**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 | 40MHz |
| Connectivity | I²C, SPI, UART/USART |
| Peripherals | Brown-out Detect/Reset, LCD, LVD, POR, PWM, WDT |
| Number of I/O | 67 |
| Program Memory Size | 32KB (16K x 16) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 2K x 8 |
| Voltage - Supply (Vcc/Vdd) | 2V ~ 3.6V |
| Data Converters | A/D 12x10b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 80-TQFP |
| Supplier Device Package | 80-TQFP (12x12) |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/pic18f85j90-i-pt |

**TABLE 6-3: PIC18F85J90 FAMILY REGISTER FILE SUMMARY (CONTINUED)**

| File Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR, BOR | Details on page |
|---|---|---|---|---|---|---|---|---|---|---|
| PORTJ(2) | RJ7 | RJ6 | RJ5 | RJ4 | RJ3 | RJ2 | RJ1 | RJ0 | xxxx xxxx | 60, 136 |
| PORTH(2) | RH7 | RH6 | RH5 | RH4 | RH3 | RH2 | RH1 | RH0 | xxxx xxxx | 60, 134 |
| PORTG | RDPU | REPU | RJPU(2) | RG4 | RG3 | RG2 | RG1 | RG0 | 000x xxxx | 60, 132 |
| PORTF | RF7 | RF6 | RF5 | RF4 | RF3 | RF2 | RF1 | — | xxxx xxx- | 60, 130 |
| PORTE | RE7 | RE6 | RE5 | RE4 | RE3 | — | RE1 | RE0 | xxxx x-xx | 61, 127 |
| PORTD | RD7 | RD6 | RD5 | RD4 | RD3 | RD2 | RD1 | RD0 | xxxx xxxx | 61, 125 |
| PORTC | RC7 | RC6 | RC5 | RC4 | RC3 | RC2 | RC1 | RC0 | xxxx xxxx | 61, 123 |
| PORTB | RB7 | RB6 | RB5 | RB4 | RB3 | RB2 | RB1 | RB0 | xxxx xxxx | 61, 120 |
| PORTA | RA7(5) | RA6(5) | RA5 | RA4 | RA3 | RA2 | RA1 | RA0 | xx0x 0000 | 61, 117 |
| SPBRGH1 | EUSART Baud Rate Generator High Byte | | | | | | | | 0000 0000 | 61, 240 |
| BAUDCON1 | ABDOVF | RCIDL | RXDTP | TXCKP | BRG16 | — | WUE | ABDEN | 0100 0-00 | 61, 238 |
| LCDDATA23(2) | S47C3 | S46C3 | S45C3 | S44C3 | S43C3 | S42C3 | S41C3 | S40C3 | xxxx xxxx | 61, 167 |
| LCDDATA22 | S39C3(2) | S38C3(2) | S37C3(2) | S36C3(2) | S35C3(2) | S34C3(2) | S33C3(2) | S32C3 | xxxx xxxx | 61, 167 |
| LCDDATA21 | S31C3 | S30C3 | S29C3 | S28C3 | S27C3 | S26C3 | S25C3 | S24C3 | xxxx xxxx | 61, 167 |
| LCDDATA20 | S23C3 | S22C3 | S21C3 | S20C3 | S19C3 | S18C3 | S17C3 | S16C3 | xxxx xxxx | 61, 167 |
| LCDDATA19 | S15C3 | S14C3 | S13C3 | S12C3 | S11C3 | S10C3 | S09C3 | S08C3 | xxxx xxxx | 61, 167 |
| LCDDATA18 | S07C3 | S06C3 | S05C3 | S04C3 | S03C3 | S02C3 | S01C3 | S00C3 | xxxx xxxx | 61, 167 |
| LCDDATA17(2) | S47C2 | S46C2 | S45C2 | S44C2 | S43C2 | S42C2 | S41C2 | S40C2 | xxxx xxxx | 61, 167 |
| LCDDATA16 | S39C2(2) | S38C2(2) | S37C2(2) | S36C2(2) | S35C2(2) | S34C2(2) | S33C2(2) | S32C2 | xxxx xxxx | 61, 167 |
| LCDDATA15 | S31C2 | S30C2 | S29C2 | S28C2 | S27C2 | S26C2 | S25C2 | S24C2 | xxxx xxxx | 61, 167 |
| LCDDATA14 | S23C2 | S22C2 | S21C2 | S20C2 | S19C2 | S18C2 | S17C2 | S16C2 | xxxx xxxx | 61, 167 |
| LCDDATA13 | S15C2 | S14C2 | S13C2 | S12C2 | S11C2 | S10C2 | S09C2 | S08C2 | xxxx xxxx | 61, 167 |
| LCDDATA12 | S07C2 | S06C2 | S05C2 | S04C2 | S03C2 | S02C2 | S01C2 | S00C2 | xxxx xxxx | 61, 167 |
| LCDDATA11(2) | S47C1 | S46C1 | S45C1 | S44C1 | S43C1 | S42C1 | S41C1 | S40C1 | xxxx xxxx | 61, 167 |
| LCDDATA10 | S39C1(2) | S38C1(2) | S37C1(2) | S36C1(2) | S35C1(2) | S34C1(2) | S33C1(2) | S32C1 | xxxx xxxx | 61, 167 |
| LCDDATA9 | S31C1 | S30C1 | S29C1 | S28C1 | S27C1 | S26C1 | S25C1 | S24C1 | xxxx xxxx | 61, 167 |
| LCDDATA8 | S23C1 | S22C1 | S21C1 | S20C1 | S19C1 | S18C1 | S17C1 | S16C1 | xxxx xxxx | 61, 167 |
| LCDDATA7 | S15C1 | S14C1 | S13C1 | S12C1 | S11C1 | S10C1 | S09C1 | S08C1 | xxxx xxxx | 61, 167 |
| LCDDATA6 | S07C1 | S06C1 | S05C1 | S04C1 | S03C1 | S02C1 | S01C1 | S00C1 | xxxx xxxx | 61, 167 |
| LCDDATA5(2) | S47C0 | S46C0 | S45C0 | S44C0 | S43C0 | S42C0 | S41C0 | S40C0 | xxxx xxxx | 61, 167 |
| CCPR1H | Capture/Compare/PWM Register 1 High Byte | | | | | | | | xxxx xxxx | 61, 154 |
| CCPR1L | Capture/Compare/PWM Register 1 Low Byte | | | | | | | | xxxx xxxx | 61, 154 |
| CCP1CON | — | — | DC1B1 | DC1B0 | CCP1M3 | CCP1M2 | CCP1M1 | CCP1M0 | --00 0000 | 61, 153 |
| CCPR2H | Capture/Compare/PWM Register 2 High Byte | | | | | | | | xxxx xxxx | 61, 154 |
| CCPR2L | Capture/Compare/PWM Register 2 Low Byte | | | | | | | | xxxx xxxx | 62, 154 |
| CCP2CON | — | — | DC2B1 | DC2B0 | CCP2M3 | CCP2M2 | CCP2M1 | CCP2M0 | --00 0000 | 62, 153 |
| SPBRG2 | AUSART Baud Rate Generator Register | | | | | | | | 0000 0000 | 62, 260 |
| RCREG2 | AUSART Receive Register | | | | | | | | 0000 0000 | 62, 265 |
| TXREG2 | AUSART Transmit Register | | | | | | | | 0000 0000 | 62, 263 |
| TXSTA2 | CSRC | TX9 | TXEN | SYNC | — | BRGH | TRMT | TX9D | 0000 -010 | 62, 258 |
| RCSTA2 | SPEN | RX9 | SREN | CREN | ADDEN | FERR | OERR | RX9D | 0000 000x | 62, 259 |

**Legend:** x = unknown, u = unchanged, - = unimplemented, q = value depends on condition, r = reserved, do not modify

**Note 1:** Bit 21 of the PC is only available in Test mode and Serial Programming modes.

**2:** These registers and/or bits are available only on 80-pin devices; otherwise, they are unimplemented and read as '0'. Reset states shown are for 80-pin devices.

**3:** Alternate names and definitions for these bits when the MSSP module is operating in $I^2C$™ Slave mode. See **Section 17.4.3.2 "Address Masking"** for details.

**4:** The PLLEN bit is only available in specific oscillator configurations; otherwise, it is disabled and reads as '0'. See **Section 3.4.3 "PLL Frequency Multiplier"** for details.

**5:** RA6/RA7 and their associated latch and direction bits are configured as port pins only when the internal oscillator is selected as the default clock source (FOSC2 Configuration bit = 0); otherwise, they are disabled and these bits read as '0'.

### 6.3.5 STATUS REGISTER

The STATUS register, shown in Register 6-2, contains the arithmetic status of the ALU. The STATUS register can be the operand for any instruction, as with any other register. If the STATUS register is the destination for an instruction that affects the Z, DC, C, OV or N bits, then the write to these five bits is disabled.

These bits are set or cleared according to the device logic. Therefore, the result of an instruction with the STATUS register as destination may be different than intended. For example, `CLRF STATUS` will set the Z bit but leave the other bits unchanged. The STATUS register then reads back as '`000u u1uu`'. It is recommended, therefore, that only `BCF`, `BSF`, `SWAPF`, `MOVFF` and `MOVWF` instructions are used to alter the STATUS register because these instructions do not affect the Z, C, DC, OV or N bits in the STATUS register.

For other instructions not affecting any Status bits, see the instruction set summaries in Table 24-2 and Table 24-3.

**Note:** The C and DC bits operate as a $\overline{\text{borrow}}$ and $\overline{\text{digit borrow}}$ bit respectively, in subtraction.

**REGISTER 6-2: STATUS REGISTER**

| U-0 | U-0 | U-0 | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x |
|---|---|---|---|---|---|---|---|
| — | — | — | N | OV | Z | DC(1) | C(2) |
| bit 7 | | | | | | | bit 0 |

| **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 |

bit 7-5 **Unimplemented:** Read as '0'

bit 4 **N:** Negative bit
This bit is used for signed arithmetic (2's complement). It indicates whether the result was negative (ALU MSB = 1).
1 = Result was negative
0 = Result was positive

bit 3 **OV:** Overflow bit
This bit is used for signed arithmetic (2's complement). It indicates an overflow of the 7-bit magnitude which causes the sign bit (bit 7) to change state.
1 = Overflow occurred for signed arithmetic (in this arithmetic operation)
0 = No overflow occurred

bit 2 **Z:** Zero bit
1 = The result of an arithmetic or logic operation is zero
0 = The result of an arithmetic or logic operation is not zero

bit 1 **DC:** Digit Carry/$\overline{\text{Borrow}}$ bit(1)
For `ADDWF, ADDLW, SUBLW` and `SUBWF` instructions:
1 = A carry-out from the 4th low-order bit of the result occurred
0 = No carry-out from the 4th low-order bit of the result

bit 0 **C:** Carry/$\overline{\text{Borrow}}$ bit(2)
For `ADDWF, ADDLW, SUBLW` and `SUBWF` instructions:
1 = A carry-out from the Most Significant bit of the result occurred
0 = No carry-out from the Most Significant bit of the result occurred

**Note 1:** For $\overline{\text{borrow}}$, the polarity is reversed. A subtraction is executed by adding the 2's complement of the second operand. For rotate (`RRF, RLF`) instructions, this bit is loaded with either bit 4 or bit 3 of the source register.

**2:** For $\overline{\text{borrow}}$, the polarity is reversed. A subtraction is executed by adding the 2's complement of the second operand. For rotate (`RRF, RLF`) instructions, this bit is loaded with either the high or low-order bit of the source register.

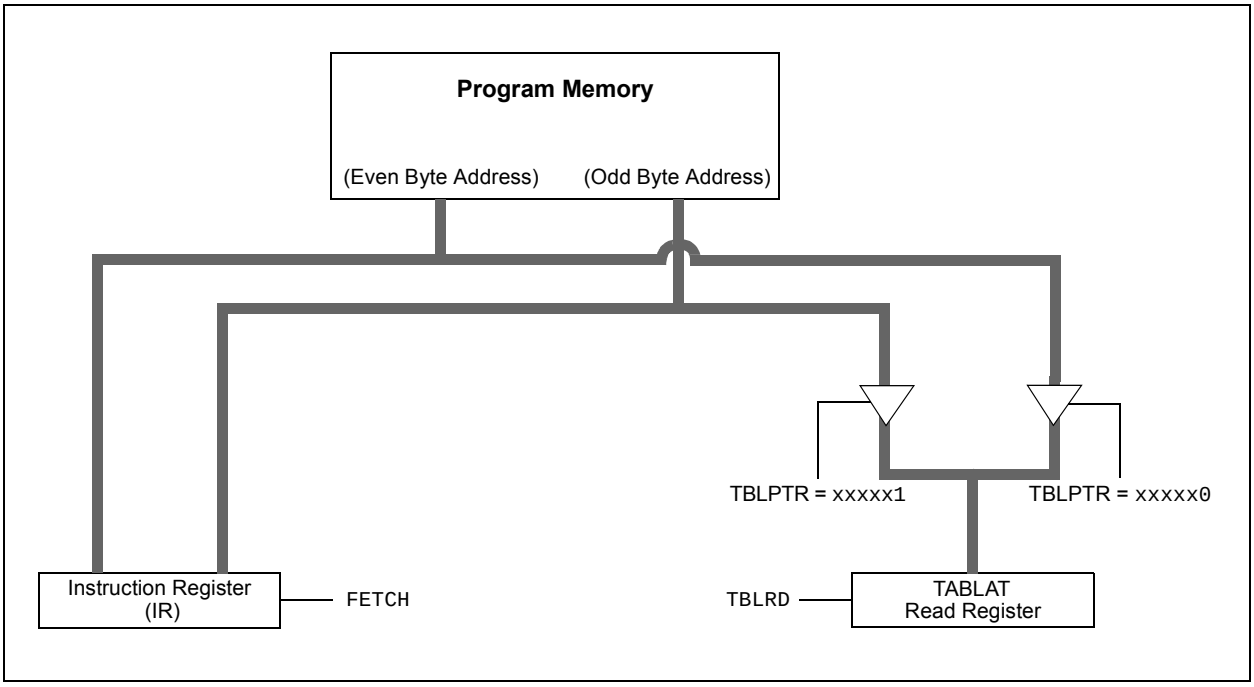
## 7.3 Reading the Flash Program Memory

The TBLRD instruction is used to retrieve data from program memory and places it into data RAM. Table reads from program memory are performed one byte at a time.

The TBLPTR points to a byte address in program space. Executing TBLRD places the byte pointed to into TABLAT. In addition, the TBLPTR can be modified automatically for the next table read operation.

The internal program memory is typically organized by words. The Least Significant bit of the address selects between the high and low bytes of the word. Figure 7-4 shows the interface between the internal program memory and the TABLAT.

**FIGURE 7-4: READS FROM FLASH PROGRAM MEMORY**




**EXAMPLE 7-1: READING A FLASH PROGRAM MEMORY WORD**

```
            MOVLW     CODE_ADDR_UPPER           ; Load TBLPTR with the base
            MOVWF     TBLPTRU                   ; address of the word
            MOVLW     CODE_ADDR_HIGH
            MOVWF     TBLPTRH
            MOVLW     CODE_ADDR_LOW
            MOVWF     TBLPTRL
READ_WORD
            TBLRD*+                             ; read into TABLAT and increment
            MOVF      TABLAT, W                 ; get data
            MOVWF     WORD_EVEN
            TBLRD*+                             ; read into TABLAT and increment
            MOVF      TABLAT, W                 ; get data
            MOVWF     WORD_ODD
```

## 7.4 Erasing Flash Program Memory

The minimum erase block is 512 words or 1024 bytes. Only through the use of an external programmer, or through ICSP control, can larger blocks of program memory be Bulk Erased. Word erase in the Flash array is not supported.

When initiating an erase sequence from the microcontroller itself, a block of 1024 bytes of program memory is erased. The Most Significant 12 bits of the TBLPTR<21:10> point to the block being erased; TBLPTR<9:0> are ignored.

The EECON1 register commands the erase operation. The WREN bit must be set to enable write operations. The FREE bit is set to select an erase operation. For protection, the write initiate sequence for EECON2 must be used.

A long write is necessary for erasing the internal Flash. Instruction execution is halted while in a long write cycle. The long write will be terminated by the internal programming timer.

### 7.4.1 FLASH PROGRAM MEMORY ERASE SEQUENCE

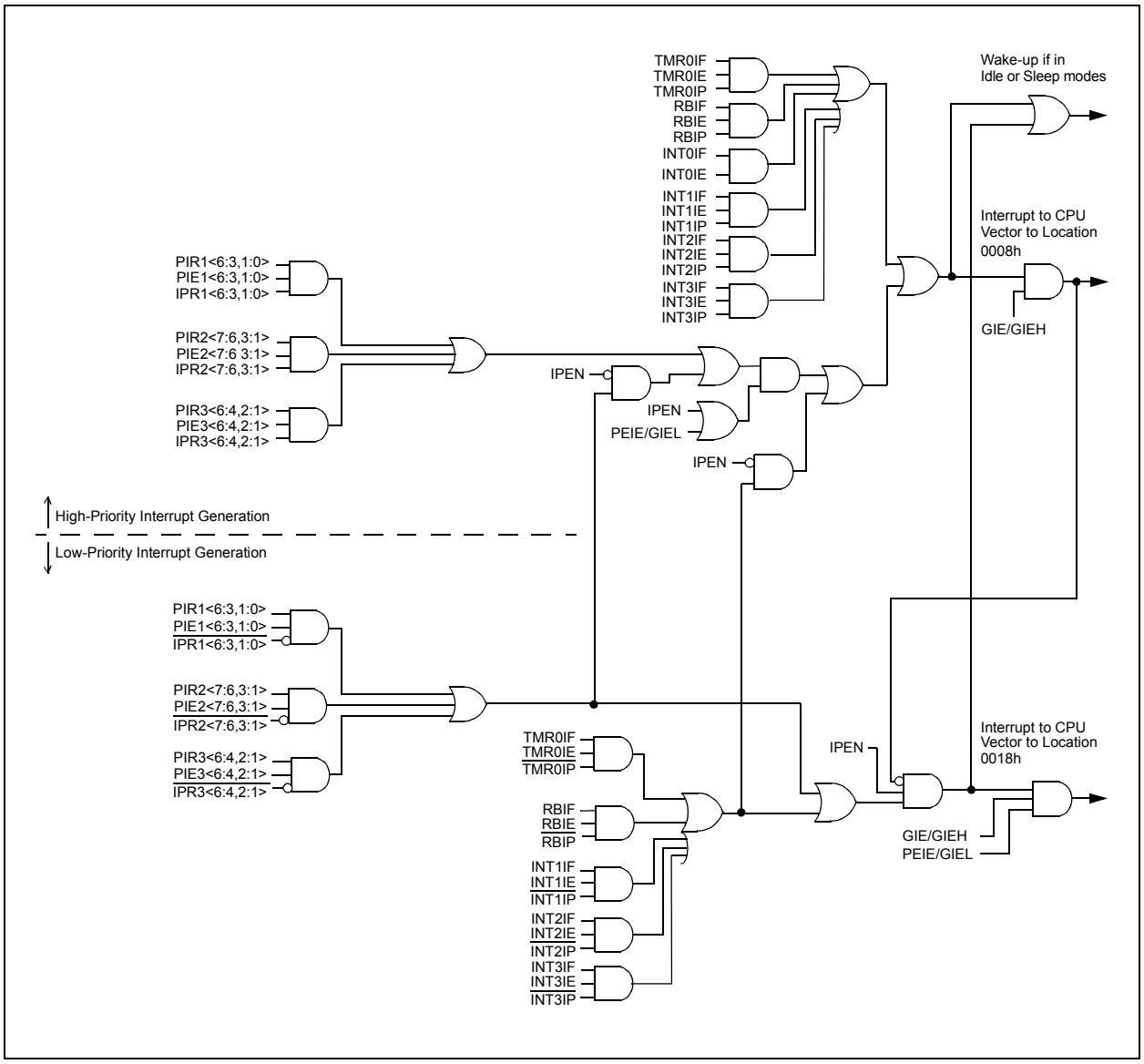
The sequence of events for erasing a block of internal program memory location is:

1. Load the Table Pointer register with the address of the block being erased.
2. Set the WREN and FREE bits (EECON1<2,4>) to enable the erase operation.
3. Disable interrupts.
4. Write 55h to EECON2.
5. Write 0AAh to EECON2.
6. Set the WR bit; this will begin the erase cycle.
7. The CPU will stall for the duration of the erase for TIE (see parameter D133B).
8. Re-enable interrupts.

**EXAMPLE 7-2: ERASING A FLASH PROGRAM MEMORY BLOCK**

```
                    MOVLW  CODE_ADDR_UPPER   ; load TBLPTR with the base
                    MOVWF  TBLPTRU           ; address of the memory block
                    MOVLW  CODE_ADDR_HIGH
                    MOVWF  TBLPTRH
                    MOVLW  CODE_ADDR_LOW
                    MOVWF  TBLPTRL
        ERASE_BLOCK
                    BSF    EECON1, WREN      ; enable write to memory
                    BSF    EECON1, FREE      ; enable Erase operation
                    BCF    INTCON, GIE       ; disable interrupts
    Required        MOVLW  55h
    Sequence        MOVWF  EECON2            ; write 55h
                    MOVLW  0AAh
                    MOVWF  EECON2            ; write 0AAh
                    BSF    EECON1, WR        ; start erase (CPU stall)
                    BSF    INTCON, GIE       ; re-enable interrupts
```

**FIGURE 9-1: PIC18F85J90 FAMILY INTERRUPT LOGIC**



**REGISTER 9-3: INTCON3: INTERRUPT CONTROL REGISTER 3**

| R/W-1 | R/W-1 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| INT2IP | INT1IP | INT3IE | INT2IE | INT1IE | INT3IF | INT2IF | INT1IF |
| bit 7 | | | | | | | bit 0 |

| **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 |

bit 7 **INT2IP:** INT2 External Interrupt Priority bit
1 = High priority
0 = Low priority

bit 6 **INT1IP:** INT1 External Interrupt Priority bit
1 = High priority
0 = Low priority

bit 5 **INT3IE:** INT3 External Interrupt Enable bit
1 = Enables the INT3 external interrupt
0 = Disables the INT3 external interrupt

bit 4 **INT2IE:** INT2 External Interrupt Enable bit
1 = Enables the INT2 external interrupt
0 = Disables the INT2 external interrupt

bit 3 **INT1IE:** INT1 External Interrupt Enable bit
1 = Enables the INT1 external interrupt
0 = Disables the INT1 external interrupt

bit 2 **INT3IF:** INT3 External Interrupt Flag bit
1 = The INT3 external interrupt occurred (must be cleared in software)
0 = The INT3 external interrupt did not occur

bit 1 **INT2IF:** INT2 External Interrupt Flag bit
1 = The INT2 external interrupt occurred (must be cleared in software)
0 = The INT2 external interrupt did not occur

bit 0 **INT1IF:** INT1 External Interrupt Flag bit
1 = The INT1 external interrupt occurred (must be cleared in software)
0 = The INT1 external interrupt did not occur

**Note:** Interrupt flag bits are set when an interrupt condition occurs regardless of the state of its corresponding enable bit or the global interrupt enable bit. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt. This feature allows for software polling.

**REGISTER 9-11: IPR2: PERIPHERAL INTERRUPT PRIORITY REGISTER 2**

| R/W-1 | R/W-1 | U-0 | U-0 | R/W-1 | R/W-1 | R/W-1 | U-0 |
|---|---|---|---|---|---|---|---|
| OSCFIP | CMIP | — | — | BCLIP | LVDIP | TMR3IP | — |
| bit 7 | | | | | | | bit 0 |

| **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 |

bit 7 **OSCFIP:** Oscillator Fail Interrupt Priority bit
1 = High priority
0 = Low priority

bit 6 **CMIP:** Comparator Interrupt Priority bit
1 = High priority
0 = Low priority

bit 5-4 **Unimplemented:** Read as '0'

bit 3 **BCLIP:** Bus Collision Interrupt Priority bit
1 = High priority
0 = Low priority

bit 2 **LVDIP:** Low-Voltage Detect Interrupt Priority bit
1 = High priority
0 = Low priority

bit 1 **TMR3IP:** TMR3 Overflow Interrupt Priority bit
1 = High priority
0 = Low priority

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

## 16.0 LIQUID CRYSTAL DISPLAY (LCD) DRIVER MODULE

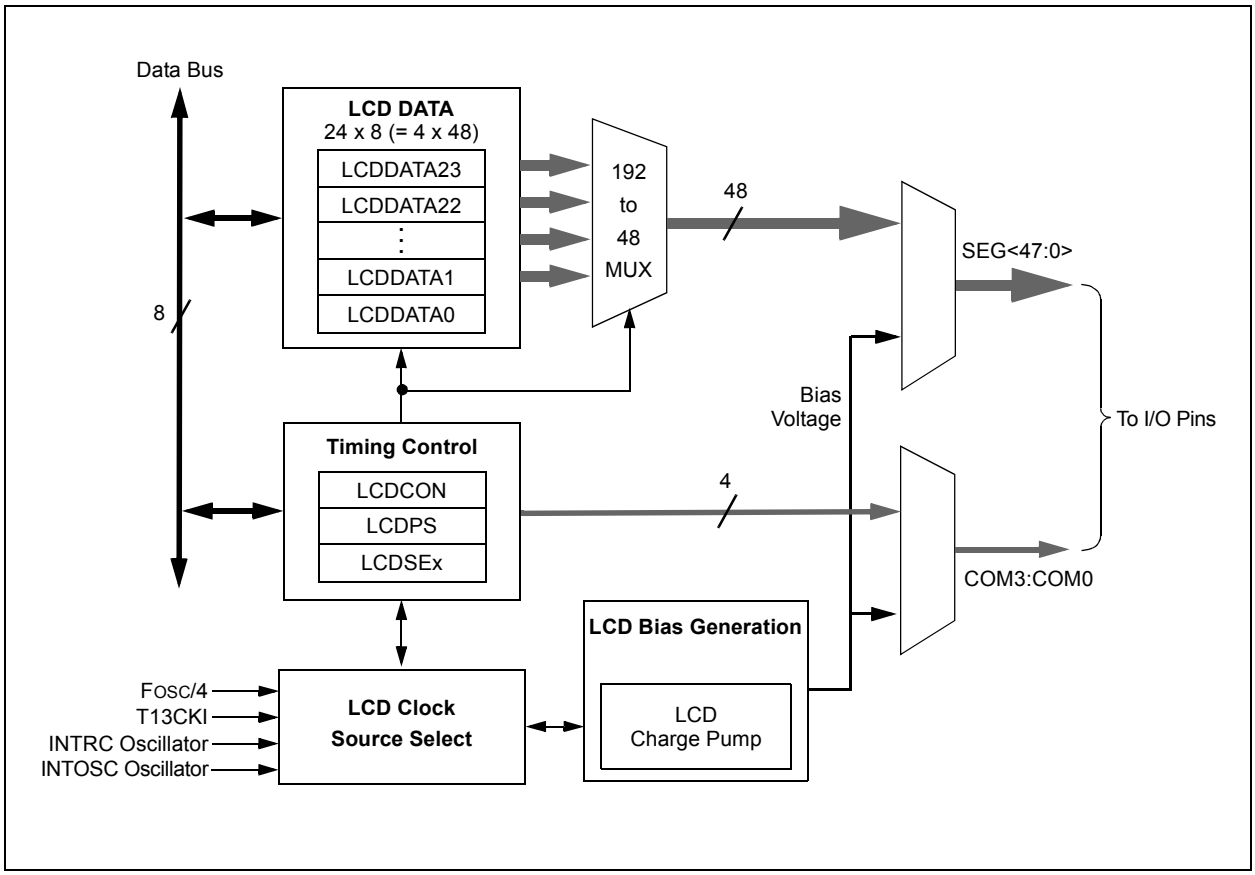
The Liquid Crystal Display (LCD) driver module generates the timing control to drive a static or multiplexed LCD panel. It also provides control of the LCD pixel data. The module can drive panels of up to 192 pixels (48 segments by 4 commons) in 80-pin devices, and 132 pixels (33 segments by 4 commons) in 64-pin devices.

The LCD driver module supports these features:

- Direct driving of LCD panel
- On-chip bias generator with dedicated charge pump to support a range of fixed and variable bias options
- Up to four commons, with four Multiplexing modes
- Up to 48 (80-pin devices) or 33 (64-pin devices) segments
- Three LCD clock sources with selectable prescaler, with a fourth source available for use with the LCD charge pump

A simplified block diagram of the module is shown in Figure 16-1.

**FIGURE 16-1: LCD DRIVER MODULE BLOCK DIAGRAM**



## 16.1 LCD Registers

The LCD driver module has 33 registers:

- LCD Control Register (LCDCON)
- LCD Phase Register (LCDPS)
- LCD Regulator Control Register (LCDREG)
- Six LCD Segment Enable Registers (LCDSE5:LCDSE0)
- 24 LCD Data Registers (LCDDATA23:LCDDATA0)

### 16.1.1 LCD CONTROL REGISTERS

The LCDCON register, shown in Register 16-1, controls the overall operation of the module. 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.

The LCDPS register, shown in Register 16-2, configures the LCD clock source prescaler and the type of waveform: Type-A or Type-B. Details on these features are provided in **Section 16.2 "LCD Clock Source"**, **Section 16.3 "LCD Bias Generation"** and **Section 16.8 "LCD Waveform Generation"**.

The LCDREG register is described in **Section 16.3 "LCD Bias Generation"**.

The LCD Segment Enable registers (LCDSEx) configure the functions of the port pins. Setting the segment enable bit for a particular segment configures that pin as an LCD driver. The prototype LCDSE register is shown in Register 16-3. There are six LCDSE registers (LCDSE5:LCDSE0) listed in Table 16-1.

**REGISTER 16-1: LCDCON: LCD CONTROL REGISTER**

| R/W-0 | R/W-0 | R/C-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| LCDEN | SLPEN | WERR | — | CS1 | CS0 | LMUX1 | LMUX0 |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | C = Clearable bit | | |
|---|---|---|---|
| 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 |

bit 7 **LCDEN:** LCD Driver Enable bit
1 = LCD driver module is enabled
0 = LCD driver module is disabled

bit 6 **SLPEN:** LCD Driver Enable in Sleep mode bit
1 = LCD driver module is disabled in Sleep mode
0 = LCD driver module is enabled in Sleep mode

bit 5 **WERR:** LCD Write Failed Error bit
1 = LCDDATAx register written while LCDPS<4> = 0 (must be cleared in software)
0 = No LCD write error

bit 4 **Unimplemented:** Read as '0'

bit 3-2 **CS<1:0>:** Clock Source Select bits
1x = INTRC (31 kHz)
01 = T13CKI (Timer1)
00 = System clock (FOSC/4)

bit 1-0 **LMUX<1:0>:** Commons Select bits

| LMUX<1:0> | Multiplex Type | Maximum Number of Pixels: | | Bias Type |
|---|---|---|---|---|
| | | PIC18F6XJ90 | PIC18F8XJ90 | |
| 00 | Static (COM0) | 33 | 48 | Static |
| 01 | 1/2 (COM1:COM0) | 66 | 96 | 1/2 or 1/3 |
| 10 | 1/3 (COM2:COM0) | 99 | 144 | 1/2 or 1/3 |
| 11 | 1/4 (COM3:COM0) | 132 | 192 | 1/3 |

## 16.2 LCD Clock Source

The LCD driver module generates its internal clock from 3 possible sources:

- System clock (FOSC/4)
- Timer1 oscillator
- INTRC source

The LCD clock generator uses a configurable divide-by-32/divide-by-8192 postscaler to produce a baseline frequency of about 1 kHz nominal, regardless of the source selected. The clock source selection and the postscaler configuration are determined by the Clock Source Select bits, CS<1:0> (LCDCON<3:2>).

An additional programmable prescaler is used to derive the LCD frame frequency from the 1 kHz baseline. The prescaler is configured using the LP<3:0> bits (LCDPS<3:0>) for any one of 16 options, ranging from 1:1 to 1:16.

Proper timing for waveform generation is set by the LMUX<1:0> bits (LCDCON<1:0>). These bits determine which Commons Multiplexing mode is to be used, and divide down the LCD clock source as required. They also determine the configuration of the ring counter that is used to switch the LCD commons on or off.

### 16.2.1 LCD VOLTAGE REGULATOR CLOCK SOURCE

In addition to the clock source for LCD timing, a separate 31 kHz nominal clock is required for the LCD charge pump. This is provided from a distinct branch of the LCD clock source.

The charge pump clock can use either the Timer1 oscillator or the INTRC source, as well as the 8 MHz INTOSC source (after being divided by 256 by a prescaler). The charge pump clock source is configured using the CKSEL<1:0> bits (LCDREG<1:0>).

### 16.2.2 CLOCK SOURCE CONSIDERATIONS

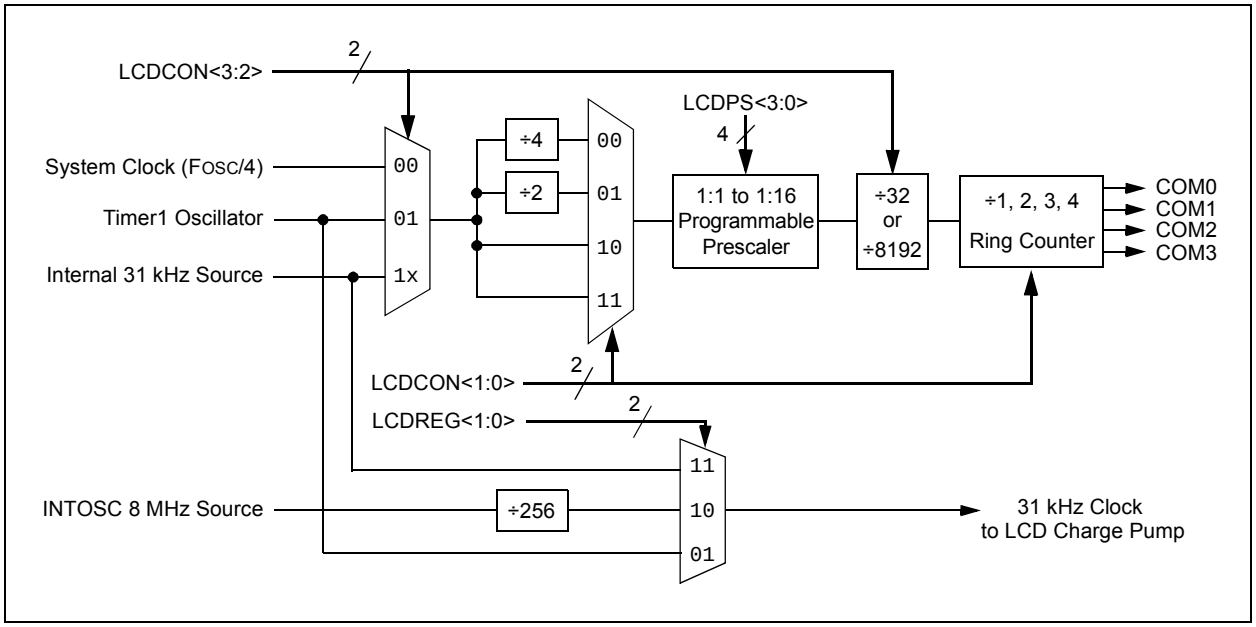
When using the system clock as the LCD clock source, it is assumed that the system clock frequency is a nominal 32 MHz (for a FOSC/4 frequency of 8 MHz). Because the prescaler option for the FOSC/4 clock selection is fixed at divide-by-8192, system clock speeds that differ from 32 MHz will produce frame frequencies and refresh rates different than discussed in this chapter. The user will need to keep this in mind when designing the display application.

The Timer1 and INTRC sources can be used as LCD clock sources when the device is in Sleep mode. To use the Timer1 oscillator, it is necessary to set the T1OSCEN bit (T1CON<3>). Selecting either Timer1 or INTRC as the LCD clock source will not automatically activate these sources.

Similarly, selecting the INTOSC as the charge pump clock source will not turn the oscillator on. To use INTOSC, it must be selected as the system clock source by using the FOSC2 Configuration bit.

If Timer1 is used as a clock source for the device, either as an LCD clock source or for any other purpose, LCD segment 32 become unavailable.

**FIGURE 16-2: LCD CLOCK GENERATION**



# 17.0 MASTER SYNCHRONOUS SERIAL PORT (MSSP) MODULE

## 17.1 Master SSP (MSSP) Module Overview

The Master Synchronous Serial Port (MSSP) module is a serial interface, useful for communicating with other peripheral or microcontroller devices. These peripheral devices may be serial EEPROMs, shift registers, display drivers, A/D Converters, etc. The MSSP module can operate in one of two modes:

- Serial Peripheral Interface (SPI)
- Inter-Integrated Circuit ($I^2C$™)
  - Full Master mode
  - Slave mode (with general address call)

The $I^2C$ interface supports the following modes in hardware:

- Master mode
- Multi-Master mode
- Slave mode

## 17.2 Control Registers

Each MSSP module has three associated control registers. These include a status register (SSPSTAT) and two control registers (SSPCON1 and SSPCON2). The use of these registers and their individual bits differ significantly depending on whether the MSSP module is operated in SPI or $I^2C$ mode.

Additional details are provided under the individual sections.

## 17.3 SPI Mode

The SPI mode allows 8 bits of data to be synchronously transmitted and received simultaneously. All four modes of SPI are supported. To accomplish communication, typically three pins are used:

- Serial Data Out (SDO) – RC5/SDO
- Serial Data In (SDI) – RC4/SDI/SDA
- Serial Clock (SCK) – RC3/SCK/SCL

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

- Slave Select ($\overline{SS}$) – RF7/$\overline{SS}$

| **Note:** | Disabling the MSSP module by clearing the SSPEN (SSPCON1<5>) bit may not reset the module. It is recommended to clear the SSPSTAT, SSPCON1 and SSPCON2 registers and select the mode prior to setting the SSPEN bit to enable the MSSP module. |
|---|---|

Figure 17-1 shows the block diagram of the MSSP module when operating in SPI mode.

| **Note:** | Disabling the MSSP module by clearing the SSPEN (SSPCON1<5>) bit may not reset the module. It is recommended to clear the SSPSTAT, SSPCON1 and SSPCON2 registers and select the mode prior to setting the SSPEN bit to enable the MSSP module. |
|---|---|

**FIGURE 17-1: MSSP BLOCK DIAGRAM (SPI MODE)**

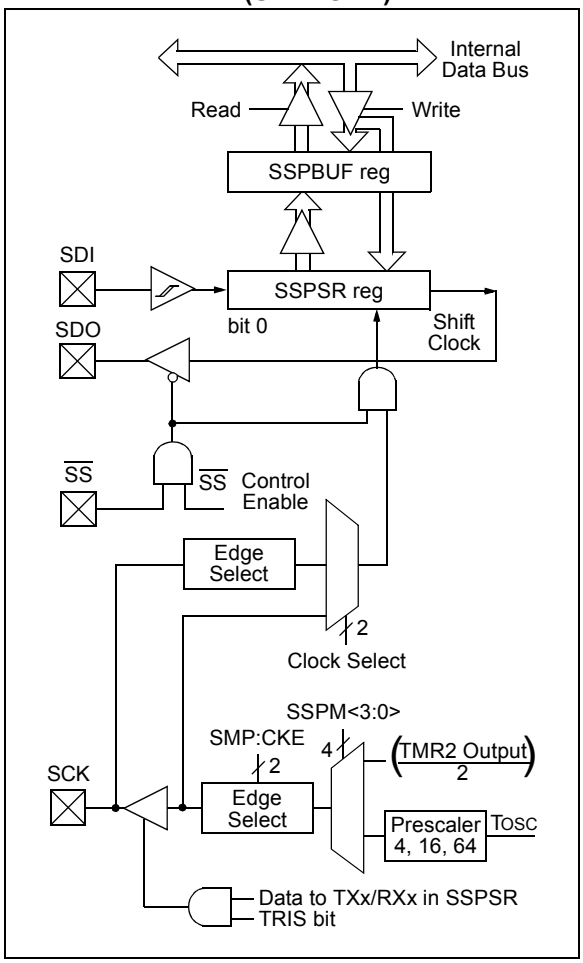
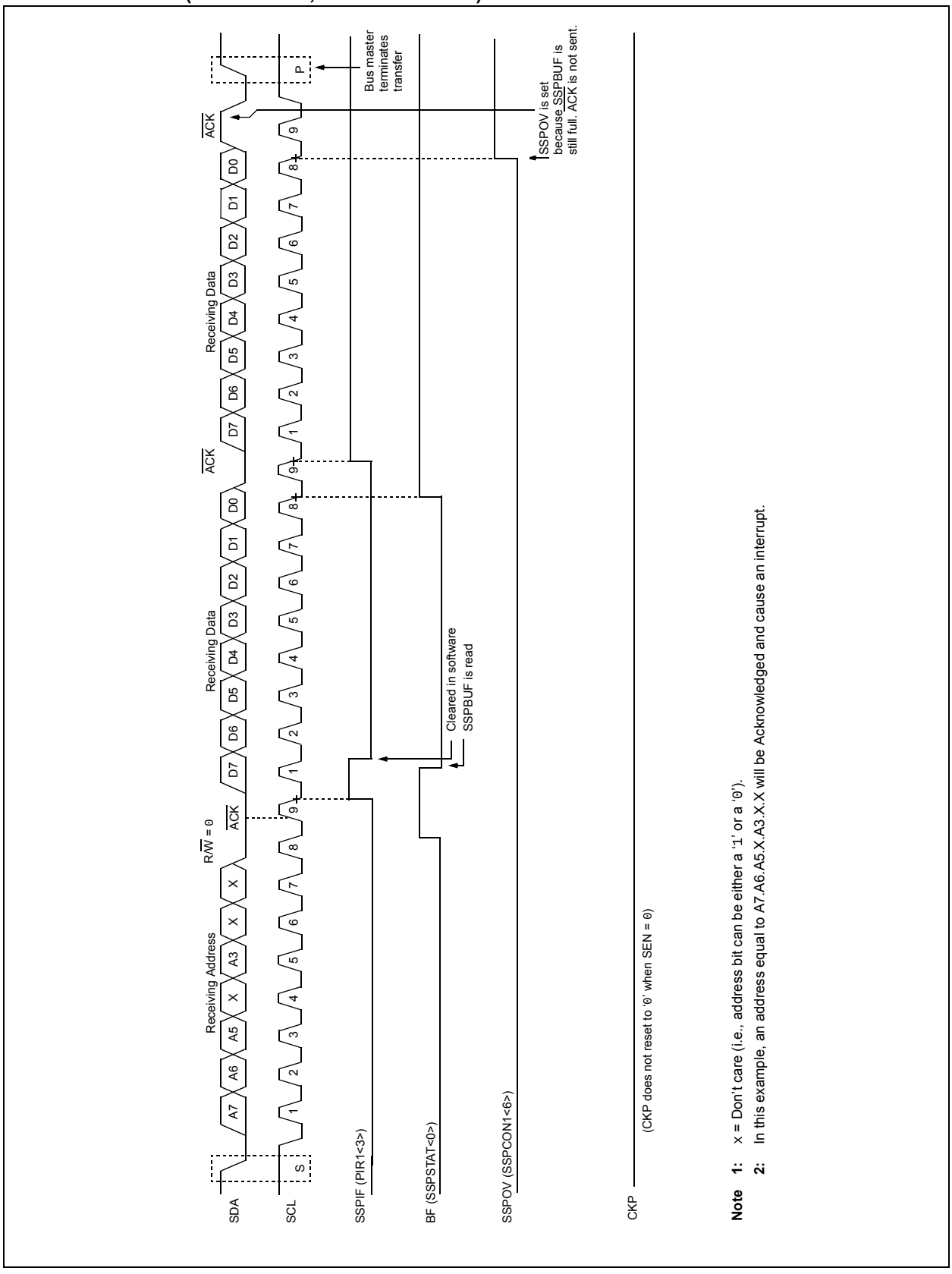


**FIGURE 17-9: I²C™ SLAVE MODE TIMING WITH SEN = 0 AND ADMSK<5:1> = 01011 (RECEPTION, 7-BIT ADDRESS)**




**Note 1:** x = Don't care (i.e., address bit can be either a '1' or a '0').

**2:** In this example, an address equal to A7.A6.A5.X.A3.X.X will be Acknowledged and cause an interrupt.

After the A/D module has been configured as desired, the selected channel must be acquired before the conversion is started. The analog input channels must have their corresponding TRIS bits selected as an input. To determine acquisition time, see **Section 20.1 "A/D Acquisition Requirements"**. After this acquisition time has elapsed, the A/D conversion can be started. An acquisition time can be programmed to occur between setting the GO/$\overline{\text{DONE}}$ bit and the actual start of the conversion.

The following steps should be followed to do an A/D conversion:

1. Configure the A/D module:
   - Configure analog pins, voltage reference and digital I/O (ADCON1)
   - Select A/D input channel (ADCON0)
   - Select A/D acquisition time (ADCON2)
   - Select A/D conversion clock (ADCON2)
   - Turn on A/D module (ADCON0)
2. Configure A/D interrupt (if desired):
   - Clear ADIF bit
   - Set ADIE bit
   - Set GIE bit
3. Wait the required acquisition time (if required).
4. Start conversion:
   - Set GO/$\overline{\text{DONE}}$ bit (ADCON0<1>)
5. Wait for A/D conversion to complete, by either:
   - Polling for the GO/$\overline{\text{DONE}}$ bit to be cleared

   OR
   - Waiting for the A/D interrupt
6. Read A/D Result registers (ADRESH:ADRESL); clear ADIF bit, if required.
7. For next conversion, go to step 1 or step 2, as required. The A/D conversion time per bit is defined as TAD. A minimum wait of 2 TAD is required before next acquisition starts.

**FIGURE 20-2: ANALOG INPUT MODEL**

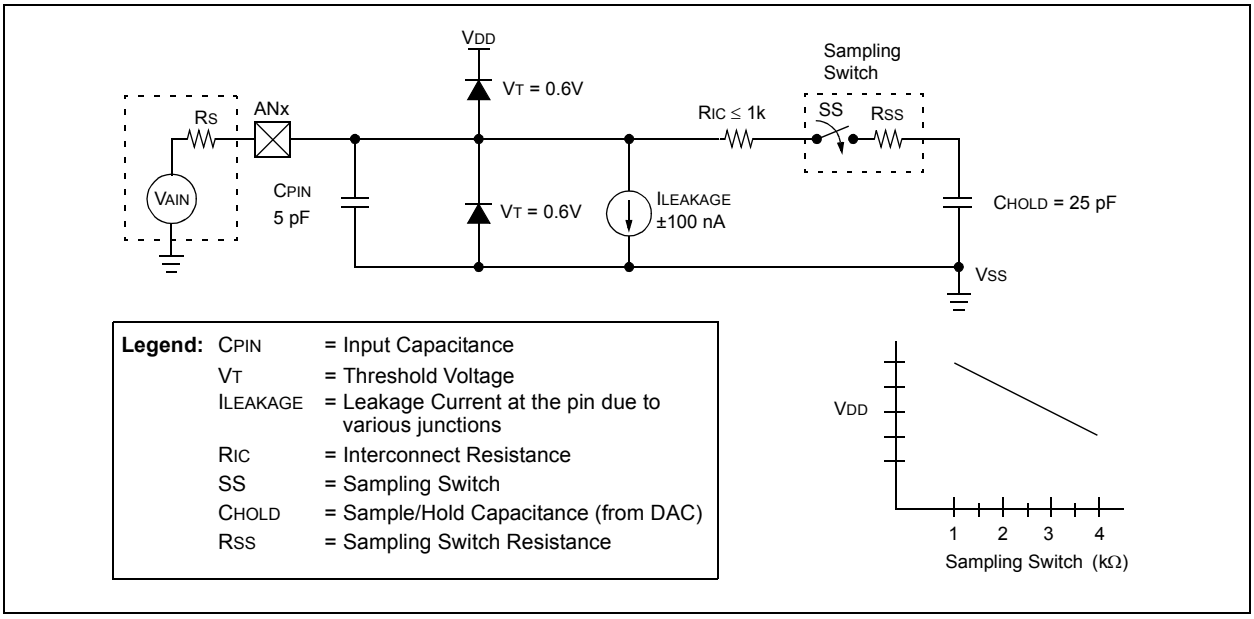
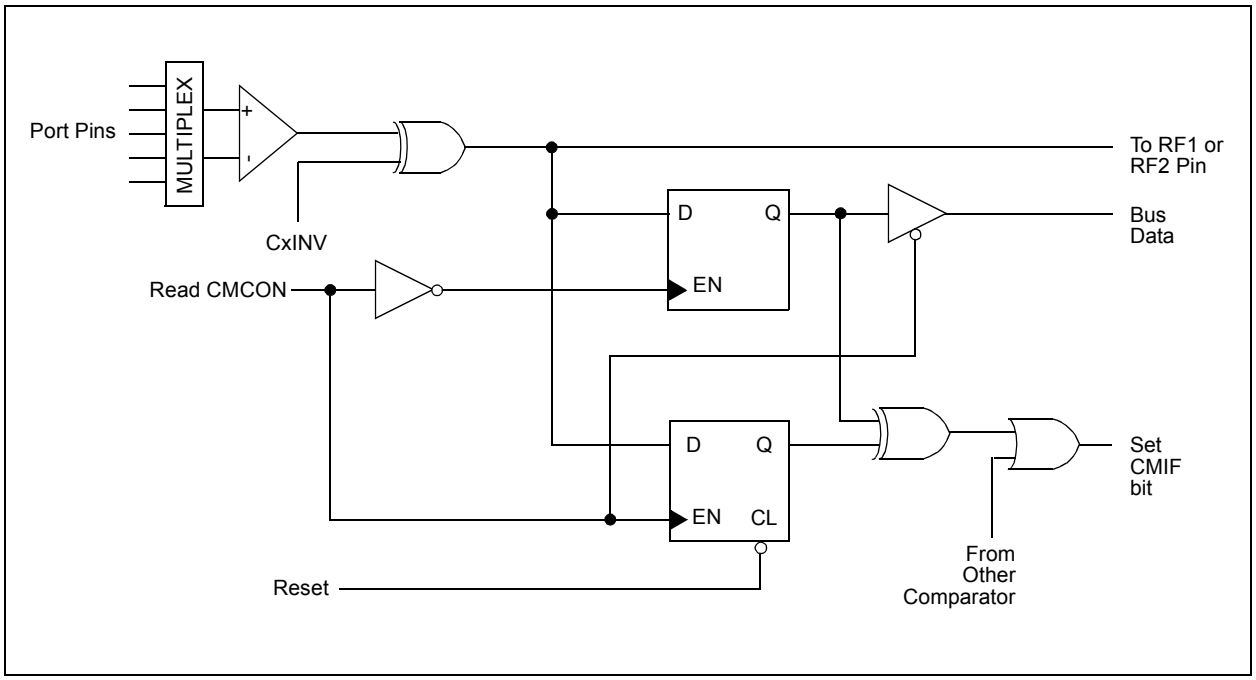


**FIGURE 21-3: COMPARATOR OUTPUT BLOCK DIAGRAM**




## 21.6 Comparator Interrupts

The comparator interrupt flag is set whenever there is a change in the output value of either comparator. Software will need to maintain information about the status of the output bits, as read from CMCON<7:6>, to determine the actual change that occurred. The CMIF bit (PIR2<6>) is the Comparator Interrupt Flag. The CMIF bit must be reset by clearing it. Since it is also possible to write a '1' to this register, a simulated interrupt may be initiated.

Both the CMIE bit (PIE2<6>) and the PEIE bit (INTCON<6>) must be set to enable the interrupt. In addition, the GIE bit (INTCON<7>) must also be set. If any of these bits are clear, the interrupt is not enabled, though the CMIF bit will still be set if an interrupt condition occurs.

| **Note:** | If a change in the CMCON register (C1OUT or C2OUT) should occur when a read operation is being executed (start of the Q2 cycle), then the CMIF (PIR2<6>) interrupt flag may not get set. |
|---|---|

The user, in the Interrupt Service Routine, can clear the interrupt in the following manner:

a) Any read or write of CMCON will end the mismatch condition.

b) Clear flag bit, CMIF.

A mismatch condition will continue to set flag bit CMIF. Reading CMCON will end the mismatch condition and allow flag bit CMIF to be cleared.

## 21.7 Comparator Operation During Sleep

When a comparator is active and the device is placed in Sleep mode, the comparator remains active and the interrupt is functional, if enabled. This interrupt will wake-up the device from Sleep mode, when enabled. Each operational comparator will consume additional current, as shown in the comparator specifications. To minimize power consumption while in Sleep mode, turn off the comparators (CM<2:0> = 111) before entering Sleep. If the device wakes up from Sleep, the contents of the CMCON register are not affected.

## 21.8 Effects of a Reset

A device Reset forces the CMCON register to its Reset state, causing the comparator modules to be turned off (CM<2:0> = 111). However, the input pins (RF3 through RF6) are configured as analog inputs by default on device Reset. The I/O configuration for these pins is determined by the setting of the PCFG<3:0> bits (ADCON1<3:0>). Therefore, device current is minimized when analog inputs are present at Reset time.

**REGISTER 23-8: WDTCON: WATCHDOG TIMER CONTROL REGISTER**

| R/W-0 | U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| REGSLP(1) | — | — | — | — | — | — | SWDTEN(2) |
| bit 7 | | | | | | | bit 0 |

| 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 |

bit 7 **REGSLP:** Voltage Regulator Low-Power Operation Enable bit(1)
1 = On-chip regulator enters low-power operation when device enters Sleep mode
0 = On-chip regulator continues to operate normally in Sleep mode

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

bit 0 **SWDTEN:** Software Controlled Watchdog Timer Enable bit(2)
1 = Watchdog Timer is on
0 = Watchdog Timer is off

**Note 1:** The REGSLP bit is automatically cleared when a Low-Voltage Detect condition occurs.
**2:** This bit has no effect if the Configuration bit, WDTEN, is enabled.

**TABLE 23-3: SUMMARY OF WATCHDOG TIMER REGISTERS**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Reset Values on page |
|---|---|---|---|---|---|---|---|---|---|
| RCON | IPEN | — | $\overline{CM}$ | $\overline{RI}$ | $\overline{TO}$ | $\overline{PD}$ | $\overline{POR}$ | $\overline{BOR}$ | 58 |
| WDTCON | REGSLP | — | — | — | — | — | — | SWDTEN | 58 |

**Legend:** — = unimplemented, read as '0'. Shaded cells are not used by the Watchdog Timer.

**BTG** **Bit Toggle f**

Syntax: BTG f, b {,a}

Operands: $0 \le f \le 255$
$0 \le b < 7$
$a \in [0,1]$

Operation: $(\overline{f<b>}) \rightarrow f<b>$

Status Affected: None

Encoding:

| 0111 | bbba | ffff | ffff |
|---|---|---|---|

Description: Bit 'b' in data memory location 'f' is inverted.

If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank.

If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \le 95$ (5Fh). See **Section 24.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1

Cycles: 1

Q Cycle Activity:

| Q1 | Q2 | Q3 | Q4 |
|---|---|---|---|
| Decode | Read register 'f' | Process Data | Write register 'f' |

Example: `BTG PORTC, 4, 0`

Before Instruction:
PORTC = 0111 0101 [75h]
After Instruction:
PORTC = 0110 0101 [65h]

**BOV** **Branch if Overflow**

Syntax: BOV n

Operands: $-128 \le n \le 127$

Operation: if Overflow bit is '1',
$(PC) + 2 + 2n \rightarrow PC$

Status Affected: None

Encoding:

| 1110 | 0100 | nnnn | nnnn |
|---|---|---|---|

Description: If the Overflow bit is '1', then the program will branch.

The 2's complement number '2n' is added to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be PC + 2 + 2n. This instruction is then a two-cycle instruction.

Words: 1

Cycles: 1(2)

Q Cycle Activity:
If Jump:

| Q1 | Q2 | Q3 | Q4 |
|---|---|---|---|
| Decode | Read literal 'n' | Process Data | Write to PC |
| No operation | No operation | No operation | No operation |

If No Jump:

| Q1 | Q2 | Q3 | Q4 |
|---|---|---|---|
| Decode | Read literal 'n' | Process Data | No operation |

Example:
```
HERE    BOV  Jump
```
Before Instruction
PC = address (HERE)
After Instruction
If Overflow = 1;
PC = address (Jump)
If Overflow = 0;
PC = address (HERE + 2)

## 25.11 PICkit 2 Development Programmer/Debugger and PICkit 2 Debug Express

The PICkit™ 2 Development Programmer/Debugger is a low-cost development tool with an easy to use interface for programming and debugging Microchip's Flash families of microcontrollers. The full featured Windows® programming interface supports baseline (PIC10F, PIC12F5xx, PIC16F5xx), midrange (PIC12F6xx, PIC16F), PIC18F, PIC24, dsPIC30, dsPIC33, and PIC32 families of 8-bit, 16-bit, and 32-bit microcontrollers, and many Microchip Serial EEPROM products. With Microchip's powerful MPLAB Integrated Development Environment (IDE) the PICkit™ 2 enables in-circuit debugging on most PIC® microcontrollers. In-Circuit-Debugging runs, halts and single steps the program while the PIC microcontroller is embedded in the application. When halted at a breakpoint, the file registers can be examined and modified.

The PICkit 2 Debug Express include the PICkit 2, demo board and microcontroller, hookup cables and CDROM with user's guide, lessons, tutorial, compiler and MPLAB IDE software.

## 25.12 MPLAB PM3 Device Programmer

The MPLAB PM3 Device Programmer is a universal, CE compliant device programmer with programmable voltage verification at $V_{DDMIN}$ and $V_{DDMAX}$ for maximum reliability. It features a large LCD display (128 x 64) for menus and error messages and a modular, detachable socket assembly to support various package types. The ICSP™ cable assembly is included as a standard item. In Stand-Alone mode, the MPLAB PM3 Device Programmer can read, verify and program PIC devices without a PC connection. It can also set code protection in this mode. The MPLAB PM3 connects to the host PC via an RS-232 or USB cable. The MPLAB PM3 has high-speed communications and optimized algorithms for quick programming of large memory devices and incorporates an MMC card for file storage and data applications.

## 25.13 Demonstration/Development Boards, Evaluation Kits, and Starter Kits

A wide variety of demonstration, development and evaluation boards for various PIC MCUs and dsPIC DSCs allows quick application development on fully functional systems. Most boards include prototyping areas for adding custom circuitry and provide application firmware and source code for examination and modification.

The boards support a variety of features, including LEDs, temperature sensors, switches, speakers, RS-232 interfaces, LCD displays, potentiometers and additional EEPROM memory.

The demonstration and development boards can be used in teaching environments, for prototyping custom circuits and for learning about various microcontroller applications.

In addition to the PICDEM™ and dsPICDEM™ demonstration/development board series of circuits, Microchip has a line of evaluation kits and demonstration software for analog filter design, KEELOQ® security ICs, CAN, IrDA®, PowerSmart battery management, SEEVAL® evaluation system, Sigma-Delta ADC, flow rate sensing, plus many more.

Also available are starter kits that contain everything needed to experience the specified device. This usually includes a single application and debug capability, all on one board.

Check the Microchip web page (www.microchip.com) for the complete list of demonstration, development and evaluation kits.

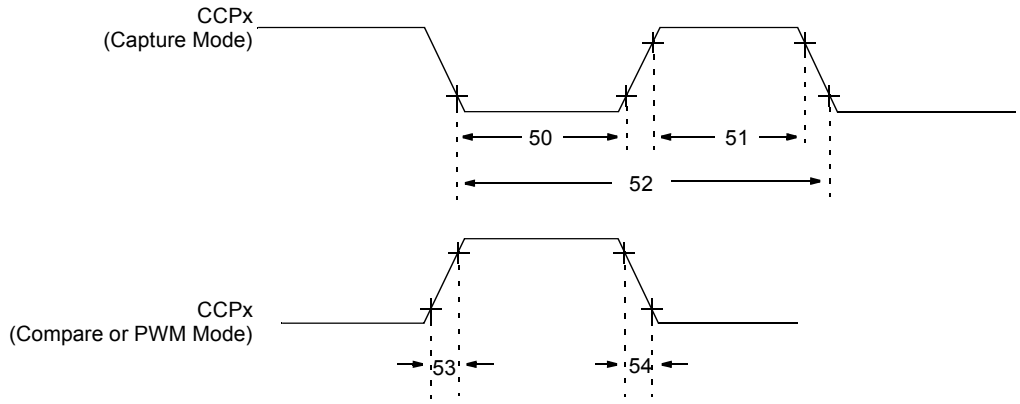
## 26.3 DC Characteristics:PIC18F84J90 Family (Industrial) (Continued)

| DC CHARACTERISTICS | | | Standard Operating Conditions (unless otherwise stated)<br>Operating temperature -40°C ≤ TA ≤ +85°C for industrial | | | |
|---|---|---|---|---|---|---|
| **Param No.** | **Symbol** | **Characteristic** | **Min** | **Max** | **Units** | **Conditions** |
| | VOL | **Output Low Voltage** | | | | |
| D080 | | I/O Ports: | | | | |
| | | PORTA, PORTF, PORTG, PORTH | — | 0.4 | V | IOL = 3.4 mA, VDD = 3.3V, -40°C to +85°C |
| | | PORTD, PORTE, PORTJ | — | 0.4 | V | IOL = 3.4 mA, VDD = 3.3V, -40°C to +85°C |
| | | PORTB, PORTC | — | 0.4 | V | IOL = 8.5 mA, VDD = 3.3V, -40°C to +85°C |
| D083 | | OSC2/CLKO (EC, ECPLL modes) | — | 0.4 | V | IOL = 1.6 mA, VDD = 3.3V, -40°C to +85°C |
| | VOH | **Output High Voltage(1)** | | | | |
| D090 | | I/O Ports: | | | V | |
| | | PORTA, PORTF, PORTG, PORTH | 2.4 | — | V | IOH = -2 mA, VDD = 3.3V, -40°C to +85°C |
| | | PORTD, PORTE, PORTJ | 2.4 | — | V | IOH = -2 mA, VDD = 3.3V, -40°C to +85°C |
| | | PORTB, PORTC | 2.4 | — | V | IOH = -6 mA, VDD = 3.3V, -40°C to +85°C |
| D092 | | OSC2/CLKO (INTOSC, EC, ECPLL modes) | 2.4 | — | V | IOH = -1 mA, VDD = 3.3V, -40°C to +85°C |
| | | **Capacitive Loading Specs on Output Pins** | | | | |
| D100 | COSC2 | OSC2 Pin | — | 15 | pF | In HS mode when external clock is used to drive OSC1 |
| D101 | CIO | All I/O Pins and OSC2 | — | 50 | pF | To meet the AC Timing Specifications |
| D102 | CB | SCL, SDA | — | 400 | pF | $I^2C$™ Specification |

**Note 1:** Negative current is defined as current sourced by the pin.

**2:** Refer to Table 10-1 for the pins that have corresponding tolerance limits.

**FIGURE 26-8: CAPTURE/COMPARE/PWM TIMINGS (CCP1, CCP2 MODULES)**




**Note:** Refer to Figure 26-3 for load conditions.

**TABLE 26-13: CAPTURE/COMPARE/PWM REQUIREMENTS (CCP1, CCP2 MODULES)**

| Param No. | Symbol | Characteristic | | Min | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|
| 50 | TccL | CCPx Input Low Time | No prescaler | 0.5 TCY + 20 | — | ns | |
| | | | With prescaler | 10 | — | ns | |
| 51 | TccH | CCPx Input High Time | No prescaler | 0.5 TCY + 20 | — | ns | |
| | | | With prescaler | 10 | — | ns | |
| 52 | TccP | CCPx Input Period | | $\frac{3\ T_{CY} + 40}{N}$ | — | ns | N = prescale value (1, 4 or 16) |
| 53 | TccR | CCPx Output Fall Time | | — | 25 | ns | |
| 54 | TccF | CCPx Output Fall Time | | — | 25 | ns | |

**FIGURE 26-11: EXAMPLE SPI SLAVE MODE TIMING (CKE = 0)**

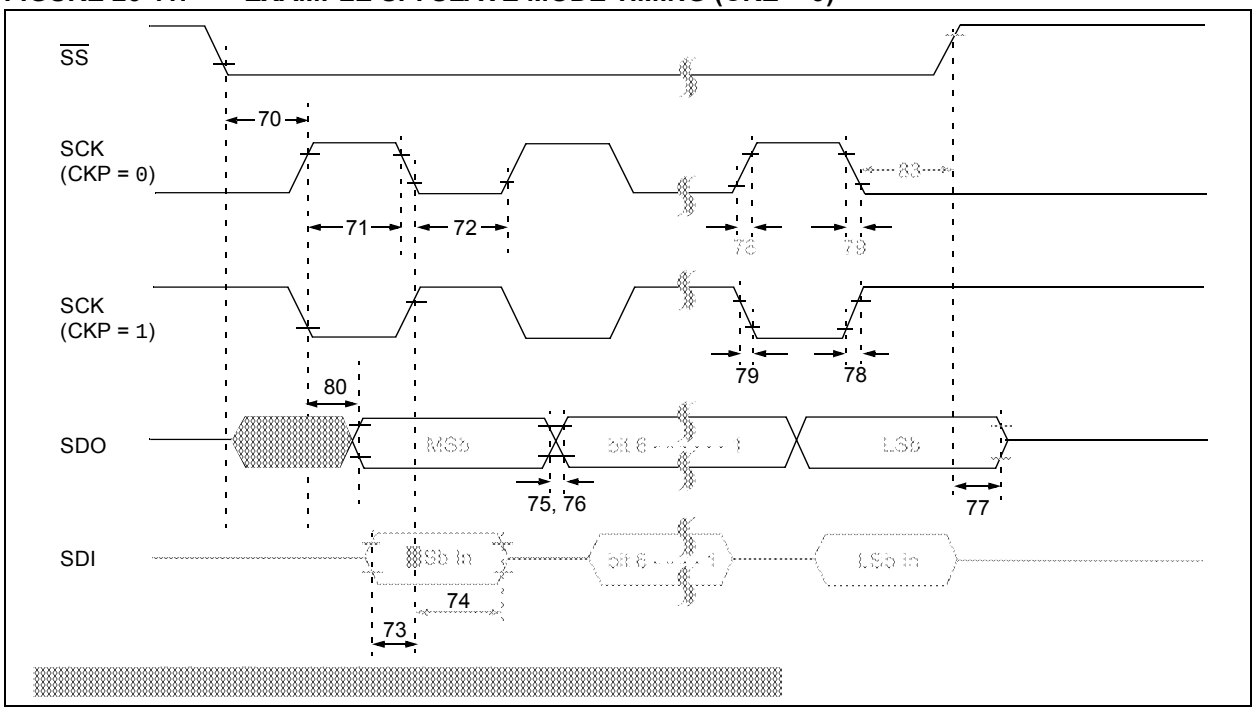



**TABLE 26-16: EXAMPLE SPI MODE REQUIREMENTS (SLAVE MODE TIMING, CKE = 0)**

| Param No. | Symbol | Characteristic | | Min | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|
| 70 | TssL2scH, TssL2scL | $\overline{SS}$ ↓ to SCK ↓ or SCK ↑ Input | | 3 Tcy | — | ns | |
| 70A | TssL2WB | $\overline{SS}$ to Write to SSPBUF | | 3 Tcy | — | ns | |
| 71 | TscH | SCK Input High Time (Slave mode) | Continuous | 1.25 Tcy + 30 | — | ns | |
| 71A | | | Single byte | 40 | — | ns | **(Note 1)** |
| 72 | TscL | SCK Input Low Time (Slave mode) | Continuous | 1.25 Tcy + 30 | — | ns | |
| 72A | | | Single byte | 40 | — | ns | **(Note 1)** |
| 73 | TdiV2scH, TdiV2scL | Setup Time of SDI Data Input to SCK Edge | | 20 | — | ns | |
| 73A | Tb2b | Last Clock Edge of Byte 1 to the First Clock Edge of Byte 2 | | 1.5 Tcy + 40 | — | ns | **(Note 2)** |
| 74 | TscH2diL, TscL2diL | Hold Time of SDI Data Input to SCK Edge | | 40 | — | ns | |
| 75 | TdoR | SDO Data Output Rise Time | | — | 25 | ns | |
| 76 | TdoF | SDO Data Output Fall Time | | — | 25 | ns | |
| 77 | TssH2doZ | $\overline{SS}$ ↑ to SDO Output High-Impedance | | 10 | 50 | ns | |
| 78 | TscR | SCK Output Rise Time (Master mode) | | — | 25 | ns | |
| 79 | TscF | SCK Output Fall Time (Master mode) | | — | 25 | ns | |
| 80 | TscH2doV, TscL2doV | SDO Data Output Valid after SCK Edge | | — | 50 | ns | |
| 83 | TscH2ssH, TscL2ssH | $\overline{SS}$ ↑ after SCK Edge | | 1.5 Tcy + 40 | — | ns | |

**Note 1:** Requires the use of Parameter #73A.

**2:** Only if Parameter #71A and #72A are used.