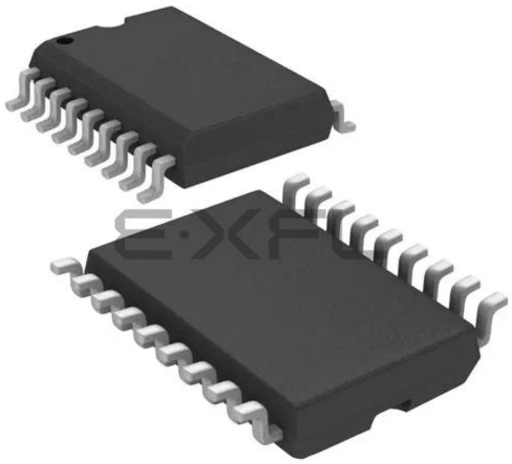



Welcome to E-XFL.COM

## 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 | - |
| Peripherals | Brown-out Detect/Reset, POR, PWM, WDT |
| Number of I/O | 13 |
| Program Memory Size | 3.5KB (2K x 14) |
| Program Memory Type | OTP |
| EEPROM Size | - |
| RAM Size | 128 x 8 |
| Voltage - Supply (Vcc/Vdd) | 4V ~ 5.5V |
| Data Converters | A/D 4x8b |
| Oscillator Type | External |
| Operating Temperature | -40°C ~ 125°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 18-SOIC (0.295", 7.50mm Width) |
| Supplier Device Package | 18-SOIC |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/pic16c716-20e-so |

### 2.2.2.3 INTCON Register

The INTCON Register is a readable and writable register which contains various enable and flag bits for the TMR0 register overflow, RB Port change and External RB0/INT pin interrupts.

**Note:** Interrupt flag bits get set when an interrupt condition occurs, regardless of the state of its corresponding enable bit or the global enable bit, GIE (INTCON<7>). User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

**FIGURE 2-6: INTCON REGISTER (ADDRESS 0Bh, 8Bh)**

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-x |
|---|---|---|---|---|---|---|---|
| GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | RBIF |
| bit7 | | | | | | | bit0 |

R = Readable bit
W = Writable bit
U = Unimplemented bit, read as '0'
-n = Value at POR Reset

bit 7: **GIE:** Global Interrupt Enable bit
1 = Enables all unmasked interrupts
0 = Disables all interrupts

bit 6: **PEIE**: Peripheral Interrupt Enable bit
1 = Enables all unmasked peripheral interrupts
0 = Disables all peripheral interrupts

bit 5: **T0IE**: TMR0 Overflow Interrupt Enable bit
1 = Enables the TMR0 interrupt
0 = Disables the TMR0 interrupt

bit 4: **IINTE**: RB0/INT External Interrupt Enable bit
1 = Enables the RB0/INT external interrupt
0 = Disables the RB0/INT external interrupt

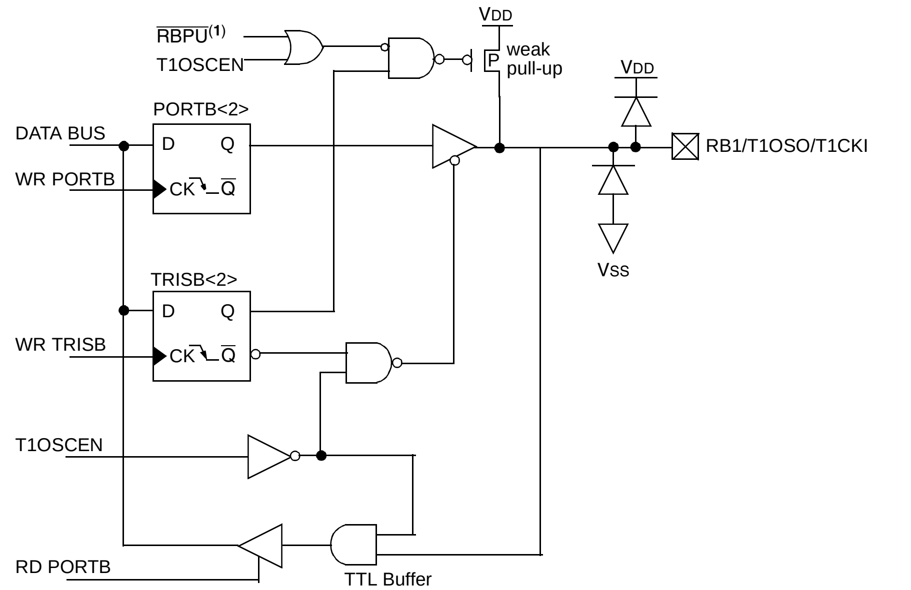
bit 3: **RBIE**: RB Port Change Interrupt Enable bit
1 = Enables the RB port change interrupt
0 = Disables the RB port change interrupt

bit 2: **T0IF**: TMR0 Overflow Interrupt Flag bit
1 = TMR0 register has overflowed (must be cleared in software)
0 = TMR0 register did not overflow

bit 1: **INTF**: RB0/INT External Interrupt Flag bit
1 = The RB0/INT external interrupt occurred (must be cleared in software)
0 = The RB0/INT external interrupt did not occur

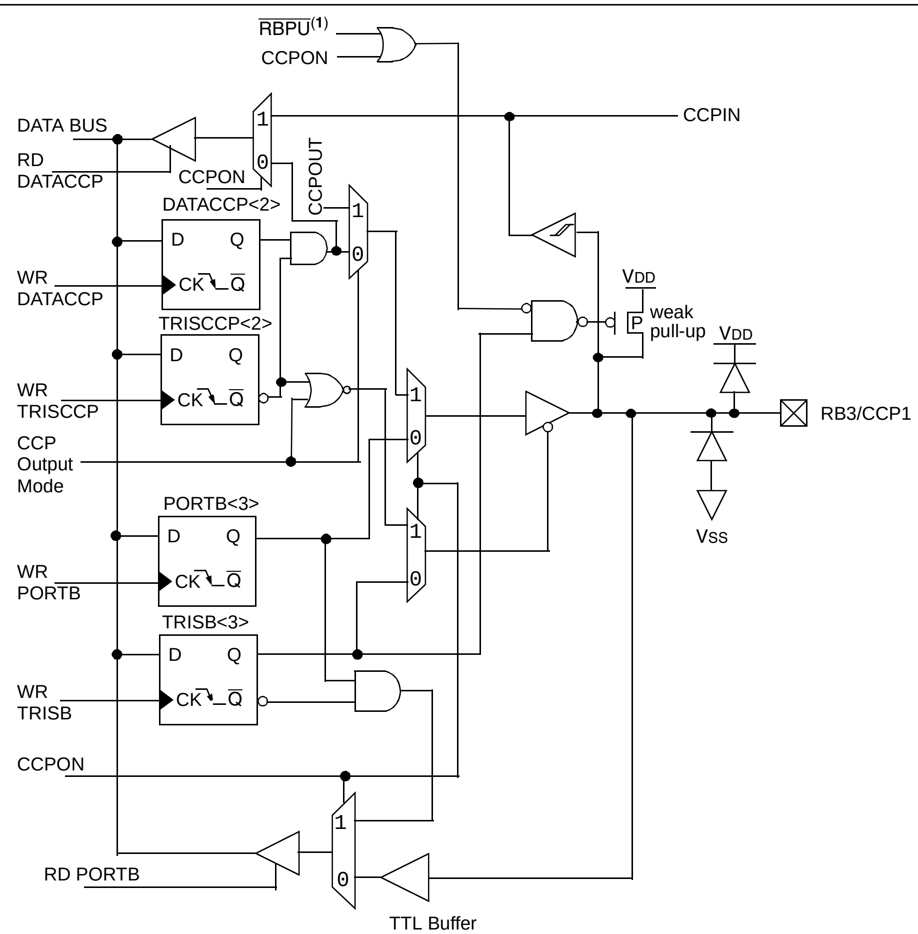
bit 0: **RBIF**: RB Port Change Interrupt Flag bit
1 = At least one of the RB7:RB4 pins changed state (must be cleared in software)
0 = None of the RB7:RB4 pins have changed state

**FIGURE 3-5: BLOCK DIAGRAM OF RB2/T1OSI PIN**




**Note 1:** To enable weak pull-ups, set the appropriate TRIS bit(s) and clear the RBPU bit (OPTION_REG<7>).

**FIGURE 3-6: BLOCK DIAGRAM OF RB3/CCP1 PIN**




**Note 1:** To enable weak pull-ups, set the appropriate TRIS bit(s) and clear the RBPU bit (OPTION_REG<7>).

## 5.0 TIMER1 MODULE

The Timer1 module timer/counter has the following features:

- 16-bit timer/counter
(Two 8-bit registers; TMR1H and TMR1L)
- Readable and writable (Both registers)
- Internal or external clock select
- Interrupt on overflow from FFFFh to 0000h
- Reset from CCP module trigger

Timer1 has a control register, shown in Figure 5-1. Timer1 can be enabled/disabled by setting/clearing control bit TMR1ON (T1CON<0>).

Figure 5-2 is a simplified block diagram of the Timer1 module.

Additional information on timer modules is available in the PIC® Mid-Range Reference Manual, (DS33023).

## 5.1 Timer1 Operation

Timer1 can operate in one of these modes:

- As a timer
- As a synchronous counter
- As an asynchronous counter

The operating mode is determined by the clock select bit, TMR1CS (T1CON<1>).

In timer mode, Timer1 increments every instruction cycle. In counter mode, it increments on every rising edge of the external clock input.

When the Timer1 oscillator is enabled (T1OSCEN is set), the RB2/T1OSI and RB1/T1OSO/T1CKI pins become inputs. That is, the TRISB<2:1> value is ignored.

Timer1 also has an internal "Reset input". This Reset can be generated by the CCP module (see **Section 7.0 "Capture/Compare/PWM (CCP) Module(s)"**).

**FIGURE 5-1: T1CON: TIMER1 CONTROL REGISTER (ADDRESS 10h)**

| U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | — | T1CKPS1 | T1CKPS0 | T1OSCEN | $\overline{\text{T1SYNC}}$ | TMR1CS | TMR1ON |
| bit7 | | | | | | | bit0 |

R = Readable bit
W = Writable bit
U = Unimplemented bit, read as '0'
-n = Value at POR Reset

bit 7-6: **Unimplemented:** Read as '0'

bit 5-4: **T1CKPS1:T1CKPS0**: Timer1 Input Clock Prescale Select bits
11 = 1:8 Prescale value
10 = 1:4 Prescale value
01 = 1:2 Prescale value
00 = 1:1 Prescale value

bit 3: **T1OSCEN**: Timer1 Oscillator Enable Control bit
1 = Oscillator is enabled
0 = Oscillator is shut off
Note: The oscillator inverter and feedback resistor are turned off to eliminate power drain

bit 2: **$\overline{\text{T1SYNC}}$**: Timer1 External Clock Input Synchronization Control bit

TMR1CS = 1
1 = Do not synchronize external clock input
0 = Synchronize external clock input

TMR1CS = 0
This bit is ignored. Timer1 uses the internal clock when TMR1CS = 0.

bit 1: **TMR1CS**: Timer1 Clock Source Select bit
1 = External clock from pin RB1/T1OSO/T1CKI (on the rising edge)
0 = Internal clock (FOSC/4)

bit 0: **TMR1ON**: Timer1 On bit
1 = Enables Timer1
0 = Stops Timer1

**TABLE 5-1: TMR1 MODULE AND PORTB OPERATION**

| TMR1 Module Mode | Clock Source | Control Bits | TMR1 Module Operation | PORTB<2:1> Operation |
|---|---|---|---|---|
| Off | N/A | T1CON = --xx 0x00 | Off | PORTB<2:1> function as normal I/O |
| Timer | FOSC/4 | T1CON = --xx 0x01 | TMR1 module uses the main oscillator as clock source. TMR1ON can turn on or turn off Timer1. | PORTB<2:1> function as normal I/O |
| Counter | External circuit | T1CON = --xx 0x11<br>TR1SCCP = ---- -x-1 | TMR1 module uses the external signal on the RB1/T1OSO/T1CKI pin as a clock source. TMR1ON can turn on or turn off Timer1. DT1CK can read the signal on the RB1/T1OSO/T1CKI pin. | PORTB<2> functions as normal I/O. PORTB<1> always reads '0' when configured as input. If PORTB<1> is configured as output, reading PORTB<1> will read the data latch. Writing to PORTB<1> will always store the result in the data latch, but not to the RB1/T1OSO/T1CKI pin. If the TMR1CS bit is cleared (TMR1 reverts to the timer mode), then pin PORTB<1> will be driven with the value in the data latch. |
| | Firmware | T1CON = --xx 0x11<br>TR1SCCP = ---- -x-0 | DATACCP<0> bit drives RB1/T1OSO/T1CKI and produces the TMR1 clock source. TMR1ON can turn on or turn off Timer1. The DATACCP<0> bit, DT1CK, can read and write to the RB1/T1OSO/T1CKI pin. | |
| | Timer1 oscillator | T1CON = --xx 1x11 | RB1/T1OSO/T1CKI and RB2/T1OSI are configured as a 2 pin crystal oscillator. RB1/T1OSI/T1CKI is the clock input for TMR1. TMR1ON can turn on or turn off Timer1. DATACCP<1> bit, DT1CK, always reads '0' as input and can not write to the RB1/T1OSO/T1CK1 pin. | PORTB<2:1> always read '0' when configured as inputs. If PORTB<2:1> are configured as outputs, reading PORTB<2:1> will read the data latches. Writing to PORTB<2:1> will always store the result in the data latches, but not to the RB2/T1OSI and RB1/T1OSO/T1CKI pins. If the TMR1CS and T1OSCEN bits are cleared (TMR1 reverts to the timer mode and TMR1 oscillator is disabled), then pin PORTB<2:1> will be driven with the value in the data latches. |

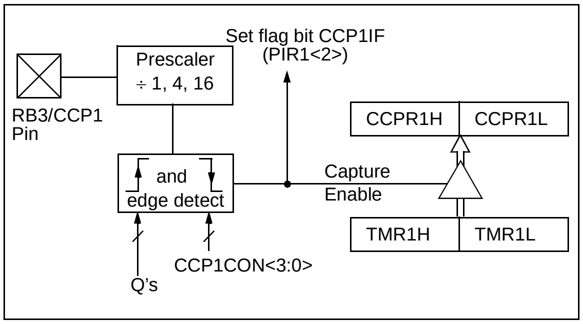
## 7.1 Capture Mode

In Capture mode, CCPR1H:CCPR1L captures the 16-bit value of the TMR1 register when an event occurs on pin RB3/CCP1. An event is defined as:

- every falling edge
- every rising edge
- every 4th rising edge
- every 16th rising edge

An event is selected by control bits CCP1M3:CCP1M0 (CCP1CON<3:0>). When a capture is made, the interrupt request flag bit CCP1IF (PIR1<2>) is set. It must be cleared in software. If another capture occurs before the value in register CCPR1 is read, the old captured value will be lost.

**FIGURE 7-3: CAPTURE MODE OPERATION BLOCK DIAGRAM**




### 7.1.1 CCP PIN CONFIGURATION

In Capture mode, the CCP output must be disabled by setting the TRISCCP<2> bit.

| | |
|---|---|
| **Note:** | If the RB3/CCP1 is configured as an output by clearing the TRISCCP<2> bit, a write to the DCCP bit can cause a capture condition. |

### 7.1.2 TIMER1 MODE SELECTION

Timer1 must be running in Timer mode or Synchronized Counter mode for the CCP module to use the capture feature. In Asynchronous Counter mode, the capture operation may not work.

### 7.1.3 SOFTWARE INTERRUPT

When the Capture mode is changed, a false capture interrupt may be generated. The user should keep bit CCP1IE (PIE1<2>) clear to avoid false interrupts and should clear the flag bit CCP1IF following any such change in Operating mode.

### 7.1.4 CCP PRESCALER

There are four prescaler settings, specified by bits CCP1M3:CCP1M0. Whenever the CCP module is turned off, or the CCP module is not in Capture mode, the prescaler counter is cleared. This means that any Reset will clear the prescaler counter.

Switching from one capture prescaler to another may generate an interrupt. Also, the prescaler counter will not be cleared, therefore the first capture may be from a non-zero prescaler. Example 7-1 shows the recommended method for switching between capture prescalers. This example also clears the prescaler counter and will not generate the “false” interrupt.

**EXAMPLE 7-1: CHANGING BETWEEN CAPTURE PRESCALERS**

```
CLRF   CCP1CON      ;Turn CCP module off
MOVLW  NEW_CAPT_PS  ;Load the W reg with
                    ; the new prescaler
                    ; mode value and CCP ON
MOVWF  CCP1CON      ;Load CCP1CON with this
                    ; value
```

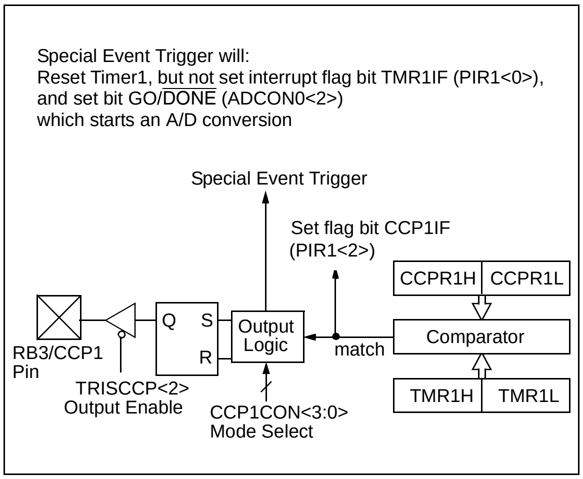
## 7.2 Compare Mode

In Compare mode, the 16-bit CCPR1 register value is constantly compared against the TMR1 register pair value. When a match occurs, the RB3/CCP1 pin is either:

- driven High
- driven Low
- remains Unchanged

The action on the pin is based on the value of control bits CCP1M3:CCP1M0 (CCP1CON<3:0>). At the same time, interrupt flag bit CCP1IF is set.

**FIGURE 7-4: COMPARE MODE OPERATION BLOCK DIAGRAM**




### 7.2.1 CCP PIN CONFIGURATION

The user must configure the RB3/CCP1 pin as the CCP output by clearing the TRISCCP<2> bit.

| **Note:** | Clearing the CCP1CON register will force the RB3/CCP1 compare output latch to the default low level. This is neither the PORTB I/O data latch nor the DATACCP latch. |
|---|---|

### 7.2.2 TIMER1 MODE SELECTION

Timer1 must be running in Timer mode or Synchronized Counter mode if the CCP module is using the compare feature. In Asynchronous Counter mode, the compare operation may not work.

### 7.2.3 SOFTWARE INTERRUPT MODE

When generate software interrupt is chosen the CCP1 pin is not affected. Only a CCP interrupt is generated (if enabled).

### 7.2.4 SPECIAL EVENT TRIGGER

In this mode, an internal hardware trigger is generated which may be used to initiate an action.

The Special Event Trigger output of CCP1 resets the TMR1 register pair. This allows the CCPR1 register to effectively be a 16-bit programmable period register for Timer1.

The Special Event Trigger output of CCP1 also starts an A/D conversion (if the A/D module is enabled).

| **Note:** | The Special Event Trigger from the CCP1 module will not set interrupt flag bit TMR1IF (PIR1<0>). |
|---|---|

**TABLE 7-2: REGISTERS ASSOCIATED WITH CAPTURE, COMPARE, AND TIMER1**

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 07h | DATACCP | — | — | — | — | — | DCCP | — | DT1CK | xxxx xxxx | xxxx xuxu |
| 0Bh,8Bh | INTCON | GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | RBIF | 0000 000x | 0000 000u |
| 0Ch | PIR1 | — | ADIF | — | — | — | CCP1IF | TMR2IF | TMR1IF | -0-- -000 | -0-- -000 |
| 0Eh | TMR1L | Holding Register for the Least Significant Byte of the 16-bit TMR1 Register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 0Fh | TMR1H | Holding Register for the Most Significant Byte of the 16-bit TMR1 Register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 10h | T1CON | — | — | T1CKPS1 | T1CKPS0 | T1OSCEN | $\overline{\text{T1SYNC}}$ | TMR1CS | TMR1ON | --00 0000 | --uu uuuu |
| 15h | CCPR1L | Capture/Compare/PWM Register 1 (LSB) | | | | | | | | xxxx xxxx | uuuu uuuu |
| 16h | CCPR1H | Capture/Compare/PWM Register 1 (MSB) | | | | | | | | xxxx xxxx | uuuu uuuu |
| 17h | CCP1CON | — | — | DC1B1 | DC1B0 | CCP1M3 | CCP1M2 | CCP1M1 | CCP1M0 | --00 0000 | --00 0000 |
| 87h | TRISCCP | — | — | — | — | — | TCCP | — | TT1CK | xxxx x1x1 | xxxx x1x1 |
| 8Ch | PIE1 | — | ADIE | — | — | — | CCP1IE | TMR2IE | TMR1IE | -0-- -000 | -0-- -000 |

**Legend:** x = unknown, u = unchanged, — = unimplemented read as '0'. Shaded cells are not used by Capture and Timer1.

## 7.4 CCP1 Module and PORTB Operation

When the CCP module is disabled, PORTB<3> operates as a normal I/O pin. When the CCP module is enabled, PORTB<3> operation is affected. Multiplexing details of the CCP1 module are shown on PORTB<3>, refer to Figure 3.6.

Table 7-5 below shows the effects of the CCP module operation on PORTB<3>

**TABLE 7-5: CCP1 MODULE AND PORTB OPERATION**

| CCP1 Module Mode | Control Bits | CCP1 Module Operation | PORTB<3> Operation |
|---|---|---|---|
| Off | CCP1CON = --xx 0000 | Off | PORTB<3> functions as normal I/O. |
| Capture | CCP1CON = --xx 01xx<br>TRISCCP = ---- -1-x | The CCP1 module will capture an event on the RB3/CCP1 pin which is driven by an external circuit. The DCCP bit can read the signal on the RB3/CCP1 pin. | PORTB<3> always reads '0' when configured as input. If PORTB<3> is configured as output, reading PORTB<3> will read the data latch. Writing to PORTB<3> will always store the result in the data latch, but it does not drive the RB3/CCP1 pin. |
| | CCP1CON = --xx 01xx<br>TRISCCP = ---- -0-x | The CCP1 module will capture an event on the RB3/CCP1 pin which is driven by the DCCP bit. The DCCP bit can read the signal on the RB3/CCP1 pin. | |
| Compare | CCP1CON = --xx 10xx<br>TRISCCP = ---- -0-x | The CCP1 module produces an output on the RB3/CCP1 pin when a compare event occurs. The DCCP bit can read the signal on the RB3/CCP1 pin. | |
| PWM | CCP1CON = --xx 11xx<br>TRISCCP = ---- -0-x | The CCP1 module produces the PWM signal on the RB3/CCP1 pin. The DCCP bit can read the signal on the RB3/CCP1 pin. | |

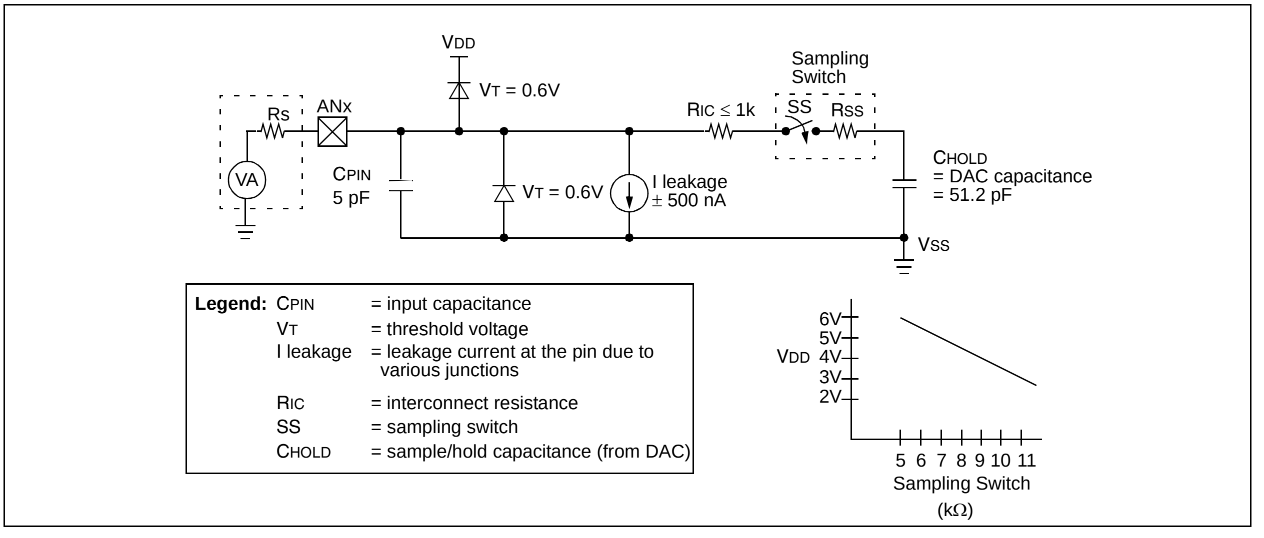
## 8.1 A/D Acquisition Requirements

For the A/D converter to meet its specified accuracy, the Charge Holding capacitor ($C_{HOLD}$) must be allowed to fully charge to the input channel voltage level. The analog input model is shown in Figure 8-4. The source impedance ($R_S$) and the internal sampling switch ($R_{SS}$) impedance directly affect the time required to charge the capacitor $C_{HOLD}$. The sampling switch ($R_{SS}$) impedance varies over the device voltage ($V_{DD}$). The source impedance affects the offset voltage at the analog input (due to pin leakage current). **The maximum recommended impedance for analog sources is 10 kΩ**. After the analog input channel is selected (changed) this acquisition must be done before the conversion can be started.

To calculate the minimum acquisition time, $T_{ACQ}$, see the PIC® Mid-Range Reference Manual, (DS33023). This equation calculates the acquisition time to within 1/2 LSb error (512 steps for the A/D). The 1/2 LSb error is the maximum error allowed for the A/D to meet its specified accuracy.

| Note: | When the conversion is started, the holding capacitor is disconnected from the input pin. |
|---|---|

**FIGURE 8-4: ANALOG INPUT MODEL**



# 9.0 SPECIAL FEATURES OF THE CPU

The PIC16C712/716 devices have a host of features intended to maximize system reliability, minimize cost through elimination of external components, provide power-saving operating modes and offer code protection. These are:

- OSC Selection
- Reset:
  - Power-on Reset (POR)
  - Power-up Timer (PWRT)
  - Oscillator Start-up Timer (OST)
  - Brown-out Reset (BOR)
- Interrupts
- Watchdog Timer (WDT)
- Sleep
- Code protection
- ID locations
- In-Circuit Serial Programming™ (ICSP™)

These devices have a Watchdog Timer, which can be shut off only through Configuration bits. It runs off its own RC oscillator for added reliability. There are two timers that offer necessary delays on power-up. One is the Oscillator Start-up Timer (OST), intended to keep the chip in Reset until the crystal oscillator is stable. The other is the Power-up Timer (PWRT), which provides a fixed delay on power-up only and is designed to keep the part in Reset while the power supply stabilizes. With these two timers on-chip, most applications need no external Reset circuitry.

Sleep mode is designed to offer a very low-current Power-Down mode. The user can wake-up from Sleep through external Reset, Watchdog Timer Wake-up, or through an interrupt. Several oscillator options are also made available to allow the part to fit the application. The RC oscillator option saves system cost, while the LP crystal option saves power. A set of Configuration bits are used to select various options.

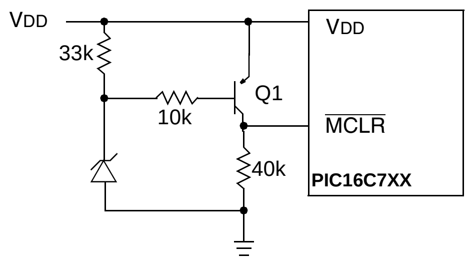
Additional information on special features is available in the PIC® Mid-Range Reference Manual, (DS33023).

## 9.1 Configuration Bits

The Configuration bits can be programmed (read as '0') or left unprogrammed (read as '1') to select various device configurations. These bits are mapped in program memory location 2007h.

The user will note that address 2007h is beyond the user program memory space. In fact, it belongs to the special test/configuration memory space (2000h-3FFFh), which can be accessed only during programming.

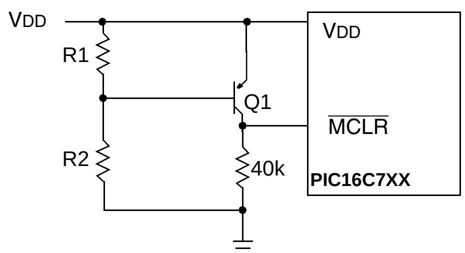
**FIGURE 9-8: EXTERNAL BROWN-OUT PROTECTION CIRCUIT 1**




**Note 1:** This circuit will activate Reset when VDD goes below (Vz + 0.7V) where Vz = Zener voltage.

**2:** Internal Brown-out Reset circuitry should be disabled when using this circuit.

**FIGURE 9-9: EXTERNAL BROWN-OUT PROTECTION CIRCUIT 2**



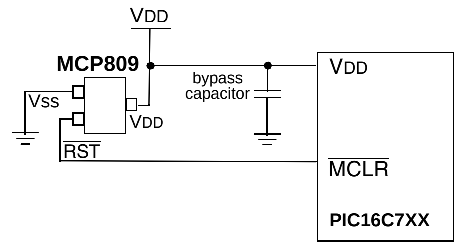

**Note 1:** This brown-out circuit is less expensive, albeit less accurate. Transistor Q1 turns off when VDD is below a certain level such that:

$$V_{DD} \times \frac{R1}{R1 + R2} = 0.7V$$

**2:** Internal Brown-out Reset should be disabled when using this circuit.

**3:** Resistors should be adjusted for the characteristics of the transistor.

**FIGURE 9-10: EXTERNAL BROWN-OUT PROTECTION CIRCUIT 3**




This brown-out protection circuit employs Microchip Technology's MCP809 microcontroller supervisor. The MCP8XX and MCP1XX families of supervisors provide push-pull and open collector outputs with both high and low active Reset pins. There are 7 different trip point selections to accommodate 5V and 3V systems

## 9.8 Time-out Sequence

On power-up the time-out sequence is as follows: First PWRT time-out is invoked after the POR time delay has expired. Then OST is activated. The total time-out will vary based on oscillator configuration and the status of the PWRT. For example, in RC mode with the PWRT disabled, there will be no time-out at all. Figure 9-11, Figure 9-12, and Figure 9-13 depict time-out sequences on power-up.

Since the time-outs occur from the POR pulse, if $\overline{\text{MCLR}}$ is kept low long enough, the time-outs will expire. Then bringing $\overline{\text{MCLR}}$ high will begin execution immediately (Figure 9-13). This is useful for testing purposes or to synchronize more than one PIC16CXXX device operating in parallel.

Table 9-5 shows the Reset conditions for some Special Function Registers, while Table 9-6 shows the Reset conditions for all the registers.

### 9.10.1 INT INTERRUPT

External interrupt on RB0/INT pin is edge triggered, either rising if bit INTEDG (OPTION_REG<6>) is set, or falling if the INTEDG bit is clear. When a valid edge appears on the RB0/INT pin, flag bit INTF (INTCON<1>) is set. This interrupt can be disabled by clearing enable bit INTE (INTCON<4>). Flag bit INTF must be cleared in software in the Interrupt Service Routine before re-enabling this interrupt. The INT interrupt can wake-up the processor from Sleep, if bit INTE was set prior to going into Sleep. The status of global interrupt enable bit GIE decides whether or not the processor branches to the interrupt vector following wake-up. See **Section 9.13 "Power-down Mode (Sleep)"** for details on Sleep mode.

### 9.10.2 TMR0 INTERRUPT

An overflow (FFh $\rightarrow$ 00h) in the TMR0 register will set flag bit T0IF (INTCON<2>). The interrupt can be enabled/disabled by setting/clearing enable bit T0IE (INTCON<5>). (**Section 4.0 "Timer0 Module"**)

### 9.10.3 PORTB INTCON CHANGE

An input change on PORTB<7:4> sets flag bit RBIF (INTCON<0>). The interrupt can be enabled/disabled by setting/clearing enable bit RBIE (INTCON<4>). (**Section 3.2 "PORTB and the TRISB Register"**)

## 9.11 Context Saving During Interrupts

During an interrupt, only the return PC value is saved on the stack. Typically, users may wish to save key registers during an interrupt, (i.e., W register and STATUS register). This will have to be implemented in software.

Example 9-1 stores and restores the W and STATUS registers. The register, W_TEMP, must be defined in each bank and must be defined at the same offset from the bank base address (i.e., if W_TEMP is defined at 0x20 in bank 0, it must also be defined at 0xA0 in bank 1).

The example:

a) Stores the W register.
b) Stores the STATUS register in bank 0.
c) Stores the PCLATH register.
d) Executes the Interrupt Service Routine code (User-generated).
e) Restores the STATUS register (and bank select bit).
f) Restores the W and PCLATH registers.

**EXAMPLE 9-1: SAVING STATUS, W, AND PCLATH REGISTERS IN RAM**

```
MOVWF   W_TEMP           ;Copy W to TEMP register, could be bank one or zero
SWAPF   STATUS,W         ;Swap status to be saved into W
CLRF    STATUS           ;bank 0, regardless of current bank, Clears IRP,RP1,RP0
MOVWF   STATUS_TEMP      ;Save status to bank zero STATUS_TEMP register
MOVF    PCLATH, W        ;Only required if using pages 1, 2 and/or 3
MOVWF   PCLATH_TEMP      ;Save PCLATH into W
CLRF    PCLATH           ;Page zero, regardless of current page
BCF     STATUS, IRP      ;Return to Bank 0
MOVF    FSR, W           ;Copy FSR to W
MOVWF   FSR_TEMP         ;Copy FSR from W to FSR_TEMP
:
:(ISR)
:
MOVF    PCLATH_TEMP, W   ;Restore PCLATH
MOVWF   PCLATH           ;Move W into PCLATH
SWAPF   STATUS_TEMP,W    ;Swap STATUS_TEMP register into W
                         ;(sets bank to original state)
MOVWF   STATUS           ;Move W into STATUS register
SWAPF   W_TEMP,F         ;Swap W_TEMP

SWAPF   W_TEMP,W         ;Swap W_TEMP into W
```

## 9.16 In-Circuit Serial Programming™

PIC16CXXX 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. 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 complete details on serial programming, please refer to the In-Circuit Serial Programming™ (ICSP™) Guide, (DS30277).

## 10.0 INSTRUCTION SET SUMMARY

Each PIC16CXXX 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 PIC16CXXX instruction set summary in Table 10-2 lists **byte-oriented**, **bit-oriented**, and **literal and control** operations. Table 10-1 shows the opcode field descriptions.

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 number of the bit affected by the operation, while 'f' represents the number 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.

**TABLE 10-1: 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)<br>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,<br>d = 1: store result in file register f.<br>Default is d = 1 |
| PC | Program Counter |
| $\overline{TO}$ | Time-out bit |
| $\overline{PD}$ | Power-down bit |
| Z | Zero bit |
| DC | Digit Carry bit |
| C | Carry bit |

The instruction set is highly orthogonal and is grouped into three basic categories:

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

All instructions are executed within one single instruction cycle, unless a conditional test is true or the program counter is changed as a result of an instruction. In this case, the execution takes two instruction cycles with the second cycle executed as a NOP. One instruction cycle consists of four oscillator periods. Thus, for an oscillator frequency of 4 MHz, the normal instruction execution time is 1 μs. If a conditional test is true or the program counter is changed as a result of an instruction, the instruction execution time is 2 μs.

Table 10-2 lists the instructions recognized by the MPASM assembler.

Figure 10-1 shows the general formats that the instructions can have.

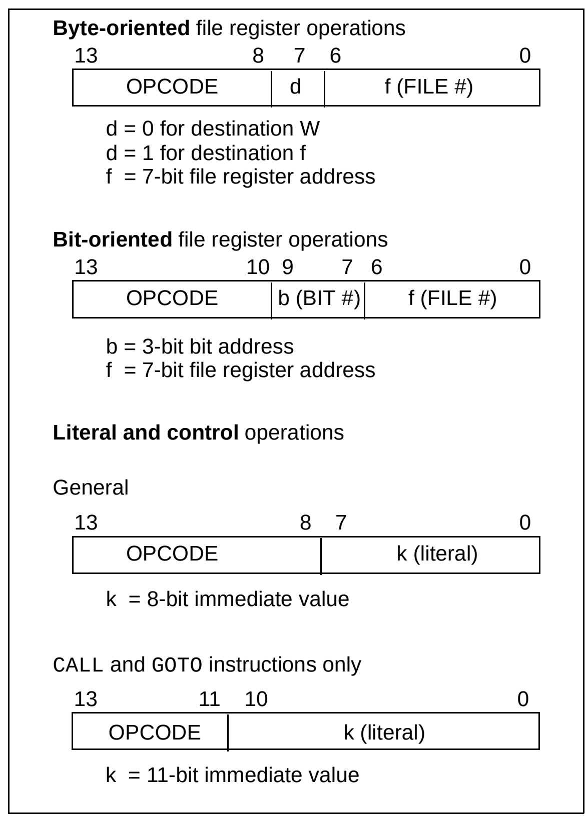
> **Note:** To maintain upward compatibility with future PIC16CXXX products, do not use the OPTION and TRIS instructions.

All examples use the following format to represent a hexadecimal number:

0xhh

where h signifies a hexadecimal digit.

**FIGURE 10-1: GENERAL FORMAT FOR INSTRUCTIONS**




A description of each instruction is available in the PIC® Mid-Range Reference Manual, (DS33023).

**TABLE 10-2: PIC16CXXX INSTRUCTION SET**

| Mnemonic, Operands | | Description | Cycles | 14-Bit Opcode MSb | | | LSb | Status Affected | Notes |
|---|---|---|---|---|---|---|---|---|---|
| **BYTE-ORIENTED FILE REGISTER OPERATIONS** | | | | | | | | | |
| **ADDWF** | **f, d** | Add W and f | 1 | 00 | 0111 | dfff | ffff | C,DC,Z | **1,2** |
| **ANDWF** | **f, d** | AND W with f | 1 | 00 | 0101 | dfff | ffff | Z | **1,2** |
| **CLRF** | **f** | Clear f | 1 | 00 | 0001 | lfff | ffff | Z | **2** |
| **CLRW** | **-** | Clear W | 1 | 00 | 0001 | 0000 | 0011 | Z | |
| **COMF** | **f, d** | Complement f | 1 | 00 | 1001 | dfff | ffff | Z | **1,2** |
| **DECF** | **f, d** | Decrement f | 1 | 00 | 0011 | dfff | ffff | Z | **1,2** |
| **DECFSZ** | **f, d** | Decrement f, Skip if 0 | 1(2) | 00 | 1011 | dfff | ffff | | **1,2,3** |
| **INCF** | **f, d** | Increment f | 1 | 00 | 1010 | dfff | ffff | Z | **1,2** |
| **INCFSZ** | **f, d** | Increment f, Skip if 0 | 1(2) | 00 | 1111 | dfff | ffff | | **1,2,3** |
| **IORWF** | **f, d** | Inclusive OR W with f | 1 | 00 | 0100 | dfff | ffff | Z | **1,2** |
| **MOVF** | **f, d** | Move f | 1 | 00 | 1000 | dfff | ffff | Z | **1,2** |
| **MOVWF** | **f** | Move W to f | 1 | 00 | 0000 | lfff | ffff | | |
| **NOP** | **-** | No Operation | 1 | 00 | 0000 | 0xx0 | 0000 | | |
| **RLF** | **f, d** | Rotate Left f through Carry | 1 | 00 | 1101 | dfff | ffff | C | **1,2** |
| **RRF** | **f, d** | Rotate Right f through Carry | 1 | 00 | 1100 | dfff | ffff | C | **1,2** |
| **SUBWF** | **f, d** | Subtract W from f | 1 | 00 | 0010 | dfff | ffff | C,DC,Z | **1,2** |
| **SWAPF** | **f, d** | Swap nibbles in f | 1 | 00 | 1110 | dfff | ffff | | **1,2** |
| **XORWF** | **f, d** | Exclusive OR W with f | 1 | 00 | 0110 | dfff | ffff | Z | **1,2** |
| **BIT-ORIENTED FILE REGISTER OPERATIONS** | | | | | | | | | |
| **BCF** | **f, b** | Bit Clear f | 1 | 01 | 00bb | bfff | ffff | | **1,2** |
| **BSF** | **f, b** | Bit Set f | 1 | 01 | 01bb | bfff | ffff | | **1,2** |
| **BTFSC** | **f, b** | Bit Test f, Skip if Clear | 1 (2) | 01 | 10bb | bfff | ffff | | **3** |
| **BTFSS** | **f, b** | Bit Test f, Skip if Set | 1 (2) | 01 | 11bb | bfff | ffff | | **3** |
| **LITERAL AND CONTROL OPERATIONS** | | | | | | | | | |
| **ADDLW** | **k** | Add literal and W | 1 | 11 | 111x | kkkk | kkkk | C,DC,Z | |
| **ANDLW** | **k** | AND literal with W | 1 | 11 | 1001 | kkkk | kkkk | Z | |
| **CALL** | **k** | Call subroutine | 2 | 10 | 0kkk | kkkk | kkkk | | |
| **CLRWDT** | **-** | Clear Watchdog Timer | 1 | 00 | 0000 | 0110 | 0100 | $\overline{TO}$,$\overline{PD}$ | |
| **GOTO** | **k** | Go to address | 2 | 10 | 1kkk | kkkk | kkkk | | |
| **IORLW** | **k** | Inclusive OR literal with W | 1 | 11 | 1000 | kkkk | kkkk | Z | |
| **MOVLW** | **k** | Move literal to W | 1 | 11 | 00xx | kkkk | kkkk | | |
| **RETFIE** | **-** | Return from interrupt | 2 | 00 | 0000 | 0000 | 1001 | | |
| **RETLW** | **k** | Return with literal in W | 2 | 11 | 01xx | kkkk | kkkk | | |
| **RETURN** | **-** | Return from Subroutine | 2 | 00 | 0000 | 0000 | 1000 | | |
| **SLEEP** | **-** | Go into standby mode | 1 | 00 | 0000 | 0110 | 0011 | $\overline{TO}$,$\overline{PD}$ | |
| **SUBLW** | **k** | Subtract W from literal | 1 | 11 | 110x | kkkk | kkkk | C,DC,Z | |
| **XORLW** | **k** | Exclusive OR literal with W | 1 | 11 | 1010 | kkkk | kkkk | Z | |

**Note 1:** When an I/O register is modified as a function of itself (e.g., `MOVF PORTB, 1`), the value used will be that value present on the pins themselves. For example, if the data latch is ‘1’ for a pin configured as input and is driven low by an external device, the data will be written back with a ‘0’.

**2:** If this instruction is executed on the TMR0 register (and, where applicable, d = 1), the prescaler will be cleared if assigned to the Timer0 Module.

**3:** If Program Counter (PC) is modified or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a `NOP`.

## 12.1 DC Characteristics: PIC16C712/716-04 (Commercial, Industrial, Extended) PIC16C712/716-20 (Commercial, Industrial, Extended)

| DC CHARACTERISTICS | | | Standard Operating Conditions (unless otherwise stated)<br>Operating temperature 0°C ≤ TA ≤ +70°C for commercial<br>-40°C ≤ TA ≤ +85°C for industrial<br>-40°C ≤ TA ≤ +125°C for extended | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Sym.** | **Characteristic** | **Min.** | **Typ†** | **Max.** | **Units** | **Conditions** |
| D001<br>D001A | VDD | Supply Voltage | 4.0<br>4.5<br>VBOR* | —<br>—<br>— | 5.5<br>5.5<br>5.5 | V<br>V<br>V | XT, RC and LP osc mode<br>HS osc mode<br>BOR enabled(7) |
| D002* | VDR | **RAM Data Retention Voltage(1)** | — | 1.5 | — | V | |
| D003 | VPOR | **VDD Start Voltage** to ensure internal Power-on Reset signal | — | VSS | — | V | See section on Power-on Reset for details |
| D004*<br>D004A* | SVDD | **VDD Rise Rate** to ensure internal Power-on Reset signal | 0.05<br>TBD | —<br>— | —<br>— | V/ms | PWRT enabled ($\overline{\text{PWRTE}}$ bit clear)<br>PWRT disabled ($\overline{\text{PWRTE}}$ bit set)<br>See section on Power-on Reset for details |
| D005 | VBOR | **Brown-out Reset** voltage trip point | 3.65 | — | 4.35 | V | BODEN bit set |
| D010<br>D013 | IDD | **Supply Current(2,5)** | —<br>— | 0.8<br>4.0 | 2.5<br>8.0 | mA<br>mA | FOSC = 4 MHz, VDD = 4.0V<br>FOSC = 20 MHz, VDD = 4.0V |
| D020<br><br>D021<br>D021B | IPD | **Power-down Current(3,5)** | —<br>—<br>—<br>— | 10.5<br>1.5<br>1.5<br>2.5 | 42<br>16<br>19<br>19 | μA<br>μA<br>μA<br>μA | VDD = 4.0V, WDT enabled,-40°C to +85°C<br>VDD = 4.0V, WDT disabled, 0°C to +70°C<br>VDD = 4.0V, WDT disabled,-40°C to +85°C<br>VDD = 4.0V, WDT disabled,-40°C to +125°C |
| <br>D022*<br>D022A* | <br>ΔIWDT<br>ΔIBOR | **Module Differential Current(6)**<br>Watchdog Timer<br>Brown-out Reset | <br>—<br>— | <br>6.0<br>TBD | <br>20<br>200 | <br>μA<br>μA | <br>WDTE bit set, VDD = 4.0V<br>BODEN bit set, VDD = 5.0V |
| 1A | FOSC | LP Oscillator Operating Frequency<br>RC Oscillator Operating Frequency<br>XT Oscillator Operating Frequency<br>HS Oscillator Operating Frequency | 0<br>0<br>0<br>0 | —<br>—<br>—<br>— | 200<br>4<br>4<br>20 | KHz<br>MHz<br>MHz<br>MHz | All temperatures<br>All temperatures<br>All temperatures<br>All temperatures |

* These parameters are characterized but not tested.

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

**Note1:** This is the limit to which VDD can be lowered without losing RAM data.

**2:** The supply current is mainly a function of the operating voltage and frequency. Other factors such as I/O pin loading and switching rate, oscillator type, internal code execution pattern, and temperature also have an impact on the current consumption.
The test conditions for all IDD measurements in active operation mode are:
OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to VDD,
$\overline{\text{MCLR}}$ = VDD; WDT enabled/disabled as specified.

**3:** The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in high-impedance state and tied to VDD and VSS.

**4:** For RC Osc mode, current through REXT is not included. The current through the resistor can be estimated by the formula Ir = VDD/2REXT (mA) with REXT in kOhm.

**5:** Timer1 oscillator (when enabled) adds approximately 20 μA to the specification. This value is from characterization and is for design guidance only. This is not tested.

**6:** The Δ current is the additional current consumed when this peripheral is enabled. This current should be added to the base IDD or IPD measurement.

**7:** This is the voltage where the device enters the Brown-out Reset. When BOR is enabled, the device will operate correctly to this trip point.

## 12.4 AC (Timing) Characteristics

### 12.4.1 TIMING PARAMETER SYMBOLOGY

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

1. TppS2ppS
2. TppS

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

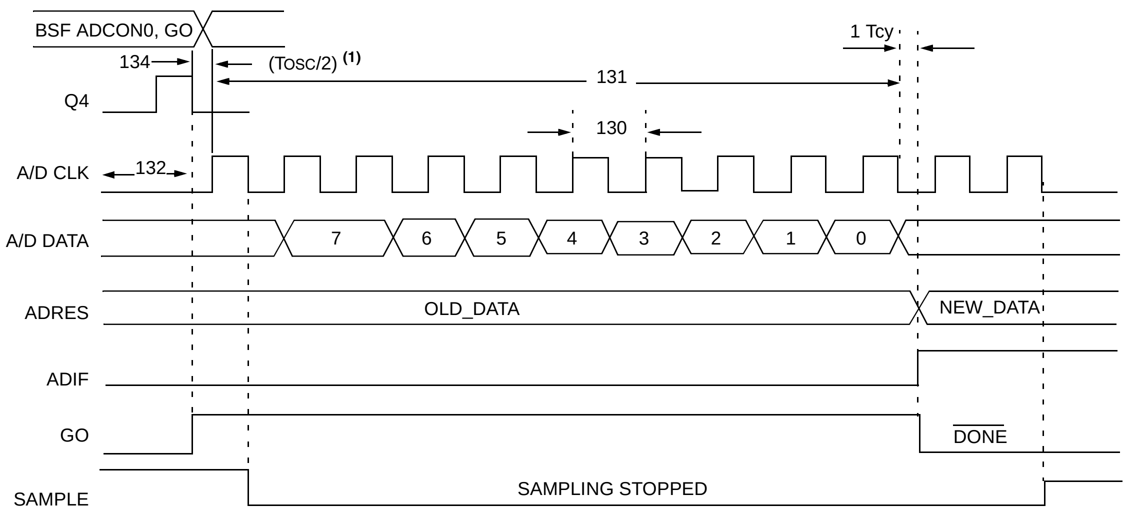
Lowercase letters (pp) and their meanings:

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

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 |

**FIGURE 12-10: A/D CONVERSION TIMING**




**Note1:** If the A/D clock source is selected as RC, a time of TCY is added before the A/D clock starts. This allows the SLEEP instruction to be executed.

**TABLE 12-8: A/D CONVERSION REQUIREMENTS**

| Param No. | Sym. | Characteristic | | Min. | Typ† | Max. | Units | Conditions |
|---|---|---|---|---|---|---|---|---|
| 130 | TAD | A/D clock period | Standard | 1.6 | — | — | μs | TOSC based, VREF ≥ 3.0V |
| | | | Extended (LC) | 2.0 | — | — | μs | TOSC based, VREF full range |
| | | | Standard | 2.0 | 4.0 | 6.0 | μs | A/D RC Mode |
| | | | Extended (LC) | 3.0 | 6.0 | 9.0 | μs | A/D RC Mode |
| 131 | TCNV | Conversion time (not including S/H time) **(Note 1)** | | 11 | — | 11 | TAD | |
| 132 | TACQ | Acquisition time | | **(Note 2)** | 20 | — | μs | |
| | | | | 5* | — | — | μs | The minimum time is the amplifier settling time. This may be used if the “new” input voltage has not changed by more than 1 LSb (i.e., 20.0 mV @ 5.12V) from the last sampled voltage (as stated on CHOLD). |
| 134 | TGO | Q4 to A/D clock start | | — | TOSC/2 § | — | — | If the A/D clock source is selected as RC, a time of TCY is added before the A/D clock starts. This allows the SLEEP instruction to be executed. |
| 135 | TSWC | Switching from convert Æ sample time | | 1.5 § | — | — | TAD | |

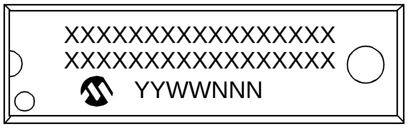
: * These parameters are characterized but not tested.
: † Data in “Typ” column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.
: § This specification ensured by design.

**Note 1:** ADRES register may be read on the following TCY cycle.
**2:** See **Section 9.1 “Configuration Bits”** for min. conditions.

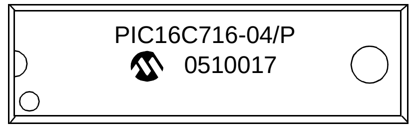
# 13.0 PACKAGING INFORMATION

## 13.1 Package Marking Information

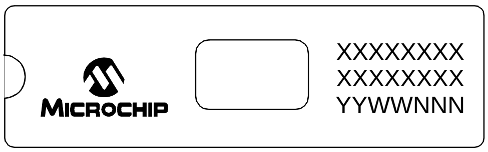
18-Lead PDIP



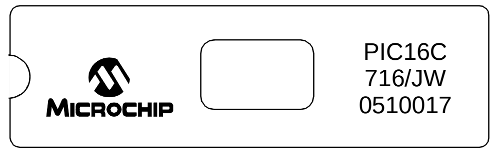

Example



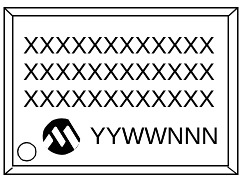

18-Lead CERDIP Windowed



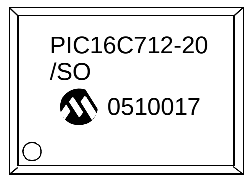

Example



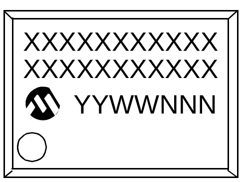

18-Lead SOIC (.300”)



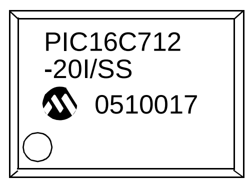

Example




20-Lead SSOP




Example




| | | |
|---|---|---|
| **Legend:** | XX...X | Customer-specific information |
| | Y | Year code (last digit of calendar year) |
| | YY | Year code (last 2 digits of calendar year) |
| | WW | Week code (week of January 1 is week ‘01’) |
| | NNN | Alphanumeric traceability code |
| | e3 | Pb-free JEDEC designator for Matte Tin (Sn) |
| | * | This package is Pb-free. The Pb-free JEDEC designator (e3) can be found on the outer packaging for this package. |

**Note**: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line, thus limiting the number of available characters for customer-specific information.

**NOTES:**

## M



## O



## P