



Welcome to [E-XFL.COM](#)

### What is "[Embedded - Microcontrollers](#)"?

"[Embedded - Microcontrollers](#)" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

### Applications of "[Embedded - Microcontrollers](#)"

#### Details

Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	20MHz
Connectivity	I <sup>2</sup> C, SPI
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	22
Program Memory Size	3.5KB (2K x 14)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	128 x 8
Voltage - Supply (Vcc/Vdd)	4V ~ 5.5V
Data Converters	-
Oscillator Type	External
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SSOP (0.209", 5.30mm Width)
Supplier Device Package	28-SSOP
Purchase URL	<a href="https://www.e-xfl.com/product-detail/microchip-technology/pic16c62b-20-ss">https://www.e-xfl.com/product-detail/microchip-technology/pic16c62b-20-ss</a>

## Table of Contents

1.0 Device Overview .....	5
2.0 Memory Organization .....	7
3.0 I/O Ports .....	19
4.0 Timer0 Module .....	25
5.0 Timer1 Module .....	27
6.0 Timer2 Module .....	31
7.0 Capture/Compare/PWM (CCP) Module .....	33
8.0 Synchronous Serial Port (SSP) Module .....	39
9.0 Analog-to-Digital Converter (A/D) Module .....	49
10.0 Special Features of the CPU .....	55
11.0 Instruction Set Summary .....	67
12.0 Development Support .....	75
13.0 Electrical Characteristics .....	81
14.0 DC and AC Characteristics Graphs and Tables .....	103
15.0 Packaging Information .....	105
Appendix A: Revision History .....	111
Appendix B: Conversion Considerations .....	111
Appendix C: Migration from Base-line to Mid-Range Devices .....	112
Index .....	113
On-Line Support .....	117
Reader Response .....	118
PIC16C62B/72A Product Identification System .....	119

### *To Our Valued Customers*

#### **Most Current Data Sheet**

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

<http://www.microchip.com>

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

#### **New Customer Notification System**

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

#### **Errata**

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

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

- Microchip's Worldwide Web site; <http://www.microchip.com>
- Your local Microchip sales office (see last page)
- The Microchip Corporate Literature Center; U.S. FAX: (480) 786-7277

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

#### **Corrections to this Data Sheet**

We constantly strive to improve the quality of all our products and documentation. We have spent a great deal of time to ensure that this document is 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:

- Fill out and mail in the reader response form in the back of this data sheet.
- E-mail us at [webmaster@microchip.com](mailto:webmaster@microchip.com).

We appreciate your assistance in making this a better document.

## 2.0 MEMORY ORGANIZATION

There are two memory blocks in each of these micro-controllers. Each block (Program Memory and Data Memory) has its own bus, so that concurrent access can occur.

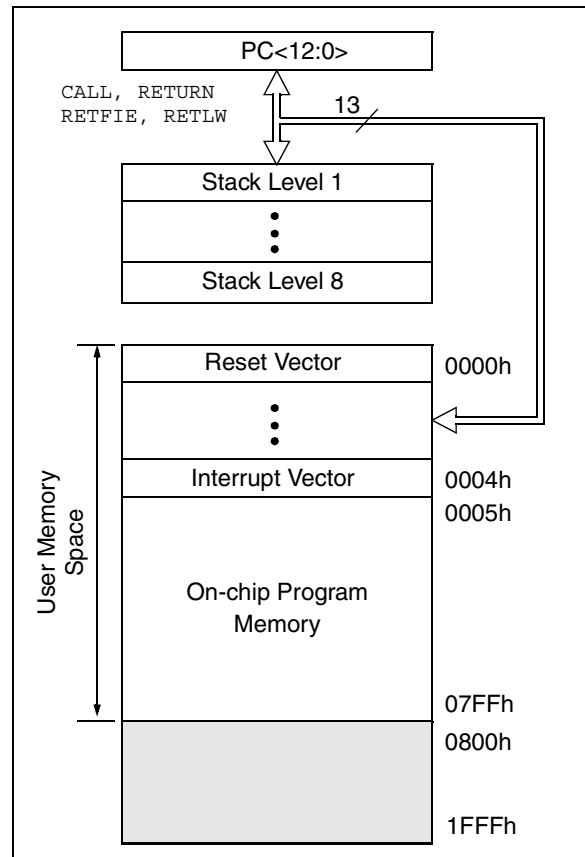
Additional information on device memory may be found in the PICmicro™ Mid-Range Reference Manual, (DS33023).

### 2.1 Program Memory Organization

The PIC16C62B/72A devices have a 13-bit program counter capable of addressing an 8K x 14 program memory space. Each device has 2K x 14 words of program memory. Accessing a location above 07FFh will cause a wraparound.

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

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



# PIC16C62B/72A

## 2.2.2.6 PCON REGISTER

The Power Control register (PCON) contains flag bits to allow differentiation between a Power-on Reset (POR), Brown-Out Reset (BOR) and resets from other sources. .

**Note:** On Power-on Reset, the state of the  $\overline{\text{BOR}}$  bit is unknown and is not predictable. If the BODEN bit in the configuration word is set, the user must first set the BOR bit on a POR, and check it on subsequent resets. If BOR is cleared while POR remains set, a Brown-out reset has occurred. If the BODEN bit is clear, the BOR bit may be ignored.

### REGISTER 2-6: PCON REGISTER (ADDRESS 8Eh)

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-q
—	—	—	—	—	—	POR	$\overline{\text{BOR}}$
bit7							bit0
<div>bit 7-2: <b>Unimplemented:</b> Read as '0'</div> <div>bit 1: <b><math>\overline{\text{POR}}</math>:</b> Power-on Reset Status bit 1 = No Power-on Reset occurred 0 = A Power-on Reset occurred (must be set in software after a Power-on Reset occurs)</div> <div>bit 0: <b><math>\overline{\text{BOR}}</math>:</b> Brown-out Reset Status bit 1 = No Brown-out Reset occurred 0 = A Brown-out Reset occurred (must be set in software after a Brown-out Reset occurs)</div>							

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

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

The 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. Table 7-1 shows the timer resources of the CCP module modes.

Capture/Compare/PWM Register 1 (CCPR1) is comprised of two 8-bit registers: CCPR1L (low byte) and CCPR1H (high byte). The CCP1CON register controls the operation of CCP1. All are readable and writable.

Additional information on the CCP module is available in the PIC® MCU Mid-Range Reference Manual, (DS33023).

**TABLE 7-1 CCP MODE - TIMER RESOURCE**

CCP Mode	Timer Resource
Capture	Timer1
Compare	Timer1
PWM	Timer2

**TABLE 7-2 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.

### REGISTER 7-1: CCP1CON REGISTER (ADDRESS 17h)

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0
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 5-4: **CCP1X:CCP1Y:** PWM Least Significant bits  
Capture Mode: Unused  
Compare Mode: Unused  
PWM Mode: These bits are the two LSBs of the PWM duty cycle. The eight MSBs are found in CCPR1L.

bit 3-0: **CCP1M3:CCP1M0:** CCP1 Mode Select bits  
0000 = Capture/Compare/PWM off (resets CCP1 module)  
0100 = Capture mode, every falling edge  
0101 = Capture mode, every rising edge  
0110 = Capture mode, every 4th rising edge  
0111 = Capture mode, every 16th rising edge  
1000 = Compare mode, set output on match (CCP1IF bit is set)  
1001 = Compare mode, clear output on match (CCP1IF bit is set)  
1010 = Compare mode, generate software interrupt on match (CCP1IF bit is set, CCP1 pin is unaffected)  
1011 = Compare mode, trigger special event (CCP1IF bit is set; CCP1 resets TMR1 and starts an A/D conversion (if A/D module is enabled))  
11xx = PWM mode

## 7.3 PWM Mode

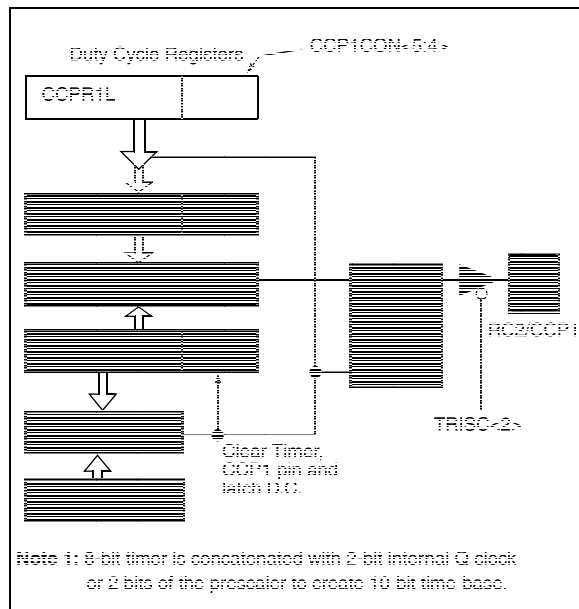
In Pulse Width Modulation (PWM) mode, the CCP1 pin produces up to a 10-bit resolution PWM output. Since the CCP1 pin is multiplexed with the PORTC data latch, the TRISC<2> bit must be cleared to make the CCP1 pin an output.

**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 7-3 shows a simplified block diagram of the CCP module in PWM mode.

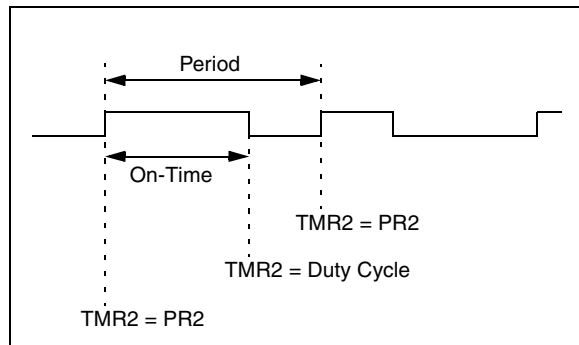
For a step by step procedure on how to set up the CCP module for PWM operation, see Section 7.3.3.

**FIGURE 7-3: SIMPLIFIED PWM BLOCK DIAGRAM**



A PWM output (Figure 7-4) has a time base (period) and a time that the output stays high (on-time). The frequency of the PWM is the inverse of the period (1/period).

**FIGURE 7-4: PWM OUTPUT**



### 7.3.1 PWM PERIOD

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

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

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 6.0) 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.

### 7.3.2 PWM ON-TIME

The PWM on-time is specified by writing to the CCPR1L register and to the CCP1CON<5:4> bits. Up to 10-bit resolution is available. CCPR1L contains eight MSBs and CCP1CON<5:4> contains 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:

$$\text{PWM on-time} = (\text{CCPR1L:CCP1CON<5:4>}) \cdot T_{osc} \cdot (\text{TMR2 prescale value})$$

CCPR1L and CCP1CON<5:4> can be written to at any time, but the on-time 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 on-time. 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:

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

**Note:** If the PWM on-time value is larger than the PWM period, the CCP1 pin will not be cleared.

For an example PWM period and on-time calculation, see the PIC® MCU Mid-Range Reference Manual, (DS33023).

NOTES:

## 8.3.2 MASTER OPERATION

Master 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 by 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 operation, the SCL and SDA lines are manipulated in software 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
- Byte transfer completed

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

For more information on master operation, see AN554 - *Software Implementation of I<sup>2</sup>C Bus Master*.

## 8.3.3 MULTI-MASTER OPERATION

In multi-master operation, the interrupt generation on the detection of the START and STOP conditions allows the determination of when the bus is free. The STOP (P) and START (S) bits are cleared from a reset or when the SSP module is disabled. The STOP (P) and START (S) bits will toggle based on the START and STOP conditions. Control of the I<sup>2</sup>C bus may be taken when bit P (SSPSTAT<4>) is set, or the bus is idle and both the S and P bits clear. When the bus is busy, enabling the SSP Interrupt will generate the interrupt when the STOP condition occurs.

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

- Address Transfer
- Data Transfer

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

For more information on master operation, see AN578 - *Use of the SSP Module in the of I<sup>2</sup>C Multi-Master Environment*.

**TABLE 8-3 REGISTERS ASSOCIATED WITH I<sup>2</sup>C OPERATION**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other resets
0Bh, 8Bh	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
13h	SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register								xxxx xxxx	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>(1)</sup>	CKE <sup>(1)</sup>	D/ $\bar{A}$	P	S	R/ $\bar{W}$	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:** Maintain these bits clear in I<sup>2</sup>C mode.



## REGISTER 8-2: 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 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 operation, 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)

bit 3-0: **SSPM3:SSPM0:** Synchronous Serial Port Mode Select bits  
0000 = SPI master operation, clock = FOSC/4  
0001 = SPI master operation, clock = FOSC/16  
0010 = SPI master operation, clock = FOSC/64  
0011 = SPI master operation, 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 operation (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

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

**Note:** This section applies to the PIC16C72A only.

The analog-to-digital (A/D) converter module has five input channels.

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 the 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.

Additional information on the A/D module is available in the PIC® MCU Mid-Range Reference Manual, (DS33023).

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)

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 ADCON0 register, shown in Figure 9-1, controls the operation of the A/D module. The ADCON1 register, shown in Figure 9-2, 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.

### REGISTER 9-1: ADCON0 REGISTER (ADDRESS 1Fh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	—	ADON
bit7							bit0

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

bit 7-6: **ADCS1:ADCS0:** A/D Conversion Clock Select bits  
00 = Fosc/2  
01 = Fosc/8  
10 = Fosc/32  
11 = FRC (clock derived from an internal RC oscillator)

bit 5-3: **CHS2:CHS0:** Analog Channel Select bits  
000 = channel 0, (RA0/AN0)  
001 = channel 1, (RA1/AN1)  
010 = channel 2, (RA2/AN2)  
011 = channel 3, (RA3/AN3)  
100 = channel 4, (RA5/AN4)

bit 2: **GO/DONE:** A/D Conversion Status bit  
If ADON = 1  
1 = A/D conversion in progress (setting this bit starts the A/D conversion)  
0 = A/D conversion not in progress (This bit is automatically cleared by hardware when the A/D conversion is complete)

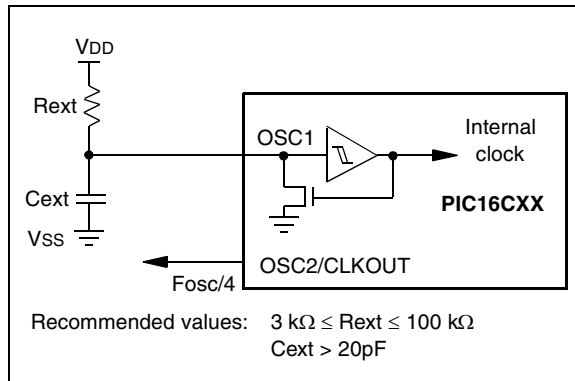
bit 1: **Unimplemented:** Read as '0'

bit 0: **ADON:** A/D On bit  
1 = A/D converter module is operating  
0 = A/D converter module is shutoff and consumes no operating current

## 10.2.3 RC OSCILLATOR

For timing insensitive applications, the “RC” device option offers additional cost savings. The RC oscillator frequency is a function of the supply voltage, the resistor (R<sub>EXT</sub>) and capacitor (C<sub>EXT</sub>) values, and the operating temperature. In addition to this, the oscillator frequency will vary from unit to unit due to normal process parameter variation. Furthermore, the difference in lead frame capacitance between package types will also affect the oscillation frequency, especially for low C<sub>EXT</sub> values. The user also needs to take into account variation due to tolerance of external R and C components used. Figure 10-4 shows how the R/C combination is connected to the PIC16CXXX.

**FIGURE 10-4: RC OSCILLATOR MODE**



## 10.3 Reset

The PIC16CXXX differentiates between various kinds of reset:

- Power-on Reset (POR)
- $\overline{\text{MCLR}}$  reset during normal operation
- $\overline{\text{MCLR}}$  reset during SLEEP
- WDT Reset (during normal operation)
- WDT Wake-up (during SLEEP)
- Brown-out Reset (BOR)

Some registers are not affected in any reset condition; their status is unknown on POR and unchanged by 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 on Brown-out Reset (BOR). They are not affected by a WDT Wake-up from SLEEP, which is viewed as the resumption of normal operation. The  $\overline{\text{TO}}$  and  $\overline{\text{PD}}$  bits are set or cleared depending on the reset situation, as indicated in Table 10-4. These bits are used in software to determine the nature of the reset. See Table 10-6 for a full description of reset states of all registers.

A simplified block diagram of the on-chip reset circuit is shown in Figure 10-5.

The PIC devices have a  $\overline{\text{MCLR}}$  noise filter in the  $\overline{\text{MCLR}}$  reset path. The filter will ignore small pulses. However, a valid  $\overline{\text{MCLR}}$  pulse must meet the minimum pulse width (T<sub>mcl</sub>, Specification #30).

No internal reset source (WDT, BOR, POR) will drive the  $\overline{\text{MCLR}}$  pin low.

**TABLE 10-6 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	62B	72A	xxxx xxxx	uuuu uuuu	uuuu uuuu
INDF	62B	72A	N/A	N/A	N/A
TMR0	62B	72A	xxxx xxxx	uuuu uuuu	uuuu uuuu
PCL	62B	72A	0000h	0000h	PC + 1 <sup>(2)</sup>
STATUS	62B	72A	0001 1xxx	000q quuu <sup>(3)</sup>	uuuq quuu <sup>(3)</sup>
FSR	62B	72A	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTA <sup>(4)</sup>	62B	72A	--0x 0000	--0u 0000	--uu uuuu
PORTB <sup>(5)</sup>	62B	72A	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTC <sup>(5)</sup>	62B	72A	xxxx xxxx	uuuu uuuu	uuuu uuuu
PCLATH	62B	72A	---0 0000	---0 0000	---u uuuu
INTCON	62B	72A	0000 000x	0000 000u	uuuu uuuu <sup>(1)</sup>
PIR1	62B	72A	---- 0000	---- 0000	---- uuuu <sup>(1)</sup>
	62B	72A	-0-- 0000	-0-- 0000	-u-- uuuu <sup>(1)</sup>
TMR1L	62B	72A	xxxx xxxx	uuuu uuuu	uuuu uuuu
TMR1H	62B	72A	xxxx xxxx	uuuu uuuu	uuuu uuuu
T1CON	62B	72A	--00 0000	--uu uuuu	--uu uuuu
TMR2	62B	72A	0000 0000	0000 0000	uuuu uuuu
T2CON	62B	72A	-000 0000	-000 0000	-uuu uuuu
SSPBUF	62B	72A	xxxx xxxx	uuuu uuuu	uuuu uuuu
SSPCON	62B	72A	0000 0000	0000 0000	uuuu uuuu
CCPR1L	62B	72A	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCPR1H	62B	72A	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCP1CON	62B	72A	--00 0000	--00 0000	--uu uuuu
ADRES	62B	72A	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADCON0	62B	72A	0000 00-0	0000 00-0	uuuu uu-u
OPTION_REG	62B	72A	1111 1111	1111 1111	uuuu uuuu
TRISA	62B	72A	--11 1111	--11 1111	--uu uuuu
TRISB	62B	72A	1111 1111	1111 1111	uuuu uuuu
TRISC	62B	72A	1111 1111	1111 1111	uuuu uuuu
PIE1	62B	72A	---- 0000	---- 0000	---- uuuu
	62B	72A	-0-- 0000	-0-- 0000	-u-- uuuu
PCON	62B	72A	---- --0q	---- --uq	---- --uq
PR2	62B	72A	1111 1111	1111 1111	1111 1111
SSPADD	62B	72A	0000 0000	0000 0000	uuuu uuuu
SSPSTAT	62B	72A	0000 0000	0000 0000	uuuu uuuu
ADCON1	62B	72A	---- -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 and/or PIR1 will be affected (to cause wake-up).

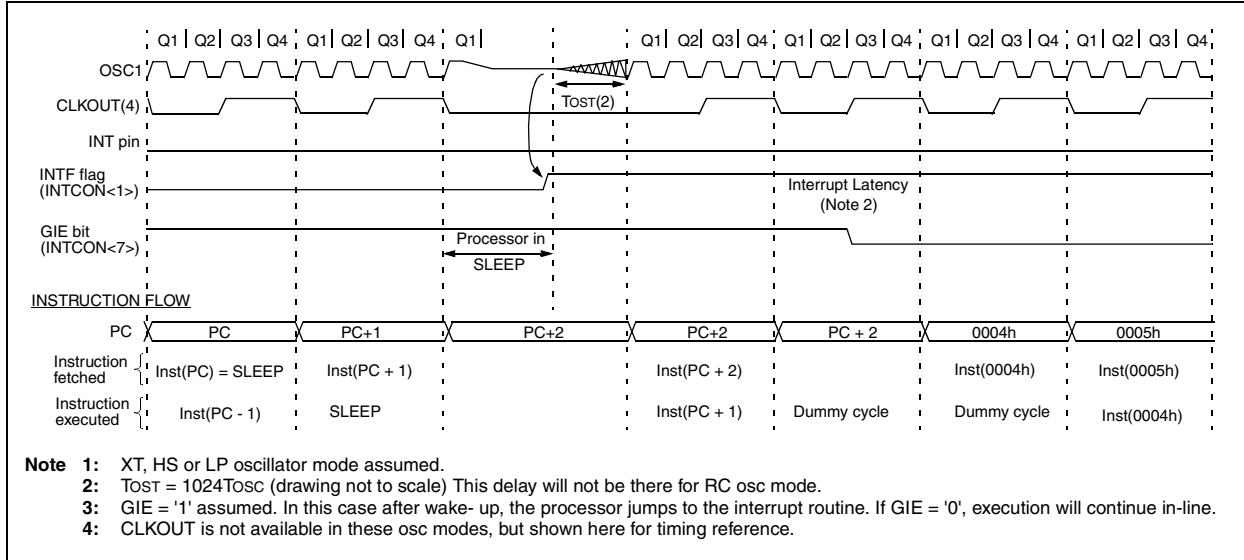
**2:** When the wake-up is due to an interrupt and the GIE bit is set, the PC is loaded with the interrupt vector (0004h).

**3:** See Table 10-5 for reset value for specific condition.

**4:** On any device reset, these pins are configured as inputs.

**5:** This is the value that will be in the port output latch.

**FIGURE 10-10: WAKE-UP FROM SLEEP THROUGH INTERRUPT**



## 10.14 Program Verification/Code Protection

If the code protection bits 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.

## 10.15 ID Locations

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

For ROM devices, these values are submitted along with the ROM code.

## 10.16 In-Circuit Serial Programming™

PIC16CXXX microcontrollers can be serially programmed while in the end application circuit. This is simply done with two lines for clock and data, and three more 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.

For complete details of serial programming, please refer to the In-Circuit Serial Programming (ICSP™) Guide, DS30277.

## 11.0 INSTRUCTION SET SUMMARY

Each PIC16CXXX 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 11-2 lists **byte-oriented**, **bit-oriented**, and **literal and control** operations. Table 11-1 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 11-1 OPCODE FIELD DESCRIPTIONS**

Field	Description
f	Register file address (0x00 to 0x7F)
w	Working register (accumulator)
b	Bit address within an 8-bit file register
k	Literal field, constant data or label
x	Don't care location (= 0 or 1) The assembler will generate code with x = 0. It is the recommended form of use for compatibility with all Microchip software tools.
d	Destination select; d = 0: store result in W, d = 1: store result in file register f. Default is d = 1
PC	Program Counter
TO	Time-out bit
PD	Power-down bit
Z	Zero bit
DC	Digit Carry bit
C	Carry bit

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  $\mu$ s. If a conditional test is true or the program counter is changed as a result of an instruction, the instruction execution time is 2  $\mu$ s.

Table 11-2 lists the instructions recognized by the MPASM assembler.

Figure 11-1 shows the general formats that the instructions can have.

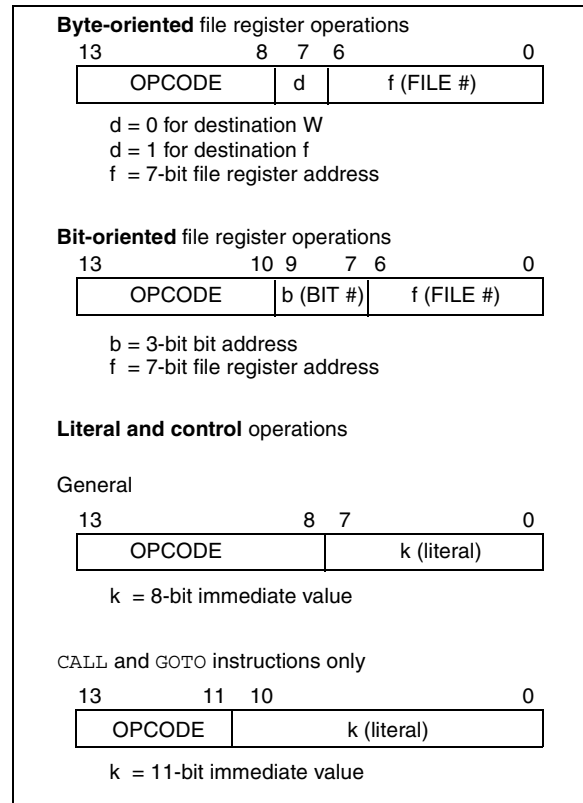
**Note:** To maintain upward compatibility with future PIC16CXXX products, do not use the **OPTION** and **TRIS** instructions.

All examples use the following format to represent a hexadecimal number:

0xhh

where h signifies a hexadecimal digit.

**FIGURE 11-1: GENERAL FORMAT FOR INSTRUCTIONS**



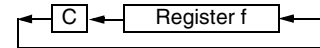
A description of each instruction is available in the PIC<sup>®</sup> MCU Mid-Range Reference Manual, (DS33023).

## RETFIE Return from Interrupt

Syntax: [ *label* ] RETFIE  
 Operands: None  
 Operation: TOS → PC,  
 1 → GIE  
 Status Affected: None

## RLF Rotate Left f through Carry

Syntax: [ *label* ] RLF f,d  
 Operands:  $0 \leq f \leq 127$   
 $d \in [0,1]$   
 Operation: See description below  
 Status Affected: C  
 Description: The contents of register 'f' are rotated one bit to the left through the Carry Flag. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is stored back in register 'f'.

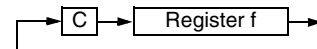


## RETLW Return with Literal in W

Syntax: [ *label* ] RETLW k  
 Operands:  $0 \leq k \leq 255$   
 Operation:  $k \rightarrow (W)$ ;  
 TOS → PC  
 Status Affected: None  
 Description: The W register is loaded with the eight bit literal 'k'. The program counter is loaded from the top of the stack (the return address). This is a two cycle instruction.

## RRF Rotate Right f through Carry

Syntax: [ *label* ] RRF f,d  
 Operands:  $0 \leq f \leq 127$   
 $d \in [0,1]$   
 Operation: See description below  
 Status Affected: C  
 Description: The contents of register 'f' are rotated one bit to the right through the Carry Flag. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is placed back in register 'f'.



## RETURN Return from Subroutine

Syntax: [ *label* ] RETURN  
 Operands: None  
 Operation: TOS → PC  
 Status Affected: None  
 Description: Return from subroutine. The stack is POPed and the top of the stack (TOS) is loaded into the program counter. This is a two cycle instruction.

## SLEEP

Syntax: [ *label* ] SLEEP  
 Operands: None  
 Operation: 00h → WDT,  
 0 → WDT prescaler,  
 $1 \rightarrow \overline{TO}$ ,  
 $0 \rightarrow \overline{PD}$   
 Status Affected:  $\overline{TO}$ ,  $\overline{PD}$   
 Description: The power-down status bit,  $\overline{PD}$  is cleared. Time-out status bit,  $\overline{TO}$  is set. Watchdog Timer and its prescaler are cleared. The processor is put into SLEEP mode with the oscillator stopped. See Section 10.13 for more details.

MPLIB is a librarian for pre-compiled code to be used with MPLINK. When a routine from a library is called from another source file, only the modules that contains that routine will be linked in with the application. This allows large libraries to be used efficiently in many different applications. MPLIB manages the creation and modification of library files.

MPLINK features include:

- MPLINK works with MPASM and MPLAB-C17 and MPLAB-C18.
- MPLINK allows all memory areas to be defined as sections to provide link-time flexibility.

MPLIB features include:

- MPLIB makes linking easier because single libraries can be included instead of many smaller files.
- MPLIB helps keep code maintainable by grouping related modules together.
- MPLIB commands allow libraries to be created and modules to be added, listed, replaced, deleted, or extracted.

## 12.5 MPLAB-SIM Software Simulator

The MPLAB-SIM Software Simulator allows code development in a PC host environment by simulating the PIC series microcontrollers on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a file or user-defined key press to any of the pins. The execution can be performed in single step, execute until break, or trace mode.

MPLAB-SIM fully supports symbolic debugging using MPLAB-C17 and MPLAB-C18 and MPASM. The Software Simulator offers the flexibility to develop and debug code outside of the laboratory environment making it an excellent multi-project software development tool.

## 12.6 MPLAB-ICE High Performance Universal In-Circuit Emulator with MPLAB IDE

The MPLAB-ICE Universal In-Circuit Emulator is intended to provide the product development engineer with a complete microcontroller design tool set for PIC microcontrollers (MCUs). Software control of MPLAB-ICE is provided by the MPLAB Integrated Development Environment (IDE), which allows editing, “make” and download, and source debugging from a single environment.

Interchangeable processor modules allow the system to be easily reconfigured for emulation of different processors. The universal architecture of the MPLAB-ICE allows expansion to support new PIC microcontrollers.

The MPLAB-ICE Emulator System has been designed as a real-time emulation system with advanced features that are generally found on more expensive devel-

opment tools. The PC platform and Microsoft® Windows 3.x/95/98 environment were chosen to best make these features available to you, the end user.

MPLAB-ICE 2000 is a full-featured emulator system with enhanced trace, trigger, and data monitoring features. Both systems use the same processor modules and will operate across the full operating speed range of the PIC MCU.

## 12.7 PICMASTER/PICMASTER CE

The PICMASTER system from Microchip Technology is a full-featured, professional quality emulator system. This flexible in-circuit emulator provides a high-quality, universal platform for emulating Microchip 8-bit PIC microcontrollers (MCUs). PICMASTER systems are sold worldwide, with a CE compliant model available for European Union (EU) countries.

## 12.8 ICEPIC

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

## 12.9 MPLAB-ICD In-Circuit Debugger

Microchip's In-Circuit Debugger, MPLAB-ICD, is a powerful, low-cost run-time development tool. This tool is based on the flash PIC16F877 and can be used to develop for this and other PIC microcontrollers from the PIC16CXXX family. MPLAB-ICD utilizes the In-Circuit Debugging capability built into the PIC16F87X. This feature, along with Microchip's In-Circuit Serial Programming protocol, offers cost-effective in-circuit flash programming and debugging from the graphical user interface of the MPLAB Integrated Development Environment. This enables a designer to develop and debug source code by watching variables, single-stepping and setting break points. Running at full speed enables testing hardware in real-time. The MPLAB-ICD is also a programmer for the flash PIC16F87X family.

## 12.10 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. PRO MATE II is CE compliant.

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 instructions and error messages, keys to enter commands and a modular detachable socket assembly to support various package types. In



stand-alone mode the PRO MATE II can read, verify or program PIC devices. It can also set code-protect bits in this mode.

## **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 supports all PIC devices with up to 40 pins. Larger pin count devices such as the PIC16C92X, and PIC17C76X may be supported with an adapter socket. PICSTART Plus is CE compliant.

## **SIMICE Entry-Level Hardware Simulator**

SIMICE is an entry-level hardware development system designed to operate in a PC-based environment with Microchip's simulator MPLAB-SIM. Both SIMICE and MPLAB-SIM run under Microchip Technology's MPLAB Integrated Development Environment (IDE) software. Specifically, SIMICE provides hardware simulation for Microchip's PIC12C5XX, PIC12CE5XX, and PIC16C5X families of PIC 8-bit microcontrollers. SIMICE works in conjunction with MPLAB-SIM to provide non-real-time I/O port emulation. SIMICE enables a developer to run simulator code for driving the target system. In addition, the target system can provide input to the simulator code. This capability allows for simple and interactive debugging without having to manually generate MPLAB-SIM stimulus files. SIMICE is a valuable debugging tool for entry-level system development.

## **PICDEM-1 Low-Cost PIC MCU 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-Plus programmer, and easily test firmware. The user can also connect the PICDEM-1 board to the MPLAB-ICE 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.

## **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-Plus, and easily test firmware. The MPLAB-ICE 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.

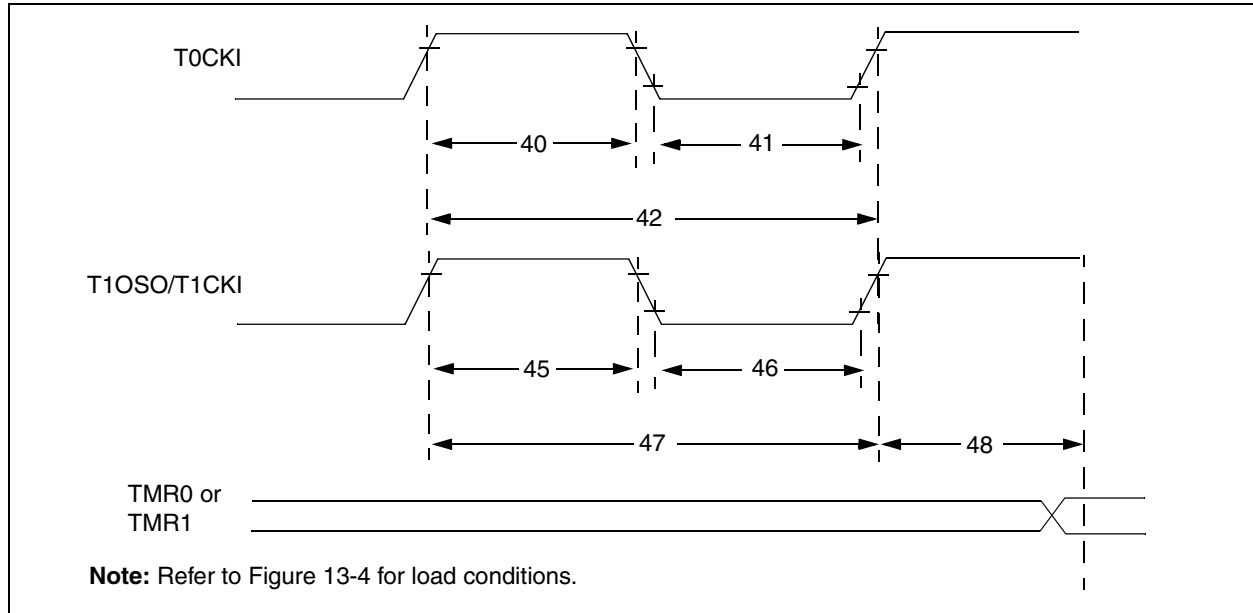
## **PICDEM-3 Low-Cost PIC16CXXX Demonstration Board**

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 MPLAB-ICE 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.

## **PICDEM-17**

The PICDEM-17 is an evaluation board that demonstrates the capabilities of several Microchip microcontrollers, including PIC17C752, PIC17C756, PIC17C762, and PIC17C766. All necessary hardware is included to run basic demo programs, which are supplied on a 3.5-inch disk. A programmed sample is included, and the user may erase it and program it with the other sample programs using the PRO MATE II or PICSTART Plus device programmers and easily debug

**FIGURE 13-9: TIMER0 AND TIMER1 EXTERNAL CLOCK TIMINGS**



**TABLE 13-5: TIMER0 AND TIMER1 EXTERNAL CLOCK REQUIREMENTS**

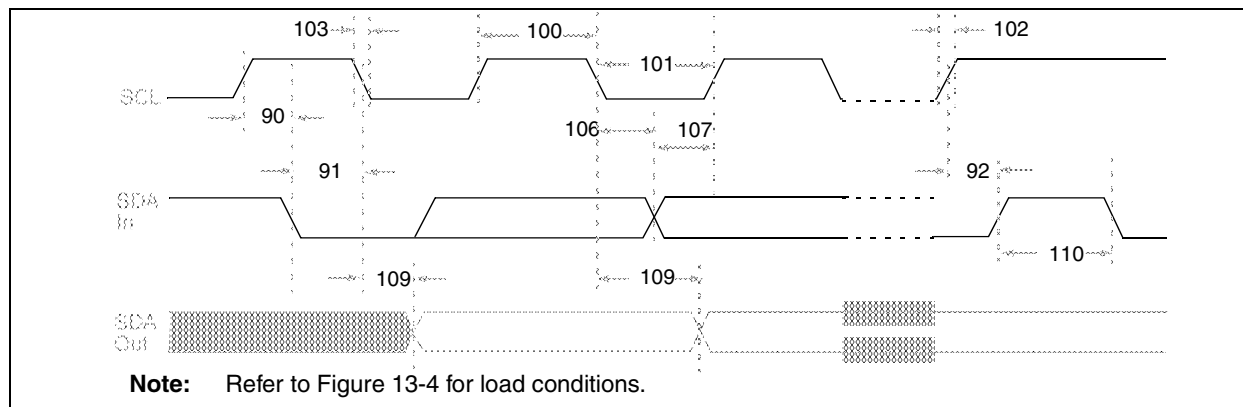
Param No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions	
40*	Tt0H	T0CKI High Pulse Width	No Prescaler	0.5Tcy + 20	—	—	ns	Must also meet parameter 42	
			With Prescaler	10	—	—	ns		
41*	Tt0L	T0CKI Low Pulse Width	No Prescaler	0.5Tcy + 20	—	—	ns	Must also meet parameter 42	
			With Prescaler	10	—	—	ns		
42*	Tt0P	T0CKI Period	No Prescaler	Tcy + 40	—	—	ns	N = prescale value (2, 4,..., 256)	
			With Prescaler	Greater of: 20 or $Tcy + 40 \over N$	—	—	ns		
45*	Tt1H	T1CKI High Time	Synchronous, Prescaler = 1	0.5Tcy + 20	—	—	ns	Must also meet parameter 47	
			Synchronous, Prescaler = 2,4,8	PIC16CXX	15	—	—		ns
				PIC16LCXX	25	—	—		ns
			Asynchronous	PIC16CXX	30	—	—		ns
				PIC16LCXX	50	—	—		ns
46*	Tt1L	T1CKI Low Time	Synchronous, Prescaler = 1	0.5Tcy + 20	—	—	ns	Must also meet parameter 47	
			Synchronous, Prescaler = 2,4,8	PIC16CXX	15	—	—		ns
				PIC16LCXX	25	—	—		ns
			Asynchronous	PIC16CXX	30	—	—		ns
				PIC16LCXX	50	—	—		ns
47*	Tt1P	T1CKI input period	Synchronous	PIC16CXX	GREATER OF: 30 OR $Tcy + 40 \over N$	—	—	ns	N = prescale value (1, 2, 4, 8)
				PIC16LCXX	GREATER OF: 50 OR $Tcy + 40 \over N$				N = prescale value (1, 2, 4, 8)
			Asynchronous	PIC16CXX	60	—	—	ns	
				PIC16LCXX	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		2Tosc	—	7Tosc	—		

\* 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.

# PIC16C62B/72A

**FIGURE 13-16: I<sup>2</sup>C BUS DATA TIMING**



**TABLE 13-12: I<sup>2</sup>C BUS DATA REQUIREMENTS**

Param. No.	Sym	Characteristic	Min	Max	Units	Conditions
100*	THIGH	Clock high time	100 kHz mode	4.0	—	μs
			400 kHz mode	0.6	—	μs
			SSP Module	1.5T <sub>CY</sub>	—	
101*	TLOW	Clock low time	100 kHz mode	4.7	—	μs
			400 kHz mode	1.3	—	μs
			SSP Module	1.5T <sub>CY</sub>	—	
102*	Tr	SDA and SCL rise time	100 kHz mode	—	1000	ns
			400 kHz mode	20 + 0.1Cb	300	ns
103*	Tf	SDA and SCL fall time	100 kHz mode	—	300	ns
			400 kHz mode	20 + 0.1Cb	300	ns
90*	TSU:STA	START condition setup time	100 kHz mode	4.7	—	μs
			400 kHz mode	0.6	—	μs
91*	THD:STA	START condition hold time	100 kHz mode	4.0	—	μs
			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
			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
			400 kHz mode	—	—	ns
110*	TBUF	Bus free time	100 kHz mode	4.7	—	μs
			400 kHz mode	1.3	—	μs
	Cb	Bus capacitive loading	—	400	pF	

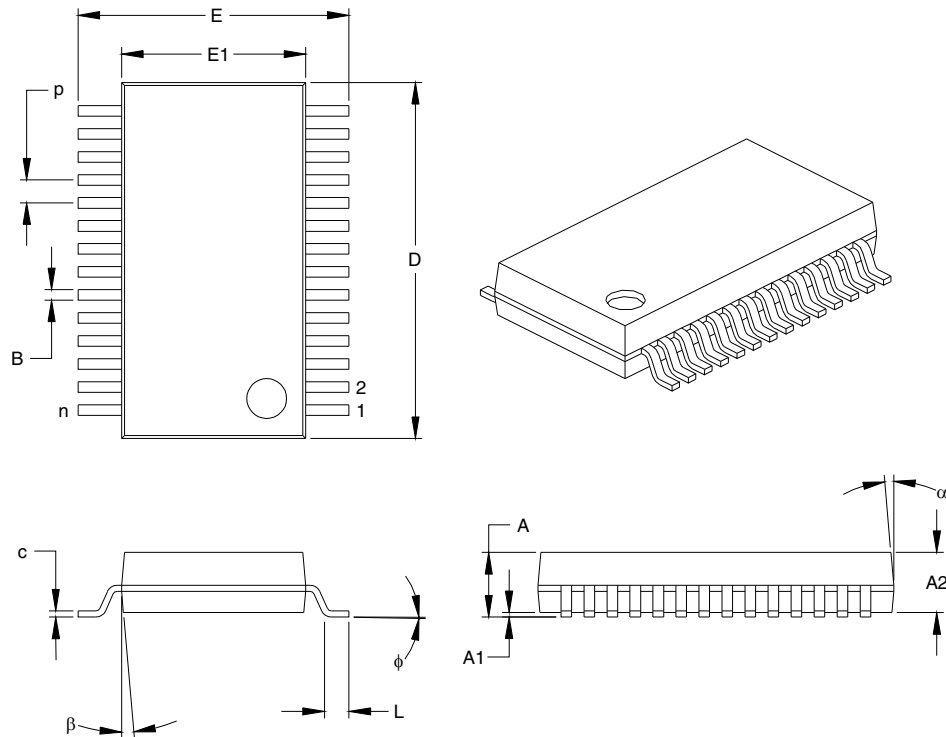
\* These parameters are characterized but not tested.

**Note 1:** As a transmitter, the device must provide this internal minimum delay time to bridge the undefined region (min. 300 ns) of the falling edge of SCL to avoid unintended generation of START or STOP conditions.

**2:** A fast-mode (400 kHz) I<sup>2</sup>C-bus device can be used in a standard-mode (100 kHz) I<sup>2</sup>C-bus system, but the requirement T<sub>su</sub>:DAT ≥ 250 ns must then be met. This will automatically be the case if the device does not stretch the LOW period of the SCL signal. If such a device does stretch the LOW period of the SCL signal, it must output the next data bit to the SDA line T<sub>r</sub> max.+t<sub>su</sub>:DAT = 1000 + 250 = 1250 ns (according to the standard-mode I<sup>2</sup>C bus specification) before the SCL line is released.

NOTES:

## 15.5 28-Lead Plastic Shrink Small Outline (SS) – 209 mil, 5.30 mm (SSOP)



Units		INCHES			MILLIMETERS*		
Dimension Limits		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		28			28	
Pitch	P		.026			0.66	
Overall Height	A	.068	.073	.078	1.73	1.85	1.98
Molded Package Thickness	A2	.064	.068	.072	1.63	1.73	1.83
Standoff	A1	.002	.006	.010	0.05	0.15	0.25
Overall Width	E	.299	.309	.319	7.59	7.85	8.10
Molded Package Width	E1	.201	.207	.212	5.11	5.25	5.38
Overall Length	D	.396	.402	.407	10.06	10.20	10.34
Foot Length	L	.022	.030	.037	0.56	0.75	0.94
Lead Thickness	c	.004	.007	.010	0.10	0.18	0.25
Foot Angle	φ	0	4	8	0.00	101.60	203.20
Lead Width	B	.010	.013	.015	0.25	0.32	0.38
Mold Draft Angle Top	α	0	5	10	0	5	10
Mold Draft Angle Bottom	β	0	5	10	0	5	10

\*Controlling Parameter

Notes:

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

JEDEC Equivalent: MS-150

Drawing No. C04-073