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	I²C, SPI
Peripherals	Brown-out Detect/Reset, LCD, POR, PWM, WDT
Number of I/O	25
Program Memory Size	14KB (8K x 14)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	336 x 8
Voltage - Supply (Vcc/Vdd)	2.5V ~ 5.5V
Data Converters	A/D 5x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°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/pic16lc926-i-l

2.3 Special Function Registers

The Special Function Registers (SFRs) are registers used by the CPU and Peripheral Modules for controlling the desired operation of the device. These registers are implemented as static RAM.

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

TABLE 2-1: SPECIAL FUNCTION REGISTER SUMMARY

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-on Reset	Details on page
Bank 0											
00h	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	26
01h	TMR0	Timer0 Module Register								xxxx xxxx	41
02h	PCL	Program Counter (PC) Least Significant Byte								0000 0000	25
03h	STATUS	IRP	RP1	RP0	TO	PD	Z	DC	C	0001 1xxx	19
04h	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	26
05h	PORTA	—	—	PORTA Data Latch when written: PORTA pins when read						--0x 0000	29
06h	PORTB	PORTB Data Latch when written: PORTB pins when read								xxxx xxxx	31
07h	PORTC	—	—	PORTC Data Latch when written: PORTC pins when read						--xx xxxx	33
08h	PORTD	PORTD Data Latch when written: PORTD pins when read								0000 0000	34
09h	PORTE	PORTE pins when read								0000 0000	36
0Ah	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	25
0Bh	INTCON	GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBIF	0000 000x	21
0Ch	PIR1	LCDIF	ADIF	—	—	SSPIF	CCP1IF	TMR2IF	TMR1IF	00-- 0000	23
0Dh	—	Unimplemented								—	—
0Eh	TMR1L	Holding register for the Least Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	47
0Fh	TMR1H	Holding register for the Most Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	47
10h	T1CON	—	—	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON	--00 0000	47
11h	TMR2	Timer2 Module Register								0000 0000	51
12h	T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	52
13h	SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register								xxxx xxxx	64, 72
14h	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	60
15h	CCPR1L	Capture/Compare/PWM Register (LSB)								xxxx xxxx	58
16h	CCPR1H	Capture/Compare/PWM Register (MSB)								xxxx xxxx	58
17h	CCP1CON	—	—	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	53
18h	—	Unimplemented								—	—
19h	—	Unimplemented								—	—
1Ah	—	Unimplemented								—	—
1Bh	—	Unimplemented								—	—
1Ch	—	Unimplemented								—	—
1Dh	—	Unimplemented								—	—
1Eh	ADRESH	A/D Result Register High								xxxx xxxx	80, 81
1Fh	ADCON0	ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/$\overline{\text{DONE}}$	—	ADON	0000 0000	75

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

Note 1: These pixels do not display, but can be used as general purpose RAM.

2.3.2 OPTION REGISTER

The OPTION register is a readable and writable register, which contains various control bits to configure the TMR0/WDT prescaler, the external RB0/INT pin interrupt, TMR0, and the weak pull-ups on PORTB.

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

REGISTER 2-2: OPTION REGISTER (ADDRESS 81h, 181h)

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
bit 7							bit 0

bit 7 **$\overline{\text{RBPU}}$:** PORTB Pull-up Enable bit
1 = PORTB pull-ups are disabled
0 = PORTB pull-ups are enabled by individual port latch values

bit 6 **INTEDG**: Interrupt Edge Select bit
1 = Interrupt on rising edge of RB0/INT pin
0 = Interrupt on falling edge of RB0/INT pin

bit 5 **T0CS**: TMR0 Clock Source Select bit
1 = Transition on RA4/T0CKI pin
0 = Internal instruction cycle clock (CLKOUT)

bit 4 **T0SE**: TMR0 Source Edge Select bit
1 = Increment on high-to-low transition on RA4/T0CKI pin
0 = Increment on low-to-high transition on RA4/T0CKI pin

bit 3 **PSA**: Prescaler Assignment bit
1 = Prescaler is assigned to the WDT
0 = Prescaler is assigned to the Timer0 module

bit 2-0 **PS2:PS0**: Prescaler Rate Select bits

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

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

4.3 PORTC and TRISC Register

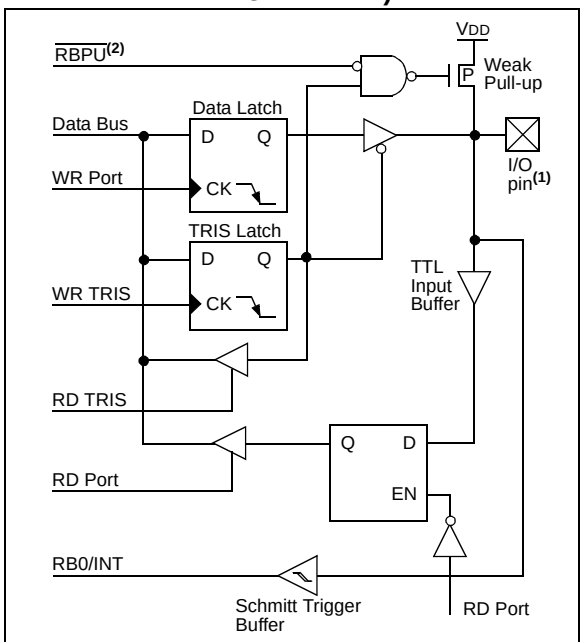
PORTC is a 6-bit, bi-directional port. Each pin is individually configurable as an input or output through the TRISC register. PORTC is multiplexed with several peripheral functions (Table 4-5). PORTC pins have Schmitt Trigger input buffers.

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

EXAMPLE 4-3: INITIALIZING PORTC

```
BCF     STATUS,RP0 ; Select Bank0
BCF     STATUS,RP1
CLRF    PORTC      ; Initialize PORTC
BSF     STATUS,RP0 ; Select Bank1
MOVLW   0xCF       ; Value used to
                   ; initialize data
                   ; direction
MOVWF   TRISC      ; Set RC<3:0> as inputs
                   ; RC<5:4> as outputs
                   ; RC<7:6> always read 0
```

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



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

2: To enable weak pull-ups, set the appropriate TRIS bit(s) and clear the RBPU bit (OPTION<7>).

TABLE 4-5: PORTC FUNCTIONS

Name	Bit#	Buffer Type	Function
RC0/T1OSO/T1CKI	bit0	ST	Input/output port pin or Timer1 oscillator output or Timer1 clock input.
RC1/T1OSI	bit1	ST	Input/output port pin or Timer1 oscillator input.
RC2/CCP1	bit2	ST	Input/output port pin or Capture input/Compare output/PWM output.
RC3/SCK/SCL	bit3	ST	Input/output port pin or the synchronous serial clock for both SPI and I^2C modes.
RC4/SDI/SDA	bit4	ST	Input/output port pin or the SPI Data In (SPI mode) or data I/O (I^2C mode).
RC5/SDO	bit5	ST	Input/output port pin or Synchronous Serial Port data out.

Legend: ST = Schmitt Trigger input

TABLE 4-6: SUMMARY OF REGISTERS ASSOCIATED WITH PORTC

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-on Reset	Value on all other RESETS
07h	PORTC	—	—	RC5	RC4	RC3	RC2	RC1	RC0	--xx xxxx	--uu uuuu
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 PORTC.

4.6 PORTF and TRISF Register

PORTF is a digital input only port. Each pin is multiplexed with an LCD segment driver. These pins have Schmitt Trigger input buffers.

Note 1: On a Power-on Reset, these pins are configured as LCD segment drivers.

2: To configure the pins as a digital port, the corresponding bits in the LCDSE register must be cleared. Any bit set in the LCDSE register overrides any bit settings in the corresponding TRIS register.

EXAMPLE 4-6: INITIALIZING PORTF

```
BCF STATUS,  RP0    ;Select Bank2
BSF STATUS,  RP1    ;
BCF LCDSE,   SE16   ;Make all PORTF
BCF LCDSE,   SE12   ;digital inputs
```

FIGURE 4-8: PORTF BLOCK DIAGRAM

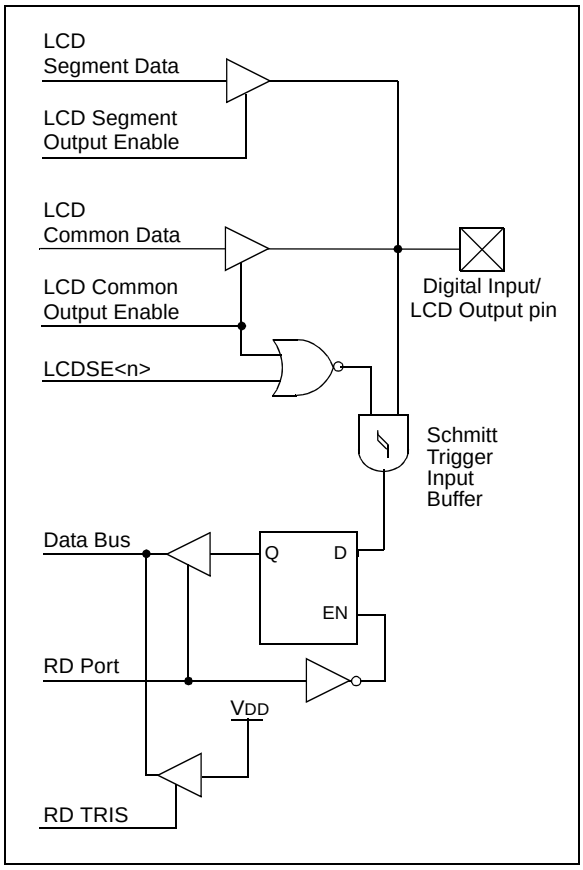


TABLE 4-11: PORTF FUNCTIONS

Name	Bit#	Buffer Type	Function
RF0/SEG12	bit0	ST	Digital input or Segment Driver12.
RF1/SEG13	bit1	ST	Digital input or Segment Driver13.
RF2/SEG14	bit2	ST	Digital input or Segment Driver14.
RF3/SEG15	bit3	ST	Digital input or Segment Driver15.
RF4/SEG16	bit4	ST	Digital input or Segment Driver16.
RF5/SEG17	bit5	ST	Digital input or Segment Driver17.
RF6/SEG18	bit6	ST	Digital input or Segment Driver18.
RF7/SEG19	bit7	ST	Digital input or Segment Driver19.

Legend: ST = Schmitt Trigger input

TABLE 4-12: SUMMARY OF REGISTERS ASSOCIATED WITH PORTF

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
107h	PORTF	RF7	RF6	RF5	RF4	RF3	RF2	RF1	RF0	0000 0000	0000 0000
187h	TRISF	PORTF Data Direction Control Register								1111 1111	1111 1111
10Dh	LCDSE	SE29	SE27	SE20	SE16	SE12	SE9	SE5	SE0	1111 1111	1111 1111

Legend: Shaded cells are not used by PORTF.

4.8 I/O Programming Considerations

4.8.1 BI-DIRECTIONAL I/O PORTS

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

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

Example 4-8 shows the effect of two sequential read-modify-write instructions on an I/O port. A pin actively outputting a Low or High should not be driven from external devices at the same time, in order to change the level on this pin ("wired-or", "wired-and"). The resulting high output currents may damage the chip.

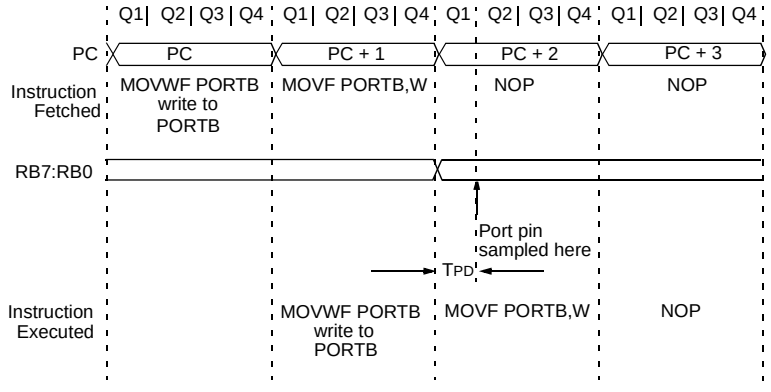
EXAMPLE 4-8: READ-MODIFY-WRITE INSTRUCTIONS ON AN I/O PORT

```
;Initial PORT settings: PORTB<7:4> Inputs
;                       PORTB<3:0> Outputs
;PORTB<7:6> have external pull-ups and are
;not connected to other circuitry
;
;                    PORT latch   PORT pins
;                    ----------   ---------
  BCF PORTB,  7   ; 01pp pppp    11pp pppp
  BCF PORTB,  6   ; 10pp pppp    11pp pppp
  BCF STATUS, RP1 ; Select Bank1
  BSF STATUS, RP0 ;
  BCF TRISB,  7   ; 10pp pppp    11pp pppp
  BCF TRISB,  6   ; 10pp pppp    10pp pppp
;
;Note that the user may have expected the
;pin values to be 00pp ppp. The 2nd BCF
;caused RB7 to be latched as the pin value
;(high).
```

4.8.2 SUCCESSIVE OPERATIONS ON I/O PORTS

The actual write to an I/O port happens at the end of an instruction cycle, whereas for reading, the data must be valid at the beginning of the instruction cycle (Figure 4-10). Therefore, care must be exercised if a write followed by a read operation is carried out on the same I/O port. The sequence of instructions should be such to allow the pin voltage to stabilize (load dependent) before the next instruction, which causes that file to be read into the CPU, is executed. Otherwise, the previous state of that pin may be read into the CPU, rather than the new state. When in doubt, it is better to separate these instructions with a NOP, or another instruction not accessing this I/O port.

FIGURE 4-10: SUCCESSIVE I/O OPERATION



Note:

This example shows a write to PORTB followed by a read from PORTB.

Note that:

data setup time = (0.25TCY - TPD)

where TCY = instruction cycle
TPD = propagation delay

Therefore, at higher clock frequencies, a write followed by a read may be problematic.

9.1 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/$\overline{SS}$

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 9-1 shows the loading of the SSPBUF (SSPSR) for data transmission. The `MOVWF RXDATA` instruction (shaded) is only required if the received data is meaningful.

EXAMPLE 9-1: 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 9-1), 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 9-1: SSP BLOCK DIAGRAM (SPI MODE)

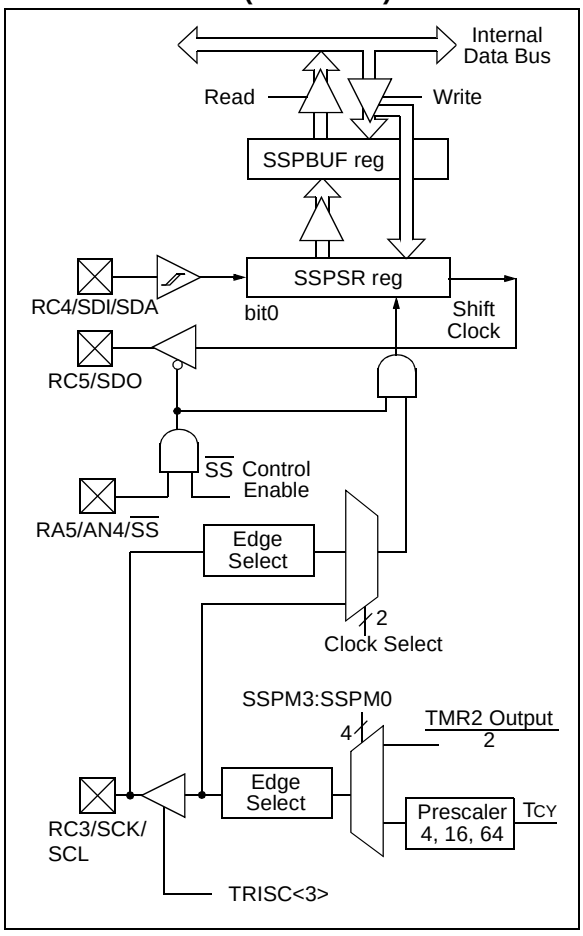


FIGURE 9-19: OPERATION OF THE I²C MODULE IN IDLE_MODE, RCV_MODE OR XMIT_MODE

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

10.3 Configuring Analog Port Pins

The ADCON1 and TRIS registers control the operation of the A/D port pins. The port pins that are desired as analog inputs must have their corresponding TRIS bits set (input). If the TRIS bit is cleared (output), the digital output level (V_{OH} or V_{OL}) will be converted.

The A/D operation is independent of the state of the CHS<2:0> bits and the TRIS bits.

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

2: Analog levels on any pin that is defined as a digital input (including the AN<4:0> pins), may cause the input buffer to consume current that is out of the device specifications.

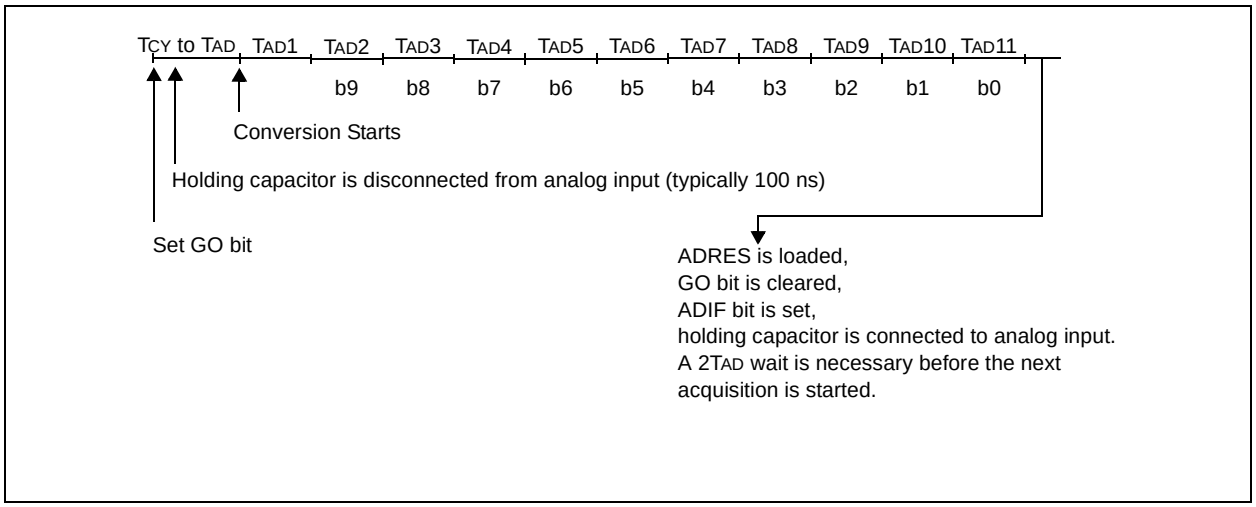
10.4 A/D Conversions

Clearing the GO/$\overline{\text{DONE}}$ bit during a conversion will abort the current conversion. The A/D result register pair will NOT be updated with the partially completed A/D conversion sample. That is, the ADRESH:ADRESL registers will continue to contain the value of the last completed conversion (or the last value written to the ADRESH:ADRESL registers). After the A/D conversion is aborted, a $2T_{AD}$ wait is required before the next acquisition is started. After this $2T_{AD}$ wait, acquisition on the selected channel is automatically started. After this, the GO/$\overline{\text{DONE}}$ bit can be set to start the conversion.

In Figure 10-3, after the GO bit is set, the first time segment has a minimum of T_{CY} and a maximum of T_{AD}.

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

FIGURE 10-3: A/D CONVERSION T_{AD} CYCLES



10.4.1 A/D RESULT REGISTERS

The ADRESH:ADRESL register pair is the location where the 10-bit A/D result is loaded at the completion of the A/D conversion. This register pair is 16-bits wide. The A/D module gives the flexibility to left or right justify the 10-bit result in the 16-bit result register. The A/D Format Select bit (ADFM) controls this justification. Figure 10-4 shows the operation of the A/D result justification. The extra bits are loaded with '0's'. When an A/D result will not overwrite these locations (A/D disable), these registers may be used as two general purpose 8-bit registers.

11.0 LCD MODULE

The LCD module generates the timing control to drive a static or multiplexed LCD panel, with support for up to 32 segments multiplexed with up to four commons. It also provides control of the LCD pixel data.

The interface to the module consists of 3 control registers (LCDCON, LCDSE, and LCDPS), used to define the timing requirements of the LCD panel and up to 16 LCD data registers (LCD00-LCD15) that represent the array of the pixel data. In normal operation, the control registers are configured to match the LCD panel being used. Primarily, the initialization information consists of selecting the number of commons required by the LCD panel, and then specifying the LCD frame clock rate to be used by the panel.

Once the module is initialized for the LCD panel, the individual bits of the LCD data registers are cleared/set to represent a clear/dark pixel, respectively.

Once the module is configured, the LCDEN (LCDCON<7>) bit is used to enable or disable the LCD module. The LCD panel can also operate during SLEEP by clearing the SLPEN (LCDCON<6>) bit.

Figure 11-2 through Figure 11-5 provides waveforms for static, half-duty cycle, one-third-duty cycle, and quarter-duty cycle drives.

REGISTER 11-1: LCDCON REGISTER (ADDRESS 10Fh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
LCDEN	SLPEN	WERR	BIAS	CS1	CS0	LMUX1	LMUX0
bit 7							bit 0

bit 7 **LCDEN**: Module Drive Enable bit
1 = LCD drive enabled
0 = LCD drive disabled

bit 6 **SLPEN**: LCD Display Enabled to SLEEP bit
1 = LCD module will stop driving in SLEEP
0 = LCD module will continue driving in SLEEP

bit 5 **WERR:** Write Failed Error bit
1 = System tried to write LCDD register during disallowed time. (Must be reset in software.)
0 = No error

bit 4 **BIAS**: Bias Generator Enable bit
0 = Internal bias generator powered down, bias is expected to be provided externally
1 = Internal bias generator enabled, powered up

bit 3-2 **CS<1:0>:** Clock Source bits
00 = FOSC/256
01 = T1CKI (Timer1)
1x = Internal RC oscillator

bit 1-0 **LMUX<1:0>:** Common Selection bits
Specifies the number of commons
00 = Static(COM0)
01 = 1/2 (COM0, 1)
10 = 1/3 (COM0, 1, 2)
11 = 1/4 (COM0, 1, 2, 3)

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

11.3 Pixel Control

11.3.1 LCDD (PIXEL DATA) REGISTERS

The pixel registers contain bits which define the state of each pixel. Each bit defines one unique pixel.

Table 11-4 shows the correlation of each bit in the LCDD registers to the respective common and segment signals.

Any LCD pixel location not being used for display can be used as general purpose RAM.

REGISTER 11-3: GENERIC LCDD REGISTER LAYOUT

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
SEGs COMc	SEGs COMc	SEGs COMc	SEGs COMc	SEGs COMc	SEGs COMc	SEGs COMc	SEGs COMc
bit 7							bit 0

bit 7-0 **SEGsCOMc**: Pixel Data bit for Segment S and Common C
1 = Pixel on (dark)
0 = Pixel off (clear)

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

11.6 Configuring the LCD Module

The following is the sequence of steps to follow to configure the LCD module.

1. Select the frame clock prescale using bits LP3:LP0 (LCDPS<3:0>).
2. Configure the appropriate pins to function as segment drivers using the LCDSE register.
3. Configure the LCD module for the following using the LCDCON register:
   - Multiplex mode and Bias, bits LMUX1:LMUX0
   - Timing source, bits CS1:CS0
   - Voltage generation, bit VGEN
   - SLEEP mode, bit SLPEN
4. Write initial values to pixel data registers, LCDD00 through LCDD15.
5. Clear LCD interrupt flag, LCDIF (PIR1<7>), and if desired, enable the interrupt by setting bit LCDIE (PIE1<7>).
6. Enable the LCD module, by setting bit LCDEN (LCDCON<7>).

TABLE 11-4: SUMMARY OF REGISTERS ASSOCIATED WITH THE LCD MODULE

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	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	LCDIF	ADIF	—	—	SSPIF	CCP1IF	TMR2IF	TMR1IF	00-- 0000	00-- 0000
8Ch	PIE1	LCDIE	ADIE	—	—	SSPIE	CCP1IE	TMR2IE	TMR1IE	00-- 0000	00-- 0000
10h	T1CON	—	—	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON	--00 0000	--uu uuuu
110h	LCDD00	SEG07 COM0	SEG06 COM0	SEG05 COM0	SEG04 COM0	SEG03 COM0	SEG02 COM0	SEG01 COM0	SEG00 COM0	xxxx xxxx	uuuu uuuu
111h	LCDD01	SEG15 COM0	SEG14 COM0	SEG13 COM0	SEG12 COM0	SEG11 COM0	SEG10 COM0	SEG09 COM0	SEG08 COM0	xxxx xxxx	uuuu uuuu
112h	LCDD02	SEG23 COM0	SEG22 COM0	SEG21 COM0	SEG20 COM0	SEG19 COM0	SEG18 COM0	SEG17 COM0	SEG16 COM0	xxxx xxxx	uuuu uuuu
113h	LCDD03	SEG31 COM0	SEG30 COM0	SEG29 COM0	SEG28 COM0	SEG27 COM0	SEG26 COM0	SEG25 COM0	SEG24 COM0	xxxx xxxx	uuuu uuuu
114h	LCDD04	SEG07 COM1	SEG06 COM1	SEG05 COM1	SEG04 COM1	SEG03 COM1	SEG02 COM1	SEG01 COM1	SEG00 COM1	xxxx xxxx	uuuu uuuu
115h	LCDD05	SEG15 COM1	SEG14 COM1	SEG13 COM1	SEG12 COM1	SEG11 COM1	SEG10 COM1	SEG09 COM1	SEG08 COM1	xxxx xxxx	uuuu uuuu
116h	LCDD06	SEG23 COM1	SEG22 COM1	SEG21 COM1	SEG20 COM1	SEG19 COM1	SEG18 COM1	SEG17 COM1	SEG16 COM1	xxxx xxxx	uuuu uuuu
117h	LCDD07	SEG31 COM1(1)	SEG30 COM1	SEG29 COM1	SEG28 COM1	SEG27 COM1	SEG26 COM1	SEG25 COM1	SEG24 COM1	xxxx xxxx	uuuu uuuu
118h	LCDD08	SEG07 COM2	SEG06 COM2	SEG05 COM2	SEG04 COM2	SEG03 COM2	SEG02 COM2	SEG01 COM2	SEG00 COM2	xxxx xxxx	uuuu uuuu
119h	LCDD09	SEG15 COM2	SEG14 COM2	SEG13 COM2	SEG12 COM2	SEG11 COM2	SEG10 COM2	SEG09 COM2	SEG08 COM2	xxxx xxxx	uuuu uuuu
11Ah	LCDD10	SEG23 COM2	SEG22 COM2	SEG21 COM2	SEG20 COM2	SEG19 COM2	SEG18 COM2	SEG17 COM2	SEG16 COM2	xxxx xxxx	uuuu uuuu
11Bh	LCDD11	SEG31 COM2(1)	SEG30 COM2(1)	SEG29 COM2	SEG28 COM2	SEG27 COM2	SEG26 COM2	SEG25 COM2	SEG24 COM2	xxxx xxxx	uuuu uuuu
11Ch	LCDD12	SEG07 COM3	SEG06 COM3	SEG05 COM3	SEG04 COM3	SEG03 COM3	SEG02 COM3	SEG01 COM3	SEG00 COM3	xxxx xxxx	uuuu uuuu
11Dh	LCDD13	SEG15 COM3	SEG14 COM3	SEG13 COM3	SEG12 COM3	SEG11 COM3	SEG10 COM3	SEG09 COM3	SEG08 COM3	xxxx xxxx	uuuu uuuu
11Eh	LCDD14	SEG23 COM3	SEG22 COM3	SEG21 COM3	SEG20 COM3	SEG19 COM3	SEG18 COM3	SEG17 COM3	SEG16 COM3	xxxx xxxx	uuuu uuuu
11Fh	LCDD15	SEG31 COM3(1)	SEG30 COM3(1)	SEG29 COM3(1)	SEG28 COM3	SEG27 COM3	SEG26 COM3	SEG25 COM3	SEG24 COM3	xxxx xxxx	uuuu uuuu
10Dh	LCDSE	SE29	SE27	SE20	SE16	SE12	SE9	SE5	SE0	1111 1111	1111 1111
10Eh	LCDPS	—	—	—	—	LP3	LP2	LP1	LP0	---- 0000	---- 0000
10Fh	LCDCON	LCDEN	SLPEN	—	VGEN	CS1	CS0	LMUX1	LMUX0	00-0 0000	00-0 0000

Legend: x = unknown, u = unchanged, - = unimplemented, read as '0'. Shaded cells are not used by the LCD module.
Note 1: These pixels do not display, but can be used as general purpose RAM.

12.2 Oscillator Configurations

12.2.1 OSCILLATOR TYPES

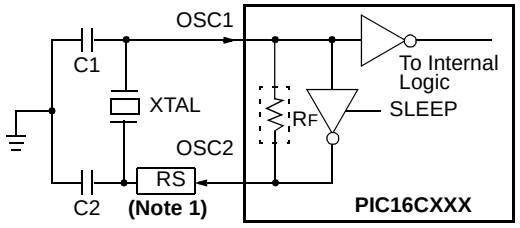
The PIC16CXXX can be operated in four different oscillator modes. The user can program two configuration bits (FOSC1 and FOSC0) to select one of these four modes:

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

12.2.2 CRYSTAL OSCILLATOR/CERAMIC RESONATORS

In XT, LP, or HS modes a crystal or ceramic resonator is connected to the OSC1/CLKIN and OSC2/CLKOUT pins to establish oscillation (Figure 12-1). The PIC16CXXX oscillator design requires the use of a parallel cut crystal. Use of a series cut crystal may give a frequency out of the crystal manufacturers specifications. When in XT, LP, or HS modes, the device can have an external clock source to drive the OSC1/CLKIN pin (Figure 12-2).

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



See Table 12-1 and Table 12-2 for recommended values of C1 and C2.

Note 1: A series resistor may be required for AT strip cut crystals.

FIGURE 12-2: EXTERNAL CLOCK INPUT OPERATION (HS, XT OR LP OSC CONFIGURATION)

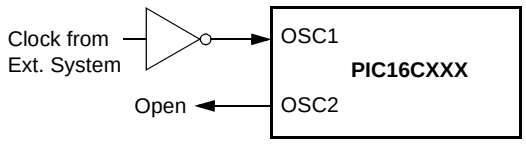


TABLE 12-1: CERAMIC RESONATORS

Ranges Tested:			
Mode	**Freq.**	**C1**	**C2**
XT	455 kHz	68 - 100 pF	68 - 100 pF
	2.0 MHz	15 - 68 pF	15 - 68 pF
	4.0 MHz	15 - 68 pF	15 - 68 pF
HS	8.0 MHz	10 - 68 pF	10 - 68 pF

These values are for design guidance only.
See notes following Table 12-2.

TABLE 12-2: CAPACITOR SELECTION FOR CRYSTAL OSCILLATOR

Osc Type	Crystal Freq.	Cap. Range C1	Cap. Range C2
LP	32 kHz	33 pF	33 pF
	200 kHz	15 pF	15 pF
XT	200 kHz	47-68 pF	47-68 pF
	1 MHz	15 pF	15 pF
	4 MHz	15 pF	15 pF
HS	4 MHz	15 pF	15 pF
	8 MHz	15-33 pF	15-33 pF

These values are for design guidance only.
See notes following this table.

Note 1: Recommended ranges of C1 and C2 are depicted in Table 12-1.

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

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

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

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

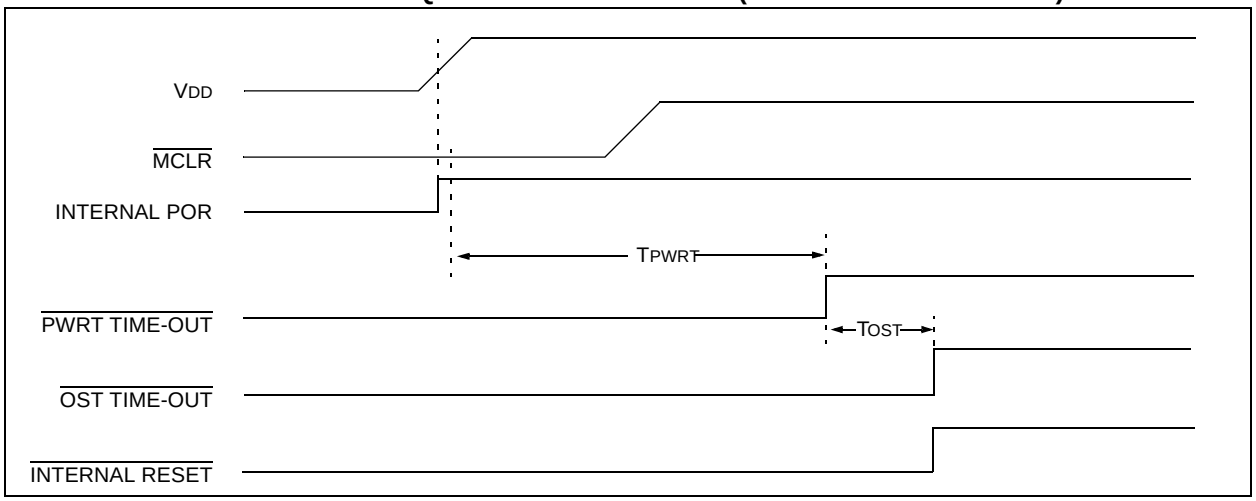


FIGURE 12-8: TIME-OUT SEQUENCE ON POWER-UP ($\overline{MCLR}$ NOT TIED TO V_{DD}): CASE 2

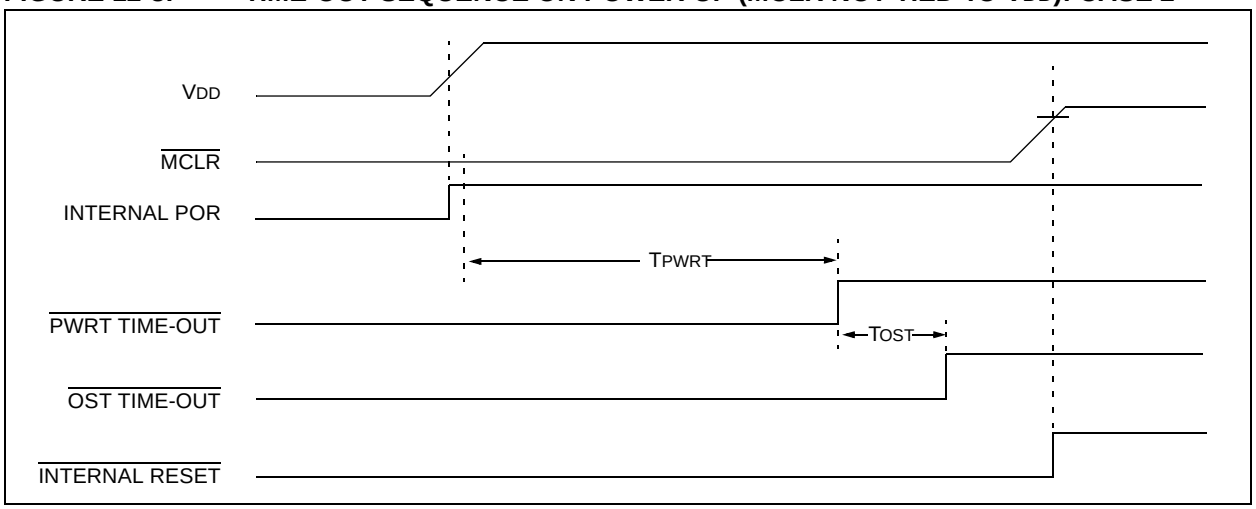


FIGURE 12-9: TIME-OUT SEQUENCE ON POWER-UP ($\overline{MCLR}$ TIED TO V_{DD})

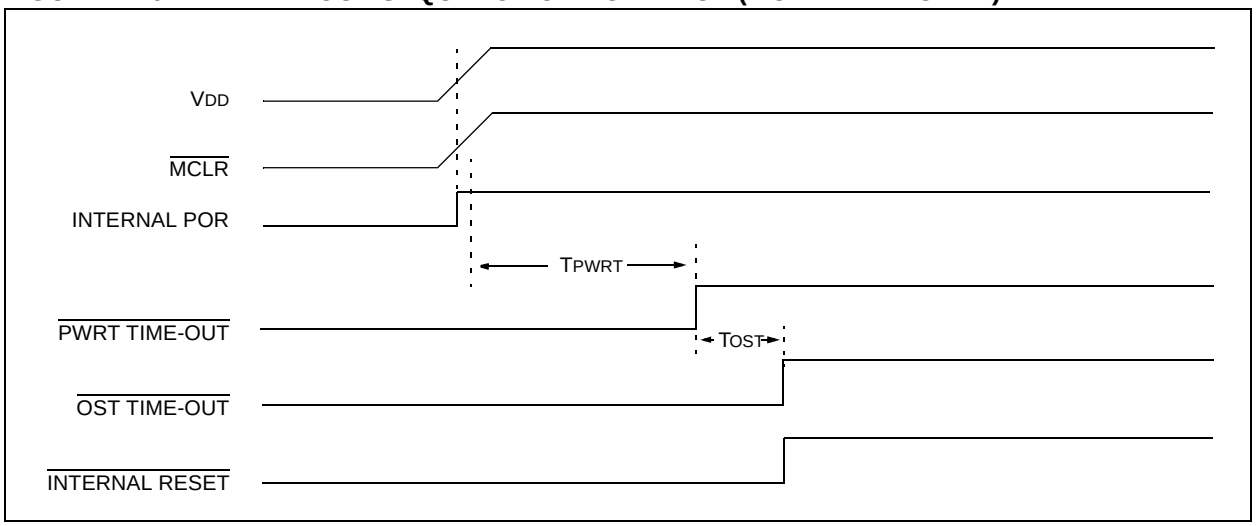
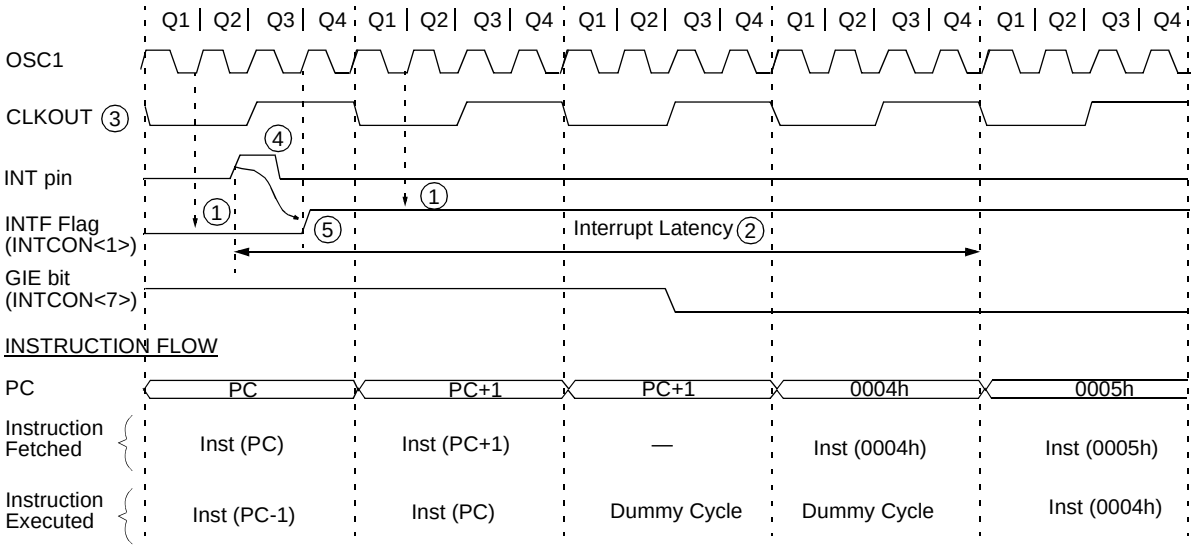


FIGURE 12-11: INT PIN INTERRUPT TIMING



Note 1: INTF flag is sampled here (every Q1).
2: Interrupt latency = 3-4 TCY where TCY = instruction cycle time. Latency is the same whether Inst (PC) is a single cycle or a 2-cycle instruction.
3: CLKOUT is available only in RC oscillator mode.
4: For minimum width of INT pulse, refer to AC specs.
5: INTF can be set any time during the Q4-Q1 cycles.

12.5.1 INT INTERRUPT

External interrupt on RB0/INT pin is edge triggered: either rising if bit INTEDG (OPTION_REG<6>) is set, or falling, if the INTEDG bit is clear. When a valid edge appears on the RB0/INT pin, flag bit INTF (INTCON<1>) is set. This interrupt can be disabled by clearing enable bit INTE (INTCON<4>). Flag bit INTF must be cleared in software in the Interrupt Service Routine before re-enabling this interrupt. The INT interrupt can wake-up the processor from SLEEP, if bit INTE was set prior to going into SLEEP. The status of global interrupt enable bit, GIE, decides whether or not the processor branches to the interrupt vector following wake-up. See Section 12.8 for details on SLEEP mode.

12.5.2 TMR0 INTERRUPT

An overflow (FFh → 00h) in the TMR0 register will set flag bit, TMR0IF (INTCON<2>). The interrupt can be enabled/disabled by setting/clearing enable bit, TMR0IE (INTCON<5>) (Section 5.0).

12.5.3 PORTB INTCON CHANGE

An input change on PORTB<7:4> sets flag bit RBIF (INTCON<0>). The interrupt can be enabled/disabled by setting/clearing enable bit, RBIE (INTCON<4>) (Section 4.2).

CLRF **Clear f**

Syntax: [*label*] CLRF f

Operands: $0 \le f \le 127$

Operation: 00h → (f)
1 → Z

Status Affected: Z

Encoding:

00	0001	1fff	ffff

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

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process data	Write register 'f'

Example `CLRF FLAG_REG`

Before Instruction:
FLAG_REG = 0x5A

After Instruction:
FLAG_REG = 0x00
Z = 1

CLRW **Clear W**

Syntax: [*label*] CLRW

Operands: None

Operation: 00h → (W)
1 → Z

Status Affected: Z

Encoding:

00	0001	0xxx	xxxx

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

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	No Operation	Process data	Write to W

Example `CLRW`

Before Instruction:
W = 0x5A

After Instruction:
W = 0x00
Z = 1

RETURN **Return from Subroutine**

Syntax: [*label*] RETURN

Operands: None

Operation: TOS → PC

Status Affected: None

Encoding:

00	0000	0000	1000

Description: Return from subroutine. The stack is POPed and the top of the stack (TOS) is loaded into the program counter. This is a two-cycle instruction.

Words: 1

Cycles: 2

Q Cycle Activity:

	Q1	Q2	Q3	Q4
1st Cycle	Decode	No Operation	No Operation	Pop from the Stack
2nd Cycle	No Operation	No Operation	No Operation	No Operation

Example `RETURN`

After Interrupt:
PC = TOS

RLF **Rotate Left f through Carry**

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

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

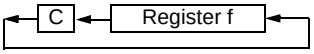
Operation: See description below

Status Affected: C

Encoding:

00	1101	dfff	ffff

Description: The contents of register 'f' are rotated one bit to the left through the Carry Flag. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is stored back in register 'f'.



Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process data	Write to destination

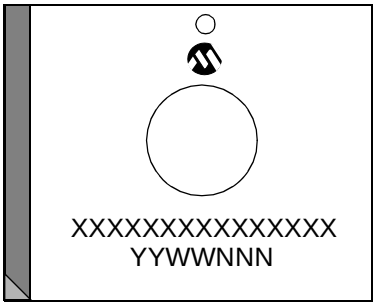
Example
```
RLF     REG1,0
```
Before Instruction:
```
REG1 =  1110 0110
C    =  0
```
After Instruction:
```
REG1 =  1110 0110
W    =  1100 1100
C    =  1
```

16.0 DC AND AC CHARACTERISTICS GRAPHS AND TABLES

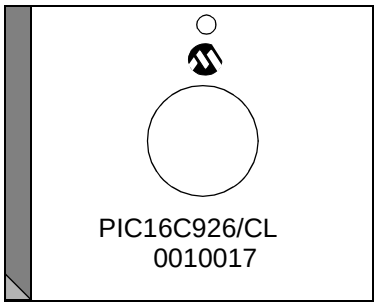
Graphs and Tables are not available at this time.

Package Marking Information (Continued)

68-Lead CERQUAD Windowed



Example



APPENDIX C: CONVERSION CONSIDERATIONS

Considerations for converting to the devices listed in this data sheet from previous device types are summarized in Table C-1.

TABLE C-1: CONVERSION CONSIDERATIONS

Feature	PIC16C923/924	PIC16C925/926
Operating Frequency	DC - 8 MHz	DC - 20 MHz
EPROM Program Memory (words)	4K	4K (925) 8K (926)
Data Memory (bytes)	176	176 (925) 336 (926)
A/D Converter Resolution	8-bit (924 only)	10-bit
A/D Converter Channels	none (923) 5 (924)	5
Interrupt Sources	8 (923) 9 (924)	9
Brown-out Reset	No	Yes

