



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.

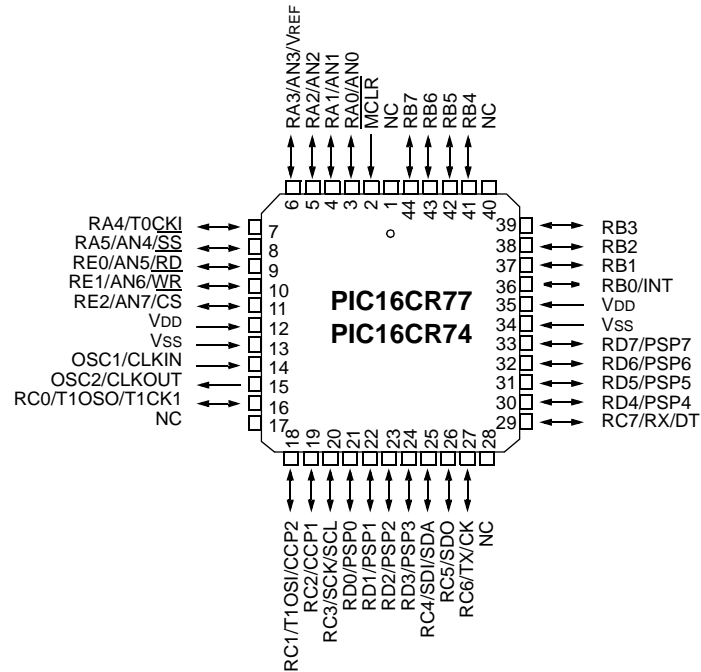
Applications of "[Embedded - Microcontrollers](#)"

Details

| | |
|----------------------------|---|
| Product Status | Obsolete |
| Core Processor | PIC |
| Core Size | 8-Bit |
| Speed | 20MHz |
| Connectivity | I ² C, SPI, UART/USART |
| Peripherals | Brown-out Detect/Reset, POR, PWM, WDT |
| Number of I/O | 33 |
| Program Memory Size | 7KB (4K x 14) |
| Program Memory Type | ROM |
| EEPROM Size | - |
| RAM Size | 192 x 8 |
| Voltage - Supply (Vcc/Vdd) | 2V ~ 5.5V |
| Data Converters | A/D 8x8b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 44-VQFN Exposed Pad |
| Supplier Device Package | 44-QFN (8x8) |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/pic16cr74-i-ml |

Pin Diagrams (Continued)

PLCC



TQFP

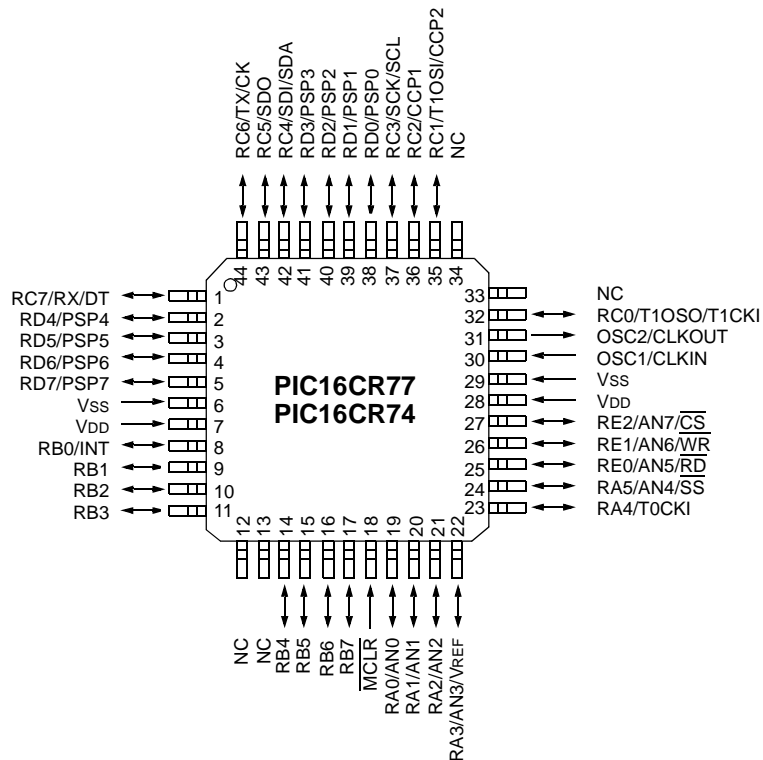
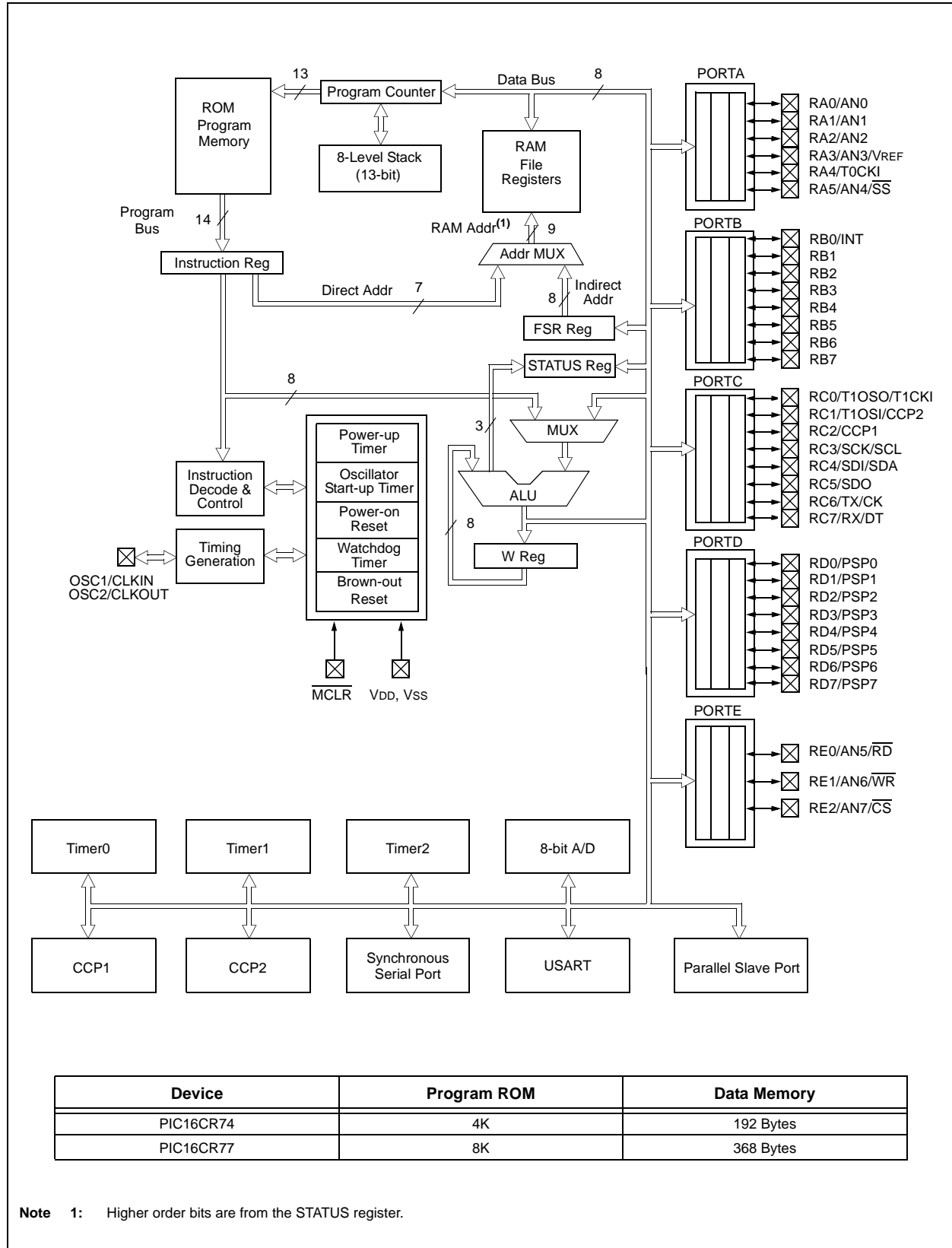


FIGURE 1-2: PIC16CR74 AND PIC16CR77 BLOCK DIAGRAM



PIC16CR7X

TABLE 1-3: PIC16CR74 AND PIC16CR77 PINOUT DESCRIPTION (CONTINUED)

| Pin Name | PDIP Pin# | PLCC Pin# | QFP Pin# | I/O/P Type | Buffer Type | Description |
|---|-----------|-----------------|------------------|-----------------|-----------------------|---|
| RD0/PSP0 RD0 PSP0 | 19 | 21 | 38 | I/O I/O | ST/TTL ⁽³⁾ | PORTD is a bidirectional I/O port or parallel slave port when interfacing to a microprocessor bus. Digital I/O. Parallel Slave Port data. |
| RD1/PSP1 RD1 PSP1 | 20 | 22 | 39 | I I/O I/O | ST/TTL ⁽³⁾ | Digital I/O. Parallel Slave Port data. |
| RD2/PSP2 RD2 PSP2 | 21 | 23 | 40 | I I/O I/O | ST/TTL ⁽³⁾ | Digital I/O. Parallel Slave Port data. |
| RD3/PSP3 RD3 PSP3 | 22 | 24 | 41 | I/O I/O | ST/TTL ⁽³⁾ | Digital I/O. Parallel Slave Port data. |
| RD4/PSP4 RD4 PSP4 | 27 | 30 | 2 | I/O I/O | ST/TTL ⁽³⁾ | Digital I/O. Parallel Slave Port data. |
| RD5/PSP5 RD5 PSP5 | 28 | 31 | 3 | I/O I/O | ST/TTL ⁽³⁾ | Digital I/O. Parallel Slave Port data. |
| RD6/PSP6 RD6 PSP6 | 29 | 32 | 4 | I/O I/O | ST/TTL ⁽³⁾ | Digital I/O. Parallel Slave Port data. |
| RD7/PSP7 RD7 PSP7 | 30 | 33 | 5 | I/O I/O | ST/TTL ⁽³⁾ | Digital I/O. Parallel Slave Port data. |
| RE0/AN5/ $\overline{\text{RD}}$ / RE0 AN5 $\overline{\text{RD}}$ | 8 | 9 | 25 | I/O I I | ST/TTL ⁽³⁾ | PORTE is a bidirectional I/O port. Digital I/O. Analog input 5. Read control for parallel slave port . |
| RE1/AN6/ $\overline{\text{WR}}$ / RE1 AN6 $\overline{\text{WR}}$ | 9 | 10 | 26 | I/O I I | ST/TTL ⁽³⁾ | Digital I/O. Analog input 6. Write control for parallel slave port . |
| RE2/AN7/ $\overline{\text{CS}}$ / RE2 AN7 $\overline{\text{CS}}$ | 10 | 11 | 27 | I/O I I | ST/TTL ⁽³⁾ | Digital I/O. Analog input 7. Chip Select control for parallel slave port . |
| VSS | 12,31 | 13,34 | 6,29 | P | — | Ground reference for logic and I/O pins. |
| VDD | 11,32 | 12,35 | 7,28 | P | — | Positive supply for logic and I/O pins. |
| NC | — | 1,17, 28, 40 | 12,13, 33, 34 | | — | These pins are not internally connected. These pins should be left unconnected. |

Legend: I = input O = output I/O = input/output P = power
— = Not used TTL = TTL input ST = Schmitt Trigger input

- Note** 1: This buffer is a Schmitt Trigger input when configured as an external interrupt.
2: This buffer is a Schmitt Trigger input when used in Serial Verify mode.
3: This buffer is a Schmitt Trigger input when configured as general purpose I/O and a TTL input when used in the Parallel Slave Port mode (for interfacing to a microprocessor bus).
4: This buffer is a Schmitt Trigger input when configured in RC Oscillator mode and a CMOS input otherwise.

TABLE 2-1: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR, BOR | Details on page | |
|----------------------|------------|--|--------|-------------------------------|--|------------------------|---------------------------|-------------------------|------------------|-------------------|-----------------|--------|
| Bank 1 | | | | | | | | | | | | |
| 80h ⁽⁴⁾ | INDF | Addressing this location uses contents of FSR to address data memory (not a physical register) | | | | | | | | 0000 0000 | 27, 96 | |
| 81h | OPTION_REG | $\overline{\text{RBPV}}$ | INTEDG | T0CS | T0SE | PSA | PS2 | PS1 | PS0 | 1111 1111 | 20, 44, 96 | |
| 82h ⁽⁴⁾ | PCL | Program Counter (PC) Least Significant Byte | | | | | | | | 0000 0000 | 26, 96 | |
| 83h ⁽⁴⁾ | STATUS | IRP | RP1 | RP0 | $\overline{\text{TO}}$ | $\overline{\text{PD}}$ | Z | DC | C ⁽²⁾ | 0001 1xxx | 19, 96 | |
| 84h ⁽⁴⁾ | FSR | Indirect data memory address pointer | | | | | | | | xxxx xxxx | 27, 96 | |
| 85h | TRISA | — | — | PORTA Data Direction Register | | | | | | | --11 1111 | 32, 96 |
| 86h | TRISB | PORTB Data Direction Register | | | | | | | | 1111 1111 | 34, 96 | |
| 87h | TRISC | PORTC Data Direction Register | | | | | | | | 1111 1111 | 35, 96 | |
| 88h ⁽⁵⁾ | TRISD | PORTD Data Direction Register | | | | | | | | 1111 1111 | 36, 96 | |
| 89h ⁽⁵⁾ | TRISE | IBF | OBF | IBOV | PSPMODE | — | PORTE Data Direction Bits | | | 0000 -111 | 38, 96 | |
| 8Ah ^(1,4) | PCLATH | — | — | — | Write Buffer for the upper 5 bits of the Program Counter | | | | | ---0 0000 | 26, 96 | |
| 8Bh ⁽⁴⁾ | INTCON | GIE | PEIE | TMR0IE | INTE | RBIE | TMR0IF | INTF | RBIF | 0000 000x | 21, 96 | |
| 8Ch | PIE1 | PSPIE ⁽³⁾ | ADIE | RCIE | TXIE | SSPIE | CCP1IE | TMR2IE | TMR1IE | 0000 0000 | 22, 97 | |
| 8Dh | PIE2 | — | — | — | — | — | — | — | CCP2IE | ---- ---0 | 24, 97 | |
| 8Eh | PCON | — | — | — | — | — | — | $\overline{\text{POR}}$ | BOR | ---- --qg | 22, 97 | |
| 8Fh | — | Unimplemented | | | | | | | | — | — | |
| 90h | — | Unimplemented | | | | | | | | — | — | |
| 91h | — | Unimplemented | | | | | | | | — | — | |
| 92h | PR2 | Timer2 Module Period Register | | | | | | | | 1111 1111 | 52, 97 | |
| 93h | SSPADD | Synchronous Serial Port (I ² C™ mode) Address Register | | | | | | | | 0000 0000 | 68, 97 | |
| 94h | SSPSTAT | SMP | CKE | D/ $\overline{\text{A}}$ | P | S | R/ $\overline{\text{W}}$ | UA | BF | 0000 0000 | 60, 97 | |
| 95h | — | Unimplemented | | | | | | | | — | — | |
| 96h | — | Unimplemented | | | | | | | | — | — | |
| 97h | — | Unimplemented | | | | | | | | — | — | |
| 98h | TXSTA | CSRC | TX9 | TXEN | SYNC | — | BRGH | TRMT | TX9D | 0000 -010 | 69, 97 | |
| 99h | SPBRG | Baud Rate Generator Register | | | | | | | | 0000 0000 | 71, 97 | |
| 9Ah | — | Unimplemented | | | | | | | | — | — | |
| 9Bh | — | Unimplemented | | | | | | | | — | — | |
| 9Ch | — | Unimplemented | | | | | | | | — | — | |
| 9Dh | — | Unimplemented | | | | | | | | — | — | |
| 9Eh | — | Unimplemented | | | | | | | | — | — | |
| 9Fh | ADCON1 | — | — | — | — | — | PCFG2 | PCFG1 | PCFG0 | ---- -000 | 84, 97 | |

Legend: x = unknown, u = unchanged, q = value depends on condition, — = unimplemented, read as '0', r = reserved.
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 during branches (CALL or GOTO).
- 2:** Other (non Power-up) Resets include external Reset through MCLR and Watchdog Timer Reset.
- 3:** Bits PSPIE and PSPIF are reserved on the 28-pin devices; always maintain these bits clear.
- 4:** These registers can be addressed from any bank.
- 5:** PORTD, PORTE, TRISD and TRISE are not physically implemented on the 28-pin devices, read as '0'.
- 6:** This bit always reads as a '1'.

TABLE 4-9: PORTE FUNCTIONS

| Name | Bit# | Buffer Type | Function |
|----------------------------------|-------|-----------------------|---|
| RE0/ $\overline{\text{RD}}$ /AN5 | bit 0 | ST/TTL ⁽¹⁾ | Input/output port pin or read control input in Parallel Slave Port mode or analog input. For $\overline{\text{RD}}$ (PSP mode): 1 = Idle 0 = Read operation. Contents of PORTD register output to PORTD I/O pins (if chip selected). |
| RE1/ $\overline{\text{WR}}$ /AN6 | bit 1 | ST/TTL ⁽¹⁾ | Input/output port pin or write control input in Parallel Slave Port mode or analog input. For $\overline{\text{WR}}$ (PSP mode): 1 = Idle 0 = Write operation. Value of PORTD I/O pins latched into PORTD register (if chip selected). |
| RE2/ $\overline{\text{CS}}$ /AN7 | bit 2 | ST/TTL ⁽¹⁾ | Input/output port pin or Chip Select control input in Parallel Slave Port mode or analog input. For $\overline{\text{CS}}$ (PSP mode): 1 = Device is not selected 0 = Device is selected |

Legend: ST = Schmitt Trigger input, TTL = TTL input

Note 1: Input buffers are Schmitt Triggers when in I/O mode and TTL buffers when in Parallel Slave Port mode.

TABLE 4-10: SUMMARY OF REGISTERS ASSOCIATED WITH PORTE

| Addr | 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 |
|------|--------|-------|-------|-------|---------|-------|---------------------------|-------|-------|-------------------|---------------------------|
| 09h | PORTE | — | — | — | — | — | RE2 | RE1 | RE0 | ---- -xxx | ---- -uuu |
| 89h | TRISE | IBF | OBF | IBOV | PSPMODE | — | PORTE Data Direction bits | | | 0000 -111 | 0000 -111 |
| 9Fh | ADCON1 | — | — | — | — | — | PCFG2 | PCFG1 | PCFG0 | ---- -000 | ---- -000 |

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

PIC16CR7X

REGISTER 8-1: CCP1CON/CCP2CON: (ADDRESS 17h/1Dh)

| | | | | | | | |
|-------|-----|-------|-------|--------|--------|--------|--------|
| U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| — | — | CCPxX | CCPY | CCPxM3 | CCPxM2 | CCPxM1 | CCPxM0 |
| 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 7-6 **Unimplemented:** Read as '0'

bit 5-4 **CCPxX:CCPY:** PWM Least Significant bits

Capture mode:

Unused

Compare mode:

Unused

PWM mode:

These bits are the two LSBs of the PWM duty cycle. The eight MSBs are found in CCPRxL.

bit 3-0 **CCPxM3:CCPxM0:** CCPx Mode Select bits

0000 = Capture/Compare/PWM disabled (resets CCPx module)

0100 = Capture mode, every falling edge

0101 = Capture mode, every rising edge

0110 = Capture mode, every 4th rising edge

0111 = Capture mode, every 16th rising edge

1000 = Compare mode, set output on match (CCPxIF bit is set)

1001 = Compare mode, clear output on match (CCPxIF bit is set)

1010 = Compare mode, generate software interrupt on match (CCPxIF bit is set, CCPx pin is unaffected)

1011 = Compare mode, trigger special event (CCPxIF bit is set, CCPx pin is unaffected); CCP1 clears Timer1; CCP2 clears Timer1 and starts an A/D conversion (if A/D module is enabled)

11xx = PWM mode

PIC16CR7X

9.3.2 MASTER MODE

Master mode of operation is supported in firmware using interrupt generation on the detection of the Start and Stop conditions. The Stop (P) and Start (S) bits are cleared from a Reset or when the SSP module is disabled. The Stop (P) and Start (S) bits will toggle based on the Start and Stop conditions. Control of the I²C bus may be taken when the P bit is set, or the bus is Idle and both the S and P bits are clear.

In Master mode, the SCL and SDA lines are manipulated by clearing the corresponding TRISC<4:3> bit(s). The output level is always low, irrespective of the value(s) in PORTC<4:3>. So when transmitting data, a '1' data bit must have the TRISC<4> bit set (input) and a '0' data bit must have the TRISC<4> bit cleared (output). The same scenario is true for the SCL line with the TRISC<3> bit. Pull-up resistors must be provided externally to the SCL and SDA pins for proper operation of the I²C module.

The following events will cause SSP Interrupt Flag bit, SSPIF, to be set (SSP Interrupt will occur if enabled):

- Start condition
- Stop condition
- Data transfer byte transmitted/received

Master mode of operation can be done with either the Slave mode Idle (SSPM3:SSPM0 = 1011), or with the Slave active. When both Master and Slave modes are enabled, the software needs to differentiate the source(s) of the interrupt.

9.3.3 MULTI-MASTER MODE

In Multi-Master mode, the interrupt generation on the detection of the Start and Stop conditions, allows the determination of when the bus is free. The Stop (P) and Start (S) bits are cleared from a Reset or when the SSP module is disabled. The Stop (P) and Start (S) bits will toggle based on the Start and Stop conditions. Control of the I²C bus may be taken when bit P (SSPSTAT<4>) is set, or the bus is Idle and both the S and P bits clear. When the bus is busy, enabling the SSP Interrupt will generate the interrupt when the Stop condition occurs.

In Multi-Master operation, the SDA line must be monitored to see if the signal level is the expected output level. This check only needs to be done when a high level is output. If a high level is expected and a low level is present, the device needs to release the SDA and SCL lines (set TRISC<4:3>). There are two stages where this arbitration can be lost, these are:

- Address Transfer
- Data Transfer

When the slave logic is enabled, the slave continues to receive. If arbitration was lost during the address transfer stage, communication to the device may be in progress. If addressed, an ACK pulse will be generated. If arbitration was lost during the data transfer stage, the device will need to retransfer the data at a later time.

TABLE 9-3: REGISTERS ASSOCIATED WITH I²C™ OPERATION

| 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 | INTE | RBIE | TMR0IF | INTF | RBIF | 0000 000x | 0000 000u |
| 0Ch | PIR1 | PSPIF ⁽¹⁾ | ADIF | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | 0000 0000 | 0000 0000 |
| 8Ch | PIE1 | PSPIE ⁽¹⁾ | ADIE | RCIE | TXIE | SSPIE | CCP1IE | TMR2IE | TMR1IE | 0000 0000 | 0000 0000 |
| 13h | SSPBUF | Synchronous Serial Port Receive Buffer/Transmit Register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 93h | SSPAD | Synchronous Serial Port (I ² C™ mode) Address Register | | | | | | | | 0000 0000 | 0000 0000 |
| 14h | SSPCON | WCOL | SSPOV | SSPEN | CKP | SSPM3 | SSPM2 | SSPM1 | SSPM0 | 0000 0000 | 0000 0000 |
| 94h | SSPSTAT | SMP ⁽²⁾ | CKE ⁽²⁾ | D/A | P | S | R/W | UA | BF | 0000 0000 | 0000 0000 |
| 87h | TRISC | PORTC Data Direction Register | | | | | | | | 1111 1111 | 1111 1111 |

Legend: x = unknown, u = unchanged, – = unimplemented locations read as '0'. Shaded cells are not used by SSP module in I²C™ mode.

- Note 1:** PSPIF and PSPIE are reserved on the PIC16CR73/76; always maintain these bits clear.
Note 2: Maintain these bits clear in I²C mode.

PIC16CR7X

11.7 Use of the CCP Trigger

An A/D conversion can be started by the “special event trigger” of the CCP2 module. This requires that the CCP2M3:CCP2M0 bits (CCP2CON<3:0>) be programmed as 1011 and that the A/D module is enabled (ADON bit is set). When the trigger occurs, the GO/DONE bit will be set, starting the A/D conversion, and the Timer1 counter will be reset to zero. Timer1 is reset to automatically repeat the A/D acquisition period

with minimal software overhead (moving the ADRES to the desired location). The appropriate analog input channel must be selected and an appropriate acquisition time should pass before the “special event trigger” sets the GO/DONE bit (starts a conversion).

If the A/D module is not enabled (ADON is cleared), then the “special event trigger” will be ignored by the A/D module, but will still reset the Timer1 counter.

TABLE 11-2: SUMMARY OF A/D REGISTERS

| 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 | INTE | RBIE | TMR0IF | INTF | RBIF | 0000 000x | 0000 000u |
| 0Ch | PIR1 | PSPIF ⁽¹⁾ | ADIF | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | 0000 0000 | 0000 0000 |
| 0Dh | PIR2 | — | — | — | — | — | — | — | CCP2IF | ---- --0 | ---- --0 |
| 8Ch | PIE1 | PSPIE ⁽¹⁾ | ADIE | RCIE | TXIE | SSPIE | CCP1IE | TMR2IE | TMR1IE | 0000 0000 | 0000 0000 |
| 8Dh | PIE2 | — | — | — | — | — | — | — | CCP2IE | ---- --0 | ---- --0 |
| 1Eh | ADRES | A/D Result Register Byte | | | | | | | | xxxx xxxx | uuuu uuuu |
| 1Fh | ADCON0 | ADCS1 | ADCS0 | CHS2 | CHS1 | CHS0 | GO/DONE | — | ADON | 0000 00-0 | 0000 00-0 |
| 9Fh | ADCON1 | — | — | — | — | — | PCFG2 | PCFG1 | PCFG0 | ---- -000 | ---- -000 |
| 05h | PORTA | — | — | RA5 | RA4 | RA3 | RA2 | RA1 | RA0 | --0x 0000 | --0u 0000 |
| 85h | TRISA | — | — | PORTA Data Direction Register | | | | | | --11 1111 | --11 1111 |
| 09h | PORTE ⁽²⁾ | — | — | — | — | — | RE2 | RE1 | RE0 | ---- -xxx | ---- -uuu |
| 89h | TRISE ⁽²⁾ | IBF | OBF | IBOV | PSPMODE | — | PORTE Data Direction Bits | | | 0000 -111 | 0000 -111 |

Legend: x = unknown, u = unchanged, — = unimplemented, read as '0'. Shaded cells are not used for A/D conversion.

- Note** 1: Bits PSPIE and PSPIF are reserved on the PIC16CR73/76; always maintain these bits clear.
2: These registers are reserved on the PIC16CR73/76.

12.3 Reset

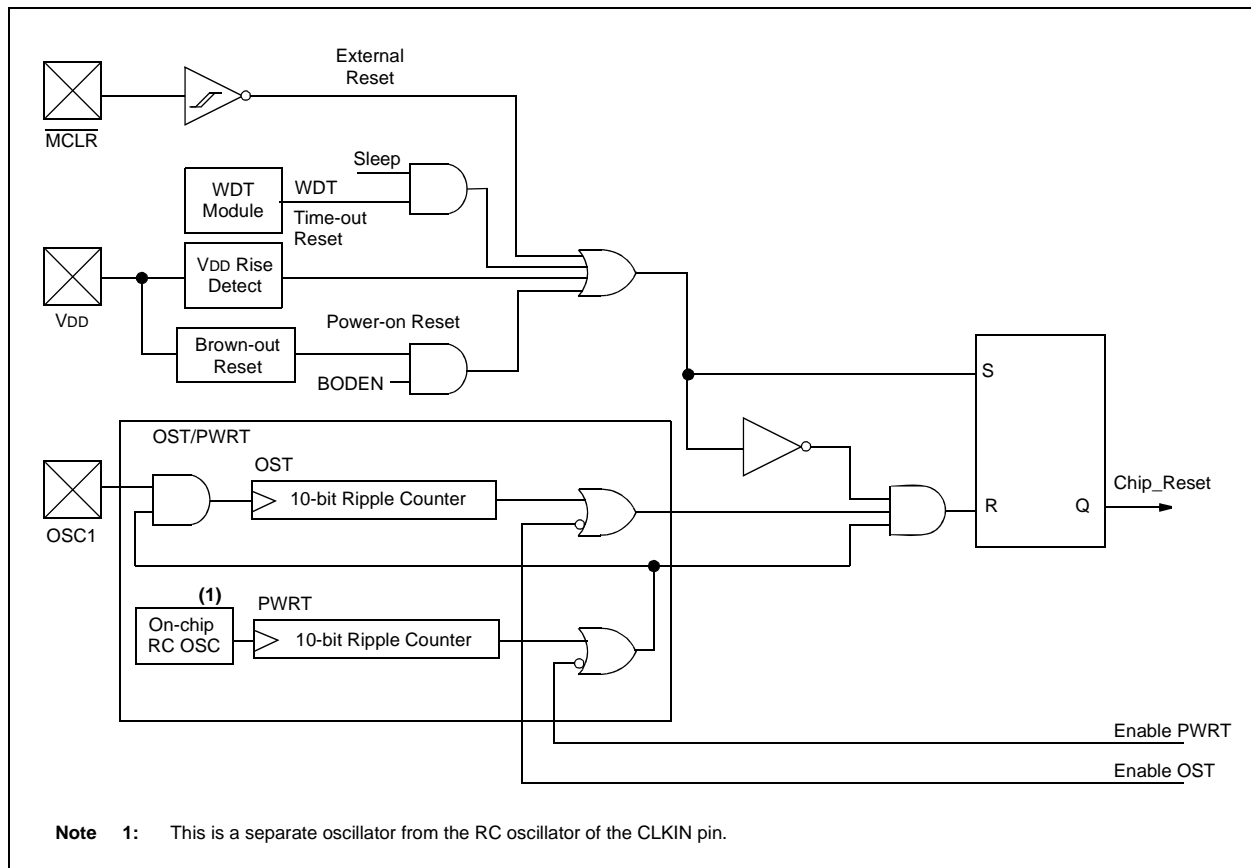
The PIC16CR7X differentiates between various kinds of Reset:

- Power-on Reset (POR)
- $\overline{\text{MCLR}}$ Reset during normal operation
- $\overline{\text{MCLR}}$ Reset during Sleep
- WDT Reset (during normal operation)
- WDT Wake-up (during Sleep)
- Brown-out Reset (BOR)

Some registers are not affected in any Reset condition. Their status is unknown on POR and unchanged in any other Reset. Most other registers are reset to a "Reset state" on Power-on Reset (POR), on the $\overline{\text{MCLR}}$ and WDT Reset, on $\overline{\text{MCLR}}$ Reset during Sleep, and Brown-out Reset (BOR). They are not affected by a WDT Wake-up, which is viewed as the resumption of normal operation. The $\overline{\text{TO}}$ and $\overline{\text{PD}}$ bits are set or cleared differently in different Reset situations, as indicated in Table 12-4. These bits are used in software to determine the nature of the Reset. See Table 12-6 for a full description of Reset states of all registers.

A simplified block diagram of the on-chip Reset circuit is shown in Figure 12-4.

FIGURE 12-4: SIMPLIFIED BLOCK DIAGRAM OF ON-CHIP RESET CIRCUIT



PIC16CR7X

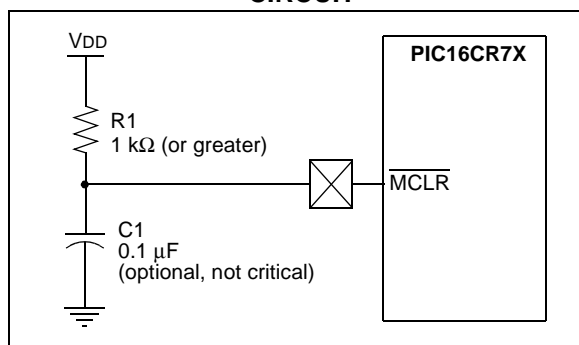
12.4 MCLR

PIC16CR7X devices have a noise filter in the $\overline{\text{MCLR}}$ Reset path. The filter will detect and ignore small pulses.

It should be noted that a WDT Reset does not drive $\overline{\text{MCLR}}$ pin low.

The behavior of the ESD protection on the $\overline{\text{MCLR}}$ pin has been altered from previous devices of this family. Voltages applied to the pin that exceed its specification can result in both $\overline{\text{MCLR}}$ Resets and excessive current beyond the device specification during the ESD event. For this reason, Microchip recommends that the $\overline{\text{MCLR}}$ pin no longer be tied directly to VDD. The use of an RC network, as shown in Figure 12-5, is suggested.

FIGURE 12-5: RECOMMENDED MCLR CIRCUIT



12.5 Power-on Reset (POR)

A Power-on Reset pulse is generated on-chip when VDD rise is detected (in the range of 1.2V-1.7V). To take advantage of the POR, tie the $\overline{\text{MCLR}}$ pin to VDD as described in **Section 12.4 "MCLR"**. A maximum rise time for VDD is specified. See the Electrical Specifications for details.

When the device starts normal operation (exits the Reset condition), device operating parameters (voltage, frequency, temperature,...) must be met to ensure operation. If these conditions are not met, the device must be held in Reset until the operating conditions are met. For additional information, refer to Application Note AN607, "Power-up Trouble Shooting" (DS00607).

12.6 Power-up Timer (PWRT)

The Power-up Timer provides a fixed 72 ms nominal time-out on power-up only from the POR. The Power-up Timer operates on an internal RC oscillator. The chip is kept in Reset as long as the PWRT is active. The PWRT's time delay allows VDD to rise to an acceptable level. A Configuration bit is provided to enable/disable the PWRT.

The power-up time delay will vary from chip-to-chip, due to VDD, temperature and process variation. See DC parameters for details (TPWRT, parameter #33).

12.7 Oscillator Start-up Timer (OST)

The Oscillator Start-up Timer (OST) provides 1024 oscillator cycles (from OSC1 input) delay after the PWRT delay is over (if enabled). This helps to ensure that the crystal oscillator or resonator has started and stabilized.

The OST time-out is invoked only for XT, LP and HS modes and only on Power-on Reset, or wake-up from Sleep.

12.8 Brown-out Reset (BOR)

The Configuration bit, BODEN, can enable or disable the Brown-out Reset circuit. If VDD falls below VBOR (parameter D005, about 4V) for longer than TBOR (parameter #35, about 100 μ S), the brown-out situation will reset the device. If VDD falls below VBOR for less than TBOR, a Reset may not occur.

Once the brown-out occurs, the device will remain in Brown-out Reset until VDD rises above VBOR. The Power-up Timer then keeps the device in Reset for TPWRT (parameter #33, about 72 mS). If VDD should fall below VBOR during TPWRT, the Brown-out Reset process will restart when VDD rises above VBOR, with the Power-up Timer Reset. The Power-up Timer is always enabled when the Brown-out Reset circuit is enabled, regardless of the state of the PWRT Configuration bit.

12.9 Time-out Sequence

On power-up, the time-out sequence is as follows: the PWRT delay starts (if enabled) when a POR Reset occurs. Then, OST starts counting 1024 oscillator cycles when PWRT ends (LP, XT, HS). When the OST ends, the device comes out of Reset.

If $\overline{\text{MCLR}}$ is kept low long enough, all delays will expire. Bringing $\overline{\text{MCLR}}$ high will begin execution immediately. This is useful for testing purposes or to synchronize more than one PIC16CR7X device operating in parallel.

Table 12-5 shows the Reset conditions for the STATUS, PCON and PC registers, while Table 12-6 shows the Reset conditions for all the registers.

12.14 Power-down Mode (Sleep)

Power-down mode is entered by executing a `SLEEP` instruction.

If enabled, the Watchdog Timer will be cleared but keeps running, the \overline{PD} bit (`STATUS<3>`) is cleared, the \overline{TO} (`STATUS<4>`) bit is set, and the oscillator driver is turned off. The I/O ports maintain the status they had before the `SLEEP` instruction was executed (driving high, low, or high-impedance).

For lowest current consumption in this mode, place all I/O pins at either V_{DD} or V_{SS} , ensure no external circuitry is drawing current from the I/O pin, power-down the A/D and disable external clocks. Pull all I/O pins that are high-impedance inputs, high or low externally, to avoid switching currents caused by floating inputs. The `T0CKI` input should also be at V_{DD} or V_{SS} for lowest current consumption. The contribution from on-chip pull-ups on `PORTB` should also be considered.

The \overline{MCLR} pin must be at a logic high level (V_{IHMC}).

12.14.1 WAKE-UP FROM SLEEP

The device can wake-up from Sleep through one of the following events:

1. External Reset input on \overline{MCLR} pin.
2. Watchdog Timer wake-up (if WDT was enabled).
3. Interrupt from INT pin, RB port change or a Peripheral Interrupt.

External \overline{MCLR} Reset will cause a device Reset. All other events are considered a continuation of program execution and cause a “wake-up”. The \overline{TO} and \overline{PD} bits in the `STATUS` register can be used to determine the cause of device Reset. The \overline{PD} bit, which is set on power-up, is cleared when Sleep is invoked. The \overline{TO} bit is cleared if a WDT time-out occurred and caused wake-up.

The following peripheral interrupts can wake the device from Sleep:

1. PSP read or write (PIC16CR74/77 only).
2. TMR1 interrupt. Timer1 must be operating as an asynchronous counter.
3. CCP Capture mode interrupt.
4. Special event trigger (Timer1 in Asynchronous mode, using an external clock).
5. SSP (Start/Stop) bit detect interrupt.
6. SSP transmit or receive in Slave mode (SPI/I²C).
7. USART RX or TX (Synchronous Slave mode).
8. A/D conversion (when A/D clock source is RC).

Other peripherals cannot generate interrupts, since during Sleep, no on-chip clocks are present.

When the `SLEEP` instruction is being executed, the next instruction (`PC + 1`) is pre-fetched. For the device to wake-up through an interrupt event, the corresponding interrupt enable bit must be set (enabled). Wake-up occurs, regardless of the state of the GIE bit. If the GIE bit is clear (disabled), the device continues execution at the instruction after the `SLEEP` instruction. If the GIE bit is set (enabled), the device executes the instruction after the `SLEEP` instruction and then branches to the interrupt address (0004h). In cases where the execution of the instruction following `SLEEP` is not desirable, the user should have a `NOP` after the `SLEEP` instruction.

12.14.2 WAKE-UP USING INTERRUPTS

When global interrupts are disabled (GIE cleared) and any interrupt source has both its interrupt enable bit and interrupt flag bit set, one of the following will occur:

- If the interrupt occurs **before** the execution of a `SLEEP` instruction, the `SLEEP` instruction will complete as a `NOP`. Therefore, the WDT and WDT postscaler will not be cleared, the \overline{TO} bit will not be set and \overline{PD} bits will not be cleared.
- If the interrupt occurs **during or after** the execution of a `SLEEP` instruction, the device will immediately wake-up from Sleep. The `SLEEP` instruction will be completely executed before the wake-up. Therefore, the WDT and WDT postscaler will be cleared, the \overline{TO} bit will be set and the \overline{PD} bit will be cleared.

Even if the flag bits were checked before executing a `SLEEP` instruction, it may be possible for flag bits to become set before the `SLEEP` instruction completes. To determine whether a `SLEEP` instruction executed, test the \overline{PD} bit. If the \overline{PD} bit is set, the `SLEEP` instruction was executed as a `NOP`.

To ensure that the WDT is cleared, a `CLRWDT` instruction should be executed before a `SLEEP` instruction.

PIC16CR7X

CALL Call Subroutine

Syntax: [*label*] CALL *k*
Operands: $0 \leq k \leq 2047$
Operation: $(PC) + 1 \rightarrow TOS$,
 $k \rightarrow PC<10:0>$,
 $(PCLATH<4:3>) \rightarrow PC<12:11>$
Status Affected: None
Description: Call Subroutine. First, return
 address $(PC + 1)$ is pushed onto
 the stack. The eleven-bit immedi-
 ate address is loaded into PC bits
 $<10:0>$. The upper bits of the PC
 are loaded from PCLATH. *CALL* is
 a two-cycle instruction.

CLRWDT Clear Watchdog Timer

Syntax: [*label*] CLRWDT
Operands: None
Operation: $00h \rightarrow WDT$
 $0 \rightarrow WDT \text{ prescaler}$,
 $1 \rightarrow \overline{TO}$
 $1 \rightarrow \overline{PD}$
Status Affected: \overline{TO} , \overline{PD}
Description: *CLRWDT* instruction resets the
 Watchdog Timer. It also resets the
 prescaler of the WDT. Status bits
 \overline{TO} and \overline{PD} are set.

CLRF Clear *f*

Syntax: [*label*] CLRF *f*
Operands: $0 \leq f \leq 127$
Operation: $00h \rightarrow (f)$
 $1 \rightarrow Z$
Status Affected: *Z*
Description: The contents of register '*f*' are
 cleared and the *Z* bit is set.

COMF Complement *f*

Syntax: [*label*] COMF *f*,*d*
Operands: $0 \leq f \leq 127$
 $d \in [0,1]$
Operation: $(\bar{f}) \rightarrow (\text{destination})$
Status Affected: *Z*
Description: The contents of register '*f*' are
 complemented. If '*d*' is '0', the
 result is stored in *W*. If '*d*' is '1',
 the result is stored back in
 register '*f*'.

CLRW Clear *W*

Syntax: [*label*] CLRW
Operands: None
Operation: $00h \rightarrow (W)$
 $1 \rightarrow Z$
Status Affected: *Z*
Description: *W* register is cleared. Zero bit (*Z*)
 is set.

DECF Decrement *f*

Syntax: [*label*] DECF *f*,*d*
Operands: $0 \leq f \leq 127$
 $d \in [0,1]$
Operation: $(f) - 1 \rightarrow (\text{destination})$
Status Affected: *Z*
Description: Decrement 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*'.

RLF Rotate Left f through Carry

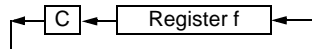
Syntax: [*label*] RLF *f*,*d*

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

Operation: See description below

Status Affected: C

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



SLEEP

Syntax: [*label*] SLEEP

Operands: None

Operation: 00h → WDT,
 0 → WDT prescaler,
 1 → \overline{TO} ,
 0 → \overline{PD}

Status Affected: \overline{TO} , \overline{PD}

Description: The power-down Status bit \overline{PD} is cleared. Time-out Status bit \overline{TO} is set. Watchdog Timer and its prescaler are cleared. The processor is put into Sleep mode with the oscillator stopped.

RETURN Return from Subroutine

Syntax: [*label*] RETURN

Operands: None

Operation: TOS → PC

Status Affected: None

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

SUBLW Subtract W from Literal

Syntax: [*label*] SUBLW *k*

Operands: $0 \leq k \leq 255$

Operation: $k - (W) \rightarrow (W)$

Status Affected: C, DC, Z

Description: The W register is subtracted (2's complement method) from the eight-bit literal 'k'. The result is placed in the W register.

RRF Rotate Right f through Carry

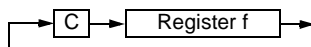
Syntax: [*label*] RRF *f*,*d*

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

Operation: See description below

Status Affected: C

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



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

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'.

PIC16CR7X

SWAPF **Swap Nibbles in f**

Syntax: [*label*] SWAPF f,d

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

Operation: (f<3:0>) → (destination<7:4>),
 (f<7:4>) → (destination<3:0>)

Status Affected: None

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

XORWF **Exclusive OR W with f**

Syntax: [*label*] XORWF f,d

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

Operation: (W) .XOR. (f) → (destination)

Status Affected: Z

Description: Exclusive OR the contents of the W register with 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'.

XORLW **Exclusive OR Literal with W**

Syntax: [*label*] XORLW k

Operands: $0 \leq k \leq 255$

Operation: (W) .XOR. k → (W)

Status Affected: Z

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

14.7 MPLAB ICE 2000 High-Performance In-Circuit Emulator

The MPLAB ICE 2000 In-Circuit Emulator is intended to provide the product development engineer with a complete microcontroller design tool set for PIC microcontrollers. Software control of the MPLAB ICE 2000 In-Circuit Emulator is advanced by the MPLAB Integrated Development Environment, which allows editing, building, downloading and source debugging from a single environment.

The MPLAB ICE 2000 is a full-featured emulator system with enhanced trace, trigger and data monitoring features. Interchangeable processor modules allow the system to be easily reconfigured for emulation of different processors. The architecture of the MPLAB ICE 2000 In-Circuit Emulator allows expansion to support new PIC microcontrollers.

The MPLAB ICE 2000 In-Circuit Emulator system has been designed as a real-time emulation system with advanced features that are typically found on more expensive development tools. The PC platform and Microsoft® Windows® 32-bit operating system were chosen to best make these features available in a simple, unified application.

14.8 MPLAB REAL ICE In-Circuit Emulator System

MPLAB REAL ICE In-Circuit Emulator System is Microchip's next generation high-speed emulator for Microchip Flash DSC® and MCU devices. It debugs and programs PIC® and dsPIC® Flash microcontrollers with the easy-to-use, powerful graphical user interface of the MPLAB Integrated Development Environment (IDE), included with each kit.

The MPLAB REAL ICE probe is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with either a connector compatible with the popular MPLAB ICD 2 system (RJ11) or with the new high speed, noise tolerant, low-voltage differential signal (LVDS) interconnection (CAT5).

MPLAB REAL ICE is field upgradeable through future firmware downloads in MPLAB IDE. In upcoming releases of MPLAB IDE, new devices will be supported, and new features will be added, such as software breakpoints and assembly code trace. MPLAB REAL ICE offers significant advantages over competitive emulators including low-cost, full-speed emulation, real-time variable watches, trace analysis, complex breakpoints, a ruggedized probe interface and long (up to three meters) interconnection cables.

14.9 MPLAB ICD 2 In-Circuit Debugger

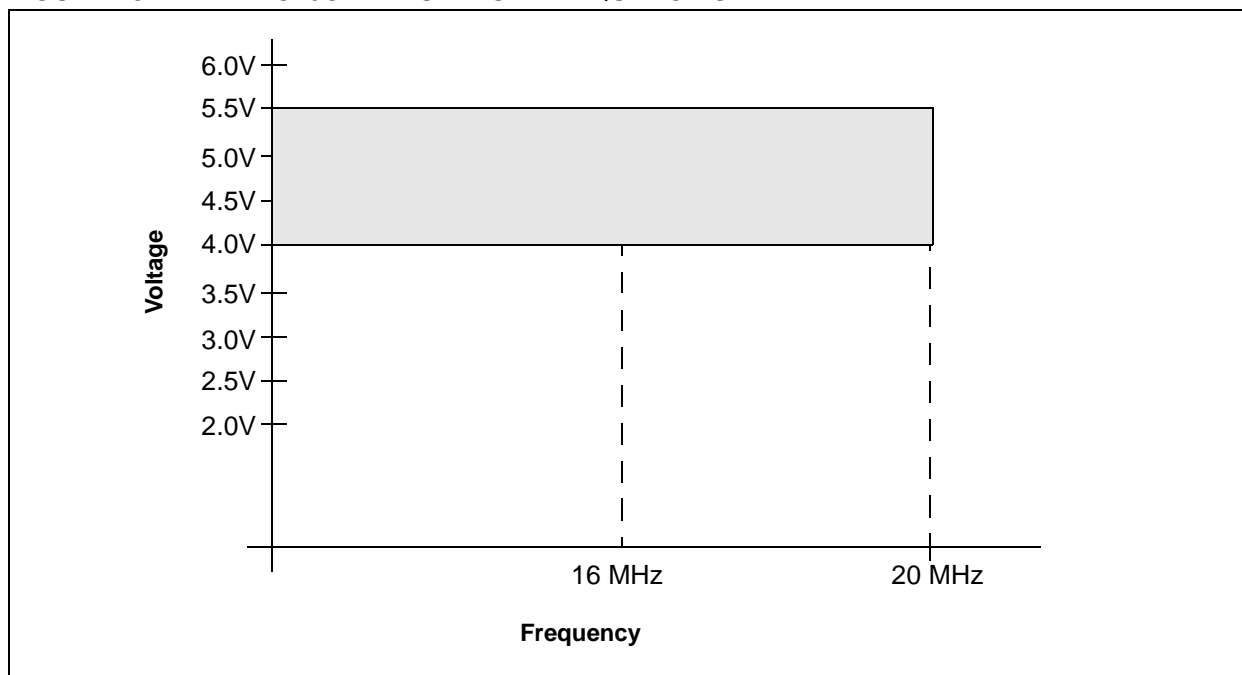
Microchip's In-Circuit Debugger, MPLAB ICD 2, is a powerful, low-cost, run-time development tool, connecting to the host PC via an RS-232 or high-speed USB interface. This tool is based on the Flash PIC MCUs and can be used to develop for these and other PIC MCUs and dsPIC DSCs. The MPLAB ICD 2 utilizes the in-circuit debugging capability built into the Flash devices. This feature, along with Microchip's In-Circuit Serial Programming™ (ICSP™) protocol, offers cost-effective, in-circuit Flash debugging from the graphical user interface of the MPLAB Integrated Development Environment. This enables a designer to develop and debug source code by setting breakpoints, single stepping and watching variables, and CPU status and peripheral registers. Running at full speed enables testing hardware and applications in real time. MPLAB ICD 2 also serves as a development programmer for selected PIC devices.

14.10 MPLAB PM3 Device Programmer

The MPLAB PM3 Device Programmer is a universal, CE compliant device programmer with programmable voltage verification at VDDMIN and VDDMAX 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 SD/MMC card for file storage and secure data applications.

PIC16CR7X

FIGURE 15-1: PIC16CR7X VOLTAGE-FREQUENCY GRAPH



15.3 Timing Parameter Symbolology

The timing parameter symbols have been created using one of the following formats:

1. TppS2ppS
2. TppS
3. TCC:ST (I²C™ specifications only)
4. Ts (I²C™ specifications only)

| | | | |
|----------|-----------|---|------|
| T | | | |
| F | Frequency | T | Time |

Lowercase letters (pp) and their meanings:

| | | | |
|-----------|-------------------|-----|------------------------------------|
| pp | | | |
| cc | CCP1 | osc | OSC1 |
| ck | CLKOUT | rd | \overline{RD} |
| cs | \overline{CS} | rw | \overline{RD} or \overline{WR} |
| di | SDI | sc | SCK |
| do | SDO | ss | \overline{SS} |
| dt | Data in | t0 | T0CKI |
| io | I/O port | t1 | T1CKI |
| mc | \overline{MCLR} | wr | \overline{WR} |

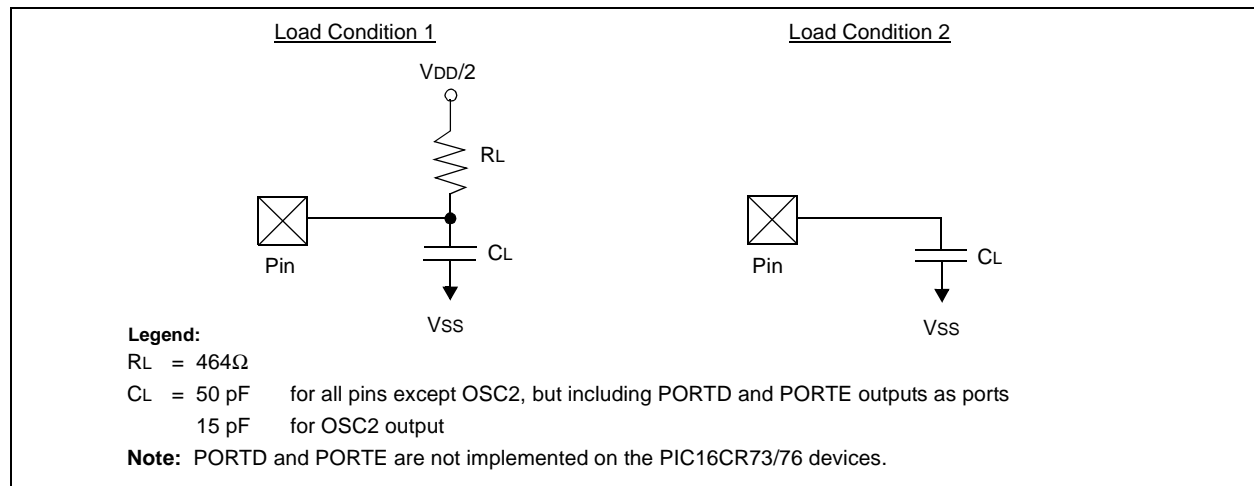
Uppercase letters and their meanings:

| | | | |
|-----------------------------|--------------------------|------|----------------|
| S | | | |
| F | Fall | P | Period |
| H | High | R | Rise |
| I | Invalid (High-impedance) | V | Valid |
| L | Low | Z | High-impedance |
| I²C™ only | | | |
| AA | output access | High | High |
| BUF | Bus free | Low | Low |

TCC:ST (I²C specifications only)

| | | | |
|-----------|-----------------|-----|----------------|
| CC | | | |
| HD | Hold | SU | Setup |
| ST | | | |
| DAT | DATA input hold | STO | Stop condition |
| STA | Start condition | | |

FIGURE 15-2: LOAD CONDITIONS



PIC16CR7X

TABLE 15-9: I²C™ BUS DATA REQUIREMENTS

| Param. No. | Symbol | Characteristic | | Min | Max | Units | Conditions |
|------------|---------|----------------------------|--------------|------------------------|------|-------|---|
| 100* | THIGH | Clock high time | 100 kHz mode | 4.0 | — | μs | Device must operate at a minimum of 1.5 MHz |
| | | | 400 kHz mode | 0.6 | — | μs | Device must operate at a minimum of 10 MHz |
| | | | SSP Module | 1.5T _{CY} | — | | |
| 101* | TLOW | Clock low time | 100 kHz mode | 4.7 | — | μs | Device must operate at a minimum of 1.5 MHz |
| | | | 400 kHz mode | 1.3 | — | μs | Device must operate at a minimum of 10 MHz |
| | | | SSP Module | 1.5T _{CY} | — | | |
| 102* | TR | SDA and SCL rise time | 100 kHz mode | — | 1000 | ns | |
| | | | 400 kHz mode | 20 + 0.1C _B | 300 | ns | C _B is specified to be from 10-400 pF |
| 103* | TF | SDA and SCL fall time | 100 kHz mode | — | 300 | ns | |
| | | | 400 kHz mode | 20 + 0.1C _B | 300 | ns | C _B is specified to be from 10-400 pF |
| 90* | TSU:STA | Start condition setup time | 100 kHz mode | 4.7 | — | μs | Only relevant for Repeated Start condition |
| | | | 400 kHz mode | 0.6 | — | μs | |
| 91* | THD:STA | Start condition hold time | 100 kHz mode | 4.0 | — | μs | After this period the first clock pulse is generated |
| | | | 400 kHz mode | 0.6 | — | μs | |
| 106* | THD:DAT | Data input hold time | 100 kHz mode | 0 | — | ns | |
| | | | 400 kHz mode | 0 | 0.9 | μs | |
| 107* | TSU:DAT | Data input setup time | 100 kHz mode | 250 | — | ns | (Note 2) |
| | | | 400 kHz mode | 100 | — | ns | |
| 92* | TSU:STO | Stop condition setup time | 100 kHz mode | 4.7 | — | μs | |
| | | | 400 kHz mode | 0.6 | — | μs | |
| 109* | TAA | Output valid from clock | 100 kHz mode | — | 3500 | ns | (Note 1) |
| | | | 400 kHz mode | — | — | ns | |
| 110* | TBUF | Bus free time | 100 kHz mode | 4.7 | — | μs | Time the bus must be free before a new transmission can start |
| | | | 400 kHz mode | 1.3 | — | μs | |
| | CB | Bus capacitive loading | | — | 400 | pF | |

* These parameters are characterized but not tested.

- Note 1:** As a transmitter, the device must provide this internal minimum delay time to bridge the undefined region (min. 300 ns) of the falling edge of SCL to avoid unintended generation of Start or Stop conditions.
- 2:** A Fast mode (400 kHz) I²C bus device can be used in a Standard mode (100 kHz) I²C bus system, but the requirement TSU:DAT ≥ 250 ns must then be met. This will automatically be the case if the device does not stretch the LOW period of the SCL signal. If such a device does stretch the LOW period of the SCL signal, it must output the next data bit to the SDA line T_R max. + TSU:DAT = 1000 + 250 = 1250 ns (according to the Standard mode I²C bus specification), before the SCL line is released.

PIC16CR7X

FIGURE 16-3: TYPICAL I_{DD} vs. F_{osc} OVER V_{DD} (XT MODE)

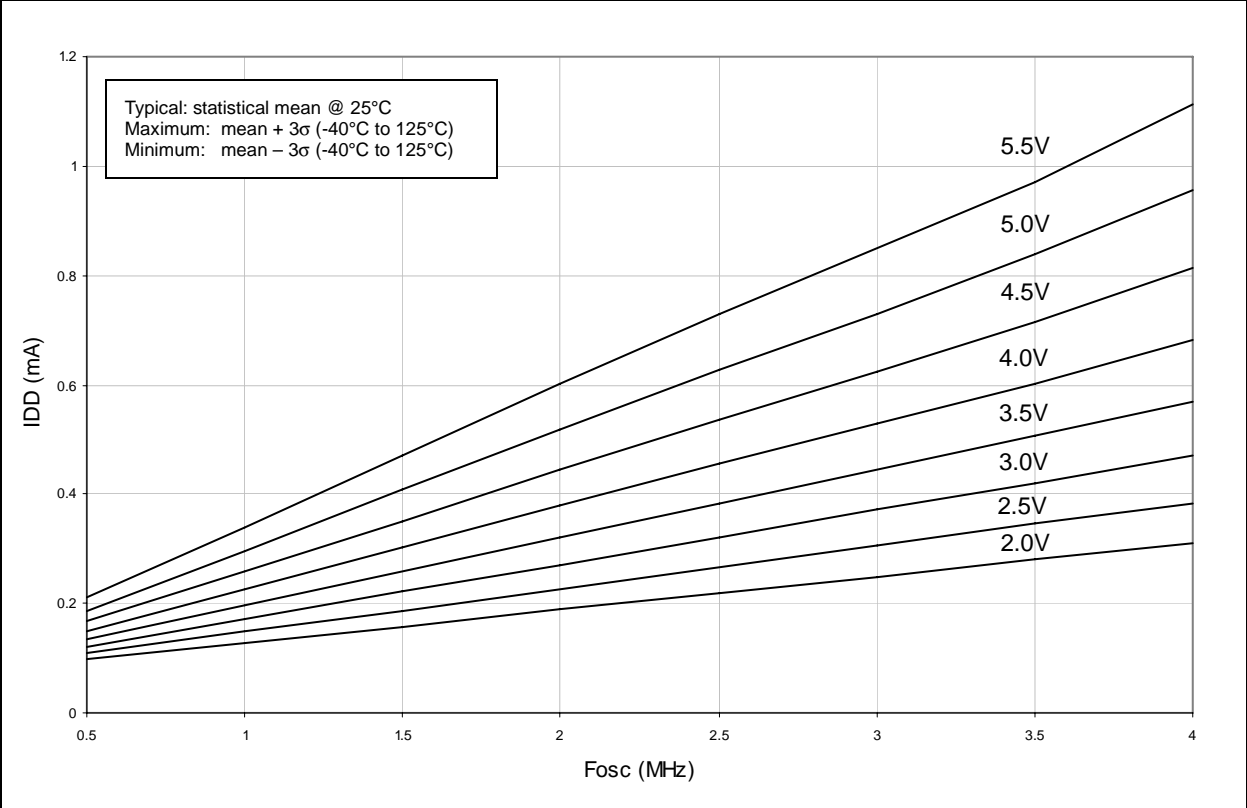
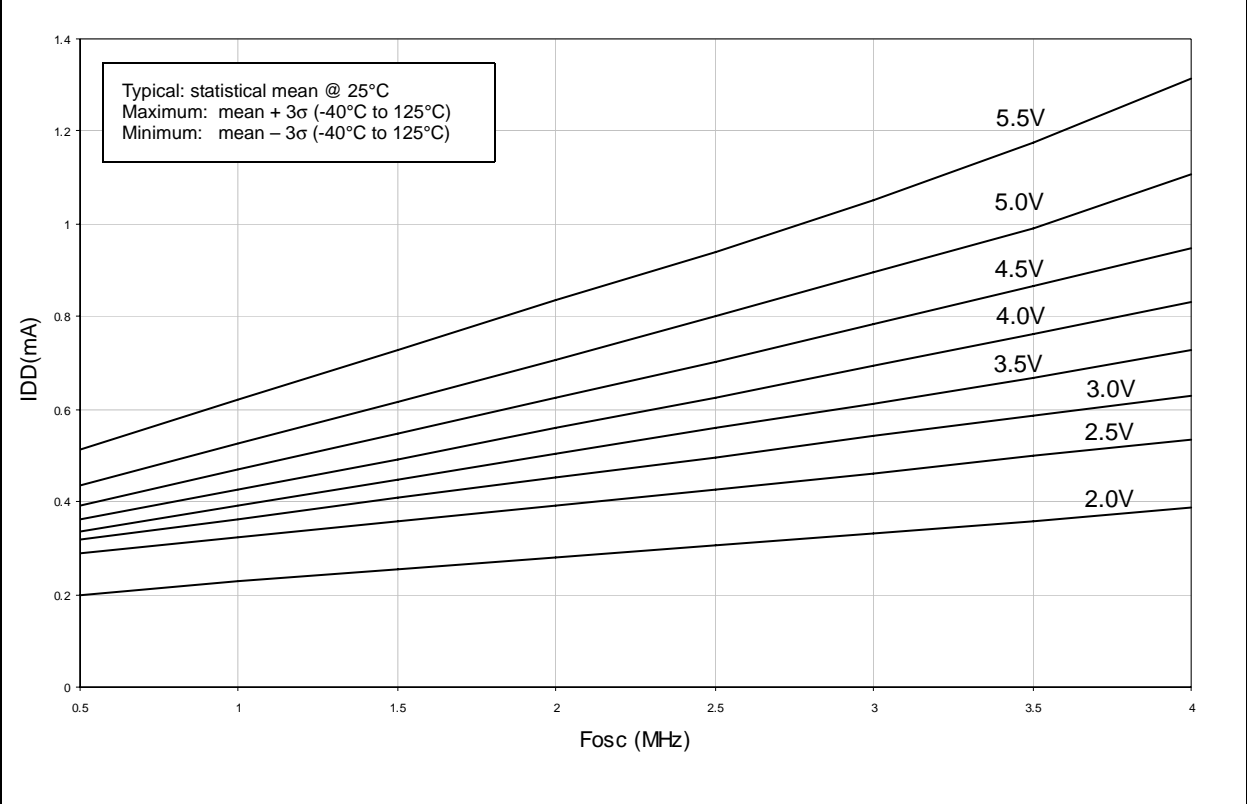


FIGURE 16-4: MAXIMUM I_{DD} vs. F_{osc} OVER V_{DD} (XT MODE)



THE MICROCHIP WEB SITE

Microchip provides online support via our WWW site at www.microchip.com. This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

CUSTOMER CHANGE NOTIFICATION SERVICE

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip web site at www.microchip.com, click on Customer Change Notification and follow the registration instructions.

CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support
- Development Systems Information Line

Customers should contact their distributor, representative or field application engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the web site at: <http://support.microchip.com>