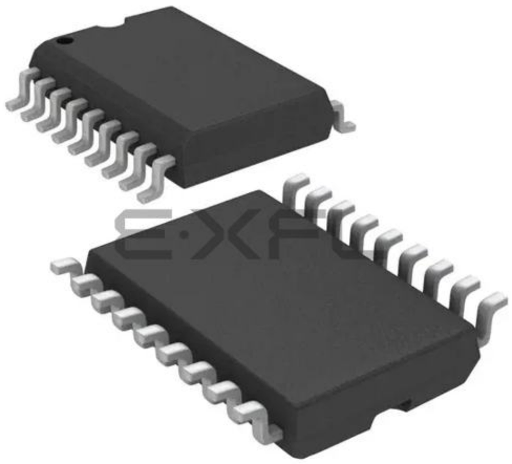


Welcome to E-XFL.COM

## 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 ~ 85°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/pic16c716-20i-so |

## Table of Contents



## TO OUR VALUED CUSTOMERS

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

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

### Most Current Data Sheet

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

http://www.microchip.com

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

### Errata

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

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

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

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

### Customer Notification System

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

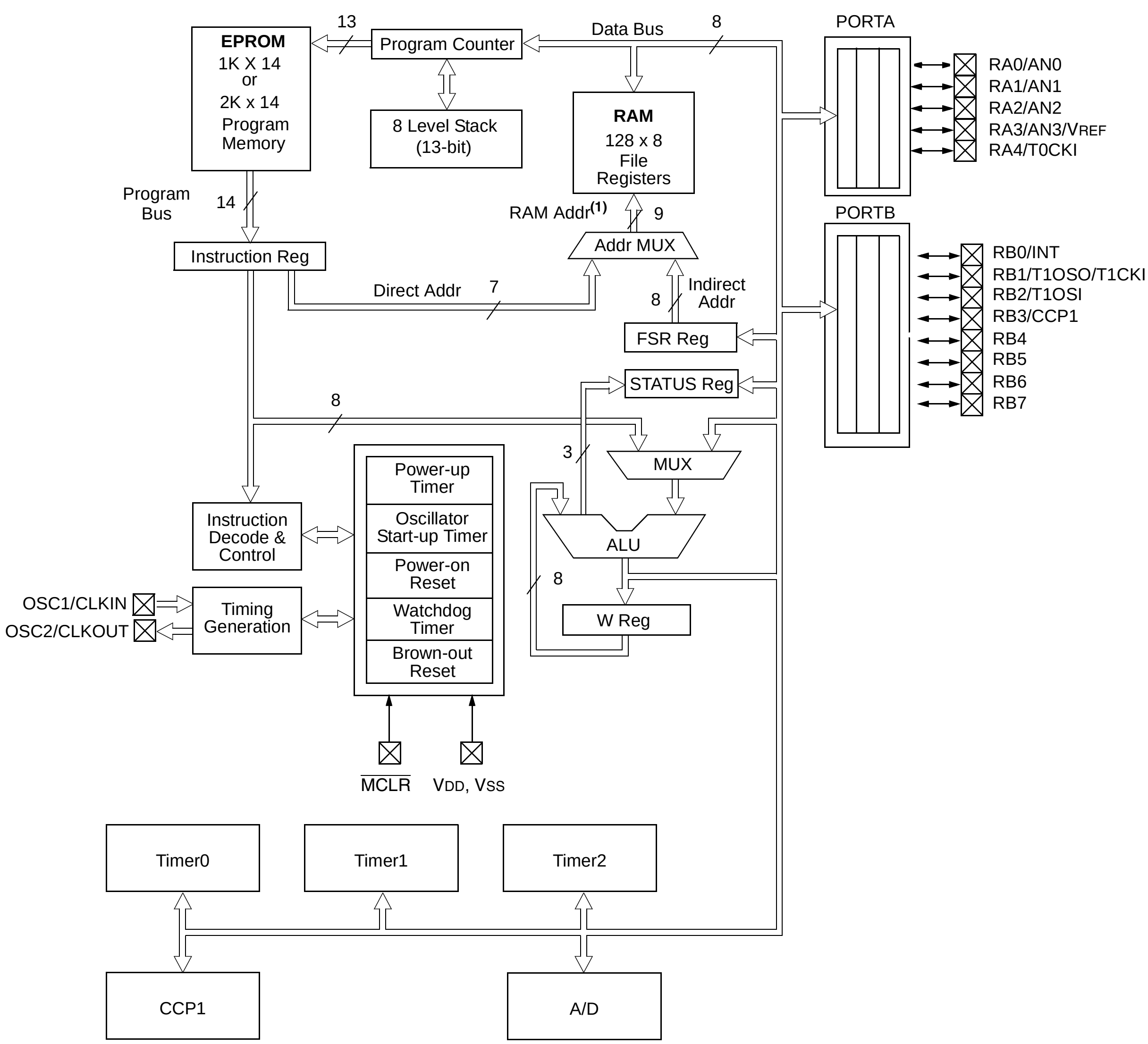
## 1.0 DEVICE OVERVIEW

This document contains device-specific information. Additional information may be found in the PIC® Mid-Range Reference Manual, (DS33023), which may be obtained from your local Microchip Sales Representative or downloaded from the Microchip web site. The Reference Manual should be considered a complementary document to this data sheet, and is highly recommended reading for a better understanding of the device architecture and operation of the peripheral modules.

There are two devices (PIC16C712, PIC16C716) covered by this data sheet.

Figure 1-1 is the block diagram for both devices. The pinouts are listed in Table 1-1.

**FIGURE 1-1: PIC16C712/716 BLOCK DIAGRAM**




**Note 1:** Higher order bits are from the STATUS register.

**TABLE 1-1: PIC16C712/716 PINOUT DESCRIPTION**

| Pin Name | PIC16C712/716 DIP, SOIC | PIC16C712/716 SSOP | Pin Type | Buffer Type | Description |
|---|---|---|---|---|---|
| $\overline{MCLR}/V_{PP}$ | 4 | 4 | | | |
| $\overline{MCLR}$ | | | I | ST | Master clear (Reset) input. This pin is an active low Reset to the device. |
| $V_{PP}$ | | | P | | Programming voltage input |
| OSC1/CLKIN | 16 | 18 | | | |
| OSC1 | | | I | ST | Oscillator crystal input or external clock source input. ST buffer when configured in RC mode. CMOS otherwise. |
| CLKIN | | | I | CMOS | External clock source input. |
| OSC2/CLKOUT | 15 | 17 | | | |
| OSC2 | | | O | — | Oscillator crystal output. Connects to crystal or resonator in crystal oscillator mode. |
| CLKOUT | | | O | — | In RC mode, OSC2 pin outputs CLKOUT which has 1/4 the frequency of OSC1, and denotes the instruction cycle rate. |
| | | | | | PORTA is a bidirectional I/O port. |
| RA0/AN0 | 17 | 19 | | | |
| RA0 | | | I/O | TTL | Digital I/O |
| AN0 | | | I | Analog | Analog input 0 |
| RA1/AN1 | 18 | 20 | | | |
| RA1 | | | I/O | TTL | Digital I/O |
| AN1 | | | I | Analog | Analog input 1 |
| RA2/AN2 | 1 | 1 | | | |
| RA2 | | | I/O | TTL | Digital I/O |
| AN2 | | | I | Analog | Analog input 2 |
| RA3/AN3/$V_{REF}$ | 2 | 2 | | | |
| RA3 | | | I/O | TTL | Digital I/O |
| AN3 | | | I | Analog | Analog input 3 |
| $V_{REF}$ | | | I | Analog | A/D Reference Voltage input. |
| RA4/T0CKI | 3 | 3 | | | |
| RA4 | | | I/O | ST/OD | Digital I/O. Open drain when configured as output. |
| T0CKI | | | I | ST | Timer0 external clock input |

**Legend:** TTL = TTL-compatible input  CMOS = CMOS compatible input or output
ST = Schmitt Trigger input with CMOS levels
OD = Open drain output
SM = SMBus compatible input. An external resistor is required if this pin is used as an output
NPU = N-channel pull-up  PU = Weak internal pull-up
No-P diode = No P-diode to $V_{DD}$  AN = Analog input or output
I = input  O = output
P = Power  L = LCD Driver

### 2.2.2.2 OPTION_REG Register

The OPTION_REG register is a readable and writable register, which contains various control bits to configure the TMR0 prescaler/WDT postscaler (single assignable register known also as the prescaler), the External INT Interrupt, TMR0 and the weak pull-ups on PORTB.

| **Note:** | To achieve a 1:1 prescaler assignment for the TMR0 register, assign the prescaler to the Watchdog Timer. |
|---|---|

**FIGURE 2-5: OPTION_REG REGISTER (ADDRESS 81h)**

| R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 |
|---|---|---|---|---|---|---|---|
| $\overline{RBPU}$ | INTEDG | T0CS | T0SE | PSA | PS2 | PS1 | PS0 |

bit7 bit0

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

bit 7: **$\overline{RBPU}$**: PORTB Pull-up Enable bit
1 = PORTB pull-ups are disabled
0 = PORTB pull-ups are enabled by individual port latch values

bit 6: **INTEDG**: Interrupt Edge Select bit
1 = Interrupt on rising edge of RB0/INT pin
0 = Interrupt on falling edge of RB0/INT pin

bit 5: **T0CS**: TMR0 Clock Source Select bit
1 = Transition on RA4/T0CKI pin
0 = Internal instruction cycle clock (CLKOUT)

bit 4: **T0SE**: TMR0 Source Edge Select bit
1 = Increment on high-to-low transition on RA4/T0CKI pin
0 = Increment on low-to-high transition on RA4/T0CKI pin

bit 3: **PSA**: Prescaler Assignment bit
1 = Prescaler is assigned to the WDT
0 = Prescaler is assigned to the Timer0 module

bit 2-0: **PS2:PS0**: Prescaler Rate Select bits

| Bit Value | TMR0 Rate | WDT Rate |
|---|---|---|
| 000 | 1 : 2 | 1 : 1 |
| 001 | 1 : 4 | 1 : 2 |
| 010 | 1 : 8 | 1 : 4 |
| 011 | 1 : 16 | 1 : 8 |
| 100 | 1 : 32 | 1 : 16 |
| 101 | 1 : 64 | 1 : 32 |
| 110 | 1 : 128 | 1 : 64 |
| 111 | 1 : 256 | 1 : 128 |

### 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

## 3.0 I/O PORTS

Some pins for these I/O ports are multiplexed with an alternate function for the peripheral features on the device. In general, when a peripheral is enabled, that pin may not be used as a general purpose I/O pin.

Additional information on I/O ports may be found in the PIC® Mid-Range Reference Manual, (DS33023).

### 3.1 PORTA and the TRISA Register

PORTA is a 5-bit wide bidirectional port. The corresponding data direction register is TRISA. Setting a TRISA bit (= 1) will make the corresponding PORTA pin an input, (i.e., put the corresponding output driver in a High-Impedance mode). Clearing a TRISA bit (= 0) will make the corresponding PORTA pin an output, (i.e., put the contents of the output latch on the selected pin).

Reading the PORTA register reads the status of the pins whereas writing to it will write to the port latch. All write operations are read-modify-write operations. Therefore, a write to a port implies that the port pins are read, the value is modified, and then written to the port data latch.

Pin RA4 is multiplexed with the Timer0 module clock input to become the RA4/T0CKI pin. The RA4/T0CKI pin is a Schmitt Trigger input and an open drain output. All other RA port pins have TTL input levels and full CMOS output drivers.

PORTA pins, RA3:0, are multiplexed with analog inputs and analog $V_{REF}$ input. The operation of each pin is selected by clearing/setting the control bits in the ADCON1 register (A/D Control Register1).

| **Note:** | On a Power-on Reset, these pins are configured as analog inputs and read as '0'. |
|---|---|

The TRISA register controls the direction of the RA pins, even when they are being used as analog inputs. The user must ensure the bits in the TRISA register are maintained set when using them as analog inputs.

**EXAMPLE 3-1: INITIALIZING PORTA**

```
BCF    STATUS, RP0  ;
CLRF   PORTA        ; Initialize PORTA by
                    ; clearing output
                    ; data latches
BSF    STATUS, RP0  ; Select Bank 1
MOVLW  0xEF         ; Value used to
                    ; initialize data
                    ; direction
MOVWF  TRISA        ; Set RA<3:0> as inputs
                    ; RA<4> as outputs

BCF    STATUS, RP0  ; Return to Bank 0
```

**FIGURE 3-1: BLOCK DIAGRAM OF RA3:RA0**

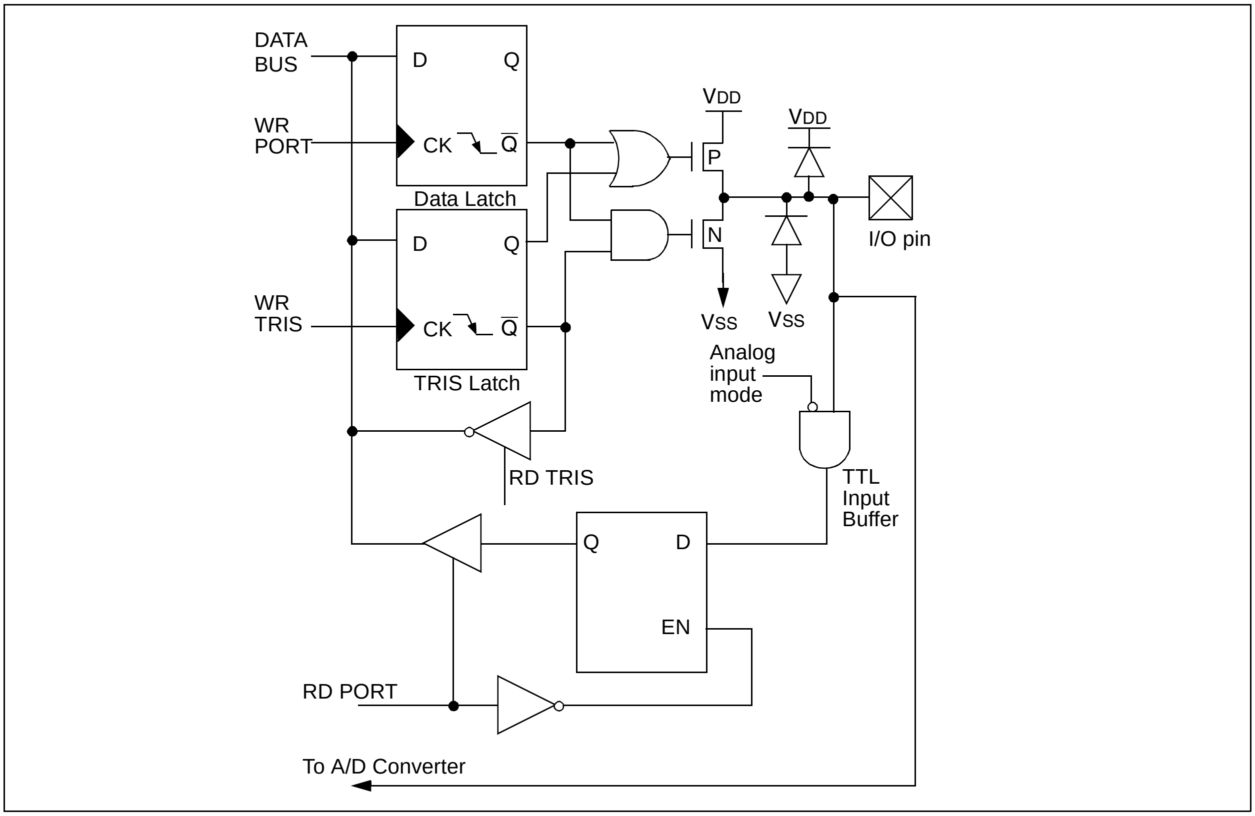
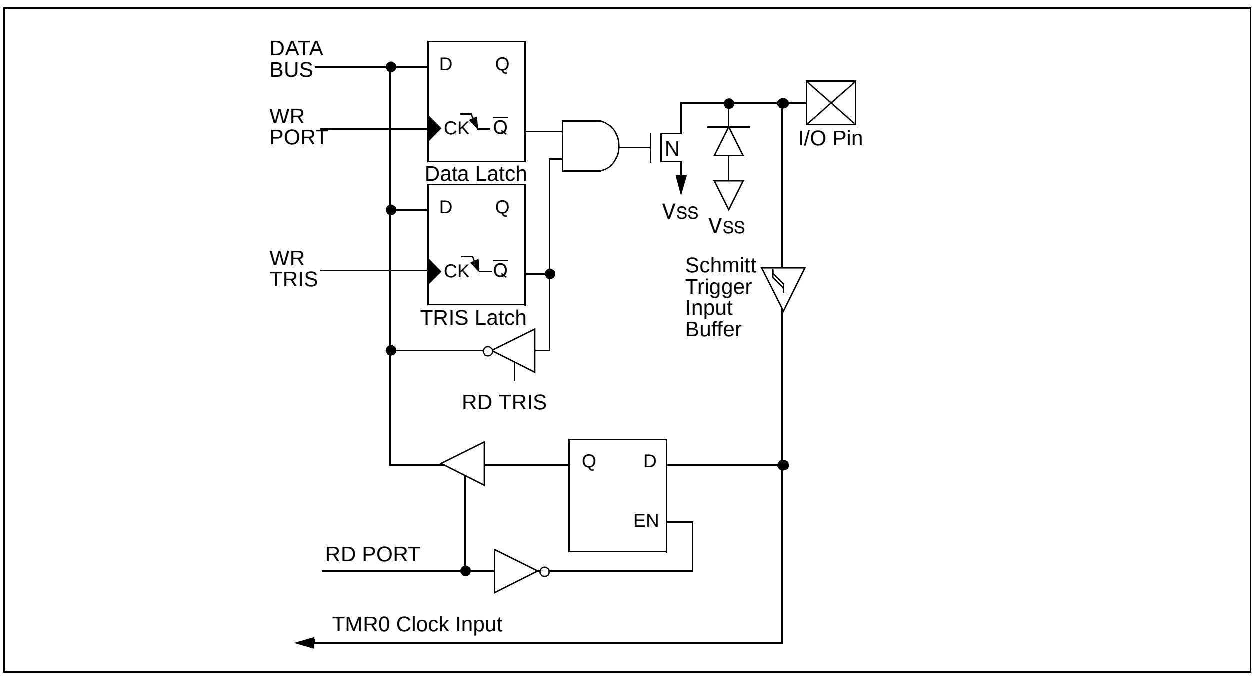



**FIGURE 3-2: BLOCK DIAGRAM OF RA4/T0CKI PIN**



## 5.3 Timer1 Oscillator

A crystal oscillator circuit is built in between pins T1OSI (input) and T1OSO (amplifier output). It is enabled by setting control bit T1OSCEN (T1CON<3>). The oscillator is a low-power oscillator rated up to 200 kHz. It will continue to run during Sleep. It is primarily intended for a 32 kHz crystal. Table 5-2 shows the capacitor selection for the Timer1 oscillator.

The Timer1 oscillator is identical to the LP oscillator. The user must provide a software time delay to ensure proper oscillator start-up.

**TABLE 5-2: CAPACITOR SELECTION FOR THE TIMER1 OSCILLATOR**

| Osc Type | Freq. | C1 | C2 |
|---|---|---|---|
| LP | 32 kHz | 33 pF | 33 pF |
| | 100 kHz | 15 pF | 15 pF |
| | 200 kHz | 15 pF | 15 pF |
| **These values are for design guidance only.** | | | |

**Note 1:** Higher capacitance increases the stability of oscillator but also increases the start-up time.

**2:** Since each resonator/crystal has its own characteristics, the user should consult the resonator/crystal manufacturer for appropriate values of external components.

## 5.4 Timer1 Interrupt

The TMR1 Register pair (TMR1H:TMR1L) increments from 0000h to FFFFh and rolls over to 0000h. The TMR1 interrupt, if enabled, is generated on overflow which is latched in interrupt flag bit TMR1IF (PIR1<0>). This interrupt can be enabled/disabled by setting/clearing TMR1 interrupt enable bit TMR1IE (PIE1<0>).

## 5.5 Resetting Timer1 using a CCP Trigger Output

If the CCP module is configured in Compare mode to generate a "Special Event Trigger" (CCP1M3:CCP1M0 = `1011`), this signal will reset Timer1 and start an A/D conversion (if the A/D module is enabled).

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

Timer1 must be configured for either Timer or Synchronized Counter mode to take advantage of this feature. If Timer1 is running in Asynchronous Counter mode, this reset operation may not work.

In the event that a write to Timer1 coincides with a Special Event Trigger from CCP1, the write will take precedence.

In this mode of operation, the CCPR1H:CCPR1L registers pair effectively becomes the period register for Timer1.

**TABLE 5-3: REGISTERS ASSOCIATED WITH TIMER1 AS A TIMER/COUNTER**

| 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 | INTCON | GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | RBIF | `0000 000x` | `0000 000u` |
| 0Ch | PIR1 | — | ADIF | — | — | — | CCP1IF | TMR2IF | TMR1IF | `-0-- -000` | `-0-- -000` |
| 8Ch | PIE1 | — | ADIE | — | — | — | CCP1IE | TMR2IE | TMR1IE | `-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` |
| 07h | DATACCP | — | — | — | — | — | DCCP | — | DT1CK | `---- -x-x` | `---- -u-u` |
| 87h | TRISCCP | — | — | — | — | — | TCCP | — | TT1CK | `---- -1-1` | `---- -1-1` |

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

## 7.0 CAPTURE/COMPARE/PWM (CCP) MODULE(S)

Each CCP (Capture/Compare/PWM) module contains a 16-bit register, which can operate as a 16-bit capture register, as a 16-bit compare register or as a PWM master/slave Duty Cycle register. Table 7-1 shows the timer resources of the CCP module modes.

Capture/Compare/PWM Register 1 (CCPR1) is comprised of two 8-bit registers: CCPR1L (low byte) and CCPR1H (high byte). The CCP1CON register controls the operation of CCP1. All are readable and writable.

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

**TABLE 7-1: CCP MODE – TIMER RESOURCE**

| CCP Mode | Timer Resource |
|---|---|
| Capture | Timer1 |
| Compare | Timer1 |
| PWM | Timer2 |

**FIGURE 7-1: CCP1CON REGISTER (ADDRESS 17h)**

| U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | — | DC1B1 | DC1B0 | CCP1M3 | CCP1M2 | CCP1M1 | CCP1M0 |
| 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: **DC1B1:DC1B0**: 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 CCPR1L.

bit 3-0: **CCP1M3:CCP1M0**: CCP1 Mode Select bits
0000 = Capture/Compare/PWM off (resets CCP1 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 (CCP1IF bit is set)
1001 = Compare mode, clear output on match (CCP1IF bit is set)
1010 = Compare mode, generate software interrupt on match (CCP1IF bit is set, CCP1 pin is unaffected)
1011 = Compare mode, trigger special event (CCP1IF bit is set; CCP1 resets TMR1 and starts an A/D conversion (if A/D module is enabled))
11xx = PWM mode

**FIGURE 7-2: TRISCCP REGISTER (ADDRESS 87H)**

| R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 |
|---|---|---|---|---|---|---|---|
| — | — | — | — | — | TCCP | — | TT1CK |
| bit7 | | | | | | | bit0 |

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

bit 7-3: **Reserved bits; Do Not Use**

bit 2: **TCCP – Tri-state control bit for CCP**
0 = Output pin driven
1 = Output pin tristated

bit 1: **Reserved bit; Do Not Use**

bit 0: **TT1CK – Tri-state control bit for T1CKI pin**
0 = T1CKI pin is an output
1 = T1CKI pin is an input

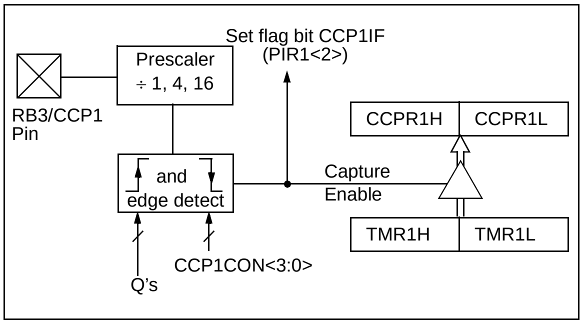
## 7.1 Capture Mode

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
```

### 7.3.3 SET-UP FOR PWM OPERATION

The following steps should be taken when configuring the CCP module for PWM operation:

1. Set the PWM period by writing to the PR2 register.
2. Set the PWM duty cycle by writing to the CCPR1L register and CCP1CON<5:4> bits.
3. Make the CCP1 pin an output by clearing the TRISCCP<2> bit.
4. Set the TMR2 prescale value and enable Timer2 by writing to T2CON.
5. Configure the CCP1 module for PWM operation.

**TABLE 7-3: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS AT 20 MHz**

| PWM Frequency | 1.22 kHz | 4.88 kHz | 19.53 kHz | 78.12 kHz | 156.3 kHz | 208.3 kHz |
|---|---|---|---|---|---|---|
| Timer Prescaler (1, 4, 16) | 16 | 4 | 1 | 1 | 1 | 1 |
| PR2 Value | 0xFF | 0xFF | 0xFF | 0x3F | 0x1F | 0x17 |
| Maximum Resolution (bits) | 10 | 10 | 10 | 8 | 7 | 5.5 |

**TABLE 7-4: REGISTERS ASSOCIATED WITH PWM AND TIMER2**

| 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 |
| 11h | TMR2 | Timer2 Module's Register | | | | | | | | 0000 0000 | 0000 0000 |
| 12h | T2CON | — | TOUTPS3 | TOUTPS2 | TOUTPS1 | TOUTPS0 | TMR2ON | T2CKPS1 | T2CKPS0 | -000 0000 | -000 0000 |
| 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 |
| 92h | PR2 | Timer2 Module's Period Register | | | | | | | | 1111 1111 | 1111 1111 |

**Legend:** x = unknown, u = unchanged, — = unimplemented read as '0'. Shaded cells are not used by PWM and Timer2.

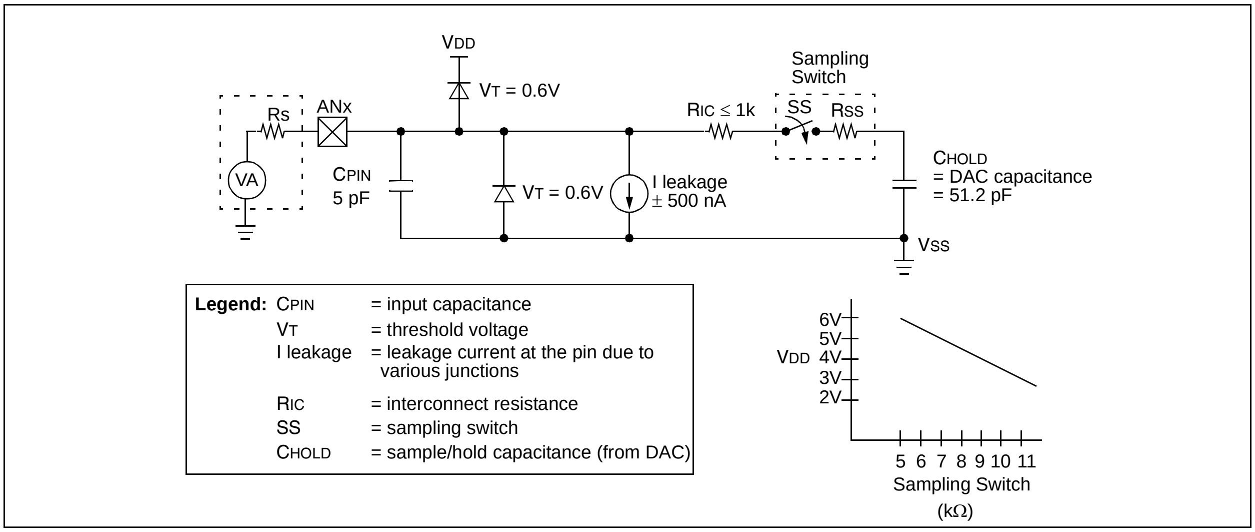
## 8.1 A/D Acquisition Requirements

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**



## 8.2 Selecting the A/D Conversion Clock

The A/D conversion time per bit is defined as $T_{AD}$. The A/D conversion requires $9.5T_{AD}$ per 8-bit conversion. The source of the A/D conversion clock is software selectable. The four possible options for $T_{AD}$ are:

- $2T_{OSC}$
- $8T_{OSC}$
- $32T_{OSC}$
- Internal RC oscillator

For correct A/D conversions, the A/D conversion clock ($T_{AD}$) must be selected to ensure a minimum $T_{AD}$ time of 1.6 μs.

Table 8-1 shows the resultant $T_{AD}$ times derived from the device operating frequencies and the A/D clock source selected.

## 8.3 Configuring Analog Port Pins

The ADCON1 and TRISA registers control the operation of the A/D port pins. The port pins that are desired as analog inputs must have their corresponding TRIS bits set (input). If the TRIS bit is cleared (output), the digital output level ($V_{OH}$ or $V_{OL}$) will be converted.

The A/D operation is independent of the state of the CHS2:CHS0 bits and the TRIS bits.

**Note 1:** When reading the port register, all pins configured as analog input channels will read as cleared (a low level). Pins configured as digital inputs, will convert an analog input. Analog levels on a digitally configured input will not affect the conversion accuracy.

**2:** Analog levels on any pin that is defined as a digital input (including the AN3:AN0 pins), may cause the input buffer to consume current that is out of the devices specification.

**TABLE 8-1: $T_{AD}$ vs. DEVICE OPERATING FREQUENCIES**

| AD Clock Source ($T_{AD}$) | | Device Frequency | | | |
|---|---|---|---|---|---|
| **Operation** | **ADCS1:ADCS0** | **20 MHz** | **5 MHz** | **1.25 MHz** | **333.33 kHz** |
| $2T_{OSC}$ | `00` | 100 ns(2) | 400 ns(2) | 1.6 μs | 6 μs |
| $8T_{OSC}$ | `01` | 400 ns(2) | 1.6 μs | 6.4 μs | 24 μs(3) |
| $32T_{OSC}$ | `10` | 1.6 μs | 6.4 μs | 25.6 μs(3) | 96 μs(3) |
| RC(5) | `11` | 2-6 μs(1,4) | 2-6 μs(1,4) | 2-6 μs(1,4) | 2-6 μs(1) |

**Legend:** Shaded cells are outside of recommended range.

**Note 1:** The RC source has a typical $T_{AD}$ time of 4 μs.

**2:** These values violate the minimum required $T_{AD}$ time.

**3:** For faster conversion times, the selection of another clock source is recommended.

**4:** When device frequency is greater than 1 MHz, the RC A/D conversion clock source is recommended for Sleep operation only.

**5:** For extended voltage devices (LC), please refer to Electrical Specifications section.

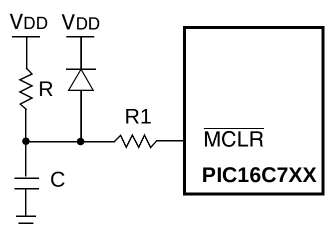
 

## 9.4 Power-On Reset (POR)

A Power-on Reset pulse is generated on-chip when $V_{DD}$ rise is detected (to a level of 1.5V-2.1V). To take advantage of the POR, just tie the $\overline{MCLR}$ pin directly (or through a resistor) to $V_{DD}$. This will eliminate external RC components usually needed to create a Power-on Reset. A maximum rise time for $V_{DD}$ is specified (parameter D004). For a slow rise time, see Figure 9-5.

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. Brown-out Reset may be used to meet the start-up conditions.

**FIGURE 9-5: EXTERNAL POWER-ON RESET CIRCUIT (FOR SLOW $V_{DD}$ POWER-UP)**




**Note 1:** External Power-on Reset circuit is required only if $V_{DD}$ power-up slope is too slow. The diode D helps discharge the capacitor quickly when $V_{DD}$ powers down.

**2:** R < 40 kΩ is recommended to make sure that voltage drop across R does not violate the device's electrical specification.

**3:** R1 = 100Ω to 1 kΩ will limit any current flowing into $\overline{MCLR}$ from external capacitor C in the event of $\overline{MCLR}$/$V_{PP}$ pin breakdown due to Electrostatic Discharge (ESD) or Electrical Overstress (EOS).

## 9.5 Power-up Timer (PWRT)

The Power-up Timer provides a fixed nominal time-out (parameter #33), 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 $V_{DD}$ 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 $V_{DD}$, temperature, and process variation. See DC parameters for details.

## 9.6 Oscillator Start-up Timer (OST)

The Oscillator Start-up Timer (OST) provides a 1024 oscillator cycle (from OSC1 input) delay after the PWRT delay is over (parameter #32). This ensures 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.

## 9.7 Brown-Out Reset (BOR)

The PIC16C712/716 members have on-chip Brown-out Reset circuitry. A Configuration bit, BODEN, can disable (if clear/programmed) or enable (if set) the Brown-out Reset circuitry. If $V_{DD}$ falls below 4.0V, refer to $V_{BOR}$ parameter D005($V_{BOR}$) for a time greater than parameter ($T_{BOR}$) in Table 12-6. The brown-out situation will reset the chip. A Reset is not guaranteed to occur if $V_{DD}$ falls below 4.0V for less than parameter ($T_{BOR}$).

On any Reset (Power-on, Brown-out, Watchdog, etc.) the chip will remain in Reset until $V_{DD}$ rises above $V_{BOR}$. The Power-up Timer will now be invoked and will keep the chip in Reset an additional 72 ms.

If $V_{DD}$ drops below $V_{BOR}$ while the Power-up Timer is running, the chip will go back into a Brown-out Reset and the Power-up Timer will be re-initialized. Once $V_{DD}$ rises above $V_{BOR}$, the Power-Up Timer will execute a 72 ms Reset. The Power-up Timer should always be enabled when Brown-out Reset is enabled. Figure 9-7 shows typical Brown-out situations.

For operations where the desired brown-out voltage is other than 4V, an external brown-out circuit must be used. Figure 9-8, 9-9 and 9-10 show examples of external brown-out protection circuits.

### 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
```

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 is 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.

### 9.13.2 WAKE-UP USING INTERRUPTS

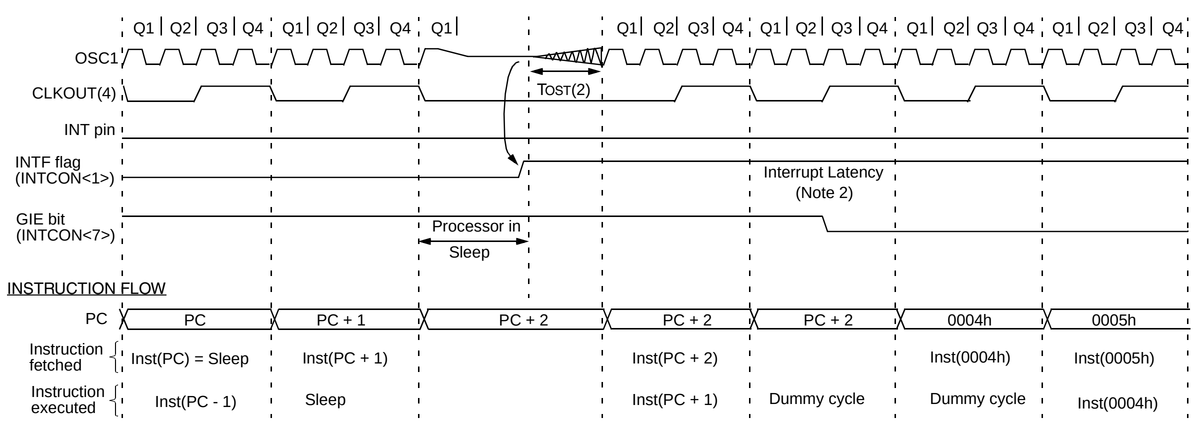
When global interrupts are disabled (GIE cleared) and any interrupt source has both its interrupt enable bit and interrupt flag bit set, one of the following will occur:

- 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 $\overline{\text{TO}}$ bit will not be set and $\overline{\text{PD}}$ 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 $\overline{\text{TO}}$ bit will be set and the $\overline{\text{PD}}$ 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 $\overline{\text{PD}}$ bit. If the $\overline{\text{PD}}$ 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.

**FIGURE 9-17: WAKE-UP FROM SLEEP THROUGH INTERRUPT**




**Note 1:** XT, HS or LP Oscillator mode assumed.
**2:** $T_{OST}$ = 1024$T_{OSC}$ (drawing not to scale) This delay will not be there for RC Osc mode.
**3:** GIE = 1 assumed. In this case after wake-up, the processor jumps to the interrupt routine. If GIE = 0, execution will continue in-line.
**4:** CLKOUT is not available in these osc modes, but shown here for timing reference.

## 9.14 Program Verification/Code Protection

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

| **Note:** | Microchip does not recommend code protecting windowed devices. |
|---|---|

## 9.15 ID Locations

Four memory locations (2000h-2003h) are designated as ID locations where the user can store checksum or other code-identification numbers. These locations are not accessible during normal execution, but are readable and writable during Program/Verify. It is recommended that only the 4 Least Significant bits of the ID location are used.

For ROM devices, these values are submitted along with the ROM code.

## 11.2 MPASM Assembler

The MPASM Assembler is a full-featured, universal macro assembler for all PIC MCUs.

The MPASM Assembler generates relocatable object files for the MPLINK Object Linker, Intel® standard HEX files, MAP files to detail memory usage and symbol reference, absolute LST files that contain source lines and generated machine code and COFF files for debugging.

The MPASM Assembler features include:

- Integration into MPLAB IDE projects
- User-defined macros to streamline assembly code
- Conditional assembly for multi-purpose source files
- Directives that allow complete control over the assembly process

## 11.3 MPLAB C18 and MPLAB C30 C Compilers

The MPLAB C18 and MPLAB C30 Code Development Systems are complete ANSI C compilers for Microchip's PIC18 family of microcontrollers and dsPIC30F family of digital signal controllers. These compilers provide powerful integration capabilities, superior code optimization and ease of use not found with other compilers.

For easy source level debugging, the compilers provide symbol information that is optimized to the MPLAB IDE debugger.

## 11.4 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK Object Linker combines relocatable objects created by the MPASM Assembler and the MPLAB C18 C Compiler. It can link relocatable objects from precompiled libraries, using directives from a linker script.

The MPLIB Object Librarian manages the creation and modification of library files of precompiled code. When a routine from a library is called from a source file, only the modules that contain that routine will be linked in with the application. This allows large libraries to be used efficiently in many different applications.

The object linker/library features include:

- Efficient linking of single libraries instead of many smaller files
- Enhanced code maintainability by grouping related modules together
- Flexible creation of libraries with easy module listing, replacement, deletion and extraction

## 11.5 MPLAB ASM30 Assembler, Linker and Librarian

MPLAB ASM30 Assembler produces relocatable machine code from symbolic assembly language for dsPIC30F devices. MPLAB C30 C Compiler uses the assembler to produce its object file. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. Notable features of the assembler include:

- Support for the entire dsPIC30F instruction set
- Support for fixed-point and floating-point data
- Command line interface
- Rich directive set
- Flexible macro language
- MPLAB IDE compatibility

## 11.6 MPLAB SIM Software Simulator

The MPLAB SIM Software Simulator allows code development in a PC-hosted environment by simulating the PIC MCUs and dsPIC® DSCs on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a comprehensive stimulus controller. Registers can be logged to files for further run-time analysis. The trace buffer and logic analyzer display extend the power of the simulator to record and track program execution, actions on I/O, as well as internal registers.

The MPLAB SIM Software Simulator fully supports symbolic debugging using the MPLAB C18 and MPLAB C30 C Compilers, and the MPASM and MPLAB ASM30 Assemblers. The software simulator offers the flexibility to develop and debug code outside of the laboratory environment, making it an excellent, economical software development tool.

## 12.2 DC Characteristics: PIC16LC712/716-04 (Commercial, Industrial)

| 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 | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Sym.** | **Characteristic** | **Min.** | **Typ†** | **Max.** | **Units** | **Conditions** |
| D001 | VDD | **Supply Voltage** | 2.5<br>VBOR* | —<br>— | 5.5<br>5.5 | V<br>V | <br>BOR enabled **(Note 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><br>D010A | IDD | **Supply Current**(2,5) | —<br><br>— | 2.0<br><br>22.5 | 3.8<br><br>48 | mA<br><br>µA | XT, RC osc modes<br>FOSC = 4 MHz, VDD = 3.0V **(Note 4)**<br>LP osc mode<br>FOSC = 32 kHz, VDD = 3.0V, WDT disabled |
| D020<br>D021<br>D021A | IPD | **Power-down Current**(3,5) | —<br>—<br>— | 7.5<br>0.9<br>0.9 | 30<br>5<br>5 | µA<br>µA<br>µA | VDD = 3.0V, WDT enabled, -40°C to +85°C<br>VDD = 3.0V, WDT disabled, 0°C to +70°C<br>VDD = 3.0V, WDT disabled, -40°C to +85°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.

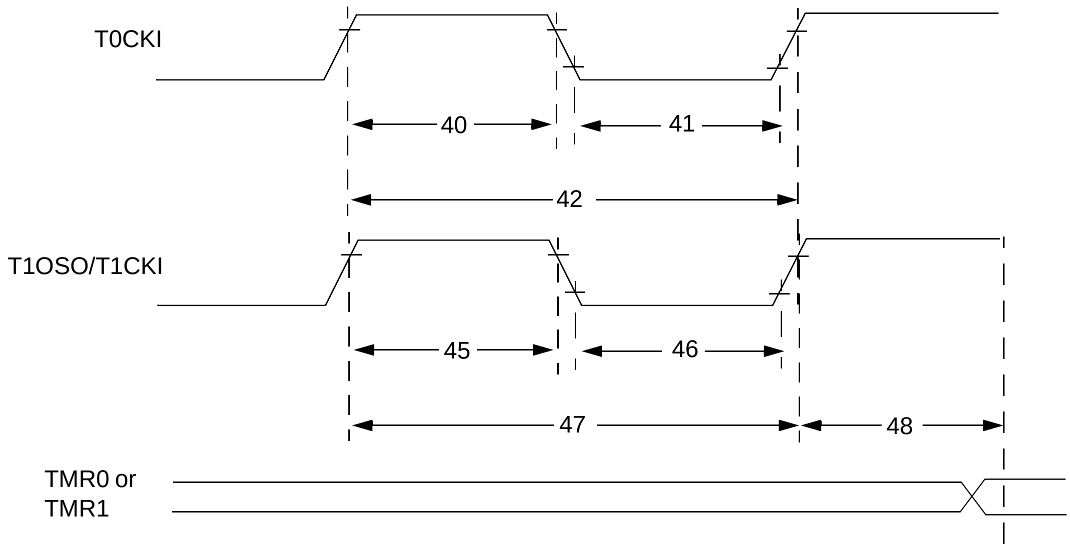
**4:** For RC Osc mode, current through REXT is not included. The current through the resistor can be estimated by the formula Ir = VDD/2REXT (mA) with REXT in kOhm.

**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.

**FIGURE 12-8: TIMER0 AND TIMER1 EXTERNAL CLOCK TIMINGS**




**Note:** Refer to Figure 12-3 for load conditions.

**TABLE 12-5: TIMER0 AND TIMER1 EXTERNAL CLOCK REQUIREMENTS**

| Param No. | Sym. | Characteristic | | | Min. | Typ† | Max. | Units | Conditions |
|---|---|---|---|---|---|---|---|---|---|
| 40* | Tt0H | T0CKI High Pulse Width | | No Prescaler | 0.5$T_{CY}$ + 20 | — | — | ns | Must also meet parameter 42 |
| | | | | With Prescaler | 10 | — | — | ns | |
| 41* | Tt0L | T0CKI Low Pulse Width | | No Prescaler | 0.5$T_{CY}$ + 20 | — | — | ns | Must also meet parameter 42 |
| | | | | With Prescaler | 10 | — | — | ns | |
| 42* | Tt0P | T0CKI Period | | No Prescaler | $T_{CY}$ + 40 | — | — | ns | |
| | | | | With Prescaler | Greater of: 20 or $\frac{T_{CY} + 40}{N}$ | — | — | ns | N = prescale value (2, 4,..., 256) |
| 45* | Tt1H | T1CKI High Time | Synchronous, Prescaler = 1 | | 0.5$T_{CY}$ + 20 | — | — | ns | Must also meet parameter 47 |
| | | | Synchronous, Prescaler = 2,4,8 | Standard | 15 | — | — | ns | |
| | | | | Extended (LC) | 25 | — | — | ns | |
| | | | Asynchronous | Standard | 30 | — | — | ns | |
| | | | | Extended (LC) | 50 | — | — | ns | |
| 46* | Tt1L | T1CKI Low Time | Synchronous, Prescaler = 1 | | 0.5$T_{CY}$ + 20 | — | — | ns | Must also meet parameter 47 |
| | | | Synchronous, Prescaler = 2,4,8 | Standard | 15 | — | — | ns | |
| | | | | Extended (LC) | 25 | — | — | ns | |
| | | | Asynchronous | Standard | 30 | — | — | ns | |
| | | | | Extended (LC) | 50 | — | — | ns | |
| 47* | Tt1P | T1CKI input period | Synchronous | Standard | Greater of: 30 OR $\frac{T_{CY} + 40}{N}$ | — | — | ns | N = prescale value (1, 2, 4, 8) |
| | | | | Extended (LC) | Greater of: 50 OR $\frac{T_{CY} + 40}{N}$ | | | | N = prescale value (1, 2, 4, 8) |
| | | | Asynchronous | Standard | 60 | — | — | ns | |
| | | | | Extended (LC) | 100 | — | — | ns | |
| | Ft1 | Timer1 oscillator input frequency range (oscillator enabled by setting bit T1OSCEN) | | | DC | — | 200 | kHz | |
| 48 | TCKEZtmr1 | Delay from external clock edge to timer increment | | | 2Tosc | — | 7Tosc | — | |

* 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.

**NOTES:**