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	4MHz
Connectivity	I²C, SPI
Peripherals	LCD, POR, PWM, WDT
Number of I/O	25
Program Memory Size	7KB (4K x 14)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	176 x 8
Voltage - Supply (Vcc/Vdd)	4V ~ 6V
Data Converters	-
Oscillator Type	External
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	68-LCC (J-Lead)
Supplier Device Package	68-PLCC (24.23x24.23)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16c923-04-l

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)
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

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

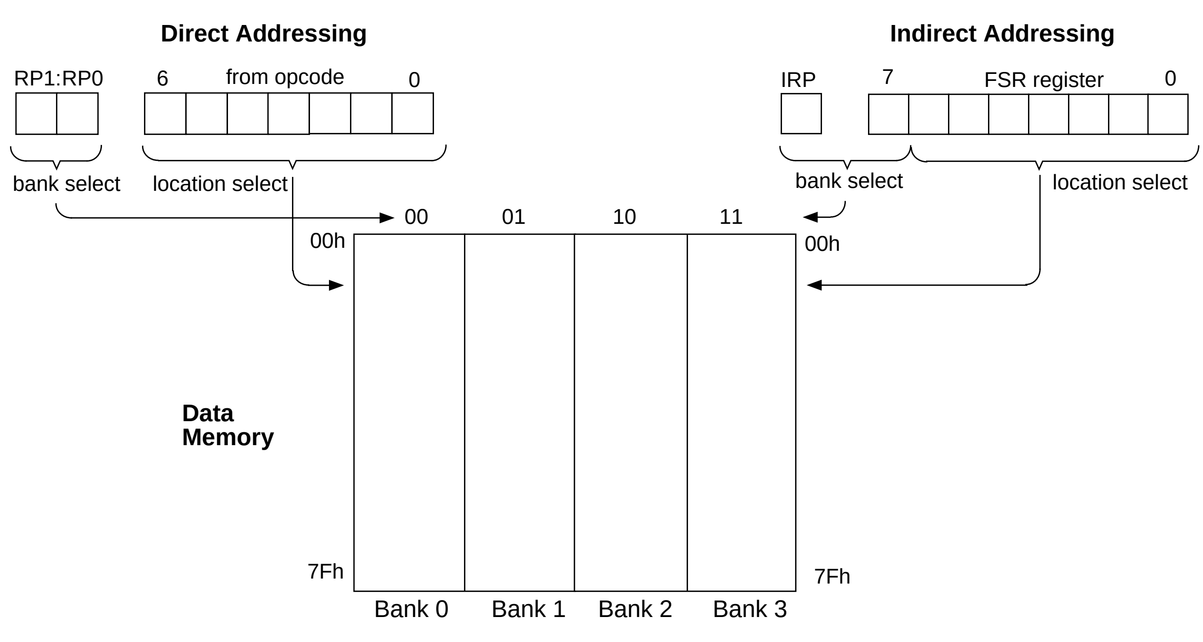
Indirect addressing is possible by using the INDF register. Any instruction using the INDF register actually accesses the register pointed to by the File Select Register (FSR). Reading the INDF register itself indirectly (FSR = '0') will produce 00h. Writing to the INDF register indirectly results in 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 4-10.

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 4-10: DIRECT/INDIRECT ADDRESSING



For memory map detail see Figure 4-2.

6.0 OVERVIEW OF TIMER MODULES

Each module can generate an interrupt to indicate that an event has occurred (e.g. 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

The Timer0 module is a simple 8-bit timer/counter. The clock source can be either the internal system clock (Fosc/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.

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

6.2 Timer1 Overview

Timer1 is a 16-bit timer/counter. The clock source can be either the internal system clock (Fosc/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. Timer1 oscillator is also one of the clock sources for the LCD module.

6.3 Timer2 Overview

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 clock source 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

The CCP module 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 CCPR1H:CCPR1L 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 CCP1 pin.

Compare mode compares the TMR1H:TMR1L register pair to the CCPR1H:CCPR1L register pair. When a match occurs an interrupt can be generated, and the output pin CCP1 can be forced to given state (High or Low), TMR1 can be reset and start A/D conversion. This depends on the control bits CCP1M3:CCP1M0.

PWM mode compares the TMR2 register to a 10-bit duty cycle register (CCPR1H:CCPR1L<5:4>) as well as to an 8-bit period register (PR2). When the TMR2 register = Duty Cycle register, the CCP1 pin will be forced low. When TMR2 = PR2, TMR2 is cleared to 00h, an interrupt can be generated, and the CCP1 pin (if an output) will be forced high.

7.3 Prescaler

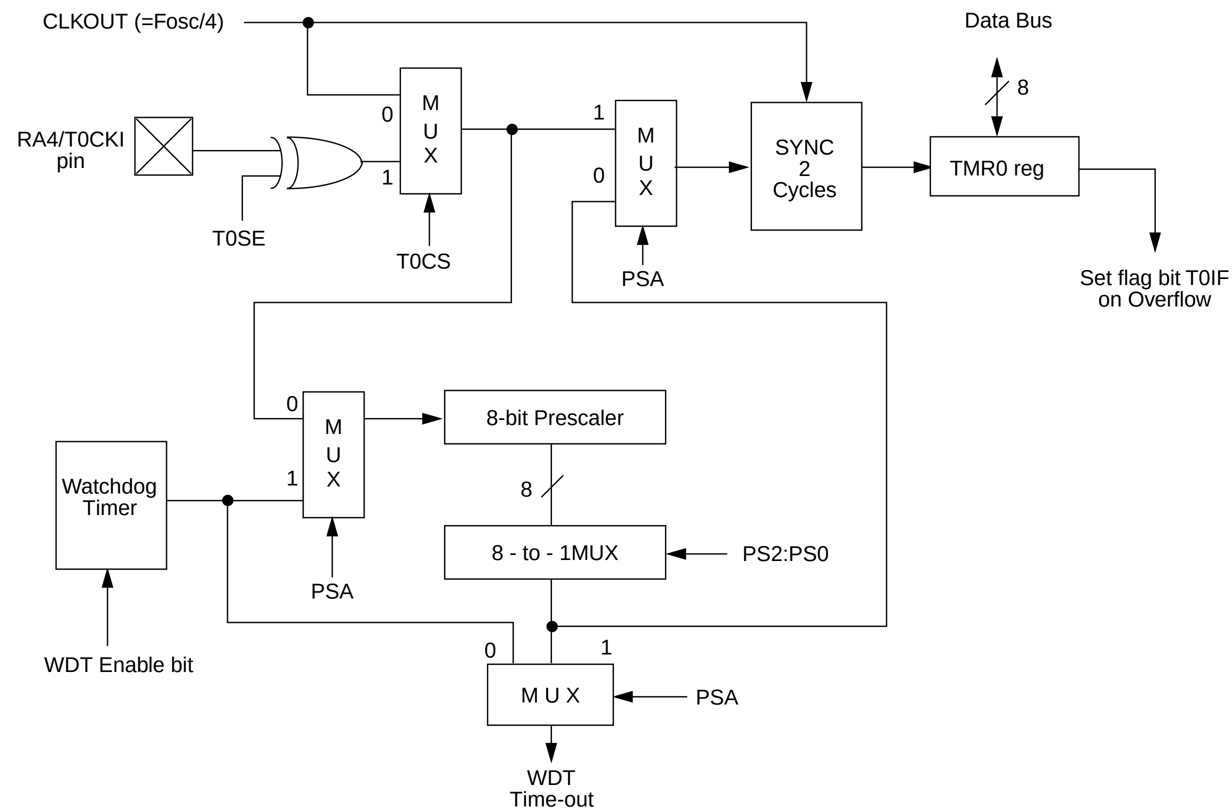
An 8-bit counter is available as a prescaler for the Timer0 module, or as a postscaler for the Watchdog Timer (Figure 7-6). For simplicity, this counter is being referred to as “prescaler” throughout this data sheet. Note that the prescaler may be used by either the Timer0 module or the WDT but not both. 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 count. When assigned to WDT, a `CLRWDT` instruction will clear the prescaler count 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



Note: T0CS, T0SE, PSA, PS2:PS0 are (OPTION<5:0>).

8.3 Timer1 Operation in Asynchronous Counter Mode

If control bit $\overline{\text{T1SYNC}}$ (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-from or write-to the Timer1 register pair (TMR1H:TMR1L) (Section 8.3.2).

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

8.3.1 EXTERNAL CLOCK INPUT TIMING WITH UNSYNCHRONIZED CLOCK

If control bit $\overline{\text{T1SYNC}}$ is set, the timer will increment completely asynchronously. The input clock must meet certain minimum high time and low time requirements, as specified in timing parameters 45, 46, and 47.

8.3.2 READING AND WRITING TMR1 IN ASYNCHRONOUS COUNTER MODE

Reading TMR1H or TMR1L while the timer is running, from an external asynchronous clock, will ensure a valid read (taken care of in hardware). However, the user should keep in mind that reading the 16-bit timer in two 8-bit values itself poses certain problems since the timer may overflow between the reads.

For writes, it is recommended that the user simply stop the timer and write the desired values. A write contention may occur by writing to the timer registers while the register is incrementing. This may produce an unpredictable value in the timer register.

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

8.4 Timer1 Oscillator

A crystal oscillator circuit is built in between pins T1OSI (input) and T1OSO (amplifier output). It is enabled by setting control bit T1OSCEN (T1CON<3>). The oscillator is a low power oscillator rated up to 200 kHz. It will continue to run during SLEEP. It is primarily intended for 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.

> **Note:** The Timer2 postscaler (Section 9.0) 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.

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 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:

PWM duty cycle = (CCPR1L:CCP1CON<5:4>) • Tosc • (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:

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

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

EXAMPLE 10-2: PWM PERIOD AND DUTY CYCLE CALCULATION

Desired PWM frequency is 31.25 kHz,
Fosc = 8 MHz
TMR2 prescale = 1

$1/31.25\text{ kHz} = [\,(PR2) + 1\,] \bullet 4 \bullet 1/8\text{ MHz} \bullet 1$

$32\ \mu s = [\,(PR2) + 1\,] \bullet 4 \bullet 125\text{ ns} \bullet 1$

$PR2 = 63$

Find the maximum resolution of the duty cycle that can be used with a 31.25 kHz frequency and 8 MHz oscillator:

$1/31.25\text{ kHz} = 2^{\text{PWM RESOLUTION}} \bullet 1/8\text{ MHz} \bullet 1$

$32\ \mu s = 2^{\text{PWM RESOLUTION}} \bullet 125\text{ ns} \bullet 1$

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

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

$8.0 = \text{PWM Resolution}$

At most, an 8-bit resolution duty cycle can be obtained from a 31.25 kHz frequency and a 8 MHz oscillator, i.e., 0 ≤ CCPR1L:CCP1CON<5:4> ≤ 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 10-2 lists example PWM frequencies and resolutions for Fosc = 8 MHz. 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 CCP module for PWM operation.

TABLE 10-2: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS AT 8 MHz

PWM Frequency	488 Hz	1.95 kHz	7.81 kHz	31.25 kHz	62.5 kHz	250 kHz
Timer Prescaler (1, 4, 16)	16	4	1	1	1	1
PR2 Value	0xFF	0xFF	0xFF	0x3F	0x1F	0x07
Maximum Resolution (bits)	10	10	10	8	7	5

NOTES:

11.1 SPI Mode

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
- Serial Clock (SCK) RC3/SCK

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

- Slave Select ($\overline{SS}$) RA5/AN4/$\overline{SS}$ (the AN4 function is implemented on the PIC16C924 only)

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 SSPIF (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 11-1 shows the loading of the SSPBUF (SSPSR) for data transmission. The shaded instruction is only required if the received data is meaningful.

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

```
     BCF   STATUS, RP1    ;Select Bank1
     BSF   STATUS, RP0    ;
LOOP BTFSS SSPSTAT, BF    ;Has data been
                          ;received
                          ;(transmit
                          ;complete)?
     GOTO  LOOP           ;No
     BCF   STATUS, RP0    ;Select Bank0
     MOVF  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 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)

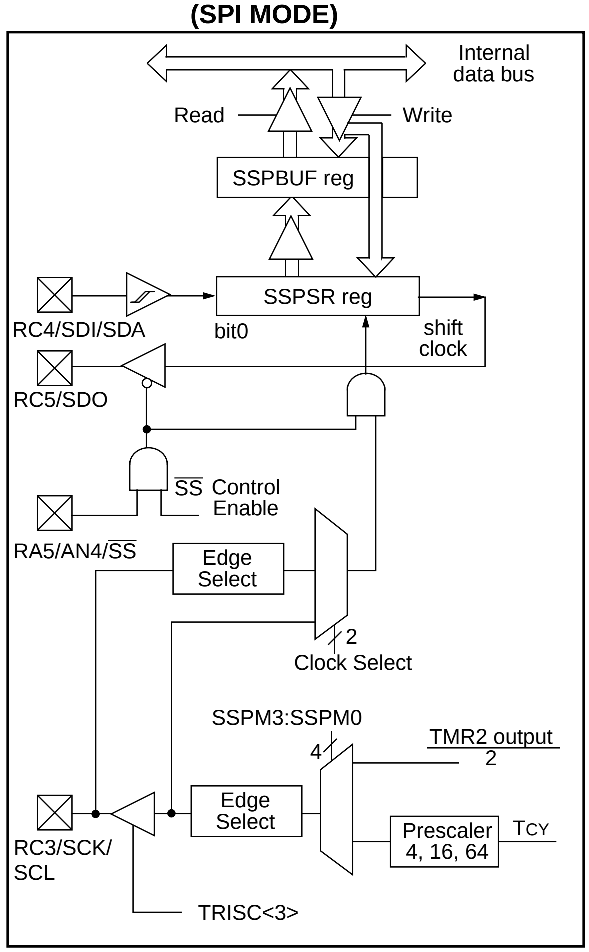


Figure 11-13 and Figure 11-14 show Master-transmitter and Master-receiver data transfer sequences.

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 (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 11-15.

FIGURE 11-13: MASTER-TRANSMITTER SEQUENCE

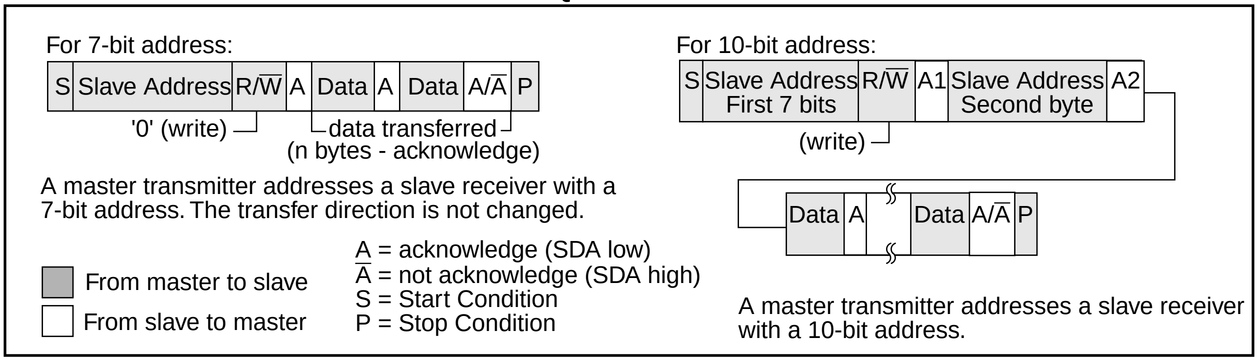


FIGURE 11-14: MASTER-RECEIVER SEQUENCE

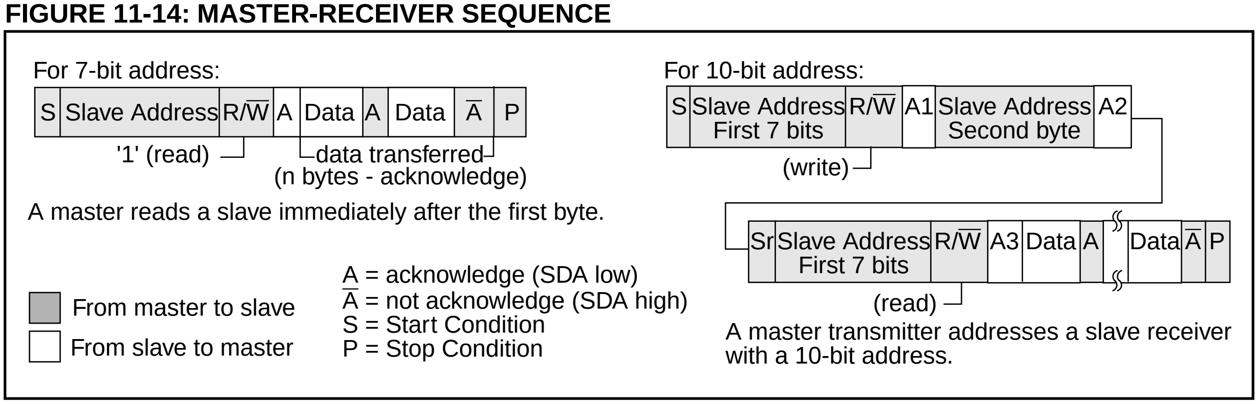
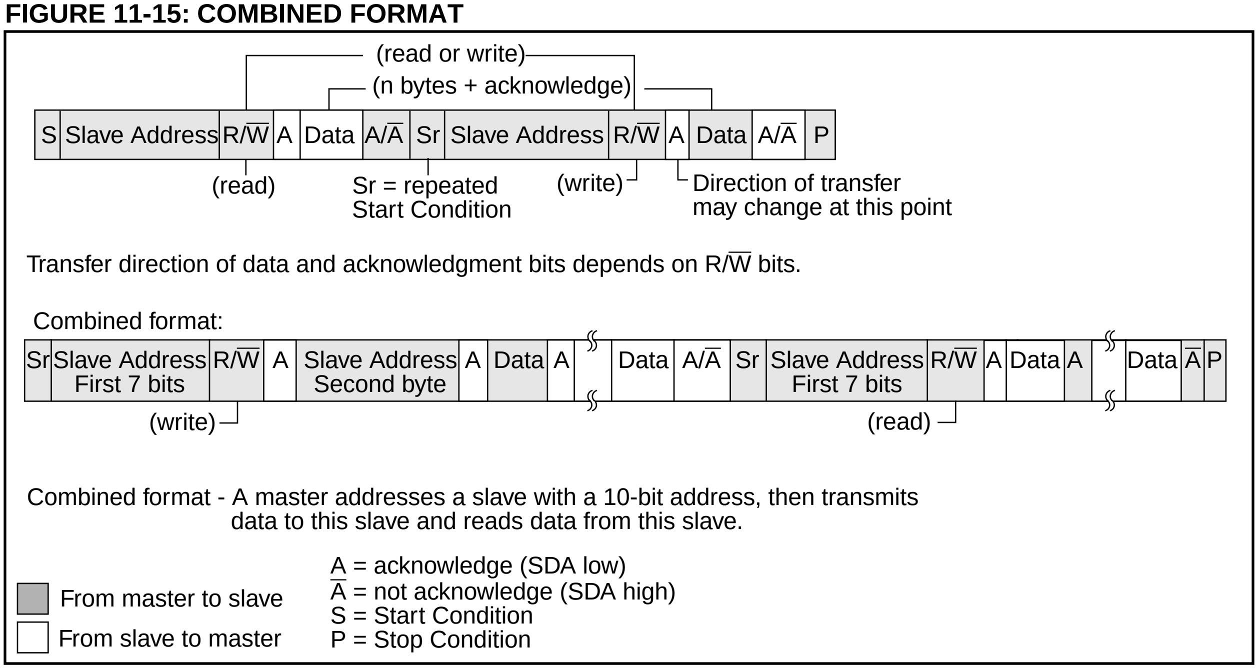


FIGURE 11-15: COMBINED FORMAT



11.3.2 MASTER MODE

Master mode of operation is supported, in firmware, using interrupt generation on the detection of the START and STOP conditions. The STOP (P) and START (S) bits are cleared from a reset or when the SSP module is disabled. The STOP and START bits will toggle based on the start and stop conditions. Control of the I^{2}C bus may be taken when the P bit is set, or the bus is idle with both the S and P bits clear.

In 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.

11.3.3 MULTI-MASTER MODE

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

In multi-master operation, the SDA line must be monitored 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, they 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 11-4: REGISTERS ASSOCIATED WITH I^{2}C OPERATION

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-on Reset	Value on all other resets
0Bh, 8Bh, 10Bh, 18Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	LCDIF	ADIF(1)	—	—	SSPIF	CCP1IF	TMR2IF	TMR1IF	00-- 0000	00-- 0000
8Ch	PIE1	LCDIE	ADIE(1)	—	—	SSPIE	CCP1IE	TMR2IE	TMR1IE	00-- 0000	00-- 0000
13h	SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register								xxxx xxxx	uuuu uuuu
93h	SSPADD	Synchronous Serial Port (I^{2}C mode) Address Register								0000 0000	0000 0000
14h	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	0000 0000
94h	SSPSTAT	SMP	CKE	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF	0000 0000	0000 0000
87h	TRISC	—	—	PORTC Data Direction Control Register						--11 1111	--11 1111

Legend: x = unknown, u = unchanged, - = unimplemented read as '0'. Shaded cells are not used by SSP in I^{2}C mode.
Note 1: Bits ADIE and ADIF are reserved on the PIC16C923, always maintain these bits clear.

12.1 A/D Acquisition Requirements

For the A/D converter to meet its specified accuracy, the charge holding capacitor (C_{HOLD}) must be allowed to fully charge to the input channel voltage level. The analog input model is shown in Figure 12-4. The source impedance (R_S) and the internal sampling switch (R_{SS}) impedance directly affect the time required to charge the capacitor C_{HOLD}. The sampling switch (R_{SS}) impedance varies over the device voltage (V_{DD}), (Figure 12-4). 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 kΩ**. After the analog input channel is selected (changed) this acquisition must be done before the conversion can be started.

To calculate the minimum acquisition time, Equation 12-1 may be used. This equation calculates the acquisition time to within 1/2 LSb error (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 12-1: A/D MINIMUM CHARGING TIME

$V_{HOLD} = (V_{REF} - (V_{REF}/512)) \bullet (1 - e^{(-T_C/C_{HOLD}(R_{IC} + R_{SS} + R_S))})$

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

The above equation reduces to:

$T_C = -(51.2\ pF)(1\ k\Omega - R_{SS} + R_S)\ \ln(1/511)$

Example 12-1 shows the calculation of the minimum required acquisition time (T_{ACQ}). This calculation is based on the following system assumptions.

$C_{HOLD} = 51.2$ pF

$R_s = 10$ kΩ

1/2 LSb error

$V_{DD} = 5V \rightarrow R_{ss} = 7$ kΩ

Temp (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 (C_{HOLD}) is not discharged after each conversion.

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

Note 4: After a conversion has completed, a 2.0 T_{AD} 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 12-1: CALCULATING THE MINIMUM REQUIRED SAMPLE TIME

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

$T_{ACQ} = 5\ \mu s + T_C + [(Temp - 25°C)(0.05\ \mu s/°C)]$

$T_C = -C_{HOLD}\ (R_{IC} + R_{SS} + R_S)\ \ln(1/511)$

$-51.2\ pF\ (1\ k\Omega + 7\ k\Omega + 10\ k\Omega)\ \ln(0.0020)$

$-51.2\ pF\ (18\ k\Omega)\ \ln(0.0020)$

$-0.921\ \mu s\ (-6.2364)$

$5.747\ \mu s$

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

$10.747\ \mu s + 1.25\ \mu s$

$11.997\ \mu s$

FIGURE 12-4: ANALOG INPUT MODEL

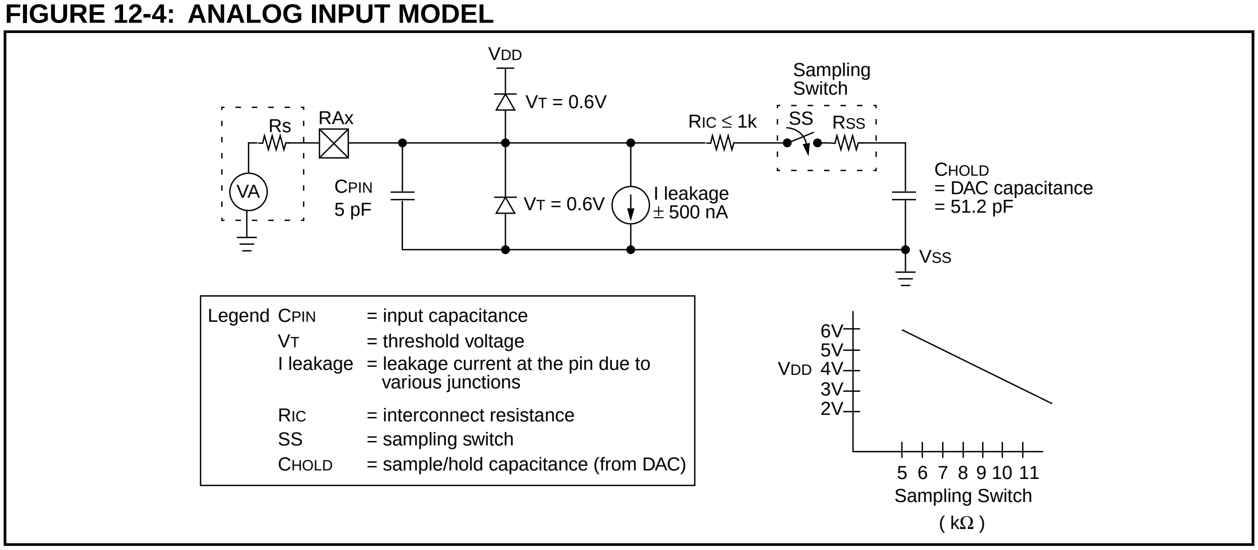


TABLE 14-6: INITIALIZATION CONDITIONS FOR ALL REGISTERS (Cont.'d)

Register	Applicable Devices		Power-on Reset	$\overline{\text{MCLR}}$ Resets WDT Reset	Wake-up via WDT or Interrupt
PORTD	923	924	0000 0000	0000 0000	uuuu uuuu
PORTE	923	924	0000 0000	0000 0000	uuuu uuuu
PCLATH	923	924	---0 0000	---0 0000	---u uuuu
INTCON	923	924	0000 000x	0000 000u	uuuu uuuu(1)
PIR1(4)	923	924	00-- 0000	00-- 0000	uu-- uuuu(1)
TMR1L	923	924	xxxx xxxx	uuuu uuuu	uuuu uuuu
TMR1H	923	924	xxxx xxxx	uuuu uuuu	uuuu uuuu
T1CON	923	924	--00 0000	--uu uuuu	--uu uuuu
TMR2	923	924	0000 0000	0000 0000	uuuu uuuu
T2CON	923	924	-000 0000	-000 0000	-uuu uuuu
SSPBUF	923	924	xxxx xxxx	uuuu uuuu	uuuu uuuu
SSPCON	923	924	0000 0000	0000 0000	uuuu uuuu
CCPR1L	923	924	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCPR1H	923	924	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCP1CON	923	924	--00 0000	--00 0000	--uu uuuu
ADRES	923	924	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADCON0	923	924	0000 00-0	0000 00-0	uuuu uu-u
OPTION	923	924	1111 1111	1111 1111	uuuu uuuu
TRISA	923	924	--11 1111	--11 1111	--uu uuuu
TRISB	923	924	1111 1111	1111 1111	uuuu uuuu
TRISC	923	924	--11 1111	--11 1111	--uu uuuu
TRISD	923	924	1111 1111	1111 1111	uuuu uuuu
TRISE	923	924	1111 1111	1111 1111	uuuu uuuu
PIE1(4)	923	924	00-- 0000	00-- 0000	uu-- uuuu
PCON	923	924	---- --0-	---- --u-	---- --u-
PR2	923	924	1111 1111	1111 1111	1111 1111
SSPADD	923	924	0000 0000	0000 0000	uuuu uuuu
SSPSTAT	923	924	0000 0000	0000 0000	uuuu uuuu
ADCON1	923	924	---- -000	---- -000	---- -uuu
PORTF	923	924	0000 0000	0000 0000	uuuu uuuu
PORTG	923	924	0000 0000	0000 0000	uuuu uuuu

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

Note 1: One or more bits in INTCON 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 14-5 for reset value for specific condition.

4: Bits PIE1<6> and PIR1<6> are reserved on the PIC16C923, always maintain these bits clear.

5: PORTA values when read.

FIGURE 14-8: TIME-OUT SEQUENCE ON POWER-UP ($\overline{MCLR}$ NOT TIED TO VDD): CASE 1

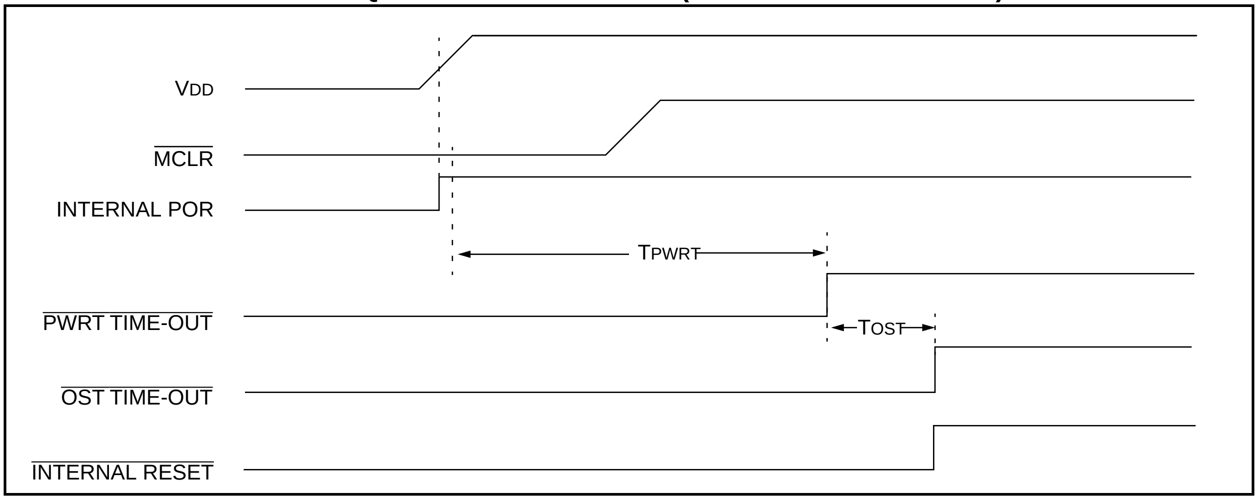


FIGURE 14-9: TIME-OUT SEQUENCE ON POWER-UP ($\overline{MCLR}$ NOT TIED TO VDD): CASE 2

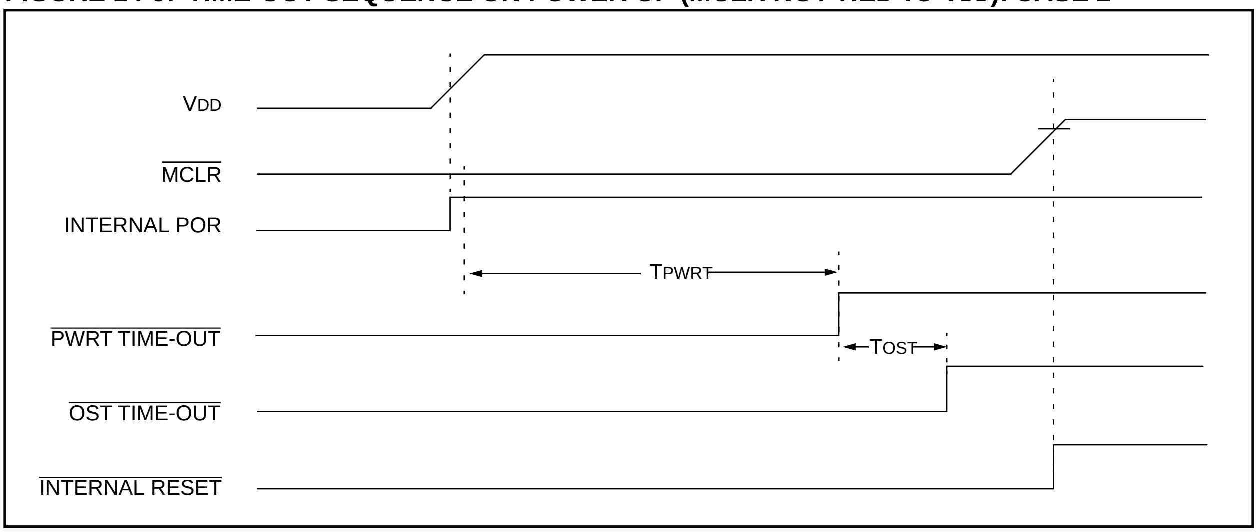
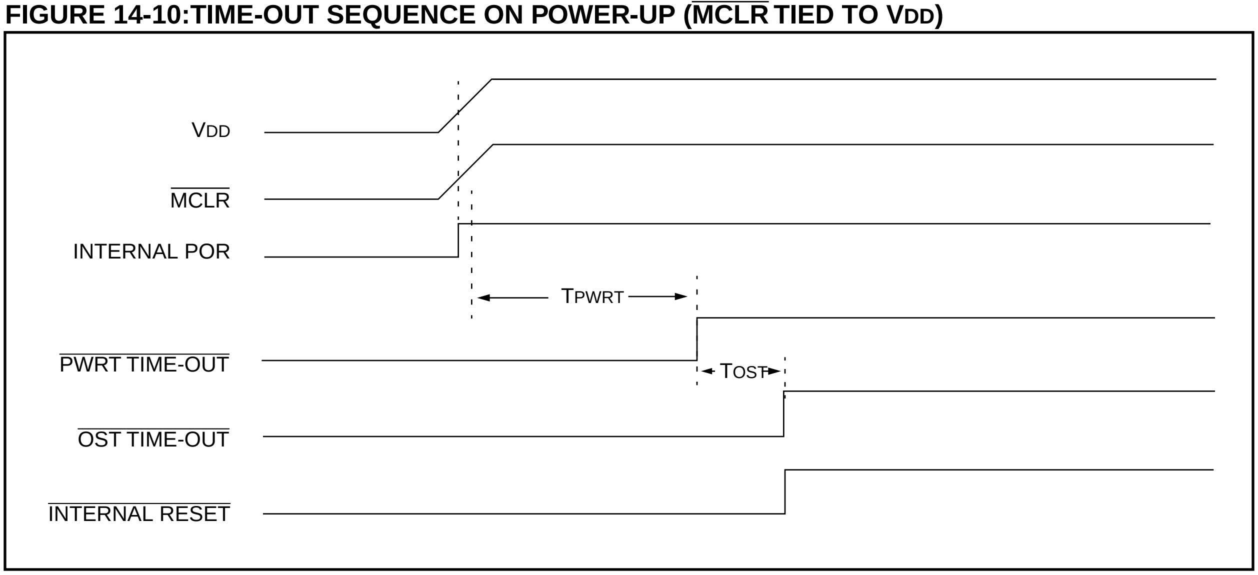


FIGURE 14-10: TIME-OUT SEQUENCE ON POWER-UP ($\overline{MCLR}$ TIED TO VDD)



15.0 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 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.

The instruction set is highly orthogonal and is grouped into three basic categories:

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

FIGURE 15-1: GENERAL FORMAT FOR INSTRUCTIONS

Byte-oriented file register operations

13 8	7	6 0
OPCODE	d	f (FILE #)

d = 0 for destination W
d = 1 for destination f
f = 7-bit file register address

Bit-oriented file register operations

13 10	9 7	6 0
OPCODE	b (BIT #)	f (FILE #)

b = 3-bit bit address
f = 7-bit file register address

Literal and control operations

General

13 8	7 0
OPCODE	k (literal)

k = 8-bit immediate value

`CALL` and `GOTO` instructions only

13 11	10 0
OPCODE	k (literal)

k = 11-bit immediate 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
$\overline{\texttt{TO}}$	Time-out bit
$\overline{\texttt{PD}}$	Power-down bit
`dest`	Destination either the W register or the specified register file location
[ ]	Options
( )	Contents
$\rightarrow$	Assigned to
< >	Register bit field
$\in$	In the set of
italics	User defined term (font is courier)

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 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.

17.1 DC Characteristics: PIC16C923/924-04 (Commercial, Industrial) PIC16C923/924-08 (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 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 Power-on Reset section for details
D004*	VDD rise rate to ensure internal Power-on Reset signal	SVDD	0.05	-	-	V/ms	(Note 6) See Power-on Reset section for details
D010	Supply Current (Note 2)	IDD	-	2.7	5	mA	XT and RC osc configuration FOSC = 4 MHz, VDD = 5.5V (Note 4)
D011			-	22.5	48	μA	LP osc configuration, FOSC = 32 kHz, VDD = 4.0V
D012			-	3.5	7	mA	HS osc configuration FOSC = 8 MHz, VDD = 5.5V
D020	Power-down Current (Note 3)	IPD	-	1.5	21	μA	VDD = 4.0V
	Module Differential Current (Note 5)						
D021	Watchdog Timer	ΔIWDT	-	6.0	20	μA	VDD = 4.0V
D022*	LCD Voltage Generation w/internal RC osc enabled	ΔILCDRC	-	40	55	μA	VDD = 4.0V (Note 7)
D024*	LCD Voltage Generation w/Timer1 @ 32.768 kHz	ΔILCDT1	-	33	60	μA	VDD = 4.0V (Note 7)
D025*	Timer1 oscillator	ΔIT1OSC	-	10.6	17	μA	VDD = 4.0V
D026*	A/D Converter	ΔIAD	-	1.0	-	μA	A/D on, not converting

* 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 in SLEEP mode 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
$\overline{\text{MCLR}}$ = VDD.

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.

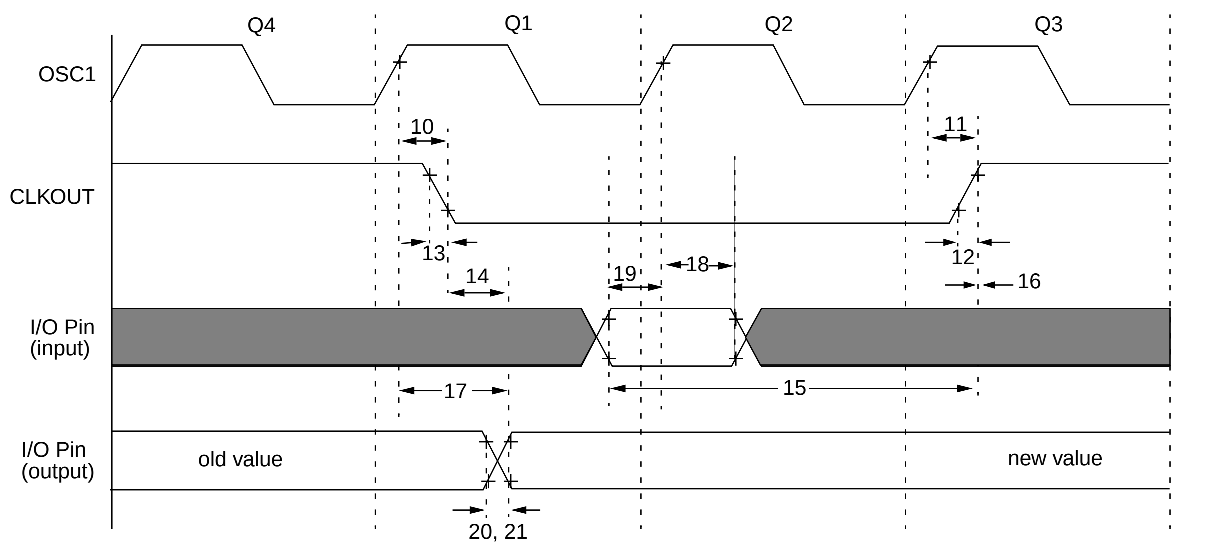
4: For RC osc configuration, current through Rext is not included. The current through the resistor can be estimated by the formula Ir = VDD/2Rext (mA) with Rext in kOhm.

5: The Δ current is the additional current consumed when this peripheral is enabled. This current should be added to the base IDD or IPD measurement.

6: PWRT must be enabled for slow ramps.

7: ΔILCDT1 and ΔILCDRC includes the current consumed by the LCD Module and the voltage generation circuitry. This does not include current dissipated by the LCD panel.

FIGURE 17-4: CLKOUT AND I/O TIMING



Refer to Figure 17-2 for load conditions.

TABLE 17-5: 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*	TckH2ioI	Port in hold after CLKOUT ↑		0	—	—	ns	Note 1
17*	TosH2ioV	OSC1↑ (Q1 cycle) to Port out valid		—	50	150	ns	
18*	TosH2ioI	OSC1↑ (Q2 cycle) to Port input invalid (I/O in hold time)	PIC16**C**923/924	100	—	—	ns	
			PIC16**LC**923/924	200	—	—	ns	
19*	TioV2osH	Port input valid to OSC1↑ (I/O in setup time)		0	—	—	ns	
20*	TioR	Port output rise time	PIC16**C**923/924	—	10	40	ns	
			PIC16**LC**923/924	—	—	80	ns	
21*	TioF	Port output fall time	PIC16**C**923/924	—	10	40	ns	
			PIC16**LC**923/924	—	—	80	ns	
22††*	Tinp	INT pin high or low time		TCY	—	—	ns	
23††*	Trbp	RB7:RB4 change INT high or low time		TCY	—	—	ns	

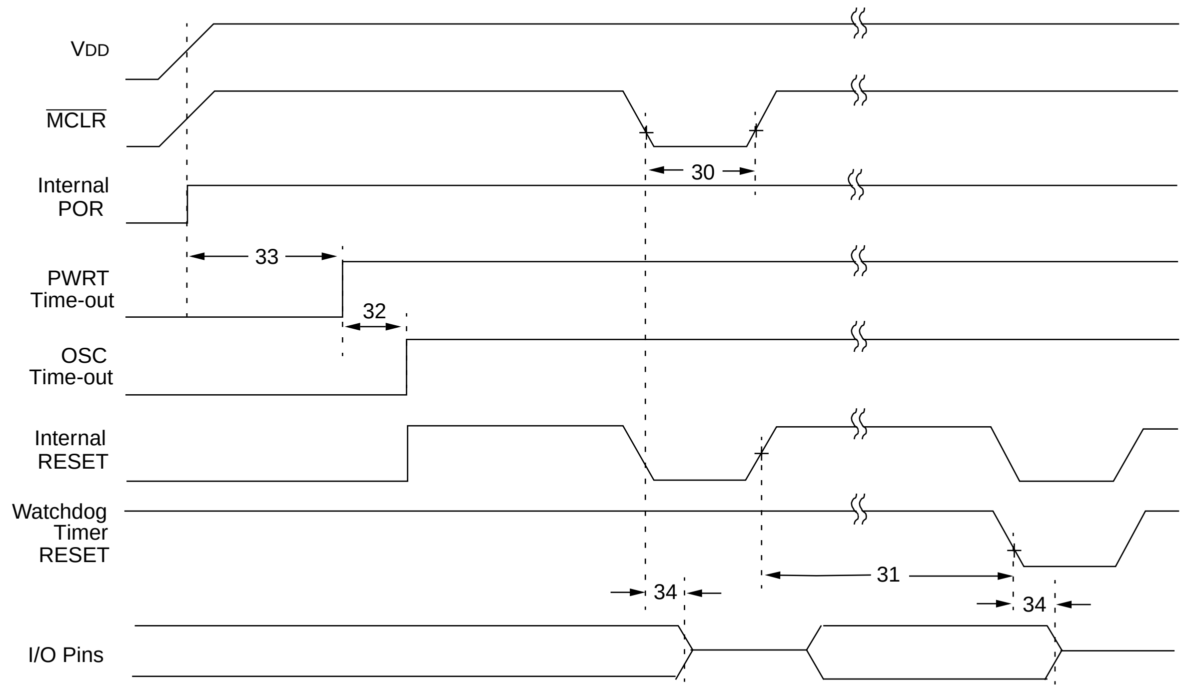
* These parameters are characterized but not tested.

† 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.

FIGURE 17-5: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING



Refer to Figure 17-2 for load conditions.

TABLE 17-6: 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)	2	—	—	µs	
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	—	—	2.1	µs	

* 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.

TABLE 17-9: SPI MODE REQUIREMENTS

Param No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions
70*	TssL2scH, TssL2scL	$\overline{SS}\downarrow$ to SCK$\downarrow$ or SCK$\uparrow$ input		TCY	—	—	ns	
71*	TscH	SCK input high time (slave mode)	Continuous	1.25TCY + 30	—	—	ns	
71A*			Single Byte	40	—	—	ns	
72*	TscL	SCK input low time (slave mode)	Continuous	1.25TCY + 30	—	—	ns	
72A*			Single Byte	40				
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	$\overline{SS}\uparrow$ 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		TCY	—	—	ns	
82*	TssL2doV	SDO data output valid after $\overline{SS}\downarrow$ edge		—	—	50	ns	
83*	TscH2ssH, TscL2ssH	$\overline{SS}\uparrow$ after SCK edge		1.5TCY + 40	—	—	ns	
84*	Tb2b	Delay between consecutive bytes		1.5TCY + 40	—	—	ns	

* 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.

APPENDIX C: WHAT'S NEW

Figure 13-13 (Resistor Ladder and Charge Pump) in LCD Section.

Parameter D150 - Open Drain High Voltage.

DC and AC Characterization Graphs and Tables.

APPENDIX D: WHAT'S CHANGED

Various descriptions for clarity.

Example code for Changing prescaler assignment between Timer0 and the WDT.

The A/D section has many changes that provide greater clarification of A/D operation.

The Instruction Set has Q-cycle activity listings for every instruction.

The following Electrical Characteristic Parameter values have changed to:

D011 (Standard Voltage Devices, C)

Typical	22.5	µA
Max	48	µA

D022 (Standard Voltage Devices)

Typical	40	µA
Max	55	µA

D024 (Standard Voltage Devices)

Typical	33	µA
Max	60	µA

D001 (Extended Voltage Devices, LC)

Min	2.5	V

D011 (Extended Voltage Devices, LC)

Typical	13.5	µA
Max	30	µA

D022 (Extended Voltage Devices, LC)

Typical	36	µA
Max	50	µA

D024 (Extended Voltage Devices, LC)

Typical	15	µA
Max	29	µA

D030 (with TTL)

Max	0.5VDD	V (ENTIRE RANGE)
Max	0.8V	V (4.5V ≤ VDD ≤ 5.5V)

D201, D202

Deleted D210 and D211, D251, D253, D260, D271

D222

Min	5	kHz
Typical	15	kHz
Max	50	kHz

D223, D224 - units to ns.

Added D265 (VLCDADJ voltage limits.

Changed parameters:

12 - TckR	35 ns Typical
13 - TckF	35 ns Typical
15 - TioV2ckH	Tosc + 200 ns Min
18 - TosH2ioL	200 ns Min (LC devices)
30 - TmcL	2 µs Min
34 - Tioz	2.1 µs Max

Timer0 and Timer1 External Clock Timings - Various.

53 - TccR,
54 - TccF

73 - TdiV2scH	50 ns Min
74 - TscH2diL	50 ns Min

Combined A/D specification tables for Standard and Extended Voltage devices.

PIC16C9XX 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

Pattern:	QTP, SQTP, ROM Code or Special Requirements	
Package:	SP	= 64-pin Shrink PDIP
	PT	= TQFP
	CL	= 68-pin Windowed CERQUAD
	L	= PLCC
Temperature Range:	-	= 0°C to +70°C (T for Tape/Reel)
	I	= -40°C to +85°C (S for Tape/Reel)
Frequency Range:	04	= 200 kHz (PIC16C9XX-04)
	04	= 4 MHz
	08	= 8 MHz
Device	PIC16C9XX	:V_{DD} range 4.0V to 6.0V
	PIC16C9XXT	:V_{DD} range 4.0V to 6.0V (Tape/Reel)
	PIC16LC9XX	:V_{DD} range 2.5V to 6.0V
	PIC16LC9XT	:V_{DD} range 2.5V to 6.0V (Tape/Reel)

Examples

a) PIC16C924 - 04/P 301 Commercial Temp., PDIP Package, 4 MHz, normal V_{DD} limits, QTP pattern #301

b) PIC16LC923 - 04/PT Commercial Temp., TQFP package, 4 MHz, extended V_{DD} limits

c) PIC16C923 - 08I/CL Industrial Temp., Windowed CERQUAD package, 8 MHz, normal V_{DD} limits

* CL Devices are UV erasable and can be programmed to any device configuration. CL 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. Your local Microchip sales office (see below)
2. The Microchip Corporate Literature Center U.S. FAX: (602) 786-7277
3. 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.

