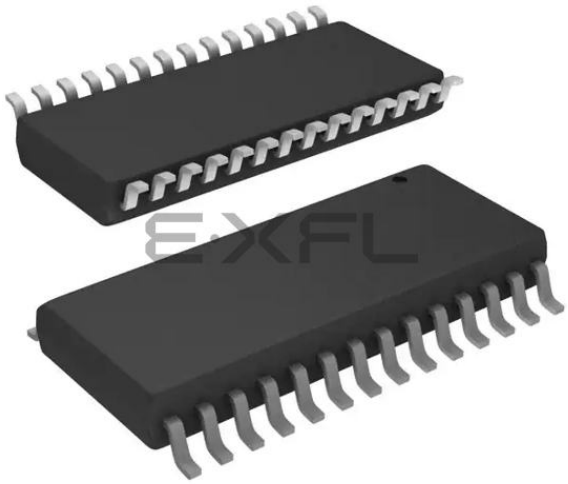


Welcome to E-XFL.COM

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, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	25
Program Memory Size	14KB (8K x 14)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	368 x 8
Voltage - Supply (Vcc/Vdd)	4V ~ 5.5V
Data Converters	A/D 11x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°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/pic16f767-i-so

Table of Contents

TO OUR VALUED CUSTOMERS

It is our intention to provide our valued customers with the best documentation possible to ensure successful use of your Microchip products. To this end, we will continue to improve our publications to better suit your needs. Our publications will be refined and enhanced as new volumes and updates are introduced.

If you have any questions or comments regarding this publication, please contact the Marketing Communications Department via E-mail at **docerrors@microchip.com** or fax the **Reader Response Form** in the back of this data sheet to (480) 792-4150. We welcome your feedback.

Most Current Data Sheet

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

http://www.microchip.com

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

Errata

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

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

- Microchip's Worldwide Web site; http://www.microchip.com
- Your local Microchip sales office (see last page)

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

Customer Notification System

Register on our web site at **www.microchip.com** to receive the most current information on all of our products.

3.0 READING PROGRAM MEMORY

The Flash program memory is readable during normal operation over the entire VDD range. It is indirectly addressed through Special Function Registers (SFR). Up to 14-bit numbers can be stored in memory for use as calibration parameters, serial numbers, packed 7-bit ASCII, etc. Executing a program memory location containing data that forms an invalid instruction results in a NOP.

There are five SFRs used to read the program and memory. These registers are:

- PMCON1
- PMDATA
- PMDATH
- PMADR
- PMADRH

The program memory allows word reads. Program memory access allows for checksum calculation and reading calibration tables.

When interfacing to the program memory block, the PMDATH:PMDATA registers form a two-byte word which holds the 14-bit data for reads. The PMADRH:PMADR registers form a two-byte word which holds the 13-bit address of the Flash location being accessed. These devices can have up to 8K words of program Flash, with an address range from 0h to 3FFFh. The unused upper bits in both the PMDATH and PMADRH registers are not implemented and read as '0's.

3.1 PMADR

The address registers can address up to a maximum of 8K words of program Flash.

When selecting a program address value, the MSB of the address is written to the PMADRH register and the LSB is written to the PMADR register. The upper Most Significant bits of PMADRH must always be clear.

3.2 PMCON1 Register

PMCON1 is the control register for memory accesses.

The control bit, RD, initiates read operations. This bit cannot be cleared, only set, in software. It is cleared in hardware at the completion of the read operation.

REGISTER 3-1: PMCON1: PROGRAM MEMORY CONTROL REGISTER 1 (ADDRESS 18Ch)

R-1	U-0	U-0	U-0	U-x	U-0	U-0	R/S-0
reserved	—	—	—	—	—	—	RD
bit 7							bit 0

bit 7 **Reserved:** Read as '1'

bit 6-1 **Unimplemented**: Read as '0'

bit 0 **RD**: Read Control bit

1 = Initiates a Flash read, RD is cleared in hardware. The RD bit can only be set (not cleared) in software.

0 = Flash read completed

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

3.3 Reading the Flash Program Memory

A program memory location may be read by writing two bytes of the address to the PMADR and PMADRH registers and then setting control bit, RD (PMCON1<0>). Once the read control bit is set, the microcontroller will use the next two instruction cycles to read the data. The data is available in the PMDATA and PMDATH registers after the second NOP instruction; therefore, it can be read as two bytes in the following instructions. The PMDATA and PMDATH registers will hold this value until the next read operation.

3.4 Operation During Code-Protect

Flash program memory has its own code-protect mechanism. External read and write operations by programmers are disabled if this mechanism is enabled.

The microcontroller can read and execute instructions out of the internal Flash program memory, regardless of the state of the code-protect configuration bits.

EXAMPLE 3-1: FLASH PROGRAM READ

```
            BSF     STATUS, RP1     ;
            BCF     STATUS, RP0     ; Bank 2
            MOVF    ADDRH, W        ;
            MOVWF   PMADRH          ; MSByte of Program Address to read
            MOVF    ADDRL, W        ;
            MOVWF   PMADR           ; LSByte of Program Address to read
            BSF     STATUS, RP0     ; Bank 3 Required

Required    BSF     PMCON1, RD      ; EEPROM Read Sequence
Sequence    NOP                     ; memory is read in the next two cycles after BSF PMCON1,RD
            NOP                     ;

            BCF     STATUS, RP0     ; Bank 2
            MOVF    PMDATA, W       ; W = LSByte of Program PMDATA
            MOVF    PMDATH, W       ; W = MSByte of Program PMDATH
```

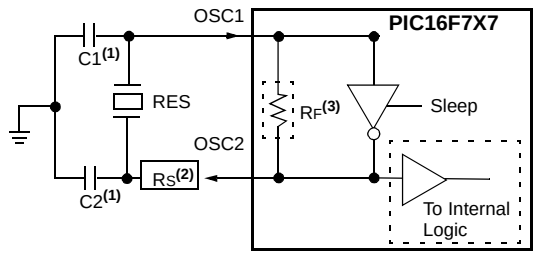
TABLE 3-1: REGISTERS ASSOCIATED WITH PROGRAM FLASH

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
10Dh	PMADR	EEPROM Address Register Low Byte								xxxx xxxx	uuuu uuuu
10Fh	PMADRH	—	—	—	—	EEPROM Address Register High Byte				---- xxxx	---u uuuu
10Ch	PMDATA	EEPROM Data Register Low Byte								xxxx xxxx	uuuu uuuu
10Eh	PMDATH	—	—	EEPROM Data Register High Byte						--xx xxxx	--uu uuuu
18Ch	PMCON1	reserved(1)	—	—	—	—	—	—	RD	1--- ---0	1--- ---0

Legend: x = unknown, u = unchanged, — = unimplemented, read as '0'. Shaded cells are not used during Flash access.

Note 1: This bit always reads as a '1'.

FIGURE 4-2: CERAMIC RESONATOR OPERATION (HS OR XT OSC CONFIGURATION)



Note 1: See Table 4-2 for typical values of C1 and C2.

2: A series resistor (RS) may be required.

3: RF varies with the resonator chosen (typically between 2 MΩ to 10 MΩ).

TABLE 4-2: CERAMIC RESONATORS (FOR DESIGN GUIDANCE ONLY)

Typical Capacitor Values Used:			
Mode	**Freq**	**OSC1**	**OSC2**
XT	455 kHz 2.0 MHz 4.0 MHz	56 pF 47 pF 33 pF	56 pF 47 pF 33 pF
HS	8.0 MHz 16.0 MHz	27 pF 22 pF	27 pF 22 pF

Capacitor values are for design guidance only.

These capacitors were tested with the resonators listed below for basic start-up and operation. These values were not optimized.

Different capacitor values may be required to produce acceptable oscillator operation. The user should test the performance of the oscillator over the expected VDD and temperature range for the application.

See the notes following this table for additional information.

Note: When using resonators with frequencies above 3.5 MHz, the use of HS mode rather than XT mode is recommended. HS mode may be used at any VDD for which the controller is rated. If HS is selected, it is possible that the gain of the oscillator will overdrive the resonator. Therefore, a series resistor should be placed between the OSC2 pin and the resonator. As a good starting point, the recommended value of RS is 330Ω.

4.3 External Clock Input

The ECIO Oscillator mode requires an external clock source to be connected to the OSC1 pin. There is no oscillator start-up time required after a Power-on Reset or after an exit from Sleep mode.

In the ECIO Oscillator mode, the OSC2 pin becomes an additional general purpose I/O pin. The I/O pin becomes bit 6 of PORTA (RA6). Figure 4-3 shows the pin connections for the ECIO Oscillator mode.

FIGURE 4-3: EXTERNAL CLOCK INPUT OPERATION (ECIO CONFIGURATION)

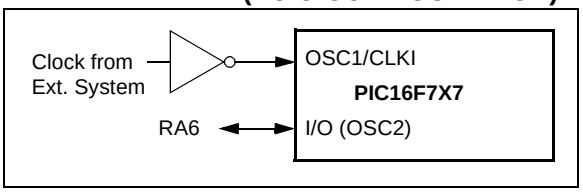
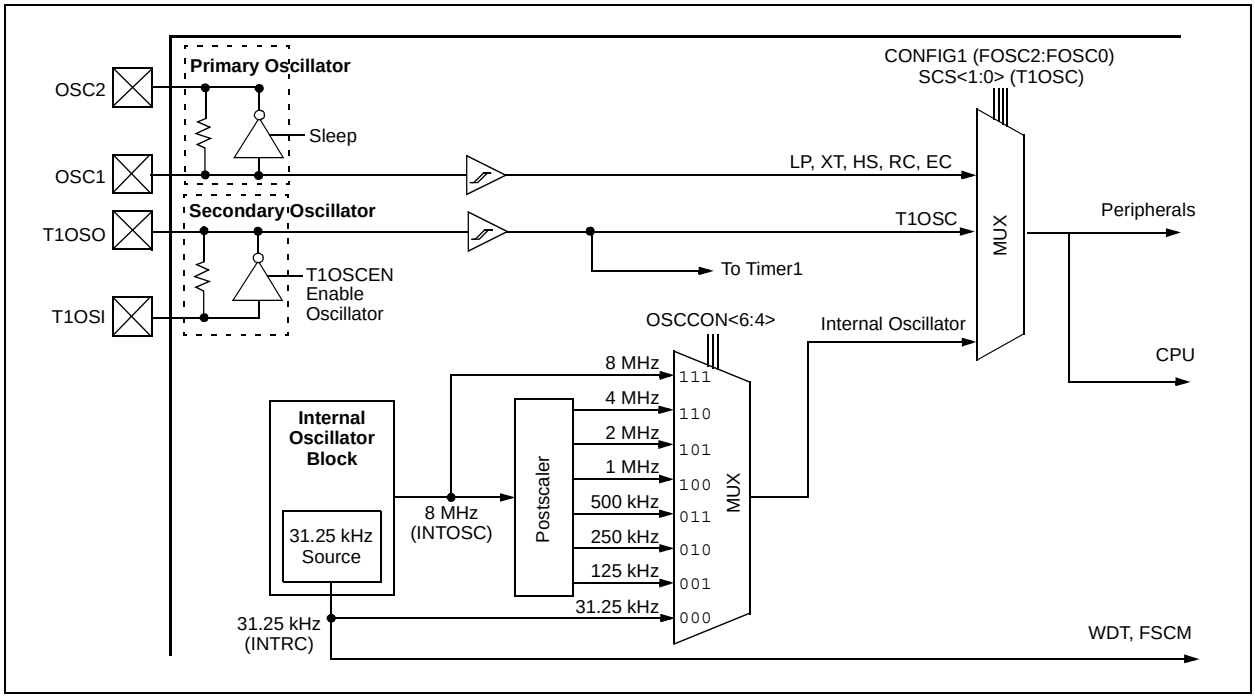


FIGURE 4-6: PIC16F7X7 CLOCK DIAGRAM



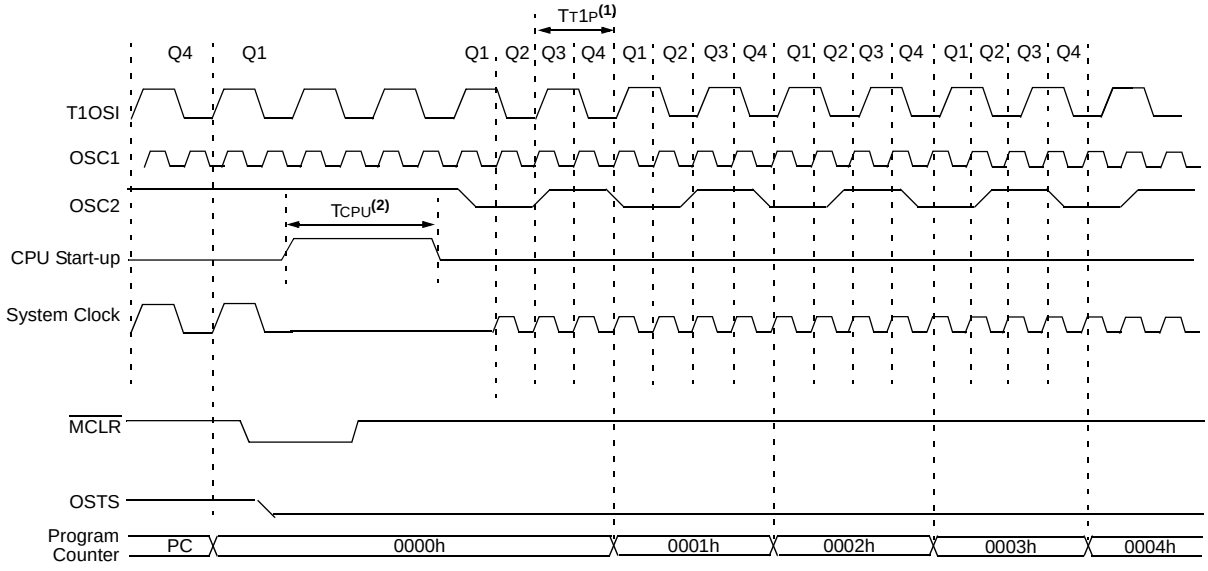
4.6.4 MODIFYING THE IRCF BITS

The IRCF bits can be modified at any time regardless of which clock source is currently being used as the system clock. The internal oscillator allows users to change the frequency during run time. This is achieved by modifying the IRCF bits in the OSCCON register. The sequence of events that occur after the IRCF bits are modified is dependent upon the initial value of the IRCF bits before they are modified. If the INTRC (31.25 kHz, IRCF<2:0> = 000) is running and the IRCF bits are modified to any other value than '000', a 4 ms (approx.) clock switch delay is turned on. Code execution continues at a higher than expected frequency while the new frequency stabilizes. Time sensitive code should wait for the IOFS bit in the OSCCON register to become set before continuing. This bit can be monitored to ensure that the frequency is stable before using the system clock in time critical applications.

If the IRCF bits are modified while the internal oscillator is running at any other frequency than INTRC (31.25 kHz, IRCF<2:0> ≠ 000), there is no need for a 4 ms (approx.) clock switch delay. The new INTOSC frequency will be stable immediately after the **eight** falling edges. The IOFS bit will remain set after clock switching occurs.

Note:	Caution must be taken when modifying the IRCF bits using BCF or BSF instructions. It is possible to modify the IRCF bits to a frequency that may be out of the V_{DD} specification range; for example: V_{DD} = 2.0V and IRCF = 111 (8 MHz).

FIGURE 4-11: TIMING LP CLOCK TO PRIMARY SYSTEM CLOCK AFTER RESET (EC, RC, INTRC)



Note 1: T_{T1P} = 30.52 μs.
2: T_{CPU} = 5-10 μs.

7.5 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 that will wake-up the processor. However, special precautions in software are needed to read/write the timer (**Section 7.5.1 "Reading and Writing Timer1 in Asynchronous Counter Mode"**).

In Asynchronous Counter mode, Timer1 cannot be used as a time base for capture or compare operations.

7.5.1 READING AND WRITING TIMER1 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. The example codes provided in Example 7-1 and Example 7-2 demonstrate how to write to and read Timer1 while it is running in Asynchronous mode.

EXAMPLE 7-1: WRITING A 16-BIT FREE RUNNING TIMER

```
; All interrupts are disabled
CLRF      TMR1L           ; Clear Low byte, Ensures no rollover into TMR1H
MOVLW     HI_BYTE         ; Value to load into TMR1H
MOVWF     TMR1H, F        ; Write High byte
MOVLW     LO_BYTE         ; Value to load into TMR1L
MOVWF     TMR1H, F        ; Write Low byte
; Re-enable the Interrupt (if required)
CONTINUE                  ; Continue with your code
```

EXAMPLE 7-2: 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
```

REGISTER 10-2: SSPCON: MSSP CONTROL (SPI MODE) REGISTER 1 (ADDRESS 14h)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0
bit 7							bit 0

bit 7 **WCOL:** Write Collision Detect bit (Transmit mode only)

1 = The SSPBUF register is written while it is still transmitting the previous word. (Must be cleared in software.)

0 = No collision

bit 6 **SSPOV:** Receive Overflow Indicator bit

SPI Slave mode:

1 = A new byte is received while the SSPBUF register is still holding the previous data. In case of overflow, the data in SSPSR is lost. Overflow can only occur in Slave mode. The user must read the SSPBUF, even if only transmitting data, to avoid setting overflow. (Must be cleared in software.)

0 = No overflow

Note: In Master mode, the overflow bit is not set since each new reception (and transmission) is initiated by writing to the SSPBUF register.

bit 5 **SSPEN:** Synchronous Serial Port Enable bit

1 = Enables serial port and configures SCK, SDO, SDI and $\overline{SS}$ as serial port pins

0 = Disables serial port and configures these pins as I/O port pins

Note: When enabled, these pins must be properly configured as input or output.

bit 4 **CKP:** Clock Polarity Select bit

1 = Idle state for clock is a high level

0 = Idle state for clock is a low level

bit 3-0 **SSPM3:SSPM0:** Synchronous Serial Port Mode Select bits

0101 = SPI Slave mode, clock = SCK pin. $\overline{SS}$ pin control disabled. $\overline{SS}$ can be used as I/O pin.

0100 = SPI Slave mode, clock = SCK pin. $\overline{SS}$ pin control enabled.

0011 = SPI Master mode, clock = TMR2 output/2

0010 = SPI Master mode, clock = FOSC/64

0001 = SPI Master mode, clock = FOSC/16

0000 = SPI Master mode, clock = FOSC/4

Note: Bit combinations not specifically listed here are either reserved or implemented in I^2C mode only.

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

FIGURE 10-5: SPI MODE WAVEFORM (SLAVE MODE WITH CKE = 0)

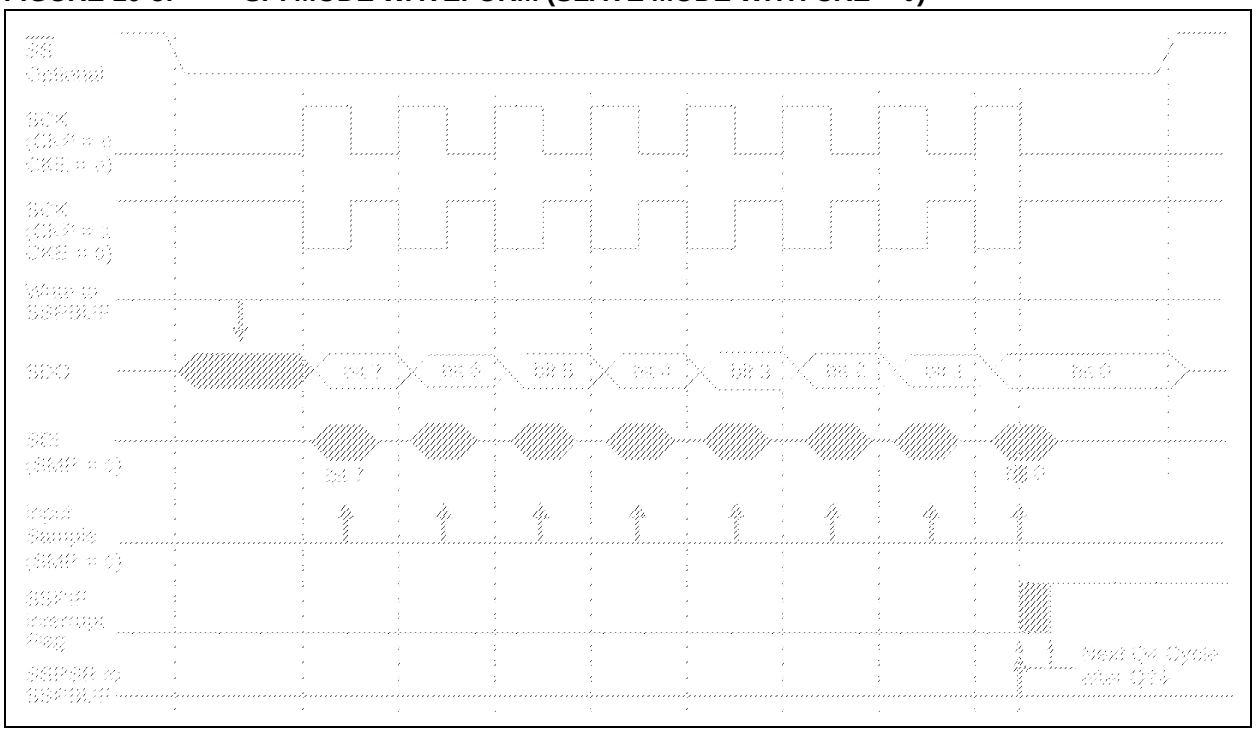
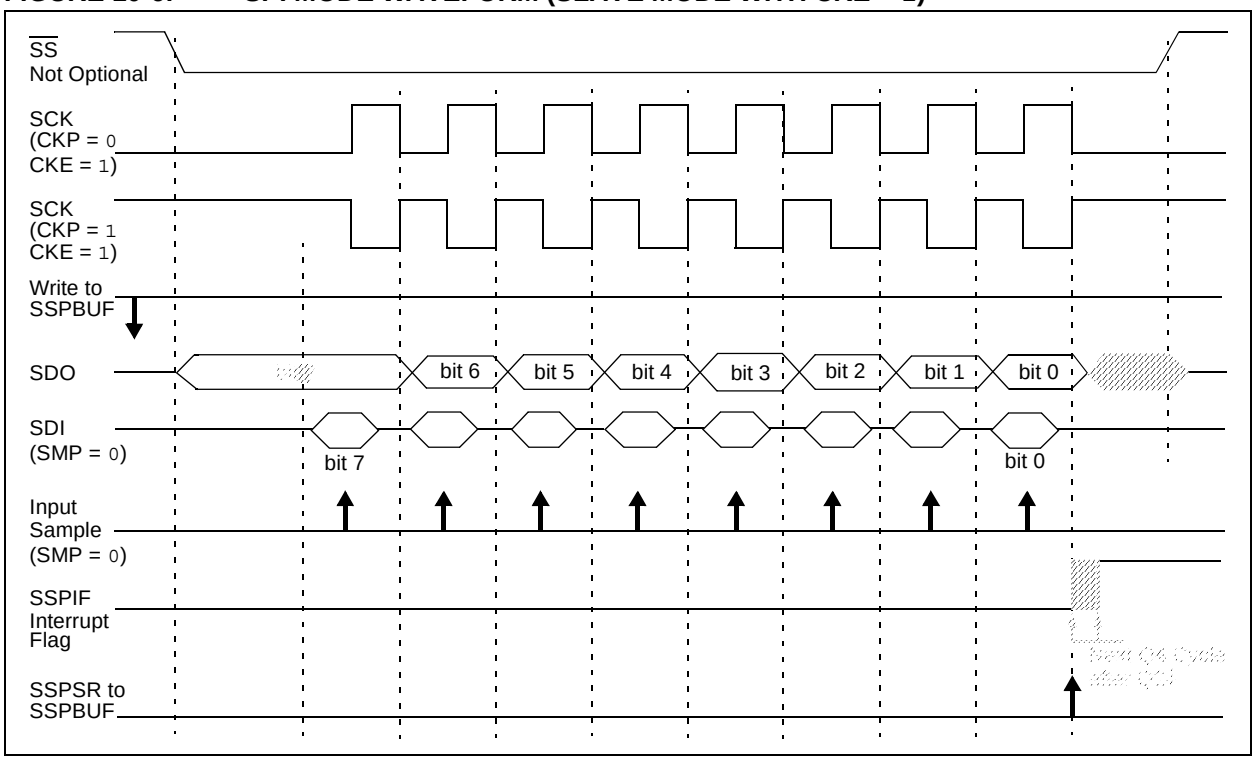


FIGURE 10-6: SPI MODE WAVEFORM (SLAVE MODE WITH CKE = 1)



10.3.8 SLEEP OPERATION

In Master mode, all module clocks are halted and the transmission/reception will remain in that state until the device wakes from Sleep. After the device returns to normal mode, the module will continue to transmit/ receive data.

In Slave mode, the SPI Transmit/Receive Shift register operates asynchronously to the device. This allows the device to be placed in Sleep mode and data to be shifted into the SPI Transmit/Receive Shift register. When all 8 bits have been received, the MSSP interrupt flag bit will be set and if enabled, will wake the device from Sleep.

10.3.9 EFFECTS OF A RESET

A Reset disables the MSSP module and terminates the current transfer.

10.3.10 BUS MODE COMPATIBILITY

Table 10-1 shows the compatibility between the standard SPI modes and the states of the CKP and CKE control bits.

TABLE 10-1: SPI BUS MODES

Standard SPI Mode Terminology	Control Bits State	
	CKP	CKE
0, 0	0	1
0, 1	0	0
1, 0	1	1
1, 1	1	0

There is also an SMP bit which controls when the data is sampled.

TABLE 10-2: REGISTERS ASSOCIATED WITH SPI OPERATION

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
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u
PIR1	PSPIF(1)	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
PIE1	PSPIE(1)	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
TRISC	PORTC Data Direction Register								1111 1111	1111 1111
SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register								xxxx xxxx	uuuu uuuu
SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	0000 0000
TRISA	PORTA Data Direction Register								1111 1111	1111 1111
SSPSTAT	SMP	CKE	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF	0000 0000	0000 0000

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

Note 1: The PSPIF and PSPIE bits are reserved on 28-pin devices; always maintain these bits clear.

When setting up an Asynchronous Transmission, follow these steps:

1. Initialize the SPBRG register for the appropriate baud rate. If a high-speed baud rate is desired, set bit BRGH (see **Section 11.1 "AUSART Baud Rate Generator (BRG)"**).
2. Enable the asynchronous serial port by clearing bit SYNC and setting bit SPEN.
3. If interrupts are desired, then set enable bit TXIE.
4. If 9-bit transmission is desired, then set transmit bit TX9.
5. Enable the transmission by setting bit TXEN which will also set bit TXIF.
6. If 9-bit transmission is selected, the ninth bit should be loaded in bit TX9D.
7. Load data to the TXREG register (starts transmission).
8. If using interrupts, ensure that GIE and PEIE (bits 7 and 6) of the INTCON register are set.

FIGURE 11-2: ASYNCHRONOUS MASTER TRANSMISSION

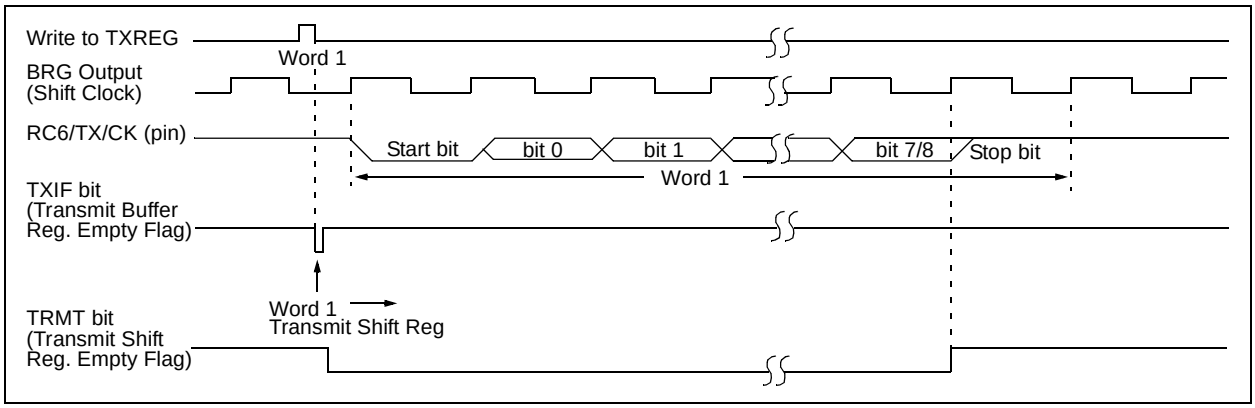
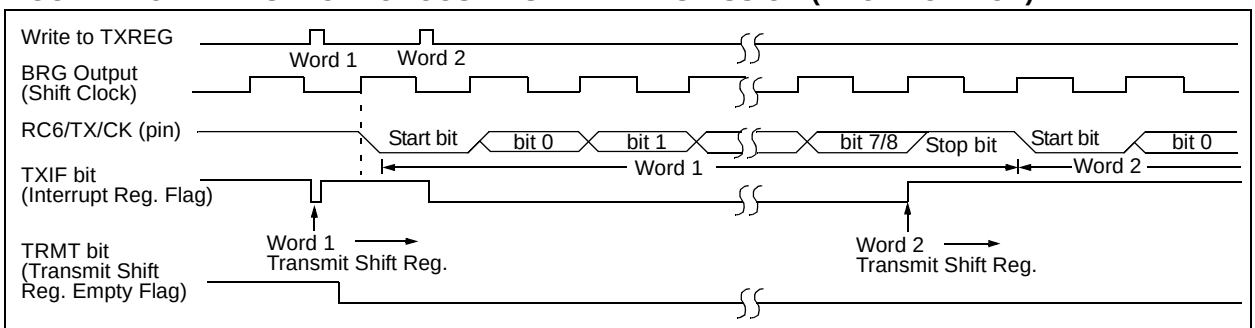


FIGURE 11-3: ASYNCHRONOUS MASTER TRANSMISSION (BACK TO BACK)



Note: This timing diagram shows two consecutive transmissions.

TABLE 11-7: REGISTERS ASSOCIATED WITH ASYNCHRONOUS TRANSMISSION

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other Resets
0Bh, 8Bh, 10Bh,18Bh	INTCON	GIE	PEIE	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF(1)	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
19h	TXREG	AUSART Transmit Data 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 locations read as '0'. Shaded cells are not used for asynchronous transmission.

Note 1: Bits PSPIE and PSPIF are reserved on 28-pin devices; always maintain these bits clear.

12.6 A/D Conversions

Figure 12-3 shows the operation of the A/D converter after the GO/$\overline{\text{DONE}}$ bit has been set and the ACQT2:ACQT0 bits are cleared. A conversion is started after the following instruction to allow entry into Sleep mode before the conversion begins.

Figure 12-4 shows the operation of the A/D converter after the GO/$\overline{\text{DONE}}$ bit has been set, the ACQT2:ACQT0 bits are set to '010' and a 4 TAD acquisition time is selected before the conversion starts.

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. This means 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 completed or aborted, a 2 TAD wait is required before the next acquisition can be started. After this wait, acquisition on the selected channel is automatically started.

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

FIGURE 12-3: A/D CONVERSION TAD CYCLES (ACQT<2:0> = 000, TACQ = 0)

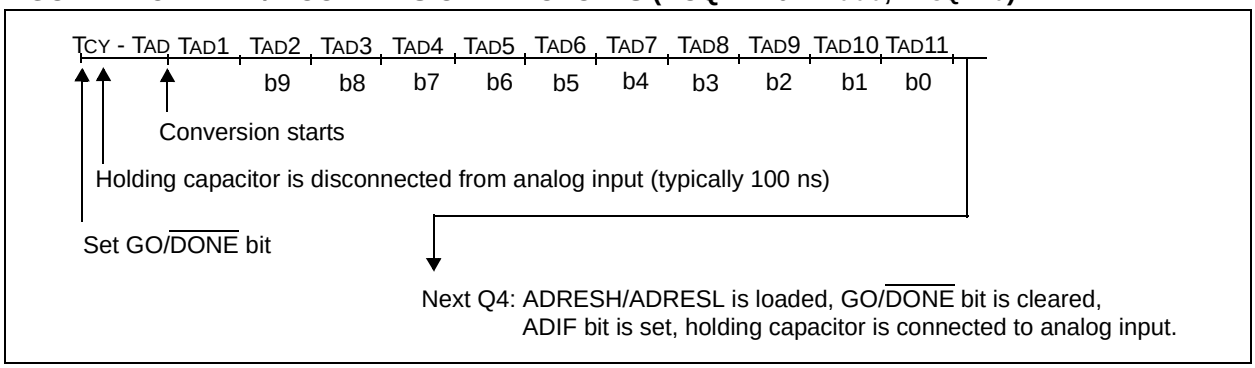
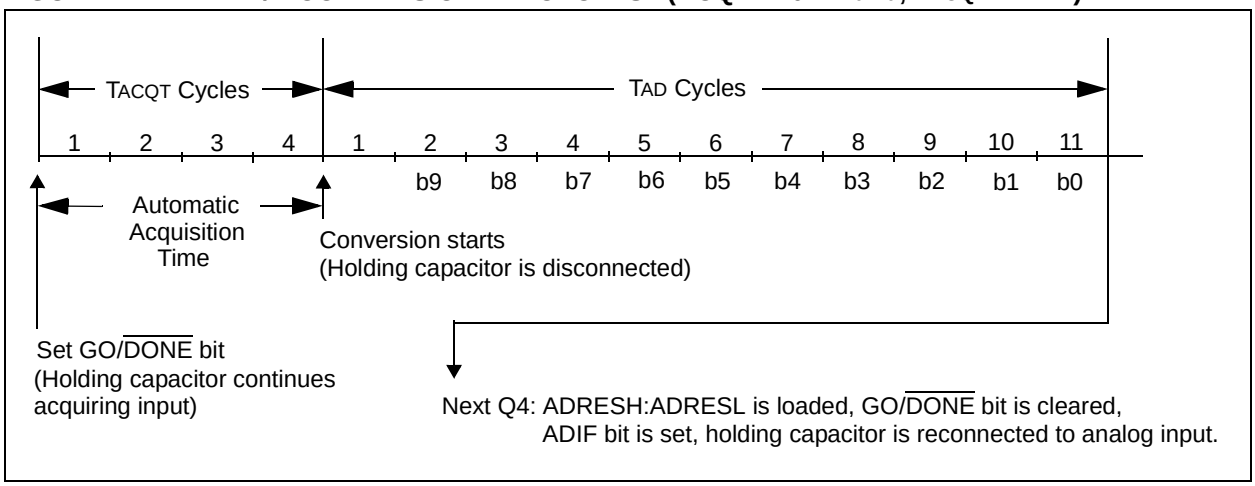


FIGURE 12-4: A/D CONVERSION TAD CYCLES (ACQT<2:0> = 010, TACQ = 4 TAD)

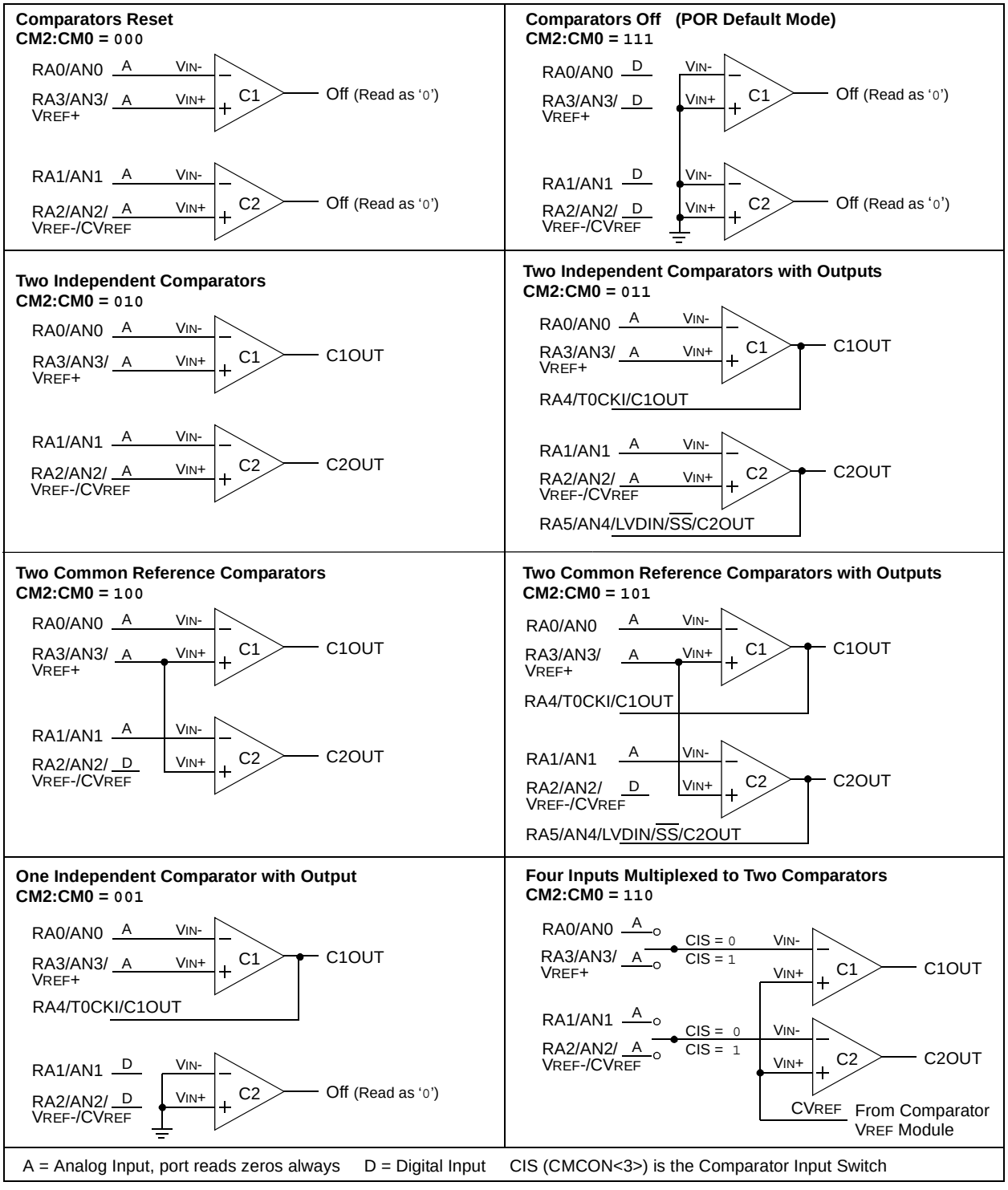


13.1 Comparator Configuration

There are eight modes of operation for the comparators. The CMCON register is used to select these modes. Figure 13-1 shows the eight possible modes. The TRISA register controls the data direction of the comparator pins for each mode. If the Comparator mode is changed, the comparator output level may not be valid for the specified mode change delay shown in the electrical specifications (**Section 18.0 "Electrical Characteristics"**).

Note:	Comparator interrupts should be disabled during a Comparator mode change. Otherwise, a false interrupt may occur.

FIGURE 13-1: COMPARATOR I/O OPERATING MODES



A = Analog Input, port reads zeros always D = Digital Input CIS (CMCON<3>) is the Comparator Input Switch

FIGURE 14-1: COMPARATOR VOLTAGE REFERENCE BLOCK DIAGRAM

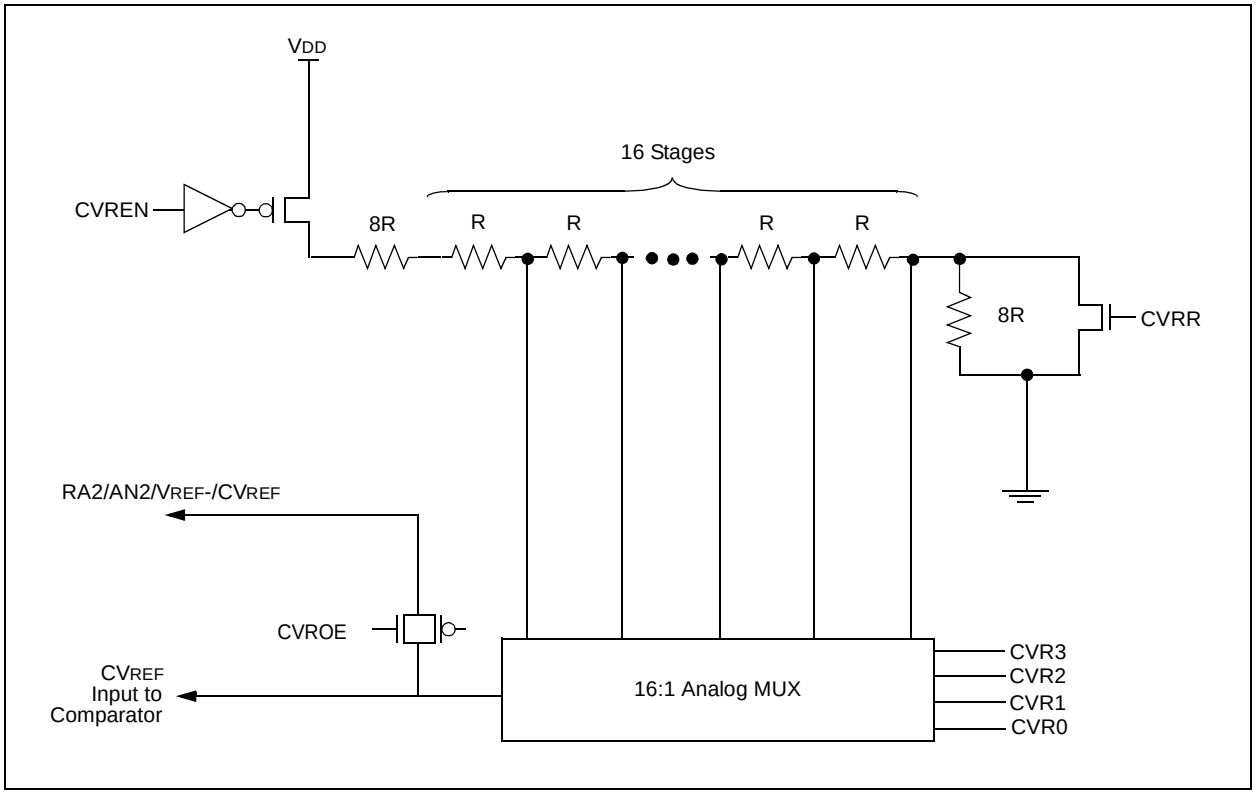


TABLE 14-1: REGISTERS ASSOCIATED WITH COMPARATOR VOLTAGE REFERENCE

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR	Value on all other Resets
9Dh	CVRCON	CVREN	CVROE	CVRR	—	CVR3	CVR2	CVR1	CVR0	000- 0000	000- 0000
9Ch	CMCON	C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0	0000 0111	0000 0111

Legend: x = unknown, u = unchanged, — = unimplemented, read as '0'.
Shaded cells are not used with the comparator voltage reference.

REGISTER 15-1: CONFIGURATION WORD REGISTER 1 (ADDRESS 2007h)

R/P-1	R/P-1	R/P-1	U-1	U-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1
CP	CCPMX	DEBUG	—	—	BORV1	BORV0	BOREN	MCLRE	FOSC2	$\overline{\text{PWRTEN}}$	WDTEN	FOSC1	FOSC0
bit 13													bit 0

bit 13 **CP:** Flash Program Memory Code Protection bits
1 = Code protection off
0 = 0000h to 1FFFh code-protected for PIC16F767/777 and 0000h to 0FFFh for PIC16F737/747 (all protected)

bit 12 **CCPMX:** CCP2 Multiplex bit
1 = CCP2 is on RC1
0 = CCP2 is on RB3

bit 11 **DEBUG:** In-Circuit Debugger Mode bit
1 = In-Circuit Debugger disabled, RB6 and RB7 are general purpose I/O pins
0 = In-Circuit Debugger enabled, RB6 and RB7 are dedicated to the debugger

bit 10-9 **Unimplemented:** Read as '1'

bit 8-7 **BORV<1:0>:** Brown-out Reset Voltage bits
11 = V_{BOR} set to 2.0V
10 = V_{BOR} set to 2.7V
01 = V_{BOR} set to 4.2V
00 = V_{BOR} set to 4.5V

bit 6 **BOREN:** Brown-out Reset Enable bit
BOREN combines with BORSEN to control when BOR is enabled and how it is controlled.
BOREN:BORSEN:
11 = BOR enabled and always on
10 = BOR enabled during operation and disabled during Sleep by hardware
01 = BOR controlled by software bit SBOREN – refer to Register 2-8 (PCON<2>)
00 = BOR disabled

bit 5 **MCLRE:** $\overline{\text{MCLR}}$/V_{PP}/RE3 Pin Function Select bit
1 = $\overline{\text{MCLR}}$/V_{PP}/RE3 pin function is $\overline{\text{MCLR}}$
0 = $\overline{\text{MCLR}}$/V_{PP}/RE3 pin function is digital input only, $\overline{\text{MCLR}}$ gated to '1'

bit 3 **$\overline{\text{PWRTEN}}$:** Power-up Timer Enable bit
1 = PWRT disabled
0 = PWRT enabled

bit 2 **WDTEN:** Watchdog Timer Enable bit
1 = WDT enabled
0 = WDT disabled

bit 4, 1-0 **FOSC2:FOSC0:** Oscillator Selection bits
111 = EXTRC oscillator; CLKO function on OSC2/CLKO/RA6
110 = EXTRC oscillator; port I/O function on OSC2/CLKO/RA6
101 = INTRC oscillator; CLKO function on OSC2/CLKO/RA6 and port I/O function on OSC1/CLKI/RA7
100 = INTRC oscillator; port I/O function on OSC1/CLKI/RA7 and OSC2/CLKO/RA6
011 = EXTCLK; port I/O function on OSC2/CLKO/RA6
010 = HS oscillator
001 = XT oscillator
000 = LP oscillator

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

15.19 In-Circuit Debugger

When the DEBUG bit in the Configuration Word is programmed to a '0', the In-Circuit Debugger functionality is enabled. This function allows simple debugging functions when used with MPLAB® ICD. When the microcontroller has this feature enabled, some of the resources are not available for general use. Table 15-7 shows which features are consumed by the background debugger.

TABLE 15-7: DEBUGGER RESOURCES

I/O pins	RB6, RB7
Stack	1 level
Program Memory	Address 0000h must be NOP
	Last 100h words
Data Memory	0x070 (0x0F0, 0x170, 0x1F0) 0x165-0x16F

To use the In-Circuit Debugger function of the microcontroller, the design must implement In-Circuit Serial Programming connections to $\overline{MCLR}$/VPP, VDD, GND, RB7 and RB6. This will interface to the In-Circuit Debugger module available from Microchip or one of the third party development tool companies.

Note:	In-Circuit Debugger operation must occur between the operating voltage range (VDD) of 4.75V-5.25V on PIC16F7X7 devices.

15.20 Program Verification/Code Protection

If the code protection bit(s) have not been programmed, the on-chip program memory can be read out for verification purposes.

15.21 ID Locations

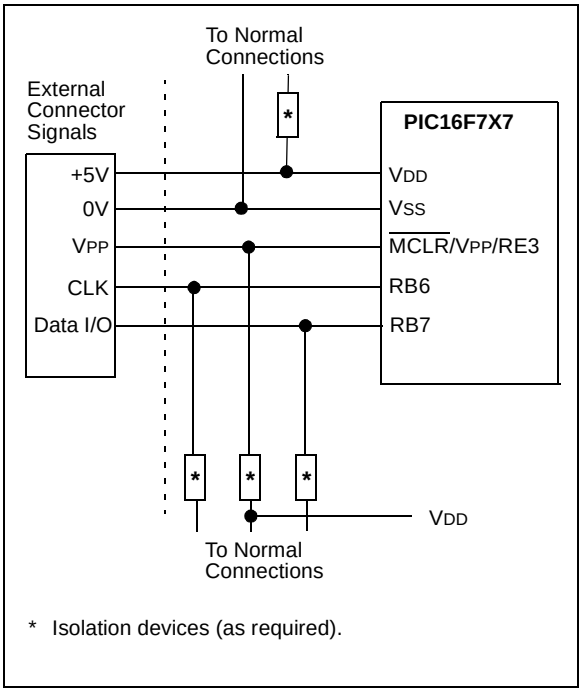
Four memory locations (2000h-2003h) are designated as ID locations where the user can store checksum or other code identification numbers. These locations are not accessible during normal execution but are readable and writable during program/verify. It is recommended that only the four Least Significant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

15.22 In-Circuit Serial Programming

PIC16F7X7 microcontrollers can be serially programmed while in the end application circuit. This is simply done with two lines for clock and data and three other lines for power, ground and the programming voltage (see Figure 15-17 for an example). This allows customers to manufacture boards with unprogrammed devices and then program the microcontroller just before shipping the product. This also allows the most recent firmware or a custom firmware to be programmed.

For general information of serial programming, please refer to the *"In-Circuit Serial Programming™ (ICSP™) Guide"* (DS30277).

FIGURE 15-17: TYPICAL IN-CIRCUIT SERIAL PROGRAMMING™ CONNECTION



* Isolation devices (as required).

16.0 INSTRUCTION SET SUMMARY

The PIC16 instruction set is highly orthogonal and is comprised of three basic categories:

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

Each PIC16 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 formats for each of the categories are presented in Figure 16-1, while the various opcode fields are summarized in Table 16-1.

Table 13-2 lists the instructions recognized by the MPASM™ Assembler. A complete description of each instruction is also available in the *"PIC® Mid-Range MCU Family Reference Manual"* (DS33023).

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 bit affected by the operation, while 'f' represents the address 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

One instruction cycle consists of four oscillator periods; for an oscillator frequency of 4 MHz, this gives a normal instruction execution time of 1 μs. All instructions are executed within a single instruction cycle, unless a conditional test is true, or the program counter is changed as a result of an instruction. When this occurs, the execution takes two instruction cycles, with the second cycle executed as a NOP.

Note:	To maintain upward compatibility with future PIC16F7X7 products, do not use the OPTION and TRIS instructions.

All instruction examples use the format '0xhh' to represent a hexadecimal number, where 'h' signifies a hexadecimal digit.

16.1 Read-Modify-Write Operations

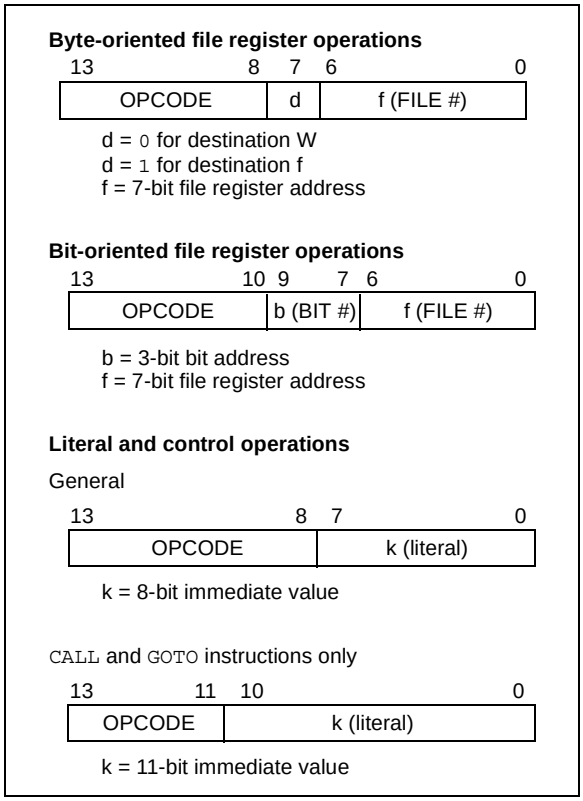
Any instruction that specifies a file register as part of the instruction performs a Read-Modify-Write (R-M-W) operation. The register is read, the data is modified and the result is stored according to either the instruction or the destination designator 'd'. A read operation is performed on a register even if the instruction writes to that register.

For example, a "CLRF PORTB" instruction will read PORTB, clear all the data bits, then write the result back to PORTB. This example would have the unintended result that the condition that sets the RBIF flag would be cleared for pins configured as inputs and using the PORTB interrupt-on-change feature.

TABLE 16-1: OPCODE FIELD DESCRIPTIONS

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

FIGURE 16-1: GENERAL FORMAT FOR INSTRUCTIONS



17.0 DEVELOPMENT SUPPORT

The PIC® microcontrollers and dsPIC® digital signal controllers are supported with a full range of software and hardware development tools:

- Integrated Development Environment
  - MPLAB® IDE Software
- Compilers/Assemblers/Linkers
  - MPLAB C Compiler for Various Device Families
  - HI-TECH C® for Various Device Families
  - MPASM™ Assembler
  - MPLINK™ Object Linker/ MPLIB™ Object Librarian
  - MPLAB Assembler/Linker/Librarian for Various Device Families
- Simulators
  - MPLAB SIM Software Simulator
- Emulators
  - MPLAB REAL ICE™ In-Circuit Emulator
- In-Circuit Debuggers
  - MPLAB ICD 3
  - PICkit™ 3 Debug Express
- Device Programmers
  - PICkit™ 2 Programmer
  - MPLAB PM3 Device Programmer
- Low-Cost Demonstration/Development Boards, Evaluation Kits, and Starter Kits

17.1 MPLAB Integrated Development Environment Software

The MPLAB IDE software brings an ease of software development previously unseen in the 8/16/32-bit microcontroller market. The MPLAB IDE is a Windows® operating system-based application that contains:

- A single graphical interface to all debugging tools
  - Simulator
  - Programmer (sold separately)
  - In-Circuit Emulator (sold separately)
  - In-Circuit Debugger (sold separately)
- A full-featured editor with color-coded context
- A multiple project manager
- Customizable data windows with direct edit of contents
- High-level source code debugging
- Mouse over variable inspection
- Drag and drop variables from source to watch windows
- Extensive on-line help
- Integration of select third party tools, such as IAR C Compilers

The MPLAB IDE allows you to:

- Edit your source files (either C or assembly)
- One-touch compile or assemble, and download to emulator and simulator tools (automatically updates all project information)
- Debug using:
  - Source files (C or assembly)
  - Mixed C and assembly
  - Machine code

MPLAB IDE supports multiple debugging tools in a single development paradigm, from the cost-effective simulators, through low-cost in-circuit debuggers, to full-featured emulators. This eliminates the learning curve when upgrading to tools with increased flexibility and power.

TABLE 18-1: COMPARATOR SPECIFICATIONS

Operating Conditions: 3.0V < V_{DD} < 5.5V, -40°C < T_A < +85°C (unless otherwise stated).							
Param No.	**Sym**	**Characteristics**	**Min**	**Typ**	**Max**	**Units**	**Comments**
D300	V_{IOFF}	Input Offset Voltage	—	± 5.0	± 10	mV	
D301	V_{ICM}	Input Common Mode Voltage*	0	—	V_{DD} – 1.5	V	
D302	CMRR	Common Mode Rejection Ratio*	55	—	—	dB	
300 300A	T_{RESP}	Response Time(1)*	—	150	400 600	ns ns	PIC16F7X7 PIC16LF7X7
301	T_{MC2OV}	Comparator Mode Change to Output Valid*	—	—	10	µs	

* These parameters are characterized but not tested.

Note 1: Response time measured with one comparator input at (V_{DD} – 1.5)/2, while the other input transitions from V_{SS} to V_{DD}.

TABLE 18-2: VOLTAGE REFERENCE SPECIFICATIONS

Operating Conditions: 3.0V < V_{DD} < 5.5V, -40°C < T_A < +85°C (unless otherwise stated).							
Param No.	**Sym**	**Characteristics**	**Min**	**Typ**	**Max**	**Units**	**Comments**
D310	V_{RES}	Resolution	V_{DD}/24	—	V_{DD}/32	LSb	
D311	V_{RAA}	Absolute Accuracy	— —	— —	1/4 1/2	LSb LSb	Low Range (CVRR = 1) High Range (CVRR = 0)
D312	V_{RUR}	Unit Resistor Value (R)*	—	2k	—	Ω	
310	T_{SET}	Settling Time(1)*	—	—	10	µs	

* These parameters are characterized but not tested.

Note 1: Settling time measured while CVRR = 1 and CVR<3:0> transition from '0000' to '1111'.

FIGURE 18-3: LOW-VOLTAGE DETECT CHARACTERISTICS

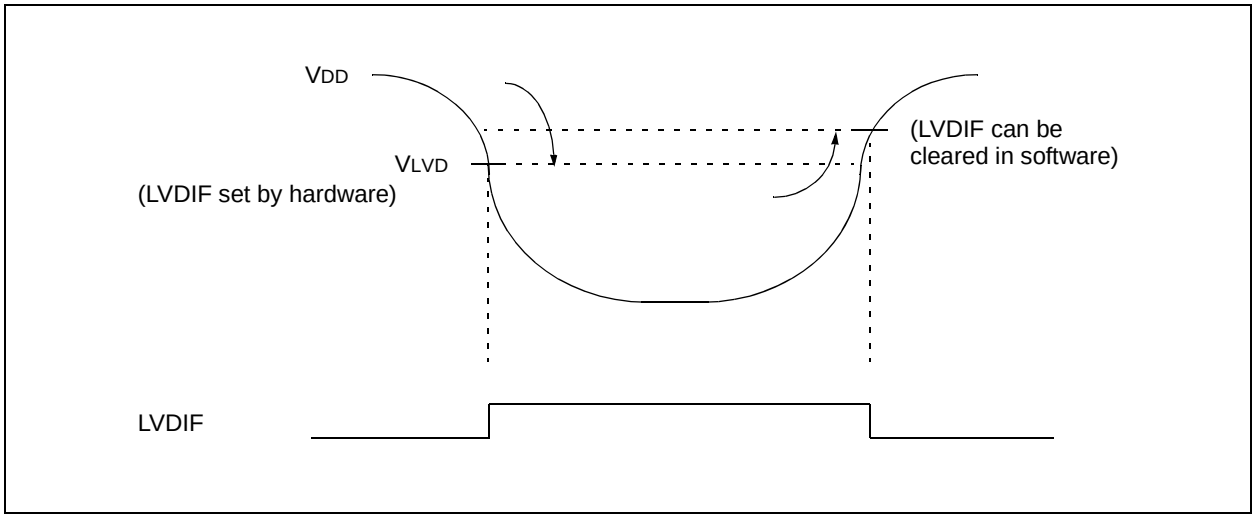


TABLE 18-3: LOW-VOLTAGE DETECT CHARACTERISTICS

Standard Operating Conditions (unless otherwise stated)
Operating temperature -40°C ≤ TA ≤ +85°C for industrial
-40°C ≤ TA ≤ +125°C for extended

Param No.	Symbol	Characteristic		Min	Typ†	Max	Units	Conditions
D420	VLVD	LVD Voltage on VDD Transition High-to-Low	LVDL<3:0> = 0000	N/A	N/A	N/A	V	Reserved
			LVDL<3:0> = 0001	1.96	2.06	2.16	V	T ≥ 25°C
			LVDL<3:0> = 0010	2.16	2.27	2.38	V	T ≥ 25°C
			LVDL<3:0> = 0011	2.35	2.47	2.59	V	T ≥ 25°C
			LVDL<3:0> = 0100	2.43	2.56	2.69	V	
			LVDL<3:0> = 0101	2.64	2.78	2.92	V	
			LVDL<3:0> = 0110	2.75	2.89	3.03	V	
			LVDL<3:0> = 0111	2.95	3.1	3.26	V	
			LVDL<3:0> = 1000	3.24	3.41	3.58	V	
			LVDL<3:0> = 1001	3.43	3.61	3.79	V	
			LVDL<3:0> = 1010	3.53	3.72	3.91	V	
			LVDL<3:0> = 1011	3.72	3.92	4.12	V	
			LVDL<3:0> = 1100	3.92	4.13	4.34	V	
			LVDL<3:0> = 1101	4.11	4.33	4.55	V	
			LVDL<3:0> = 1110	4.41	4.64	4.87	V	

Legend: Shading of rows is to assist in readability of the table.

† Production tested at TAMB = 25°C. Specifications over temperature limits ensured by characterization.