



Welcome to [E-XFL.COM](https://www.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](#)"

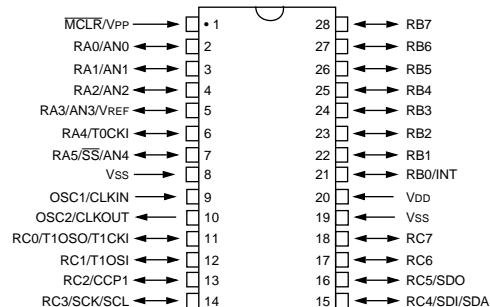
Details

Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	4MHz
Connectivity	I ² C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	22
Program Memory Size	7KB (4K x 14)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	192 x 8
Voltage - Supply (Vcc/Vdd)	2.5V ~ 6V
Data Converters	A/D 5x8b
Oscillator Type	External
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SOIC (0.295", 7.50mm Width)
Supplier Device Package	28-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16lc73a-04-so

PIC16C7X

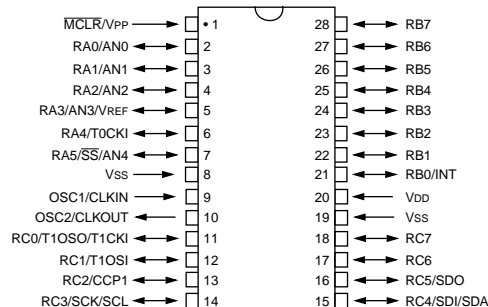
Pin Diagrams

SDIP, SOIC, Windowed Side Brazed Ceramic



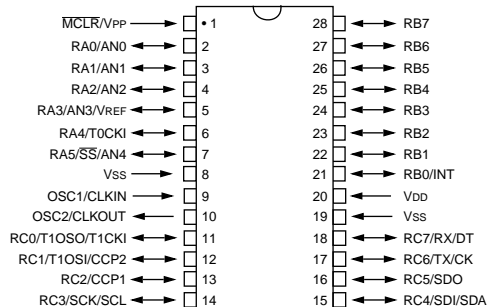
PIC16C72

SSOP



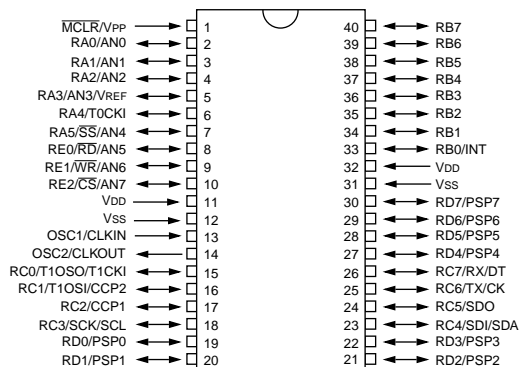
PIC16C72

SDIP, SOIC, Windowed Side Brazed Ceramic



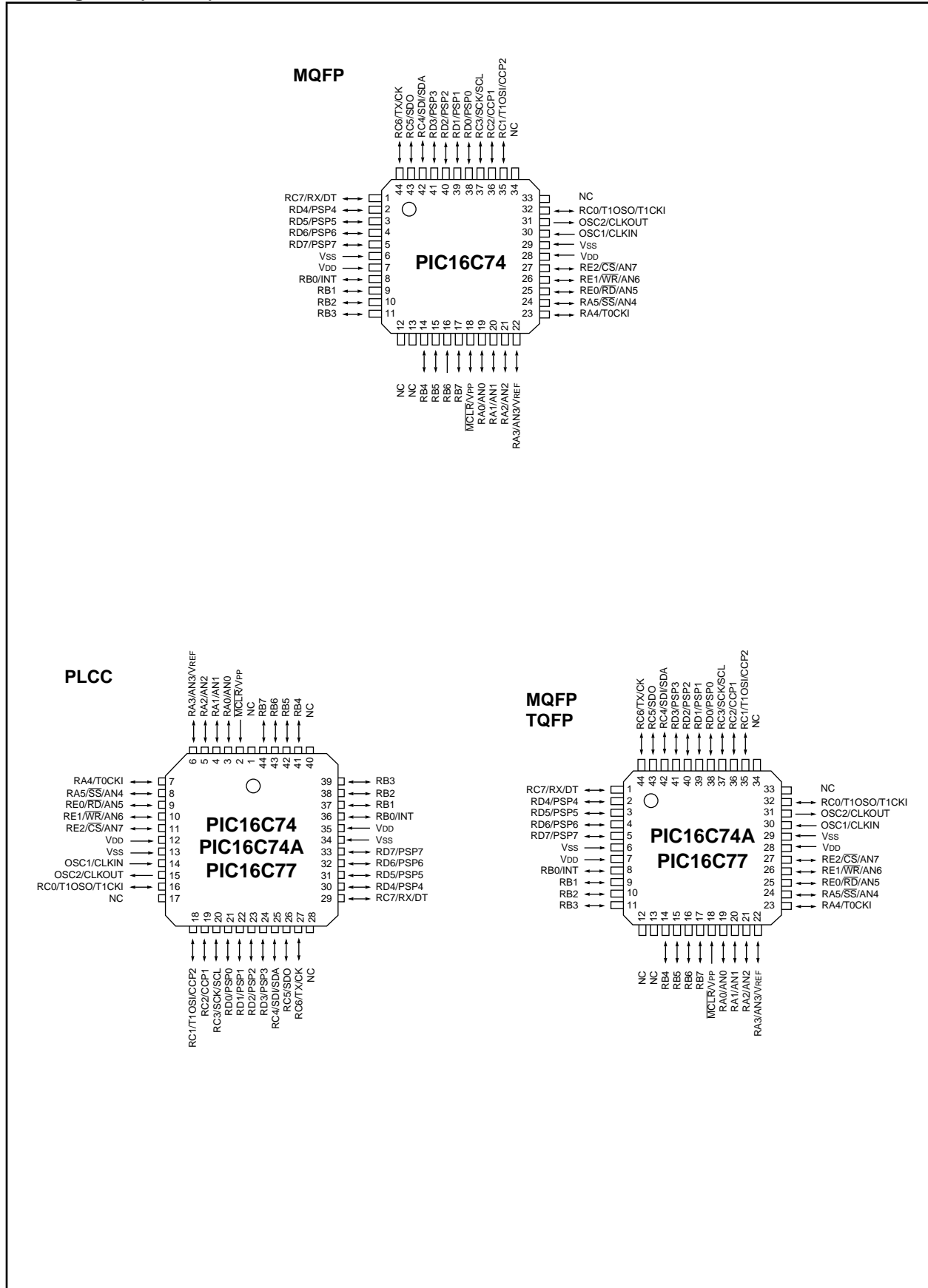
**PIC16C73
 PIC16C73A
 PIC16C76**

PDIP, Windowed Cerdip



**PIC16C74
 PIC16C74A
 PIC16C77**

Pin Diagrams (Cont.'d)



4.0 MEMORY ORGANIZATION

Applicable Devices							
72	73	73A	74	74A	76	77	

4.1 Program Memory Organization

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

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

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

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

FIGURE 4-1: PIC16C72 PROGRAM MEMORY MAP AND STACK

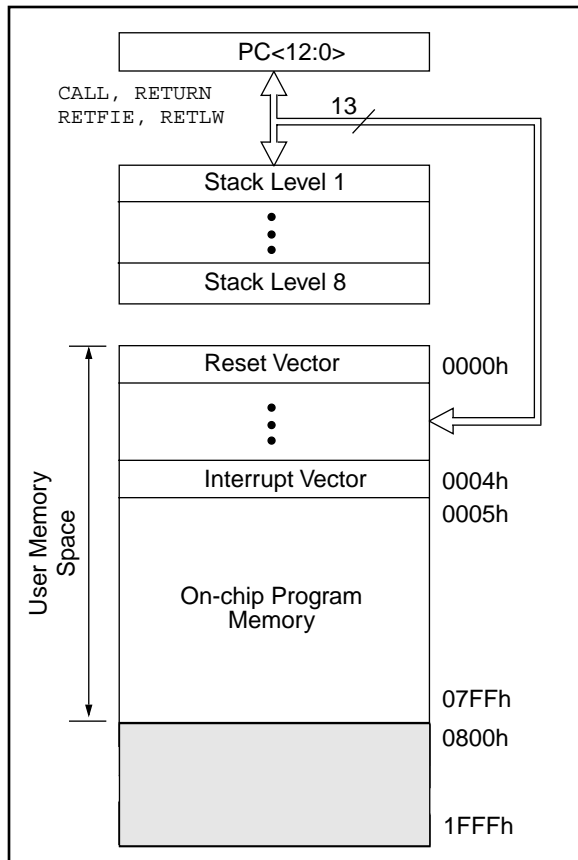


FIGURE 4-2: PIC16C73/73A/74/74A PROGRAM MEMORY MAP AND STACK

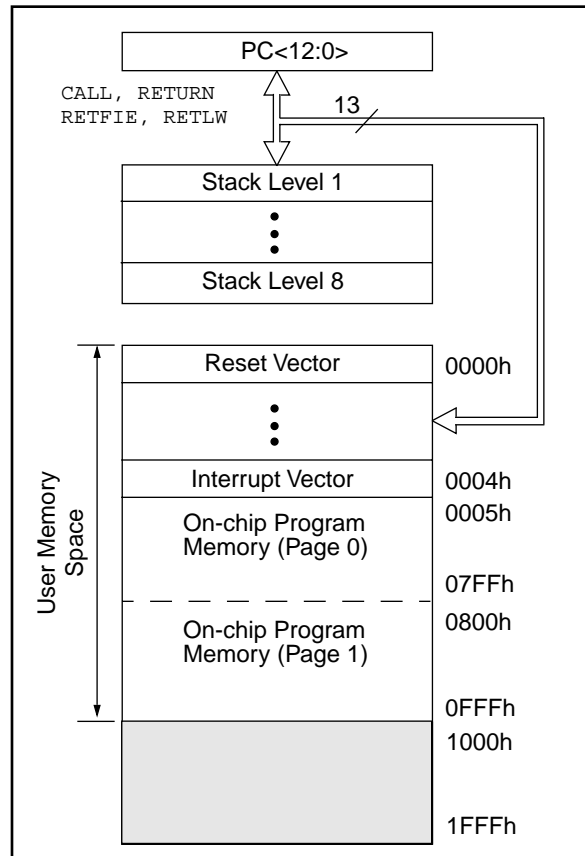


TABLE 4-3: PIC16C76/77 SPECIAL FUNCTION REGISTER SUMMARY

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other resets (2)
Bank 0											
00h ⁽⁴⁾	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	0000 0000
01h	TMR0	Timer0 module's register								xxxx xxxx	uuuu uuuu
02h ⁽⁴⁾	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	0000 0000
03h ⁽⁴⁾	STATUS	IRP	RP1	RP0	T ₀	P _D	Z	DC	C	0001 1xxx	000q quuu
04h ⁽⁴⁾	FSR	Indirect data memory address pointer								xxxx xxxx	uuuu uuuu
05h	PORTA	—	—	PORTA Data Latch when written: PORTA pins when read						--0x 0000	--0u 0000
06h	PORTB	PORTB Data Latch when written: PORTB pins when read								xxxx xxxx	uuuu uuuu
07h	PORTC	PORTC Data Latch when written: PORTC pins when read								xxxx xxxx	uuuu uuuu
08h ⁽⁵⁾	PORTD	PORTD Data Latch when written: PORTD pins when read								xxxx xxxx	uuuu uuuu
09h ⁽⁵⁾	PORTE	—	—	—	—	—	RE2	RE1	RE0	---- -xxx	---- -uuu
0Ah ^(1,4)	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	---0 0000
0Bh ⁽⁴⁾	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF ⁽³⁾	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
0Dh	PIR2	—	—	—	—	—	—	—	CCP2IF	---- --0	---- --0
0Eh	TMR1L	Holding register for the Least Significant Byte of the 16-bit TMR1 register								xxxx xxxx	uuuu uuuu
0Fh	TMR1H	Holding register for the Most Significant Byte of the 16-bit TMR1 register								xxxx xxxx	uuuu uuuu
10h	T1CON	—	—	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON	--00 0000	--uu uuuu
11h	TMR2	Timer2 module's register								0000 0000	0000 0000
12h	T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	-000 0000
13h	SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register								xxxx xxxx	uuuu uuuu
14h	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	0000 0000
15h	CCPR1L	Capture/Compare/PWM Register1 (LSB)								xxxx xxxx	uuuu uuuu
16h	CCPR1H	Capture/Compare/PWM Register1 (MSB)								xxxx xxxx	uuuu uuuu
17h	CCP1CON	—	—	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	--00 0000
18h	RCSTA	SPEN	RX9	SREN	CREN	—	FERR	OERR	RX9D	0000 -00x	0000 -00x
19h	TXREG	USART Transmit Data Register								0000 0000	0000 0000
1Ah	RCREG	USART Receive Data Register								0000 0000	0000 0000
1Bh	CCPR2L	Capture/Compare/PWM Register2 (LSB)								xxxx xxxx	uuuu uuuu
1Ch	CCPR2H	Capture/Compare/PWM Register2 (MSB)								xxxx xxxx	uuuu uuuu
1Dh	CCP2CON	—	—	CCP2X	CCP2Y	CCP2M3	CCP2M2	CCP2M1	CCP2M0	--00 0000	--00 0000
1Eh	ADRES	A/D Result Register								xxxx xxxx	uuuu uuuu
1Fh	ADCON0	ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	—	ADON	0000 00-0	0000 00-0

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

Shaded locations are unimplemented, read as '0'.

Note 1: The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8> whose contents are transferred to the upper byte of the program counter.

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

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

4: These registers can be addressed from any bank.

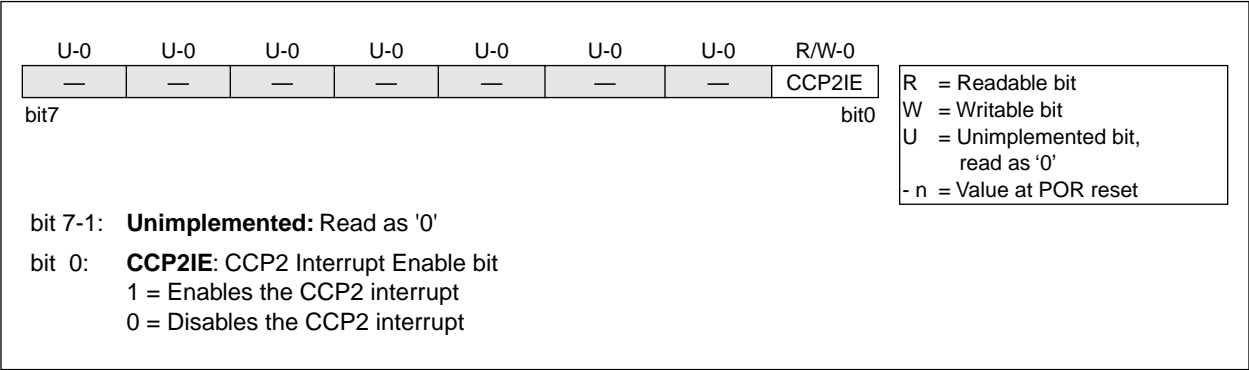
5: PORTD and PORTE are not physically implemented on the PIC16C76, read as '0'.

4.2.2.6 PIE2 REGISTER

Applicable Devices						
72	73	73A	74	74A	76	77

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

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



Example 4-1 shows the calling of a subroutine in page 1 of the program memory. This example assumes that PCLATH is saved and restored by the interrupt service routine (if interrupts are used).

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

```

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

4.5 Indirect Addressing, INDF and FSR Registers

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

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

Indirect addressing is possible by using the INDF register. Any instruction using the INDF register actually accesses the register pointed to by the File Select Register, FSR. Reading the INDF register itself indirectly (FSR = '0') will read 00h. Writing to the INDF register indirectly results in a no-operation (although status bits may be affected). An effective 9-bit address is obtained by concatenating the 8-bit FSR register and the IRP bit (STATUS<7>), as shown in Figure 4-18.

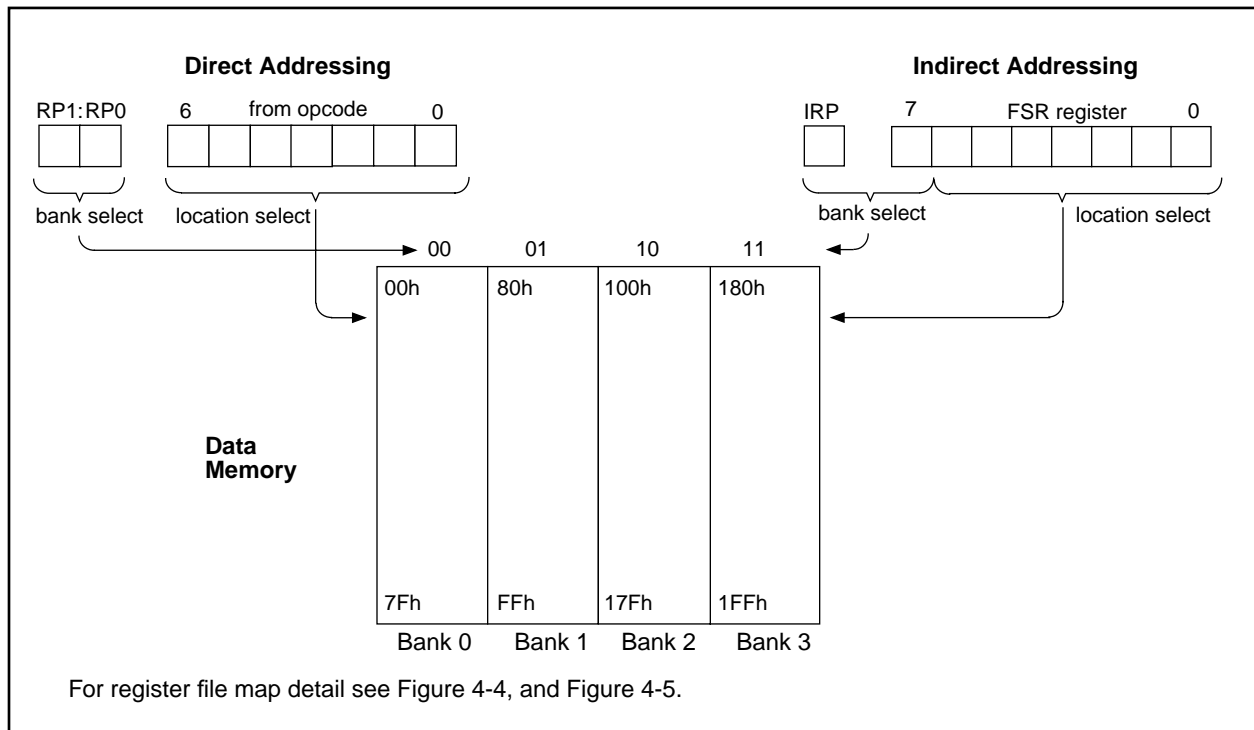
A simple program to clear RAM locations 20h-2Fh using indirect addressing is shown in Example 4-2.

EXAMPLE 4-2: INDIRECT ADDRESSING

```

movlw  0x20    ;initialize pointer
movwf  FSR     ;to RAM
NEXT   clrf    INDF ;clear INDF register
       incf    FSR,F ;inc pointer
       btfss   FSR,4 ;all done?
       goto    NEXT ;no clear next
CONTINUE
:         ;yes continue
    
```

FIGURE 4-18: DIRECT/INDIRECT ADDRESSING



PIC16C7X

5.7 Parallel Slave Port

Applicable Devices						
72	73	73A	74	74A	76	77

PORTD operates as an 8-bit wide Parallel Slave Port, or microprocessor port when control bit PSPMODE (TRISE<4>) is set. In slave mode it is asynchronously readable and writable by the external world through \overline{RD} control input pin RE0/ \overline{RD} /AN5 and \overline{WR} control input pin RE1/ \overline{WR} /AN6.

It can directly interface to an 8-bit microprocessor data bus. The external microprocessor can read or write the PORTD latch as an 8-bit latch. Setting bit PSPMODE enables port pin RE0/RD/AN5 to be the RD input, RE1/ $\overline{\text{WR}}$ /AN6 to be the $\overline{\text{WR}}$ input and RE2/ $\overline{\text{CS}}$ /AN7 to be the $\overline{\text{CS}}$ (chip select) input. For this functionality, the corresponding data direction bits of the TRISE register (TRISE<2:0>) must be configured as inputs (set) and the A/D port configuration bits PCFG2:PCFG0 (ADCON1<2:0>) must be set, which will configure pins RE2:RE0 as digital I/O.

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

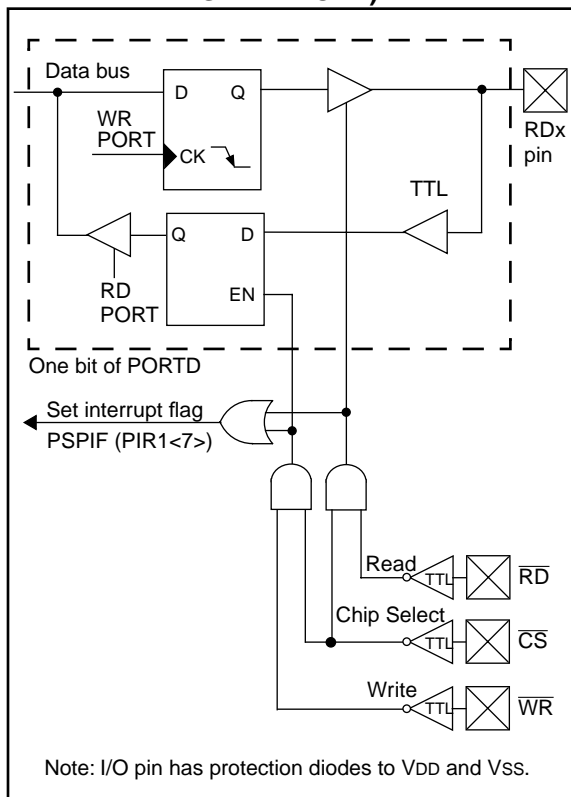
A write to the PSP occurs when both the $\overline{\text{CS}}$ and $\overline{\text{WR}}$ lines are first detected low. When either the $\overline{\text{CS}}$ or $\overline{\text{WR}}$ lines become high (level triggered), then the Input Buffer Full status flag bit IBF (TRISE<7>) is set on the Q4 clock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

A read from the PSP occurs when both the $\overline{\text{CS}}$ and $\overline{\text{RD}}$ lines are first detected low. The Output Buffer Full status flag bit OBF (TRISE<6>) is cleared immediately (Figure 5-13) indicating that the PORTD latch is waiting to be read by the external bus. When either the $\overline{\text{CS}}$ or $\overline{\text{RD}}$ pin becomes high (level triggered), the interrupt flag bit PSPIF is set on the Q4 clock cycle, following the next Q2 cycle, indicating that the read is complete. OBF remains low until data is written to PORTD by the user firmware.

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

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

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



Applicable Devices						
72	73	73A	74	74A	76	77

- Serial Data Out (SDO)
- Serial Data In (SDI)
- Serial Clock (SCK)

- Slave Select (\overline{SS})

- Master Mode (SCK is the clock output)
- Slave Mode (SCK is the clock input)
- Clock Polarity (Output/Input data on the Rising/Falling edge of SCK)
- Clock Rate (Master mode only)
- Slave Select Mode (Slave mode only)

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

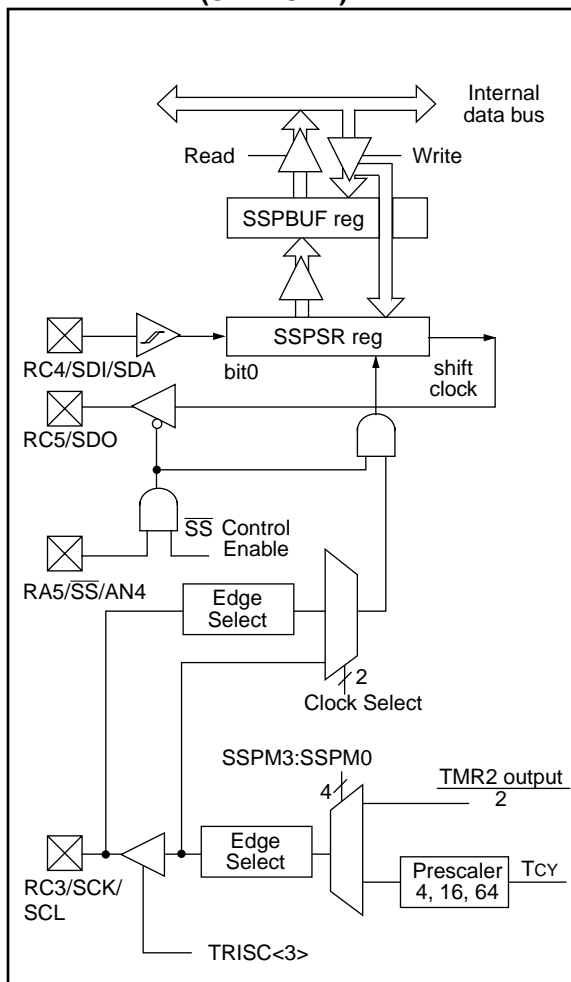
```

        BSF      STATUS, RP0      ;Specify Bank 1
LOOP    BTFSS   SSPSTAT, BF      ;Has data been
                                           ;received
                                           ;(transmit
                                           ;complete)?
        GOTO    LOOP             ;No
        BCF      STATUS, RP0      ;Specify Bank 0
        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 register is not directly readable or writable, and can only be accessed from addressing the SSPBUF register. Additionally, the SSP status register (SSPSTAT) indicates the various status conditions.

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



11.5.1 SLAVE MODE

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

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

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

- The buffer full bit BF (SSPSTAT<0>) was set before the transfer was received.
- The overflow bit SSPOV (SSPCON<6>) was set before the transfer was received.

In this case, the SSPSR register value is not loaded into the SSPBUF, but bit SSPIF (PIR1<3>) is set. Table 11-4 shows what happens when a data transfer byte is received, given the status of bits BF and SSPOV. The shaded cells show the condition where user software did not properly clear the overflow condition. Flag bit BF is cleared by reading the SSPBUF register while bit SSPOV is cleared through software.

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

11.5.1.1 ADDRESSING

Once the SSP module has been enabled, it waits for a START condition to occur. Following the START condition, the 8-bits are shifted into the SSPSR register. All incoming bits are sampled with the rising edge of the clock (SCL) line. The value of register SSPSR<7:1> is compared to the value of the SSPADD register. The

address is compared on the falling edge of the eighth clock (SCL) pulse. If the addresses match, and the BF and SSPOV bits are clear, the following events occur:

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

In 10-bit address mode, two address bytes need to be received by the slave (Figure 11-16). The five Most Significant bits (MSBs) of the first address byte specify if this is a 10-bit address. Bit R/W (SSPSTAT<2>) must specify a write so the slave device will receive the second address byte. For a 10-bit address the first byte would equal '1111 0 A9 A8 0', where A9 and A8 are the two MSBs of the address. The sequence of events for 10-bit address is as follows, with steps 7- 9 for slave-transmitter:

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

TABLE 11-4: DATA TRANSFER RECEIVED BYTE ACTIONS

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

12.3 USART Synchronous Master Mode

Applicable Devices							
72	73	73A	74	74A	76	77	

In Synchronous Master mode, the data is transmitted in a half-duplex manner i.e. transmission and reception do not occur at the same time. When transmitting data, the reception is inhibited and vice versa. Synchronous mode is entered by setting bit SYNC (TXSTA<4>). In addition enable bit SPEN (RCSTA<7>) is set in order to configure the RC6/TX/CK and RC7/RX/DT I/O pins to CK (clock) and DT (data) lines respectively. The Master mode indicates that the processor transmits the master clock on the CK line. The Master mode is entered by setting bit CSRC (TXSTA<7>).

12.3.1 USART SYNCHRONOUS MASTER TRANSMISSION

The USART transmitter block diagram is shown in Figure 12-7. The heart of the transmitter is the transmit (serial) shift register (TSR). The shift register obtains its data from the read/write transmit buffer register TXREG. The TXREG register is loaded with data in software. The TSR register is not loaded until the last bit has been transmitted from the previous load. As soon as the last bit is transmitted, the TSR is loaded with new data from the TXREG (if available). Once the TXREG register transfers the data to the TSR register (occurs in one Tcycle), the TXREG is empty and interrupt bit, TXIF (PIR1<4>) is set. The interrupt can be enabled/disabled by setting/clearing enable bit TXIE (PIE1<4>). Flag bit TXIF will be set regardless of the state of enable bit TXIE and cannot be cleared in software. It will reset only when new data is loaded into the TXREG register. While flag bit TXIF indicates the status of the TXREG register, another bit TRMT (TXSTA<1>) shows the status of the TSR register. TRMT is a read only bit which is set when the TSR is empty. No interrupt logic is tied to this bit, so the user has to poll this bit in order to determine if the TSR register is empty. The TSR is not mapped in data memory so it is not available to the user.

Transmission is enabled by setting enable bit TXEN (TXSTA<5>). The actual transmission will not occur until the TXREG register has been loaded with data. The first data bit will be shifted out on the next available rising edge of the clock on the CK line. Data out is stable around the falling edge of the synchronous clock (Figure 12-12). The transmission can also be started by first loading the TXREG register and then setting bit TXEN (Figure 12-13). This is advantageous when slow baud rates are selected, since the BRG is kept in reset when bits TXEN, CREN, and SREN are clear. Setting enable bit TXEN will start the BRG, creating a shift clock immediately. Normally when transmission is first started, the TSR register is empty, so a transfer to the TXREG register will result in an immediate transfer to TSR resulting in an empty TXREG. Back-to-back transfers are possible.

Clearing enable bit TXEN, during a transmission, will cause the transmission to be aborted and will reset the transmitter. The DT and CK pins will revert to hi-impedance. If either bit CREN or bit SREN is set, during a transmission, the transmission is aborted and the DT pin reverts to a hi-impedance state (for a reception). The CK pin will remain an output if bit CSRC is set (internal clock). The transmitter logic however is not reset although it is disconnected from the pins. In order to reset the transmitter, the user has to clear bit TXEN. If bit SREN is set (to interrupt an on-going transmission and receive a single word), then after the single word is received, bit SREN will be cleared and the serial port will revert back to transmitting since bit TXEN is still set. The DT line will immediately switch from hi-impedance receive mode to transmit and start driving. To avoid this, bit TXEN should be cleared.

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

Steps to follow when setting up a Synchronous Master Transmission:

1. Initialize the SPBRG register for the appropriate baud rate (Section 12.1).
2. Enable the synchronous master serial port by setting bits SYNC, SPEN, and CSRC.
3. If interrupts are desired, then set enable bit TXIE.
4. If 9-bit transmission is desired, then set bit TX9.
5. Enable the transmission by setting bit TXEN.
6. If 9-bit transmission is selected, the ninth bit should be loaded in bit TX9D.
7. Start transmission by loading data to the TXREG register.

12.3.2 USART SYNCHRONOUS MASTER RECEPTION

Once Synchronous mode is selected, reception is enabled by setting either enable bit SREN (RCSTA<5>) or enable bit CREN (RCSTA<4>). Data is sampled on the RC7/RX/DT pin on the falling edge of the clock. If enable bit SREN is set, then only a single word is received. If enable bit CREN is set, the reception is continuous until CREN is cleared. If both bits are set then CREN takes precedence. After clocking the last bit, the received data in the Receive Shift Register (RSR) is transferred to the RCREG register (if it is empty). When the transfer is complete, interrupt flag bit RCIF (PIR1<5>) is set. The actual interrupt can be enabled/disabled by setting/clearing enable bit RCIE (PIE1<5>). Flag bit RCIF is a read only bit which is reset by the hardware. In this case it is reset when the RCREG register has been read and is empty. The RCREG is a double buffered register, i.e. it is a two deep FIFO. It is possible for two bytes of data to be received and transferred to the RCREG FIFO and a third byte to begin shifting into the RSR register. On the clocking of the last bit of the third byte, if the RCREG register is still full then overrun error bit OERR (RCSTA<1>) is set. The word in the RSR will be lost. The RCREG register can be read twice to retrieve the two bytes in the FIFO. Bit OERR has to be cleared in software (by clearing bit CREN). If bit OERR is set, transfers from the RSR to the RCREG are inhibited, so it is essential to clear bit OERR if it is set. The 9th receive bit is buffered the same way as the receive data. Reading the RCREG register, will load bit RX9D with a new value, therefore it is essential for the user to read the RCSTA register before reading RCREG in order not to lose the old RX9D information.

Steps to follow when setting up a Synchronous Master Reception:

1. Initialize the SPBRG register for the appropriate baud rate. (Section 12.1)
2. Enable the synchronous master serial port by setting bits SYNC, SPEN, and CSRC.
3. Ensure bits CREN and SREN are clear.
4. If interrupts are desired, then set enable bit RCIE.
5. If 9-bit reception is desired, then set bit RX9.
6. If a single reception is required, set bit SREN. For continuous reception set bit CREN.
7. Interrupt flag bit RCIF will be set when reception is complete and an interrupt will be generated if enable bit RCIE was set.
8. Read the RCSTA register to get the ninth bit (if enabled) and determine if any error occurred during reception.
9. Read the 8-bit received data by reading the RCREG register.
10. If any error occurred, clear the error by clearing bit CREN.

TABLE 12-9: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER RECEPTION

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

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

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

13.4.1 FASTER CONVERSION - LOWER RESOLUTION TRADE-OFF

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

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

Where: N = number of bits of resolution required.

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

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

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

	Freq. (MHz) ⁽¹⁾	Resolution	
		4-bit	8-bit
TAD	20	1.6 μ s	1.6 μ s
	16	2.0 μ s	2.0 μ s
TOSC	20	50 ns	50 ns
	16	62.5 ns	62.5 ns
$2T_{AD} + N \cdot T_{AD} + (8 - N)(2T_{OSC})$	20	10 μ s	16 μ s
	16	12.5 μ s	20 μ s

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

INCFSZ Increment f, Skip if 0

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

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

Operation: $(f) + 1 \rightarrow (\text{destination})$,
 skip if result = 0

Status Affected: None

Encoding:

00	1111	dfff	ffff
----	------	------	------

Description: The contents of register 'f' are incremented. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is placed back in register 'f'. If the result is 1, the next instruction is executed. If the result is 0, a NOP is executed instead making it a 2Tcy instruction.

Words: 1

Cycles: 1(2)

Q Cycle Activity:

Q1	Q2	Q3	Q4
----	----	----	----

Decode	Read register 'f'	Process data	Write to destination
--------	-------------------	--------------	----------------------

If Skip: (2nd Cycle)

Q1	Q2	Q3	Q4
No-Operation	No-Operation	No-Operation	No-Operation

Example

```
HERE      INCFSZ    CNT, 1
          GOTO      LOOP
CONTINUE  •
          •
          •
```

Before Instruction

PC = address HERE

After Instruction

```
CNT = CNT + 1
if CNT= 0,
PC = address CONTINUE
if CNT≠ 0,
PC = address HERE +1
```

IORLW Inclusive OR Literal with W

Syntax: [*label*] IORLW *k*

Operands: $0 \leq k \leq 255$

Operation: $(W) .OR. k \rightarrow (W)$

Status Affected: Z

Encoding:

11	1000	kkkk	kkkk
----	------	------	------

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

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
----	----	----	----

Decode	Read literal 'k'	Process data	Write to W
--------	------------------	--------------	------------

Example

```
IORLW    0x35
```

Before Instruction

W = 0x9A

After Instruction

W = 0xBF
 Z = 1

SUBWF Subtract W from f

Syntax: `[label] SUBWF f,d`

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

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

Status Affected: C, DC, Z

Encoding:

00	0010	dfff	ffff
----	------	------	------

Description: Subtract (2's complement method) W register from register 'f'. If 'd' is 0 the result is stored in the W register. If 'd' is 1 the result is stored back in register 'f'.

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example 1: `SUBWF REG1, 1`

Before Instruction

REG1 = 3
W = 2
C = ?
Z = ?

After Instruction

REG1 = 1
W = 2
C = 1; result is positive
Z = 0

Example 2: Before Instruction

REG1 = 2
W = 2
C = ?
Z = ?

After Instruction

REG1 = 0
W = 2
C = 1; result is zero
Z = 1

Example 3: Before Instruction

REG1 = 1
W = 2
C = ?
Z = ?

After Instruction

REG1 = 0xFF
W = 2
C = 0; result is negative
Z = 0

SWAPF Swap Nibbles in f

Syntax: `[label] SWAPF f,d`

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

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

Status Affected: None

Encoding:

00	1110	dfff	ffff
----	------	------	------

Description: The upper and lower nibbles of register 'f' are exchanged. If 'd' is 0 the result is placed in W register. If 'd' is 1 the result is placed in register 'f'.

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example `SWAPF REG, 0`

Before Instruction

REG1 = 0xA5

After Instruction

REG1 = 0xA5
W = 0x5A

TRIS Load TRIS Register

Syntax: `[label] TRIS f`

Operands: $5 \leq f \leq 7$

Operation: $(W) \rightarrow \text{TRIS register } f$;

Status Affected: None

Encoding:

00	0000	0110	0fff
----	------	------	------

Description: The instruction is supported for code compatibility with the PIC16C5X products. Since TRIS registers are readable and writable, the user can directly address them.

Words: 1

Cycles: 1

Example

To maintain upward compatibility with future PIC16CXX products, do not use this instruction.

FIGURE 17-9: I²C BUS START/STOP BITS TIMING

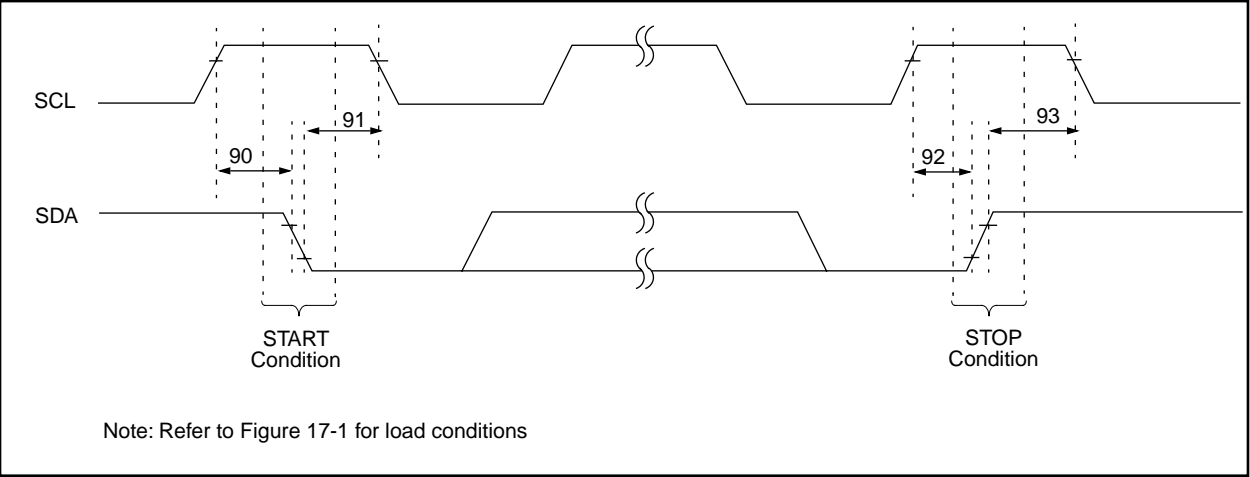


TABLE 17-8: I²C BUS START/STOP BITS REQUIREMENTS

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

FIGURE 18-6: CAPTURE/COMPARE/PWM TIMINGS (CCP1 AND CCP2)

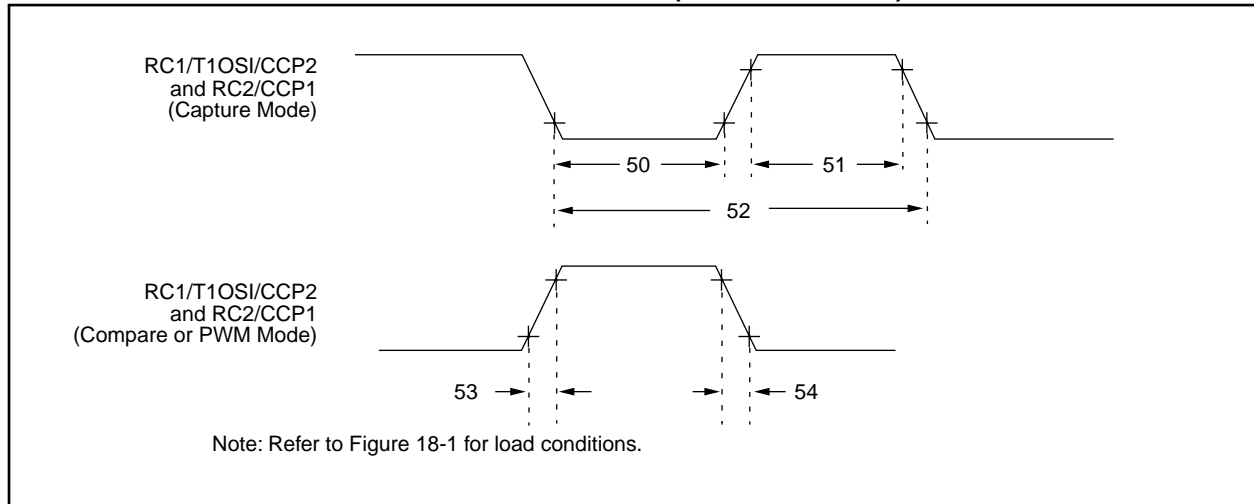


TABLE 18-6: CAPTURE/COMPARE/PWM REQUIREMENTS (CCP1 AND CCP2)

Parameter No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions	
50*	TccL	CCP1 and CCP2 input low time	No Prescaler		0.5Tcy + 20	—	—	ns	
			With Prescaler	PIC16C73/74	10	—	—	ns	
				PIC16LC73/74	20	—	—	ns	
51*	TccH	CCP1 and CCP2 input high time	No Prescaler		0.5Tcy + 20	—	—	ns	
			With Prescaler	PIC16C73/74	10	—	—	ns	
				PIC16LC73/74	20	—	—	ns	
52*	TccP	CCP1 and CCP2 input period			$\frac{3Tcy + 40}{N}$	—	—	ns	N = prescale value (1,4 or 16)
53*	TccR	CCP1 and CCP2 output fall time	PIC16C73/74		—	10	25	ns	
			PIC16LC73/74		—	25	45	ns	
54*	TccF	CCP1 and CCP2 output fall time	PIC16C73/74		—	10	25	ns	
			PIC16LC73/74		—	25	45	ns	

* These parameters are characterized but not tested.

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

FIGURE 19-7: CAPTURE/COMPARE/PWM TIMINGS (CCP1 AND CCP2)

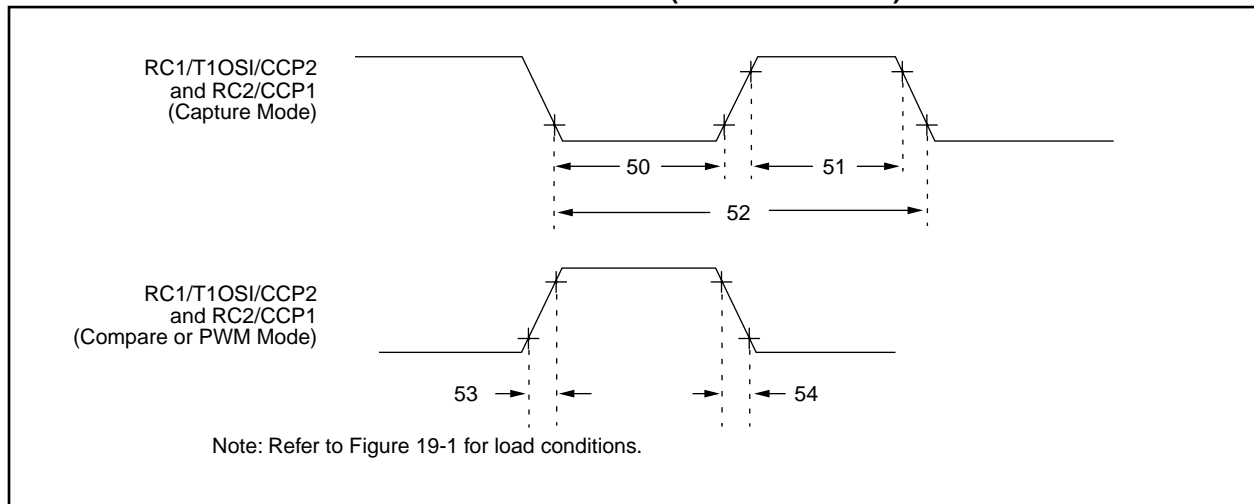


TABLE 19-6: CAPTURE/COMPARE/PWM REQUIREMENTS (CCP1 AND CCP2)

Param No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions	
50*	TccL	CCP1 and CCP2 input low time	No Prescaler		0.5TCY + 20	—	—	ns	
			With Prescaler	PIC16C73A/74A	10	—	—	ns	
				PIC16LC73A/74A	20	—	—	ns	
51*	TccH	CCP1 and CCP2 input high time	No Prescaler		0.5TCY + 20	—	—	ns	
			With Prescaler	PIC16C73A/74A	10	—	—	ns	
				PIC16LC73A/74A	20	—	—	ns	
52*	TccP	CCP1 and CCP2 input period			$\frac{3TCY + 40}{N}$	—	—	ns	N = prescale value (1,4 or 16)
53*	TccR	CCP1 and CCP2 output rise time	PIC16C73A/74A		—	10	25	ns	
			PIC16LC73A/74A		—	25	45	ns	
54*	TccF	CCP1 and CCP2 output fall time	PIC16C73A/74A		—	10	25	ns	
			PIC16LC73A/74A		—	25	45	ns	

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated.

These parameters are for design guidance only and are not tested.

PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

FIGURE 19-14: A/D CONVERSION TIMING

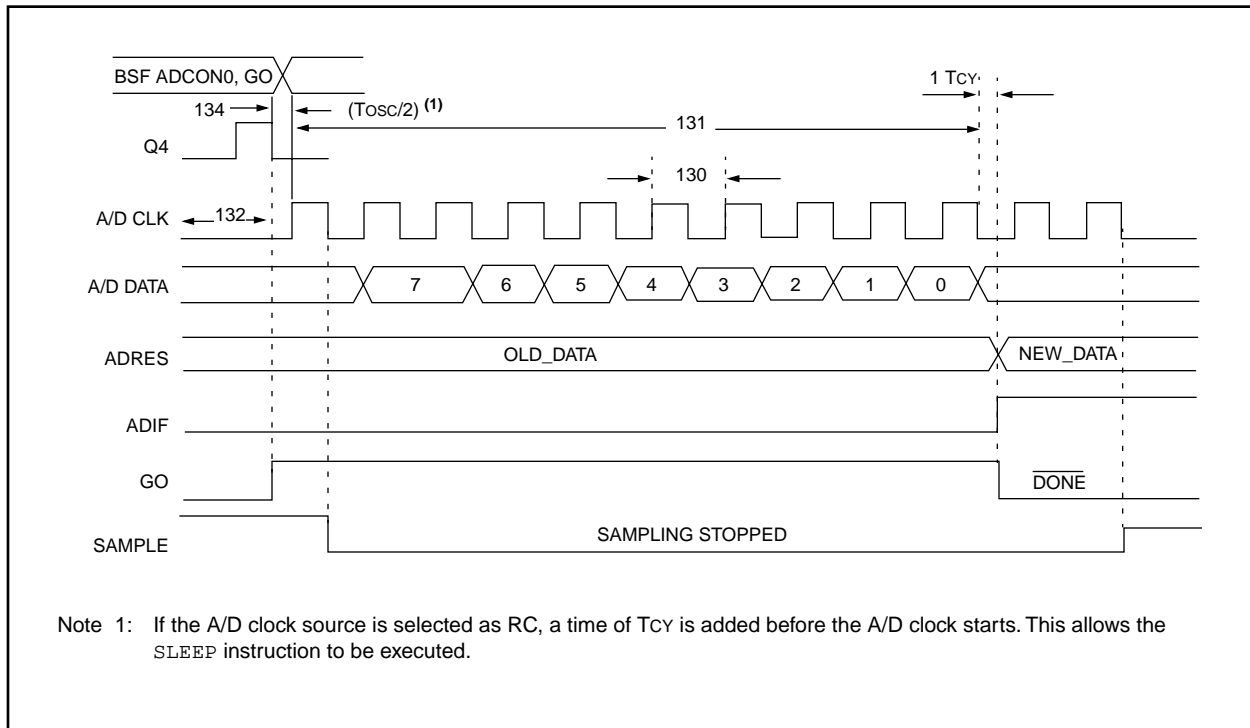


TABLE 19-14: A/D CONVERSION REQUIREMENTS

Param No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions
130	TAD	A/D clock period	PIC16C73A/74A	1.6	—	—	μs	TOSC based, VREF ≥ 3.0V
			PIC16LC73A/74A	2.0	—	—	μs	TOSC based, VREF full range
			PIC16C73A/74A	2.0	4.0	6.0	μs	A/D RC Mode
			PIC16LC73A/74A	3.0	6.0	9.0	μs	A/D RC Mode
131	TCNV	Conversion time (not including S/H time) (Note 1)		—	9.5	—	TAD	
132	TACQ	Acquisition time		Note 2	20	—	μs	The minimum time is the amplifier settling time. This may be used if the "new" input voltage has not changed by more than 1 LSb (i.e., 20.0 mV @ 5.12V) from the last sampled voltage (as stated on CHOLD).
				5*	—	—	μs	
134	TGO	Q4 to A/D clock start		—	TOSC/2 §	—	—	If the A/D clock source is selected as RC, a time of T _{CY} is added before the A/D clock starts. This allows the SLEEP instruction to be executed.
135	Tswc	Switching from convert → sample time		1.5 §	—	—	TAD	

* These parameters are characterized but not tested.

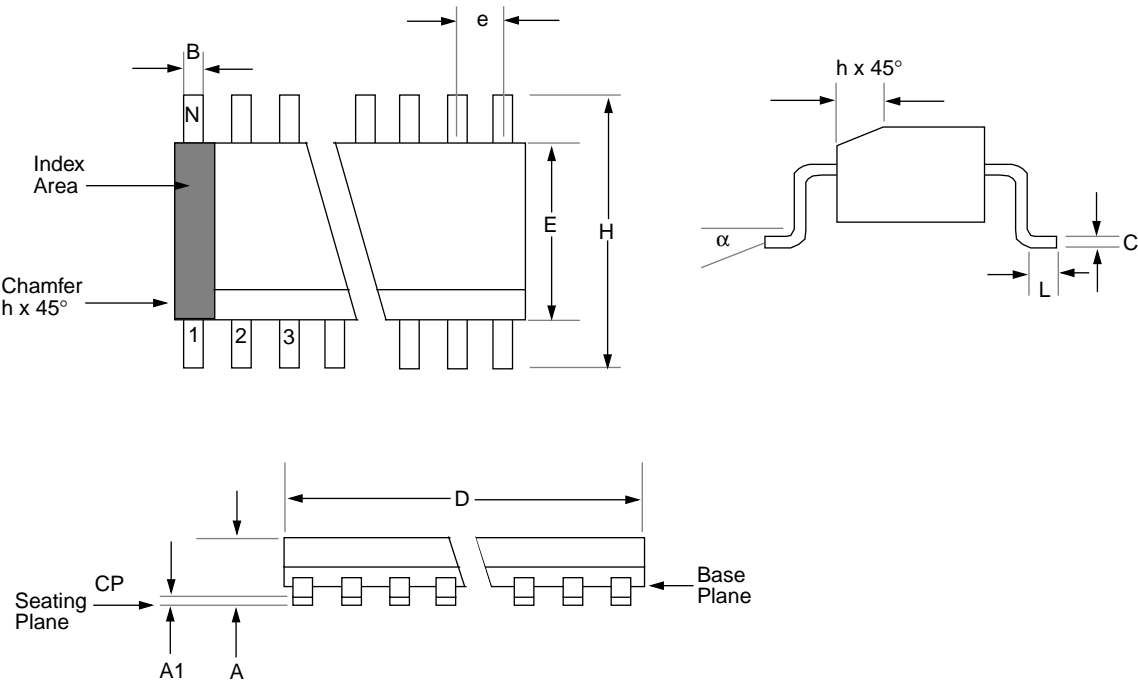
† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

§ This specification ensured by design.

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

2: See Section 13.1 for min conditions.

22.5 28-Lead Plastic Surface Mount (SOIC - Wide, 300 mil Body) (SO)



Package Group: Plastic SOIC (SO)						
Symbol	Millimeters			Inches		
	Min	Max	Notes	Min	Max	Notes
α	0°	8°		0°	8°	
A	2.362	2.642		0.093	0.104	
A1	0.101	0.300		0.004	0.012	
B	0.355	0.483		0.014	0.019	
C	0.241	0.318		0.009	0.013	
D	17.703	18.085		0.697	0.712	
E	7.416	7.595		0.292	0.299	
e	1.270	1.270	Typical	0.050	0.050	Typical
H	10.007	10.643		0.394	0.419	
h	0.381	0.762		0.015	0.030	
L	0.406	1.143		0.016	0.045	
N	28	28		28	28	
CP	—	0.102		—	0.004	

INDEX

A

A/D

Accuracy/Error	124
ADCON0 Register	117
ADCON1 Register	118
ADIF bit	119
Analog Input Model Block Diagram	120
Analog-to-Digital Converter	117
Block Diagram	119
Configuring Analog Port Pins	121
Configuring the Interrupt	119
Configuring the Module	119
Connection Considerations	125
Conversion Clock	121
Conversion Time	123
Conversions	122
Converter Characteristics	181, 199, 217, 238
Delays	120
Effects of a Reset	124
Equations	120
Faster Conversion - Lower Resolution Tradeoff	123
Flowchart of A/D Operation	126
GO/DONE bit	119
Internal Sampling Switch (Rss) Impedance	120
Operation During Sleep	124
Sampling Requirements	120
Sampling Time	120
Source Impedance	120
Time Delays	120
Transfer Function	125
Using the CCP Trigger	125
Absolute Maximum Ratings	167, 183, 201, 219
ACK	90, 94, 95
ADIE bit	33
ADIF bit	35
ADRES Register	23, 25, 27, 117, 119
ALU	9
Application Notes	
AN546 (Using the Analog-to-Digital Converter)	117
AN552 (Implementing Wake-up on Key Strokes Using PIC16CXXX)	45
AN556 (Table Reading Using PIC16CXX)	40
AN578 (Use of the SSP Module in the I ² C Multi-Master Environment)	77
AN594 (Using the CCP Modules)	71
AN607, Power-up Trouble Shooting	134
Architecture	
Harvard	9
Overview	9
von Neumann	9
Assembler	
MPASM Assembler	164

B

Baud Rate Error	101
Baud Rate Formula	101
Baud Rates	
Asynchronous Mode	102
Synchronous Mode	102
BF	78, 83, 94
Block Diagrams	
A/D	119
Analog Input Model	120
Capture	72

Compare	73
I ² C Mode	93
On-Chip Reset Circuit	133
PIC16C72	10
PIC16C73	11
PIC16C73A	11
PIC16C74	12
PIC16C74A	12
PIC16C76	11
PIC16C77	12
PORTC	48
PORTD (In I/O Port Mode)	50
PORTD and PORTE as a Parallel Slave Port	54
PORTE (In I/O Port Mode)	51
PWM	74
RA3:RA0 and RA5 Port Pins	43
RA4/T0CKI Pin	43
RB3:RB0 Port Pins	45
RB7:RB4 Port Pins	46
SPI Master/Slave Connection	81
SSP in I ² C Mode	93
SSP in SPI Mode	80, 85
Timer0	59
Timer0/WDT Prescaler	62
Timer1	66
Timer2	69
USART Receive	108
USART Transmit	106
Watchdog Timer	144
BOR bit	39, 135
BRGH bit	101
Buffer Full Status bit, BF	78, 83

C

C bit	30
C Compiler	165
Capture/Compare/PWM	
Capture	
Block Diagram	72
CCP1CON Register	72
CCP1IF	72
CCPR1	72
CCPR1H:CCPR1L	72
Mode	72
Prescaler	73
CCP Timer Resources	71
Compare	
Block Diagram	73
Mode	73
Software Interrupt Mode	73
Special Event Trigger	73
Special Trigger Output of CCP1	73
Special Trigger Output of CCP2	73
Interaction of Two CCP Modules	71
Section	71
Special Event Trigger and A/D Conversions	73
Capture/Compare/PWM (CCP)	
PWM Block Diagram	74
PWM Mode	74
PWM, Example Frequencies/Resolutions	75
Carry bit	9
CCP1CON	29
CCP1IE bit	33
CCP1IF bit	35, 36
CCP2CON	29
CCP2IE bit	37