Flatpack) SO = SOIC SP = Skinny plastic dip P = PDIP L = PLCC SS = SSOP	b) PIC16LC76 - 041/SO Industrial Temp., SOIC package, 4 MHz, extended V <sub>DD</sub> limits
					<b>Temperature Range:</b>	- = 0°C to +70°C I = -40°C to +85°C E = -40°C to +125°C	c) PIC16C74A - 10E/P Automotive Temp., PDIP package, 10 MHz, normal V <sub>DD</sub> limits
					<b>Frequency Range:</b>	04 = 200 kHz (PIC16C7X-04) 04 = 4 MHz 10 = 10 MHz 20 = 20 MHz	
					<b>Device</b>	PIC16C7X :V <sub>DD</sub> range 4.0V to 6.0V PIC16C7XT :V <sub>DD</sub> range 4.0V to 6.0V (Tape/Reel) PIC16LC7X :V <sub>DD</sub> range 2.5V to 6.0V PIC16LC7XT :V <sub>DD</sub> range 2.5V to 6.0V (Tape/Reel)	

\* JW Devices are UV erasable and can be programmed to any device configuration. JW Devices meet the electrical requirement of each oscillator type (including LC devices).

## Sales and Support

Products supported by a preliminary Data Sheet may possibly 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. The Microchip Website at [www.microchip.com](http://www.microchip.com)
2. Your local Microchip sales office (see following page)
3. The Microchip Corporate Literature Center U.S. FAX: (602) 786-7277
4. The Microchip's Bulletin Board, via your local CompuServe number (CompuServe membership NOT required).

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

For latest version information and upgrade kits for Microchip Development Tools, please call 1-800-755-2345 or 1-602-786-7302.

---

---

**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, microID, 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.

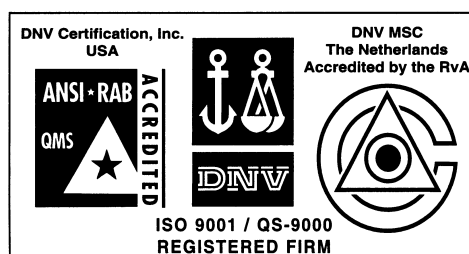
dsPIC, ECONOMONITOR, FanSense, FlexROM, fuzzyLAB, In-Circuit Serial Programming, ICSP, ICEPIC, microPort, Migratable Memory, MPASM, MPLIB, MPLINK, MPSIM, MXDEV, PICC, PICDEM, PICDEM.net, rPIC, Select Mode and Total Endurance are trademarks of Microchip Technology Incorporated in the U.S.A.

Serialized Quick Turn 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.*

---

---



---

---

**WORLDWIDE SALES AND SERVICE**

---

---