# PIC16F685/687/689/690
# Data Sheet

## 20-Pin Flash-Based, 8-Bit CMOS Microcontrollers with nanoWatt Technology

**Note the following details of the code protection feature on Microchip devices:**

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.



**Trademarks**



QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
ISO/TS 16949:2002

*Microchip received ISO/TS-16949:2002 quality system certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona and Mountain View, California in October 2003. The Company's quality system processes and procedures are for its PICmicro® 8-bit MCUs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.*

# PIC16F685/687/689/690

## 20-Pin Flash-Based, 8-Bit CMOS Microcontrollers with nanoWatt Technology

### High-Performance RISC CPU:

- Only 35 instructions to learn:
  - All single-cycle instructions except branches
- Operating speed:
  - DC – 20 MHz oscillator/clock input
  - DC – 200 ns instruction cycle
- Interrupt capability
- 8-level deep hardware stack
- Direct, Indirect and Relative Addressing modes

### Special Microcontroller Features:

- Precision Internal Oscillator:
  - Factory calibrated to ± 1%
  - Software selectable frequency range of 8 MHz to 32 kHz
  - Software tunable
  - Two-Speed Start-up mode
  - Crystal fail detect for critical applications
  - Clock mode switching during operation for power savings
- Power-saving Sleep mode
- Wide operating voltage range (2.0V-5.5V)
- Industrial and Extended Temperature range
- Power-on Reset (POR)
- Power-up Timer (PWRTE) and Oscillator Start-up Timer (OST)
- Brown-out Reset (BOR) with software control option
- Enhanced low-current Watchdog Timer (WDT) with on-chip oscillator (software selectable nominal 268 seconds with full prescaler) with software enable
- Multiplexed Master Clear/Input pin
- Programmable code protection
- High Endurance Flash/EEPROM cell:
  - 100,000 write Flash endurance
  - 1,000,000 write EEPROM endurance
  - Flash/Data EEPROM retention: > 40 years
- Enhanced USART Module:
  - Supports RS-485, RS-232, and LIN 2.0
  - Auto-Baud Detect
  - Auto-wake-up on Start bit

### Low-Power Features:

- **Standby Current:**
  - 1 nA @ 2.0V, typical
- Operating Current:
  - 20 μA @ 32 kHz, 2.0V, typical
  - <1 mA @ 4 MHz, 5.5V, typical
- Watchdog Timer Current:
  - <1 μA @ 2.0V, typical

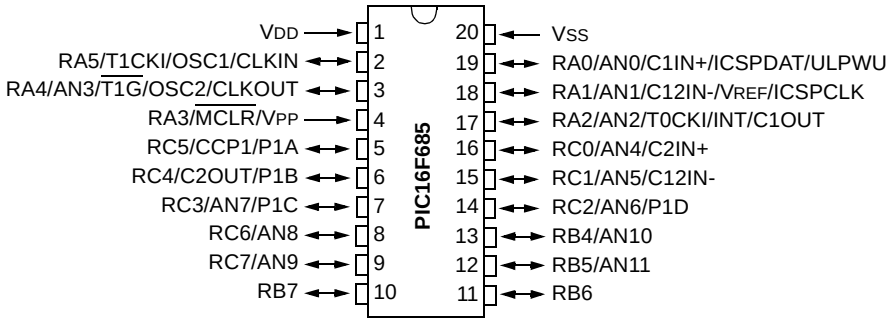
### Peripheral Features:

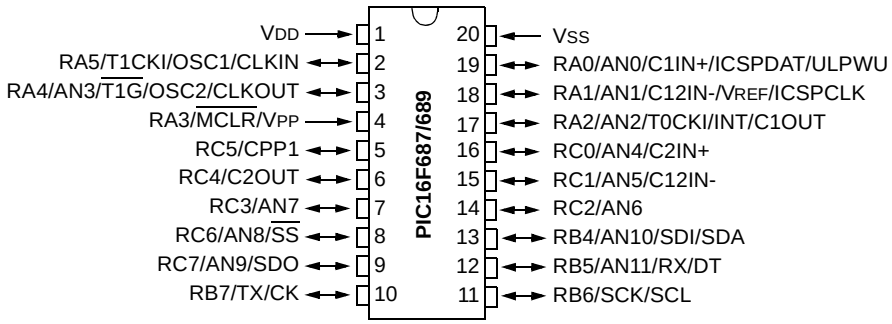
- 17 I/O pins and 1 input only pin:
  - High current source/sink for direct LED drive
  - Interrupt-on-pin change
  - Individually programmable weak pull-ups
  - Ultra Low-Power Wake-up (ULPWU)
- Analog comparator module with:
  - Two analog comparators
  - Programmable on-chip voltage reference ($CV_{REF}$) module (% of $V_{DD}$)
  - Comparator inputs and outputs externally accessible
  - SR Latch mode
  - Timer 1 Gate Sync Latch
- A/D Converter:
  - 10-bit resolution and 12 channels
- Timer0: 8-bit timer/counter with 8-bit programmable prescaler
- Enhanced Timer1:
  - 16-bit timer/counter with prescaler
  - External Gate Input mode
  - Option to use OSC1 and OSC2 in LP mode as Timer1 oscillator if INTOSC mode selected
- Timer2: 8-bit timer/counter with 8-bit period register, prescaler and postscaler
- Enhanced Capture, Compare, PWM+ module:
  - 16-bit Capture, max resolution 12.5 ns
  - Compare, max resolution 200 ns
  - 10-bit PWM with 1, 2 or 4 output channels, programmable “dead time”, max frequency 20 kHz
  - PWM output steering control
- Synchronous Serial Port (SSP):
  - SPI™ mode (Master and Slave)
- $I^2C$™ (Master/Slave modes):
  - $I^2C$™ address mask
- In-Circuit Serial Programming™ (ICSP™) via two pins

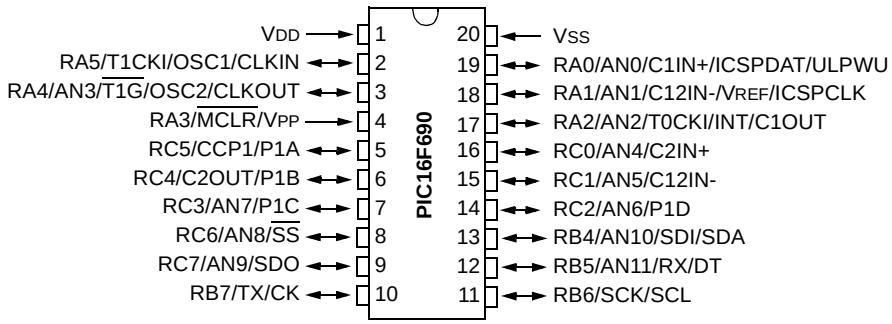
| Device | Program Memory | Data Memory | | I/O | 10-bit A/D (ch) | Comparators | Timers 8/16-bit | SSP | ECCP+ | EUSART |
|---|---|---|---|---|---|---|---|---|---|---|
| | Flash (words) | SRAM (bytes) | EEPROM (bytes) | | | | | | | |
| PIC16F685 | 4096 | 256 | 256 | 18 | 12 | 2 | 2/1 | No | Yes | No |
| PIC16F687 | 2048 | 128 | 256 | 18 | 12 | 2 | 1/1 | Yes | No | Yes |
| PIC16F689 | 4096 | 256 | 256 | 18 | 12 | 2 | 1/1 | Yes | No | Yes |
| PIC16F690 | 4096 | 256 | 256 | 18 | 12 | 2 | 2/1 | Yes | Yes | Yes |

## Pin Diagrams

**20-pin PDIP, SOIC, SSOP**



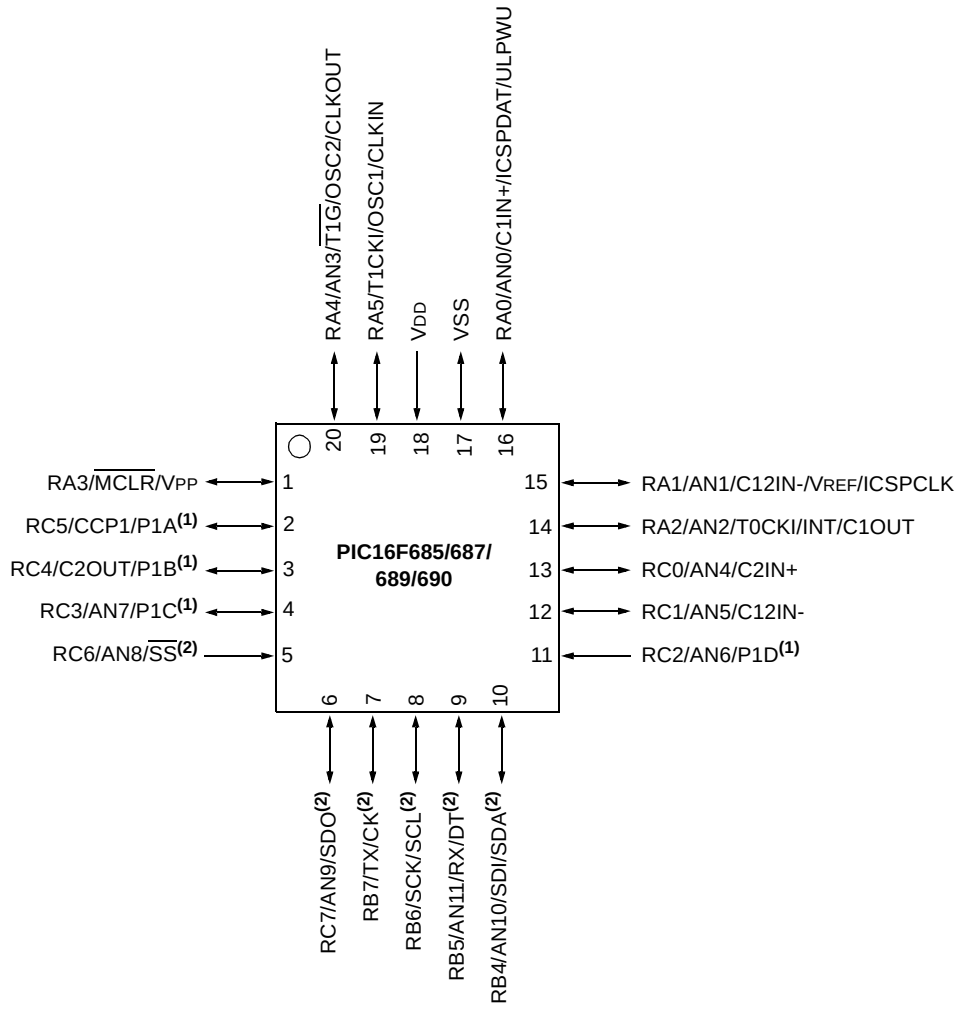



## Pin Diagrams (Continued)

**20-pin QFN**




**Note 1:** P1A, P1B, P1C and P1D are available on PIC16F685/PIC16F690 only.

**2:** $\overline{SS}$, SDO, SDA, RX and DT available on PIC16F687/PIC16F689/PIC16F690 only.

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

The PIC16F685/687/689/690 devices are covered by this data sheet. They are available in 20-pin PDIP, SOIC, TSSOP and QFN packages.

Block Diagrams and pinout descriptions of the devices are as follows:

- PIC16F685 (Figure 1-1, Table 1-1)
- PIC16F687/PIC16F689 (Figure 1-2, Table 1-2)
- PIC16F690 (Figure 1-3, Table 1-3)

**FIGURE 1-1: PIC16F685 BLOCK DIAGRAM**

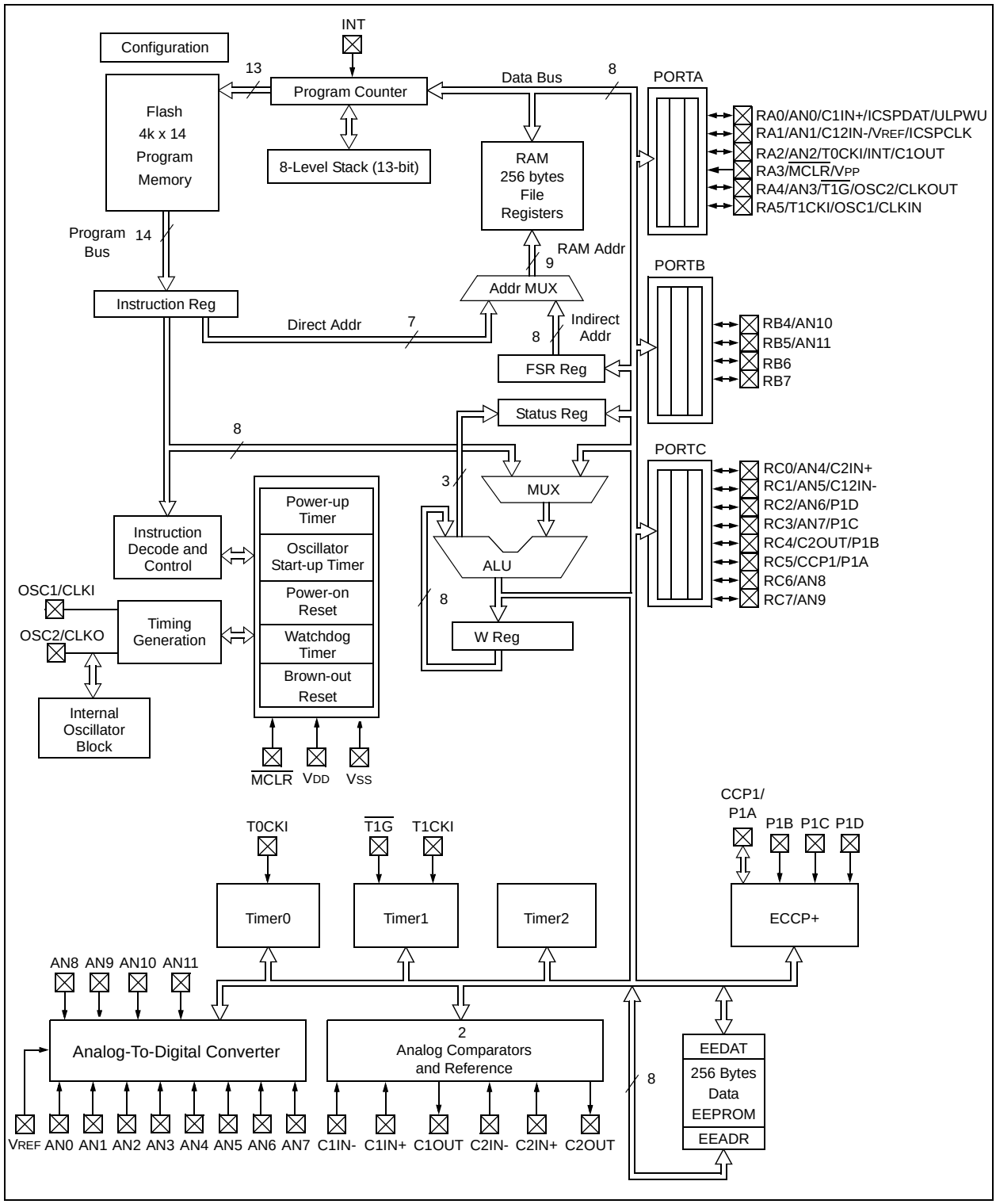
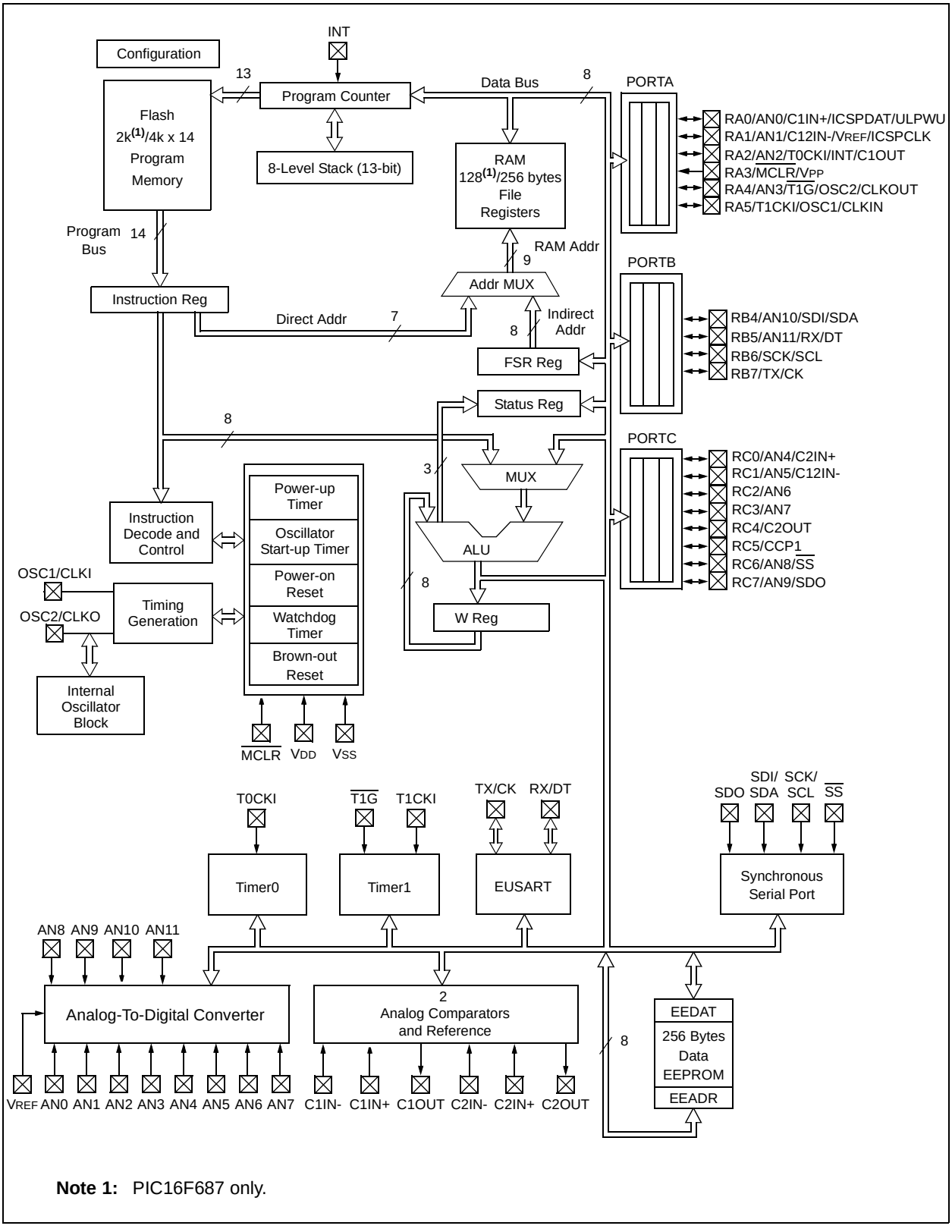


**FIGURE 1-2: PIC16F687/PIC16F689 BLOCK DIAGRAM**




**Note 1:** PIC16F687 only.

**FIGURE 1-3: PIC16F690 BLOCK DIAGRAM**

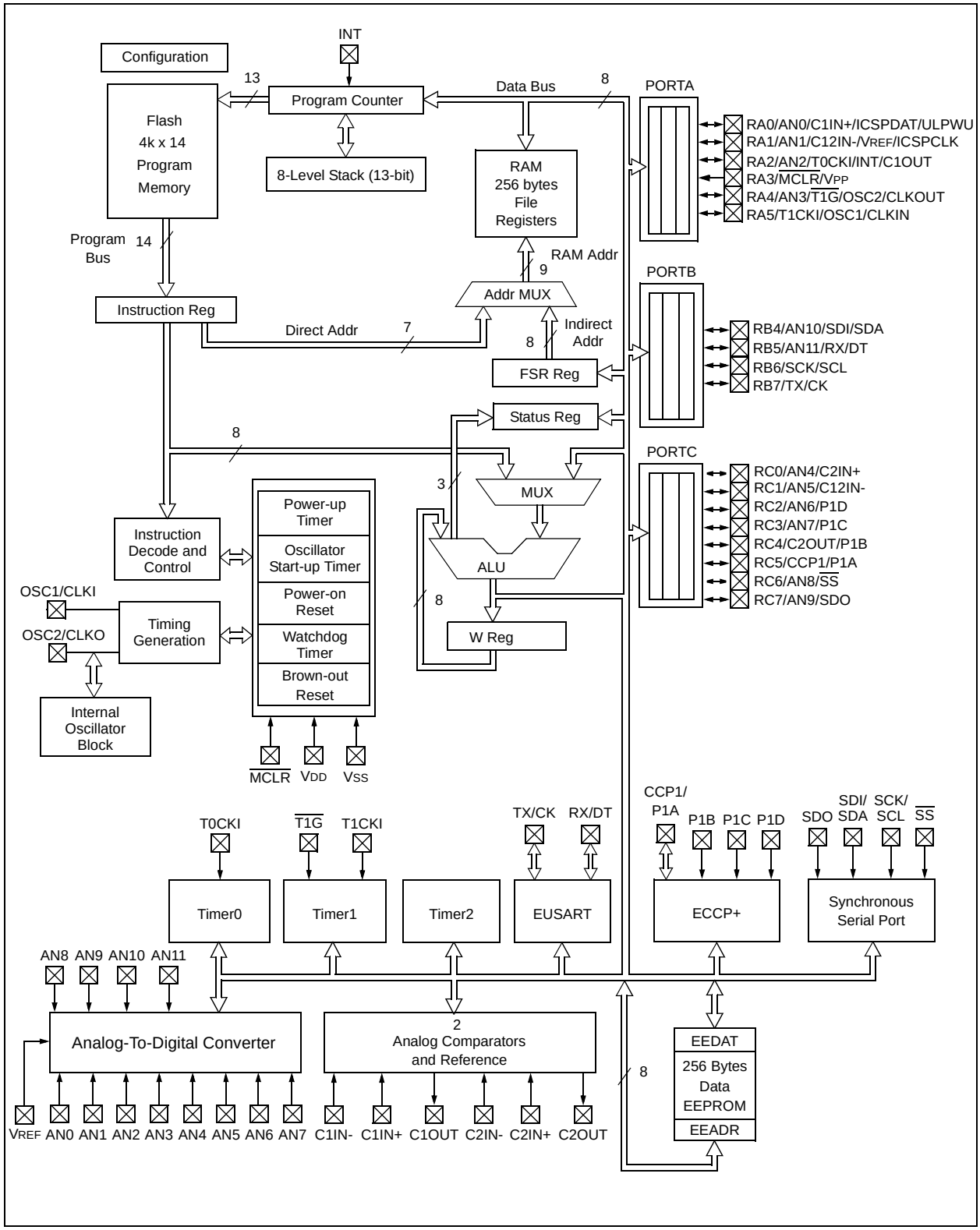


**TABLE 1-1: PINOUT DESCRIPTION – PIC16F685**

| Name | Function | Input Type | Output Type | Description |
|---|---|---|---|---|
| RA0/AN0/C1IN+/ICSPDAT/ ULPWU | RA0 | TTL | — | General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up. |
| | AN0 | AN | — | A/D Channel 0 input. |
| | C1IN+ | AN | — | Comparator 1 positive input. |
| | ICSPDAT | TTL | CMOS | ICSP™ Data I/O. |
| | ULPWU | AN | — | Ultra Low-Power Wake-up input. |
| RA1/AN1/C12IN-/VREF/ICSPCLK | RA1 | TTL | CMOS | General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up. |
| | AN1 | AN | — | A/D Channel 1 input. |
| | C12IN- | AN | — | Comparator 1 or 2 negative input. |
| | VREF | AN | — | External Voltage Reference for A/D. |
| | ICSPCLK | ST | — | ICSP™ clock. |
| RA2/AN2/T0CKI/INT/C1OUT | RA2 | ST | CMOS | General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up. |
| | AN2 | AN | — | A/D Channel 2 input. |
| | T0CKI | ST | — | Timer0 clock input. |
| | INT | ST | — | External interrupt pin. |
| | C1OUT | — | CMOS | Comparator 1 output. |
| RA3/$\overline{\text{MCLR}}$/VPP | RA3 | TTL | — | General purpose input. Individually controlled interrupt-on-change. |
| | $\overline{\text{MCLR}}$ | ST | — | Master Clear with internal pull-up. |
| | VPP | HV | — | Programming voltage. |
| RA4/AN3/$\overline{\text{T1G}}$/OSC2/CLKOUT | RA4 | TTL | CMOS | General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up. |
| | AN3 | AN | — | A/D Channel 3 input. |
| | $\overline{\text{T1G}}$ | ST | — | Timer1 gate input. |
| | OSC2 | — | XTAL | Crystal/Resonator. |
| | CLKOUT | — | CMOS | FOSC/4 output. |
| RA5/T1CKI/OSC1/CLKIN | RA5 | TTL | CMOS | General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up. |
| | T1CKI | ST | — | Timer1 clock input. |
| | OSC1 | XTAL | — | Crystal/Resonator. |
| | CLKIN | ST | — | External clock input/RC oscillator connection. |
| RB4/AN10 | RB4 | TTL | CMOS | General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up. |
| | AN10 | AN | — | A/D Channel 10 input. |
| RB5/AN11 | RB5 | TTL | CMOS | General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up. |
| | AN11 | AN | — | A/D Channel 11 input. |
| RB6 | RB6 | TTL | CMOS | General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up. |
| RB7 | RB7 | TTL | CMOS | General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up. |
| RC0/AN4/C2IN+ | RC0 | ST | CMOS | General purpose I/O. |
| | AN4 | AN | — | A/D Channel 4 input. |
| | C2IN+ | AN | — | Comparator 2 positive input. |

**Legend:** AN = Analog input or output; CMOS = CMOS compatible input or output; TTL = TTL compatible input; ST = Schmitt Trigger input with CMOS levels; HV = High Voltage; XTAL = Crystal

**TABLE 1-1: PINOUT DESCRIPTION – PIC16F685 (CONTINUED)**

| Name | Function | Input Type | Output Type | Description |
|---|---|---|---|---|
| RC1/AN5/C12IN- | RC1 | ST | CMOS | General purpose I/O. |
| | AN5 | AN | — | A/D Channel 5 input. |
| | C12IN- | AN | — | Comparator 1 or 2 negative input. |
| RC2/AN6/P1D | RC2 | ST | CMOS | General purpose I/O. |
| | AN6 | AN | — | A/D Channel 6 input. |
| | P1D | — | CMOS | PWM output. |
| RC3/AN7/P1C | RC3 | ST | CMOS | General purpose I/O. |
| | AN7 | AN | — | A/D Channel 7 input. |
| | P1C | — | CMOS | PWM output. |
| RC4/C2OUT/P1B | RC4 | ST | CMOS | General purpose I/O. |
| | C2OUT | — | CMOS | Comparator 2 output. |
| | P1B | — | CMOS | PWM output. |
| RC5/CCP1/P1A | RC5 | ST | CMOS | General purpose I/O. |
| | CCP1 | ST | CMOS | Capture/Compare input. |
| | P1A | ST | CMOS | PWM output. |
| RC6/AN8 | RC6 | ST | CMOS | General purpose I/O. |
| | AN8 | AN | — | A/D Channel 8 input. |
| RC7/AN9 | RC7 | ST | CMOS | General purpose I/O. |
| | AN9 | AN | — | A/D Channel 9 input. |
| VSS | VSS | Power | — | Ground reference. |
| VDD | VDD | Power | — | Positive supply. |

**Legend:** AN = Analog input or output; CMOS = CMOS compatible input or output; TTL = TTL compatible input; ST = Schmitt Trigger input with CMOS levels; HV = High Voltage; XTAL = Crystal

**TABLE 1-2: PINOUT DESCRIPTION – PIC16F687/PIC16F689**

| Name | Function | Input Type | Output Type | Description |
|---|---|---|---|---|
| RA0/AN0/C1IN+/ICSPDAT/ ULPWU | RA0 | TTL | — | General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up. |
| | AN0 | AN | — | A/D Channel 0 input. |
| | C1IN+ | AN | — | Comparator 1 positive input. |
| | ICSPDAT | TTL | CMOS | ICSP Data I/O. |
| | ULPWU | AN | — | Ultra Low-Power Wake-up input. |
| RA1/AN1/C12IN-/VREF/ICSPCLK | RA1 | TTL | CMOS | General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up. |
| | AN1 | AN | — | A/D Channel 1 input. |
| | C12IN- | AN | — | Comparator 1 or 2 negative input. |
| | VREF | AN | — | External Voltage Reference for A/D. |
| | ICSPCLK | ST | — | ICSP™ clock. |
| RA2/AN2/T0CKI/INT/C1OUT | RA2 | ST | CMOS | General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up. |
| | AN2 | AN | — | A/D Channel 2 input. |
| | T0CKI | ST | — | Timer0 clock input. |
| | INT | ST | — | External Interrupt. |
| | C1OUT | — | CMOS | Comparator 1 output. |
| RA3/$\overline{\text{MCLR}}$/VPP | RA3 | TTL | — | General purpose input. Individually controlled interrupt-on-change. |
| | $\overline{\text{MCLR}}$ | ST | — | Master Clear with internal pull-up. |
| | VPP | HV | — | Programming voltage. |
| RA4/AN3/$\overline{\text{T1G}}$/OSC2/CLKOUT | RA4 | TTL | CMOS | General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up. |
| | AN3 | AN | — | A/D Channel 3 input. |
| | $\overline{\text{T1G}}$ | ST | — | Timer1 gate input. |
| | OSC2 | — | XTAL | Crystal/Resonator. |
| | CLKOUT | — | CMOS | FOSC/4 output. |
| RA5/T1CKI/OSC1/CLKIN | RA5 | TTL | CMOS | General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up. |
| | T1CKI | ST | — | Timer1 clock input. |
| | OSC1 | XTAL | — | Crystal/Resonator. |
| | CLKIN | ST | — | External clock input/RC oscillator connection. |
| RB4/AN10/SDI/SDA | RB4 | TTL | CMOS | General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up. |
| | AN10 | AN | — | A/D Channel 10 input. |
| | SDI | ST | — | SPI™ data input. |
| | SDA | ST | OD | $I^2C$ data input/output. |
| RB5/AN11/RX/DT | RB5 | TTL | CMOS | General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up. |
| | AN11 | AN | — | A/D Channel 11 input. |
| | RX | ST | — | EUSART asynchronous input. |
| | DT | ST | CMOS | EUSART synchronous data. |

**Legend:** AN = Analog input or output CMOS = CMOS compatible input or output OD = Open Drain
TTL = TTL compatible input ST = Schmitt Trigger input with CMOS levels
HV = High Voltage XTAL = Crystal

**TABLE 1-2: PINOUT DESCRIPTION – PIC16F687/PIC16F689 (CONTINUED)**

| Name | Function | Input Type | Output Type | Description |
|---|---|---|---|---|
| RB6/SCK/SCL | RB6 | TTL | CMOS | General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up. |
| | SCK | ST | CMOS | SPI™ clock. |
| | SCL | ST | OD | $I^2C$™ clock. |
| RB7/TX/CK | RB7 | TTL | CMOS | General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up. |
| | TX | — | CMOS | EUSART asynchronous output. |
| | CK | ST | CMOS | EUSART synchronous clock. |
| RC0/AN4/C2IN+ | RC0 | ST | CMOS | General purpose I/O. |
| | AN4 | AN | — | A/D Channel 4 input. |
| | C2IN+ | AN | — | Comparator 2 positive input. |
| RC1/AN5/C12IN- | RC1 | ST | CMOS | General purpose I/O. |
| | AN5 | AN | — | A/D Channel 5 input. |
| | C12IN- | AN | — | Comparator 1 or 2 negative input. |
| RC2/AN6 | RC2 | ST | CMOS | General purpose I/O. |
| | AN6 | AN | — | A/D Channel 6 input. |
| RC3/AN7 | RC3 | ST | CMOS | General purpose I/O. |
| | AN7 | AN | — | A/D Channel 7 input. |
| RC4/C2OUT | RC4 | ST | CMOS | General purpose I/O. |
| | C2OUT | — | CMOS | Comparator 2 output. |
| RC5/CCP1 | RC5 | ST | CMOS | General purpose I/O. |
| | CCP1 | ST | CMOS | Capture/Compare input. |
| RC6/AN8/$\overline{SS}$ | RC6 | ST | CMOS | General purpose I/O. |
| | AN8 | AN | — | A/D Channel 8 input. |
| | $\overline{SS}$ | ST | — | Slave Select input. |
| RC7/AN9/SDO | RC7 | ST | CMOS | General purpose I/O. |
| | AN9 | AN | — | A/D Channel 9 input. |
| | SDO | — | CMOS | SPI data output. |
| VSS | VSS | Power | — | Ground reference. |
| VDD | VDD | Power | — | Positive supply. |

**Legend:** AN = Analog input or output CMOS = CMOS compatible input or output OD = Open Drain
TTL = TTL compatible input ST = Schmitt Trigger input with CMOS levels
HV = High Voltage XTAL = Crystal

**TABLE 1-3: PINOUT DESCRIPTION – PIC16F690**

| Name | Function | Input Type | Output Type | Description |
|---|---|---|---|---|
| RA0/AN0/C1IN+/ICSPDAT/ ULPWU | RA0 | TTL | — | General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up. |
| | AN0 | AN | — | A/D Channel 0 input. |
| | C1IN+ | AN | — | Comparator 1 positive input. |
| | ICSPDAT | TTL | CMOS | ICSP Data I/O. |
| | ULPWU | AN | — | Ultra Low-Power Wake-up input. |
| RA1/AN1/C12IN-/VREF/ICSPCLK | RA1 | TTL | CMOS | General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up. |
| | AN1 | AN | — | A/D Channel 1 input. |
| | C12IN- | AN | — | Comparator 1 or 2 negative input. |
| | VREF | AN | — | External Voltage Reference for A/D. |
| | ICSPCLK | ST | — | ICSP™ clock. |
| RA2/AN2/T0CKI/INT/C1OUT | RA2 | ST | CMOS | General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up. |
| | AN2 | AN | — | A/D Channel 2 input. |
| | T0CKI | ST | — | Timer0 clock input. |
| | INT | ST | — | External Interrupt. |
| | C1OUT | — | CMOS | Comparator 1 output. |
| RA3/$\overline{\text{MCLR}}$/VPP | RA3 | TTL | — | General purpose input. Individually controlled interrupt-on-change. |
| | $\overline{\text{MCLR}}$ | ST | — | Master Clear with internal pull-up. |
| | VPP | HV | — | Programming voltage. |
| RA4/AN3/$\overline{\text{T1G}}$/OSC2/CLKOUT | RA4 | TTL | CMOS | General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up. |
| | AN3 | AN | — | A/D Channel 3 input. |
| | $\overline{\text{T1G}}$ | ST | — | Timer1 gate input. |
| | OSC2 | — | XTAL | Crystal/Resonator. |
| | CLKOUT | — | CMOS | FOSC/4 output. |
| RA5/T1CKI/OSC1/CLKIN | RA5 | TTL | CMOS | General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up. |
| | T1CKI | ST | — | Timer1 clock input. |
| | OSC1 | XTAL | — | Crystal/Resonator. |
| | CLKIN | ST | — | External clock input/RC oscillator connection. |
| RB4/AN10/SDI/SDA | RB4 | TTL | CMOS | General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up. |
| | AN10 | AN | — | A/D Channel 10 input. |
| | SDI | ST | — | SPI data input. |
| | SDA | ST | OD | $I^2C$ data input/output. |
| RB5/AN11/RX/DT | RB5 | TTL | CMOS | General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up. |
| | AN11 | AN | — | A/D Channel 11 input. |
| | RX | ST | — | EUSART asynchronous input. |
| | DT | ST | CMOS | EUSART synchronous data. |

**Legend:** AN = Analog input or output CMOS = CMOS compatible input or output OD = Open Drain
TTL = TTL compatible input ST = Schmitt Trigger input with CMOS levels
HV = High Voltage XTAL = Crystal

**TABLE 1-3: PINOUT DESCRIPTION – PIC16F690 (CONTINUED)**

| Name | Function | Input Type | Output Type | Description |
|---|---|---|---|---|
| RB6/SCK/SCL | RB6 | TTL | CMOS | General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up. |
| | SCK | ST | CMOS | SPI™ clock. |
| | SCL | ST | OD | $I^2C$™ clock. |
| RB7/TX/CK | RB7 | TTL | CMOS | General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up. |
| | TX | — | CMOS | EUSART asynchronous output. |
| | CK | ST | CMOS | EUSART synchronous clock. |
| RC0/AN4/C2IN+ | RC0 | ST | CMOS | General purpose I/O. |
| | AN4 | AN | — | A/D Channel 4 input. |
| | C2IN+ | AN | — | Comparator 2 positive input. |
| RC1/AN5/C12IN- | RC1 | ST | CMOS | General purpose I/O. |
| | AN5 | AN | — | A/D Channel 5 input. |
| | C12IN- | AN | — | Comparator 1 or 2 negative input. |
| RC2/AN6/P1D | RC2 | ST | CMOS | General purpose I/O. |
| | AN6 | AN | — | A/D Channel 6 input. |
| | P1D | — | CMOS | PWM output. |
| RC3/AN7/P1C | RC3 | ST | CMOS | General purpose I/O. |
| | AN7 | AN | — | A/D Channel 7 input. |
| | P1C | — | CMOS | PWM output. |
| RC4/C2OUT/P1B | RC4 | ST | CMOS | General purpose I/O. |
| | C2OUT | — | CMOS | Comparator 2 output. |
| | P1B | — | CMOS | PWM output. |
| RC5/CCP1/P1A | RC5 | ST | CMOS | General purpose I/O. |
| | CCP1 | ST | CMOS | Capture/Compare input. |
| | P1A | ST | CMOS | PWM output. |
| RC6/AN8/$\overline{SS}$ | RC6 | ST | CMOS | General purpose I/O. |
| | AN8 | AN | — | A/D Channel 8 input. |
| | $\overline{SS}$ | ST | — | Slave Select input. |
| RC7/AN9/SDO | RC7 | ST | CMOS | General purpose I/O. |
| | AN9 | AN | — | A/D Channel 9 input. |
| | SDO | — | CMOS | SPI data output. |
| VSS | VSS | Power | — | Ground reference. |
| VDD | VDD | Power | — | Positive supply. |

**Legend:** AN = Analog input or output CMOS = CMOS compatible input or output OD = Open Drain
TTL = TTL compatible input ST = Schmitt Trigger input with CMOS levels
HV = High Voltage XTAL = Crystal

**NOTES:**

## 2.0 MEMORY ORGANIZATION

### 2.1 Program Memory Organization

The PIC16F685/687/689/690 has a 13-bit program counter capable of addressing an 8k x 14 program memory space. Only the first 2k x 14 (0000h-07FFh) for the PIC16F687 is physically implemented and first 4k x 14 (0000h-0FFFh) for the PIC16F685/PIC16F689/ PIC16F690. Accessing a location above these boundaries will cause a wrap around. The Reset vector is at 0000h and the interrupt vector is at 0004h (see Figures 2-1 and 2-2).

**FIGURE 2-1: PROGRAM MEMORY MAP AND STACK FOR THE PIC16F685/689/690**

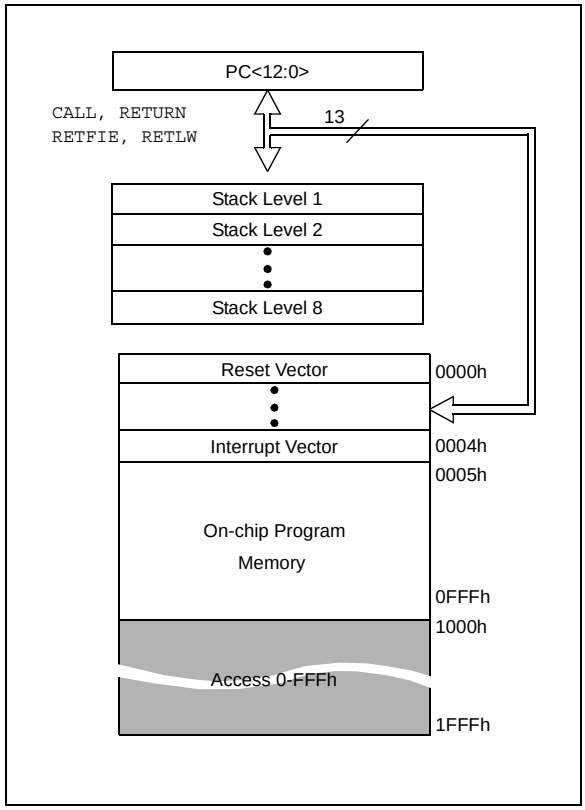
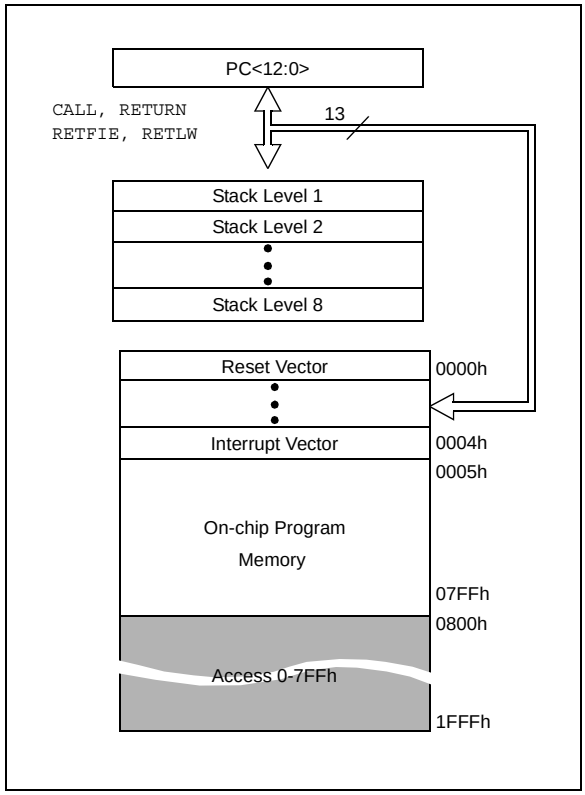



**FIGURE 2-2: PROGRAM MEMORY MAP AND STACK FOR THE PIC16F687**



## 2.2 Data Memory Organization

The data memory (see Figures 2-3, 2-4 and 2-5) is partitioned into four banks which contain the General Purpose Registers (GPR) and the Special Function Registers (SFR). The Special Function Registers are located in the first 32 locations of each bank. Register locations 20h-7Fh in Bank 0 and A0h-EFh (A0-BF, PIC16F687 only) in Bank 1 are General Purpose Registers, implemented as static RAM. Register locations F0h-FFh in Bank 1, 170h-17Fh in Bank 2 and 1F0h-1FFh in Bank 3 point to addresses 70h-7Fh in Bank 0. Other General Purpose Resisters (GPR) are also available in Bank 1 and Bank 2, depending on the device. Details are shown in Figures 2-3, 2-4 and 2-5. All other RAM is unimplemented and returns '0' when read. RP<1:0> (STATUS<6:5>) are the bank select bits:

| RP1 | RP0 | | |
|---|---|---|---|
| 0 | 0 | → | Bank 0 is selected |
| 0 | 1 | → | Bank 1 is selected |
| 1 | 0 | → | Bank 2 is selected |
| 1 | 1 | → | Bank 3 is selected |

### 2.2.1 GENERAL PURPOSE REGISTER FILE

The register file is organized as 128 x 8 in the PIC16F687 and 256 x 8 in the PIC16F685/PIC16F689/PIC16F690. Each register is accessed, either directly or indirectly, through the File Select Register (FSR) (see **Section 2.4 "Indirect Addressing, INDF and FSR Registers"**).

### 2.2.2 SPECIAL FUNCTION REGISTERS

The Special Function Registers are registers used by the CPU and peripheral functions for controlling the desired operation of the device (see Tables 2-1, 2-2, 2-3 and 2-4). These registers are static RAM.

The special registers can be classified into two sets: core and peripheral. The Special Function Registers associated with the "core" are described in this section. Registers related to the operation of peripheral features are described in the section of that peripheral feature.

**FIGURE 2-3: PIC16F685 SPECIAL FUNCTION REGISTERS**

| | File Address | | File Address | | File Address | | File Address |
|---|---|---|---|---|---|---|---|
| Indirect addr. (1) | 00h | Indirect addr. (1) | 80h | Indirect addr. (1) | 100h | Indirect addr. (1) | 180h |
| TMR0 | 01h | OPTION_REG | 81h | TMR0 | 101h | OPTION_REG | 181h |
| PCL | 02h | PCL | 82h | PCL | 102h | PCL | 182h |
| STATUS | 03h | STATUS | 83h | STATUS | 103h | STATUS | 183h |
| FSR | 04h | FSR | 84h | FSR | 104h | FSR | 184h |
| PORTA | 05h | TRISA | 85h | PORTA | 105h | TRISA | 185h |
| PORTB | 06h | TRISB | 86h | PORTB | 106h | TRISB | 186h |
| PORTC | 07h | TRISC | 87h | PORTC | 107h | TRISC | 187h |
| | 08h | | 88h | | 108h | | 188h |
| | 09h | | 89h | | 109h | | 189h |
| PCLATH | 0Ah | PCLATH | 8Ah | PCLATH | 10Ah | PCLATH | 18Ah |
| INTCON | 0Bh | INTCON | 8Bh | INTCON | 10Bh | INTCON | 18Bh |
| PIR1 | 0Ch | PIE1 | 8Ch | EEDAT | 10Ch | EECON1 | 18Ch |
| PIR2 | 0Dh | PIE2 | 8Dh | EEADR | 10Dh | EECON2(1) | 18Dh |
| TMR1L | 0Eh | PCON | 8Eh | EEDATH | 10Eh | | 18Eh |
| TMR1H | 0Fh | OSCCON | 8Fh | EEADRH | 10Fh | | 18Fh |
| T1CON | 10h | OSCTUNE | 90h | | 110h | | 190h |
| TMR2 | 11h | | 91h | | 111h | | 191h |
| T2CON | 12h | PR2 | 92h | | 112h | | 192h |
| | 13h | | 93h | | 113h | | 193h |
| | 14h | | 94h | | 114h | | 194h |
| CCPR1L | 15h | WPUA | 95h | WPUB | 115h | | 195h |
| CCPR1H | 16h | IOCA | 96h | IOCB | 116h | | 196h |
| CCP1CON | 17h | WDTCON | 97h | | 117h | | 197h |
| | 18h | | 98h | VRCON | 118h | | 198h |
| | 19h | | 99h | CM1CON0 | 119h | | 199h |
| | 1Ah | | 9Ah | CM2CON0 | 11Ah | | 19Ah |
| | 1Bh | | 9Bh | CM2CON1 | 11Bh | | 19Bh |
| PWM1CON | 1Ch | | 9Ch | | 11Ch | | 19Ch |
| ECCPAS | 1Dh | | 9Dh | | 11Dh | PSTRCON | 19Dh |
| ADRESH | 1Eh | ADRESL | 9Eh | ANSEL | 11Eh | SRCON | 19Eh |
| ADCON0 | 1Fh | ADCON1 | 9Fh | ANSELH | 11Fh | | 19Fh |
| General Purpose Register 96 Bytes | 20h – 7Fh | General Purpose Register 80 Bytes | A0h – EFh | General Purpose Register 80 Bytes | 120h – 16Fh | | 1A0h |
| | | accesses 70h-7Fh | F0h – FFh | accesses 70h-7Fh | 170h – 17Fh | accesses 70h-7Fh | 1F0h – 1FFh |
| Bank 0 | | Bank1 | | Bank2 | | Bank3 | |

Unimplemented data memory locations, read as '0'.

**Note 1:** Not a physical register.

**FIGURE 2-4: PIC16F687/PIC16F689 SPECIAL FUNCTION REGISTERS**

| | File Address | | File Address | | File Address | | File Address |
|---|---|---|---|---|---|---|---|
| Indirect addr. (1) | 00h | Indirect addr. (1) | 80h | Indirect addr. (1) | 100h | Indirect addr. (1) | 180h |
| TMR0 | 01h | OPTION_REG | 81h | TMR0 | 101h | OPTION_REG | 181h |
| PCL | 02h | PCL | 82h | PCL | 102h | PCL | 182h |
| STATUS | 03h | STATUS | 83h | STATUS | 103h | STATUS | 183h |
| FSR | 04h | FSR | 84h | FSR | 104h | FSR | 184h |
| PORTA | 05h | TRISA | 85h | PORTA | 105h | TRISA | 185h |
| PORTB | 06h | TRISB | 86h | PORTB | 106h | TRISB | 186h |
| PORTC | 07h | TRISC | 87h | PORTC | 107h | TRISC | 187h |
| | 08h | | 88h | | 108h | | 188h |
| | 09h | | 89h | | 109h | | 189h |
| PCLATH | 0Ah | PCLATH | 8Ah | PCLATH | 10Ah | PCLATH | 18Ah |
| INTCON | 0Bh | INTCON | 8Bh | INTCON | 10Bh | INTCON | 18Bh |
| PIR1 | 0Ch | PIE1 | 8Ch | EEDAT | 10Ch | EECON1 | 18Ch |
| PIR2 | 0Dh | PIE2 | 8Dh | EEADR | 10Dh | EECON2(1) | 18Dh |
| TMR1L | 0Eh | PCON | 8Eh | EEDATH(3) | 10Eh | | 18Eh |
| TMR1H | 0Fh | OSCCON | 8Fh | EEADRH(3) | 10Fh | | 18Fh |
| T1CON | 10h | OSCTUNE | 90h | | 110h | | 190h |
| | 11h | | 91h | | 111h | | 191h |
| | 12h | | 92h | | 112h | | 192h |
| SSPBUF | 13h | SSPADD(2) | 93h | | 113h | | 193h |
| SSPCON | 14h | SSPSTAT | 94h | | 114h | | 194h |
| | 15h | WPUA | 95h | WPUB | 115h | | 195h |
| | 16h | IOCA | 96h | IOCB | 116h | | 196h |
| | 17h | WDTCON | 97h | | 117h | | 197h |
| RCSTA | 18h | TXSTA | 98h | VRCON | 118h | | 198h |
| TXREG | 19h | SPBRG | 99h | CM1CON0 | 119h | | 199h |
| RCREG | 1Ah | SPBRGH | 9Ah | CM2CON0 | 11Ah | | 19Ah |
| | 1Bh | BAUDCTL | 9Bh | CM2CON1 | 11Bh | | 19Bh |
| | 1Ch | | 9Ch | | 11Ch | | 19Ch |
| | 1Dh | | 9Dh | | 11Dh | | 19Dh |
| ADRESH | 1Eh | ADRESL | 9Eh | ANSEL | 11Eh | SRCON | 19Eh |
| ADCON0 | 1Fh | ADCON1 | 9Fh | ANSELH | 11Fh | | 19Fh |
| General Purpose Register 96 Bytes | 20h – 7Fh | General Purpose Register 32 Bytes | A0h – BFh | General Purpose Register 80 Bytes (PIC16F689 only) | 120h – 16Fh | | 1A0h – 1EFh |
| | | 48 Bytes (PIC16F689 only) | C0h – EFh | | | | |
| accesses 70h-7Fh | 70h – 7Fh | accesses 70h-7Fh | F0h – FFh | accesses 70h-7Fh | 170h – 17Fh | accesses 70h-7Fh | 1F0h – 1FFh |
| Bank 0 | | Bank1 | | Bank2 | | Bank3 | |

Unimplemented data memory locations, read as '0'.

**Note 1:** Not a physical register.

**2:** Address 93h also accesses the SSP Mask (SSPMSK) register under certain conditions. See Registers 13-2 and 13-3 for more details.

**3:** PIC16F689 only.

**FIGURE 2-5: PIC16F690 SPECIAL FUNCTION REGISTERS**

| | File Address | | File Address | | File Address | | File Address |
|---|---|---|---|---|---|---|---|
| Indirect addr. (1) | 00h | Indirect addr. (1) | 80h | Indirect addr. (1) | 100h | Indirect addr. (1) | 180h |
| TMR0 | 01h | OPTION_REG | 81h | TMR0 | 101h | OPTION_REG | 181h |
| PCL | 02h | PCL | 82h | PCL | 102h | PCL | 182h |
| STATUS | 03h | STATUS | 83h | STATUS | 103h | STATUS | 183h |
| FSR | 04h | FSR | 84h | FSR | 104h | FSR | 184h |
| PORTA | 05h | TRISA | 85h | PORTA | 105h | TRISA | 185h |
| PORTB | 06h | TRISB | 86h | PORTB | 106h | TRISB | 186h |
| PORTC | 07h | TRISC | 87h | PORTC | 107h | TRISC | 187h |
| | 08h | | 88h | | 108h | | 188h |
| | 09h | | 89h | | 109h | | 189h |
| PCLATH | 0Ah | PCLATH | 8Ah | PCLATH | 10Ah | PCLATH | 18Ah |
| INTCON | 0Bh | INTCON | 8Bh | INTCON | 10Bh | INTCON | 18Bh |
| PIR1 | 0Ch | PIE1 | 8Ch | EEDAT | 10Ch | EECON1 | 18Ch |
| PIR2 | 0Dh | PIE2 | 8Dh | EEADR | 10Dh | EECON2(1) | 18Dh |
| TMR1L | 0Eh | PCON | 8Eh | EEDATH | 10Eh | | 18Eh |
| TMR1H | 0Fh | OSCCON | 8Fh | EEADRH | 10Fh | | 18Fh |
| T1CON | 10h | OSCTUNE | 90h | | 110h | | 190h |
| TMR2 | 11h | | 91h | | 111h | | 191h |
| T2CON | 12h | PR2 | 92h | | 112h | | 192h |
| SSPBUF | 13h | SSPADD(2) | 93h | | 113h | | 193h |
| SSPCON | 14h | SSPSTAT | 94h | | 114h | | 194h |
| CCPR1L | 15h | WPUA | 95h | WPUB | 115h | | 195h |
| CCPR1H | 16h | IOCA | 96h | IOCB | 116h | | 196h |
| CCP1CON | 17h | WDTCON | 97h | | 117h | | 197h |
| RCSTA | 18h | TXSTA | 98h | VRCON | 118h | | 198h |
| TXREG | 19h | SPBRG | 99h | CM1CON0 | 119h | | 199h |
| RCREG | 1Ah | SPBRGH | 9Ah | CM2CON0 | 11Ah | | 19Ah |
| | 1Bh | BAUDCTL | 9Bh | CM2CON1 | 11Bh | | 19Bh |
| PWM1CON | 1Ch | | 9Ch | | 11Ch | | 19Ch |
| ECCPAS | 1Dh | | 9Dh | | 11Dh | PSTRCON | 19Dh |
| ADRESH | 1Eh | ADRESL | 9Eh | ANSEL | 11Eh | SRCON | 19Eh |
| ADCON0 | 1Fh | ADCON1 | 9Fh | ANSELH | 11Fh | | 19Fh |
| General Purpose Register 96 Bytes | 20h – 7Fh | General Purpose Register 80 Bytes | A0h – EFh | General Purpose Register 80 Bytes | 120h – 16Fh | | 1A0h |
| | | accesses 70h-7Fh | F0h – FFh | accesses 70h-7Fh | 170h – 17Fh | accesses 70h-7Fh | 1F0h – 1FFh |
| Bank 0 | | Bank1 | | Bank2 | | Bank3 | |

Unimplemented data memory locations, read as '0'.

**Note 1:** Not a physical register.

**2:** Address 93h also accesses the SSP Mask (SSPMSK) register under certain conditions. See Registers 13-2 and 13-3 for more details.

**TABLE 2-1: PIC16F685/687/689/690 SPECIAL REGISTERS SUMMARY BANK 0**

| Addr | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR/BOR Reset | Value on all other Resets(1) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Bank 0** | | | | | | | | | | | |
| 00h | INDF | Addressing this location uses contents of FSR to address data memory (not a physical register) | | | | | | | | xxxx xxxx | xxxx xxxx |
| 01h | TMR0 | Timer0 Module Register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 02h | PCL | Program Counter's (PC) Least Significant Byte | | | | | | | | 0000 0000 | 0000 0000 |
| 03h | STATUS | IRP | RP1 | RP0 | $\overline{TO}$ | $\overline{PD}$ | Z | DC | C | 0001 1xxx | 000q quuu |
| 04h | FSR | Indirect Data Memory Address Pointer | | | | | | | | xxxx xxxx | uuuu uuuu |
| 05h | PORTA | — | — | RA5 | RA4 | RA3 | RA2 | RA1 | RA0 | --xx xxxx | --uu uuuu |
| 06h | PORTB | RB7 | RB6 | RB5 | RB4 | — | — | — | — | xxxx ---- | uuuu ---- |
| 07h | PORTC | RC7 | RC6 | RC5 | RC4 | RC3 | RC2 | RC1 | RC0 | xxxx xxxx | uuuu uuuu |
| 08h | — | Unimplemented | | | | | | | | — | — |
| 09h | — | Unimplemented | | | | | | | | — | — |
| 0Ah | PCLATH | — | — | — | Write Buffer for upper 5 bits of Program Counter | | | | | ---0 0000 | ---0 0000 |
| 0Bh | INTCON | GIE | PEIE | T0IE | INTE | RABIE | T0IF | INTF | RABIF(2) | 0000 000x | 0000 000x |
| 0Ch | PIR1 | — | ADIF | RCIF(3) | TXIF(3) | SSPIF(3) | CCP1IF(4) | TMR2IF(4) | TMR1IF | -000 0000 | -000 0000 |
| 0Dh | PIR2 | OSFIF | C2IF | C1IF | EEIF | — | — | — | — | 0000 ---- | 0000 ---- |
| 0Eh | TMR1L | Holding Register for the Least Significant Byte of the 16-bit TMR1 | | | | | | | | xxxx xxxx | uuuu uuuu |
| 0Fh | TMR1H | Holding Register for the Most Significant Byte of the 16-bit TMR1 | | | | | | | | xxxx xxxx | uuuu uuuu |
| 10h | T1CON | T1GINV | TMR1GE | T1CKPS1 | T1CKPS0 | T1OSCEN | $\overline{T1SYNC}$ | TMR1CS | TMR1ON | 0000 0000 | uuuu uuuu |
| 11h | TMR2 | Timer2 Module Register | | | | | | | | 0000 0000 | 0000 0000 |
| 12h | T2CON | — | TOUTPS3 | TOUTPS2 | TOUTPS1 | TOUTPS0 | TMR2ON | T2CKPS1 | T2CKPS0 | -000 0000 | -000 0000 |
| 13h | SSPBUF(3) | Synchronous Serial Port Receive Buffer/Transmit Register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 14h | SSPCON(3, 5) | WCOL | SSPOV | SSPEN | CKP | SSPM3 | SSPM2 | SSPM1 | SSPM0 | 0000 0000 | 0000 0000 |
| 15h | CCPR1L(4) | Capture/Compare/PWM Register 1 (LSB) | | | | | | | | xxxx xxxx | uuuu uuuu |
| 16h | CCPR1H(4) | Capture/Compare/PWM Register 1 (MSB) | | | | | | | | xxxx xxxx | uuuu uuuu |
| 17h | CCP1CON(4) | P1M1 | P1M0 | DC1B1 | DC1B0 | CCP1M3 | CCP1M2 | CCP1M1 | CCP1M0 | 0000 0000 | 0000 0000 |
| 18h | RCSTA(3) | SPEN | RX9 | SREN | CREN | ADDEN | FERR | OERR | RX9D | 0000 000x | 0000 000x |
| 19h | TXREG(3) | EUSART Transmit Data Register | | | | | | | | 0000 0000 | 0000 0000 |
| 1Ah | RCREG(3) | EUSART Receive Data Register | | | | | | | | 0000 0000 | 0000 0000 |
| 1Bh | — | Unimplemented | | | | | | | | — | — |
| 1Ch | PWM1CON(4) | PRSEN | PDC6 | PDC5 | PDC4 | PDC3 | PDC2 | PDC1 | PDC0 | 0000 0000 | 0000 0000 |
| 1Dh | ECCPAS(4) | ECCPASE | ECCPAS2 | ECCPAS1 | ECCPAS0 | PSSAC1 | PSSAC0 | PSSBD1 | PSSBD0 | 0000 0000 | 0000 0000 |
| 1Eh | ADRESH | A/D Result Register High Byte | | | | | | | | xxxx xxxx | uuuu uuuu |
| 1Fh | ADCON0 | ADFM | VCFG | CHS3 | CHS2 | CHS1 | CHS0 | GO/$\overline{DONE}$ | ADON | 0000 0000 | 0000 0000 |

**Legend:** – = Unimplemented locations read as '0', u = unchanged, x = unknown, q = value depends on condition, shaded = unimplemented

**Note 1:** Other (non Power-up) Resets include $\overline{MCLR}$ Reset and Watchdog Timer Reset during normal operation.

**2:** $\overline{MCLR}$ and WDT Reset do not affect the previous value data latch. The RABIF bit will be cleared upon Reset but will set again if the mismatched exists.

**3:** PIC16F687/PIC16F689/PIC16F690 only.

**4:** PIC16F685/PIC16F690 only.

**5:** When SSPCON bits SSPM<3:0> = 1001, any reads or writes to the SSPADD SFR address are accessed through the SSPMSK register. See Registers 13-2 and 13-3 for more detail.

**TABLE 2-2: PIC16F685/687/689/690 SPECIAL FUNCTION REGISTERS SUMMARY BANK 1**

| Addr | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR/BOR Reset | Value on all other Resets(1) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Bank 1** | | | | | | | | | | | |
| 80h | INDF | Addressing this location uses contents of FSR to address data memory (not a physical register) | | | | | | | | xxxx xxxx | xxxx xxxx |
| 81h | OPTION_REG | $\overline{\text{RABPU}}$ | INTEDG | T0CS | T0SE | PSA | PS2 | PS1 | PS0 | 1111 1111 | 1111 1111 |
| 82h | PCL | Program Counter's (PC) Least Significant Byte | | | | | | | | 0000 0000 | 0000 0000 |
| 83h | STATUS | IRP | RP1 | RP0 | $\overline{\text{TO}}$ | $\overline{\text{PD}}$ | Z | DC | C | 0001 1xxx | 000q quuu |
| 84h | FSR | Indirect Data Memory Address Pointer | | | | | | | | xxxx xxxx | uuuu uuuu |
| 85h | TRISA | — | — | TRISA5 | TRISA4 | TRISA3 | TRISA2 | TRISA1 | TRISA0 | --11 1111 | --11 1111 |
| 86h | TRISB | TRISB7 | TRISB6 | TRISB5 | TRISB4 | — | — | — | — | 1111 ---- | 1111 ---- |
| 87h | TRISC | TRISC7 | TRISC6 | TRISC5 | TRISC4 | TRISC3 | TRISC2 | TRISC1 | TRISC0 | 1111 1111 | 1111 1111 |
| 88h | — | Unimplemented | | | | | | | | — | — |
| 89h | — | Unimplemented | | | | | | | | — | — |
| 8Ah | PCLATH | — | — | — | Write Buffer for the upper 5 bits of the Program Counter | | | | | ---0 0000 | ---0 0000 |
| 8Bh | INTCON | GIE | PEIE | T0IE | INTE | RABIE | T0IF | INTF | RABIF(2) | 0000 000x | 0000 000x |
| 8Ch | PIE1 | — | ADIE | RCIE(3) | TXIE(3) | SSPIE(3) | CCP1IE(4) | TMR2IE(4) | TMR1IE | -000 0000 | -000 0000 |
| 8Dh | PIE2 | OSFIE | C2IE | C1IE | EEIE | — | — | — | — | 0000 ---- | 0000 ---- |
| 8Eh | PCON | — | — | ULPWUE | SBOREN | — | — | $\overline{\text{POR}}$ | $\overline{\text{BOR}}$ | --01 --qq | --0u --uu |
| 8Fh | OSCCON | — | IRCF2 | IRCF1 | IRCF0 | OSTS | HTS | LTS | SCS | -110 q000 | -110 x000 |
| 90h | OSCTUNE | — | — | — | TUN4 | TUN3 | TUN2 | TUN1 | TUN0 | ---0 0000 | ---u uuuu |
| 91h | — | Unimplemented | | | | | | | | — | — |
| 92h | PR2(4) | Timer2 Period Register | | | | | | | | 1111 1111 | 1111 1111 |
| 93h | SSPADD(3, 6) | Synchronous Serial Port ($I^2C$ mode) Address Register | | | | | | | | 0000 0000 | 0000 0000 |
| 93h | SSPMSK(3, 6) | MSK7 | MSK6 | MSK5 | MSK4 | MSK3 | MSK2 | MSK1 | MSK0 | 1111 1111 | 1111 1111 |
| 94h | SSPSTAT(3) | SMP | CKE | D/$\overline{\text{A}}$ | P | S | R/$\overline{\text{W}}$ | UA | BF | 0000 0000 | 0000 0000 |
| 95h | WPUA(5) | — | — | WPUA5 | WPUA4 | — | WPUA2 | WPUA1 | WPUA0 | --11 -111 | --11 -111 |
| 96h | IOCA | — | — | IOCA5 | IOCA4 | IOCA3 | IOCA2 | IOCA1 | IOCA0 | --00 0000 | --00 0000 |
| 97h | WDTCON | — | — | — | WDTPS3 | WDTPS2 | WDTPS1 | WDTPS0 | SWDTEN | ---0 1000 | ---0 1000 |
| 98h | TXSTA(3) | CSRC | TX9 | TXEN | SYNC | SENB | BRGH | TRMT | TX9D | 0000 0010 | 0000 0010 |
| 99h | SPBRG(3) | BRG7 | BRG6 | BRG5 | BRG4 | BRG3 | BRG2 | BRG1 | BRG0 | 0000 0000 | 0000 0000 |
| 9Ah | SPBRGH(3) | BRG15 | BRG14 | BRG13 | BRG12 | BRG11 | BRG10 | BRG9 | BRG8 | 0000 0000 | 0000 0000 |
| 9Bh | BAUDCTL(3) | ABDOVF | RCIDL | — | SCKP | BRG16 | — | WUE | ABDEN | 01-0 0-00 | 01-0 0-00 |
| 9Ch | — | Unimplemented | | | | | | | | — | — |
| 9Dh | — | Unimplemented | | | | | | | | — | — |
| 9Eh | ADRESL | A/D Result Register Low Byte | | | | | | | | xxxx xxxx | uuuu uuuu |
| 9Fh | ADCON1 | — | ADCS2 | ADCS1 | ADCS0 | — | — | — | — | -000 ---- | -000 --- |

**Legend:** – = Unimplemented locations read as '0', u = unchanged, x = unknown, q = value depends on condition, shaded = unimplemented

**Note 1:** Other (non Power-up) Resets include $\overline{\text{MCLR}}$ Reset and Watchdog Timer Reset during normal operation.

**2:** $\overline{\text{MCLR}}$ and WDT Reset do not affect the previous value data latch. The RABIF bit will be cleared upon Reset but will set again if the mismatched exists.

**3:** PIC16F687/PIC16F689/PIC16F690 only.

**4:** PIC16F685/PIC16F690 only.

**5:** RA3 pull-up is enabled when pin is configured as $\overline{\text{MCLR}}$ in Configuration Word.

**6:** Accessible only when SSPM<3:0> = 1001.

**TABLE 2-3: PIC16F685/687/689/690 SPECIAL REGISTERS SUMMARY BANK 2**

| Addr | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR/BOR Reset | Value on all other Resets(1) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Bank 2** | | | | | | | | | | | |
| 100h | INDF | Addressing this location uses contents of FSR to address data memory (not a physical register) | | | | | | | | xxxx xxxx | xxxx xxxx |
| 101h | TMR0 | Timer0 Module Register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 102h | PCL | Program Counter's (PC) Least Significant Byte | | | | | | | | 0000 0000 | 0000 0000 |
| 103h | STATUS | IRP | RP1 | RP0 | $\overline{TO}$ | $\overline{PD}$ | Z | DC | C | 0001 1xxx | 000q quuu |
| 104h | FSR | Indirect Data Memory Address Pointer | | | | | | | | xxxx xxxx | uuuu uuuu |
| 105h | PORTA | — | — | RA5 | RA4 | RA3 | RA2 | RA1 | RA0 | --xx xxxx | --uu uuuu |
| 106h | PORTB | RB7 | RB6 | RB5 | RB4 | — | — | — | — | xxxx ---- | uuuu ---- |
| 107h | PORTC | RC7 | RC6 | RC5 | RC4 | RC3 | RC2 | RC1 | RC0 | xxxx xxxx | uuuu uuuu |
| 108h | — | Unimplemented | | | | | | | | — | — |
| 109h | — | Unimplemented | | | | | | | | — | — |
| 10Ah | PCLATH | — | — | — | Write Buffer for the upper 5 bits of the Program Counter | | | | | ---0 0000 | ---0 0000 |
| 10Bh | INTCON | GIE | PEIE | T0IE | INTE | RABIE | T0IF | INTF | RABIF(2) | 0000 000x | 0000 000x |
| 10Ch | EEDAT | EEDAT7 | EEDAT6 | EEDAT5 | EEDAT4 | EEDAT3 | EEDAT2 | EEDAT1 | EEDAT0 | 0000 0000 | 0000 0000 |
| 10Dh | EEADR | EEADR7 | EEADR6 | EEADR5 | EEADR4 | EEADR3 | EEADR2 | EEADR1 | EEADR0 | 0000 0000 | 0000 0000 |
| 10Eh | EEDATH(3) | — | — | EEDATH5 | EEDATH4 | EEDATH3 | EEDATH2 | EEDATH1 | EEDATH0 | --00 0000 | --00 0000 |
| 10Fh | EEADRH(3) | — | — | — | — | EEADRH3 | EEADRH2 | EEADRH1 | EEADRH0 | ---- 0000 | ---- 0000 |
| 110h | — | Unimplemented | | | | | | | | — | — |
| 111h | — | Unimplemented | | | | | | | | — | — |
| 112h | — | Unimplemented | | | | | | | | — | — |
| 113h | — | Unimplemented | | | | | | | | — | — |
| 114h | — | Unimplemented | | | | | | | | — | — |
| 115h | WPUB | WPUB7 | WPUB6 | WPUB5 | WPUB4 | — | — | — | — | 1111 ---- | 1111 ---- |
| 116h | IOCB | IOCB7 | IOCB6 | IOCB5 | IOCB4 | — | — | — | — | 0000 ---- | 0000 ---- |
| 117h | — | Unimplemented | | | | | | | | — | — |
| 118h | VRCON | C1VREN | C2VREN | VRR | VP6EN | VR3 | VR2 | VR1 | VR0 | 0000 0000 | 0000 0000 |
| 119h | CM1CON0 | C1ON | C1OUT | C1OE | C1POL | — | C1R | C1CH1 | C1CH0 | 0000 -000 | 0000 -000 |
| 11Ah | CM2CON0 | C2ON | C2OUT | C2OE | C2POL | — | C2R | C2CH1 | C2CH0 | 0000 -000 | 0000 -000 |
| 11Bh | CM2CON1 | MC1OUT | MC2OUT | — | — | — | — | T1GSS | C2SYNC | 00-- --10 | 00-- --10 |
| 11Ch | — | Unimplemented | | | | | | | | — | — |
| 11Dh | — | Unimplemented | | | | | | | | — | — |
| 11Eh | ANSEL | ANS7 | ANS6 | ANS5 | ANS4 | ANS3 | ANS2 | ANS1 | ANS0 | 1111 1111 | 1111 1111 |
| 11Fh | ANSELH | — | — | — | — | ANS11 | ANS10 | ANS9 | ANS8 | ---- 1111 | ---- 1111 |

**Legend:** – = Unimplemented locations read as '0', u = unchanged, x = unknown, q = value depends on condition, shaded = unimplemented

**Note 1:** Other (non Power-up) Resets include $\overline{MCLR}$ Reset and Watchdog Timer Reset during normal operation.

**2:** $\overline{MCLR}$ and WDT Reset does not affect the previous value data latch. The RABIF bit will be cleared upon Reset but will set again if the mismatched exists.

**3:** PIC16F685/PIC16F689/PIC16F690 only.

**TABLE 2-4: PIC16F685/687/689/690 SPECIAL FUNCTION REGISTERS SUMMARY BANK 3**

| Addr | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR/BOR Reset | Value on all other Resets(1) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Bank 3** | | | | | | | | | | | |
| 180h | INDF | Addressing this location uses contents of FSR to address data memory (not a physical register) | | | | | | | | xxxx xxxx | xxxx xxxx |
| 181h | OPTION_REG | $\overline{\text{RABPU}}$ | INTEDG | T0CS | T0SE | PSA | PS2 | PS1 | PS0 | 1111 1111 | 1111 1111 |
| 182h | PCL | Program Counter's (PC) Least Significant Byte | | | | | | | | 0000 0000 | 0000 0000 |
| 183h | STATUS | IRP | RP1 | RP0 | $\overline{\text{TO}}$ | $\overline{\text{PD}}$ | Z | DC | C | 0001 1xxx | 000q quuu |
| 184h | FSR | Indirect Data Memory Address Pointer | | | | | | | | xxxx xxxx | uuuu uuuu |
| 185h | TRISA | — | — | TRISA5 | TRISA4 | TRISA3 | TRISA2 | TRISA1 | TRISA0 | --11 1111 | --11 1111 |
| 186h | TRISB | TRISB7 | TRISB6 | TRISB5 | TRISB4 | — | — | — | — | 1111 ---- | 1111 ---- |
| 187h | TRISC | TRISC7 | TRISC6 | TRISC5 | TRISC4 | TRISC3 | TRISC2 | TRISC1 | TRISC0 | 1111 1111 | 1111 1111 |
| 188h | — | Unimplemented | | | | | | | | — | — |
| 189h | — | Unimplemented | | | | | | | | — | — |
| 18Ah | PCLATH | — | — | — | Write Buffer for the upper 5 bits of the Program Counter | | | | | ---0 0000 | ---0 0000 |
| 18Bh | INTCON | GIE | PEIE | T0IE | INTE | RABIE | T0IF | INTF | RABIF(2) | 0000 000x | 0000 000x |
| 18Ch | EECON1 | EEPGD | — | — | — | WRERR | WREN | WR | RD | x--- x000 | 0--- q000 |
| 18Dh | EECON2 | EEPROM Control Register 2 (not a physical register) | | | | | | | | ---- ---- | ---- ---- |
| 18Eh | — | Unimplemented | | | | | | | | — | — |
| 18Fh | — | Unimplemented | | | | | | | | — | — |
| 190h | — | Unimplemented | | | | | | | | — | — |
| 191h | — | Unimplemented | | | | | | | | — | — |
| 192h | — | Unimplemented | | | | | | | | — | — |
| 193h | — | Unimplemented | | | | | | | | — | — |
| 194h | — | Unimplemented | | | | | | | | — | — |
| 195h | — | Unimplemented | | | | | | | | — | — |
| 196h | — | Unimplemented | | | | | | | | — | — |
| 197h | — | Unimplemented | | | | | | | | — | — |
| 198h | — | Unimplemented | | | | | | | | — | — |
| 199h | — | Unimplemented | | | | | | | | — | — |
| 19Ah | — | Unimplemented | | | | | | | | — | — |
| 19Bh | — | Unimplemented | | | | | | | | — | — |
| 19Ch | — | Unimplemented | | | | | | | | — | — |
| 19Dh | PSTRCON(3) | — | — | — | STRSYNC | STRD | STRC | STRB | STRA | ---0 0001 | ---0 0001 |
| 19Eh | SRCON | SR1 | SR0 | C1SEN | C2REN | PULSS | PULSR | — | — | 0000 00-- | 0000 00-- |
| 19Fh | — | Unimplemented | | | | | | | | — | — |

**Legend:** – = Unimplemented locations read as '0', u = unchanged, x = unknown, q = value depends on condition, shaded = unimplemented

**Note 1:** Other (non Power-up) Resets include MCLR Reset and Watchdog Timer Reset during normal operation.

**2:** MCLR and WDT Reset does not affect the previous value data latch. The RABIF bit will be cleared upon Reset but will set again if the mismatched exists.

**3:** PIC16F685/PIC16F690 only.

#### 2.2.2.1 Status Register

The Status register, shown in Register 2-1, contains:

- the arithmetic status of the ALU
- the Reset status
- the bank select bits for data memory (GPR and SFR)

The Status register can be the destination for any instruction, like any other register. If the Status register is the destination for an instruction that affects the Z, DC or C bits, then the write to these three bits is disabled. These bits are set or cleared according to the device logic. Furthermore, the $\overline{TO}$ and $\overline{PD}$ bits are not writable. Therefore, the result of an instruction with the Status register as destination may be different than intended.

For example, `CLRF STATUS`, will clear the upper three bits and set the Z bit. This leaves the Status register as '000u u1uu' (where u = unchanged).

It is recommended, therefore, that only `BCF`, `BSF`, `SWAPF` and `MOVWF` instructions are used to alter the Status register, because these instructions do not affect any Status bits. For other instructions not affecting any Status bits, see the "Instruction Set Summary."

**Note 1:** The C and DC bits operate as a $\overline{Borrow}$ and Digit Borrow out bit, respectively, in subtraction. See the `SUBLW` and `SUBWF` instructions for examples.

**REGISTER 2-1: STATUS – STATUS REGISTER (ADDRESS: 03h, 83h, 103h OR 183h)**

| R/W-0 | R/W-0 | R/W-0 | R-1 | R-1 | R/W-x | R/W-x | R/W-x |
|---|---|---|---|---|---|---|---|
| IRP | RP1 | RP0 | $\overline{TO}$ | $\overline{PD}$ | Z | DC(1) | C(1) |
| bit 7 | | | | | | | bit 0 |

bit 7 **IRP:** Register Bank Select bit (used for indirect addressing)
1 = Bank 2, 3 (100h-1FFh)
0 = Bank 0, 1 (00h-FFh)

bit 6-5 **RP<1:0>**: Register Bank Select bits (used for direct addressing)
00 = Bank 0 (00h-7Fh)
01 = Bank 1 (80h-FFh)
10 = Bank 2 (100h-17Fh)
11 = Bank 3 (180h-1FFh)

bit 4 **$\overline{TO}$:** Time-out bit
1 = After power-up, `CLRWDT` instruction or `SLEEP` instruction
0 = A WDT time-out occurred

bit 3 **$\overline{PD}$**: Power-down bit
1 = After power-up or by the `CLRWDT` instruction
0 = By execution of the `SLEEP` instruction

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

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

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

**Note 1:** For $\overline{Borrow}$, the polarity is reversed. A subtraction is executed by adding the two's complement of the second operand. For rotate (`RRF`, `RLF`) instructions, this bit is loaded with either the high or low-order bit of the source register.

| **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 |

### 2.2.2.2 OPTION Register

The OPTION register is a readable and writable register, which contains various control bits to configure:

- TMR0/WDT prescaler
- External RA2/INT interrupt
- TMR0
- Weak pull-ups on PORTA/PORTB

**Note:** To achieve a 1:1 prescaler assignment for TMR0, assign the prescaler to the WDT by setting PSA bit to '1' (OPTION_REG<3>). See **Section 5.4 "Prescaler"**.

**REGISTER 2-2: OPTION_REG – OPTION REGISTER (ADDRESS: 81h OR 181h)**

| R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 |
|---|---|---|---|---|---|---|---|
| $\overline{\text{RABPU}}$ | INTEDG | T0CS | T0SE | PSA | PS2 | PS1 | PS0 |
| bit 7 | | | | | | | bit 0 |

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

bit 6 **INTEDG:** Interrupt Edge Select bit
1 = Interrupt on rising edge of RA2/AN2/T0CKI/INT/C1OUT pin
0 = Interrupt on falling edge of RA2/AN2/T0CKI/INT/C1OUT pin

bit 5 **T0CS:** TMR0 Clock Source Select bit
1 = Transition on RA2/AN2/T0CKI/INT/C1OUT pin
0 = Internal instruction cycle clock (CLKOUT)

bit 4 **T0SE:** TMR0 Source Edge Select bit
1 = Increment on high-to-low transition on RA2/AN2/T0CKI/INT/C1OUT pin
0 = Increment on low-to-high transition on RA2/AN2/T0CKI/INT/C1OUT 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 **PS<2:0>:** Prescaler Rate Select bits

| Bit Value | TMR0 Rate | WDT Rate(1) |
|---|---|---|
| 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 |

**Note 1:** A dedicated 16-bit WDT postscaler is available. See **Section 14.5 "Watchdog Timer (WDT)"** for more information.

| **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 |

### 2.2.2.3 INTCON Register

The INTCON register is a readable and writable register, which contains the various enable and flag bits for TMR0 register overflow, PORTA change and external RA2/AN2/T0CKI/INT/C1OUT pin interrupts.

| **Note:** | Interrupt flag bits are 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. |
|---|---|

**REGISTER 2-3: INTCON – INTERRUPT CONTROL REGISTER (ADDRESS: 0Bh, 8Bh, 10Bh OR 18Bh)**

| 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 | RABIE(1,3) | T0IF(2) | INTF | RABIF |
| bit 7 | | | | | | | bit 0 |

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 **INTE:** RA2/INT External Interrupt Enable bit
1 = Enables the RA2/INT external interrupt
0 = Disables the RA2/INT external interrupt

bit 3 **RABIE:** PORTA/PORTB Change Interrupt Enable bit(1, 3)
1 = Enables the PORTA/PORTB change interrupt
0 = Disables the PORTA/PORTB change interrupt

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

bit 1 **INTF:** RA2/INT External Interrupt Flag bit
1 = The RA2/INT external interrupt occurred (must be cleared in software)
0 = The RA2/INT external interrupt did not occur

bit 0 **RABIF:** PORTA/PORTB Change Interrupt Flag bit
1 = When at least one of the PORTA or PORTB general purpose I/O pins changed state (must be cleared in software)
0 = None of the PORTA or PORTB general purpose I/O pins have changed state

**Note 1:** IOCA or IOCB register must also be enabled.

**2:** T0IF bit is set when Timer0 rolls over. Timer0 is unchanged on Reset and should be initialized before clearing T0IF bit.

**3:** Includes ULPWU interrupt.

| **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 |

### 2.2.2.4 PIE1 Register

The PIE1 register contains the interrupt enable bits, as shown in Register 2-4.

**Note:** Bit PEIE (INTCON<6>) must be set to enable any peripheral interrupt.

**REGISTER 2-4: PIE1 – PERIPHERAL INTERRUPT ENABLE REGISTER 1 (ADDRESS: 8Ch)**

| U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | ADIE | RCIE(2) | TXIE(2) | SSPIE(2) | CCP1IE(1) | TMR2IE(1) | TMR1IE |
| bit 7 | | | | | | | bit 0 |

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

bit 6 **ADIE:** A/D Converter Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 5 **RCIE:** EUSART Receive Interrupt Enable bit(2)
1 = Enabled
0 = Disabled

bit 4 **TXIE:** EUSART Transmit Interrupt Enable bit(2)
1 = Enabled
0 = Disabled

bit 3 **SSPIE:** Synchronous Serial Port (SSP) Interrupt Enable bit(2)
1 = Enabled
0 = Disabled

bit 2 **CCP1IE:** CCP1 Interrupt Enable bit(1)
1 = Enabled
0 = Disabled

bit 1 **TMR2IE:** TMR2 to PR2 Match Interrupt Enable bit(1)
1 = Enabled
0 = Disabled

bit 0 **TMR1IE:** TMR1 Overflow Interrupt Enable bit
1 = Enabled
0 = Disabled

**Note 1:** PIC16F685/PIC16F690 only.
**2:** PIC16F687/PIC16F689/PIC16F690 only.

| **Legend:** | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' | |
| - n = Value at POR | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |

### 2.2.2.5 PIE2 Register

The PIE2 register contains the interrupt enable bits, as shown in Register 2-5.

**Note:** Bit PEIE (INTCON<6>) must be set to enable any peripheral interrupt.

**REGISTER 2-5: PIE2 – PERIPHERAL INTERRUPT ENABLE REGISTER 2 (ADDRESS: 8Dh)**

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | U-0 | U-0 | U-0 | U-0 |
|---|---|---|---|---|---|---|---|
| OSFIE | C2IE | C1IE | EEIE | — | — | — | — |
| bit 7 | | | | | | | bit 0 |

bit 7 **OSFIE:** Oscillator Fail Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 6 **C2IE:** Comparator 2 Interrupt Enable bit
1 = Enables Comparator 2 interrupt
0 = Disables Comparator 2 interrupt

bit 5 **C1IE:** Comparator 1 Interrupt Enable bit
1 = Enables Comparator 1 interrupt
0 = Disables Comparator 1 interrupt

bit 4 **EEIE:** EE Write Operation Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 3-0 **Unimplemented:** Read as '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 |

### 2.2.2.6 PIR1 Register

The PIR1 register contains the interrupt flag bits, as shown in Register 2-6.

| | |
|---|---|
| **Note:** | Interrupt flag bits are 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. |

**REGISTER 2-6: PIR1 – PERIPHERAL INTERRUPT REQUEST REGISTER 1 (ADDRESS: 0Ch)**

| U-0 | R/W-0 | R-0 | R-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | ADIF | RCIF(1) | TXIF(1) | SSPIF(1) | CCP1IF(2) | TMR2IF(2) | TMR1IF |
| bit 7 | | | | | | | bit 0 |

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

bit 6 **ADIF:** A/D Converter Interrupt Flag bit
1 = The A/D conversion completed (must be cleared in software)
0 = The A/D conversion is not complete

bit 5 **RCIF:** EUSART Receive Interrupt Flag bit(1)
1 = The EUSART receive buffer is full (cleared by reading RCREG)
0 = The EUSART receive buffer is not full

bit 4 **TXIF:** EUSART Transmit Interrupt Flag bit(1)
1 = The EUSART transmit buffer is empty (cleared by writing to TXREG)
0 = The EUSART transmit buffer is full

bit 3 **SSPIF:** Synchronous Serial Port (SSP) Interrupt Flag bit(1)
1 = The Transmission/Reception is complete (must be cleared in software)
0 = Waiting to Transmit/Receive

bit 2 **CCP1IF:** CCP1 Interrupt Flag bit(2)
Capture mode
1 = A TMR1 register capture occurred (must be cleared in software)
0 = No TMR1 register capture occurred
Compare mode
1 = A TMR1 register compare match occurred (must be cleared in software)
0 = No TMR1 register compare match occurred
PWM mode
Unused in this mode

bit 1 **TMR2IF:** TMR2 to PR2 Interrupt Flag bit(2)
1 = A TMR2 to PR2 match occurred (must be cleared in software)
0 = No TMR2 to PR2 match occurred

bit 0 **TMR1IF:** TMR1 Overflow Interrupt Flag bit
1 = The TMR1 register overflowed (must be cleared in software)
0 = The TMR1 register did not overflow

**Note 1:** PIC16F687/PIC16F689/PIC16F690 only.
**2:** PIC16F685/PIC16F690 only.

| **Legend:** | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' | |
| - n = Value at POR | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |

### 2.2.2.7 PIR2 Register

The PIR2 register contains the interrupt flag bits, as shown in Register 2-7.

| | |
|---|---|
| **Note:** | Interrupt flag bits are 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. |

**REGISTER 2-7: PIR2 – PERIPHERAL INTERRUPT REQUEST REGISTER 2 (ADDRESS: 0Dh)**

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | U-0 | U-0 | U-0 | U-0 |
|---|---|---|---|---|---|---|---|
| OSFIF | C2IF | C1IF | EEIF | — | — | — | — |
| bit 7 | | | | | | | bit 0 |

bit 7 **OSFIF:** Oscillator Fail Interrupt Flag bit
1 = System oscillator failed, clock input has changed to INTOSC (must be cleared in software)
0 = System clock operating

bit 6 **C2IF:** Comparator 2 Interrupt Flag bit
1 = Comparator output (C2OUT bit) has changed (must be cleared in software)
0 = Comparator output (C2OUT bit) has not changed

bit 5 **C1IF:** Comparator 1 Interrupt Flag bit
1 = Comparator output (C1OUT bit) has changed (must be cleared in software)
0 = Comparator output (C1OUT bit) has not changed

bit 4 **EEIF:** EE Write Operation Interrupt Flag bit
1 = Write operation completed (must be cleared in software)
0 = Write operation has not completed or has not started

bit 3-0 **Unimplemented:** Read as '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 |

### 2.2.2.8 PCON Register

The Power Control (PCON) register (see Register 2-8) contains flag bits to differentiate between a:

- Power-on Reset ($\overline{POR}$)
- Brown-out Reset ($\overline{BOR}$)
- Watchdog Timer Reset (WDT)
- External $\overline{MCLR}$ Reset

The PCON register also controls the Ultra Low-Power Wake-up and software enable of the $\overline{BOR}$.

**REGISTER 2-8: PCON — POWER CONTROL REGISTER (ADDRESS: 8Eh)**

| U-0 | U-0 | R/W-0 | R/W-1 | U-0 | U-0 | R/W-0 | R/W-x |
|---|---|---|---|---|---|---|---|
| — | — | ULPWUE | SBOREN(1) | — | — | $\overline{POR}$ | $\overline{BOR}$ |
| bit 7 | | | | | | | bit 0 |

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

bit 5 **ULPWUE:** Ultra Low-Power Wake-up Enable bit
1 = Ultra Low-Power Wake-up enabled
0 = Ultra Low-Power Wake-up disabled

bit 4 **SBOREN:** Software BOR Enable bit(1)
1 = BOR enabled
0 = BOR disabled

bit 3-2 **Unimplemented:** Read as '0'

bit 1 **$\overline{POR}$:** Power-on Reset Status bit
1 = No Power-on Reset occurred
0 = A Power-on Reset occurred (must be set in software after a Power-on Reset occurs)

bit 0 **$\overline{BOR}$:** Brown-out Reset Status bit
1 = No Brown-out Reset occurred
0 = A Brown-out Reset occurred (must be set in software after a Brown-out Reset occurs)

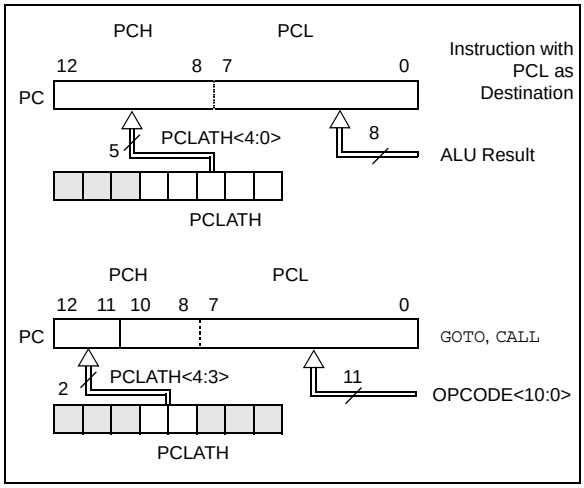
**Note 1:** BOREN<1:0> = 01 in the Configuration Word register for this bit to control the BOR.

| **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 |

## 2.3 PCL and PCLATH

The Program Counter (PC) is 13 bits wide. The low byte comes from the PCL register, which is a readable and writable register. The high byte (PC<12:8>) is not directly readable or writable and comes from PCLATH. On any Reset, the PC is cleared. Figure 2-6 shows the two situations for the loading of the PC. The upper example in Figure 2-6 shows how the PC is loaded on a write to PCL (PCLATH<4:0> → PCH). The lower example in Figure 2-6 shows how the PC is loaded during a CALL or GOTO instruction (PCLATH<4:3> → PCH).

**FIGURE 2-6: LOADING OF PC IN DIFFERENT SITUATIONS**




### 2.3.1 COMPUTED GOTO

A computed GOTO is accomplished by adding an offset to the program counter (ADDWF PCL). When performing a table read using a computed GOTO method, care should be exercised if the table location crosses a PCL memory boundary (each 256-byte block). Refer to the Application Note AN556, *"Implementing a Table Read"* (DS00556).

### 2.3.2 STACK

The PIC16F685/687/689/690 devices have an 8-level x 13-bit wide hardware stack (see Figures 2-1 and 2-2). The stack space is not part of either program or data space and the Stack Pointer is not readable or writable. The PC is PUSHed onto the stack when a CALL instruction is executed or an interrupt causes a branch. The stack is POPed in the event of a RETURN, RETLW or a RETFIE instruction execution. PCLATH is not affected by a PUSH or POP operation.

The stack operates as a circular buffer. This means that after the stack has been PUSHed eight times, the ninth push overwrites the value that was stored from the first push. The tenth push overwrites the second push (and so on).

| | |
|---|---|
| **Note 1:** | There are no Status bits to indicate stack overflow or stack underflow conditions. |
| **2:** | There are no instructions/mnemonics called PUSH or POP. These are actions that occur from the execution of the CALL, RETURN, RETLW and RETFIE instructions or the vectoring to an interrupt address. |

## 2.4 Indirect Addressing, INDF and FSR Registers

The INDF register is not a physical register. Addressing the INDF register will cause indirect addressing.

Indirect addressing is possible by using the INDF register. Any instruction using the INDF register actually accesses data pointed to by the File Select Register (FSR). Reading INDF itself indirectly will produce 00h. Writing to the INDF register indirectly results in a no operation (although Status bits may be affected). An effective 9-bit address is obtained by concatenating the 8-bit FSR and the IRP bit (STATUS<7>), as shown in Figure 2-7.

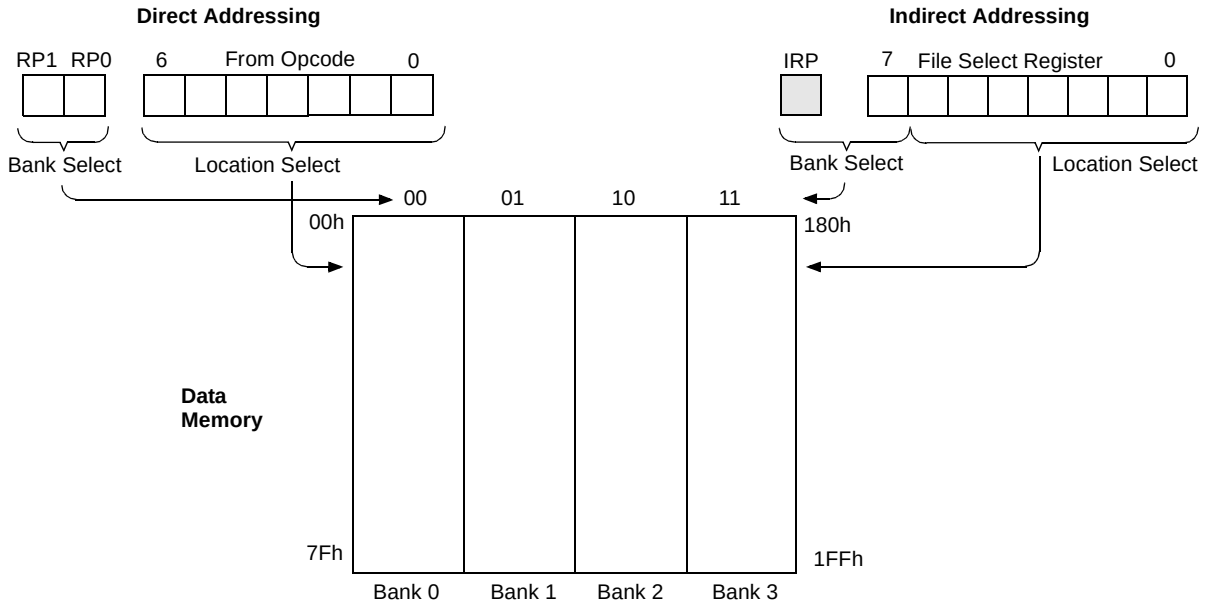
A simple program to clear RAM location 20h-2Fh using indirect addressing is shown in Example 2-1.

**EXAMPLE 2-1: INDIRECT ADDRESSING**

```
        MOVLW   0x20    ;initialize pointer
        MOVWF   FSR     ;to RAM
NEXT    CLRF    INDF    ;clear INDF register
        INCF    FSR     ;inc pointer
        BTFSS   FSR,4   ;all done?
        GOTO    NEXT    ;no clear next
CONTINUE                ;yes continue
```

**FIGURE 2-7: DIRECT/INDIRECT ADDRESSING PIC16F685/687/689/690**




For memory map detail, see Figures 2-3, 2-4 and 2-5.

**NOTES:**

## 3.0 CLOCK SOURCES

### 3.1 Overview

The PIC16F685/687/689/690 devices have a wide variety of clock sources and selection features to allow it to be used in a wide range of applications while maximizing performance and minimizing power consumption. Figure 3-1 illustrates a block diagram of the PIC16F685/687/689/690 clock sources.

Clock sources can be configured from external oscillators, quartz crystal resonators, ceramic resonators and Resistor-Capacitor (RC) circuits. In addition, the system clock source can be configured from one of two internal oscillators, with a choice of speeds selectable via software. Additional clock features include:

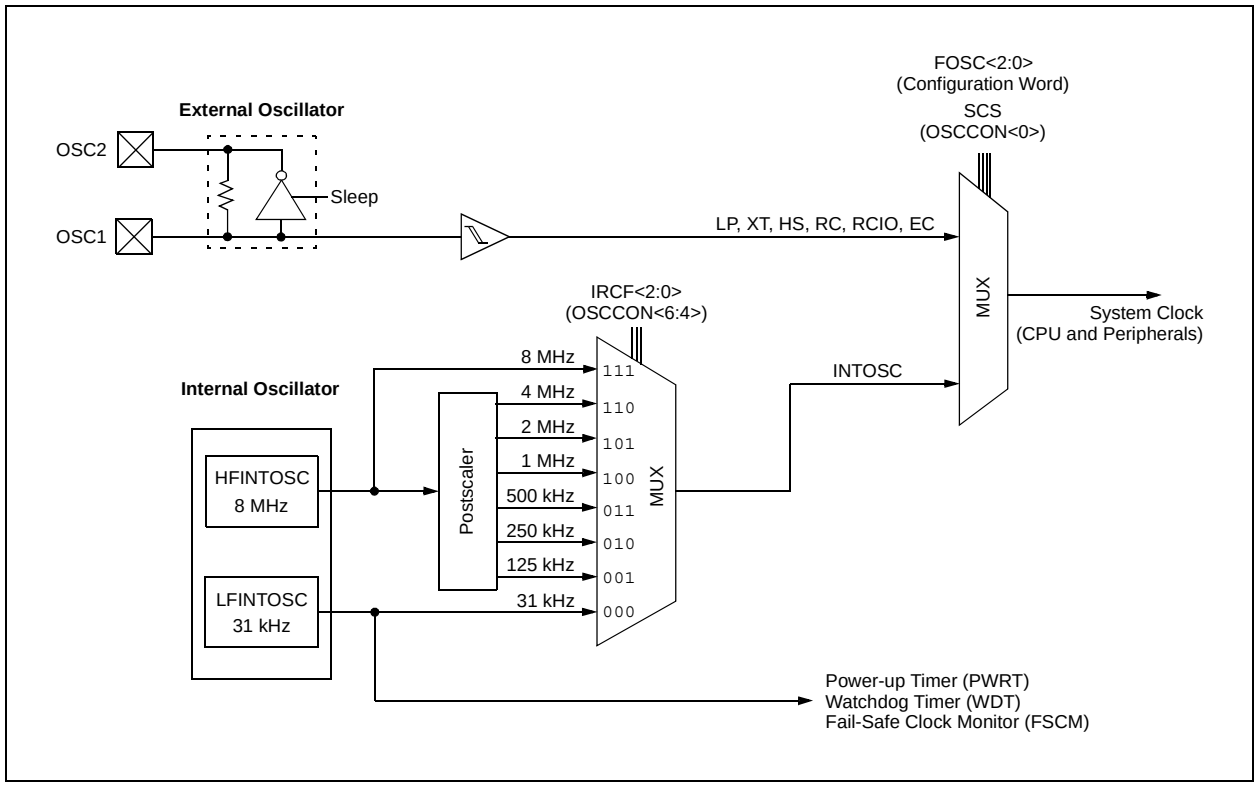
- Selectable system clock source between external or internal via software.
- Two-Speed Clock Start-up mode, which minimizes latency between external oscillator start-up and code execution.
- Fail-Safe Clock Monitor (FSCM) designed to detect a failure of the external clock source (LP, XT, HS, EC or RC modes) and switch to the internal oscillator.

The PIC16F685/687/689/690 can be configured in one of eight clock modes.

1. EC – External clock with I/O on RA4.
2. LP – 32 kHz low-power Crystal mode.
3. XT – Medium gain Crystal or Ceramic Resonator Oscillator mode.
4. HS – High gain Crystal or Ceramic Resonator mode.
5. RC – External Resistor-Capacitor (RC) with FOSC/4 output on RA4.
6. RCIO – External Resistor-Capacitor with I/O on RA4.
7. INTOSC – Internal oscillator with FOSC/4 output on RA4 and I/O on RA5.
8. INTOSCIO – Internal oscillator with I/O on RA4 and RA5.

Clock Source modes are configured by the FOSC<2:0> bits in the Configuration Word register (see **Section 14.0 "Special Features of the CPU"**). The internal clock can be generated from two internal oscillators. The HFINTOSC is a high-frequency calibrated oscillator. The LFINTOSC is a low-frequency uncalibrated oscillator.

**FIGURE 3-1: PIC16F685/687/689/690 CLOCK SOURCE BLOCK DIAGRAM**



## 3.2 Clock Source Modes

Clock Source modes can be classified as external or internal.

- External Clock modes rely on external circuitry for the clock source. Examples are oscillator modules (EC mode), quartz crystal resonators or ceramic resonators (LP, XT and HS modes), and Resistor-Capacitor (RC mode) circuits.
- Internal clock sources are contained internally within the PIC16F685/687/689/690. The PIC16F685/687/689/690 has two internal oscillators, the 8 MHz High-Frequency Internal Oscillator (HFINTOSC) and 31 kHz Low-Frequency Internal Oscillator (LFINTOSC).

The system clock can be selected between external or internal clock sources via the System Clock Selection (SCS) bit (see **Section 3.5 "Clock Switching"**).

## 3.3 External Clock Modes

### 3.3.1 OSCILLATOR START-UP TIMER (OST)

If the PIC16F685/687/689/690 is configured for LP, XT or HS modes, the Oscillator Start-up Timer (OST) counts 1024 oscillations from the OSC1 pin, following a Power-on Reset (POR) and the Power-up Timer (PWRT) has expired (if configured), or a wake-up from Sleep. During this time, the program counter does not increment and program execution is suspended. The OST ensures that the oscillator circuit, using a quartz crystal resonator or ceramic resonator, has started and is providing a stable system clock to the PIC16F685/687/689/690. When switching between clock sources a delay is required to allow the new clock to stabilize. These oscillator delays are shown in Table 3-1.

In order to minimize latency between external oscillator start-up and code execution, the Two-Speed Clock Start-up mode can be selected (see **Section 3.6 "Two-Speed Clock Start-up Mode"**).

**TABLE 3-1: OSCILLATOR DELAY EXAMPLES**

| Switch From | Switch To | Frequency | Oscillator Delay |
|---|---|---|---|
| Sleep/POR | LFINTOSC<br>HFINTOSC | 31 kHz<br>125 kHz to 8 MHz | 5 µs-10 µs (approx.) CPU Start-up(1) |
| Sleep/POR | EC, RC | DC – 20 MHz | |
| LFINTOSC (31 kHz) | EC, RC | DC – 20 MHz | |
| Sleep/POR | LP, XT, HS | 32 kHz to 20 MHz | 1024 Clock Cycles (OST) |
| LFINTOSC (31 kHz) | HFINTOSC | 125 kHz to 8 MHz | 1 µs (approx.) |

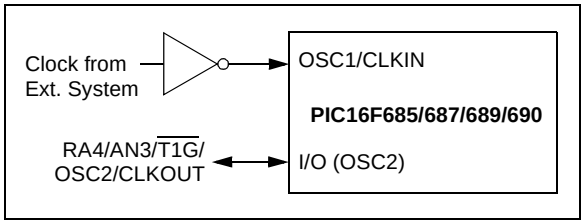
**Note 1:** The 5 µs to 10 µs start-up delay is based on a 1 MHz system clock.

### 3.3.2 EC MODE

The External Clock (EC) mode allows an externally generated logic level as the system clock source. When operating in this mode, an external clock source is connected to the OSC1 pin and the RA4/AN3/$\overline{T1G}$/OSC2/CLKOUT pin is available for general purpose I/O. Figure 3-2 shows the pin connections for EC mode.

The Oscillator Start-up Timer (OST) is disabled when EC mode is selected. Therefore, there is no delay in operation after a Power-on Reset (POR) or wake-up from Sleep. Because the PIC16F685/687/689/690 design is fully static, stopping the external clock input will have the effect of halting the device while leaving all data intact. Upon restarting the external clock, the device will resume operation as if no time had elapsed.

**FIGURE 3-2: EXTERNAL CLOCK (EC) MODE OPERATION**



### 3.3.3 LP, XT, HS MODES

The LP, XT and HS modes support the use of quartz crystal resonators or ceramic resonators connected to the OSC1 and OSC2 pins (Figure 3-3). The mode selects a low, medium or high gain setting of the internal inverter-amplifier to support various resonator types and speed.

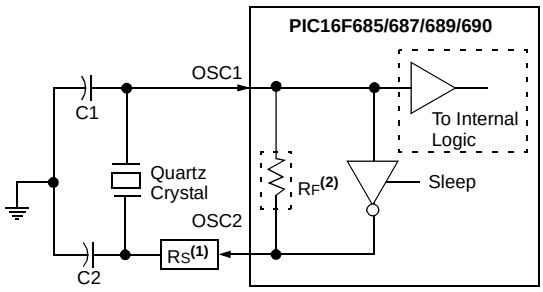
**LP** Oscillator mode selects the lowest gain setting of the internal inverter-amplifier. LP mode current consumption is the least of the three modes. This mode is best suited to drive resonators with a low drive level specification, for example, tuning fork type crystals.

**XT** Oscillator mode selects the intermediate gain setting of the internal inverter-amplifier. XT mode current consumption is the medium of the three modes. This mode is best suited to drive resonators with a medium drive level specification, for example, low-frequency/AT-cut quartz crystal resonators.

**HS** Oscillator mode selects the highest gain setting of the internal inverter-amplifier. HS mode current consumption is the highest of the three modes. This mode is best suited for resonators that require a high drive setting, for example, high-frequency/AT-cut quartz crystal resonators or ceramic resonators.

Figure 3-3 and Figure 3-4 show typical circuits for quartz crystal and ceramic resonators, respectively.

**FIGURE 3-3: QUARTZ CRYSTAL OPERATION (LP, XT OR HS MODE)**

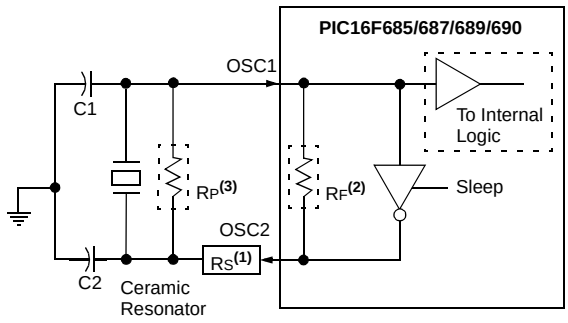



**Note 1:** A series resistor (RS) may be required for quartz crystals with low drive level.

**2:** The value of RF varies with the Oscillator mode selected (typically between 2 MΩ to 10 MΩ).

| **Note 1:** | Quartz crystal characteristics vary according to type, package and manufacturer. The user should consult the manufacturer data sheets for specifications and recommended application. |
|---|---|
| **2:** | Always verify oscillator performance over the VDD and temperature range that is expected for the application. |

**FIGURE 3-4: CERAMIC RESONATOR OPERATION (XT OR HS MODE)**




**Note 1:** A series resistor (RS) may be required for ceramic resonators with low drive level.

**2:** The value of RF varies with the Oscillator mode selected (typically between 2 MΩ to 10 MΩ).

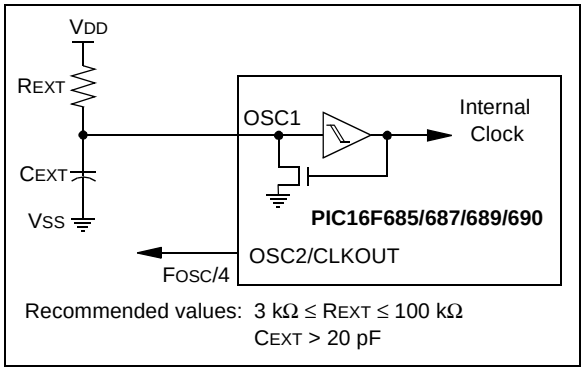
**3:** An additional parallel feedback resistor (RP) may be required for proper ceramic resonator operation (typical value 1 MΩ).

### 3.3.4 EXTERNAL RC MODES

The External Resistor-Capacitor (RC) modes support the use of an external RC circuit. This allows the designer maximum flexibility in frequency choice while keeping costs to a minimum when clock accuracy is not required. There are two modes, RC and RCIO.

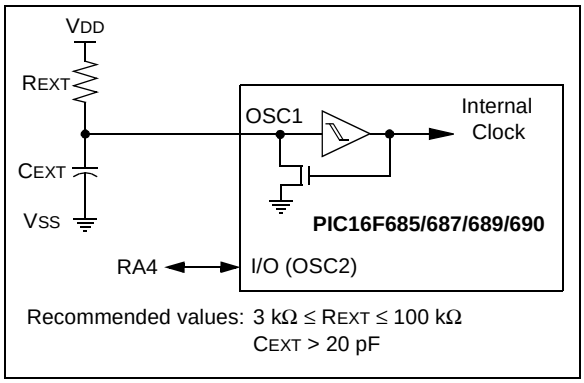
In RC mode, the RC circuit connects to the OSC1 pin. The OSC2/CLKOUT pin outputs the RC oscillator frequency divided by 4. This signal may be used to provide a clock for external circuitry, synchronization, calibration, test or other application requirements. Figure 3-5 shows the RC mode connections.

**FIGURE 3-5: RC MODE**




In RCIO mode, the RC circuit is connected to the OSC1 pin. The OSC2 pin becomes an additional general purpose I/O pin. The I/O pin becomes bit 4 of PORTA (RA4). Figure 3-6 shows the RCIO mode connections.

**FIGURE 3-6: RCIO MODE**




The RC oscillator frequency is a function of the supply voltage, the resistor (REXT) and capacitor (CEXT) values and the operating temperature. Other factors affecting the oscillator frequency are:

- threshold voltage variation
- component tolerances
- packaging variations in capacitance

The user also needs to take into account variation due to tolerance of external RC components used.

## 3.4 Internal Clock Modes

The PIC16F685/687/689/690 has two independent, internal oscillators that can be configured or selected as the system clock source.

1. The **HFINTOSC** (High-Frequency Internal Oscillator) is factory calibrated and operates at 8 MHz. The frequency of the HFINTOSC can be user adjusted ±12% via software using the OSCTUNE register (Register 3-1).
2. The **LFINTOSC** (Low-Frequency Internal Oscillator) is uncalibrated and operates at approximately 31 kHz.

The system clock speed can be selected via software using the Internal Oscillator Frequency Select (IRCF) bits.

The system clock can be selected between external or internal clock sources via the System Clock Selection (SCS) bit (see **Section 3.5 "Clock Switching"**).

### 3.4.1 INTOSC AND INTOSCIO MODES

The INTOSC and INTOSCIO modes configure the internal oscillators as the system clock source when the device is programmed using the Oscillator Selection (FOSC) bits in the Configuration Word register (Register 14-1).

In **INTOSC** mode, the OSC1 pin is available for general purpose I/O. The OSC2/CLKOUT pin outputs the selected internal oscillator frequency divided by 4. The CLKOUT signal may be used to provide a clock for external circuitry, synchronization, calibration, test or other application requirements.

In **INTOSCIO** mode, the OSC1 and OSC2 pins are available for general purpose I/O.

### 3.4.2 HFINTOSC

The High-Frequency Internal Oscillator (HFINTOSC) is a factory calibrated 8 MHz internal clock source. The frequency of the HFINTOSC can be altered approximately ±12% via software using the OSCTUNE register (Register 3-1).

The output of the HFINTOSC connects to a postscaler and multiplexer (see Figure 3-1). One of seven frequencies can be selected via software using the IRCF bits (see **Section 3.4.4 "Frequency Select Bits (IRCF)"**).

The HFINTOSC is enabled by selecting any frequency between 8 MHz and 125 kHz (IRCF ≠ 000) as the system clock source (SCS = 1), or when Two-Speed Start-up is enabled (IESO = 1 and IRCF ≠ 000).

The HF Internal Oscillator (HTS) bit (OSCCON<2>) indicates whether the HFINTOSC is stable or not.

#### 3.4.2.1 OSCTUNE Register

The HFINTOSC is factory calibrated but can be adjusted in software by writing to the OSCTUNE register (Register 3-1).

The OSCTUNE register has a tuning range of ±12%. The default value of the OSCTUNE register is '0'. The value is a 5-bit two's complement number. Due to process variation, the monotonicity and frequency step cannot be specified.

When the OSCTUNE register is modified, the HFINTOSC frequency will begin shifting to the new frequency. The HFINTOSC clock will stabilize within 1 ms. Code execution continues during this shift. There is no indication that the shift has occurred.

OSCTUNE does not affect the LFINTOSC frequency. Operation of features that depend on the LFINTOSC clock source frequency, such as the Power-up Timer (PWRT), Watchdog Timer (WDT), Fail-Safe Clock Monitor (FSCM) and peripherals, are *not* affected by the change in frequency.

**REGISTER 3-1: OSCTUNE – OSCILLATOR TUNING RESISTOR (ADDRESS: 90h)**

| U-0 | U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | — | — | TUN4 | TUN3 | TUN2 | TUN1 | TUN0 |
| bit 7 | | | | | | | bit 0 |

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

bit 4-0 **TUN<4:0>:** Frequency Tuning bits

01111 = Maximum frequency
01110 =
•
•
•
00001 =
00000 = Oscillator module is running at the calibrated frequency.
11111 =
•
•
•
10000 = Minimum frequency

| **Legend:** | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' | |
| -n = Value at POR | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |

### 3.4.3 LFINTOSC

The Low-Frequency Internal Oscillator (LFINTOSC) is an uncalibrated (approximate) 31 kHz internal clock source.

The output of the LFINTOSC connects to a postscaler and multiplexer (see Figure 3-1). 31 kHz can be selected via software using the IRCF bits (see **Section 3.4.4 "Frequency Select Bits (IRCF)"**). The LFINTOSC is also the frequency for the Power-up Timer (PWRT), Watchdog Timer (WDT) and Fail-Safe Clock Monitor (FSCM).

The LFINTOSC is enabled by selecting 31 kHz (IRCF = 000) as the system clock source (SCS = 1), or when any of the following are enabled:

- Two-Speed Start-up (IESO = 1 and IRCF = 000)
- Power-up Timer (PWRT)
- Watchdog Timer (WDT)
- Fail-Safe Clock Monitor (FSCM)

The LF Internal Oscillator (LTS) bit (OSCCON<1>) indicates whether the LFINTOSC is stable or not.

### 3.4.4 FREQUENCY SELECT BITS (IRCF)

The output of the 8 MHz HFINTOSC and 31 kHz LFINTOSC connects to a postscaler and multiplexer (see Figure 3-1). The Internal Oscillator Frequency select bits, IRCF<2:0> (OSCCON<6:4>), select the frequency output of the internal oscillators. One of eight frequencies can be selected via software:

- 8 MHz
- 4 MHz (Default after Reset)
- 2 MHz
- 1 MHz
- 500 kHz
- 250 kHz
- 125 kHz
- 31 kHz

| | |
|---|---|
| **Note:** | Following any Reset, the IRCF bits are set to '110' and the frequency selection is set to 4 MHz. The user can modify the IRCF bits to select a different frequency. |

### 3.4.5 HF AND LF INTOSC CLOCK SWITCH TIMING

When switching between the LFINTOSC and the HFINTOSC, the new oscillator may already be shut down to save power. If this is the case, there is a 10 µs delay after the IRCF bits are modified before the frequency selection takes place. The LTS/HTS bits will reflect the current active status of the LFINTOSC and the HFINTOSC oscillators. The timing of a frequency selection is as follows:

1. IRCF bits are modified.
2. If the new clock is shut down, a 10 µs clock start-up delay is started.
3. Clock switch circuitry waits for a falling edge of the current clock.
4. CLKOUT is held low and the clock switch circuitry waits for a rising edge in the new clock.
5. CLKOUT is now connected with the new clock. HTS/LTS bits are updated as required.
6. Clock switch is complete.

If the internal oscillator speed selected is between 8 MHz and 125 kHz, there is no start-up delay before the new frequency is selected. This is because the old and the new frequencies are derived from the HFINTOSC via the postscaler and multiplexer.

## 3.5 Clock Switching

The system clock source can be switched between external and internal clock sources via software using the System Clock Select (SCS) bit.

### 3.5.1 SYSTEM CLOCK SELECT (SCS) BIT

The System Clock Select (SCS) bit (OSCCON<0>) selects the system clock source that is used for the CPU and peripherals.

- When SCS = 0, the system clock source is determined by configuration of the FOSC<2:0> bits in the Configuration Word register (CONFIG).
- When SCS = 1, the system clock source is chosen by the internal oscillator frequency selected by the IRCF bits. After a Reset, SCS is always cleared.

| | |
|---|---|
| **Note:** | Any automatic clock switch, which may occur from Two-Speed Start-up or Fail-Safe Clock Monitor, does not update the SCS bit. The user can monitor the OSTS (OSCCON<3>) to determine the current system clock source. |

### 3.5.2 OSCILLATOR START-UP TIME-OUT STATUS BIT

The Oscillator Start-up Time-out Status (OSTS) bit (OSCCON<3>) indicates whether the system clock is running from the external clock source, as defined by the FOSC bits, or from internal clock source. In particular, OSTS indicates that the Oscillator Start-up Timer (OST) has timed out for LP, XT or HS modes.

## 3.6 Two-Speed Clock Start-up Mode

Two-Speed Start-up mode provides additional power savings by minimizing the latency between external oscillator start-up and code execution. In applications that make heavy use of the Sleep mode, Two-Speed Start-up will remove the external oscillator start-up time from the time spent awake and can reduce the overall power consumption of the device.

This mode allows the application to wake-up from Sleep, perform a few instructions using the INTOSC as the clock source and go back to Sleep without waiting for the primary oscillator to become stable.

| | |
|---|---|
| **Note:** | Executing a SLEEP instruction will abort the oscillator start-up time and will cause the OSTS bit (OSCCON<3>) to remain clear. |

When the PIC16F685/687/689/690 is configured for LP, XT or HS modes, the Oscillator Start-up Timer (OST) is enabled (see **Section 3.3.1 "Oscillator Start-up Timer (OST)"**). The OST timer will suspend program execution until 1024 oscillations are counted. Two-Speed Start-up mode minimizes the delay in code execution by operating from the internal oscillator as the OST is counting. When the OST count reaches 1024 and the OSTS bit (OSCCON<3>) is set, program execution switches to the external oscillator.

### 3.6.1 TWO-SPEED START-UP MODE CONFIGURATION

Two-Speed Start-up mode is configured by the following settings:

- IESO = 1 (CONFIG<10>) Internal/External Switchover bit.
- SCS = 0.
- FOSC configured for LP, XT or HS mode.

Two-Speed Start-up mode is entered after:

- Power-on Reset (POR) and, if enabled, after PWRT has expired, or
- Wake-up from Sleep.

If the external clock oscillator is configured to be anything other than LP, XT or HS mode, then Two-Speed Start-up is disabled. This is because the external clock oscillator does not require any stabilization time after POR or an exit from Sleep.

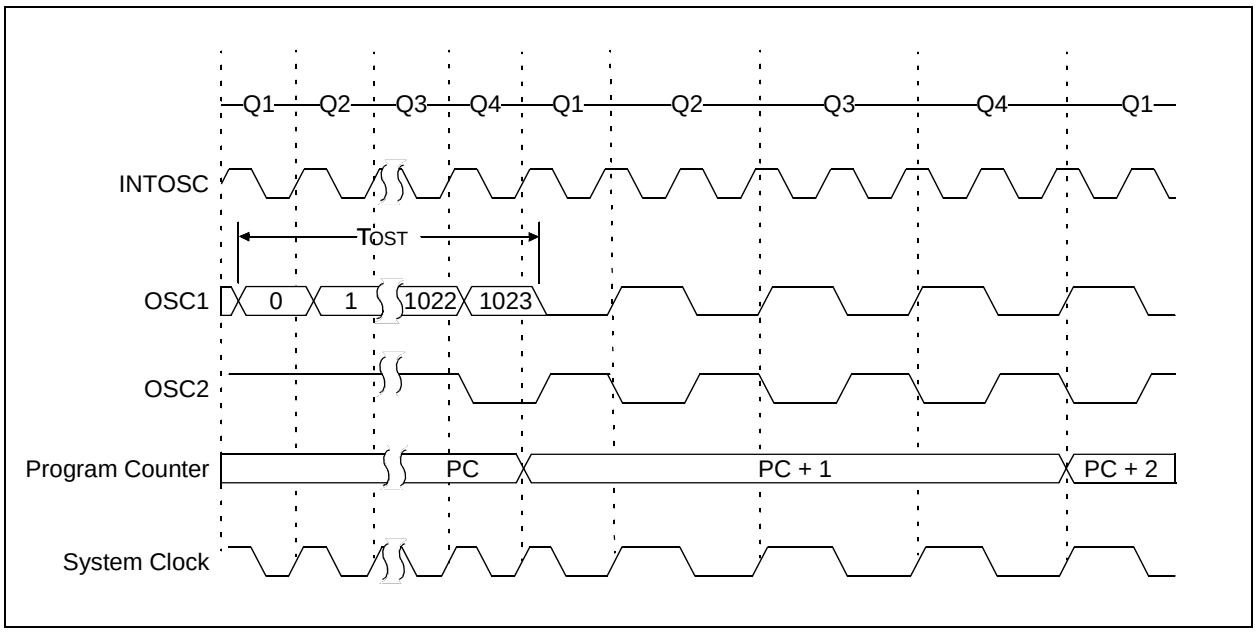
### 3.6.2 TWO-SPEED START-UP SEQUENCE

1. Wake-up from Power-on Reset or Sleep.
2. Instructions begin execution by the internal oscillator at the frequency set in the IRCF bits (OSCCON<6:4>).
3. OST enabled to count 1024 clock cycles.
4. OST timed out, wait for falling edge of the internal oscillator.
5. OSTS is set.
6. System clock held low until the next falling edge of new clock (LP, XT or HS mode).
7. System clock is switched to external clock source.

### 3.6.3 CHECKING EXTERNAL/INTERNAL CLOCK STATUS

Checking the state of the OSTS bit (OSCCON<3>) will confirm if the PIC16F685/687/689/690 is running from the external clock source as defined by the FOSC bits in the Configuration Word register (CONFIG) or the internal oscillator.

**FIGURE 3-7: TWO-SPEED START-UP**

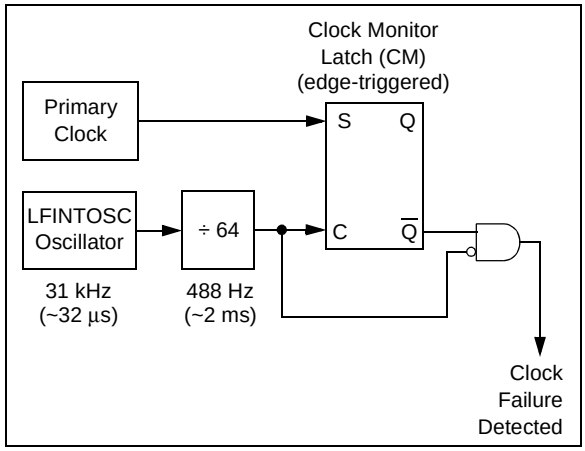


## 3.7 Fail-Safe Clock Monitor

The Fail-Safe Clock Monitor (FSCM) allows the device to continue operating should the external oscillator fail. The FSCM can detect oscillator failure anytime after the Oscillator Start-up Timer (OST) has expired. The FSCM is enabled by setting the FCMEN bit in the Configuration Word register (CONFIG). The FSCM is applicable to all external oscillator modes (LP, XT, HS, EC, RC and RCIO).

**FIGURE 3-8: FSCM BLOCK DIAGRAM**




### 3.7.1 FAIL-SAFE DETECTION

The FSCM module detects a failed oscillator by comparing the external oscillator to the FSCM sample clock. The sample clock is generated by dividing the LFINTOSC by 64. See Figure 3-8. Inside the fail detector block is a latch. The external clock sets the latch on each falling edge of the external clock. The sample clock clears the latch on each falling edge of the sample clock. If a sample clock edge occurs while the latch is cleared, a failure has occurred.

### 3.7.2 FAIL-SAFE OPERATION

When the external clock fails, the FSCM switches the device clock to an internal clock source and sets the OSFIF (PIR2<7>) flag. Setting this flag will generate an interrupt if the OSFIE (PIE2<7>) bit is also set. The device firmware can then take steps to mitigate the problems that may arise from a failed clock. The system clock will continue to be sourced from the internal clock source until the device firmware successfully restarts the external oscillator and switches back to external operation.

The internal clock source chosen by the FSCM is determined by the IRCF bits (OSCCON<6:4>). This allows the internal oscillator to be configured before a failure occurs.

### 3.7.3 FAIL-SAFE CONDITION CLEARING

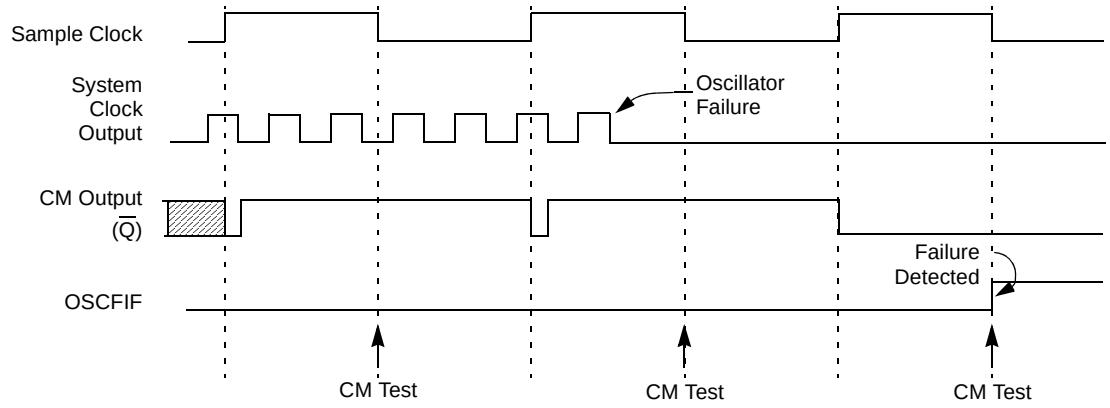
The Fail-Safe condition is cleared after a Reset, executing a `SLEEP` instruction or toggling the SCS bit (OSCCON<0>). When the SCS bit is toggled, the OST is restarted. While the OST is running, the device continues to operate from the INTOSC selected in OSCCON. When the OST times out, the Fail-Safe condition is cleared and the device will be operating from the external clock source. The Fail-Safe condition must be cleared before the OSFIF flag can be cleared.

### 3.7.4 RESET OR WAKE-UP FROM SLEEP

The FSCM is designed to detect an oscillator failure after the Oscillator Start-up Time (OST) has expired. The OST is used after waking up from Sleep and after any type of Reset. The OST is not used with the EC or RC clock modes so the FSCM will be active as soon as the Reset or wake-up have completed. When the FSCM is enabled, the Two-Speed Start-up is also enabled. Therefore, the device will always be executing code while the OST is operating.

| | |
|---|---|
| **Note:** | Due to the wide range of oscillator start-up times, the Fail-Safe circuit is not active during oscillator start-up (i.e., after exiting Reset or Sleep). After an appropriate amount of time, the user should check the OSTS bit (OSCCON<3>) to verify the oscillator start-up and system clock switchover has successfully completed. |

**FIGURE 3-9: FSCM TIMING DIAGRAM**




**Note:** The system clock is normally at a much higher frequency than the sample clock. The relative frequencies in this example have been chosen for clarity.

**REGISTER 3-2: OSCCON – OSCILLATOR CONTROL REGISTER (ADDRESS: 8Fh)**

| U-0 | R/W-1 | R/W-1 | R/W-0 | R-1 | R-0 | R-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | IRCF2 | IRCF1 | IRCF0 | OSTS(1) | HTS | LTS | SCS |
| bit 7 | | | | | | | bit 0 |

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

bit 6-4 **IRCF<2:0>:** Internal Oscillator Frequency Select bits
000 = 31 kHz
001 = 125 kHz
010 = 250 kHz
011 = 500 kHz
100 = 1 MHz
101 = 2 MHz
110 = 4 MHz (default)
111 = 8 MHz

bit 3 **OSTS:** Oscillator Start-up Time-out Status bit(1)
1 = Device is running from the external clock defined by FOSC<2:0>
0 = Device is running from the internal oscillator (HFINTOSC or LFINTOSC)

bit 2 **HTS:** HFINTOSC (High Frequency – 8 MHz to 125 kHz) Status bit
1 = HFINTOSC is stable
0 = HFINTOSC is not stable

bit 1 **LTS:** LFINTOSC (Low Frequency – 31 kHz) Stable bit
1 = LFINTOSC is stable
0 = LFINTOSC is not stable

bit 0 **SCS:** System Clock Select bit
1 = Internal oscillator is used for system clock
0 = Clock source defined by FOSC<2:0>

**Note 1:** Bit resets to '0' with Two-Speed Start-up and LP, XT or HS selected as the Oscillator mode or Fail-Safe mode is enabled.

| **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 |

**TABLE 3-2: SUMMARY OF REGISTERS ASSOCIATED WITH CLOCK SOURCES**

| 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(1) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0Ch | PIR1 | — | ADIF | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | -000 0000 | -000 0000 |
| 8Ch | PIE1 | — | ADIE | RCIE | TXIE | SSPIE | CCP1IE | TMR2IE | TMR1IE | -000 0000 | -000 0000 |
| 8Fh | OSCCON | — | IRCF2 | IRCF1 | IRCF0 | OSTS | HTS | LTS | SCS | -110 x000 | -110 x000 |
| 90h | OSCTUNE | — | — | — | TUN4 | TUN3 | TUN2 | TUN1 | TUN0 | ---0 0000 | ---u uuuu |
| 2007h(2) | CONFIG | CPD | CP | MCLRE | $\overline{\text{PWRTE}}$ | WDTE | FOSC2 | FOSC1 | FOSC0 | — | — |

**Legend:** x = unknown, u = unchanged, – = unimplemented locations read as '0'. Shaded cells are not used by oscillators.
**Note 1:** Other (non Power-up) Resets include $\overline{\text{MCLR}}$ Reset and Watchdog Timer Reset during normal operation.
**2:** See Register 14-1 for operation of all Configuration Word register bits.

**NOTES:**

## 4.0 I/O PORTS

There are as many as eighteen general purpose I/O pins available. Depending on which peripherals are enabled, some or all of the pins may not be available as general purpose I/O. In general, when a peripheral is enabled, the associated pin may not be used as a general purpose I/O pin.

### 4.1 PORTA and the TRISA Registers

PORTA is a 6-bit wide, bidirectional port. The corresponding data direction register is TRISA (Register 4-2). 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). The exception is RA3, which is input only and its TRIS bit will always read as '1'. Example 4-1 shows how to initialize PORTA.

Reading the PORTA register (Register 4-1) 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, this value is modified and then written to the port data latch. RA3 reads '0' when MCLRE = 1.

The TRISA register controls the direction of the PORTA 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. I/O pins configured as analog input always read '0'.

| **Note:** | The ANSEL (11Eh) register must be initialized to configure an analog channel as a digital input. Pins configured as analog inputs will read '0'. |
|---|---|

**EXAMPLE 4-1: INITIALIZING PORTA**

```
BCF   STATUS,RP0  ;Bank 0
BCF   STATUS,RP1  ;
CLRF  PORTA       ;Init PORTA
BSF   STATUS,RP1  ;Bank 2
CLRF  ANSEL       ;digital I/O
BSF   STATUS,RP0  ;Bank 1
BCF   STATUS,RP1  ;
MOVLW 0Ch         ;Set RA<3:2> as inputs
MOVWF TRISA       ;and set RA<5:4,1:0>
                  ;as outputs
BCF   STATUS,RP0  ;Bank 0
```

### 4.2 Additional Pin Functions

Every PORTA pin on the PIC16F685/687/689/690 has an interrupt-on-change option and a weak pull-up option. RA0 also has an Ultra Low-Power Wake-up option. The next three sections describe these functions.

#### 4.2.1 WEAK PULL-UPS

Each of the PORTA pins, except RA3, has an individually configurable internal weak pull-up. Control bits WPUAx enable or disable each pull-up. Refer to Register 4-3. Each weak pull-up is automatically turned off when the port pin is configured as an output. The pull-ups are disabled on a Power-on Reset by the RABPU bit (OPTION_REG<7>). A weak pull-up is automatically enabled for RA3 when configured as MCLR and disabled when RA3 is an I/O. There is no software control of the MCLR pull-up.

**REGISTER 4-1: PORTA – PORTA REGISTER (ADDRESS: 05h OR 105h)**

| U-0 | U-0 | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x |
|---|---|---|---|---|---|---|---|
| — | — | RA5 | RA4 | RA3 | RA2 | RA1 | RA0 |
| bit 7 | | | | | | | bit 0 |

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

bit 5-0: **RA<5:0>**: PORTA I/O Pin bit

1 = Port pin is > $V_{IH}$

0 = Port pin is < $V_{IL}$

| **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 |

**REGISTER 4-2: TRISA – PORTA TRI-STATE PORTA REGISTER (ADDRESS: 85h OR 185h)**

| U-0 | U-0 | R/W-1 | R/W-1 | R-1 | R/W-1 | R/W-1 | R/W-1 |
|---|---|---|---|---|---|---|---|
| — | — | TRISA5 | TRISA4 | TRISA3 | TRISA2 | TRISA1 | TRISA0 |
| bit 7 | | | | | | | bit 0 |

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

bit 5-0 **TRISA<5:0>:** PORTA Tri-State Control bit
1 = PORTA pin configured as an input (tri-stated)
0 = PORTA pin configured as an output

**Note:** TRISA<5:4> always reads '1' in XT, HS and LP OSC modes.

| **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 |

**REGISTER 4-3: WPUA – WEAK PULL-UP PORTA REGISTER (ADDRESS: 95h)**

| U-0 | U-0 | R/W-1 | R/W-1 | U-0 | R/W-1 | R/W-1 | R/W-1 |
|---|---|---|---|---|---|---|---|
| — | — | WPUA5 | WPUA4 | — | WPUA2 | WPUA1 | WPUA0 |
| bit 7 | | | | | | | bit 0 |

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

bit 5-4 **WPUA<5:4>:** Weak Pull-up Register bit
1 = Pull-up enabled
0 = Pull-up disabled

bit 3 **Unimplemented:** Read as '0'

bit 2-0 **WPUA<2:0>:** Weak Pull-up Register bit
1 = Pull-up enabled
0 = Pull-up disabled

**Note 1:** Global $\overline{\text{RABPU}}$ must be enabled for individual pull-ups to be enabled.

**2:** The weak pull-up device is automatically disabled if the pin is in Output mode (TRISA = 0).

**3:** The RA3 pull-up is enabled when configured as $\overline{\text{MCLR}}$ and disabled as an I/O in the Configuration Word.

**4:** WPUA<5:4> always reads '1' in XT, HS and LP OSC modes.

| **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 |

### 4.2.2 INTERRUPT-ON-CHANGE

Each of the PORTA pins is individually configurable as an interrupt-on-change pin. Control bits IOCAx enable or disable the interrupt function for each pin. Refer to Register 4-4. The interrupt-on-change is disabled on a Power-on Reset.

For enabled interrupt-on-change pins, the values are compared with the old value latched on the last read of PORTA. The 'mismatch' outputs of the last read are OR'd together to set the PORTA Change Interrupt Flag bit (RABIF) in the INTCON register (Register 2-3).

This interrupt can wake the device from Sleep. The user, in the Interrupt Service Routine, clears the interrupt by:

a) Any read or write of PORTA. This will end the mismatch condition, then,

b) Clear the flag bit RABIF.

A mismatch condition will continue to set flag bit RABIF. Reading PORTA will end the mismatch condition and allow flag bit RABIF to be cleared. The latch holding the last read value is not affected by a $\overline{MCLR}$ nor BOR Reset. After these Resets, the RABIF flag will continue to be set if a mismatch is present.

| **Note:** | If a change on the I/O pin should occur when the read operation is being executed (start of the Q2 cycle), then the RABIF interrupt flag may not get set. |
|---|---|

**REGISTER 4-4: IOCA – INTERRUPT-ON-CHANGE PORTA REGISTER (ADDRESS: 96h)**

| U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | — | IOCA5 | IOCA4 | IOCA3 | IOCA2 | IOCA1 | IOCA0 |
| bit 7 | | | | | | | bit 0 |

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

bit 5-0 **IOCA<5:0>:** Interrupt-on-change PORTA Control bit

1 = Interrupt-on-change enabled
0 = Interrupt-on-change disabled

**Note 1:** Global Interrupt Enable (GIE) must be enabled for individual interrupts to be recognized.

| **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 |

### 4.2.3 ULTRA LOW-POWER WAKE-UP

The Ultra Low-Power Wake-up (ULPWU) on RA0 allows a slow falling voltage to generate an interrupt-on-change on RA0 without excess current consumption. The mode is selected by setting the ULPWUE bit (PCON<5>). This enables a small current sink which can be used to discharge a capacitor on RA0.

To use this feature, the RA0/AN0/C1IN+/ICSPDAT/ULPWU pin is configured to output '1' to charge the capacitor, interrupt-on-change for RA0 is enabled, and RA0 is configured as an input. The ULPWUE bit is set to begin the discharge and a SLEEP instruction is performed. When the voltage on RA0 drops below VIL, an interrupt will be generated which will cause the device to wake-up. Depending on the state of the GIE bit (INTCON<7>), the device will either jump to the interrupt vector (0004h) or execute the next instruction when the interrupt event occurs. See **Section 4.2.2 "Interrupt-on-change"** and **Section 14.3.3 "PORTA/PORTB Interrupt"** for more information.

This feature provides a low-power technique for periodically waking up the device from Sleep. The time-out is dependent on the discharge time of the RC circuit on RA0. See Example 4-2 for initializing the Ultra Low-Power Wake-up module.

The series resistor provides overcurrent protection for the RA0/AN0/C1IN+/ICSPDAT/ULPWU pin and can allow for software calibration of the time-out (see Figure 4-1). A timer can be used to measure the charge time and discharge time of the capacitor. The charge time can then be adjusted to provide the desired interrupt delay. This technique will compensate for the affects of temperature, voltage and component accuracy. The Ultra Low-Power Wake-up peripheral can also be configured as a simple Programmable Low Voltage Detect or temperature sensor.

| **Note:** | For more information, refer to AN879, "*Using the Microchip Ultra Low-Power Wake-up Module*" Application Note (DS00879). |
|---|---|

**EXAMPLE 4-2: ULTRA LOW-POWER WAKE-UP INITIALIZATION**

```
BCF     STATUS,RP0  ;Bank 0
BCF     STATUS,RP1  ;
BSF     PORTA,0     ;Set RA0 data latch
BSF     STATUS,RP1  ;BANK 2
BCF     ANSEL,0     ;RA0 to digital I/O
BSF     STATUS,RP0  ;BANK 1
BCF     STATUS,RP1  ;
BCF     TRISA,0     ;Output high to
CALL    CapDelay    ;charge capacitor
BSF     PCON,ULPWUE ;Enable ULP Wake-up
BSF     IOCA,0      ;Select RA0 IOC
BSF     TRISA,0     ;RA0 to input
MOVLW   B'10001000' ;Enable interrupt
MOVWF   INTCON      ;and clear flag
BCF     STATUS,RP0  ;BANK 0
SLEEP               ;Wait for IOC
```

### 4.2.4 PIN DESCRIPTIONS AND DIAGRAMS

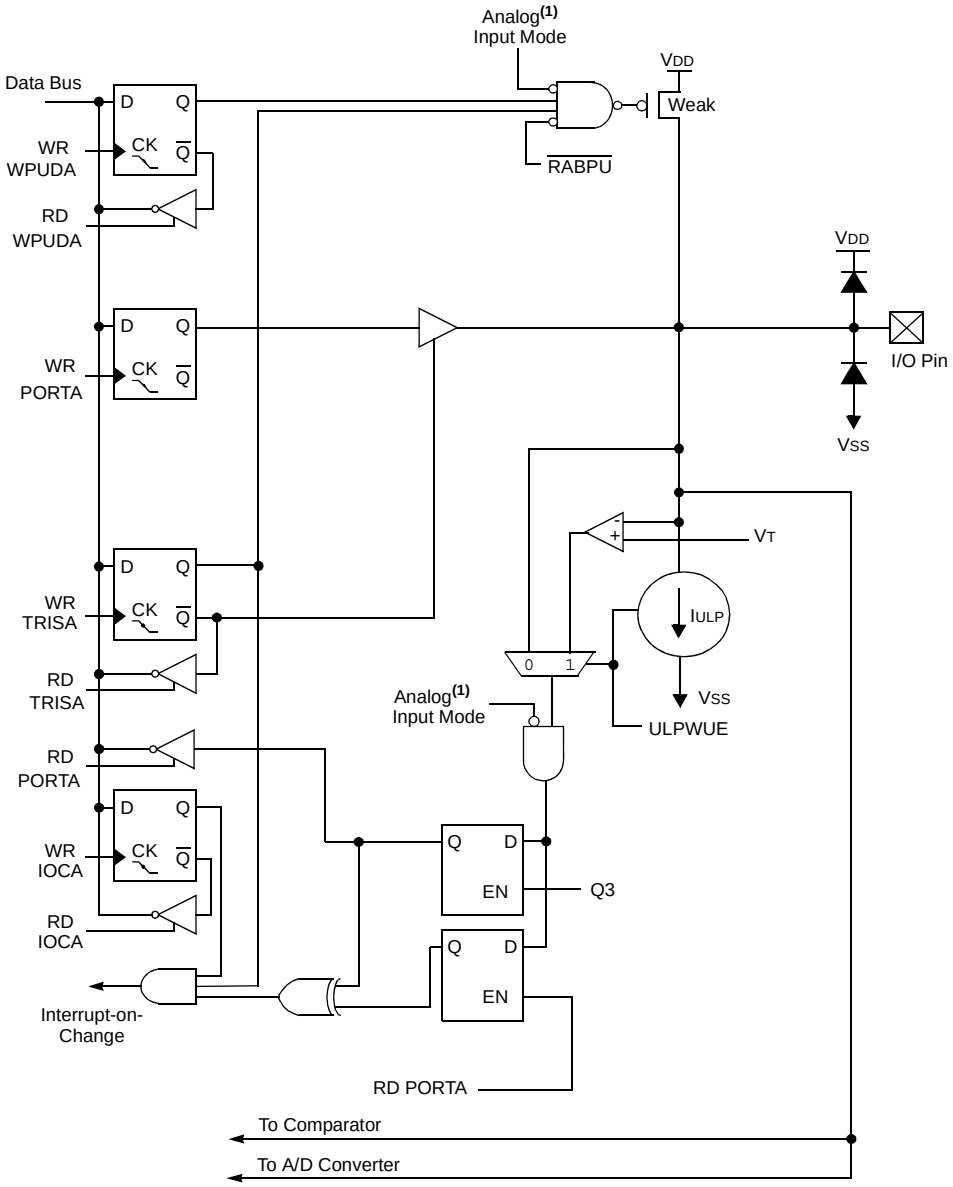
Each PORTA pin is multiplexed with other functions. The pins and their combined functions are briefly described here. For specific information about individual functions such as the comparator or the A/D Converter, refer to the appropriate section in this data sheet.

#### 4.2.4.1 RA0/AN0/C1IN+/ICSPDAT/ULPWU

Figure 4-2 shows the diagram for this pin. The RA0/AN0/C1IN+/ICSPDAT/ULPWU pin is configurable to function as one of the following:

- a general purpose I/O
- an analog input for the A/D
- an analog input to Comparator 1
- In-Circuit Serial Programming data
- an analog input for the Ultra Low-Power Wake-up

**FIGURE 4-1: BLOCK DIAGRAM OF RA0**



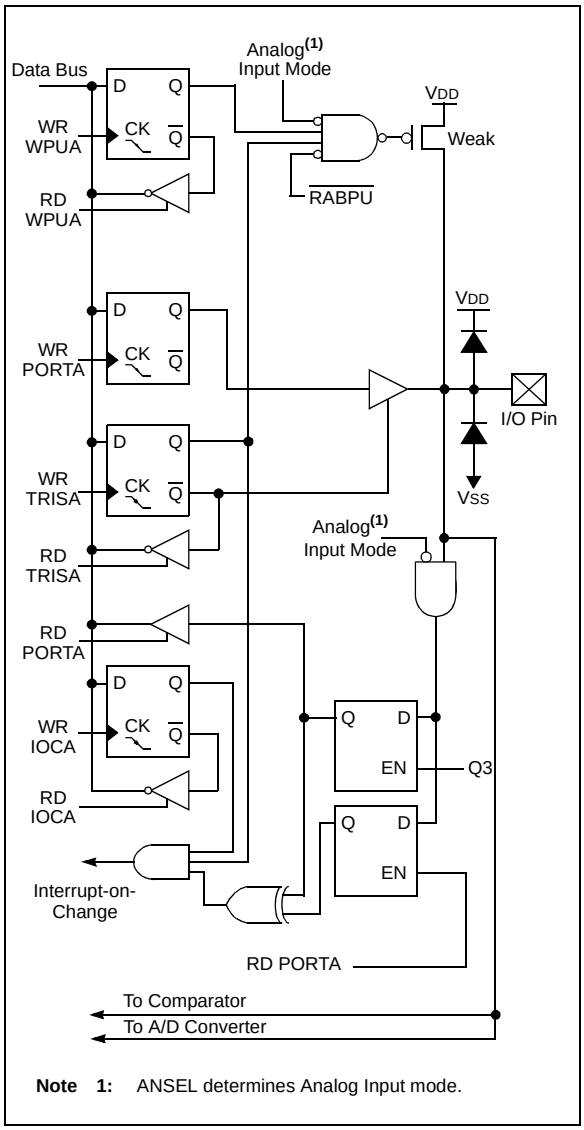

**Note 1:** ANSEL determines Analog Input mode.

#### 4.2.4.2 RA1/AN1/C12IN-/VREF/ICSPCLK

Figure 4-2 shows the diagram for this pin. The RA1/AN1/C12IN-/VREF/ICSPCLK pin is configurable to function as one of the following:

- a general purpose I/O
- an analog input for the A/D
- an analog input to Comparator 1 or 2
- a voltage reference input for the A/D
- In-Circuit Serial Programming clock

**FIGURE 4-2: BLOCK DIAGRAM OF RA1**



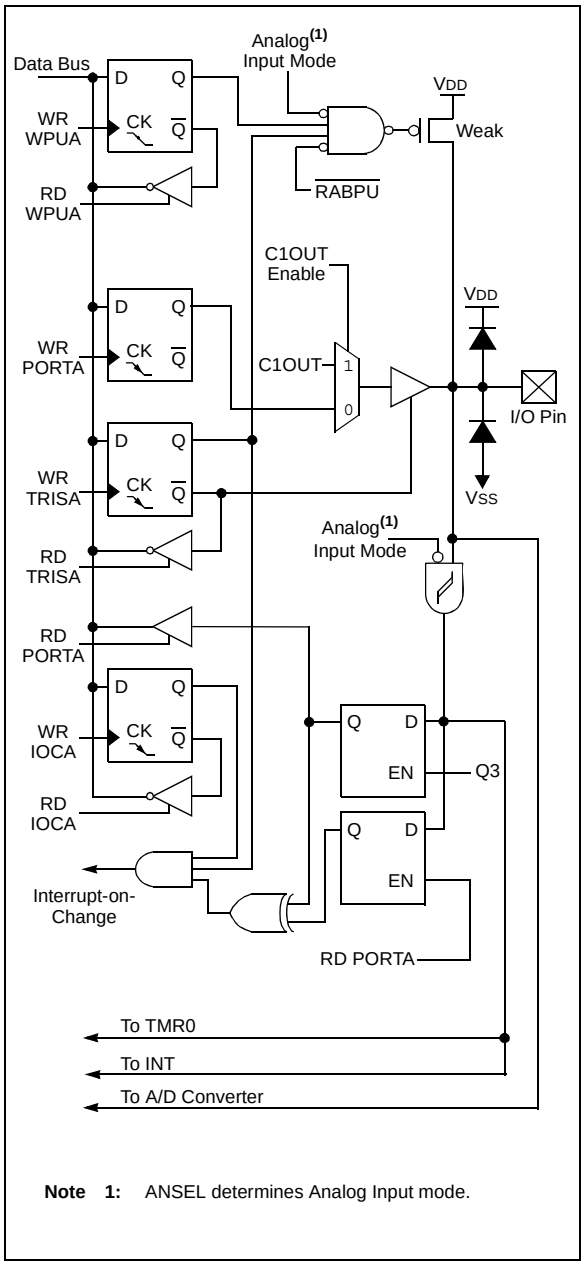

**Note 1:** ANSEL determines Analog Input mode.

#### 4.2.4.3 RA2/AN2/T0CKI/INT/C1OUT

Figure 4-3 shows the diagram for this pin. The RA2/AN2/T0CKI/INT/C1OUT pin is configurable to function as one of the following:

- a general purpose I/O
- an analog input for the A/D
- the clock input for TMR0
- an external edge triggered interrupt
- a digital output from Comparator 1

**FIGURE 4-3: BLOCK DIAGRAM OF RA2**




**Note 1:** ANSEL determines Analog Input mode.

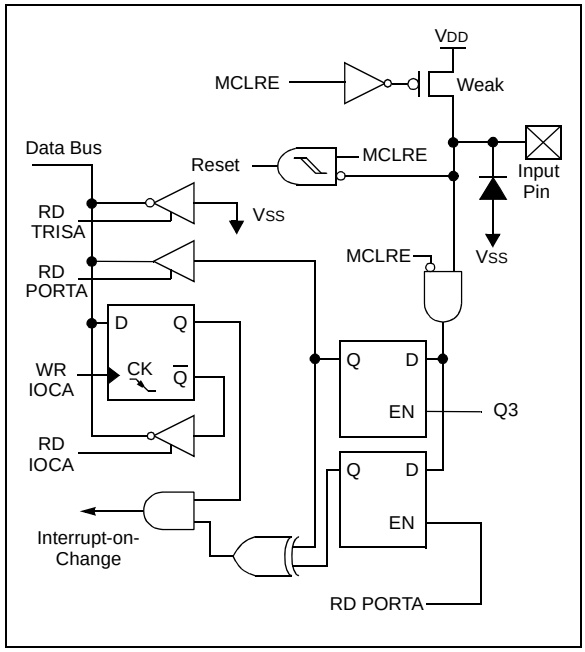
 

#### 4.2.4.4 RA3/$\overline{MCLR}$/$V_{PP}$

Figure 4-4 shows the diagram for this pin. The RA3/$\overline{MCLR}$/$V_{PP}$ pin is configurable to function as one of the following:

- a general purpose input
- as Master Clear Reset with weak pull-up

**FIGURE 4-4: BLOCK DIAGRAM OF RA3**

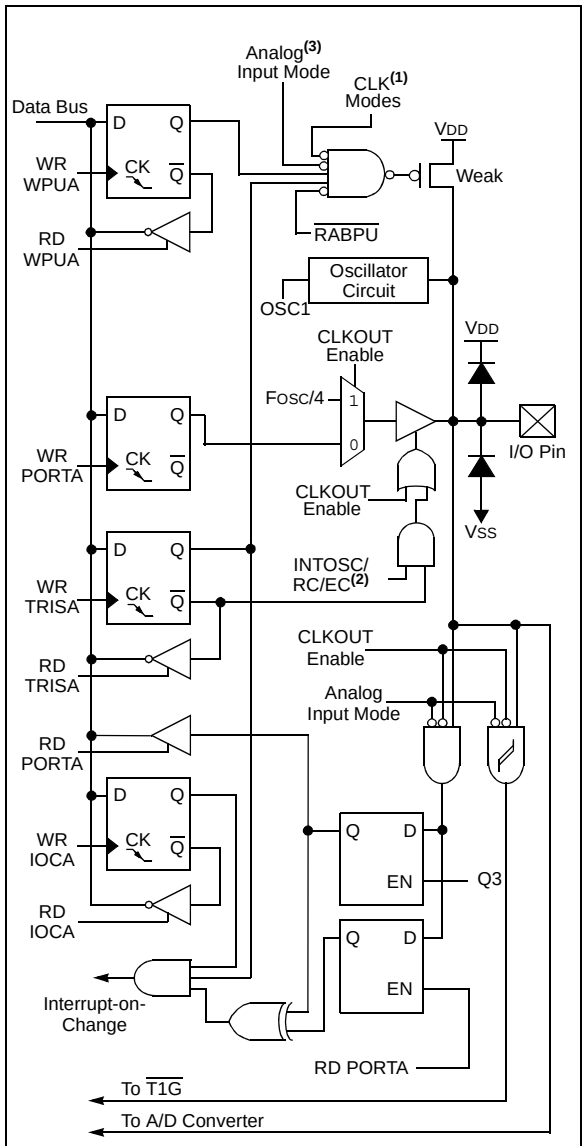



#### 4.2.4.5 RA4/AN3/$\overline{T1G}$/OSC2/CLKOUT

Figure 4-5 shows the diagram for this pin. The RA4/AN3/$\overline{T1G}$/OSC2/CLKOUT pin is configurable to function as one of the following:

- a general purpose I/O
- an analog input for the A/D
- a TMR1 gate input
- a crystal/resonator connection
- a clock output

**FIGURE 4-5: BLOCK DIAGRAM OF RA4**




**Note 1:** CLK modes are XT, HS, LP, LPTMR1 and CLKOUT Enable.

**2:** With CLKOUT option.

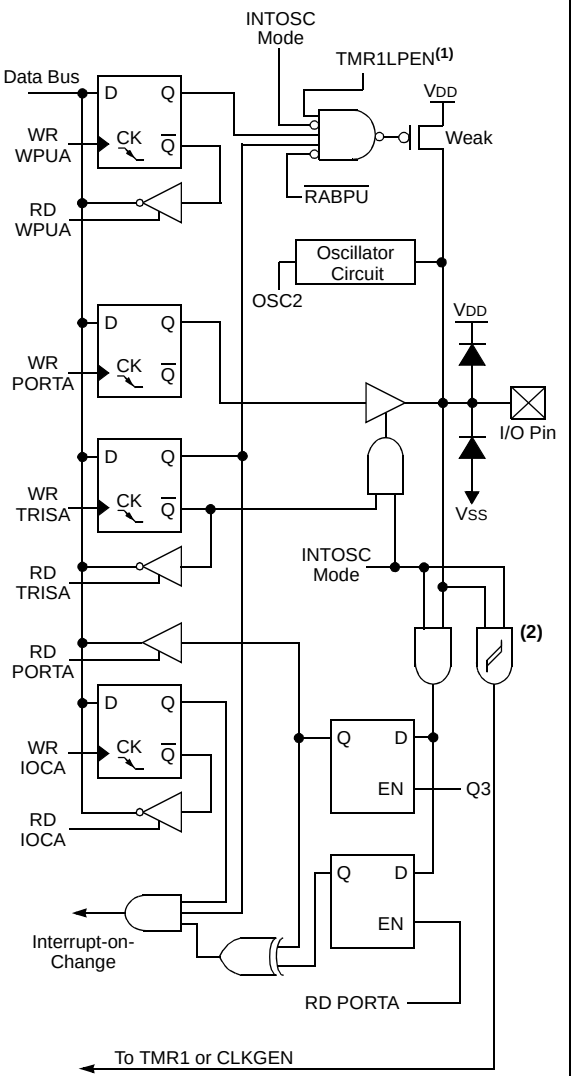
**3:** ANSEL determines Analog Input mode.

#### 4.2.4.6 RA5/T1CKI/OSC1/CLKIN

Figure 4-6 shows the diagram for this pin. The RA5/T1CKI/OSC1/CLKIN pin is configurable to function as one of the following:

- a general purpose I/O
- a TMR1 clock input
- a crystal/resonator connection
- a clock input

**FIGURE 4-6: BLOCK DIAGRAM OF RA5**




**Note 1:** Timer1 LP Oscillator enabled.

**2:** When using Timer1 with LP oscillator, the Schmitt Trigger is bypassed.

**TABLE 4-1: SUMMARY OF REGISTERS ASSOCIATED WITH PORTA**

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 05h/105h | PORTA | — | — | RA5 | RA4 | RA3 | RA2 | RA1 | RA0 | --xx xxxx | --uu uuuu |
| 0Bh/8Bh/ 10Bh/18Bh | INTCON | GIE | PEIE | T0IE | INTE | RABIE | T0IF | INTF | RABIF | 0000 000x | 0000 000x |
| 10h | T1CON | T1GINV | TMR1GE | T1CKPS1 | T1CKPS0 | T1OSCEN | $\overline{\text{T1SYNC}}$ | TMR1CS | TMR1ON | 0000 0000 | uuuu uuuu |
| 14h | SSPCON | WCOL | SSPOV | SSPEN | CKP | SSPM3 | SSPM2 | SSPM1 | SSPM0 | 0000 0000 | 0000 0000 |
| 1Fh | ADCON0 | ADFM | VCFG | CHS3 | CHS2 | CHS1 | CHS0 | GO/$\overline{\text{DONE}}$ | ADON | 0000 0000 | 0000 0000 |
| 81h/181h | OPTION_REG | $\overline{\text{RABPU}}$ | INTEDG | T0CS | T0SE | PSA | PS2 | PS1 | PS0 | 1111 1111 | 1111 1111 |
| 85h/185h | TRISA | — | — | TRISA5 | TRISA4 | TRISA3 | TRISA2 | TRISA1 | TRISA0 | --11 1111 | --11 1111 |
| 95h | WPUA | — | — | WPUA5 | WPUA4 | — | WPUA2 | WPUA1 | WPUA0 | --11 -111 | --11 -111 |
| 96h | IOCA | — | — | IOCA5 | IOCA4 | IOCA3 | IOCA2 | IOCA1 | IOCA0 | --00 0000 | --00 0000 |
| 119h | CM1CON0 | C1ON | C1OUT | C1OE | C1POL | — | C1R | C1CH1 | C1CH0 | 0000 -000 | 0000 -000 |
| 11Eh | ANSEL | ANS7 | ANS6 | ANS5 | ANS4 | ANS3 | ANS2 | ANS1 | ANS0 | 1111 1111 | 1111 1111 |

**Legend:** x = unknown, u = unchanged, – = unimplemented locations read as '0'. Shaded cells are not used by PORTA.

## 4.3 PORTB and TRISB Registers

PORTB is a 4-bit wide, bidirectional port. The corresponding data direction register is TRISB (Register 4-6). Setting a TRISB bit (= 1) will make the corresponding PORTB pin an input (i.e., put the corresponding output driver in a High-impedance mode). Clearing a TRISB bit (= 0) will make the corresponding PORTB pin an output (i.e., put the contents of the output latch on the selected pin). Example 4-3 shows how to initialize PORTB. Reading the PORTB register (Register 4-5) 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, this value is modified and then written to the port data latch.

The TRISB register controls the direction of the PORTB pins, even when they are being used as analog inputs. The user must ensure the bits in the TRISB register are maintained set when using them as analog inputs. I/O pins configured as analog input always read '0'.

**EXAMPLE 4-3: INITIALIZING PORTB**

```
BCF     STATUS,RP0  ;Bank 0
BCF     STATUS,RP1  ;
CLRF    PORTB       ;Init PORTB
BSF     STATUS,RP0  ;Bank 1
MOVLW   FFh         ;Set RB<7:4> as inputs
MOVWF   TRISB       ;
BCF     STATUS,RP0  ;Bank 0
```

| **Note:** | The ANSELH (11Fh) register must be initialized to configure an analog channel as a digital input. Pins configured as analog inputs will read '0'. |
|---|---|

## 4.4 Additional PORTB Pin Functions

PORTB pins RB<7:4> on the PIC16F685/687/689/690 have an interrupt-on-change option and a weak pull-up option. The following three sections describe these PORTB pin functions.

### 4.4.1 WEAK PULL-UPS

Each of the PORTB pins has an individually configurable internal weak pull-up. Control bits WPUB<7:4> enable or disable each pull-up. Refer to Register 4-7. Each weak pull-up is automatically turned off when the port pin is configured as an output. The pull-ups are disabled on a Power-on Reset by the $\overline{\text{RABPU}}$ bit (OPTION_REG<7>).

### 4.4.2 INTERRUPT-ON-CHANGE

Four of the PORTB pins are individually configurable as an interrupt-on-change pin. Control bits IOCB<7:4> enable or disable the interrupt function for each pin. Refer to Register 4-8. The interrupt-on-change feature is disabled on a Power-on Reset.

For enabled interrupt-on-change pins, the values are compared with the old value latched on the last read of PORTB. The 'mismatch' outputs of the last read are OR'd together to set the PORTB Change Interrupt flag bit (RABIF) in the INTCON register (Register 2-3).

This interrupt can wake the device from Sleep. The user, in the Interrupt Service Routine, clears the interrupt by:

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

b) Clear the flag bit RABIF.

A mismatch condition will continue to set flag bit RABIF. Reading or writing PORTB will end the mismatch condition and allow flag bit RABIF to be cleared. The latch holding the last read value is not affected by a $\overline{\text{MCLR}}$ nor Brown-out Reset. After these Resets, the RABIF flag will continue to be set if a mismatch is present.

| **Note:** | If a change on the I/O pin should occur when the read operation is being executed (start of the Q2 cycle), then the RABIF interrupt flag may not get set. Furthermore, since a read or write on a port affects all bits of that port, care must be taken when using multiple pins in Interrupt-on-change mode. Changes on one pin may not be seen while servicing changes on another pin. |
|---|---|

**REGISTER 4-5: PORTB – PORTB REGISTER (ADDRESS: 06h OR 106h)**

| R/W-x | R/W-x | R/W-x | R/W-x | U-0 | U-0 | U-0 | U-0 |
|---|---|---|---|---|---|---|---|
| RB7 | RB6 | RB5 | RB4 | — | — | — | — |
| bit 7 | | | | | | | bit 0 |

bit 7-4 **RB<7:4>**: PORTB I/O Pin bits
1 = Port pin is > VIH
0 = Port pin is < VIL

bit 3-0 **Unimplemented:** Read as '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 |

**REGISTER 4-6: TRISB – TRI-STATE PORTB REGISTER (ADDRESS: 86h OR 186h)**

| R/W-1 | R/W-1 | R/W-1 | R/W-1 | U-0 | U-0 | U-0 | U-0 |
|---|---|---|---|---|---|---|---|
| TRISB7 | TRISB6 | TRISB5 | TRISB4 | — | — | — | — |
| bit 7 | | | | | | | bit 0 |

bit 7-4 **TRISB<7:4>:** PORTB Tri-State Control bits
1 = PORTB pin configured as an input (tri-stated)
0 = PORTB pin configured as an output

bit 3-0 **Unimplemented:** Read as '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 |

**REGISTER 4-7: WPUB – WEAK PULL-UP PORTB REGISTER (ADDRESS: 115h)**

| R/W-1 | R/W-1 | R/W-1 | R/W-1 | U-0 | U-0 | U-0 | U-0 |
|---|---|---|---|---|---|---|---|
| WPUB7 | WPUB6 | WPUB5 | WPUB4 | — | — | — | — |
| bit 7 | | | | | | | bit 0 |

bit 7-4 **WPUB<7:4>:** Weak Pull-up Register bits
1 = Pull-up enabled
0 = Pull-up disabled

bit 3-0 **Unimplemented:** Read as '0'

**Note 1:** Global $\overline{\text{RABPU}}$ must be enabled for individual pull-ups to be enabled.

**2:** The weak pull-up device is automatically disabled if the pin is in Output mode (TRISB<7:4> = 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 |

**REGISTER 4-8: IOCB – INTERRUPT-ON-CHANGE PORTB REGISTER (ADDRESS: 116h)**

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | U-0 | U-0 | U-0 | U-0 |
|---|---|---|---|---|---|---|---|
| IOCB7 | IOCB6 | IOCB5 | IOCB4 | — | — | — | — |
| bit 7 | | | | | | | bit 0 |

bit 7-4 **IOCB<7:4>:** Interrupt-on-Change bits
1 = Interrupt-on-change enabled
0 = Interrupt-on-change disabled

bit 3-0 **Unimplemented:** Read as '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 |

### 4.4.3 PIN DESCRIPTIONS AND DIAGRAMS

Each PORTB pin is multiplexed with other functions. The pins and their combined functions are briefly described here. For specific information about individual functions such as the SSP, $I^2C$ or interrupts, refer to the appropriate section in this data sheet.

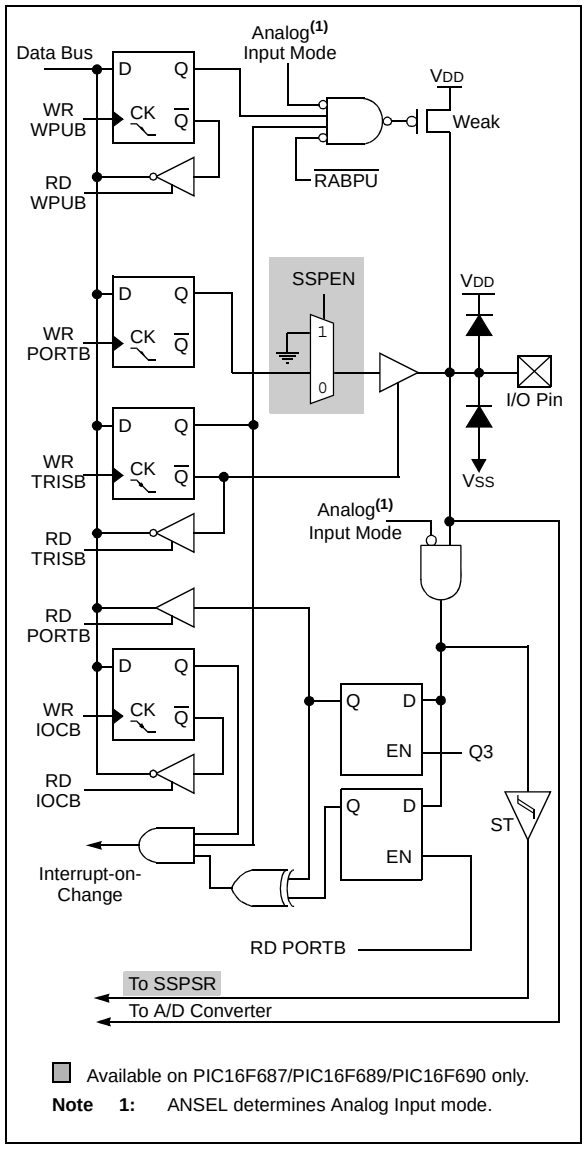
#### 4.4.3.1 RB4/AN10/SDI/SDA

Figure 4-7 shows the diagram for this pin. The RB4/AN10/SDI/SDA**(1)** pin is configurable to function as one of the following:

- a general purpose I/O
- an analog input for the A/D
- a SPI data I/O
- an $I^2C$ data I/O

**Note 1:** SDI and SDA are available on PIC16F687/PIC16F689/PIC16F690 only.

**FIGURE 4-7: BLOCK DIAGRAM OF RB4**



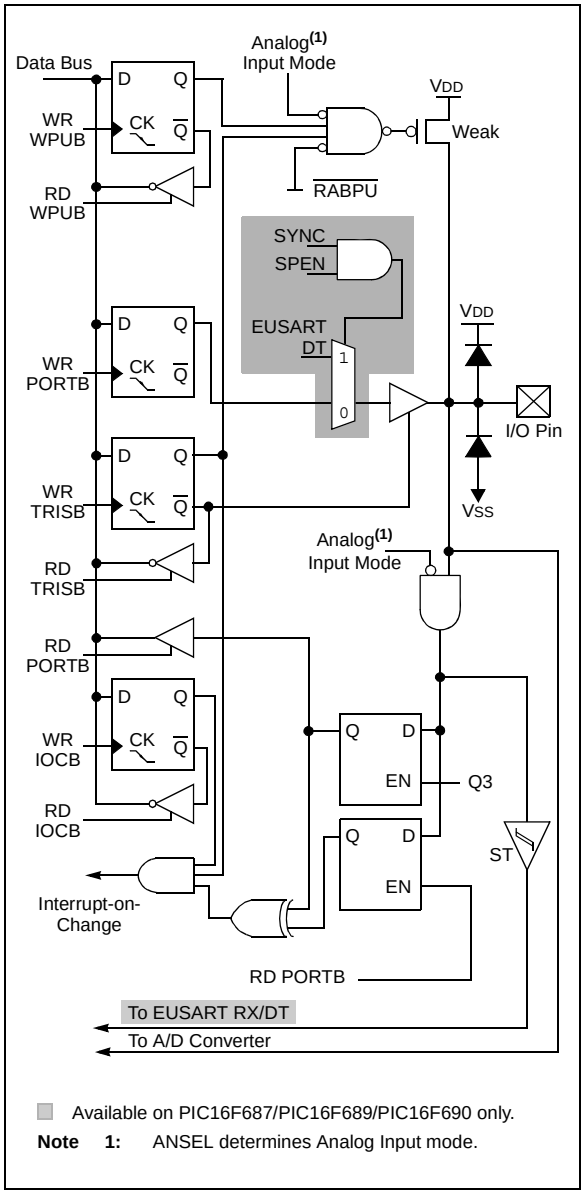
#### 4.4.3.2 RB5/AN11/RX/DT

Figure 4-8 shows the diagram for this pin. The RB5/AN11/RX/DT[1] pin is configurable to function as one of the following:

- a general purpose I/O
- an analog input for the A/D
- an asynchronous serial input
- a synchronous serial data I/O

**Note 1:** RX and DT are available on PIC16F687/PIC16F689/PIC16F690 only.

**FIGURE 4-8: BLOCK DIAGRAM OF RB5**



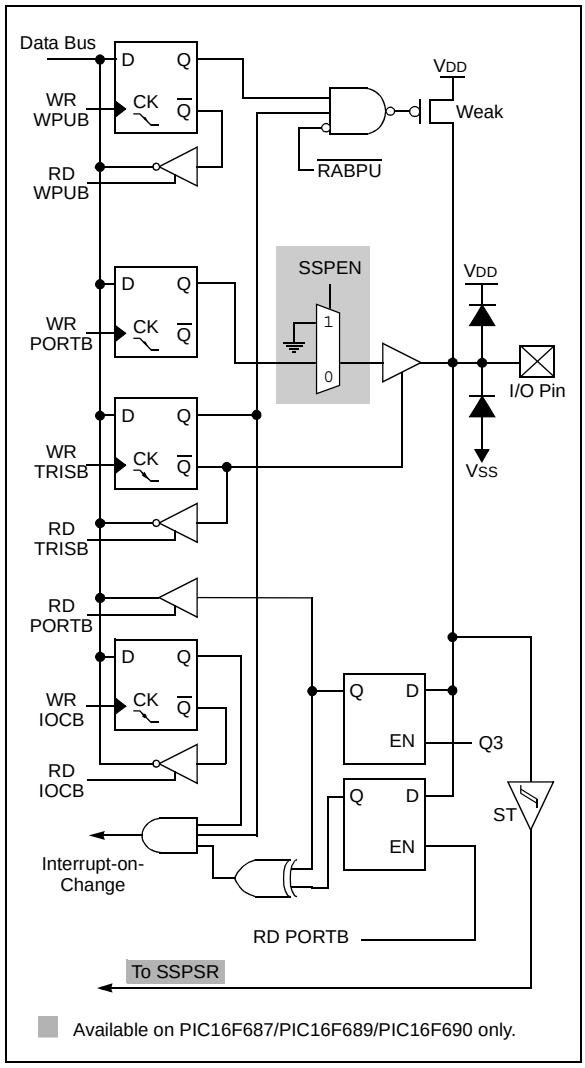
#### 4.4.3.3 RB6/SCK/SCL

Figure 4-9 shows the diagram for this pin. The RB6/SCK/SCL(1) pin is configurable to function as one of the following:

- a general purpose I/O
- a SPI™ clock
- an I²C™ clock

| | |
|---|---|
| **Note 1:** | SCK and SCL are available on PIC16F687/PIC16F689/PIC16F690 only. |

**FIGURE 4-9: BLOCK DIAGRAM OF RB6**



#### 4.4.3.4 RB7/TX/CK

Figure 4-10 shows the diagram for this pin. The RB7/TX/CK[(1)] pin is configurable to function as one of the following:

- a general purpose I/O
- an asynchronous serial output
- a synchronous clock I/O

**Note 1:** TX and CK are available on PIC16F687/PIC16F689/PIC16F690 only.

**FIGURE 4-10: BLOCK DIAGRAM OF RB7**

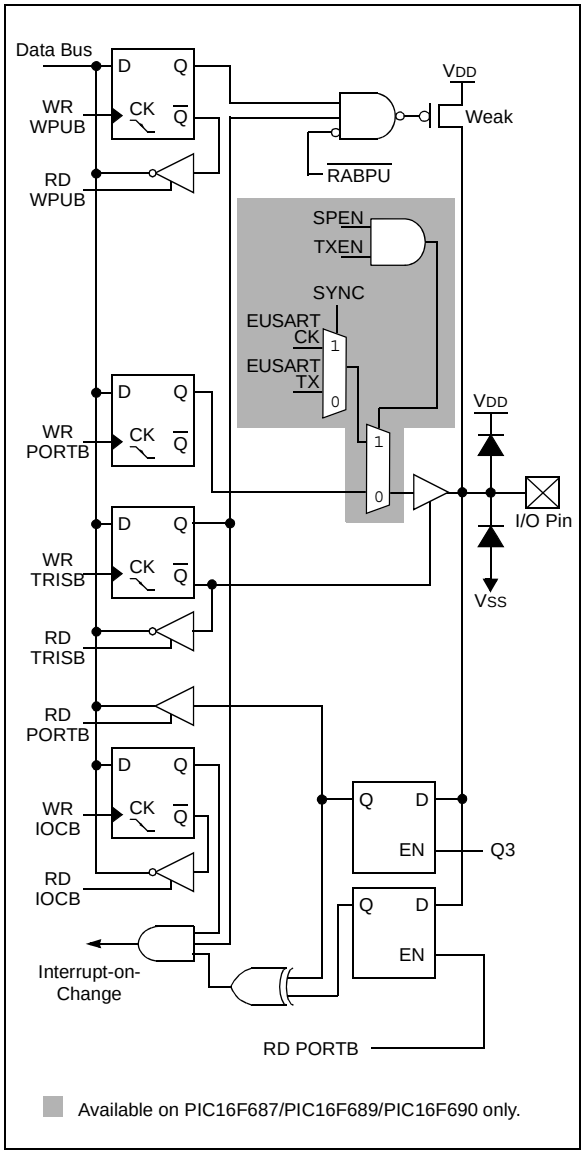


**TABLE 4-2: SUMMARY OF REGISTERS ASSOCIATED WITH PORTB**

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 06h/106h | PORTB | RB7 | RB6 | RB5 | RB4 | — | — | — | — | xxxx ---- | uuuu ---- |
| 86h/186h | TRISB | TRISB7 | TRISB6 | TRISB5 | TRISB4 | — | — | — | — | 1111 ---- | 1111 ---- |
| 0Bh/8Bh/ 10Bh/18Bh | INTCON | GIE | PEIE | T0IE | INTE | RABIE | T0IF | INTF | RABIF | 0000 000x | 0000 000x |
| 115h | WPUB | WPUB7 | WPUB6 | WPUB5 | WPUB4 | — | — | — | — | 1111 ---- | 1111 ---- |
| 116h | IOCB | IOCB7 | IOCB6 | IOCB5 | IOCB4 | — | — | — | — | 0000 ---- | 0000 ---- |

**Legend:** x = unknown, u = unchanged, – = unimplemented locations read as '0'. Shaded cells are not used by PORTB.

## 4.5 PORTC and TRISC Registers

PORTC is a 8-bit wide, bidirectional port. The corresponding data direction register is TRISC (Register 4-10). Setting a TRISC bit (= 1) will make the corresponding PORTC pin an input (i.e., put the corresponding output driver in a High-impedance mode). Clearing a TRISC bit (= 0) will make the corresponding PORTC pin an output (i.e., put the contents of the output latch on the selected pin). Example 4-4 shows how to initialize PORTC. Reading the PORTC register (Register 4-9) 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, this value is modified and then written to the port data latch.

The TRISC register controls the direction of the PORTC pins, even when they are being used as analog inputs. The user must ensure the bits in the TRISC register are maintained set when using them as analog inputs. I/O pins configured as analog input always read '0'.

| **Note:** | The ANSEL (11Eh) and ANSELH (11Fh) registers must be initialized to configure an analog channel as a digital input. Pins configured as analog inputs will read '0'. |
|---|---|

**EXAMPLE 4-4: INITIALIZING PORTC**

```
BCF     STATUS,RP0    ;Bank 0
BCF     STATUS,RP1    ;
CLRF    PORTC         ;Init PORTC
BSF     STATUS,RP1    ;Bank 2
CLRF    ANSEL         ;digital I/O
BSF     STATUS,RP0    ;Bank 1
BCF     STATUS,RP1    ;
MOVLW   0Ch           ;Set RC<3:2> as inputs
MOVWF   TRISC         ;and set RC<5:4,1:0>
                      ;as outputs
BCF     STATUS,RP0    ;Bank 0
```

**REGISTER 4-9: PORTC – PORTC REGISTER (ADDRESS: 07h OR 107h)**

| R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x |
|---|---|---|---|---|---|---|---|
| RC7 | RC6 | RC5 | RC4 | RC3 | RC2 | RC1 | RC0 |
| bit 7 | | | | | | | bit 0 |

bit 7-0 **RC<7:0>:** PORTC General Purpose I/O Pin bits
1 = Port pin is > $V_{IH}$
0 = Port pin is < $V_{IL}$

| **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 |

**REGISTER 4-10: TRISC – TRI-STATE PORTC REGISTER (ADDRESS: 87h OR 187h)**

| R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 |
|---|---|---|---|---|---|---|---|
| TRISC7 | TRISC6 | TRISC5 | TRISC4 | TRISC3 | TRISC2 | TRISC1 | TRISC0 |
| bit 7 | | | | | | | bit 0 |

bit 7-0 **TRISC<7:0>:** PORTC Tri-State Control bit
1 = PORTC pin configured as an input (tri-stated)
0 = PORTC pin configured as an output

| **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 |

### 4.5.1 RC0/AN4/C2IN+

The RC0 is configurable to function as one of the following:

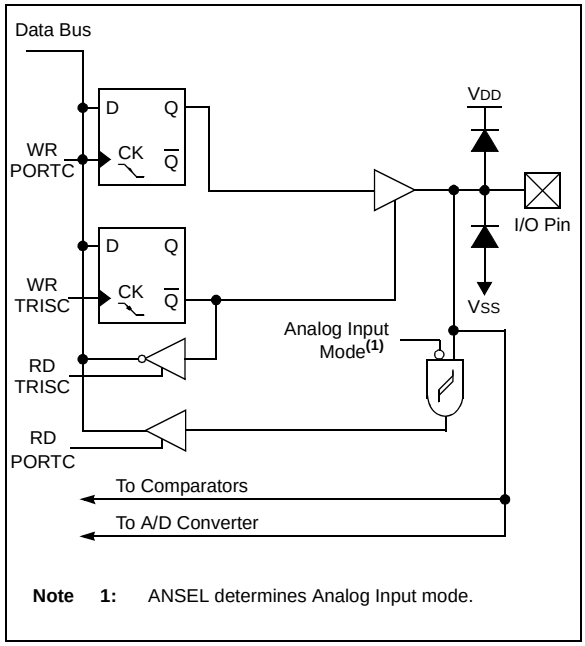
- a general purpose I/O
- an analog input for the A/D
- an analog input to Comparator 2

### 4.5.2 RC1/AN5/C12IN-

The RC1 is configurable to function as one of the following:

- a general purpose I/O
- an analog input for the A/D
- an analog input to Comparator 1 or 2

**FIGURE 4-11: BLOCK DIAGRAM OF RC0 AND RC1**




**Note 1:** ANSEL determines Analog Input mode.

### 4.5.3 RC2/AN6/P1D

The RC2/AN6/P1D**(1)** is configurable to function as one of the following:

- a general purpose I/O
- an analog input for the A/D
- a PWM output

**Note 1:** P1D is available on PIC16F685/PIC16F690 only.

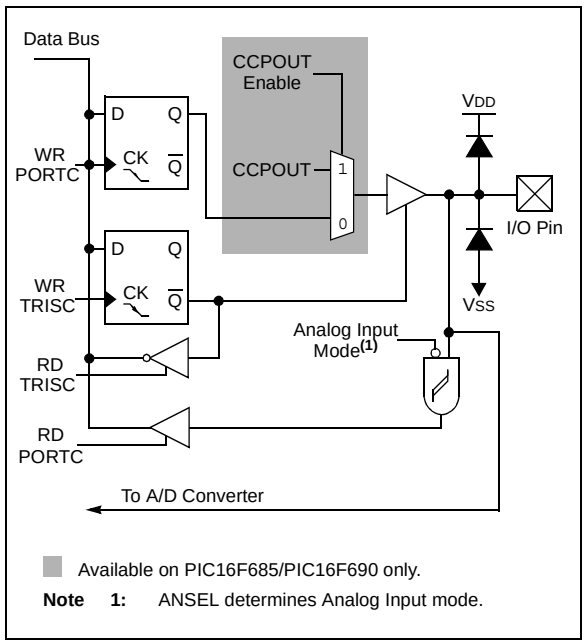
### 4.5.4 RC3/AN7/P1C

The RC3/AN7/P1C**(1)** is configurable to function as one of the following:

- a general purpose I/O
- an analog input for the A/D
- a PWM output

**Note 1:** P1C is available on PIC16F685/PIC16F690 only.

**FIGURE 4-12: BLOCK DIAGRAM OF RC2 AND RC3**




Available on PIC16F685/PIC16F690 only.

**Note 1:** ANSEL determines Analog Input mode.

### 4.5.5 RC4/C2OUT/P1B

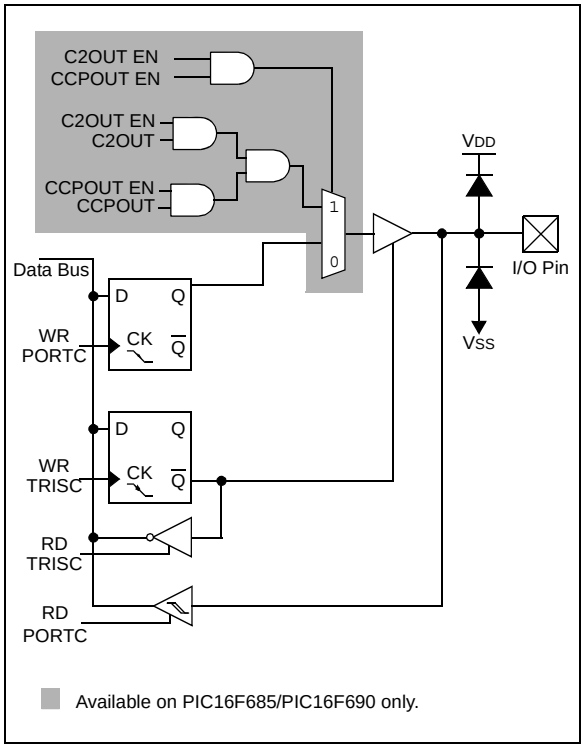
The RC4/C2OUT/P1B[1, 2] is configurable to function as one of the following:

- a general purpose I/O
- a digital output from Comparator 2
- a PWM output

**Note 1:** Enabling both C2OUT and P1B will cause a conflict on RC4 and create unpredictable results. Therefore, if C2OUT is enabled, the ECCP+ can not be used in Half-bridge or Full-bridge mode and vise-versa.

**2:** P1B is available on PIC16F685/ PIC16F690 only.

**FIGURE 4-13: BLOCK DIAGRAM OF RC4**



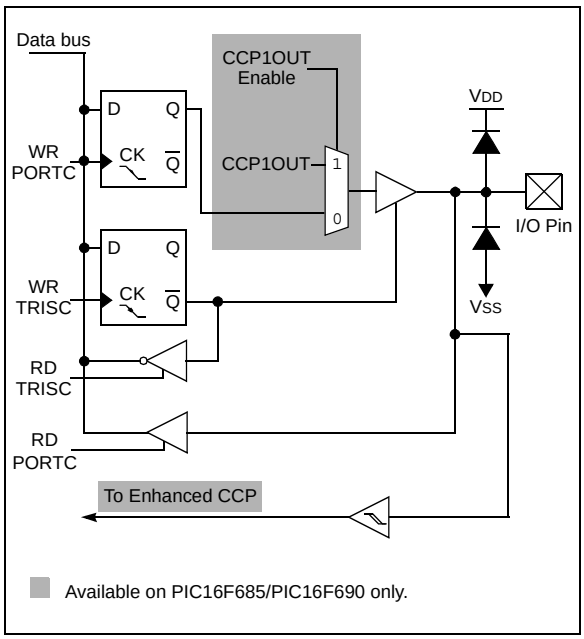

### 4.5.6 RC5/CCP1/P1A

The RC5/CCP1/P1A[1] is configurable to function as one of the following:

- a general purpose I/O
- a digital input/output for the Enhanced CCP
- a PWM output

**Note 1:** CCP1 and P1A are available on PIC16F685/PIC16F690 only.

**FIGURE 4-14: BLOCK DIAGRAM OF RC5**



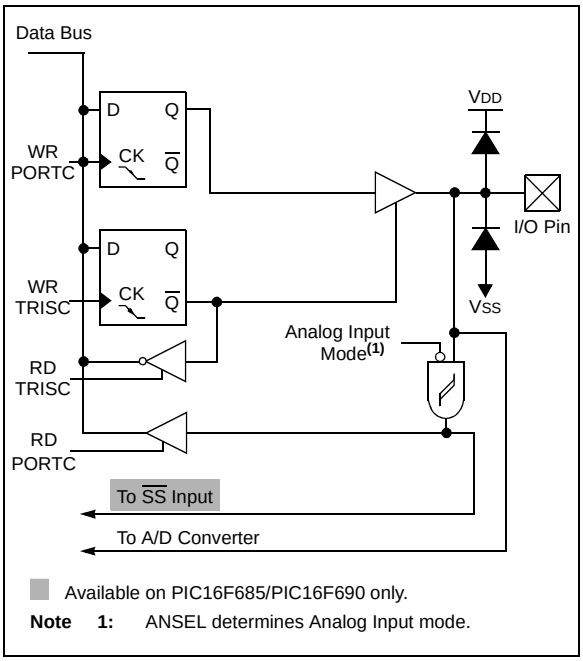
### 4.5.7 RC6/AN8/$\overline{SS}$

The RC6/AN8/$\overline{SS}$**(1)** is configurable to function as one of the following:

- a general purpose I/O
- an analog input for the A/D
- a slave select input

**Note 1:** $\overline{SS}$ is available on PIC16F687/PIC16F689/PIC16F690 only.

**FIGURE 4-15: BLOCK DIAGRAM OF RC6**




### 4.5.8 RC7/AN9/SDO

The RC7/AN9/SDO**(1)** is configurable to function as one of the following:

- a general purpose I/O
- an analog input for the A/D
- a serial data output

**Note 1:** SDO is available on PIC16F687/PIC16F689/PIC16F690 only.

**FIGURE 4-16: BLOCK DIAGRAM OF RC7**

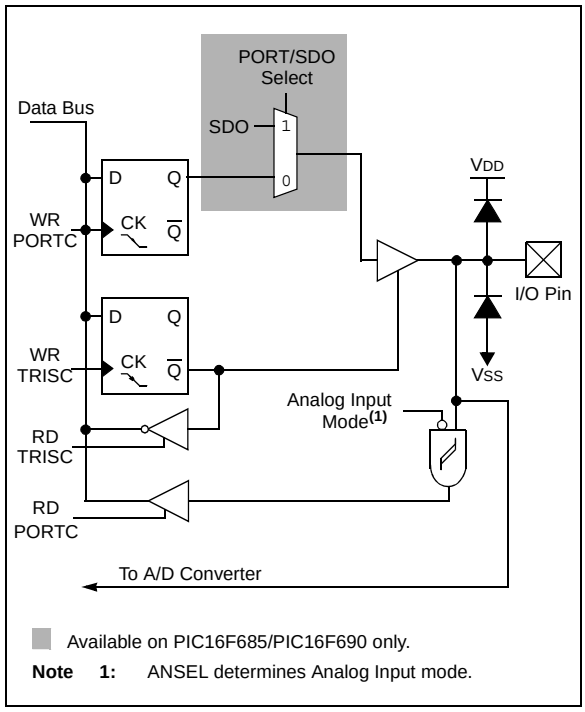


**TABLE 4-3: SUMMARY OF REGISTERS ASSOCIATED WITH PORTC**

| 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/107h | PORTC | RC7 | RC6 | RC5 | RC4 | RC3 | RC2 | RC1 | RC0 | xxxx xxxx | uuuu uuuu |
| 14h | SSPCON(1) | WCOL | SSPOV | SSPEN | CKP | SSPM3 | SSPM2 | SSPM1 | SSPM0 | 0000 0000 | 0000 0000 |
| 17h | CCP1CON(2) | P1M1 | P1M0 | DC1B1 | DC1B0 | CCP1M3 | CCP1M2 | CCP1M1 | CCP1M0 | 0000 0000 | 0000 0000 |
| 1Dh | ECCPAS(2) | ECCPASE | ECCPAS2 | ECCPAS1 | ECCPAS0 | PSSAC1 | PSSAC0 | PSSBD1 | PSSBD0 | 0000 0000 | 0000 0000 |
| 87h/187h | TRISC | TRISC7 | TRISC6 | TRISC5 | TRISC4 | TRISC3 | TRISC2 | TRISC1 | TRISC0 | 1111 1111 | 1111 1111 |
| 11Ah | CM2CON0 | C2ON | C2OUT | C2OE | C2POL | — | C2R | C2CH1 | C2CH0 | 0000 -000 | 0000 -000 |
| 11Bh | CM2CON1 | MC1OUT | MC2OUT | — | — | — | — | T1GSS | C2SYNC | 00-- --10 | 00-- --10 |
| 11Eh | ANSEL | ANS7 | ANS6 | ANS5 | ANS4 | ANS3 | ANS2 | ANS1 | ANS0 | 1111 1111 | 1111 1111 |
| 11Fh | ANSELH | — | — | — | — | ANS11 | ANS10 | ANS9 | ANS8 | ---- 1111 | ---- 1111 |
| 19Dh | PSTRCON | — | — | — | STRSYNC | STRD | STRC | STRB | STRA | ---0 0001 | ---0 0001 |
| 19Eh | SRCON | SR1 | SR0 | C1SEN | C2REN | PULSS | PULSR | — | — | 0000 00-- | 0000 00-- |
| 118h | VRCON | C1VREN | C2VREN | VRR | VP6EN | VR3 | VR2 | VR1 | VR0 | 0000 0000 | 0000 0000 |

**Legend:** x = unknown, u = unchanged, – = unimplemented locations read as '0'. Shaded cells are not used by PORTC.

**Note 1:** PIC16F687/PIC16F689/PIC16F690 only.

**2:** PIC16F685/PIC16F690 only.

## 5.0 TIMER0 MODULE

The Timer0 module timer/counter has the following features:

- 8-bit timer/counter
- Readable and writable
- 8-bit software programmable prescaler
- Internal or external clock select
- Interrupt on overflow from FFh to 00h
- Edge select for external clock

Figure 5-1 is a block diagram of the Timer0 module and the prescaler shared with the WDT.

### 5.1 Timer0 Operation

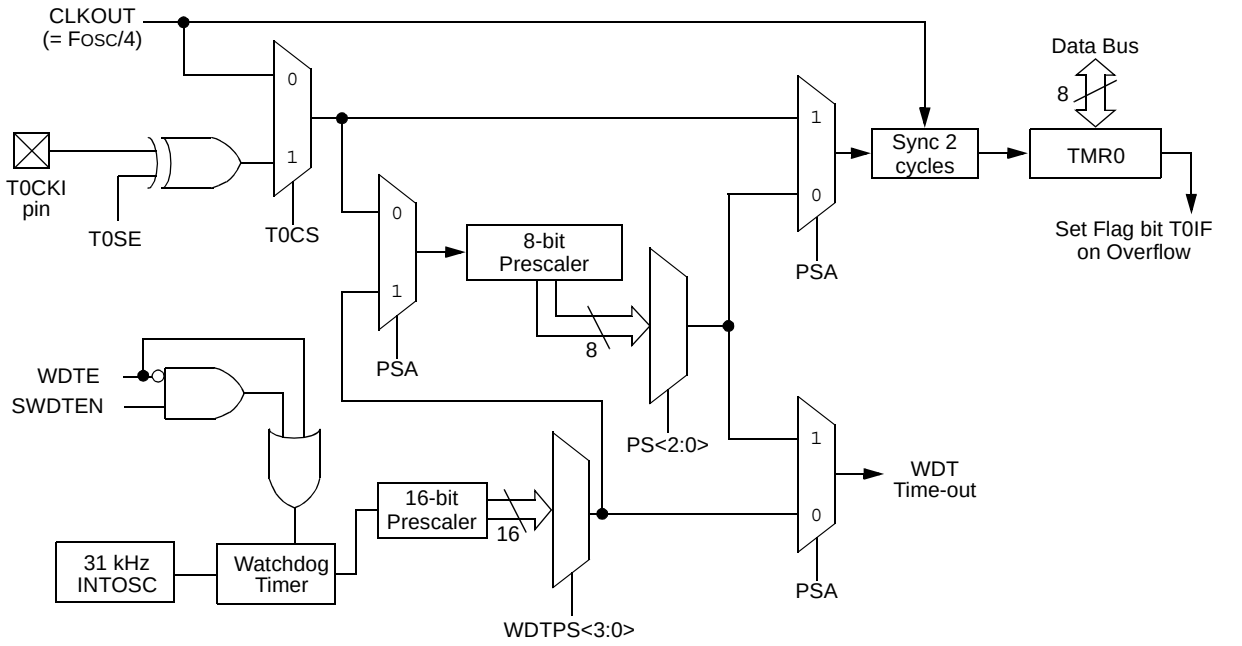
Timer mode is selected by clearing the T0CS bit (OPTION_REG<5>). In Timer mode, the Timer0 module will increment every instruction cycle (without prescaler). If TMR0 is written, the increment is inhibited for the following two instruction cycles. The user can work around this by writing an adjusted value to the TMR0 register.

Counter mode is selected by setting the T0CS bit (OPTION_REG<5>). In this mode, the Timer0 module will increment either on every rising or falling edge of pin RA2/AN1/T0CKI/INT/C1OUT. The incrementing edge is determined by the source edge (T0SE) control bit (OPTION_REG<4>). Clearing the T0SE bit selects the rising edge.

### 5.2 Timer0 Interrupt

A Timer0 interrupt is generated when the TMR0 register timer/counter overflows from FFh to 00h. This overflow sets the T0IF bit (INTCON<2>). The interrupt can be masked by clearing the T0IE bit (INTCON<5>). The T0IF bit must be cleared in software by the Timer0 module Interrupt Service Routine before re-enabling this interrupt. The Timer0 interrupt cannot wake the processor from Sleep since the timer is shut off during Sleep.

**FIGURE 5-1: BLOCK DIAGRAM OF THE TIMER0/WDT PRESCALER**




**Note 1:** T0SE, T0CS, PSA, PS<2:0> are bits in the OPTION register, WDTPS<3:0> are bits in the WDTCON register.

## 5.3 Using Timer0 with an External Clock

When no prescaler is used, the external clock input is the same as the prescaler output. The synchronization of T0CKI, with the internal phase clocks, is accomplished by sampling the prescaler output on the Q2 and Q4 cycles of the internal phase clocks. Therefore, it is necessary for T0CKI to be high for at least 2 TOSC (and a small RC delay of 20 ns) and low for at least 2 TOSC (and a small RC delay of 20 ns). Refer to the electrical specification of the desired device.

**Note:** The ANSEL (11Eh) register must be initialized to configure an analog channel as a digital input. Pins configured as analog inputs will read '0'.

**REGISTER 5-1: OPTION_REG – OPTION REGISTER (ADDRESS: 81h OR 181h)**

| R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 |
|---|---|---|---|---|---|---|---|
| $\overline{\text{RABPU}}$ | INTEDG | T0CS | T0SE | PSA | PS2 | PS1 | PS0 |
| bit 7 | | | | | | | bit 0 |

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

bit 6 **INTEDG:** Interrupt Edge Select bit
1 = Interrupt on rising edge of RA2/AN2/T0CKI/INT/C1OUT pin
0 = Interrupt on falling edge of RA2/AN2/T0CKI/INT/C1OUT pin

bit 5 **T0CS:** TMR0 Clock Source Select bit
1 = Transition on RA2/AN2/T0CKI/INT/C1OUT pin
0 = Internal instruction cycle clock (CLKOUT)

bit 4 **T0SE:** TMR0 Source Edge Select bit
1 = Increment on high-to-low transition on RA2/AN2/T0CKI/INT/C1OUT pin
0 = Increment on low-to-high transition on RA2/AN2/T0CKI/INT/C1OUT 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 **PS<2:0>:** Prescaler Rate Select bits

| BIT VALUE | TMR0 RATE | WDT RATE(1) |
|---|---|---|
| 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 |

**Note 1:** A dedicated 16-bit WDT postscaler is available. See **Section 14.5 "Watchdog Timer (WDT)"** for more information.

| **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 |

## 5.4 Prescaler

An 8-bit counter is available as a prescaler for the Timer0 module, or as a postscaler for the Watchdog Timer. For simplicity, this counter will be referred to as "prescaler" throughout this data sheet. The prescaler assignment is controlled in software by the control bit PSA (OPTION_REG<3>). Clearing the PSA bit will assign the prescaler to Timer0. Prescale values are selectable via the PS<2:0> bits (OPTION_REG<2:0>).

The prescaler is not readable or writable. When assigned to the Timer0 module, all instructions writing to the TMR0 register (e.g., `CLRF 1, MOVWF 1, BSF 1, x....etc.`) will clear the prescaler. When assigned to WDT, a `CLRWDT` instruction will clear the prescaler along with the Watchdog Timer.

### 5.4.1 SWITCHING PRESCALER ASSIGNMENT

The prescaler assignment is fully under software control (i.e., it can be changed "on the fly" during program execution). To avoid an unintended device Reset, the following instruction sequence (Example 5-1 and Example 5-2) must be executed when changing the prescaler assignment from Timer0 to WDT.

**EXAMPLE 5-1: CHANGING PRESCALER (TIMER0 → WDT)**

```
BCF     STATUS,RP0      ;Bank 0
BCF     STATUS,RP1      ;
CLRWDT                  ;Clear WDT
CLRF    TMR0            ;Clear TMR0 and
                        ;prescaler
BSF     STATUS,RP0      ;Bank 1
MOVLW   b'00101111'     ;Required if desired
MOVWF   OPTION_REG      ;PS<2:0> is
CLRWDT                  ;000 or 001
                        ;
MOVLW   b'00101xxx'     ;Set postscaler to
MOVWF   OPTION_REG      ;desired WDT rate
BCF     STATUS,RP0      ;Bank 0
```

To change prescaler from the WDT to the TMR0 module, use the sequence shown in Example 5-2. This precaution must be taken even if the WDT is disabled.

**EXAMPLE 5-2: CHANGING PRESCALER (WDT → TIMER0)**

```
CLRWDT                  ;Clear WDT and
                        ;prescaler
BSF     STATUS,RP0      ;Bank 1
BCF     STATUS,RP1      ;
MOVLW   b'xxxx0xxx'     ;Select TMR0,
                        ;prescale, and
                        ;clock source
MOVWF   OPTION_REG      ;
BCF     STATUS,RP0      ;Bank 0
```

**TABLE 5-1: REGISTERS ASSOCIATED WITH TIMER0**

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 01h/101h | TMR0 | Timer0 Module Register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 0Bh/8Bh/ 10Bh/18Bh | INTCON | GIE | PEIE | T0IE | INTE | RABIE | T0IF | INTF | RABIF | 0000 000x | 0000 000x |
| 81h/181h | OPTION_REG | $\overline{\text{RABPU}}$ | INTEDG | T0CS | T0SE | PSA | PS2 | PS1 | PS0 | 1111 1111 | 1111 1111 |
| 85h/185h | TRISA | — | — | TRISA5 | TRISA4 | TRISA3 | TRISA2 | TRISA1 | TRISA0 | --11 1111 | --11 1111 |

**Legend:** – = Unimplemented locations, read as '0', u = unchanged, x = unknown. Shaded cells are not used by the Timer0 module.

**NOTES:**

## 6.0 TIMER1 MODULE WITH GATE CONTROL

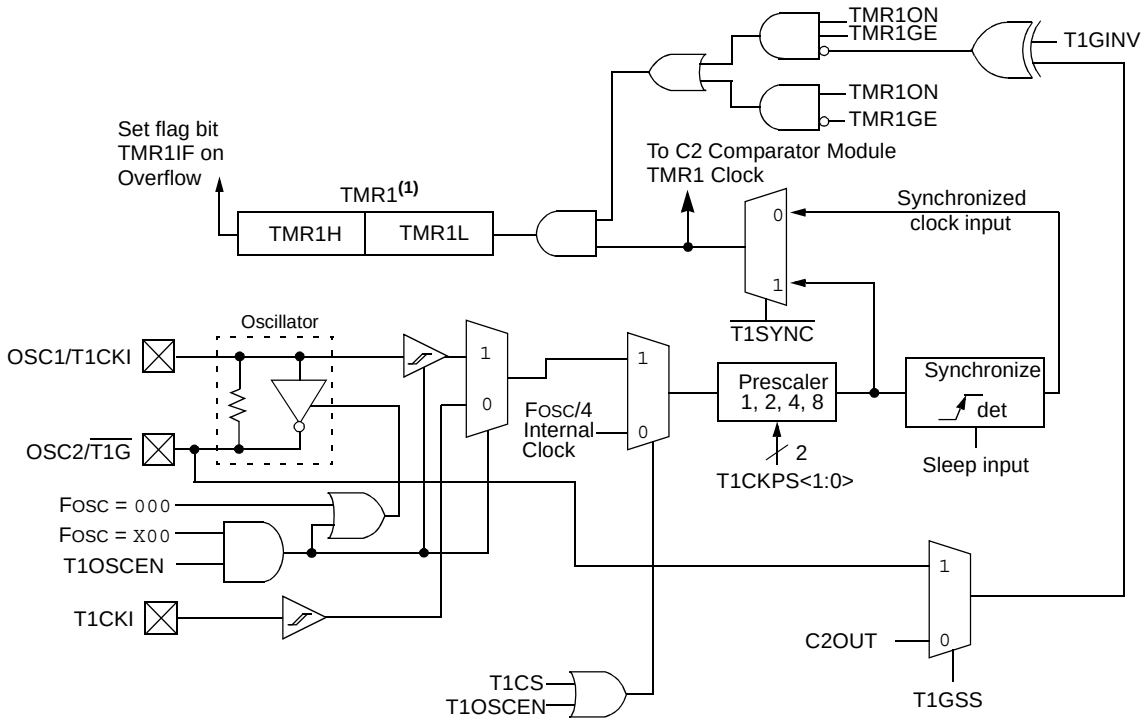
The Timer1 module has the following features:

- 16-bit timer/counter (TMR1H:TMR1L)
- Readable and writable
- Internal or external clock selection
- Synchronous or asynchronous operation
- Interrupt on overflow from FFFFh to 0000h
- Wake-up upon overflow (Asynchronous mode)
- Optional external enable input
  - Selectable gate source: $\overline{T1G}$ or C2 output (T1GSS)
  - Selectable gate polarity (T1GINV)
- Optional LP oscillator

Figure 6-1 shows the block diagram of the Timer1 module.

The Timer1 Control register (T1CON), shown in Register 6-1, is used to enable/disable Timer1 and select the various features of the Timer1 module.

**FIGURE 6-1: TIMER1 BLOCK DIAGRAM**




* ST Buffer is low power type when using LP osc, or high speed type when using T1CKI.

**Note 1:** Timer1 register increments on rising edge

## 6.1 Timer1 Modes of Operation

Timer1 can operate in one of three modes:

- 16-bit Timer with prescaler
- 16-bit Synchronous counter
- 16-bit Asynchronous counter

In Timer mode, Timer1 is incremented on every instruction cycle. In Counter mode, Timer1 is incremented on the rising edge of the external clock input T1CKI. In addition, the Counter mode clock can be synchronized to the microcontroller system clock or run asynchronously.

In Counter and Timer modules, the counter/timer clock can be gated by the Timer1 gate, which can be selected as either the T1G pin or Comparator 2 output.

If an external clock oscillator is needed (and the microcontroller is using the INTOSC without CLKOUT), Timer1 can use the LP oscillator as a clock source.

| **Note:** | In Counter mode, a falling edge must be registered by the counter prior to the first incrementing rising edge. |
|---|---|

## 6.2 Timer1 Interrupt

The Timer1 register pair (TMR1H:TMR1L) increments to FFFFh and rolls over to 0000h. When Timer1 rolls over, the Timer1 interrupt flag bit (PIR1<0>) is set. To enable the interrupt on rollover, you must set these bits:

- Timer1 interrupt enable bit (PIE1<0>)
- PEIE bit (INTCON<6>)
- GIE bit (INTCON<7>)

The interrupt is cleared by clearing the TMR1IF bit in the Interrupt Service Routine.

| **Note:** | The TMR1H:TTMR1L register pair and the TMR1IF bit should be cleared before enabling interrupts. |
|---|---|

## 6.3 Timer1 Prescaler

Timer1 has four prescaler options allowing 1, 2, 4 or 8 divisions of the clock input. The T1CKPS bits (T1CON<5:4>) control the prescale counter. The prescale counter is not directly readable or writable; however, the prescaler counter is cleared upon a write to TMR1H or TMR1L.

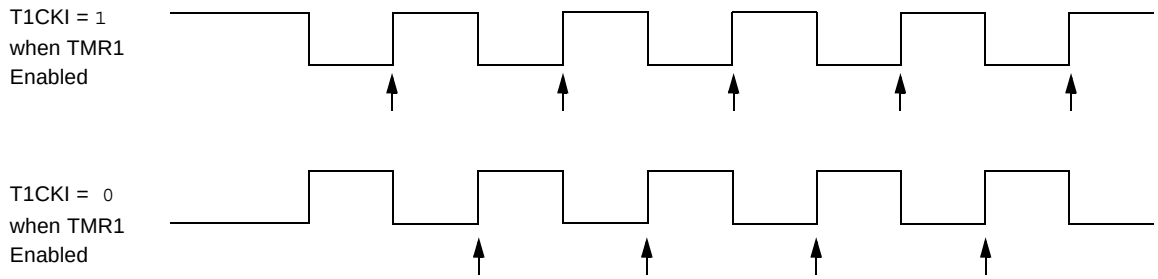
## 6.4 Timer1 Gate

Timer1 gate source is software configurable to be the T1G pin or the output of Comparator 2. This allows the device to directly time external events using T1G or analog events using Comparator 2. See CM2CON1 (Register 8-3) for selecting the Timer1 gate source. This feature can simplify the software for a Delta-Sigma A/D converter and many other applications. For more information on Delta-Sigma A/D converters, see the Microchip web site (www.microchip.com).

| **Note:** | TMR1GE bit (T1CON<6>) must be set to use either T1G or C2OUT as the Timer1 gate source. See Register 8-3 for more information on selecting the Timer1 gate source. |
|---|---|

Timer1 gate can be inverted using the T1GINV bit (T1CON<7>), whether it originates from the T1G pin or Comparator 2 output. This configures Timer1 to measure either the active-high or active-low time between events.

**FIGURE 6-2: TIMER1 INCREMENTING EDGE**




**Note 1:** Arrows indicate counter increments.

**2:** In Counter mode, a falling edge must be registered by the counter prior to the first incrementing rising edge of the clock.

**REGISTER 6-1: T1CON – TIMER1 CONTROL REGISTER (ADDRESS: 10h)**

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| T1GINV(1) | TMR1GE(2) | T1CKPS1 | T1CKPS0 | T1OSCEN | $\overline{\text{T1SYNC}}$ | TMR1CS | TMR1ON |
| bit 7 | | | | | | | bit 0 |

bit 7 **T1GINV:** Timer1 Gate Invert bit(1)
1 = Timer1 gate is inverted
0 = Timer1 gate is not inverted

bit 6 **TMR1GE:** Timer1 Gate Enable bit(2)
If TMR1ON = 0:
This bit is ignored
If TMR1ON = 1:
1 = Timer1 is on if Timer1 gate is not active
0 = Timer1 is on

bit 5-4 **T1CKPS<1:0>:** 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:** LP Oscillator Enable Control bit
If INTOSC without CLKOUT oscillator is active:
1 = LP oscillator is enabled for Timer1 clock
0 = LP oscillator is off
Else:
This bit is ignored

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.

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

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

**Note 1:** T1GINV bit inverts the Timer1 gate logic, regardless of source.

**2:** TMR1GE bit must be set to use either $\overline{\text{T1G}}$ pin or C2OUT, as selected by the T1GSS bit (CM2CON1<1>), as a Timer1 gate source.

| **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 |

## 6.5 Timer1 Operation in Asynchronous Counter Mode

If control bit $\overline{\text{T1SYNC}}$ (T1CON<2>) is set, the external clock input is not synchronized. The timer continues to increment asynchronous to the internal phase clocks. The timer will continue to run during Sleep and can generate an interrupt on overflow, which will wake-up the processor. However, special precautions in software are needed to read/write the timer (see **Section 6.5.1 "Reading and Writing Timer1 in Asynchronous Counter Mode"**).

| **Note:** | The ANSEL (11Eh) register must be initialized to configure an analog channel as a digital input. Pins configured as analog inputs will read '0'. |
|---|---|

### 6.5.1 READING AND WRITING TIMER1 IN ASYNCHRONOUS COUNTER MODE

Reading TMR1H or TMR1L while the timer is running from an external asynchronous clock will ensure a valid read (taken care of in hardware). However, the user should keep in mind that reading the 16-bit timer in two 8-bit values itself, poses certain problems, since the timer may overflow between the reads.

For writes, it is recommended that the user simply stop the timer and write the desired values. A write contention may occur by writing to the timer registers, while the register is incrementing. This may produce an unpredictable value in the timer register.

## 6.6 Timer1 Oscillator

A crystal oscillator circuit is built-in between pins OSC1 (input) and OSC2 (amplifier output). It is enabled by setting control bit, T1OSCEN (T1CON<3>). The oscillator is a low-power oscillator rated up to 32 kHz. It will continue to run during Sleep. It is primarily intended for a 32 kHz crystal. Table 3-1 shows the capacitor selection for the Timer1 oscillator.

The Timer1 oscillator is shared with the system LP oscillator. Thus, Timer1 can use this mode only when the primary system clock is derived from the internal oscillator. As with the system LP oscillator, the user must provide a software time delay to ensure proper oscillator start-up.

TRISA5 and TRISA4 bits are set when the Timer1 oscillator is enabled. RA5 and RA4 bits read as '0' and TRISA5 and TRISA4 bits read as '1'.

| **Note:** | The oscillator requires a start-up and stabilization time before use. Thus, T1OSCEN should be set and a suitable delay observed prior to enabling Timer1. |
|---|---|

## 6.7 Timer1 Operation During Sleep

Timer1 can only operate during Sleep when setup in Asynchronous Counter mode. In this mode, an external crystal or clock source can be used to increment the counter. To set up the timer to wake the device:

- Timer1 must be on (T1CON<0>)
- TMR1IE bit (PIE1<0>) must be set
- PEIE bit (INTCON<6>) must be set

The device will wake-up on an overflow. If the GIE bit (INTCON<7>) is set, the device will wake-up and jump to the Interrupt Service Routine (0004h) on an overflow. If the GIE bit is clear, execution will continue with the next instruction.

**TABLE 6-1: REGISTERS ASSOCIATED WITH TIMER1**

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0Bh/8Bh/ 10Bh/18Bh | INTCON | GIE | PEIE | T0IE | INTE | RABIE | T0IF | INTF | RABIF | 0000 000x | 0000 000x |
| 0Ch | PIR1 | — | ADIF | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | -000 0000 | -000 0000 |
| 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 | T1GINV | TMR1GE | T1CKPS1 | T1CKPS0 | T1OSCEN | $\overline{\text{T1SYNC}}$ | TMR1CS | TMR1ON | 0000 0000 | uuuu uuuu |
| 11Bh | CM2CON1 | MC1OUT | MC2OUT | — | — | — | — | T1GSS | C2SYNC | 00-- --10 | 00-- --10 |
| 8Ch | PIE1 | — | ADIE | RCIE | TXIE | SSPIE | CCP1IE | TMR2IE | TMR1IE | -000 0000 | -000 0000 |

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

## 7.0 TIMER2 MODULE

The Timer2 module timer has the following features:

- 8-bit timer (TMR2 register)
- 8-bit period register (PR2)
- Readable and writable (both registers)
- Software programmable prescaler (1:1, 1:4, 1:16)
- Software programmable postscaler (1:1 to 1:16)
- Interrupt on TMR2 match with PR2

Timer2 has a control register shown in Register 7-1. TMR2 can be shut-off by clearing control bit, TMR2ON (T2CON<2>), to minimize power consumption. Figure 7-1 is a simplified block diagram of the Timer2 module. The prescaler and postscaler selection of Timer2 are controlled by this register.

### 7.1 Timer2 Operation

Timer2 can be used as the PWM time base for the PWM mode of the ECCP+ module. The TMR2 register is readable and writable, and is cleared on any device Reset. The input clock (FOSC/4) has a prescale option of 1:1, 1:4 or 1:16, selected by control bits T2CKPS<1:0> (T2CON<1:0>). The match output of TMR2 goes through a 4-bit postscaler (which gives a 1:1 to 1:16 scaling inclusive) to generate a TMR2 interrupt (latched in flag bit, TMR2IF (PIR1<1>)).

The prescaler and postscaler counters are cleared when any of the following occurs:

- A write to the TMR2 register
- A write to the T2CON register
- Any device Reset (Power-on Reset, $\overline{\text{MCLR}}$ Reset, Watchdog Timer Reset, or Brown-out Reset)

TMR2 is not cleared when T2CON is written.

**REGISTER 7-1: T2CON — TIMER2 CONTROL REGISTER (ADDRESS: 12h)**

| U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | TOUTPS3 | TOUTPS2 | TOUTPS1 | TOUTPS0 | TMR2ON | T2CKPS1 | T2CKPS0 |
| bit 7 | | | | | | | bit 0 |

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

bit 6-3 **TOUTPS<3:0>:** Timer2 Output Postscale Select bits

0000 =1:1 postscale
0001 =1:2 postscale
•
•
•
1111 =1:16 postscale

bit 2 **TMR2ON:** Timer2 On bit

1 = Timer2 is on
0 = Timer2 is off

bit 1-0 **T2CKPS<1:0>:** Timer2 Clock Prescale Select bits

00 = Prescaler is 1
01 = Prescaler is 4
1x = Prescaler is 16

| **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 |

## 7.2 Timer2 Interrupt

The Timer2 module has an 8-bit period register, PR2. Timer2 increments from 00h until it matches PR2 and then resets to 00h on the next increment cycle. PR2 is a readable and writable register. The PR2 register is initialized to FFh upon Reset.

**FIGURE 7-1: TIMER2 BLOCK DIAGRAM**

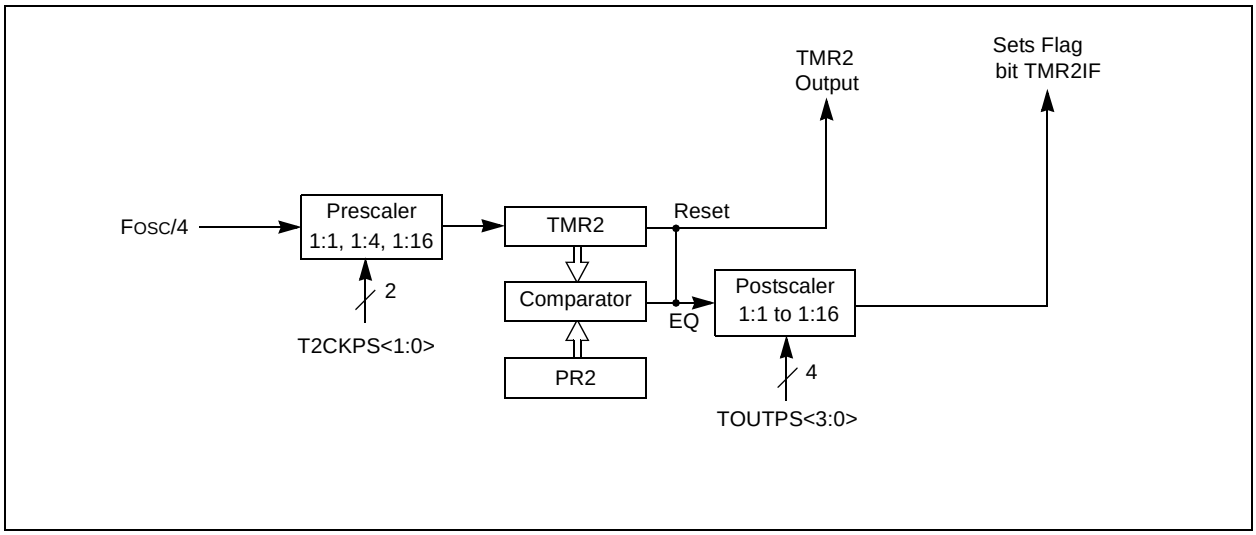



**TABLE 7-1: REGISTERS ASSOCIATED WITH TIMER2**

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0Bh/8Bh/ 10Bh/18Bh | INTCON | GIE | PEIE | T0IE | INTE | RABIE | T0IF | INTF | RABIF | 0000 000x | 0000 000x |
| 0Ch | PIR1 | — | ADIF | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | -000 0000 | -000 0000 |
| 11h | TMR2 | Holding Register for the 8-bit TMR2 Register | | | | | | | | 0000 0000 | 0000 0000 |
| 12h | T2CON | — | TOUTPS3 | TOUTPS2 | TOUTPS1 | TOUTPS0 | TMR2ON | T2CKPS1 | T2CKPS0 | -000 0000 | -000 0000 |
| 8Ch | PIE1 | — | ADIE | RCIE | TXIE | SSPIE | CCP1IE | TMR2IE | TMR1IE | -000 0000 | -000 0000 |
| 92h | PR2 | Timer2 Module Period Register | | | | | | | | 1111 1111 | 1111 1111 |

**Legend:** x = unknown, u = unchanged, – = unimplemented, read as '0'. Shaded cells are not used by the Timer2 module.

## 8.0 COMPARATOR MODULE

The comparator module has two separate voltage comparators: Comparator C1 and Comparator C2.

Each comparator offers the following list of features:

- Control and configuration register
- Comparator output available externally
- Programmable output polarity
- Interrupt-on-change flags
- Wake-up from Sleep
- Configurable as feedback input to the PWM
- Programmable four input multiplexer
- Programmable two input reference selections
- Timer1 gate
- Output synchronization to Timer1 clock input (Comparator C2 only)

| **Note:** | C2 can be linked to Timer1Gate. |
|---|---|

## 8.1 Control Registers

Both comparators have separate control and configuration registers: CM1CON0 for C1 and CM2CON0 for C2. In addition, Comparator C2 has a second control register, CM2CON1, for synchronization control and simultaneous reading of both comparator outputs.

### 8.1.1 COMPARATOR C1 CONTROL REGISTER

The CM1CON0 register (shown in Register 8-1) contains the control and Status bits for the following:

- Comparator enable
- Comparator input selection
- Comparator reference selection
- Output mode

Setting C1ON (CM1CON0<7>) enables Comparator C1 for operation.

Bits C1CH<1:0> (CM1CON0<1:0>) select the comparator input from the four analog pins AN<7:5,1>.

| **Note:** | To use AN<7:5,1> as analog inputs the appropriate bits must be programmed to '1' in the ANSEL register. |
|---|---|

Setting C1R (CM1CON0<2>) selects the C1VREF output of the comparator voltage reference module as the reference voltage for the comparator. Clearing C1R selects the C1IN+ input on the RA0/AN0/C1IN+/ICSPDAT/ULPWU pin.

The output of the comparator is available internally via the C1OUT flag (CM1CON0<6>). To make the output available for an external connection, the C1OE bit (CM1CON0<5>) must be set.

The polarity of the comparator output can be inverted by setting the C1POL bit (CM1CON0<4>). Clearing C1POL results in a non-inverted output.

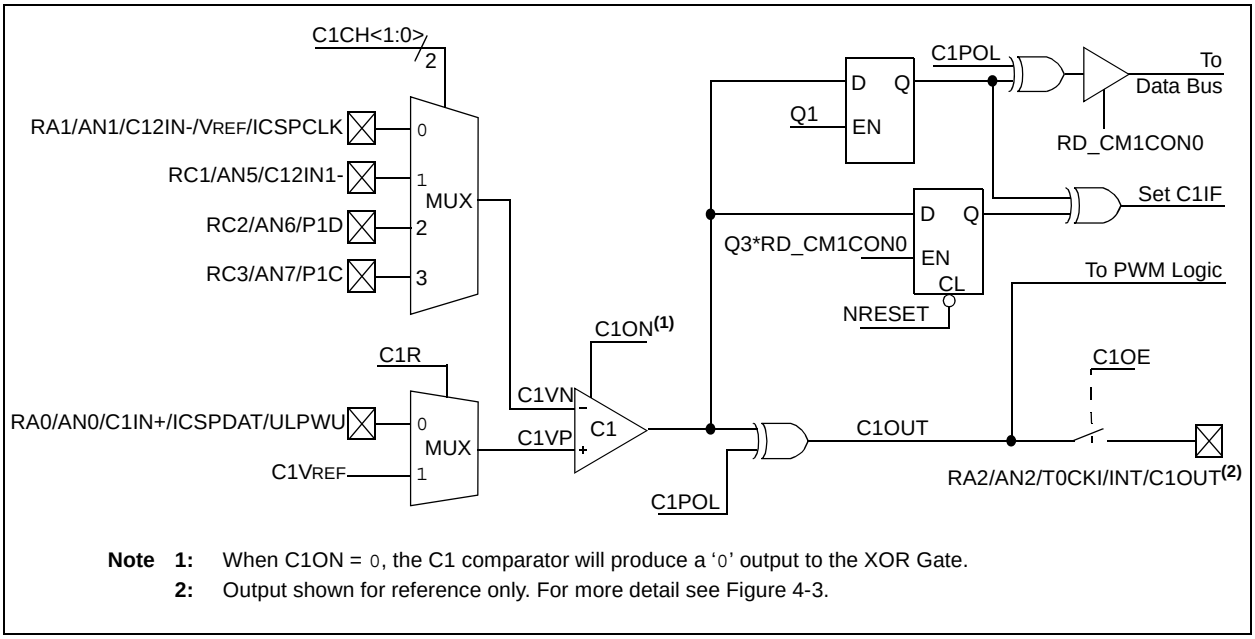
A complete table showing the output state versus input conditions and the polarity bit is shown in Table 8-1.

**TABLE 8-1: C1 OUTPUT STATE VS. INPUT CONDITIONS**

| Input Condition | C1POL | C1OUT |
|---|---|---|
| C1VN > C1VP | 0 | 0 |
| C1VN < C1VP | 0 | 1 |
| C1VN > C1VP | 1 | 1 |
| C1VN < C1VP | 1 | 0 |

| **Note 1:** | The internal output of the comparator is latched at the end of each instruction cycle. External outputs are not latched. |
|---|---|
| **2:** | The C1 interrupt will operate correctly with C1OE set or cleared. |
| **3:** | For C1 output on RA2/AN2/T0CKI/INT/C1OUT: C1OE = 1, C1ON = 1 and TRISA<2> = 0. |

**FIGURE 8-1: COMPARATOR C1 SIMPLIFIED BLOCK DIAGRAM**




**Note 1:** When C1ON = 0, the C1 comparator will produce a '0' output to the XOR Gate.
**2:** Output shown for reference only. For more detail see Figure 4-3.

**REGISTER 8-1: CM1CON0 – COMPARATOR C1 CONTROL REGISTER 0 (ADDRESS: 119h)**

| R/W-0 | R-0 | R/W-0 | R/W-0 | U-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| C1ON | C1OUT | C1OE | C1POL | — | C1R | C1CH1 | C1CH0 |
| bit 7 | | | | | | | bit 0 |

bit 7 **C1ON:** Comparator C1 Enable bit
1 = C1 Comparator is enabled
0 = C1 Comparator is disabled

bit 6 **C1OUT:** Comparator C1 Output bit
If C1POL = 1 (inverted polarity):
C1OUT = 1, C1VP < C1VN
C1OUT = 0, C1VP > C1VN
If C1POL = 0 (non-inverted polarity):
C1OUT = 1, C1VP > C1VN
C1OUT = 0, C1VP < C1VN

bit 5 **C1OE:** Comparator C1 Output Enable bit
1 = C1OUT is present on the RA2/AN2/T0CKI/INT/C1OUT pin(1)
0 = C1OUT is internal only

bit 4 **C1POL:** Comparator C1 Output Polarity Select bit
1 = C1OUT logic is inverted
0 = C1OUT logic is not inverted

bit 3 **Unimplemented**: Read as '0'

bit 2 **C1R:** Comparator C1 Reference Select bit (non-inverting input)
1 = C1VP connects to C1VREF output
0 = C1VP connects to RA0/AN0/C1IN+/ICSPDAT/ULPWU

bit 1-0 **C1CH<1:0>:** Comparator C1 Channel Select bit
00 = C1VN of C1 connects to RA1/AN1/C12IN-/VREF/ICSPCLK
01 = C1VN of C1 connects to RC1/AN5/C12IN-
10 = C1VN of C1 connects to RC2/AN6/P1D
11 = C1VN of C1 connects to RC3/AN7/P1C

**Note 1:** C1OUT will only drive RA2/AN2/T0CKI/INT/C1OUT if: C1OE = 1, C1ON = 1 and TRISA<2> = 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 |

### 8.1.2 COMPARATOR 2 CONTROL REGISTERS

The Comparator 2 (C2) register (CM2CON0) is a functional copy of the CM1CON0 register described in **Section 8.1.1 "Comparator C1 Control Register"**. A second control register, CM2CON1, is also present for control of an additional synchronizing feature, as well as mirrors of both comparator outputs.

#### 8.1.2.1 Comparator 2 Control Register 0

The CM2CON0 register, shown in Register 8-2, contains the control and Status bits for Comparator C2.

Setting C2ON (CM2CON0<7>) enables Comparator C2 for operation.

Bits C2CH<1:0> (CM2CON0<1:0>) select the comparator input from the four analog pins, AN<7:5,1>.

**Note 1:** To use AN<7:5,1> as analog inputs, the appropriate bits must be programmed to 1 in the ANSEL register.

C2R (CM2CON0<2>) selects the reference to be used with the comparator. Setting C2R (CM2CON0<2>) selects the C2VREF output of the comparator voltage reference module as the reference voltage for the comparator. Clearing C2R selects the C2IN+ input on the RC0/AN4/C2IN+ pin.

The output of the comparator is available internally via the C2OUT bit (CM2CON0<6>). To make the output available for an external connection, the C2OE bit (CM2CON0<5>) must be set.

The comparator output, C2OUT, can be inverted by setting the C2POL bit (CM2CON0<4>). Clearing C2POL results in a non-inverted output.

A complete table showing the output state versus input conditions and the polarity bit is shown in Table 8-2.

**TABLE 8-2: C2 OUTPUT STATE VS. INPUT CONDITIONS**

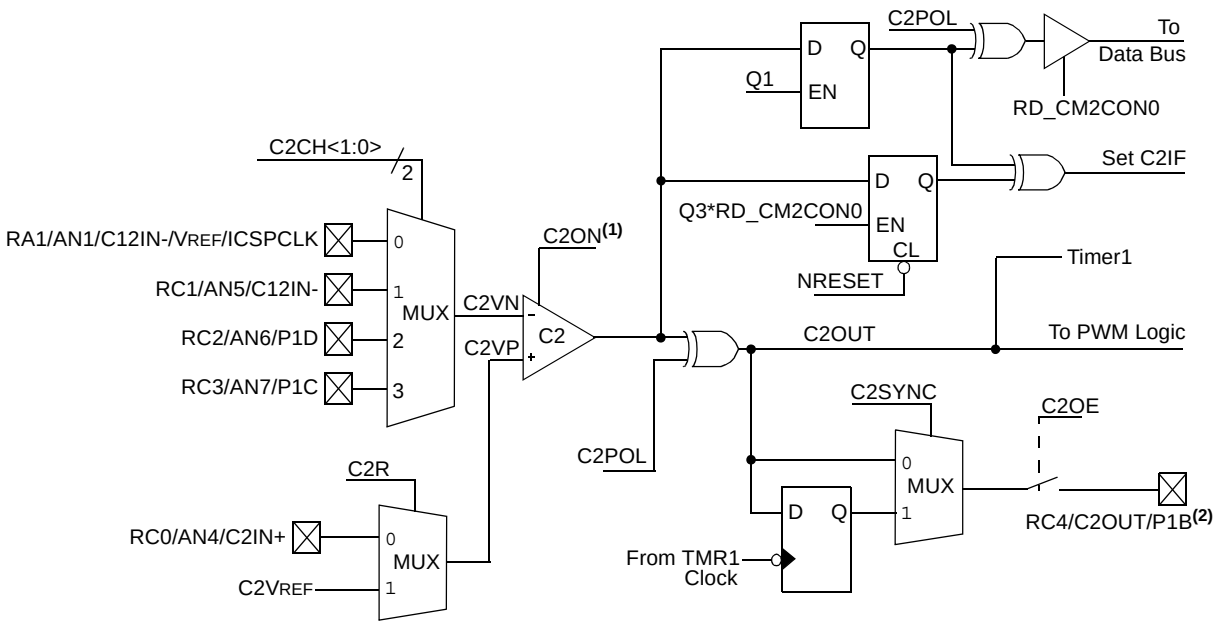
| Input Condition | C2POL | C2OUT |
|---|---|---|
| C2VN > C2VP | 0 | 0 |
| C2VN < C2VP | 0 | 1 |
| C2VN > C2VP | 1 | 1 |
| C2VN < C2VP | 1 | 0 |

**Note 1:** The internal output of the comparator is latched at the end of each instruction cycle. External outputs are not latched.

**2:** The C2 interrupt will operate correctly with C2OE set or cleared. An external output is not required for the C2 interrupt.

**3:** For C2 output on RC4/C2OUT/P1B: C2OE = 1, C2ON = 1 and TRISC<4> = 0.

**FIGURE 8-2: COMPARATOR C2 SIMPLIFIED BLOCK DIAGRAM**




**Note 1:** When C2ON = 0, the C2 comparator will produce a '0' output to the XOR Gate.

**2:** Output shown for reference only. See Figure 4-14 for more detail.

**REGISTER 8-2: CM2CON0 – COMPARATOR 2 CONTROL REGISTER 0 (ADDRESS: 11AH)**

| R/W-0 | R-0 | R/W-0 | R/W-0 | U-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| C2ON | C2OUT | C2OE | C2POL | — | C2R | C2CH1 | C2CH0 |
| bit 7 | | | | | | | bit 0 |

bit 7 **C2ON:** Comparator C2 Enable bit
1 = C2 Comparator is enabled
0 = C2 Comparator is disabled

bit 6 **C2OUT:** Comparator C2 Output bit
If C2POL = 1 (inverted polarity):
C2OUT = 1, C2VP < C2VN
C2OUT = 0, C2VP > C2VN
If C2POL = 0 (non-inverted polarity):
C2OUT = 1, C2VP > C2VN
C2OUT = 0, C2VP < C2VN

bit 5 **C2OE:** Comparator C2 Output Enable bit
1 = C2OUT is present on RC4/C2OUT/P1B(1)
0 = C2OUT is internal only

bit 4 **C2POL:** Comparator C2 Output Polarity Select bit
1 = C2OUT logic is inverted
0 = C2OUT logic is not inverted

bit 3 **Unimplemented**: Read as '0'

bit 2 **C2R:** Comparator C2 Reference Select bits (non-inverting input)
1 = C2VP connects to C2VREF
0 = C2VP connects to RC0/AN4/C2IN+

bit 1-0 **C2CH<1:0>:** Comparator C2 Channel Select bits
00 = C2VN of C2 connects to RA1/AN1/C12IN-/VREF/ICSPCLK
01 = C2VN of C2 connects to RC1/AN5/C12IN-
10 = C2VN of C2 connects to RC2/AN6/P1D
11 = C2VN of C2 connects to RC3/AN7/P1C

**Note 1:** C2OUT will only drive RC4/C2OUT/P1B if:
C2OE = 1, C2ON = 1 and TRISC<4> = 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 |

#### 8.1.2.2 Comparator 2 Control Register 1

Comparator 2 has one additional feature: its output can be synchronized to the Timer1 clock input. Setting C2SYNC (CM2CON1<0>) synchronizes the output of Comparator 2 to the falling edge of Timer1's clock input (see Figure 8-2 and Register 8-3).

The CM2CON1 register also contains mirror copies of both comparator outputs, MC1OUT and MC2OUT (CM2CON1<7:6>). The ability to read both outputs simultaneously from a single register eliminates the timing skew of reading separate registers.

| **Note 1:** | Obtaining the status of C1OUT or C2OUT by reading CM2CON1 does not affect the comparator interrupt mismatch registers. |
|---|---|

**REGISTER 8-3: CM2CON1 – COMPARATOR 2 CONTROL REGISTER 1 (ADDRESS: 11Bh)**

| R-0 | R-0 | U-0 | U-0 | U-0 | U-0 | R/W-1 | R/W-0 |
|---|---|---|---|---|---|---|---|
| MC1OUT | MC2OUT | — | — | — | — | T1GSS | C2SYNC |
| bit 7 | | | | | | | bit 0 |

bit 7 **MC1OUT:** Mirror Copy of C1OUT bit (CM1CON0<6>)

bit 6 **MC2OUT:** Mirror Copy of C2OUT bit (CM2CON0<6>)

bit 5-2 **Unimplemented:** Read as '0'

bit 1 **T1GSS:** Timer1 Gate Source Select bit
1 = Timer1 gate source is RA4/AN3/$\overline{T1G}$/OSC2/CLKOUT
0 = Timer1 gate source is C2OUT.

bit 0 **C2SYNC:** C2 Output Synchronous Mode bit
1 = C2 output is synchronous to falling edge of TMR1 clock
0 = C2 output is asynchronous

| **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 |

## 8.2 Comparator Outputs

The comparator outputs are read through the CM1CON0, COM2CON0 or CM2CON1 registers. CM1CON0 and CM2CON0 each contain the individual comparator output of Comparator 1 and Comparator 2, respectively. CM2CON1 contains a mirror copy of both comparator outputs facilitating a simultaneous read of both comparators. These bits are read-only. The comparator outputs may also be directly output to the RA2/AN2/T0CKI/INT/C1OUT and RC4/C2OUT/P1B I/O pins. When enabled, multiplexers in the output path of the RA2/AN2/T0CKI/INT/C1OUT and RC4/C2OUT/P1B pins will switch and the output of each pin will be the unsynchronized output of the comparator. The uncertainty of each of the comparators is related to the input offset voltage and the response time given in the specifications. Figure 8-1 and Figure 8-2 show the output block diagrams for Comparators 1 and 2, respectively.

The TRIS bits will still function as an output enable/disable for the RA2/AN2/T0CKI/INT/C1OUT and RC4/C2OUT/P1B pins while in this mode.

The polarity of the comparator outputs can be changed using the C1POL and C2POL bits (CMxCON0<4>).

Timer1 gate source can be configured to use the T1G pin or Comparator 2 output as selected by the T1GSS bit (CM2CON1<1>). The Timer1 gate feature can be used to time the duration or interval of analog events. The output of Comparator 2 can also be synchronized with Timer1 by setting the C2SYNC bit (CM2CON1<0>). When enabled, the output of Comparator 2 is latched on the falling edge of Timer1 clock source. If a prescaler is used with Timer1, Comparator 2 is latched after the prescaler. To prevent a race condition, the Comparator 2 output is latched on the falling edge of the Timer1 clock source and Timer1 increments on the rising edge of its clock source. See the Comparator 2 Block Diagram (Figure 8-2) and the Timer1 Block Diagram (Figure 6-1) for more information.

It is recommended to synchronize Comparator 2 with Timer1 by setting the C2SYNC bit when Comparator 2 is used as the Timer1 gate source. This ensures Timer1 does not miss an increment if Comparator 2 changes during an increment.

### 8.2.1 COMPARATOR INTERRUPT OPERATION

The comparator interrupt flags are set whenever there is a change in the output value of its respective comparator. Software will need to maintain information about the status of the output bits, as read from CM2CON0<7:6>, to determine the actual change that has occurred. The CxIF bits, PIR2<6:5>, are the Comparator Interrupt Flags. Each comparator interrupt bit must be reset in software by clearing it to '0'. Since it is also possible to write a '1' to this register, a simulated interrupt may be initiated.

The CxIE bits (PIE2<6:5>) and the PEIE bit (INTCON<6>) must be set to enable the interrupts. In addition, the GIE bit must also be set. If any of these bits are cleared, the interrupt is not enabled, though the CxIF bits will still be set if an interrupt condition occurs.

The comparator interrupt of the PIC16F685/687/689/690 differs from previous designs in that the interrupt flag is set by the mismatch edge and not the mismatch level. This means that the interrupt flag can be reset without the additional step of reading or writing the CMxCON0 register to clear the mismatch registers. When the mismatch registers are not cleared, an interrupt will not occur when the comparator output returns to the previous state. When the mismatch registers are cleared, an interrupt will occur when the comparator returns to the previous state.

**Note 1:** If a change in the CMxCON0 register (CxOUT) should occur when a read operation is being executed (start of the Q2 cycle), then the CxIF (PIR2<5:6>) interrupt flag may not get set.

**2:** When either comparator is first enabled, bias circuitry in the comparator module may cause an invalid output from the comparator until the bias circuitry is stable. Allow about 1 µs for bias settling then clear the mismatch condition and interrupt flags before enabling comparator interrupts.

## 8.3 SR Latch Output

An SR latch is connected to the comparator outputs C1OUT and C2OUT. Upon any Reset, the SR latch is always disabled. As a result, the latch output must be initialized before the outputs are made available to the output pins. Additionally, the applicable TRIS bits of the corresponding ports must be set to output ('0') and the respective comparator output enable bits (C1OE and/or C2OE) must be initialized in order to make the latch outputs available on the output pins. The four different configurations available for the SR latch are shown in Figure 8-5, and the SR<1:0> bits in the SRCON register (Register 8-4) control whether or not the latch is enabled. The latch enable state is completely independent of the enable state for the comparators.

The SR latch is a Reset-dominant latch that does not depend on a clock source. Each of the Set and Reset inputs are active-high. The Set input is driven by the C1 comparator output following the inversion gate, which is accounted for with the C1INV bit. If the effective comparator output signal is low, then the latch can be set by writing '1' to the PULSS bit. Conversely, the Reset input is driven by the C1 comparator output following the inversion gate, which is accounted for with the C2INV bit. If the comparator output signal is low, then the latch can be reset by writing '1' to the PULSR bit.

**REGISTER 8-4: SRCON – SR LATCH CONTROL REGISTER (ADDRESS: 19Eh)**

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | U-0 | U-0 |
|---|---|---|---|---|---|---|---|
| SR1(2) | SR0(2) | C1SEN | C2REN | PULSS | PULSR | — | — |
| bit 7 | | | | | | | bit 0 |

bit 7-6 **SR<1:0>:** SR Latch Configuration bits(2)
- 00 = SR latch is disabled
- 01 = SR latch is enabled. C1OUT pin is the latch non-inverting output. C2OUT pin is the C2 comparator output.
- 10 = SR latch is enabled. C1OUT pin is the C1 comparator output. C2OUT pin is the latch inverting output.
- 11 = SR latch is enabled. C1OUT pin is the latch non-inverting output. C2OUT pin is the latch inverting output.

bit 5 **C1SEN:** C1 Set Enable bit
- 1 = C1 comparator output sets SR latch
- 0 = C1 comparator output has no effect on SR latch

bit 4 **C2REN:** C2 Reset Enable bit
- 1 = C2 comparator output resets SR latch
- 0 = C2 comparator output has no effect on SR latch

bit 3 **PULSS:** Pulse the SET Input of the SR Latch bit
- 1 = Pulse input
- 0 = Always reads back '0'

bit 2 **PULSR:** Pulse the Reset Input of the SR Latch bit
- 1 = Pulse input
- 0 = Always reads back '0'

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

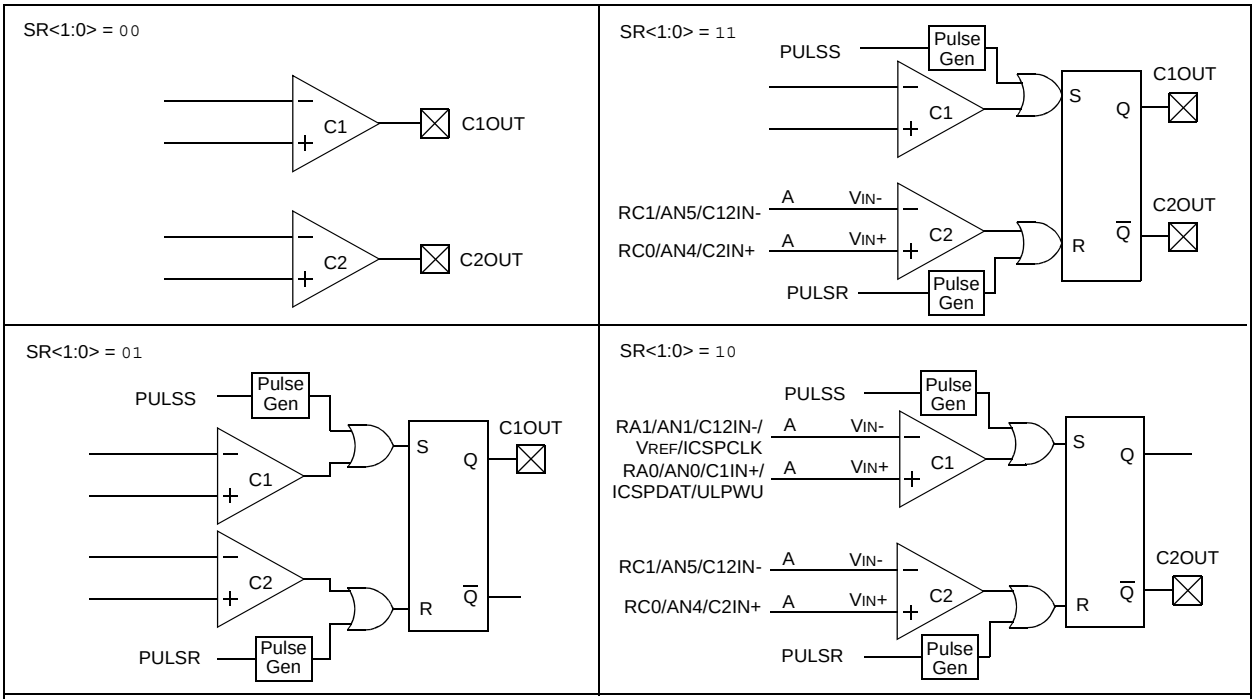
**Note 1:** The C1OUT or C2OUT bits in the CM1CON0 and CM2CON0 registers, respectively, will always reflect the actual comparator outputs (not the pins), regardless the SR latch operation.

**2:** To enable the SR Latch output to the pins, the appropriate C1OE, C2OE, TRISA2 and TRISC4 bits (CM1CON0, CM2CON0, TRISA and TRISC registers, respectively) must be properly configured.

| **Legend:** | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' | |
| -n = Value at POR | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |

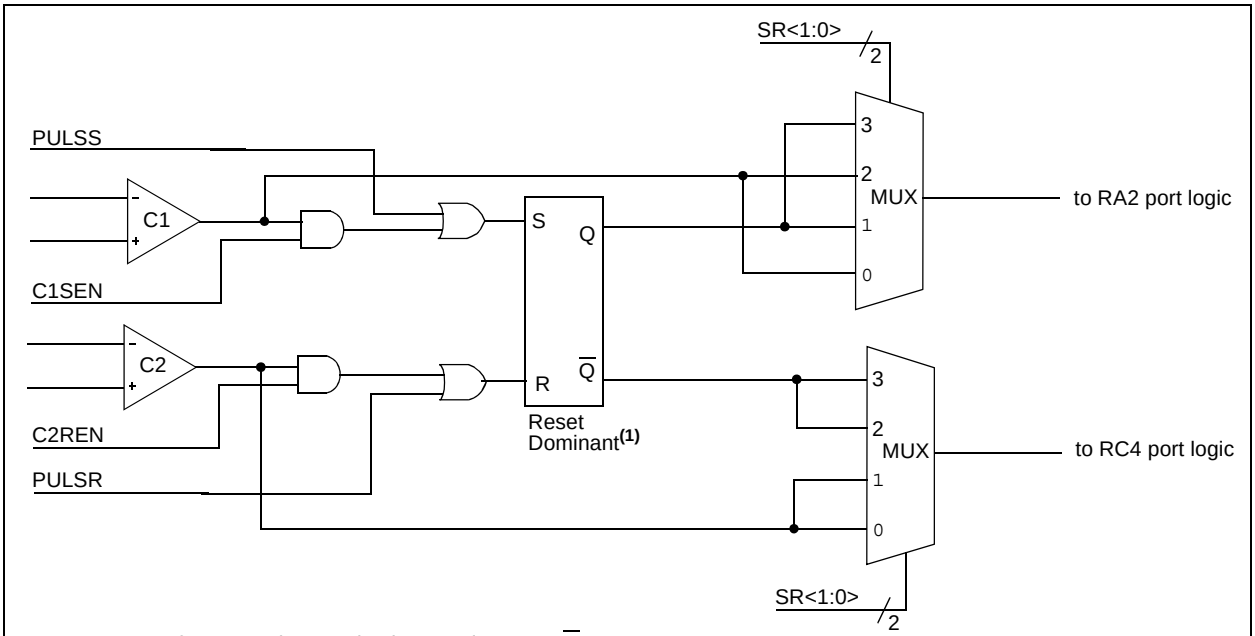
 

**FIGURE 8-3: SR LATCH CONFIGURATIONS**




**Note:** Pulse Generator causes a 1/2 Q-state (1 TOSC) pulse width.

**FIGURE 8-4: SR LATCH SIMPLIFIED BLOCK DIAGRAM**




**Note 1:** If R = 1 and S = 1 simultaneously, Q = 0, $\overline{Q}$ = 1

## 8.4 Comparator Reference

The comparator module also allows the selection of an internally generated voltage reference for one of the comparator inputs. There are two voltage references available in the PIC16F685/687/689/690: The voltage referred to as the comparator reference ($CV_{REF}$) is a variable voltage based on $V_{DD}$; The voltage referred to as the VP6 reference is a fixed voltage derived from a stable band gap source. Each source may be individually routed internally to the comparators. The VRCON register (Register 8-5) controls the voltage reference module shown in Figure 8-5.

### 8.4.1 CONFIGURING THE VOLTAGE REFERENCE

The voltage reference can output 32 distinct voltage levels, 16 in a high range and 16 in a low range.

The following equation determines the output voltages:

**EQUATION 8-1: VOLTAGE REFERENCE OUTPUT VOLTAGE**

*VRR* = 1 *(low range):* $CV_{REF} = (VR<3:0>/24) \times V_{DD}$

*VRR* = 0 *(high range):*

$CV_{REF} = (V_{DD}/4) + (VR<3:0> \times V_{DD}/32)$

### 8.4.2 VOLTAGE REFERENCE ACCURACY/ERROR

The full range of $V_{SS}$ to $V_{DD}$ cannot be realized due to the construction of the module. The transistors on the top and bottom of the resistor ladder network (Figure 8-5) keep $CV_{REF}$ from approaching $V_{SS}$ or $V_{DD}$. The exception is when the module is disabled by clearing C1VREN and C2VREN bits (VRCON<7:6>). When disabled, the reference voltage is $V_{SS}$ when VR<3:0> is '0000' and the VRR (VRCON<5>) bit is set. This allows the comparators to detect a zero-crossing and not consume $CV_{REF}$ module current.

The voltage reference is $V_{DD}$ derived and therefore, the $CV_{REF}$ output changes with fluctuations in $V_{DD}$. The tested absolute accuracy of the comparator voltage Reference can be found in **Section 17.0 "Electrical Specifications"**.

### 8.4.3 VP6 REFERENCE

The VP6 reference has a constant voltage output of 0.6V nominal. This reference can be enabled by setting the VP6EN bit to '1' (VRCON<4>). This reference is always enabled when the HFINTOSC oscillator is active.

### 8.4.4 VP6 STABILIZATION PERIOD

When the voltage reference module is enabled, it will require some time for the reference and its amplifier circuits to stabilize. The user program must include a small delay routine to allow the module to settle. See the electrical specifications section for the minimum delay requirement.

**REGISTER 8-5: VRCON – VOLTAGE REFERENCE CONTROL REGISTER (ADDRESS: 118h)**

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| C1VREN | C2VREN | VRR | VP6EN | VR3 | VR2 | VR1 | VR0 |
| bit 7 | | | | | | | bit 0 |

bit 7 **C1VREN**: Comparator 1 Voltage Reference Enable bit
1 = $CV_{REF}$ circuit powered on and routed to $C1V_{REF}$ input of Comparator 1.
0 = 0.6 Volt constant reference routed to $C1V_{REF}$ input of Comparator 1.

bit 6 **C2VREN**: Comparator 2 Voltage Reference Enable bit
1 = $CV_{REF}$ circuit powered on and routed to $C2V_{REF}$ input of Comparator 2.
0 = 0.6 Volt constant reference routed to $C2V_{REF}$ input of Comparator 2.

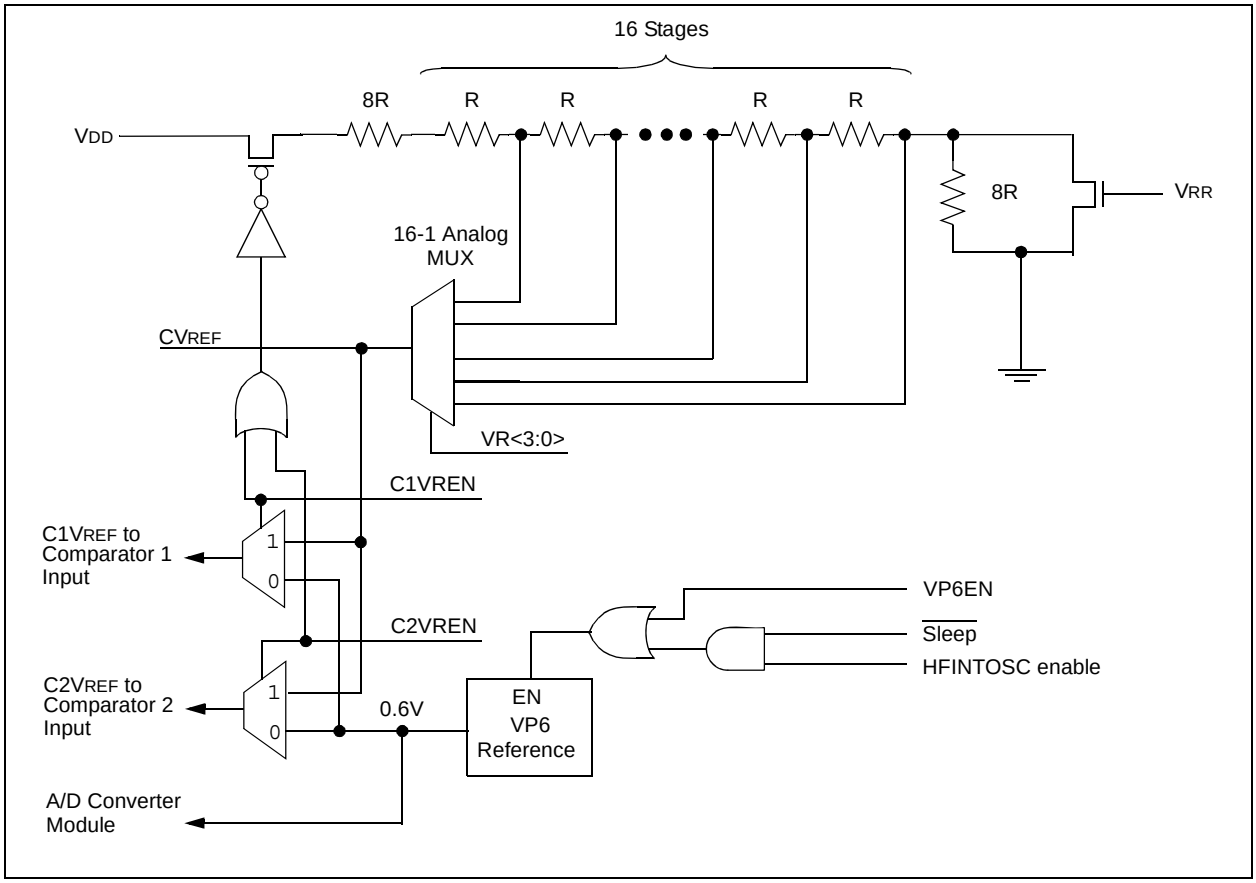
bit 5 **VRR**: Comparator Voltage Reference $CV_{REF}$ Range Selection bit
1 = Low Range
0 = High Range

bit 4 **VP6EN**: 0.6V Reference Enable bit
1 = enabled
0 = disabled

bit 3-0 **VR<3:0>**: Comparator Voltage Reference $CV_{REF}$ Value Selection $0 \leq$ VR<3:0> $\leq 15$
When VRR = 1: $CV_{REF} = (VR<3:0>/24) * V_{DD}$
When VRR = 0: $CV_{REF} = V_{DD}/4 + (VR<3:0>/32) * V_{DD}$

| **Legend:** | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as ‘0’ | |
| -n = Value at POR | ‘1’ = Bit is set | ‘0’ = Bit is cleared | x = Bit is unknown |

**FIGURE 8-5: COMPARATOR VOLTAGE REFERENCE BLOCK DIAGRAM**



## 8.5 Comparator Response Time

Response time is the minimum time, after selecting a new reference voltage or input source, before the comparator output is ensured to have a valid level. If the internal reference is changed, the maximum delay of the internal voltage reference must be considered when using the comparator outputs. Otherwise, the maximum delay of the comparators should be used (Table 17-8).

## 8.6 Operation During Sleep

The comparators and voltage reference, if enabled before entering Sleep mode, remain active during Sleep. This results in higher Sleep currents than shown in the power-down specifications. The additional current consumed by the comparator and the voltage reference is shown separately in the specifications. To minimize power consumption while in Sleep mode, turn off the comparator, CMxCON0<7> = 0, and voltage reference, VRCON<7:6> = 00.

While the comparator is enabled during Sleep, an interrupt will wake-up the device. If the GIE bit (INTCON<7>) is set, the device will jump to the interrupt vector (0004h), and if clear, continues execution with the next instruction. If the device wakes up from Sleep, the contents of the CM1CON0, CM2CON0 and VRCON registers are not affected.

## 8.7 Effects of a Reset

A device Reset forces the CM1CON0, CM2CON0 and VRCON registers to their Reset states. This forces the comparator module to be in the Comparator Reset mode, CMxCON0<7> = 0, and the voltage reference to its OFF state. Thus, all potential inputs are analog

**TABLE 8-3: REGISTERS ASSOCIATED WITH COMPARATOR MODULE**

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 05h, 105h | PORTA | — | — | RA5 | RA4 | RA3 | RA2 | RA1 | RA0 | --xx xxxx | --uu uuuu |
| 07h, 107h | PORTC | RC7 | RC6 | RC5 | RC4 | RC3 | RC2 | RC1 | RC0 | xxxx xxxx | uuuu uuuu |
| 0Bh/8Bh/ 10Bh/18Bh | INTCON | GIE | PEIE | T0IE | INTE | RABIE | T0IF | INTF | RABIF | 0000 000x | 0000 000x |
| 0Ch | PIR1 | — | ADIF | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | -000 0000 | -000 0000 |
| 8Ch | PIE1 | — | ADIE | RCIE | TXIE | SSPIE | CCP1IE | TMR2IE | TMR1IE | -000 0000 | -000 0000 |
| 85h/185h | TRISA | — | — | TRISA5 | TRISA4 | TRISA3 | TRISA2 | TRISA1 | TRISA0 | --11 1111 | --11 1111 |
| 87h/187h | TRISC | TRISC7 | TRISC6 | TRISC5 | TRISC4 | TRISC3 | TRISC2 | TRISC1 | TRISC0 | 1111 1111 | 1111 1111 |
| 118h | VRCON | C1VREN | C2VREN | VRR | VP6EN | VR3 | VR2 | VR1 | VR0 | 0000 0000 | 0000 0000 |
| 119h | CM1CON0 | C1ON | C1OUT | C1OE | C1POL | — | C1R | C1CH1 | C1CH0 | 0000 0000 | 0000 -000 |
| 11Ah | CM2CON0 | C2ON | C2OUT | C2OE | C2POL | — | C2R | C2CH1 | C2CH0 | 0000 0000 | 0000 -000 |
| 11Bh | CM2CON1 | MC1OUT | MC2OUT | — | — | — | — | T1GSS | C2SYNC | 00-- --10 | 00-- --10 |
| 11Eh | ANSEL | ANS7 | ANS6 | ANS5 | ANS4 | ANS3 | ANS2 | ANS1 | ANS0 | 1111 1111 | 1111 1111 |
| 19Eh | SRCON | SR1 | SR0 | C1SEN | C2SEN | PULSS | PULSR | — | — | 0000 00-- | 0000 00-- |

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

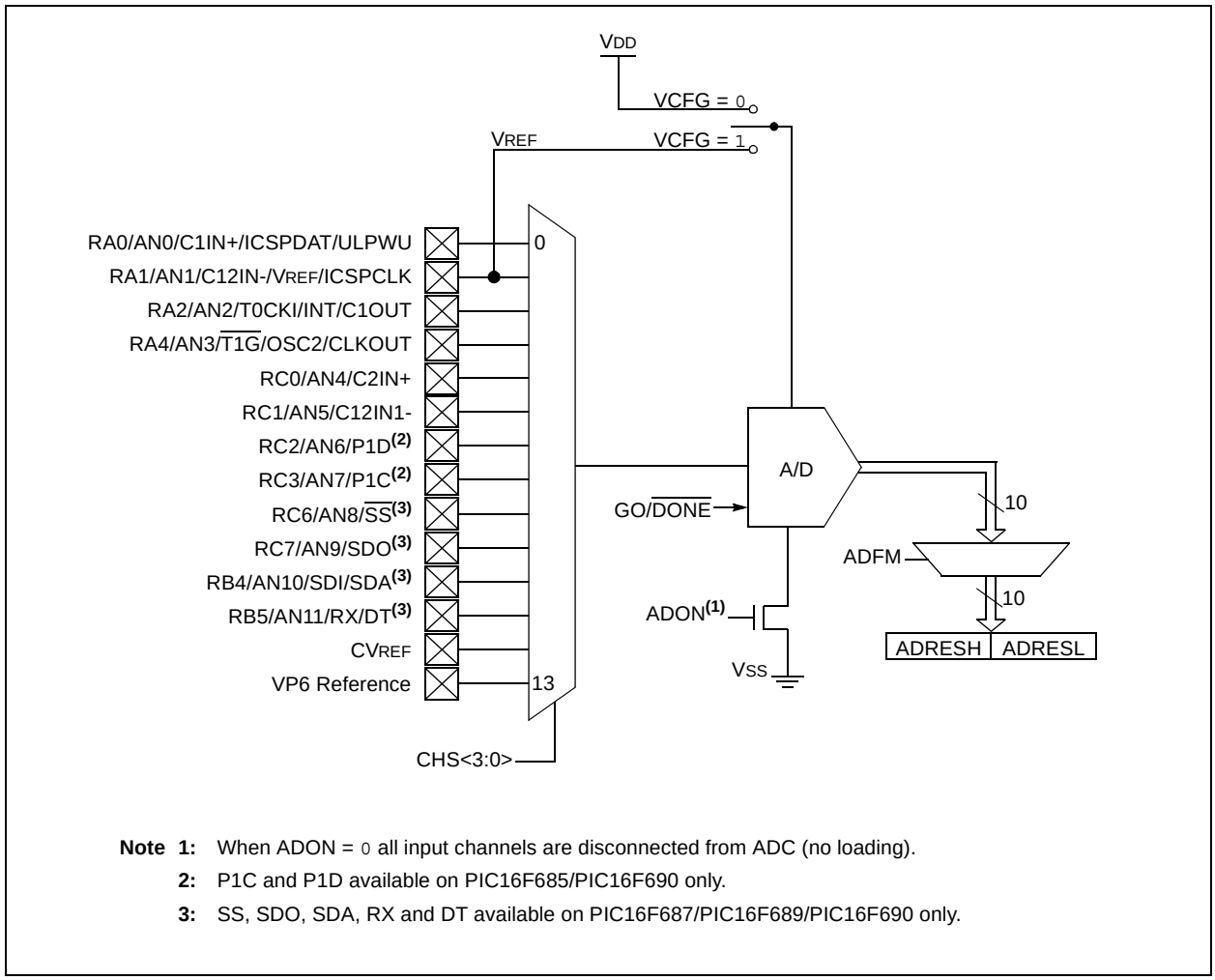
**NOTES:**

## 9.0 ANALOG-TO-DIGITAL CONVERTER (A/D) MODULE

The analog-to-digital converter (A/D) allows conversion of an analog input signal to a 10-bit binary representation of that signal. The PIC16F685/687/689/690 has twelve analog I/O inputs, plus two internal inputs, multiplexed into one sample and hold circuit. The output of the sample and hold is connected to the input of the converter. The converter generates a binary result via successive approximation and stores the resulting or remaining 10 bits of data into ADRESL (9Eh) and ADRESH (1Eh). The voltage reference used in the conversion is software selectable to either VDD or a voltage applied by the VREF pin. Figure 9-1 shows the block diagram of the A/D on the PIC16F685/687/689/690.

**FIGURE 9-1: A/D BLOCK DIAGRAM**




**Note 1:** When ADON = 0 all input channels are disconnected from ADC (no loading).

**2:** P1C and P1D available on PIC16F685/PIC16F690 only.

**3:** SS, SDO, SDA, RX and DT available on PIC16F687/PIC16F689/PIC16F690 only.

## 9.1 A/D Configuration and Operation

There are four registers available to control the functionality of the A/D module:

1. ANSEL (Register 9-1)
2. ANSELH (Register 9-2)
3. ADCON0 (Register 9-3)
4. ADCON1 (Register 9-4)

### 9.1.1 ANALOG PORT PINS

The ANS<11:0> bits (ANSEL<7:0> and ANSELH<3:0>) and the TRISA<4,2:0>, TRISB<5:4> and TRISC<7:6,3:0>> bits control the operation of the A/D port pins. Set the corresponding TRISx bits to '1' to set the pin output driver to its high-impedance state. Likewise, set the corresponding ANSx bit to disable the digital input buffer.

| **Note:** | Analog voltages on any pin that is defined as a digital input may cause the input buffer to conduct excess current. |
|---|---|

### 9.1.2 CHANNEL SELECTION

There are fourteen analog channels on PIC16F685/687/689/690. The CHS<3:0> bits (ADCON0<5:2>) control which channel is connected to the sample and hold circuit.

### 9.1.3 VOLTAGE REFERENCE

There are two options for the voltage reference to the A/D converter: either VDD is used or an analog voltage applied to VREF is used. The VCFG bit (ADCON0<6>) controls the voltage reference selection. If VCFG is set, then the voltage on the VREF pin is the reference; otherwise, VDD is the reference.

### 9.1.4 CONVERSION CLOCK

The A/D conversion cycle requires 11 TAD. The source of the conversion clock is software selectable via the ADCS bits (ADCON1<6:4>). There are seven possible clock options:

- FOSC/2
- FOSC/4
- FOSC/8
- FOSC/16
- FOSC/32
- FOSC/64
- FRC (dedicated internal oscillator)

For correct conversion, the A/D conversion clock (1/TAD) must be selected to ensure a minimum TAD of 1.6 µs. Table 9-1 shows a few TAD calculations for selected frequencies.

**TABLE 9-1: TAD VS. DEVICE OPERATING FREQUENCIES**

| A/D Clock Source (TAD) | | Device Frequency | | | |
|---|---|---|---|---|---|
| Operation | ADCS<2:0> | 20 MHz | 5 MHz | 4 MHz | 1.25 MHz |
| 2 TOSC | 000 | 100 ns(2) | 400 ns(2) | 500 ns(2) | 1.6 µs |
| 4 TOSC | 100 | 200 ns(2) | 800 ns(2) | 1.0 µs(2) | 3.2 µs |
| 8 TOSC | 001 | 400 ns(2) | 1.6 µs | 2.0 µs | 6.4 µs |
| 16 TOSC | 101 | 800 ns(2) | 3.2 µs | 4.0 µs | 12.8 µs(3) |
| 32 TOSC | 010 | 1.6 µs | 6.4 µs | 8.0 µs(3) | 25.6 µs(3) |
| 64 TOSC | 110 | 3.2 µs | 12.8 µs(3) | 16.0 µs(3) | 51.2 µs(3) |
| A/D RC | x11 | 2-6 µs(1,4) | 2-6 µs(1,4) | 2-6 µs(1,4) | 2-6 µs(1,4) |

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

**Note 1:** The A/D RC source has a typical TAD time of 4 µs for VDD > 3.0V.

**2:** These values violate the minimum required TAD time.

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

**4:** When the device frequency is greater than 1 MHz, the A/D RC clock source is only recommended if the conversion will be performed during Sleep.

### 9.1.5 STARTING A CONVERSION

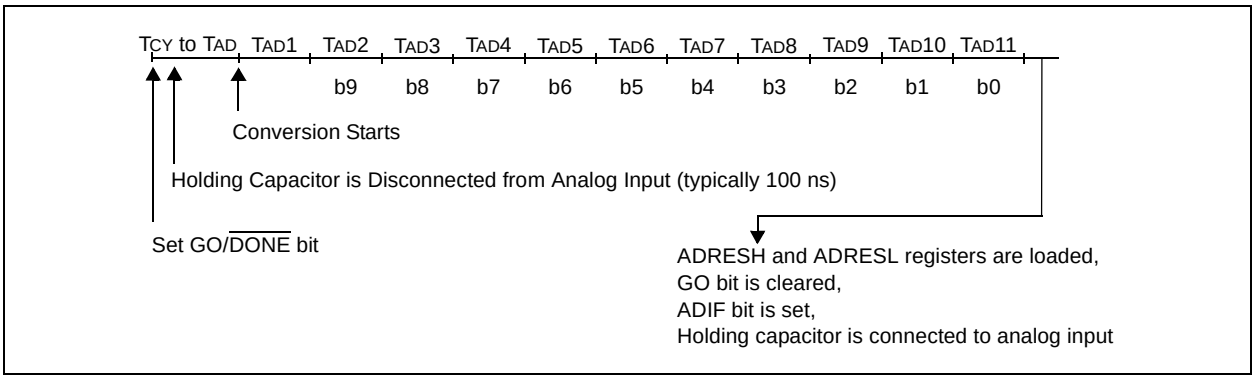
The A/D conversion is initiated by setting the GO/$\overline{\text{DONE}}$ bit (ADCON0<1>). When the conversion is complete, the A/D module:

- Clears the GO/$\overline{\text{DONE}}$ bit
- Sets the ADIF flag (PIR1<6>)
- Generates an interrupt (if enabled)

If the conversion must be aborted, the GO/$\overline{\text{DONE}}$ bit can be cleared in software. The ADRESH:ADRESL registers will not be updated with the partially complete A/D conversion sample. Instead, the ADRESH:ADRESL registers will retain the value of the previous conversion. After an aborted conversion, a 2 TAD delay is required before another acquisition can be initiated. Following the delay, an input acquisition is automatically started on the selected channel.

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

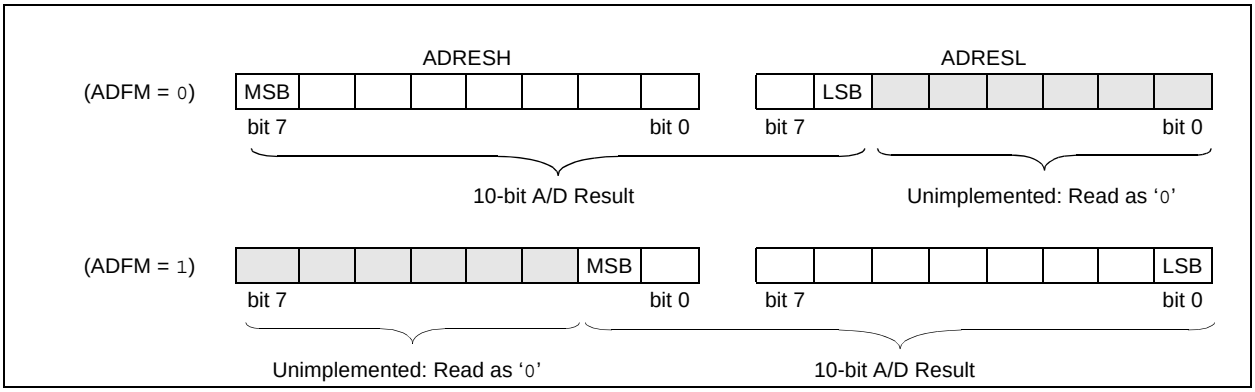
**FIGURE 9-2: A/D CONVERSION TAD CYCLES**




### 9.1.6 CONVERSION OUTPUT

The A/D conversion can be supplied in two formats: left or right justified. The ADFM bit (ADCON0<7>) controls the output format. Figure 9-3 shows the output formats.

**FIGURE 9-3: 10-BIT A/D RESULT FORMAT**



**REGISTER 9-1: ANSEL – ANALOG SELECT REGISTER (ADDRESS: 11Eh)**

| R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 |
|---|---|---|---|---|---|---|---|
| ANS7 | ANS6 | ANS5 | ANS4 | ANS3 | ANS2 | ANS1 | ANS0 |
| bit 7 | | | | | | | bit 0 |

bit 7-0 **ANS<7:0>**: Analog Select bits

Select between analog or digital function on pins AN<7:0>, respectively.

1 = Analog input. Pin is assigned as analog input.(1)

0 = Digital I/O. Pin is assigned to port or special function.

**Note 1:** Setting a pin to an analog input automatically disables the digital input circuitry, weak pull-ups, and interrupt-on-change if available. The corresponding TRIS bit must be set to Input mode in order to allow external control of the voltage on the pin.

| **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 |

**REGISTER 9-2: ANSELH – ANALOG SELECT HIGH REGISTER (ADDRESS: 11Fh)**

| U-0 | U-0 | U-0 | U-0 | R/W-1 | R/W-1 | R/W-1 | R/W-1 |
|---|---|---|---|---|---|---|---|
| — | — | — | — | ANS11 | ANS10 | ANS9 | ANS8 |
| bit 7 | | | | | | | bit 0 |

bit 7-4 **Unimplemented:** Read as '0'.

bit 3-0 **ANS<11:8>**: Analog Select bits

Select between analog or digital function on pins AN<11:8>, respectively.

1 = Analog input. Pin is assigned as analog input.(1)

0 = Digital I/O. Pin is assigned to port or special function.

**Note 1:** Setting a pin to an analog input automatically disables the digital input circuitry, weak pull-ups, and interrupt-on-change if available. The corresponding TRIS bit must be set to Input mode in order to allow external control of the voltage on the pin.

| **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 |

**TABLE 9-2: ANALOG SELECT CROSS REFERENCE**

| Analog | I/O Pins | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **RB5** | **RB4** | **RC7** | **RC6** | **RC3** | **RC2** | **RC1** | **RC0** | **RA4** | **RA2** | **RA1** | **RA0** |
| Select | ANS11 | ANS10 | ANS9 | ANS8 | ANS7 | ANS6 | ANS5 | ANS4 | ANS3 | ANS2 | ANS1 | ANS0 |
| Channel | AN11 | AN10 | AN9 | AN8 | AN7 | AN6 | AN5 | AN4 | AN3 | AN2 | AN1 | AN0 |

**REGISTER 9-3: ADCON0 – A/D CONTROL REGISTER (ADDRESS: 1Fh)**

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| ADFM | VCFG | CHS3 | CHS2 | CHS1 | CHS0 | GO/$\overline{\text{DONE}}$ | ADON |
| bit 7 | | | | | | | bit 0 |

bit 7 **ADFM:** A/D Result Formed Select bit
1 = Right justified
0 = Left justified

bit 6 **VCFG:** Voltage Reference bit
1 = $V_{REF}$ pin
0 = $V_{DD}$

bit 5-2 **CHS<3:0>:** Analog Channel Select bits
0000 = Channel 00 (AN0)
0001 = Channel 01 (AN1)
0010 = Channel 02 (AN2)
0011 = Channel 03 (AN3)
0100 = Channel 04 (AN4)
0101 = Channel 05 (AN5)
0110 = Channel 06 (AN6)
0111 = Channel 07 (AN7)
1000 = Channel 08 (AN8)
1001 = Channel 09 (AN9)
1010 = Channel 10 (AN10)
1011 = Channel 11 (AN11)
1100 = $CV_{REF}$
1101 = VP6
1110 = Reserved. Do not use.
1111 = Reserved. Do not use.

bit 1 **GO/$\overline{\text{DONE}}$:** A/D Conversion Status bit
1 = A/D conversion cycle in progress. Setting this bit starts an A/D conversion cycle.
This bit is automatically cleared by hardware when the A/D conversion has completed.
0 = A/D conversion completed/not in progress

bit 0 **ADON:** A/D Enable bit
1 = A/D converter module is enabled
0 = A/D converter is shut off and consumes no operating current

| **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 |

**REGISTER 9-4: ADCON1 – A/D CONTROL REGISTER 1 (ADDRESS: 9Fh)**

| U-0 | R/W-0 | R/W-0 | R/W-0 | U-0 | U-0 | U-0 | U-0 |
|---|---|---|---|---|---|---|---|
| — | ADCS2 | ADCS1 | ADCS0 | — | — | — | — |
| bit 7 | | | | | | | bit 0 |

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

bit 6-4 **ADCS<2:0>:** A/D Conversion Clock Select bits

000 = FOSC/2
001 = FOSC/8
010 = FOSC/32
x11 = FRC (clock derived from a dedicated internal oscillator = 500 kHz max)
100 = FOSC/4
101 = FOSC/16
110 = FOSC/64

bit 3-0 **Unimplemented:** Read as '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 |

### 9.1.7 CONFIGURING THE A/D

After the A/D module has been configured as desired, the selected channel must be acquired before the conversion is started. The analog input channels must have their corresponding TRIS bits selected as inputs.

To determine sample time, see Tables 17-16 and 17-17. After this sample time has elapsed the A/D conversion can be started.

These steps should be followed for an A/D conversion:

1. Configure the A/D module:
   - Configure analog/digital I/O (ANSx)
   - Select A/D conversion clock (ADCON1<6:4>)
   - Configure voltage reference (ADCON0<6>)
   - Select A/D input channel (ADCON0<5:2>)
   - Select result format (ADCON0<7>)
   - Turn on A/D module (ADCON0<0>)
2. Configure A/D interrupt (if desired):
   - Clear ADIF bit (PIR1<6>)
   - Set ADIE bit (PIE1<6>)
   - Set PEIE and GIE bits (INTCON<7:6>)
3. Wait the required acquisition time.
4. Start conversion:
   - Set GO/$\overline{\text{DONE}}$ bit (ADCON0<1>)
5. Wait for A/D conversion to complete, by either:
   - Polling for the GO/$\overline{\text{DONE}}$ bit to be cleared (with interrupts disabled); OR
   - Waiting for the A/D interrupt
6. Read A/D Result register pair (ADRESH:ADRESL), clear bit ADIF if required.
7. For next conversion, go to step 1 or step 2 as required. The A/D conversion time per bit is defined as TAD. A minimum wait of 2 TAD is required before the next acquisition starts.

**EXAMPLE 9-1: A/D CONVERSION**

```
;This code block configures the A/D
;for polling, Vdd reference, R/C clock
;and RA0 input.
;
;Conversion start & wait for complete
;polling code included.
;
BSF    STATUS,RP0     ;Bank 1
BCF    STATUS,RP1     ;
MOVLW  B'01110000'    ;A/D RC clock
MOVWF  ADCON1         ;
BSF    TRISA,0        ;Set RA0 to input
BCF    STATUS,RP0     ;Bank 2
BSF    STATUS,RP1     ;
BSF    ANSEL,0        ;Set RA0 to analog
BCF    STATUS,RP0     ;Bank 0
MOVLW  B'10000001'    ;Right, Vdd Vref, AN0
MOVWF  ADCON0         ;
CALL   SampleTime     ;Wait min sample time
BSF    ADCON0,GO      ;Start conversion
BCF    STATUS,RP1     ;
BTFSC  ADCON0,GO      ;Is conversion done?
GOTO   $-1            ;No, test again
MOVF   ADRESH,W       ;Read upper 2 bits
MOVWF  RESULTHI       ;
BSF    STATUS,RP0     ;Bank 1
MOVF   ADRESL,W       ;Read lower 8 bits
BCF    STATUS,RP0     ;Bank 0
MOVWF  RESULTLO
```

## 9.2 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 9-4. 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 ($V_{DD}$), see Figure 9-4. **The maximum recommended impedance for analog sources is 10 kΩ**. As the impedance is decreased, the acquisition time may be decreased. 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, Equation 9-1 may be used. This equation assumes that 1/2 LSb error is used (1024 steps for the A/D). The 1/2 LSb error is the maximum error allowed for the A/D to meet its specified resolution.

**EQUATION 9-1: ACQUISITION TIME EXAMPLE**

*Assumptions:* $Temperature = 50°C$ *and external impedance of* $10k\Omega$ $5.0V$ $V_{DD}$

$$T_{ACQ} = \text{Amplifier Settling Time} + \text{Hold Capacitor Charging Time} + \text{Temperature Coefficient}$$
$$= T_{AMP} + T_C + T_{COFF}$$
$$= 2\mu s + T_C + [(\text{Temperature} - 25°C)(0.05\mu s/°C)]$$

*The value for* $T_C$ *can be approximated with the following equations:*

$$V_{APPLIED}\left(1 - \frac{1}{2047}\right) = V_{CHOLD} \quad ;[1]\ V_{CHOLD}\ \text{charged to within 1/2 lsb}$$

$$V_{APPLIED}\left(1 - e^{\frac{-T_C}{RC}}\right) = V_{CHOLD} \quad ;[2]\ V_{CHOLD}\ \text{charge response to}\ V_{APPLIED}$$

$$V_{APPLIED}\left(1 - e^{\frac{-T_C}{RC}}\right) = V_{APPLIED}\left(1 - \frac{1}{2047}\right) \quad ;\text{combining [1] and [2]}$$

*Solving for* $T_C$*:*

$$T_C = -C_{HOLD}(R_{IC} + R_{SS} + R_S)\ ln(1/2047)$$
$$= -10pF(1k\Omega + 7k\Omega + 10k\Omega)\ ln(0.0004885)$$
$$= 1.37\mu s$$

*Therefore:*

$$T_{ACQ} = 2\mu s + 1.37\mu s + [(50°C - 25°C)(0.05\mu s/°C)]$$
$$= 4.67\mu s$$

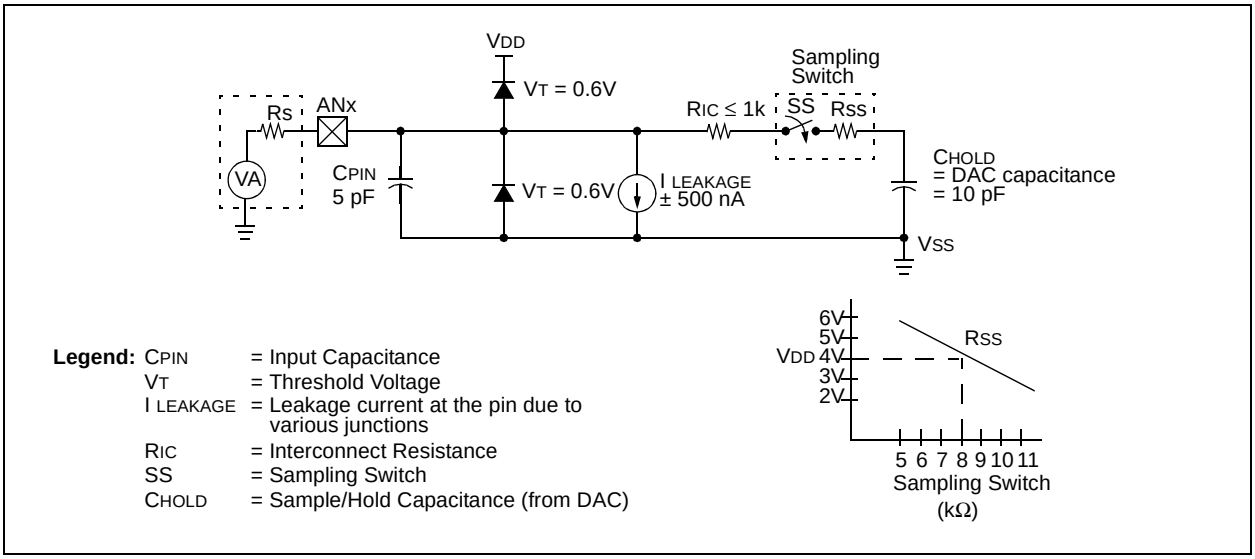
 

**Note 1:** The reference voltage ($V_{REF}$) has no effect on the equation, since it cancels itself out.

**2:** The charge holding capacitor ($C_{HOLD}$) is not discharged after each conversion.

**3:** The maximum recommended impedance for analog sources is 10 kΩ. This is required to meet the pin leakage specification.

**FIGURE 9-4: ANALOG INPUT MODEL**

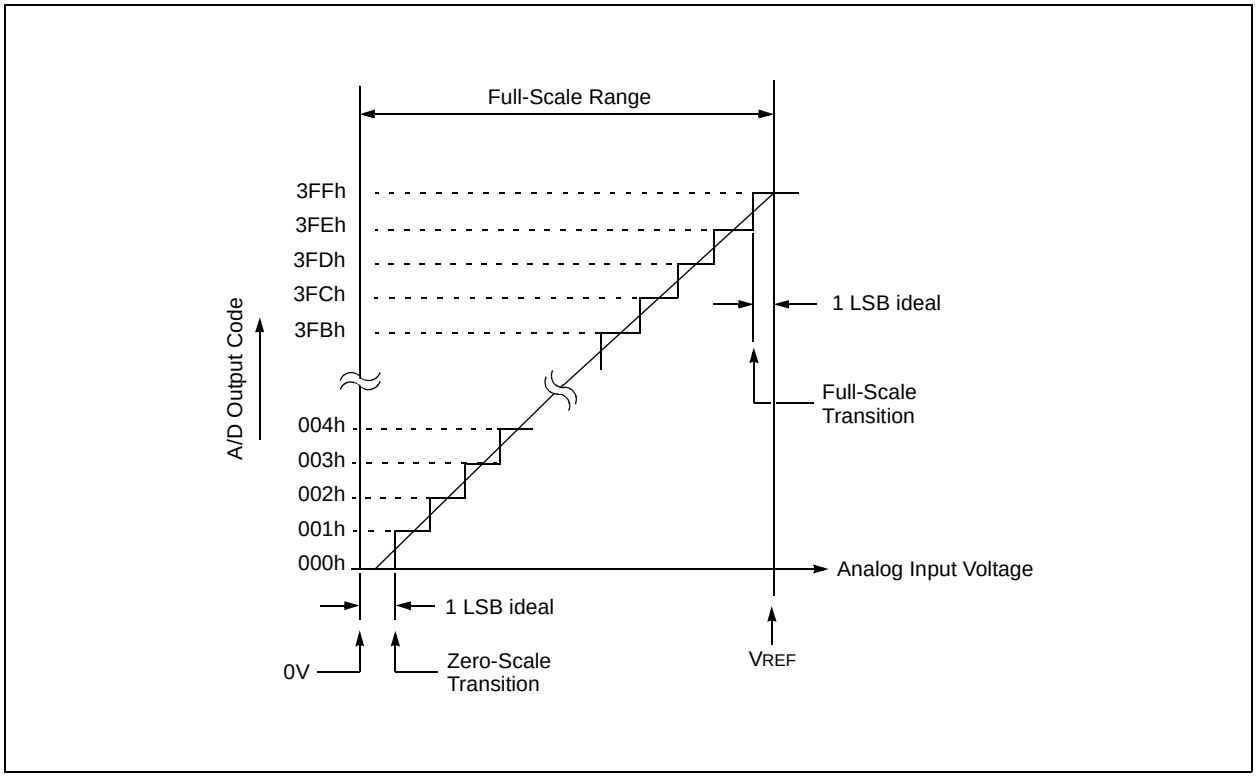


## 9.3 A/D Operation During Sleep

The A/D converter module can operate during Sleep. This requires the A/D clock source to be set to the FRC option. When the RC clock source is selected, the A/D waits one instruction before starting the conversion. This allows the SLEEP instruction to be executed, thus eliminating much of the switching noise from the conversion. When the conversion is complete, the GO/$\overline{\text{DONE}}$ bit is cleared and the result is loaded into the ADRESH:ADRESL registers. If the A/D interrupt is enabled, the device awakens from Sleep. If the GIE bit (INTCON<7>) is set, the program counter is set to the interrupt vector (0004h). If GIE is clear, the next instruction is executed. If the A/D interrupt is not enabled (ADIE and PEIE bits set), the A/D module is turned off, although the ADON bit remains set.

When the A/D clock source is something other than RC, a SLEEP instruction causes the present conversion to be aborted and the A/D module is turned off. The ADON bit remains set.

**FIGURE 9-5: A/D TRANSFER FUNCTION**



## 9.4 Effects of Reset

A device Reset forces all registers to their Reset state. Thus, the A/D module is turned off and any pending conversion is aborted. The ADRESH:ADRESL registers are unchanged.

## 9.5 Use of the CCP Trigger

An A/D conversion can be started by the "special event trigger" of the CCP module. This requires that the CCP1M<3:0> bits (CCP1CON<3:0>) be programmed as '1011' and that the A/D module is enabled (ADON bit is set). When the trigger occurs, the GO/$\overline{\text{DONE}}$ 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 ADRESH:ADRESL to the desired location).

The appropriate analog input channel must be selected and the minimum acquisition done before the "special event trigger" sets the GO/$\overline{\text{DONE}}$ 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. See **Section 11.0 "Enhanced Capture/Compare/PWM+ (ECCP+) Module"** for more information.

**TABLE 9-3: SUMMARY OF A/D REGISTERS**

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 05h/105h | PORTA | — | — | RA5 | RA4 | RA3 | RA2 | RA1 | RA0 | --xx xxxx | --uu uuuu |
| 06h/106h | PORTB | RB7 | RB6 | RB5 | RB4 | — | — | — | — | xxxx ---- | uuuu ---- |
| 07h/107h | PORTC | RC7 | RC6 | RC5 | RC4 | RC3 | RC2 | RC1 | RC0 | xxxx xxxx | uuuu uuuu |
| 0Bh/8Bh/ 10Bh/18Bh | INTCON | GIE | PEIE | T0IE | INTE | RABIE | T0IF | INTF | RABIF | 0000 000x | 0000 000x |
| 0Ch | PIR1 | — | ADIF | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | -000 0000 | -000 0000 |
| 11Eh | ANSEL | ANS7 | ANS6 | ANS5 | ANS4 | ANS3 | ANS2 | ANS1 | ANS0 | 1111 1111 | 1111 1111 |
| 11Fh | ANSELH | — | — | — | — | ANS11 | ANS10 | ANS9 | ANS8 | ---- 1111 | ---- 1111 |
| 1Eh | ADRESH | A/D Result Register High Byte | | | | | | | | xxxx xxxx | uuuu uuuu |
| 1Fh | ADCON0 | ADFM | VCFG | CHS3 | CHS2 | CHS1 | CHS0 | GO/$\overline{\text{DONE}}$ | ADON | 0000 0000 | 0000 0000 |
| 85h/185h | TRISA | — | — | TRISA5 | TRISA4 | TRISA3 | TRISA2 | TRISA1 | TRISA0 | --11 1111 | --11 1111 |
| 86h/186h | TRISB | TRISB7 | TRISB6 | TRISB5 | TRISB4 | — | — | — | — | 1111 ---- | 1111 ---- |
| 87h/187h | TRISC | TRISC7 | TRISC6 | TRISC5 | TRISC4 | TRISC3 | TRISC2 | TRISC1 | TRISC0 | 1111 1111 | 1111 1111 |
| 8Ch | PIE1 | — | ADIE | RCIE | TXIE | SSPIE | CCP1IE | TMR2IE | TMR1IE | -000 0000 | -000 0000 |
| 9Eh | ADRESL | A/D Result Register Low Byte | | | | | | | | xxxx xxxx | uuuu uuuu |
| 9Fh | ADCON1 | — | ADCS2 | ADCS1 | ADCS0 | — | — | — | — | -000 ---- | -000 ---- |

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

**NOTES:**

## 10.0 DATA EEPROM AND FLASH PROGRAM MEMORY CONTROL

Data EEPROM memory is readable and writable and the Flash program memory is readable during normal operation (full VDD range). These memories are not directly mapped in the register file space. Instead, they are indirectly addressed through the Special Function Registers. There are six SFRs used to access these memories:

- EECON1
- EECON2
- EEDAT
- EEDATH
- EEADR
- EEADRH

When interfacing the data memory block, EEDAT holds the 8-bit data for read/write, and EEADR holds the address of the EEDAT location being accessed. This device has 256 bytes of data EEPROM with an address range from 0h to 0FFh.

When interfacing the program memory block, the EEDAT and EEDATH registers form a 2-byte word that holds the 14-bit data for read/write, and the EEADR and EEADRH registers form a 2-byte word that holds the 12-bit address of the EEPROM location being accessed. This device has 4K words of program EEPROM with an address range from 0h to 0FFFh. The program memory allows one-word reads.

The EEPROM data memory allows byte read and write. A byte write automatically erases the location and writes the new data (erase before write).

The write time is controlled by an on-chip timer. The write/erase voltages are generated by an on-chip charge pump rated to operate over the voltage range of the device for byte or word operations.

When the device is code-protected, the CPU may continue to read and write the data EEPROM memory and read the program memory. When code-protected, the device programmer can no longer access data or program memory.

### 10.1 EEADR and EEADRH Registers

The EEADR and EEADRH registers can address up to a maximum of 256 bytes of data EEPROM or up to a maximum of 4K words of program EEPROM.

When selecting a program address value, the MSB of the address is written to the EEADRH register and the LSB is written to the EEADR register. When selecting a data address value, only the LSB of the address is written to the EEADR register.

#### 10.1.1 EECON1 AND EECON2 REGISTERS

EECON1 is the control register for EE memory accesses.

Control bit EEPGD determines if the access will be a program or data memory access. When clear, as it is when reset, any subsequent operations will operate on the data memory. When set, any subsequent operations will operate on the program memory. Program memory can only be read.

Control bits RD and WR initiate read and write, respectively. These bits cannot be cleared, only set, in software. They are cleared in hardware at completion of the read or write operation. The inability to clear the WR bit in software prevents the accidental, premature termination of a write operation.

The WREN bit, when set, will allow a write operation to data EEPROM. On power-up, the WREN bit is clear. The WRERR bit is set when a write operation is interrupted by a $\overline{\text{MCLR}}$ or a WDT Time-out Reset during normal operation. In these situations, following Reset, the user can check the WRERR bit and rewrite the location. The data and address will be unchanged in the EEDAT and EEADR registers.

Interrupt flag bit EEIF (PIR2<4>), is set when write is complete. It must be cleared in the software.

EECON2 is not a physical register. Reading EECON2 will read all '0's. The EECON2 register is used exclusively in the data EEPROM write sequence.

**REGISTER 10-1: EEDAT – EEPROM DATA REGISTER (ADDRESS: 10Ch)**

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| EEDAT7 | EEDAT6 | EEDAT5 | EEDAT4 | EEDAT3 | EEDAT2 | EEDAT1 | EEDAT0 |
| bit 7 | | | | | | | bit 0 |

bit 7-0 **EEDATn**: Byte value to Write to or Read from data EEPROM bits

| **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 |

**REGISTER 10-2: EEADR – EEPROM DATA REGISTER (ADDRESS: 10Dh)**

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| EEDAR7 | EEDAR6 | EEDAR5 | EEDAR4 | EEDAR3 | EEDAR2 | EEDAR1 | EEDAR0 |
| bit 7 | | | | | | | bit 0 |

bit 7-0 **EEDARn**: Byte value to Write to or Read from data EEPROM bits

| **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 |

**REGISTER 10-3: EEDATH – EEPROM DATA HIGH BYTE REGISTER(1) (ADDRESS: 10Eh)**

| U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | — | EEDATH5 | EEDATH4 | EEDATH3 | EEDATH2 | EEDATH1 | EEDATH0 |
| bit 7 | | | | | | | bit 0 |

bit 5-0 **EEDATH<5:0>**: Byte value to Write to or Read from data EEPROM bits or to Read from program memory

**Note 1:** PIC16F685/PIC16F689/PIC16F690 only.

| **Legend:** | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' | |
| - n = Value at POR | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |

**REGISTER 10-4: EEADRH – EEPROM ADDRESS HIGH BYTE REGISTER(1) (ADDRESS: 10Fh)**

| U-0 | U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | — | — | — | EEADRH3 | EEADRH2 | EEADRH1 | EEADRH0 |
| bit 7 | | | | | | | bit 0 |

bit 3-0 **EEADRH<3:0>**: Specifies one of 256 locations for EEPROM Read/Write Operation bits or high bits for program memory reads

**Note 1:** PIC16F685/PIC16F689/PIC16F690 only.

| **Legend:** | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' | |
| - n = Value at POR | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |

**REGISTER 10-5: EECON1 – EEPROM CONTROL REGISTER 1 (ADDRESS: 18Ch)**

| R/W-x | U-0 | U-0 | U-0 | R/W-x | R/W-0 | R/S-0 | R/S-0 |
|---|---|---|---|---|---|---|---|
| EEPGD | — | — | — | WRERR | WREN | WR | RD |
| bit 7 | | | | | | | bit 0 |

bit 7 **EEPGD:** Program/Data EEPROM Select bit
1 = Accesses program memory
0 = Accesses data memory

bit 6-4 **Unimplemented:** Read as '0'

bit 3 **WRERR:** EEPROM Error Flag bit
1 = A write operation is prematurely terminated (any $\overline{\text{MCLR}}$ Reset, any WDT Reset during normal operation or BOR)
0 = The write operation completed

bit 2 **WREN:** EEPROM Write Enable bit
1 = Allows write cycles
0 = Inhibits write to the data EEPROM

bit 1 **WR**: Write Control bit
EEPGD = 1:
This bit is ignored
EEPGD = 0:
1 = Initiates a write cycle (The bit is cleared by hardware once write is complete. The WR bit can only be set, not cleared, in software.)
0 = Write cycle to the data EEPROM is complete

bit 0 **RD**: Read Control bit
1 = Initiates a memory read (the RD is cleared in hardware and can only be set, not cleared, in software.)
0 = Does not initiate a memory read

| **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 |

### 10.1.2 READING THE DATA EEPROM MEMORY

To read a data memory location, the user must write the address to the EEADR register, clear the EEPGD control bit (EECON1<7>), and then set control bit RD (EECON1<0>). The data is available in the very next cycle, in the EEDAT register; therefore, it can be read in the next instruction. EEDAT will hold this value until another read or until it is written to by the user (during a write operation).

**EXAMPLE 10-1: DATA EEPROM READ**

```
BSF     STATUS, RP1     ;Bank 2
BCF     STATUS, RP0     ;
MOVLW   DATA_EE_ADDR    ;
MOVWF   EEADR           ;Data Memory
                        ;Address to read
BSF     STATUS, RP0     ;Bank 3
BCF     EECON1, EEPGD   ;Point to DATA
                        ;memory
BCF     EECON1, RD      ;EE Read
BCF     STATUS, RP1     ;Bank 2
MOVF    EEDAT, W        ;W = EEDAT
BCF     STATUS, RP0     ;Bank 0
```

### 10.1.3 WRITING TO THE DATA EEPROM MEMORY

To write an EEPROM data location, the user must first write the address to the EEADR register and the data to the EEDAT register. Then the user must follow a specific sequence to initiate the write for each byte.

The write will not initiate if the above sequence is not followed exactly (write 55h to EECON2, write AAh to EECON2, then set WR bit) for each byte. Interrupts should be disabled during this code segment.

Additionally, the WREN bit in EECON1 must be set to enable write. This mechanism prevents accidental writes to data EEPROM due to errant (unexpected) code execution (i.e., lost programs). The user should keep the WREN bit clear at all times, except when updating EEPROM. The WREN bit is not cleared by hardware.

After a write sequence has been initiated, clearing the WREN bit will not affect this write cycle. The WR bit will be inhibited from being set unless the WREN bit is set.

At the completion of the write cycle, the WR bit is cleared in hardware and the EE Write Complete Interrupt Flag bit (EEIF) is set. The user can either enable this interrupt or poll this bit. EEIF must be cleared by software.

**EXAMPLE 10-2: DATA EEPROM WRITE**

```
        BCF     STATUS, RP0     ;Bank 2
        BSF     STATUS, RP1     ;
        MOVLW   DATA_EE_ADDR    ;
        MOVWF   EEADR           ;Data Memory Address to write
        MOVLW   DATA_EE_DATA    ;
        MOVWF   EEDAT           ;Data Memory Value to write
        BSF     STATUS, RP0     ;Bank 3
        BCF     EECON1, EEPGD   ;Point to DATA memory
        BSF     EECON1, WREN    ;Enable writes

Required
Sequence
        BCF     INTCON, GIE     ;Disable INTs.
        MOVLW   55h             ;
        MOVWF   EECON2          ;Write 55h
        MOVLW   AAh             ;
        MOVWF   EECON2          ;Write AAh
        BSF     EECON1, WR      ;Set WR bit to begin write
        BSF     INTCON, GIE     ;Enable INTs.

        SLEEP                   ;Wait for interrupt to signal write complete
        BCF     EECON1, WREN    ;Disable writes
        BCF     STATUS, RP0     ;Bank 0
```

### 10.1.4 READING THE FLASH PROGRAM MEMORY

To read a program memory location, the user must write two bytes of the address to the EEADR and EEADRH registers, set the EEPGD control bit (EECON1<7>), and then set control bit RD (EECON1<0>). Once the read control bit is set, the program memory Flash controller will use the second instruction cycle to read the data. This causes the second instruction immediately following the "BSF EECON1,RD" instruction to be ignored. The data is available in the very next cycle, in the EEDAT and EEDATH registers; therefore, it can be read as two bytes in the following instructions.

EEDAT and EEDATH registers will hold this value until another read or until it is written to by the user (during a write operation).

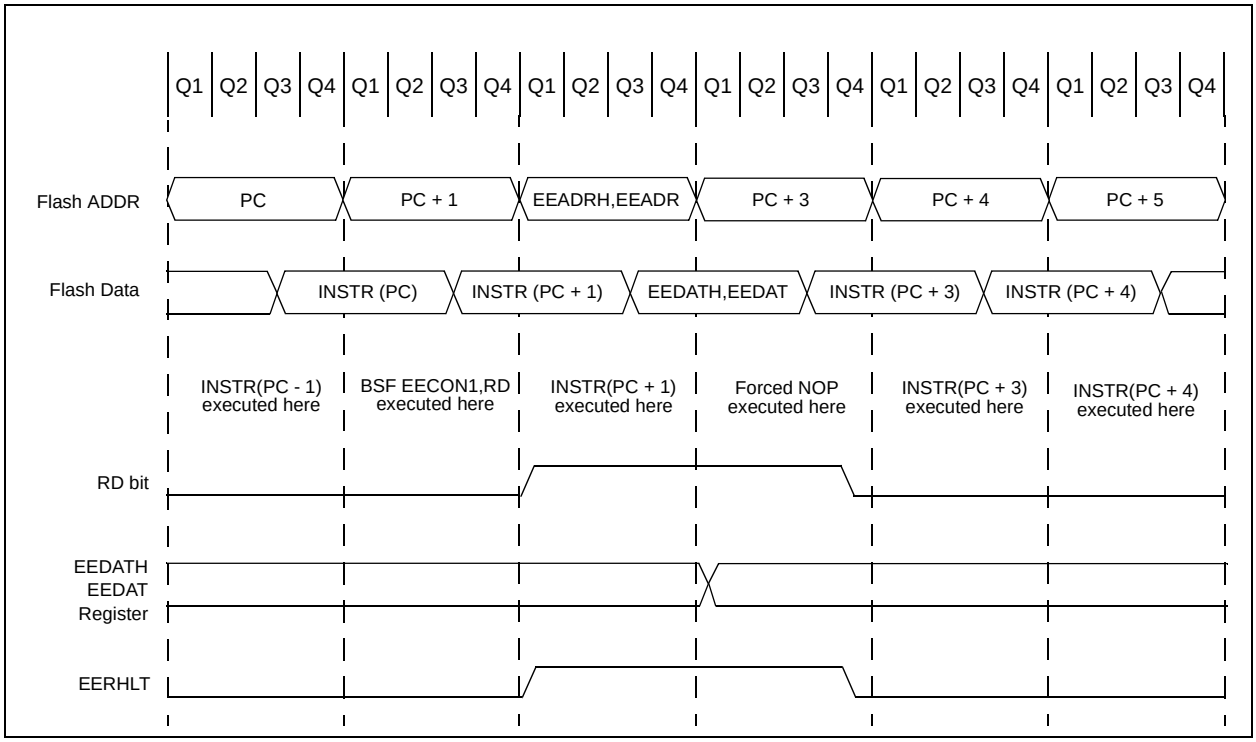
**Note 1:** The two instructions following a program memory read are required to be NOP's. This prevents the user from executing a two-cycle instruction on the next instruction after the RD bit is set.

**2:** If the WR bit is set when EEPGD = 1, it will be immediately reset to '0' and no operation will take place.

**EXAMPLE 10-3: FLASH PROGRAM READ**

```
        BCF     STATUS, RP0             ;Bank 2
        BSF     STATUS, RP1             ;
        MOVLW   MS_PROG_EE_ADDR         ;
        MOVWF   EEADRH                  ;MS Byte of Program Address to read
        MOVLW   LS_PROG_EE_ADDR         ;
        MOVWF   EEADR                   ;LS Byte of Program Address to read
        BSF     STATUS, RP0             ;Bank 3
Required
Sequence
        BSF     EECON1, EEPGD           ;Point to PROGRAM memory
        BSF     EECON1, RD              ;EE Read
                                        ;First instruction after BSF EECON1,RD executes normally
        NOP
        NOP                             ;Any instructions here are ignored as program
                                        ;memory is read in second cycle after BSF EECON1,RD
;
        BCF     STATUS, RP0             ;Bank 2
        MOVF    EEDAT, W                ;W = LS Byte of Program EEDAT
        MOVF    EEDATH, W               ;W = MS Byte of Program EEDAT
        BCF     STATUS, RP1             ;Bank 0
```

**FIGURE 10-1: FLASH PROGRAM MEMORY READ CYCLE EXECUTION**



## 10.2 Write Verify

Depending on the application, good programming practice may dictate that the value written to the data EEPROM should be verified (see Example 10-4) to the desired value to be written.

**EXAMPLE 10-4: WRITE VERIFY**

```
BCF     STATUS, RP0     ;Bank 2
BSF     STATUS, RP1     ;
MOVF    EEDAT, W        ;EEDAT not changed
                        ;from previous write
BSF     STATUS, RP0     ;Bank 3
BSF     EECON1, RD      ;YES, Read the
                        ;value written
BCF     STATUS, RP0     ;Bank 2
XORWF   EEDAT, W        ;
BTFSS   STATUS, Z       ;Is data the same
GOTO    WRITE_ERR       ;No, handle error
:                       ;Yes, continue
BCF     STATUS, RP1     ;Bank 0
```

### 10.2.1 USING THE DATA EEPROM

The data EEPROM is a high-endurance, byte addressable array that has been optimized for the storage of frequently changing information. The maximum endurance for any EEPROM cell is specified as D120 and D120A. D120 or D120A specify a maximum number of writes to any EEPROM location before a refresh is required of infrequently changing memory locations.

### 10.2.2 EEPROM ENDURANCE

A hypothetical data EEPROM is 64 bytes long and has an endurance of 1M writes. It also has a refresh parameter of 10M writes. If every memory location in the cell were written the maximum number of times, the data EEPROM would fail after 64M write cycles. If every memory location save one were written the maximum number of times, the data EEPROM would fail after 63M write cycles, but the one remaining location could fail after 10M cycles. If proper refreshes occurred, then the lone memory location would have to be refreshed six times for the data to remain correct.

## 10.3 Protection Against Spurious Write

There are conditions when the user may not want to write to the data EEPROM memory. To protect against spurious EEPROM writes, various mechanisms have been built in. On power-up, WREN is cleared. Also, the Power-up Timer (64 ms duration) prevents EEPROM write.

The write initiate sequence and the WREN bit together help prevent an accidental write during:

- Brown-out
- Power Glitch
- Software Malfunction

## 10.4 Data EEPROM Operation During Code-Protect

Data memory can be code-protected by programming the CPD bit in the Configuration Word register (Register 14-1) to ‘0’.

When the data memory is code-protected, the CPU is able to read and write data to the data EEPROM. It is recommended to code-protect the program memory when code-protecting data memory. This prevents anyone from programming zeroes over the existing code (which will execute as NOPs) to reach an added routine, programmed in unused program memory, which outputs the contents of data memory. Programming unused locations in program memory to ‘0’ will also help prevent data memory code protection from becoming breached.

**TABLE 10-1: REGISTERS/BITS ASSOCIATED WITH DATA EEPROM**

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0Bh/8Bh/ 10Bh/18Bh | INTCON | GIE | PEIE | T0IE | INTE | RABIE | T0IF | INTF | RABIF | 0000 000x | 0000 000x |
| 0Dh | PIR2 | OSFIF | C2IF | C1IF | EEIF | — | — | — | — | 0000 ---- | 0000 ---- |
| 8Dh | PIE2 | OSFIE | C2IE | C1IE | EEIE | — | — | — | — | 0000 ---- | 0000 ---- |
| 10Eh | EEDATH(1) | — | — | EEDATH5 | EEDATH4 | EEDATH3 | EEDATH2 | EEDATH1 | EEDATH0 | --00 0000 | --00 0000 |
| 10Fh | EEADRH(1) | — | — | — | — | EEADRH3 | EEADRH2 | EEADRH1 | EEADRH0 | ---- 0000 | ---- 0000 |
| 10Ch | EEDAT | EEDAT7 | EEDAT6 | EEDAT5 | EEDAT4 | EEDAT3 | EEDAT2 | EEDAT1 | EEDAT0 | 0000 0000 | 0000 0000 |
| 10Dh | EEADR | EEADR7 | EEADR6 | EEADR5 | EEADR4 | EEADR3 | EEADR2 | EEADR1 | EEADR0 | 0000 0000 | 0000 0000 |
| 18Ch | EECON1 | EEPGD | — | — | — | WRERR | WREN | WR | RD | x--- x000 | 0--- q000 |
| 18Dh | EECON2 | EEPROM Control Register 2 (not a physical register) | | | | | | | | ---- ---- | ---- ---- |

**Legend:** x = unknown, u = unchanged, – = unimplemented read as '0', q = value depends upon condition.
Shaded cells are not used by data EEPROM module.

**Note 1:** PIC16F685/PIC16F689/PIC16F690 only.

## 11.0 ENHANCED CAPTURE/COMPARE/PWM+ (ECCP+) MODULE

The enhanced Capture/Compare/PWM+ (ECCP+) module contains a 16-bit register which can operate as a:

- 16-bit Capture register
- 16-bit Compare register
- PWM Master/Slave Duty Cycle register

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 ECCP+. The special event trigger is generated by a compare match and will clear both TMR1H and TMR1L registers.

**TABLE 11-1: ECCP MODE – TIMER RESOURCES REQUIRED**

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

**REGISTER 11-1: CCP1CON – ENHANCED CCP OPERATION REGISTER(1) (ADDRESS: 17h)**

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| P1M1 | P1M0 | DC1B1 | DC1B0 | CCP1M3 | CCP1M2 | CCP1M1 | CCP1M0 |
| bit 7 | | | | | | | bit 0 |

bit 7-6 **P1M<1:0>:** PWM Output Configuration bits

If CCP1M<3:2> = 00, 01, 10:

xx = P1A assigned as Capture/Compare input; P1B, P1C, P1D assigned as port pins

If CCP1M<3:2> = 11:

00 = Single output; P1A modulated; P1B, P1C, P1D assigned as port pins
01 = Full-bridge output forward; P1D modulated; P1A active; P1B, P1C inactive
10 = Half-bridge output; P1A, P1B modulated with dead band control; P1C, P1D assigned as port pins
11 = Full-bridge output reverse; P1B modulated; P1C active; P1A, P1D inactive

bit 5-4 **DC1B<1:0>:** PWM Duty Cycle 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 **CCP1M<3:0>:** ECCP Mode Select bits

0000 = Capture/Compare/PWM off (resets ECCP module)
0001 = Unused (reserved)
0010 = Compare mode, toggle output on match (CCP1IF bit is set)
0011 = Unused (reserved)
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 TMR1or TMR2, and starts an A/D conversion, if the A/D module is enabled)
1100 = PWM mode; P1A, P1C active-high; P1B, P1D active-high
1101 = PWM mode; P1A, P1C active-high; P1B, P1D active-low
1110 = PWM mode; P1A, P1C active-low; P1B, P1D active-high
1111 = PWM mode; P1A, P1C active-low; P1B, P1D active-low

**Note 1:** PIC16F685/PIC16F690 only.

| **Legend:** | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' | |
| -n = Value at POR | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |

## 11.1 Capture Mode

In Capture mode, CCPR1H:CCPR1L captures the 16-bit value of the TMR1 register when an event occurs on pin RC5/CCP1/P1A. An event is defined as one of the following and is configured by CCP1CON<3:0>:

- Every falling edge
- Every rising edge
- Every 4th rising edge
- Every 16th rising edge

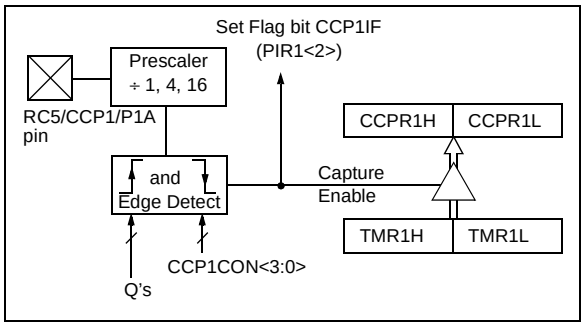
When a capture is made, the interrupt request flag bit, CCP1IF (PIR1<2>), is set. The interrupt flag must be cleared in software. If another capture occurs before the value in register CCPR1 is read, the old captured value is overwritten by the new captured value.

### 11.1.1 CCP1 PIN CONFIGURATION

In Capture mode, the RC5/CCP1/P1A pin should be configured as an input by setting the TRISC<5> bit.

| **Note:** | If the RC5/CCP1/P1A pin is configured as an output, a write to the port can cause a capture condition. |
|---|---|

**FIGURE 11-1: CAPTURE MODE OPERATION BLOCK DIAGRAM**




### 11.1.2 TIMER1 MODE SELECTION

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

### 11.1.3 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 (PIR1<2>) following any such change in operating mode.

### 11.1.4 ECCP PRESCALER

There are four prescaler settings specified by bits CCP1M<3:0> (CCP1CON<3:0>). Whenever the ECCP module is turned off, or the ECCP module is not in Capture mode, the prescaler counter is cleared. 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 11-1 shows the recommended method for switching between capture prescalers. This example also clears the prescaler counter and will not generate the "false" interrupt.

**EXAMPLE 11-1: CHANGING BETWEEN CAPTURE PRESCALERS**

```
BCF   STATUS, RP0  ;Bank 0
BCF   STATUS, RP1  ;
CLRF  CCP1CON      ;Turn ECCP module off
MOVLW NEW_CAPT_PS  ;Load the W reg with
                   ;the new prescaler
                   ;move value and ECCP ON
MOVWF CCP1CON      ;Load CCP1CON with this
                   ;value
```

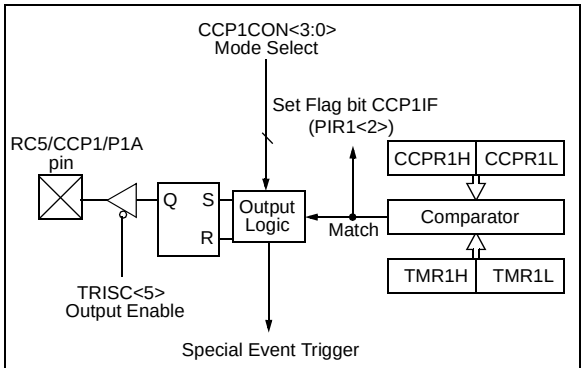
## 11.2 Compare Mode

In Compare mode, the 16-bit CCPR1 register value is constantly compared against the TMR1 register pair value. When a match occurs, the RC5/CCP1/P1A pin is:

- Driven high
- Driven low
- Remains unchanged

The action on the pin is based on the value of control bits, CCP1M<3:0> (CCP1CON<3:0>). At the same time, interrupt flag bit, CCP1IF (PIR1<2>), is set.

**FIGURE 11-2: COMPARE MODE OPERATION BLOCK DIAGRAM**



### 11.2.1 CCP1 PIN CONFIGURATION

The user must configure the RC5/CCP1/P1A pin as an output by clearing the TRISC<5> bit.

| | |
|---|---|
| **Note:** | Clearing the CCP1CON register will force the RC5/CCP1/P1A compare output latch to the default low level. This is not the PORTC I/O data latch. |

### 11.2.2 TIMER1 MODE SELECTION

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

### 11.2.3 SOFTWARE INTERRUPT MODE

When Generate Software Interrupt mode is chosen (CCP1M<3:0> = 1010), the CCP1 pin is not affected. The CCP1IF (PIR1<2>) bit is set, causing a ECCP interrupt (if enabled). See Register 11-1.

### 11.2.4 SPECIAL EVENT TRIGGER

In this mode (CCP1M<3:0> = 1011), an internal hardware trigger is generated, which may be used to initiate an action. See Register 11-1.

The special event trigger output of the CCP occurs immediately upon a match between the TMR1H, TMR1L register pair and CCPR1H, CCPR1L register pair. The TMR1H, TMR1L register pair is not reset until the next rising edge of the TMR1 clock. This allows the CCPR1H, CCPR1L register pair to effectively provide a 16-bit programmable period register for Timer1. The special event trigger output also starts an A/D conversion provided that the A/D module is enabled.

| | |
|---|---|
| **Note 1:** | The special event trigger from the CCP module will not set interrupt flag bit TMR1IF (PIR1<0>). |
| **2:** | Removing the match condition by changing the contents of the CCPR1H and CCPR1L register pair between the clock edge that generates the special event trigger and the clock edge that generates the TMR1 Reset, will preclude the Reset from occurring. |

**TABLE 11-2: REGISTERS ASSOCIATED WITH CAPTURE, COMPARE AND TIMER1(1)**

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0Bh/8Bh/ 10Bh/18Bh | INTCON | GIE | PEIE | T0IE | INTE | RABIE | T0IF | INTF | RABIF | 0000 000x | 0000 000x |
| 0Ch | PIR1 | — | ADIF | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | -000 0000 | -000 0000 |
| 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 | T1GINV | TMR1GE | T1CKPS1 | T1CKPS0 | T1OSCEN | T1SYNC | TMR1CS | TMR1ON | 0000 0000 | uuuu uuuu |
| 11Bh | CM2CON1 | MC1OUT | MC2OUT | — | — | — | — | T1GSS | C2SYNC | 00-- --10 | 00-- --10 |
| 15h | CCPR1L | Capture/Compare/PWM Register 1 Low Byte | | | | | | | | xxxx xxxx | uuuu uuuu |
| 16h | CCPR1H | Capture/Compare/PWM Register 1 High Byte | | | | | | | | xxxx xxxx | uuuu uuuu |
| 17h | CCP1CON | P1M1 | P1M0 | DC1B1 | DC1B0 | CCP1M3 | CCP1M2 | CCP1M1 | CCP1M0 | 0000 0000 | 0000 0000 |
| 87h/187h | TRISC | TRISC7 | TRISC6 | TRISC5 | TRISC4 | TRISC3 | TRISC2 | TRISC1 | TRISC0 | 1111 1111 | 1111 1111 |
| 8Ch | PIE1 | — | ADIE | RCIE | TXIE | SSPIE | CCP1IE | TMR2IE | TMR1IE | -000 0000 | -000 0000 |

**Legend:** – = Unimplemented locations, read as '0', u = unchanged, x = unknown. Shaded cells are not used by the Capture, Compare or Timer1 module.

**Note 1:** PIC16F685/PIC16F690 only.

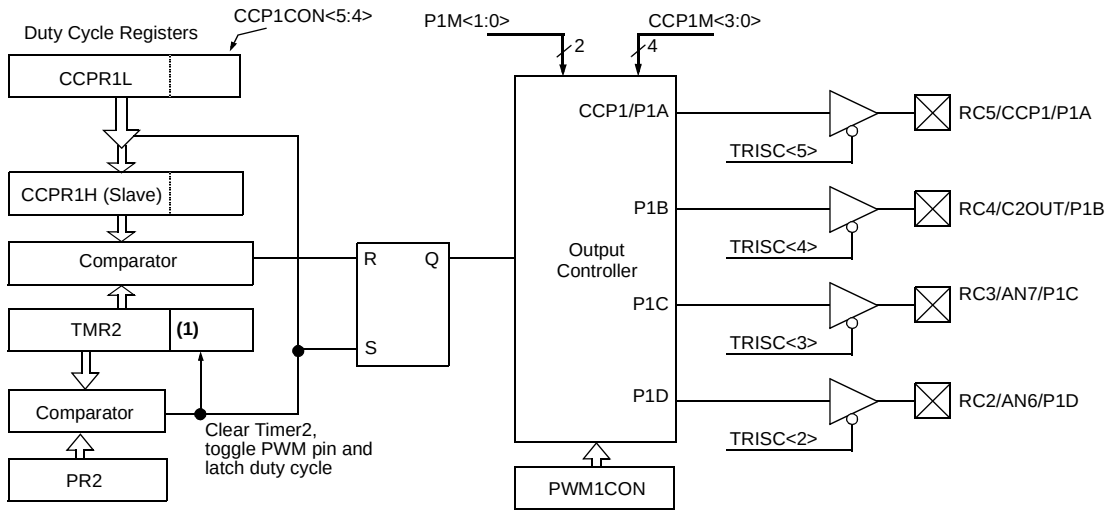
## 11.3 Enhanced PWM Mode

The Enhanced CCP module produces up to a 10-bit resolution PWM output and may have up to four outputs, depending on the selected operating mode. These outputs, designated P1A through P1D, are multiplexed with I/O pins on PORTC. The pin assignments are summarized in Table 11-3.

Figure 11-3 shows a simplified block diagram of PWM operation.

To configure I/O pins as PWM outputs, the proper PWM mode must be selected by setting the P1M<1:0> and CCP1M<3:0> bits (CCP1CON<7:6> and CCP1CON <3:0>, respectively). The appropriate TRISC bits must also be set as outputs.

**FIGURE 11-3: SIMPLIFIED BLOCK DIAGRAM OF THE ENHANCED PWM MODULE**




**Note 1:** The 8-bit timer TMR2 register is concatenated with the 2-bit internal Q clock, or 2 bits of the prescaler to create the 10-bit time base.

### 11.3.1 PWM OUTPUT CONFIGURATIONS

The P1M<1:0> bits in the CCP1CON register allows one of four configurations:

- Single Output
- Half-bridge Output
- Full-bridge Output, Forward mode
- Full-bridge Output, Reverse mode

The general relationship of the outputs in all configurations is summarized in Figure 11-3.

| **Note:** | Clearing the CCP1CON register will force the PWM output latches to their default inactive levels. This is not the PORTC I/O data latch. |
|---|---|

**TABLE 11-3: PIN ASSIGNMENTS FOR VARIOUS ENHANCED CCP MODES**

| ECCP Mode | CCP1CON Configuration | RC5 | RC4 | RC3 | RC2 |
|---|---|---|---|---|---|
| Compatible CCP | 00xx11xx | CCP1 | RC4/C2OUT | RC3/AN7 | RC2/AN6 |
| Dual PWM | 10xx11xx | P1A | P1B | RC3/AN7 | RC2/AN6 |
| Quad PWM | x1xx11xx | P1A | P1B | P1C | P1D |

**Legend:** x = Don't care. Shaded cells indicate pin assignments not used by ECCP in a given mode.

**Note 1:** TRIS register values must be configured appropriately.

**2:** With ECCP in Dual or Quad PWM mode, the C2OUT output control of PORTC must be disabled.

### 11.3.2 PWM PERIOD

A PWM output (Figure 11-4 and Figure 11-5) has a time base (period) and a time that the output is active (duty cycle). The PWM period is specified by writing to the PR2 register. The PWM period can be calculated using the following formula:

**EQUATION 11-1: PWM PERIOD (TIME BASE)**

$$PWM\ period = [(PR2) + 1] \bullet 4 \bullet T_{OSC} \bullet (TMR2\ prescale\ value)$$

PWM frequency is defined as 1 / [PWM period].

When TMR2 is equal to PR2, the following three events occur on the next increment cycle:

- TMR2 is cleared
- The appropriate PWM pin toggles. In Dual PWM mode, this occurs after the dead band delay expires (exception: if PWM duty cycle = 0%, the pin will not be set)
- The PWM duty cycle is latched from CCPR1L into CCPR1H

**Note:** The Timer2 postscaler (see **Section 7.1 "Timer2 Operation"**) is not used in the determination of the PWM frequency. The postscaler could be used to have a servo update rate at a different frequency than the PWM output.

### 11.3.3 PWM DUTY CYCLE

The PWM duty cycle is specified by writing to the CCPR1L register and to the DC1B<1:0> (CCP1CON<5:4>) bits. Up to 10 bits of resolution is available. The CCPR1L contains the eight MSbs and the DC1B<1:0> contains the two LSbs. CCPR1L and DC1B<1:0> can be written to at any time. In PWM mode, CCPR1H is a read-only register. This 10-bit value is represented by CCPR1L (CCP1CON<5:4>).

The following equation is used to calculate the PWM duty cycle in time:

**EQUATION 11-2: PWM DUTY CYCLE TIME**

$$PWM\ duty\ cycle = (CCPR1L{:}CCP1CON\langle 5{:}4\rangle) \bullet T_{OSC} \bullet (TMR2\ prescale\ value)$$

When the CCPR1H and 2-bit latch match TMR2, concatenated with an internal 2-bit Q clock or 2 bits of the TMR2 prescaler, the appropriate PWM pin is toggled. In Dual PWM mode, the pin will be toggled after the dead band time has expired.

The polarity (active-high or active-low) and mode of the signal are configured by the P1M<1:0> (CCP1CON<7:6>) and CCP1M<3:0> (CCP1CON<3:0>) bits.

The maximum PWM resolution for a given PWM frequency is given by the formula:

**EQUATION 11-3: MAX. PWM RESOLUTION PER FREQUENCY**

$$Resolution = \frac{\log\left(\frac{F_{OSC}}{F_{PWM} \bullet TMR2\ Prescaler}\right)}{\log(2)}\ bits$$

All control registers are double buffered and are loaded at the beginning of a new PWM cycle (the period boundary when Timer2 resets) in order to prevent glitches on any of the outputs. The exception is the PWM delay register, which is loaded at either the duty cycle boundary or the period boundary (whichever comes first). Because of the buffering, the module waits until the timer resets, instead of starting immediately. This means that enhanced PWM waveforms do not exactly match the standard PWM waveforms, but are instead offset by one full instruction cycle (4 TOSC).

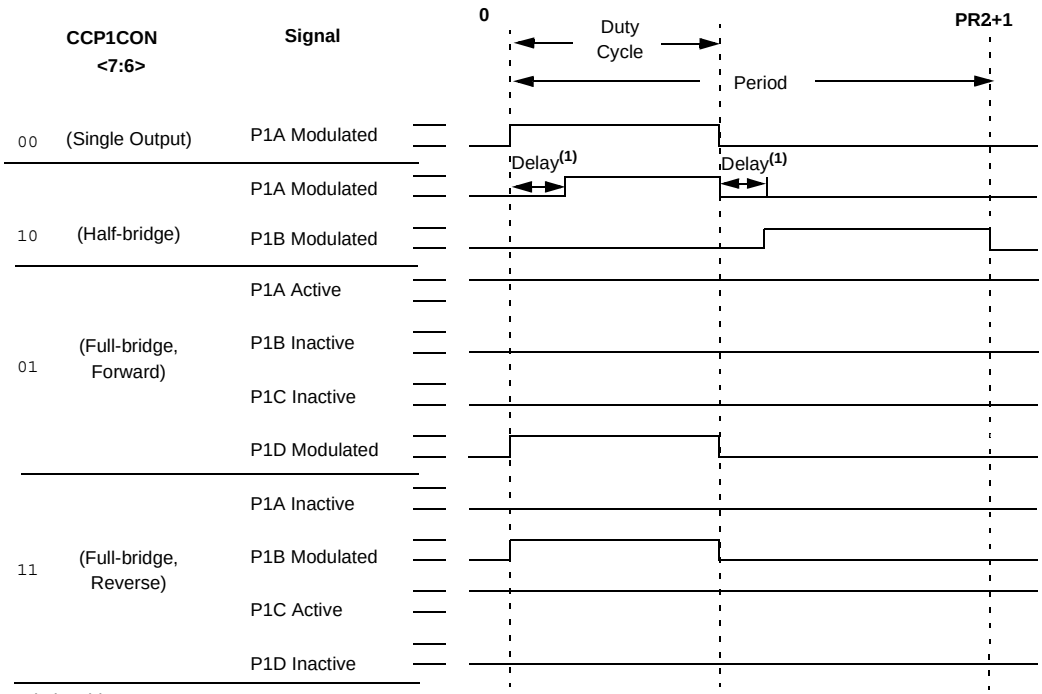
**Note:** If the PWM duty cycle value is longer than the PWM period, the assigned PWM pin(s) will remain unchanged.

**TABLE 11-4: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS (FOSC = 20 MHz)**

| PWM Frequency | 1.22 kHz(1) | 4.88 kHz(1) | 19.53 kHz | 78.12 kHz | 156.3 kHz | 208.3 kHz |
|---|---|---|---|---|---|---|
| Timer Prescale (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 | 6.6 |

**Note 1:** Changing duty cycle will cause a glitch.

**FIGURE 11-4: PWM OUTPUT RELATIONSHIPS (ACTIVE-HIGH STATE)**

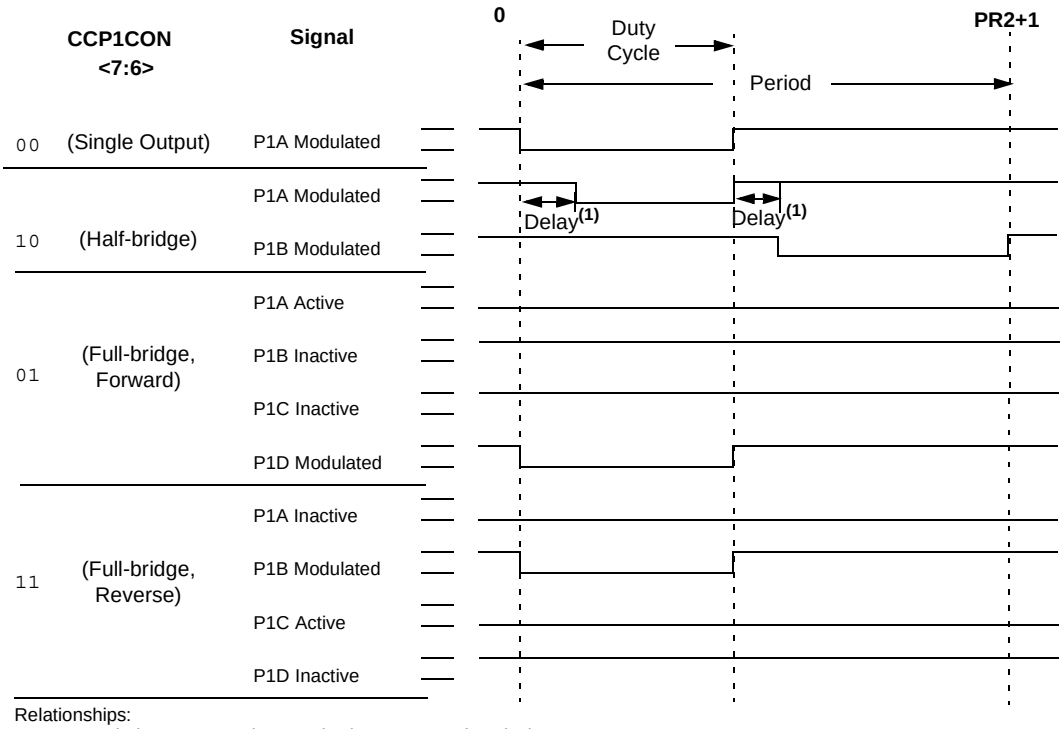



Relationships:

- Period = 4 * Tosc * (PR2 + 1) * (TMR2 prescale value)
- Duty Cycle = Tosc * (CCPR1L<7:0>:CCP1CON<5:4>) * (TMR2 prescale value)
- Delay = 4 * Tosc * (PWM1CON<6:0>)

**Note 1:** Dead band delay is programmed using the PWM1CON register (**Section 11.3.7 "Programmable Dead Band Delay"**).

**FIGURE 11-5: PWM OUTPUT RELATIONSHIPS (ACTIVE-LOW STATE)**




Relationships:

- Period = 4 * Tosc * (PR2 + 1) * (TMR2 prescale value)
- Duty Cycle = Tosc * (CCPR1L<7:0>:CCP1CON<5:4>) * (TMR2 prescale value)
- Delay = 4 * Tosc * (PWM1CON<6:0>)

**Note 1:** Dead band delay is programmed using the PWM1CON register (**Section 11.3.7 "Programmable Dead Band Delay"**).

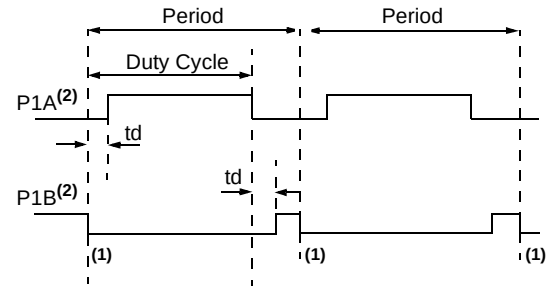
### 11.3.4 HALF-BRIDGE MODE

In the Half-bridge Output mode, two pins are used as outputs to drive push-pull loads. The PWM output signal is output on the RC5/CCP1/P1A pin, while the complementary PWM output signal is output on the RC4/C2OUT/P1B pin (Figure 11-6). This mode can be used for half-bridge applications, as shown in Figure 11-7, or for full-bridge applications, where four power switches are being modulated with two PWM signals.

In Half-bridge Output mode, the programmable dead band delay can be used to prevent shoot-through current in half-bridge power devices. The value of bits PDC<6:0> (PWM1CON<6:0>) sets the number of instruction cycles before the output is driven active. If the value is greater than the duty cycle, the corresponding output remains inactive during the entire cycle. See **Section 11.3.7 "Programmable Dead Band Delay"** for more details of the dead band delay operations.

Since the P1A and P1B outputs are multiplexed with the PORTC<5:4> data latches, the TRISC<5:4> bits must be cleared to configure P1A and P1B as outputs.

**FIGURE 11-6: HALF-BRIDGE PWM OUTPUT**

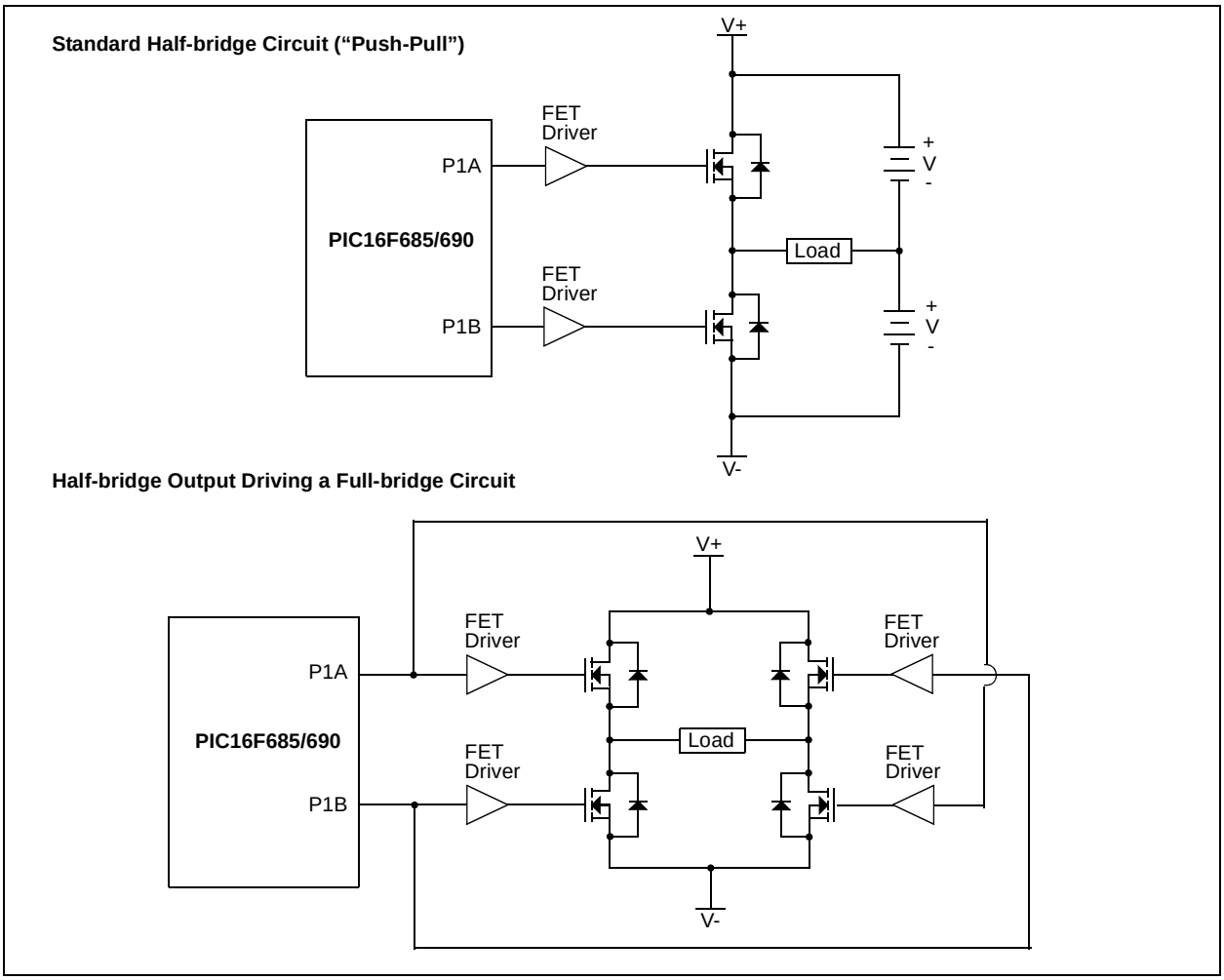



td = Dead Band Delay

**Note 1:** At this time, the TMR2 register is equal to the PR2 register.

**2:** Output signals are shown as active-high.

**FIGURE 11-7: EXAMPLES OF HALF-BRIDGE APPLICATIONS**



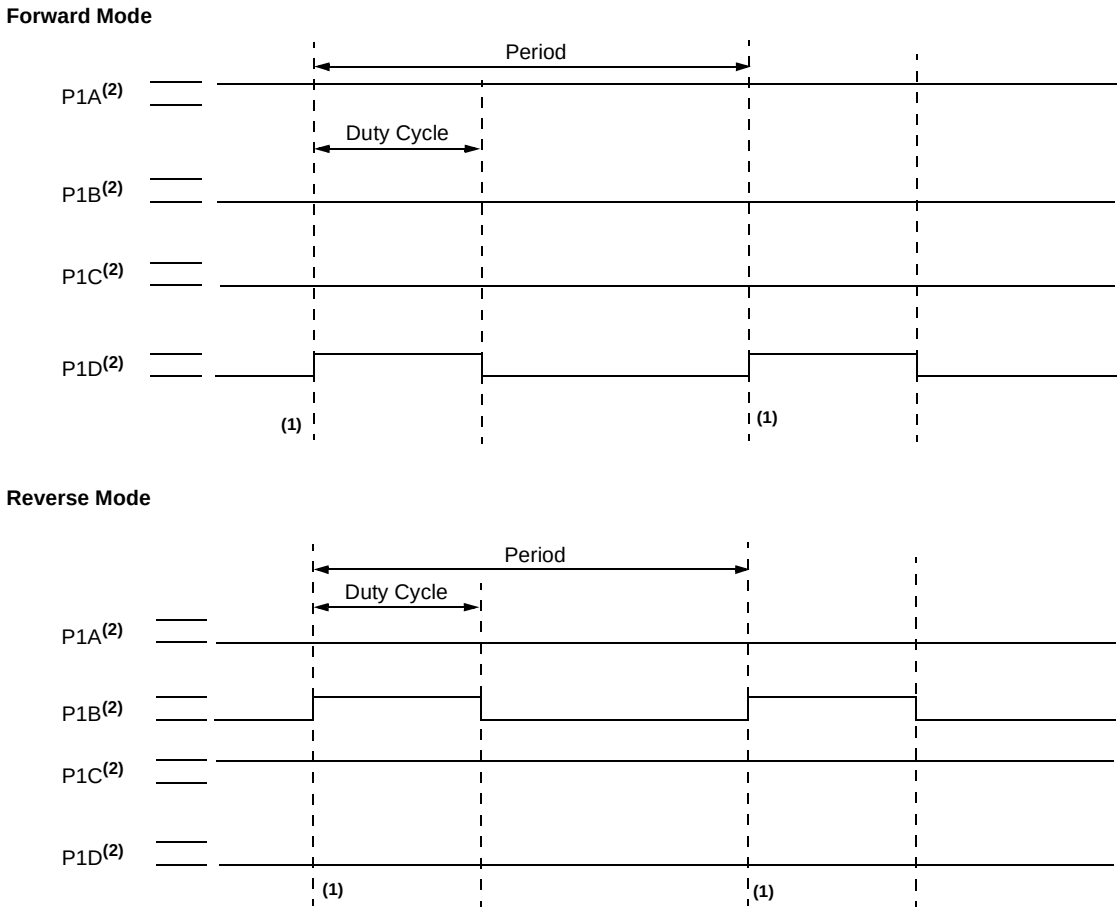
### 11.3.5 FULL-BRIDGE MODE

In Full-bridge Output mode, four pins are used as outputs; however, only two outputs are active at a time. In the Forward mode, pin RC5/CCP1/P1A is continuously active and pin RC2/AN6/P1D is modulated.

In the Reverse mode, RC3/AN7/P1C pin is continuously active and RC4/C2OUT/P1B pin is modulated. These are illustrated in Figure 11-8.

P1A, P1B, P1C and P1D outputs are multiplexed with the PORTC<5:2> data latches. The TRISC<5:2> bits must be cleared to make the P1A, P1B, P1C and P1D pins output.

**FIGURE 11-8: FULL-BRIDGE PWM OUTPUT**

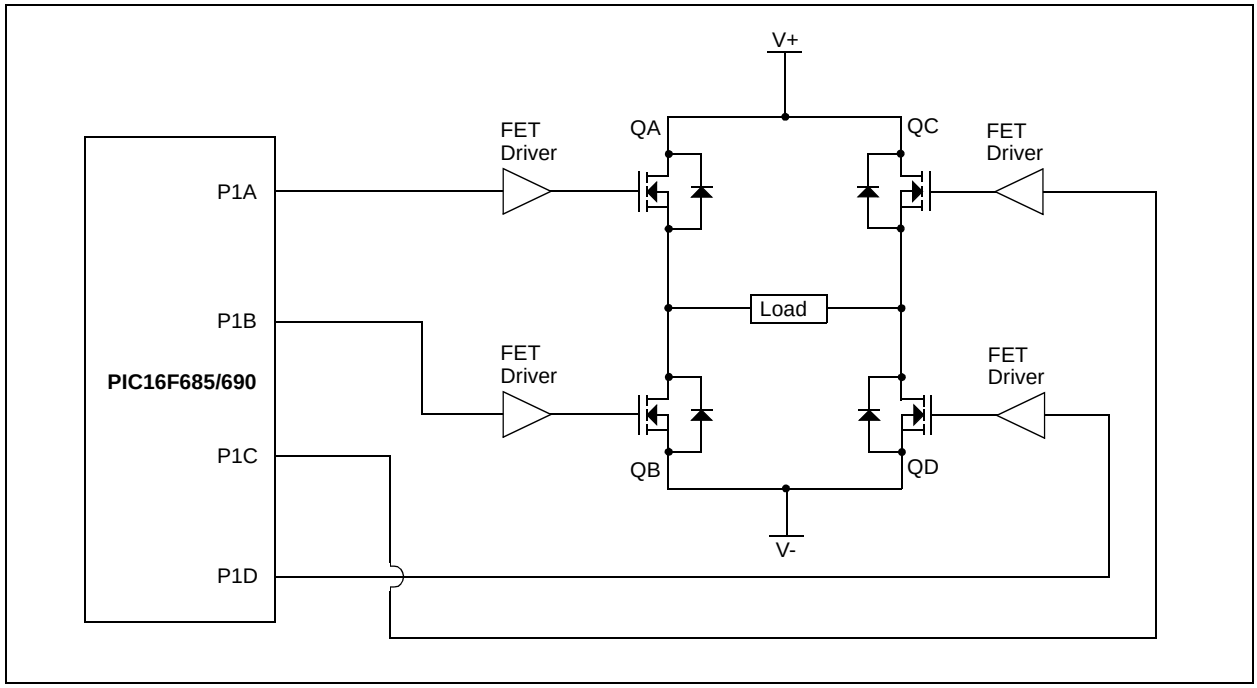



**Note 1:** At this time, the TMR2 register is equal to the PR2 register.

**2:** Output signal is shown as active-high.

**FIGURE 11-9: EXAMPLE OF FULL-BRIDGE APPLICATION**




### 11.3.5.1 Direction Change in Full-Bridge Mode

In the Full-bridge Output mode, the P1M1 bit (CCP1CON<7>) allows user to control the Forward/ Reverse direction. When the application firmware changes this direction control bit, the module will assume the new direction on the next PWM cycle.

Just before the end of the current PWM period, the modulated outputs (P1B and P1D) are placed in their inactive state, while the unmodulated outputs (P1A and P1C) are switched to drive in the opposite direction. This occurs in a time interval of (4 TOSC*(Timer2 Prescale value)) before the next PWM period begins. The Timer2 prescaler will be either 1, 4 or 16, depending on the value of the T2CKPS<1:0> bits (T2CON<1:0>). During the interval from the switch of the unmodulated outputs to the beginning of the next period, the modulated outputs (P1B and P1D) remain inactive. This relationship is shown in Figure 11-10.

Note that in the Full-bridge Output mode, the ECCP+ module does not provide any dead band delay. In general, since only one output is modulated at all times, dead band delay is not required. However, there is a situation where a dead band delay might be required. This situation occurs when both of the following conditions are true:

1. The direction of the PWM output changes when the duty cycle of the output is at or near 100%.
2. The turn off time of the power switch, including the power device and driver circuit, is greater than the turn on time.

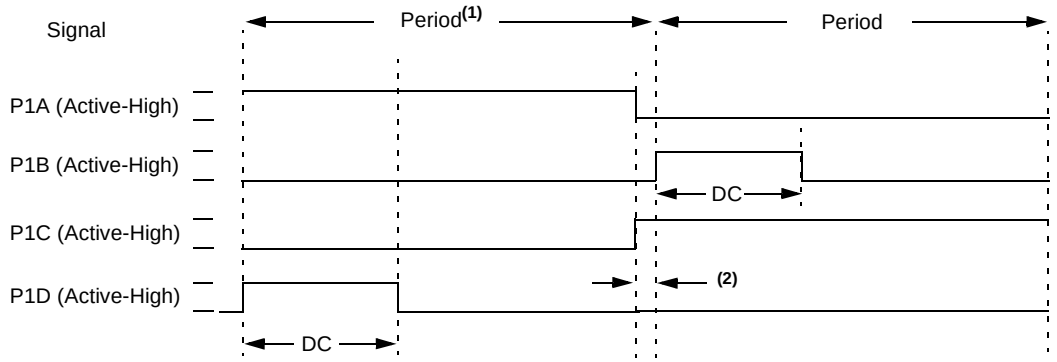
Figure 11-11 shows an example where the PWM direction changes from forward to reverse, at a near 100% duty cycle. At time t1, the output P1A and P1D become inactive, while output P1C becomes active. In this example, since the turn off time of the power devices is longer than the turn on time, a shoot-through current may flow through power devices QC and QD (see Figure 11-9) for the duration of 't'. The same phenomenon will occur to power devices QA and QB for PWM direction change from reverse to forward.

If changing PWM direction at high duty cycle is required for an application, one of the following requirements must be met:

1. Reduce PWM duty cycle for one PWM period before changing directions.
2. Use switch drivers that can drive the switches off faster than they can drive them on.

Other options to prevent shoot-through current may exist.

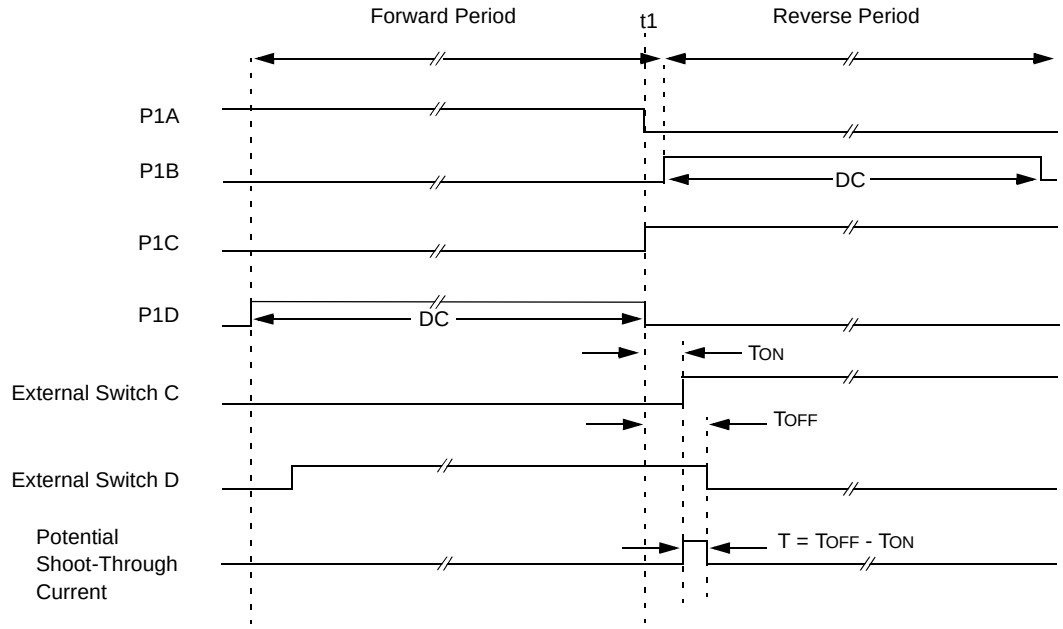
**FIGURE 11-10: PWM DIRECTION CHANGE**




**Note 1:** The direction bit in the ECCP Control register (CCP1CON<7>) is written any time during the PWM cycle.

**2:** When changing directions, the P1A and P1C signals switch before the end of the current PWM cycle at intervals of 4 TOSC, 16 TOSC or 64 TOSC, depending on the Timer2 prescaler value. The modulated P1B and P1D signals are inactive at this time.

**FIGURE 11-11: PWM DIRECTION CHANGE AT NEAR 100% DUTY CYCLE**




**Note 1:** All signals are shown as active-high.

**2:** TON is the turn on delay of power switch QC and its driver.

**3:** TOFF is the turn off delay of power switch QD and its driver.

### 11.3.6 PULSE STEERING MODE

The PWM Steering is available only when the CCP1M<3:2> = 11 and P1M<1:0> = 00 (CCP1CON register). Upon any chip Reset, the PSTRCON register is initialized to enable the PWM output to P1A only.

Once the Single Output mode is selected by CCP1M<3:0>, the user firmware can bring out the same PWM signal to one, two, three or four output pins by setting the appropriate STR<D:A> bits, as shown in Table 11-5.

| **Note:** | The relevant TRIS bits must be set to output ('0') to enable the pin output driver, in order to see the PWM signal on the pin. |
|---|---|

While the PWM Steering mode is active, CCP1M<1:0> selects the PWM output polarity for the P1<D:A> pins. See Register 11-1 (CCP1CON) for details.

The PWM auto-shutdown operation also applies to this PWM Steering mode as described in the **Section 11.3.8 "Enhanced PWM Auto-shutdown"** and **Section 11.3.11 "Effects of a Reset"** and follows ECCPAS values without regard to CCP1M<3:0>. An Auto-Shutdown event will only affect pins that have PWM outputs enabled.

**REGISTER 11-2: PSTRCON – PULSE STEERING CONTROL REGISTER(1, 2) (ADDRESS: 19Dh)**

| U-0 | U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-1 |
|---|---|---|---|---|---|---|---|
| — | — | — | STRSYNC | STRD | STRC | STRB | STRA |
| bit 7 | | | | | | | bit 0 |

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

bit 4 **STRSYNC:** Steering Sync bit
1 = Output steering update occurs on next PWM period
0 = Output steering update occurs at the beginning of the instruction cycle boundary

bit 3 **STRD:** Steering Enable bit D
1 = P1D pin has the PWM waveform with polarity control from CCPM<1:0>
0 = P1D pin is assigned to port pin

bit 2 **STRC:** Steering Enable bit C
1 = P1C pin has the PWM waveform with polarity control from CCPM<1:0>
0 = P1C pin is assigned to port pin

bit 1 **STRB:** Steering Enable bit B
1 = P1B pin has the PWM waveform with polarity control from CCPM<1:0>
0 = P1B pin is assigned to port pin

bit 0 **STRA:** Steering Enable bit A
1 = P1A pin has the PWM waveform with polarity control from CCPM<1:0>
0 = P1A pin is assigned to port pin

**Note 1:** PIC16F685/PIC16F690 only.

**2:** The PWM Steering is available only when the CCP1M<3:2> = 11 and P1M<1:0> = 00 (CCP1CON register).

| **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 |

**TABLE 11-5: PWM STEERING OPERATION WHEN CCP1M<3:2>=11 AND P1M<1:0>=00 (CCP1CON REGISTER)**

| STRD | STRC | STRB | STRA | P1D | P1C | P1B | P1A |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | Port | Port | Port | Port |
| 0 | 0 | 0 | 1 | Port | Port | Port | P1A |
| 0 | 0 | 1 | 0 | Port | Port | P1B | Port |
| 0 | 0 | 1 | 1 | Port | Port | P1B | P1A |
| 0 | 1 | 0 | 0 | Port | P1C | Port | Port |
| 0 | 1 | 0 | 1 | Port | P1C | Port | P1A |
| 0 | 1 | 1 | 0 | Port | P1C | P1B | Port |
| 0 | 1 | 1 | 1 | Port | P1C | P1B | P1A |
| 1 | 0 | 0 | 0 | P1D | Port | Port | Port |
| 1 | 0 | 0 | 1 | P1D | Port | Port | P1A |
| 1 | 0 | 1 | 0 | P1D | Port | P1B | Port |
| 1 | 0 | 1 | 1 | P1D | Port | P1B | P1A |
| 1 | 1 | 0 | 0 | P1D | P1C | Port | Port |
| 1 | 1 | 0 | 1 | P1D | P1C | Port | P1A |
| 1 | 1 | 1 | 0 | P1D | P1C | P1B | Port |
| 1 | 1 | 1 | 1 | P1D | P1C | P1B | P1A |

**Note:** 'Port' as described when NOT in PWM mode.

#### 11.3.6.1 Steering Synchronization

The STRSYNC bit gives the user two selections of when the steering event will happen. When the STRSYNC bit is '0', the steering event will happen at the end of the instruction that writes to the STRCON register. In this case, the output signal at the P1<D:A> pins may be an incomplete PWM waveform. This operation is useful when the user firmware needs to immediately remove a PWM signal from the pin.

When the STRSYNC bit is '1', the effective steering update will happen at the beginning of the next PWM period. In this case, steering on/off the PWM output will always produce a complete PWM waveform.

Figures 11-12 and 11-13 illustrates the timing diagrams of the PWM steering depending on the STRSYNC setting.

**FIGURE 11-12: STEERING EVENT AT END OF INSTRUCTION (STRSYNC = 0)**

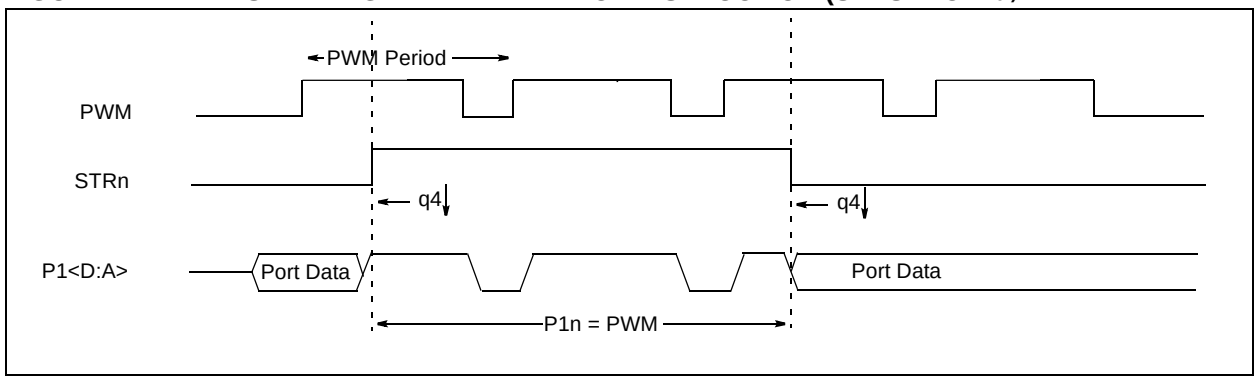
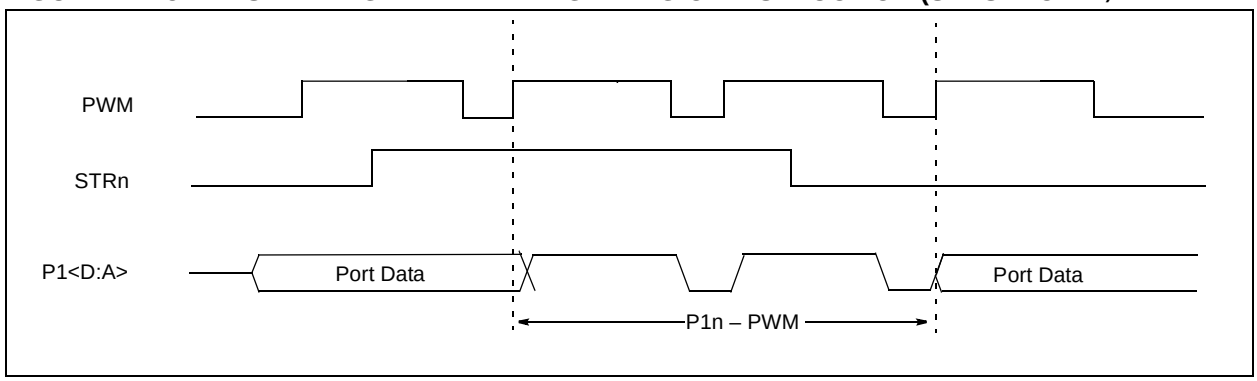



**FIGURE 11-13: STEERING EVENT AT BEGINNING OF INSTRUCTION (STRSYNC = 1)**



### 11.3.7 PROGRAMMABLE DEAD BAND DELAY

In half-bridge applications where all power switches are modulated at the PWM frequency at all times, the power switches normally require more time to turn off than to turn on. If both the upper and lower power switches are switched at the same time (one turned on, and the other turned off), both switches may be on for a short period of time until one switch completely turns off. During this brief interval, a very high current (*shoot-through current*) may flow through both power switches, shorting the bridge supply. To avoid this potentially destructive shoot-through current from flowing during switching, turning on either of the power switches is normally delayed to allow the other switch to completely turn off.

In the Half-bridge Output mode, a digitally programmable dead band delay is available to avoid shoot-through current from destroying the bridge power switches. The delay occurs at the signal transition from the non-active state to the active state. See Figure 11-6 for illustration. The lower seven bits of the PWM1CON register (Register 11-3) sets the delay period in terms of microcontroller instruction cycles (TCY or 4 TOSC).

**REGISTER 11-3: PWM1CON – ENHANCED PWM CONFIGURATION REGISTER(1) (ADDRESS: 1Ch)**

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| PRSEN | PDC6 | PDC5 | PDC4 | PDC3 | PDC2 | PDC1 | PDC0 |
| bit 7 | | | | | | | bit 0 |

bit 7 **PRSEN:** PWM Restart Enable bit

1 = Upon auto-shutdown, the ECCPASE bit clears automatically once the shutdown event goes away; the PWM restarts automatically.

0 = Upon auto-shutdown, ECCPASE must be cleared in software to restart the PWM.

bit 6-0 **PDC<6:0>:** PWM Delay Count bits

PDCn = Number of FOSC/4 (4*TOSC) cycles between the scheduled time when a PWM signal **should** transition active, and the **actual** time it transitions active.

**Note 1:** PIC16F685/PIC16F690 only.

| **Legend:** | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' | |
| - n = Value at POR | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |

### 11.3.8 ENHANCED PWM AUTO-SHUTDOWN

When the ECCP is programmed for any of the enhanced PWM modes, the active output pins may be configured for auto-shutdown. Auto-shutdown immediately places the enhanced PWM output pins into a defined shutdown state when a shutdown event occurs.

A shutdown event can be caused by either of the two comparators or the INT pin (or any combination of these three sources). The comparators may be used to monitor a voltage input proportional to a current being monitored in the bridge circuit. If the voltage exceeds a threshold, the comparator switches state and triggers a shutdown. Alternatively, a digital signal on the INT pin can also trigger a shutdown. The auto-shutdown feature can be disabled by not selecting any auto-shutdown sources. The auto-shutdown sources to be used are selected using the ECCPAS<2:0> bits (ECCPAS<6:4>).

When a shutdown occurs, the output pins are asynchronously placed in their shutdown states, specified by the PSSAC<3:2> and PSSBD<1:0> bits (ECCPAS<3:0>). Each pin pair (P1A/P1C and P1B/P1D) may be set to drive high, drive low, or be tri-stated (not driving). The ECCPASE bit (ECCPAS<7>) is also set to hold the enhanced PWM outputs in their shutdown states.

The ECCPASE bit is set by hardware when a shutdown event occurs. If Auto-restarts are not enabled, the ECCPASE bit is cleared by firmware when the cause of the shutdown clears. If Auto-restarts are enabled, the ECCPASE bit is automatically cleared when the cause of the auto-shutdown has cleared. See **Section 11.3.8.1 "Auto-shutdown and Auto-restart"** for more information.

**REGISTER 11-4: ECCPAS – ENHANCED CAPTURE/COMPARE/PWM+ AUTO-SHUTDOWN CONTROL REGISTER(1) (ADDRESS: 1Dh)**

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| ECCPASE | ECCPAS2 | ECCPAS1 | ECCPAS0 | PSSAC1 | PSSAC0 | PSSBD1 | PSSBD0 |
| bit 7 | | | | | | | bit 0 |

bit 7 **ECCPASE:** ECCP Auto-shutdown Event Status bit
1 = A shutdown event has occurred; ECCP outputs are in shutdown state
0 = ECCP outputs are operating

bit 6-4 **ECCPAS<2:0>:** ECCP Auto-shutdown Source Select bits
000 = Auto-shutdown is disabled
001 = Comparator 1 output change
010 = Comparator 2 output change
011 = Either Comparator 1 or 2 change
100 = VIL on INT pin
101 = VIL on INT pin or Comparator 1 change
110 = VIL on INT pin or Comparator 2 change
111 = VIL on INT pin or Comparator 1 or Comparator 2 change

bit 3-2 **PSSACn:** Pin P1A and P1C Shutdown State Control bits
00 = Drive Pins P1A and P1C to '0'
01 = Drive Pins P1A and P1C to '1'
1x = Pins P1A and P1C tri-state

bit 1-0 **PSSBDn:** Pin P1B and P1D Shutdown State Control bits
00 = Drive Pins P1B and P1D to '0'
01 = Drive Pins P1B and P1D to '1'
1x = Pins P1B and P1D tri-state

**Note 1:** PIC16F685/PIC16F690 only.

| **Legend:** | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' | |
| - n = Value at POR | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |

#### 11.3.8.1 Auto-shutdown and Auto-restart

The auto-shutdown feature can be configured to allow auto-restarts of the module following a shutdown event. This is enabled by setting the PRSEN bit of the PWM1CON register (PWM1CON<7>).

In Shutdown mode with PRSEN = 1 (Figure 11-14), the ECCPASE bit will remain set for as long as the cause of the shutdown continues. When the shutdown condition clears, the ECCPASE bit is cleared. If PRSEN = 0 (Figure 11-15), once a shutdown condition occurs, the ECCPASE bit will remain set until it is cleared by firmware. Once ECCPASE is cleared, the enhanced PWM will resume at the beginning of the next PWM period.

| **Note:** | Writing to the ECCPASE bit is disabled while a shutdown condition is active. |
|---|---|

Independent of the PRSEN bit setting, whether the auto-shutdown source is one of the comparators or INT, the shutdown condition is a level. The ECCPASE bit cannot be cleared as long as the cause of the shutdown persists.

The Auto-shutdown mode can be forced by writing a '1' to the ECCPASE bit.

### 11.3.9 START-UP CONSIDERATIONS

When the ECCP+ module is used in the PWM mode, the application hardware must use the proper external pull-up and/or pull-down resistors on the PWM output pins. When the microcontroller is released from Reset, all of the I/O pins are in the high-impedance state. The external circuits must keep the power switch devices in the OFF state, until the microcontroller drives the I/O pins with the proper signal levels, or activates the PWM output(s).

The CCP1M<1:0> bits (CCP1CON<1:0>) allow the user to choose whether the PWM output signals are active-high or active-low for each pair of PWM output pins (P1A/P1C and P1B/P1D). The PWM output polarities must be selected before the PWM pins are configured as outputs. Changing the polarity configuration while the PWM pins are configured as outputs is not recommended since it may result in damage to the application circuits.

The P1A, P1B, P1C and P1D output latches may not be in the proper states when the PWM module is initialized. Enabling the PWM pins for output at the same time as the ECCP+ module may cause damage to the application circuit. The ECCP+ module must be enabled in the proper Output mode and complete a full PWM cycle before configuring the PWM pins as outputs. The completion of a full PWM cycle is indicated by the TMR2IF bit being set as the second PWM period begins.

**FIGURE 11-14: PWM AUTO-SHUTDOWN (PRSEN = 1, AUTO-RESTART ENABLED)**

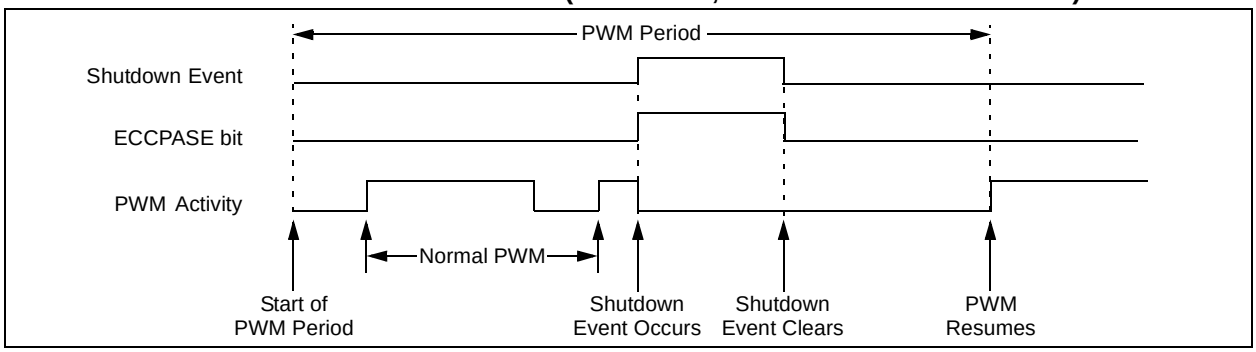
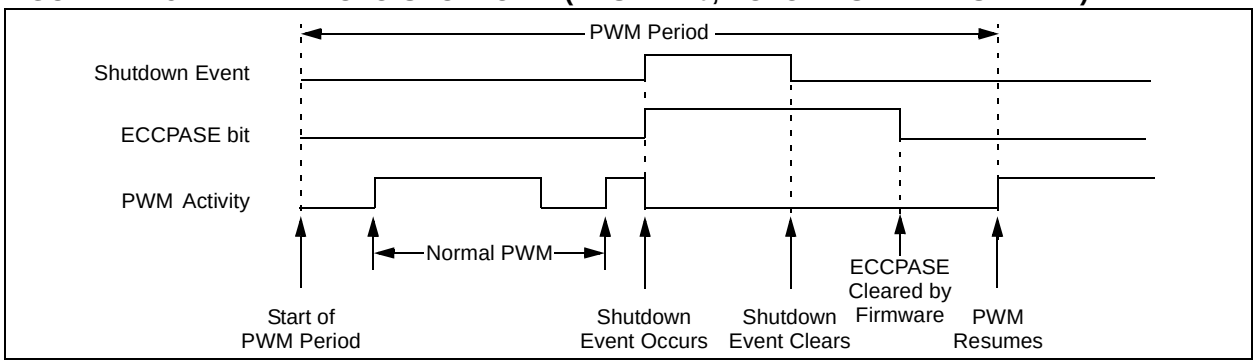



**FIGURE 11-15: PWM AUTO-SHUTDOWN (PRSEN = 0, AUTO-RESTART DISABLED)**



### 11.3.10 OPERATION IN SLEEP MODE

In Sleep mode, all clock sources are disabled. Timer2 will not increment, and the state of the module will not change. If the ECCP pin is driving a value, it will continue to drive that value. When the device wakes up, it will continue from this state.

#### 11.3.10.1 OPERATION WITH FAIL-SAFE CLOCK MONITOR

If the Fail-Safe Clock Monitor is enabled, a clock failure will force the ECCP to be clocked from the internal oscillator clock source, which may have a different clock frequency than the primary clock.

See **Section 3.0 "Clock Sources"** for additional details.

### 11.3.11 EFFECTS OF A RESET

Both Power-on Reset and Resets will force all ports to Input mode and the ECCP registers to their Reset states.

This forces the Enhanced CCP module to reset to a state compatible with the standard CCP module.

### 11.3.12 SETUP FOR PWM OPERATION

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

1. Configure the PWM pins P1A and P1B (and P1C and P1D, if used) as inputs by setting the corresponding TRISC bits.
2. Set the PWM period by loading the PR2 register.
3. Configure the ECCP+ module for the desired PWM mode and configuration by loading the CCP1CON register with the appropriate values:
   - Select one of the available output configurations and direction with the P1M<1:0> bits.
   - Select the polarities of the PWM output signals with the CCP1M<3:0> bits.
4. Set the PWM duty cycle by loading the CCPR1L register and CCP1CON<5:4> bits.
5. For Half-bridge Output mode, set the dead band delay by loading PWM1CON<6:0> with the appropriate value.
6. If auto-shutdown operation is required, load the ECCPAS register:
   - Select the auto-shutdown sources using the ECCPAS<2:0> bits.
   - Select the shutdown states of the PWM output pins using PSSAC<3:2> and PSSBD<1:0> bits.
   - Set the ECCPASE bit (ECCPAS<7>).
   - Configure the comparators using the CM1CON0 and CM2CON0 registers (Registers 8-1 and 8-2).
   - Configure the comparator inputs as analog inputs.
7. If auto-restart operation is required, set the PRSEN bit (PWM1CON<7>).
8. Configure and start TMR2:
   - Clear the TMR2 interrupt flag bit by clearing the TMR2IF bit (PIR1<1>).
   - Set the TMR2 prescale value by loading the T2CKPS bits (T2CON<1:0>).
   - Enable Timer2 by setting the TMR2ON bit (T2CON<2>).
9. Enable PWM outputs after a new PWM cycle has started:
   - Wait until TMR2 overflows (TMR2IF bit is set).
   - Enable the CCP1/P1A, P1B, P1C and/or P1D pin outputs by clearing the respective TRISC bits.
   - Clear the ECCPASE bit (ECCPAS<7>).

**TABLE 11-6: REGISTERS ASSOCIATED WITH PWM AND TIMER2(1)**

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0Bh/8Bh/ 10Bh/18Bh | INTCON | GIE | PEIE | T0IE | INTE | RABIE | T0IF | INTF | RABIF | 0000 000x | 0000 000x |
| 0Ch | PIR1 | — | ADIF | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | -000 0000 | -000 0000 |
| 11h | TMR2 | Timer2 Module Register | | | | | | | | 0000 0000 | 0000 0000 |
| 12h | T2CON | — | TOUTPS3 | TOUTPS2 | TOUTPS1 | TOUTPS0 | TMR2ON | T2CKPS1 | T2CKPS0 | -000 0000 | -000 0000 |
| 15h | CCPR1L | Capture/Compare/PWM Register1 Low Byte | | | | | | | | xxxx xxxx | uuuu uuuu |
| 16h | CCPR1H | Capture/Compare/PWM Register1 High Byte | | | | | | | | xxxx xxxx | uuuu uuuu |
| 17h | CCP1CON | P1M1 | P1M0 | DC1B1 | DC1B0 | CCP1M3 | CCP1M2 | CCP1M1 | CCP1M0 | 0000 0000 | 0000 0000 |
| 1Ch | PWM1CON | PRSEN | PDC6 | PDC5 | PDC4 | PDC3 | PDC2 | PDC1 | PDC0 | 0000 0000 | 0000 0000 |
| 1Dh | ECCPAS | ECCPASE | ECCPAS2 | ECCPAS1 | ECCPAS0 | PSSAC1 | PSSAC0 | PSSBD1 | PSSBD0 | 0000 0000 | 0000 0000 |
| 87h/187h | TRISC | TRISC7 | TRISC6 | TRISC5 | TRISC4 | TRISC3 | TRISC2 | TRISC1 | TRISC0 | 1111 1111 | 1111 1111 |
| 8Ch | PIE1 | — | ADIE | RCIE | TXIE | SSPIE | CCP1IE | TMR2IF | TMR1IF | -000 0000 | -000 0000 |
| 92h | PR2 | Timer2 Module Period Register | | | | | | | | 1111 1111 | 1111 1111 |
| 19Dh | PSTRCON | — | — | — | STRSYNC | STRD | STRC | STRB | STRA | ---0 0001 | ---0 0001 |

**Legend:** – = Unimplemented locations, read as '0', u = unchanged, x = unknown. Shaded cells are not used by the Capture, Compare or Timer1 module.

**Note 1:** PIC16F685/PIC16F690 only.

## 12.0 ENHANCED UNIVERSAL SYNCHRONOUS ASYNCHRONOUS RECEIVER TRANSMITTER (EUSART)

The Enhanced Universal Synchronous Asynchronous Receiver Transmitter (EUSART) module is the serial I/O module available for PIC16F685/687/689/690. (EUSART is also known as a Serial Communications Interface or SCI). The EUSART can be configured in full-duplex Asynchronous mode that can communicate with peripheral devices, such as CRT terminals and personal computers, or it can also be configured as a half-duplex Synchronous mode, which can communicate with peripheral devices, such as A/D or D/A integrated circuits, serial EEPROMs, etc.

The EUSART module implements additional features including automatic baud rate detection and calibration, automatic wake-up on Break reception and 13-bit Break character transmit. These features make the EUSART ideally suited for use in Local Interconnect Network (LIN) bus systems.

The EUSART can be configured in the following modes:

- Asynchronous (full-duplex) with:
  - Auto-wake-up on Break
  - Auto-baud calibration
  - 13-bit Break character transmission
- Synchronous – Master (half-duplex) with selectable clock polarity
- Synchronous – Slave (half-duplex) with selectable clock polarity

In order to configure pins RB6/SCK/SCL and RB7/TX/CK as the Universal Synchronous Asynchronous Receiver Transmitter:

- SPEN (RCSTA<7>) bit must be set (= 1),
- TRISB<6> bit must be set (= 1), and
- TRISB<7> bit must be set (= 1).

| **Note:** | The EUSART control will automatically reconfigure the I/O pin from input to output as needed. |
|---|---|

The operation of the EUSART module is controlled through three registers:

- Transmit Status and Control (TXSTA)
- Receive Status and Control (RCSTA)
- Baud Rate Control (BAUDCTL)
- Baud Rate registers (SPBRGH:SPBRG)

See Registers 12-1, 12-2 and 12-3 for more detail.

### 12.1 Clock Accuracy With Asynchronous Operation

The factory calibrates the internal oscillator block output (INTOSC) for 8 MHz. However, this frequency may drift as $V_{DD}$ or temperature changes, and this directly affects the asynchronous baud rate generator. Two methods may be used to adjust the baud rate clock, but both require a reference clock source of some kind.

The first (preferred) method uses the OSCTUNE register to adjust the INTOSC output back to 8 MHz. Adjusting the value in the OSCTUNE register allows for fine resolution changes to the system clock source (see **Section 3.4 "Internal Clock Modes"** for more information).

The other method adjusts the value in the baud rate generator. There may not be fine enough resolution when adjusting the Baud Rate Generator to compensate for a gradual change in the peripheral clock frequency.

**REGISTER 12-1: TXSTA – TRANSMIT STATUS AND CONTROL REGISTER(1) (ADDRESS: 98h)**

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R-1 | R/W-0 |
|---|---|---|---|---|---|---|---|
| CSRC | TX9 | TXEN | SYNC | SENB | BRGH | TRMT | TX9D |
| bit 7 | | | | | | | bit 0 |

bit 7 **CSRC:** Clock Source Select bit

Asynchronous mode:

Don't care

Synchronous mode:

1 = Master mode (clock generated internally from BRG)
0 = Slave mode (clock from external source)

bit 6 **TX9:** 9-bit Transmit Enable bit

1 = Selects 9-bit transmission
0 = Selects 8-bit transmission

bit 5 **TXEN:** Transmit Enable bit

1 = Transmit enabled
0 = Transmit disabled

**Note:** SREN/CREN overrides TXEN in Sync mode.

bit 4 **SYNC:** EUSART Mode Select bit

1 = Synchronous mode
0 = Asynchronous mode

bit 3 **SENB:** Send Break Character bit

Asynchronous mode:

1 = Send Sync Break on next transmission (cleared by hardware upon completion)
0 = Sync Break transmission completed

Synchronous mode:

Don't care

bit 2 **BRGH:** High Baud Rate Select bit

Asynchronous mode:

1 = High speed
0 = Low speed

Synchronous mode:

Unused in this mode

bit 1 **TRMT:** Transmit Shift Register Status bit

1 = TSR empty
0 = TSR full

bit 0 **TX9D:** 9th bit of Transmit Data

Can be address/data bit or a parity bit.

**Note 1:** PIC16F687/PIC16F689/PIC16F690 only.

| **Legend:** | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' | |
| - n = Value at POR | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |

**REGISTER 12-2: RCSTA – RECEIVE STATUS AND CONTROL REGISTER(1) (ADDRESS: 18h)**

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R-0 | R-0 | R-x |
|---|---|---|---|---|---|---|---|
| SPEN | RX9 | SREN | CREN | ADDEN | FERR | OERR | RX9D |
| bit 7 | | | | | | | bit 0 |

bit 7 **SPEN:** Serial Port Enable bit
1 = Serial port enabled
0 = Serial port disabled (holds module in Reset)

bit 6 **RX9:** 9-bit Receive Enable bit
1 = Selects 9-bit reception
0 = Selects 8-bit reception

bit 5 **SREN:** Single Receive Enable bit
Asynchronous mode:
Don't care
Synchronous mode – Master:
1 = Enables single receive
0 = Disables single receive
This bit is cleared after reception is complete.
Synchronous mode – Slave:
Don't care

bit 4 **CREN:** Continuous Receive Enable bit
Asynchronous mode:
1 = Enables receiver
0 = Disables receiver
Synchronous mode:
1 = Enables continuous receive until enable bit CREN is cleared (CREN overrides SREN)
0 = Disables continuous receive

bit 3 **ADDEN:** Address Detect Enable bit
Asynchronous mode 9-bit (RX9 = 1):
1 = Enables address detection, enable interrupt and load the receive buffer when RSR<8> is set
0 = Disables address detection, all bytes are received and ninth bit can be used as parity bit
Asynchronous mode 8-bit (RX9 = 0):
Don't care

bit 2 **FERR:** Framing Error bit
1 = Framing error (can be updated by reading RCREG register and receive next valid byte)
0 = No framing error

bit 1 **OERR:** Overrun Error bit
1 = Overrun error (can be cleared by clearing bit CREN)
0 = No overrun error

bit 0 **RX9D:** 9th bit of Received Data
This can be address/data bit or a parity bit and must be calculated by user firmware.

**Note 1:** PIC16F687/PIC16F689/PIC16F690 only.

| **Legend:** | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' | |
| - n = Value at POR | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |

**REGISTER 12-3: BAUDCTL – BAUD RATE CONTROL REGISTER(1) (ADDRESS: 9Bh)**

| R-0 | R-1 | U-0 | R/W-0 | R/W-0 | U-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| ABDOVF | RCIDL | — | SCKP | BRG16 | — | WUE | ABDEN |
| bit 7 | | | | | | | bit 0 |

bit 7 **ABDOVF:** Auto-Baud Detect Overflow bit

Asynchronous mode:
1 = Auto-baud timer overflowed
0 = Auto-baud timer did not overflow
Synchronous mode:
Don't care

bit 6 **RCIDL**: Receive IDLE Flag bit

Asynchronous mode:
1 = Receiver is IDLE
0 = Start bit has been received and the receiver is receiving
Synchronous mode:
Don't care

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

bit 4 **SCKP**: Synchronous Clock Polarity Select bit

Asynchronous mode:
1 = Transmit inverted data to the RB7/TX/CK pin
0 = Transmit non-inverted data to the RB7/TX/CK pin
Synchronous mode:
1 = Data is clocked on rising edge of the clock
0 = Data is clocked on falling edge of the clock

bit 3 **BRG16:** 16-bit Baud Rate Generator bit

1 = 16-bit baud rate generator is used
0 = 8-bit baud rate generator is used

bit 2 **Unimplemented:** Read as '0'

bit 1 **WUE:** Wake-up Enable bit

1 = Next falling RX/DT edge will set RCIF and wake-up device if it is asleep (automatically cleared on next rising edge after falling edge)
0 = RX/DT edges do not generate interrupts

bit 0 **ABDEN**: Auto-Baud Detect Enable bit

Asynchronous mode:
1 = Auto-Baud mode is enabled (clears when auto-baud is complete)
0 = Auto-Baud mode is disabled
Synchronous mode:
Don't care

**Note 1:** PIC16F687/PIC16F689/PIC16F690 only.

| **Legend:** | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' | |
| - n = Value at POR | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |

## 12.2 EUSART Baud Rate Generator (BRG)

The BRG is a dedicated 8-bit or 16-bit generator that supports both the Asynchronous and Synchronous modes of the EUSART. By default, the BRG operates in 8-bit mode; setting the BRG16 bit (BAUDCTL<3>) selects 16-bit mode.

The SPBRGH:SPBRG register pair controls the period of a free running timer. In Asynchronous mode, bits BRGH (TXSTA<2>) and BRG16 also control the baud rate. In Synchronous mode, bit BRGH is ignored. Table 12-1 shows the formula for computation of the baud rate for different EUSART modes, which only apply in Synchronous Master mode (internally generated clock).

Given the desired baud rate and FOSC, the nearest integer value for the SPBRGH:SPBRG registers can be calculated using the formulas in Table 12-1. From this, the error in baud rate can be determined. An example calculation is shown in Example 12-1. Typical baud rates and error values for the various asynchronous modes are shown in Table 12-2. It may be advantageous to use the high baud rate (BRGH = 1), or the 16-bit BRG to reduce the baud rate error, or achieve a slow baud rate for a fast oscillator frequency.

Writing a new value to the SPBRGH:SPBRG registers causes the BRG timer to be reset (or cleared). This ensures the BRG does not wait for a timer overflow before outputting the new baud rate.

If the system clock is changed during an active receive operation, a receive error or data loss may result. To avoid this problem, check the status of the RCIDL bit and make sure that the receive operation is IDLE before changing the system clock.

### 12.2.1 SAMPLING

The data on the RB5/AN11/RX/DT pin is sampled three times by a majority detect circuit to determine if a high or a low level is present at the RX pin.

**EXAMPLE 12-1: CALCULATING BAUD RATE ERROR**

For a device with FOSC of 16 MHz, desired baud rate of 9600, Asynchronous mode, 8-bit BRG:

$$\text{Desired Baud Rate} = \frac{FOSC}{64([SPBRGH:SPBRG] + 1)}$$

Solving for SPBRGH:SPBRG:

$$X = \frac{\frac{FOSC}{\text{Desired Baud Rate}}}{64} - 1$$

$$= \frac{\frac{16000000}{9600}}{64} - 1$$

$$= [25.042] = 25$$

$$\text{Calculated Baud Rate} = \frac{16000000}{64(25 + 1)}$$

$$= 9615$$

$$\text{Error} = \frac{\text{Calc. Baud Rate} - \text{Desired Baud Rate}}{\text{Desired Baud Rate}}$$

$$= \frac{(9615 - 9600)}{9600} = 0.16\%$$

**Note:** When BRGH = 1 and BRG16 = 1 then SPBRGH:SPBRG values ≤ 4 are invalid.

**TABLE 12-1: BAUD RATE FORMULAS**

| Configuration Bits | | | BRG/EUSART Mode | Baud Rate Formula |
|---|---|---|---|---|
| SYNC | BRG16 | BRGH | | |
| 0 | 0 | 0 | 8-bit/Asynchronous | FOSC/[64 (n+1)] |
| 0 | 0 | 1 | 8-bit/Asynchronous | FOSC/[16 (n+1)] |
| 0 | 1 | 0 | 16-bit/Asynchronous | |
| 0 | 1 | 1 | 16-bit/Asynchronous | FOSC/[4 (n+1)] |
| 1 | 0 | x | 8-bit/Synchronous Master | |
| 1 | 1 | x | 16-bit/Synchronous Master | |

**Legend:** x = Don't care, n = value of SPBRGH:SPBRG register pair

**TABLE 12-2: REGISTERS ASSOCIATED WITH BAUD RATE GENERATOR(1)**

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 18h | RCSTA | SPEN | RX9 | SREN | CREN | ADDEN | FERR | OERR | RX9D | 0000 000x | 0000 000x |
| 98h | TXSTA | CSRC | TX9 | TXEN | SYNC | SENB | BRGH | TRMT | TX9D | 0000 0010 | 0000 0010 |
| 99h | SPBRG | BRG7 | BRG6 | BRG5 | BRG4 | BRG3 | BRG2 | BRG1 | BRG0 | 0000 0000 | 0000 0000 |
| 9Ah | SPBRGH | BRG15 | BRG14 | BRG13 | BRG12 | BRG11 | BRG10 | BRG9 | BRG8 | 0000 0000 | 0000 0000 |
| 9Bh | BAUDCTL | ABDOVF | RCIDL | — | SCKP | BRG16 | — | WUE | ABDEN | 01-0 0-00 | 01-0 0-00 |

**Legend:** x = unknown, u = unchanged, - = unimplemented locations read as '0'. Shaded cells are not used by oscillators.

**Note 1:** PIC16F687/PIC16F689/PIC16F690 only.

**TABLE 12-3: BAUD RATES FOR ASYNCHRONOUS MODES**

| BAUD RATE (K) | SYNC = 0, BRGH = 0, BRG16 = 0 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | FOSC = 20.000 MHz | | | FOSC = 10.000 MHz | | | FOSC = 8.000 MHz | | |
| | Actual Rate (K) | % Error | SPBRG value (decimal) | Actual Rate (K) | % Error | SPBRG value (decimal) | Actual Rate (K) | % Error | SPBRG value (decimal) |
| 0.3 | — | — | — | — | — | — | — | — | — |
| 1.2 | 1.221 | 1.73 | 255 | 1.202 | 0.16 | 129 | 1201 | -0.16 | 103 |
| 2.4 | 2.404 | 0.16 | 129 | 2.404 | 0.16 | 64 | 2403 | -0.16 | 51 |
| 9.6 | 9.766 | 1.73 | 31 | 9.766 | 1.73 | 15 | 9615 | -0.16 | 12 |
| 19.2 | 19.531 | 1.73 | 15 | 19.531 | 1.73 | 7 | — | — | — |
| 57.6 | 62.500 | 8.51 | 4 | 52.083 | -9.58 | 2 | — | — | — |
| 115.2 | 104.167 | -9.58 | 2 | 78.125 | -32.18 | 1 | — | — | — |

| BAUD RATE (K) | SYNC = 0, BRGH = 0, BRG16 = 0 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | FOSC = 4.000 MHz | | | FOSC = 2.000 MHz | | | FOSC = 1.000 MHz | | |
| | Actual Rate (K) | % Error | SPBRG value (decimal) | Actual Rate (K) | % Error | SPBRG value (decimal) | Actual Rate (K) | % Error | SPBRG value (decimal) |
| 0.3 | 0.300 | 0.16 | 207 | 300 | -0.16 | 103 | 300 | -0.16 | 51 |
| 1.2 | 1.202 | 0.16 | 51 | 1201 | -0.16 | 25 | 1201 | -0.16 | 12 |
| 2.4 | 2.404 | 0.16 | 25 | 2403 | -0.16 | 12 | — | — | — |
| 9.6 | 8.929 | -6.99 | 6 | — | — | — | — | — | — |
| 19.2 | 20.833 | 8.51 | 2 | — | — | — | — | — | — |
| 57.6 | 62.500 | 8.51 | 0 | — | — | — | — | — | — |
| 115.2 | 62.500 | -45.75 | 0 | — | — | — | — | — | — |

| BAUD RATE (K) | SYNC = 0, BRGH = 1, BRG16 = 0 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | FOSC = 20.000 MHz | | | FOSC = 10.000 MHz | | | FOSC = 8.000 MHz | | |
| | Actual Rate (K) | % Error | SPBRG value (decimal) | Actual Rate (K) | % Error | SPBRG value (decimal) | Actual Rate (K) | % Error | SPBRG value (decimal) |
| 2.4 | — | — | — | 2.441 | 1.73 | 255 | 2403 | -0.16 | 207 |
| 9.6 | 9.615 | 0.16 | 129 | 9.615 | 0.16 | 64 | 9615 | -0.16 | 51 |
| 19.2 | 19.231 | 0.16 | 64 | 19.531 | 1.73 | 31 | 19230 | -0.16 | 25 |
| 57.6 | 56.818 | -1.36 | 21 | 56.818 | -1.36 | 10 | 55555 | 3.55 | 8 |
| 115.2 | 113.636 | -1.36 | 10 | 125.000 | 8.51 | 4 | — | — | — |

| BAUD RATE (K) | SYNC = 0, BRGH = 1, BRG16 = 0 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | FOSC = 4.000 MHz | | | FOSC = 2.000 MHz | | | FOSC = 1.000 MHz | | |
| | Actual Rate (K) | % Error | SPBRG value (decimal) | Actual Rate (K) | % Error | SPBRG value (decimal) | Actual Rate (K) | % Error | SPBRG value (decimal) |
| 0.3 | — | — | — | — | — | — | 300 | -0.16 | 207 |
| 1.2 | 1.202 | 0.16 | 207 | 1201 | -0.16 | 103 | 1201 | -0.16 | 51 |
| 2.4 | 2.404 | 0.16 | 103 | 2403 | -0.16 | 51 | 2403 | -0.16 | 25 |
| 9.6 | 9.615 | 0.16 | 25 | 9615 | -0.16 | 12 | — | — | — |
| 19.2 | 19.231 | 0.16 | 12 | — | — | — | — | — | — |
| 57.6 | 62.500 | 8.51 | 3 | — | — | — | — | — | — |
| 115.2 | 125.000 | 8.51 | 1 | — | — | — | — | — | — |

**TABLE 12-3: BAUD RATES FOR ASYNCHRONOUS MODES (CONTINUED)**

| BAUD RATE (K) | SYNC = 0, BRGH = 0, BRG16 = 1 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | FOSC = 20.000 MHz | | | FOSC = 10.000 MHz | | | FOSC = 8.000 MHz | | |
| | Actual Rate (K) | % Error | SPBRG value (decimal) | Actual Rate (K) | % Error | SPBRG value (decimal) | Actual Rate (K) | % Error | SPBRG value (decimal) |
| 0.3 | 0.300 | 0.02 | 4165 | 0.300 | 0.02 | 2082 | 300 | -0.04 | 1665 |
| 1.2 | 1.200 | -0.03 | 1041 | 1.200 | -0.03 | 520 | 1201 | -0.16 | 415 |
| 2.4 | 2.399 | -0.03 | 520 | 2.404 | 0.16 | 259 | 2403 | -0.16 | 207 |
| 9.6 | 9.615 | 0.16 | 129 | 9.615 | 0.16 | 64 | 9615 | -0.16 | 51 |
| 19.2 | 19.231 | 0.16 | 64 | 19.531 | 1.73 | 31 | 19230 | -0.16 | 25 |
| 57.6 | 56.818 | -1.36 | 21 | 56.818 | -1.36 | 10 | 55555 | 3.55 | 8 |
| 115.2 | 113.636 | -1.36 | 10 | 125.000 | 8.51 | 4 | — | — | — |

| BAUD RATE (K) | SYNC = 0, BRGH = 0, BRG16 = 1 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | FOSC = 4.000 MHz | | | FOSC = 2.000 MHz | | | FOSC = 1.000 MHz | | |
| | Actual Rate (K) | % Error | SPBRG value (decimal) | Actual Rate (K) | % Error | SPBRG value (decimal) | Actual Rate (K) | % Error | SPBRG value (decimal) |
| 0.3 | 0.300 | 0.04 | 832 | 300 | -0.16 | 415 | 300 | -0.16 | 207 |
| 1.2 | 1.202 | 0.16 | 207 | 1201 | -0.16 | 103 | 1201 | -0.16 | 51 |
| 2.4 | 2.404 | 0.16 | 103 | 2403 | -0.16 | 51 | 2403 | -0.16 | 25 |
| 9.6 | 9.615 | 0.16 | 25 | 9615 | -0.16 | 12 | — | — | — |
| 19.2 | 19.231 | 0.16 | 12 | — | — | — | — | — | — |
| 57.6 | 62.500 | 8.51 | 3 | — | — | — | — | — | — |
| 115.2 | 125.000 | 8.51 | 1 | — | — | — | — | — | — |

| BAUD RATE (K) | SYNC = 0, BRGH = 1, BRG16 = 1 or SYNC = 1, BRG16 = 1 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | FOSC = 20.000 MHz | | | FOSC = 10.000 MHz | | | FOSC = 8.000 MHz | | |
| | Actual Rate (K) | % Error | SPBRG value (decimal) | Actual Rate (K) | % Error | SPBRG value (decimal) | Actual Rate (K) | % Error | SPBRG value (decimal) |
| 0.3 | 0.300 | 0.00 | 16665 | 0.300 | 0.00 | 8332 | 300 | -0.01 | 6665 |
| 1.2 | 1.200 | 0.02 | 4165 | 1.200 | 0.02 | 2082 | 1200 | -0.04 | 1665 |
| 2.4 | 2.400 | 0.02 | 2082 | 2.402 | 0.06 | 1040 | 2400 | -0.04 | 832 |
| 9.6 | 9.596 | -0.03 | 520 | 9.615 | 0.16 | 259 | 9615 | -0.16 | 207 |
| 19.2 | 19.231 | 0.16 | 259 | 19.231 | 0.16 | 129 | 19230 | -0.16 | 103 |
| 57.6 | 57.471 | -0.22 | 86 | 58.140 | 0.94 | 42 | 57142 | 0.79 | 34 |
| 115.2 | 116.279 | 0.94 | 42 | 113.636 | -1.36 | 21 | 117647 | -2.12 | 16 |

| BAUD RATE (K) | SYNC = 0, BRGH = 1, BRG16 = 1 or SYNC = 1, BRG16 = 1 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | FOSC = 4.000 MHz | | | FOSC = 2.000 MHz | | | FOSC = 1.000 MHz | | |
| | Actual Rate (K) | % Error | SPBRG value (decimal) | Actual Rate (K) | % Error | SPBRG value (decimal) | Actual Rate (K) | % Error | SPBRG value (decimal) |
| 0.3 | 0.300 | 0.01 | 3332 | 300 | -0.04 | 1665 | 300 | -0.04 | 832 |
| 1.2 | 1.200 | 0.04 | 832 | 1201 | -0.16 | 415 | 1201 | -0.16 | 207 |
| 2.4 | 2.404 | 0.16 | 415 | 2403 | -0.16 | 207 | 2403 | -0.16 | 103 |
| 9.6 | 9.615 | 0.16 | 103 | 9615 | -0.16 | 51 | 9615 | -0.16 | 25 |
| 19.2 | 19.231 | 0.16 | 51 | 19230 | -0.16 | 25 | 19230 | -0.16 | 12 |
| 57.6 | 58.824 | 2.12 | 16 | 55555 | 3.55 | 8 | — | — | — |
| 115.2 | 111.111 | -3.55 | 8 | — | — | — | — | — | — |

### 12.2.2 AUTO-BAUD DETECT

The EUSART module supports the automatic detection and calibration of baud rate. This feature is active only in Asynchronous mode and while the WUE bit is clear.

The automatic baud rate measurement sequence (Figure 12-1) begins whenever a Start bit is received and the ABDEN bit is set. The calculation is self-averaging.

In the Auto-Baud Detect (ABD) mode, the clock to the BRG is reversed. Rather than the BRG clocking the incoming RX signal, the RX signal is timing the BRG. In ABD mode, the internal baud rate generator is used as a counter to time the bit period of the incoming serial byte stream.

If the BRG counter rolls over, the ABDOVF (BAUDCTL<7>) and the RCIF bits are set to indicate BRG has overflowed. The ABDOVF bit is set by hardware and can only be cleared by the user. When an overflow occurs, Auto-baud Detect remains active and the ABDEN (BAUDCTL<0>) bit remains set. The ABDOVF will remain set and not able to be cleared until the ABDEN is reset to '0'. The RCIF must be cleared by reading the RCREG or clearing the SPEN bit.

Once the ABDEN bit is set, the state machine will clear the BRG and look for a Start bit. The Auto-Baud Detect must receive a byte with the value 55h (ASCII "U", which is also the LIN bus Sync character), in order to calculate the proper bit rate. The measurement is taken over both a low and a high bit time in order to minimize any effects caused by asymmetry of the incoming signal. After a Start bit, the SPBRG begins counting up using the preselected clock source on the first rising edge of RX. After eight bits on the RX pin, or the fifth rising edge, an accumulated value totalling the proper BRG period is left in the SPBRGH:SPBRG registers. Once the 5th edge is seen (should correspond to the Stop bit), the ABDEN bit is automatically cleared.

While calibrating the baud rate period, the BRG registers are clocked at 1/8th the pre-configured clock rate. Note that the BRG clock will be configured by the BRG16 and BRGH bits. Independent of the BRG16 bit setting, both the SPBRG and SPBRGH will be used as a 16-bit counter. This allows the user to verify that no carry occurred for 8-bit modes, by checking for 00h in the SPBRGH register. Refer to Table 12-4 for counter clock rates to the BRG.

While the ABD sequence takes place, the EUSART state machine is held in IDLE. The RCIF interrupt is set once the fifth rising edge on RX is detected. The value in the RCREG needs to be read to clear the RCIF interrupt. RCREG content should be discarded.

**Note 1:** If the WUE bit is set with the ABDEN bit, auto-baud rate detection will occur on the byte *following* the Break character (see **Section 12.3.4 "Auto-Wake-up on RX Pin Falling Edge"**).

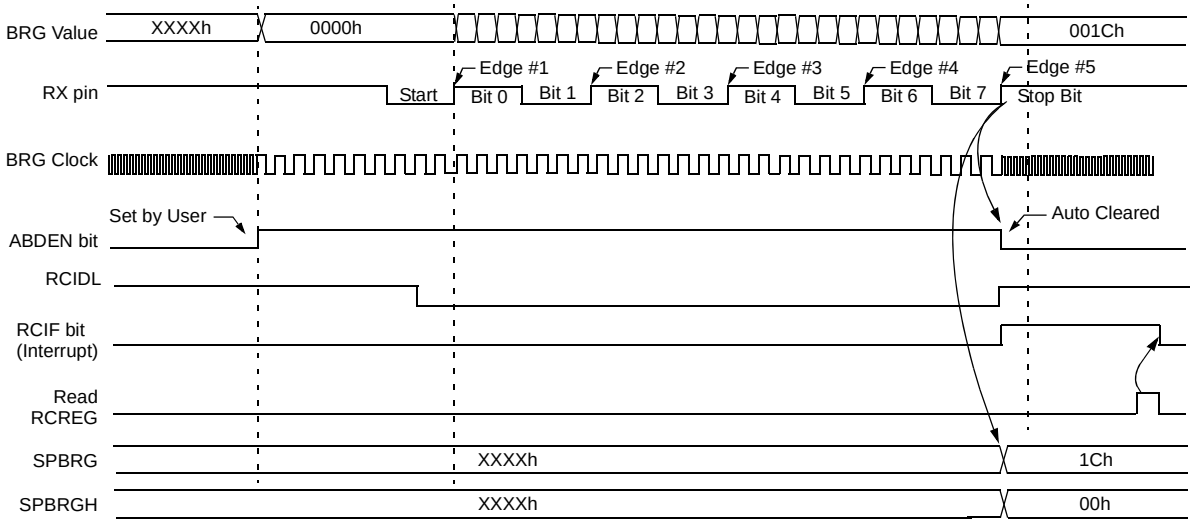
**2:** It is up to the user to determine that the incoming character baud rate is within the range of the selected BRG clock source. Some combinations of oscillator frequency and EUSART baud rates are not possible due to bit error rates. Overall system timing and communication baud rates must be taken into consideration when using the Auto-Baud Detect feature.

**TABLE 12-4: BRG COUNTER CLOCK RATES**

| BRG16 | BRGH | BRG Counter Clock |
|---|---|---|
| 0 | 0 | FOSC/512 |
| 0 | 1 | FOSC/128 |
| 1 | 0 | FOSC/128 |
| 1 | 1 | FOSC/32 |

**Note:** During the ABD sequence, SPBRG and SPBRGH are both used as a 16-bit counter, independent of BRG16 setting.

**FIGURE 12-1: AUTOMATIC BAUD RATE CALCULATION**




**Note 1:** The ABD sequence requires the EUSART module to be configured in Asynchronous mode and WUE = 0.

## 12.3 EUSART Asynchronous Mode

The Asynchronous mode of operation is selected by clearing the SYNC bit (TXSTA<4>). In this mode, the EUSART uses standard non-return-to-zero (NRZ) format (one Start bit, eight or nine data bits and one Stop bit). The most common data format is 8 bits. An on-chip dedicated 8-bit/16-bit baud rate generator can be used to derive standard baud rate frequencies from the oscillator.

The EUSART transmits and receives the LSb first. The EUSART's transmitter and receiver are functionally independent, but use the same data format and baud rate. The baud rate generator produces a clock, either x16 or x64 of the bit shift rate, depending on the BRGH and BRG16 bits (TXSTA<2> and BAUDCTL<3>). Parity is not supported by the hardware, but can be implemented in software and stored as the 9th data bit.

Asynchronous mode is available in all times. It is available in Sleep mode only when auto-wake-up on Sync Break is enabled. The baud rate generator values may need to be adjusted if the clocks are changed.

When operating in Asynchronous mode, the EUSART module consists of the following important elements:

- Baud Rate Generator
- Sampling Circuit
- Asynchronous Transmitter
- Asynchronous Receiver
- Auto-wake-up on Sync Break Character
- 13-bit Break Character Transmit
- Auto-Baud Detection

### 12.3.1 EUSART ASYNCHRONOUS TRANSMITTER

The EUSART transmitter block diagram is shown in Figure 12-2. The heart of the transmitter is the Transmit (serial) Shift Register (TSR). The shift register obtains its data from the read/write transmit buffer, TXREG. The TXREG register is loaded with data in software. The TSR register is not loaded until the Stop bit has been transmitted from the previous load. As soon as the Stop bit is transmitted, the TSR is loaded with new data from the TXREG register (if available).

Once the TXREG register transfers the data to the TSR register (occurs in one $T_{CY}$), the TXREG register is empty and flag bit TXIF (PIR1<4>) is set. This interrupt can be enabled/disabled by setting/clearing enable bit TXIE (PIE1<4>). Flag bit TXIF will be set, regardless of the state of enable bit TXIE and cannot be cleared in software. Flag bit TXIF is not cleared immediately upon loading the transmit buffer register TXREG. TXIF becomes valid in the second instruction cycle following the load instruction. Polling TXIF immediately following a load of TXREG will return invalid results.

While flag bit TXIF indicates the status of the TXREG register, another bit, TRMT (TXSTA<1>), shows the status of the TSR register. Status bit TRMT is a read-only bit, which is set when the TSR register is empty. No interrupt logic is tied to this bit, so the user has to poll this bit in order to determine if the TSR register is empty.

| | |
|---|---|
| **Note 1:** | The TSR register is not mapped in data memory, so it is not available to the user. |
| **2:** | Flag bit TXIF is set when enable bit TXEN is set. |

To set up an Asynchronous Transmission:

1. Initialize the SPBRGH:SPBRG registers for the appropriate baud rate. Set or clear the BRGH and BRG16 bits, as required, to achieve the desired baud rate.
2. Enable the asynchronous serial port by clearing bit SYNC and setting bit SPEN.
3. If interrupts are desired, set enable bit TXIE.
4. If 9-bit transmission is desired, set transmit bit TX9. Can be used as address/data bit.
5. Enable the transmission by setting bit TXEN, which will also set bit TXIF.
6. If 9-bit transmission is selected, the ninth bit should be loaded in bit TX9D.
7. Load data to the TXREG register (starts transmission).

If using interrupts, ensure that the GIE and PEIE bits in the INTCON register (INTCON<7:6>) are set.

**FIGURE 12-2: EUSART TRANSMIT BLOCK DIAGRAM**

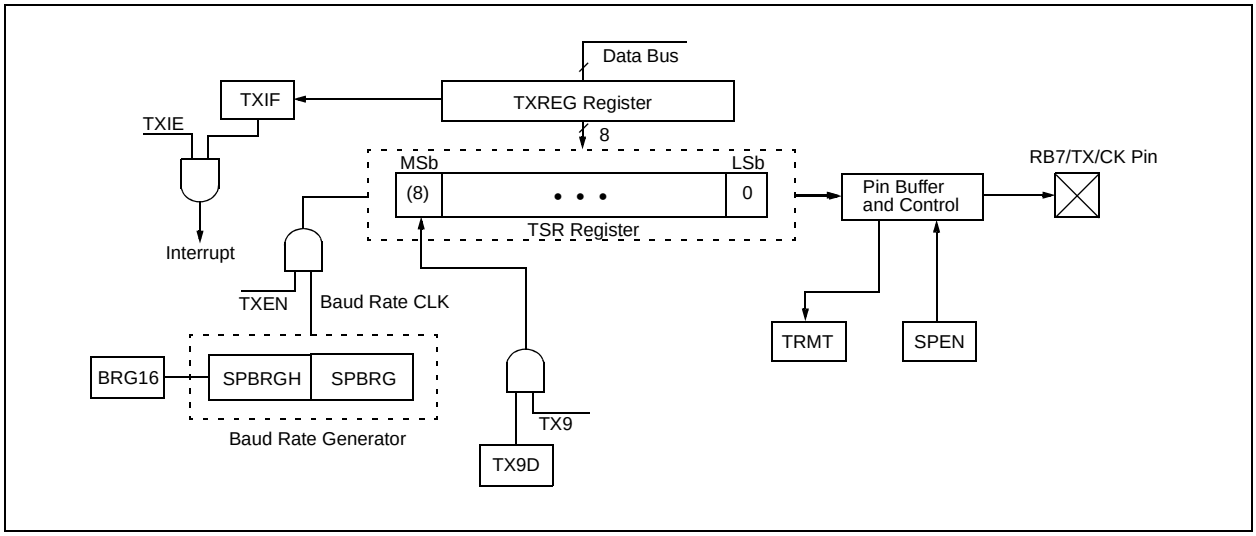



**FIGURE 12-3: ASYNCHRONOUS TRANSMISSION**

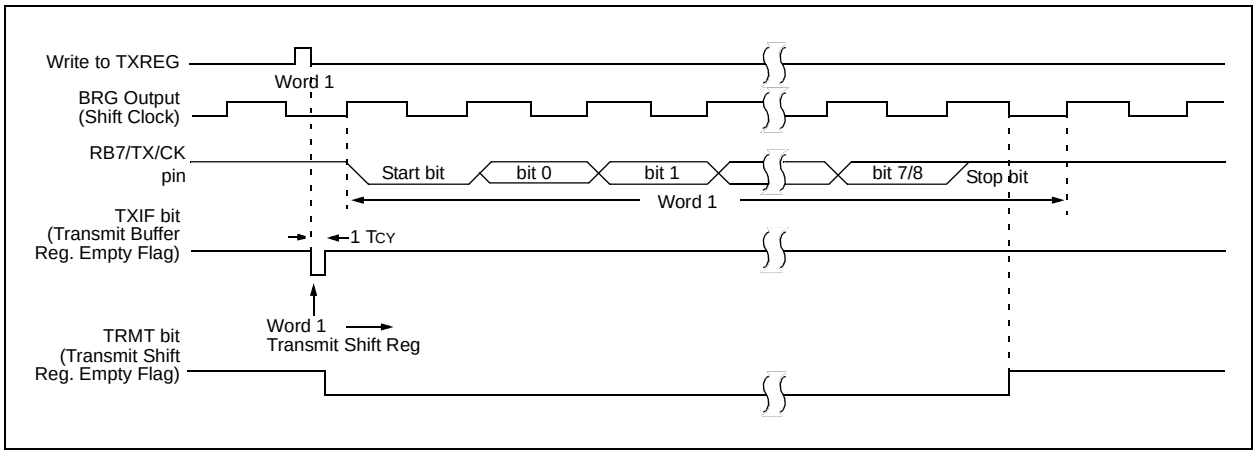
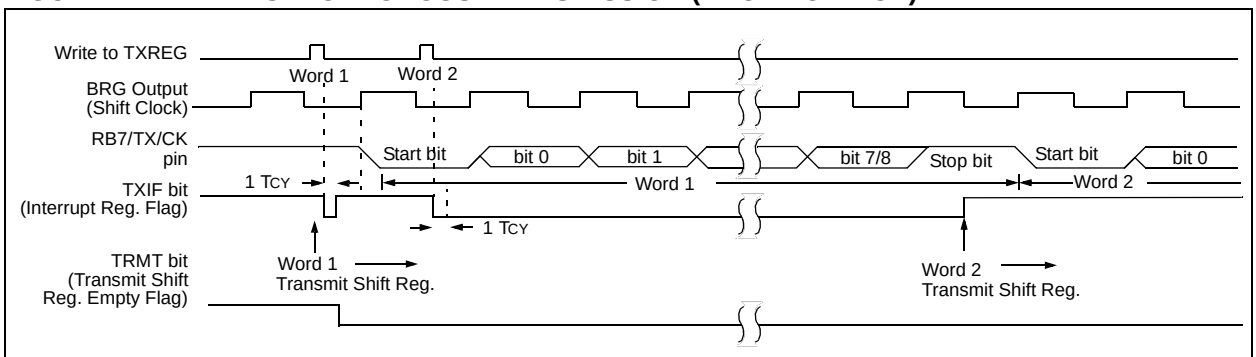



**FIGURE 12-4: ASYNCHRONOUS TRANSMISSION (BACK-TO-BACK)**




**Note:** This timing diagram shows two consecutive transmissions.

**TABLE 12-5: REGISTERS ASSOCIATED WITH ASYNCHRONOUS TRANSMISSION(1)**

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0Ch | PIR1 | — | ADIF | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | -000 0000 | -000 0000 |
| 18h | RCSTA | SPEN | RX9 | SREN | CREN | ADDEN | FERR | OERR | RX9D | 0000 000X | 0000 000X |
| 19h | TXREG | EUSART Transmit Data Register | | | | | | | | 0000 0000 | 0000 0000 |
| 1Ah | RCREG | EUSART Receive Data Register | | | | | | | | 0000 0000 | 0000 0000 |
| 86h | TRISB | TRISB7 | TRISB6 | TRISB5 | TRISB4 | — | — | — | — | 1111 ---- | 1111 ---- |
| 8Ch | PIE1 | — | ADIE | RCIE | TXIE | SSPIE | CCP1IE | TMR2IE | TMR1IE | -000 0000 | -000 0000 |
| 98h | TXSTA | CSRC | TX9 | TXEN | SYNC | SENB | BRGH | TRMT | TX9D | 0000 0010 | 0000 0010 |
| 99h | SPBRG | BRG7 | BRG6 | BRG5 | BRG4 | BRG3 | BRG2 | BRG1 | BRG0 | 0000 0000 | 0000 0000 |
| 9Ah | SPBRGH | BRG15 | BRG14 | BRG13 | BRG12 | BRG11 | BRG10 | BRG9 | BRG8 | 0000 0000 | 0000 0000 |
| 9Bh | BAUDCTL | ABDOVF | RCIDL | — | SCKP | BRG16 | — | WUE | ABDEN | 01-0 0-00 | 01-0 0-00 |

**Legend:** x = unknown, - = unimplemented locations read as '0'. Shaded cells are not used for Asynchronous Transmission.

**Note 1:** PIC16F687/PIC16F689/PIC16F690 only.

### 12.3.2 EUSART ASYNCHRONOUS RECEIVER

The receiver block diagram is shown in Figure 12-5. The data is received on the RB5/AN11/RX/DT pin and drives the data recovery block. The data recovery block is actually a high-speed shifter operating at 16 times the baud rate, whereas the main receive serial shifter operates at the bit rate or at FOSC. This mode would typically be used in RS-232 systems.

To set up an Asynchronous Reception:

1. Initialize the SPBRGH:SPBRG registers for the appropriate baud rate. Set or clear the BRGH and BRG16 bits, as required, to achieve the desired baud rate.
2. Enable the asynchronous serial port by clearing bit SYNC and setting bit SPEN.
3. If interrupts are desired, set enable bit RCIE.
4. If 9-bit reception is desired, set bit RX9.
5. Enable the reception by setting bit CREN.
6. Flag bit RCIF will be set when reception is complete and an interrupt will be generated if enable bit RCIE was set.
7. Read the RCSTA register to get the 9th bit (if enabled) and determine if any error occurred during reception.
8. Read the 8-bit received data by reading the RCREG register.
9. If any error occurred, clear the error by clearing enable bit CREN.
10. If using interrupts, ensure that the GIE and PEIE bits in the INTCON register (INTCON<7:6>) are set.

### 12.3.3 SETTING UP 9-BIT MODE WITH ADDRESS DETECT

This mode would typically be used in RS-485 systems. To set up an Asynchronous Reception with Address Detect Enable:

1. Initialize the SPBRGH:SPBRG registers for the appropriate baud rate. Set or clear the BRGH and BRG16 bits, as required, to achieve the desired baud rate.
2. Enable the asynchronous serial port by clearing the SYNC bit and setting the SPEN bit.
3. If interrupts are required, set the RCEN bit and select the desired priority level with the RCIP bit.
4. Set the RX9 bit to enable 9-bit reception.
5. Set the ADDEN bit to enable address detect.
6. Enable reception by setting the CREN bit.
7. The RCIF bit will be set when reception is complete. The interrupt will be acknowledged if the RCIE and GIE bits are set.
8. Read the RCSTA register to determine if any error occurred during reception, as well as read bit 9 of data (if applicable).
9. Read RCREG to determine if the device is being addressed.
10. If any error occurred, clear the CREN bit.
11. If the device has been addressed, clear the ADDEN bit to allow all received data into the receive buffer and interrupt the CPU.

**FIGURE 12-5: EUSART RECEIVE BLOCK DIAGRAM**

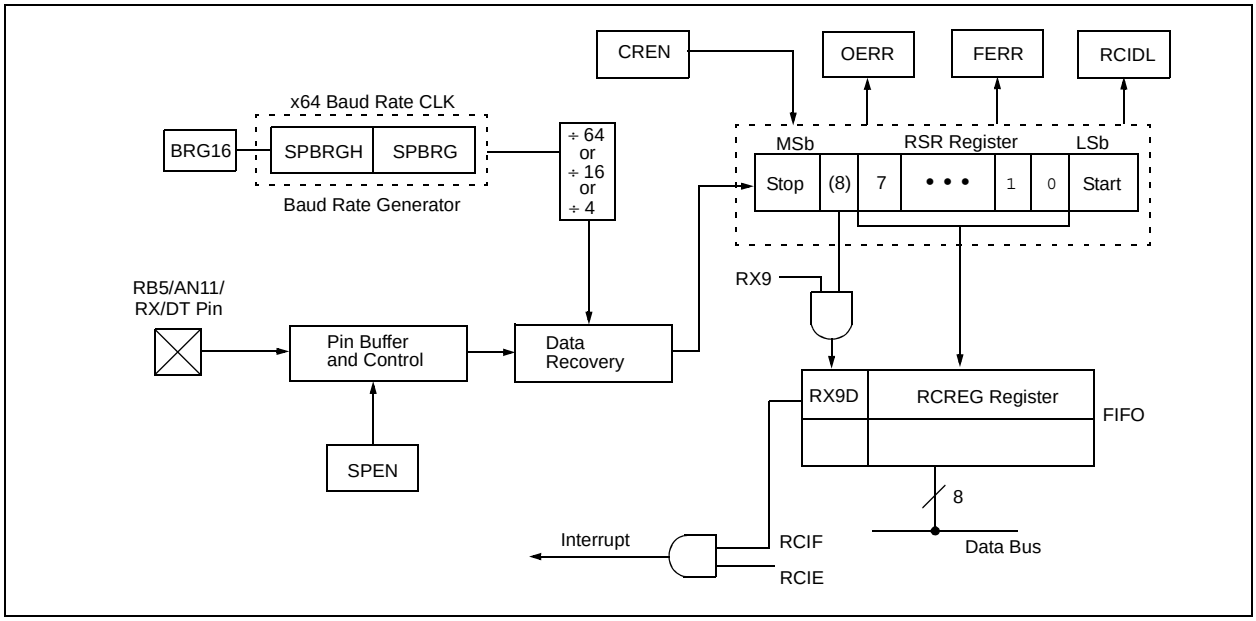
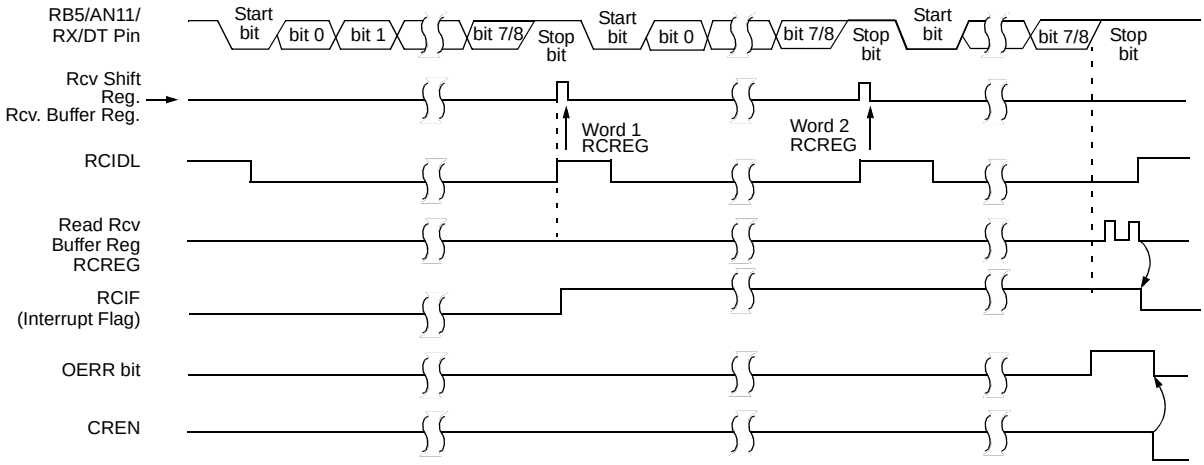


**FIGURE 12-6: ASYNCHRONOUS RECEPTION**




**Note:** This timing diagram shows three words appearing on the RX input. The RCREG (receive buffer) is read after the third word, causing the OERR (overrun) bit to be set.

**TABLE 12-6: REGISTERS ASSOCIATED WITH ASYNCHRONOUS RECEPTION(1)**

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0Ch | PIR1 | — | ADIF | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | -000 0000 | -000 0000 |
| 18h | RCSTA | SPEN | RX9 | SREN | CREN | ADDEN | FERR | OERR | RX9D | 0000 000X | 0000 000X |
| 19h | TXREG | EUSART Transmit Data Register | | | | | | | | 0000 0000 | 0000 0000 |
| 1Ah | RCREG | EUSART Receive Data Register | | | | | | | | 0000 0000 | 0000 0000 |
| 86h | TRISB | TRISB7 | TRISB6 | TRISB5 | TRISB4 | — | — | — | — | 1111 ---- | 1111 ---- |
| 8Ch | PIE1 | — | ADIE | RCIE | TXIE | SSPIE | CCPIE | TMR2IE | TMR1IE | -000 0000 | -000 0000 |
| 98h | TXSTA | CSRC | TX9 | TXEN | SYNC | SENB | BRGH | TRMT | TX9D | 0000 0010 | 0000 0010 |
| 99h | SPBRG | BRG7 | BRG6 | BRG5 | BRG4 | BRG3 | BRG2 | BRG1 | BRG0 | 0000 0000 | 0000 0000 |
| 9Ah | SPBRGH | BRG15 | BRG14 | BRG13 | BRG12 | BRG11 | BRG10 | BRG9 | BRG8 | 0000 0000 | 0000 0000 |
| 9Bh | BAUDCTL | ABDOVF | RCIDL | — | SCKP | BRG16 | — | WUE | ABDEN | 01-0 0-00 | 01-0 0-00 |

**Legend:** x = unknown, – = unimplemented locations read as '0'. Shaded cells are not used for Asynchronous Reception.

**Note 1:** PIC16F687/PIC16F689/PIC16F690 only.

### 12.3.4 AUTO-WAKE-UP ON RX PIN FALLING EDGE

The auto-wake-up feature allows the controller to wake-up due to activity on the RX/DT line, despite the baud clock being turned off. This allows communications systems to save power by only responding to direct requests.

Setting the WUE bit (BAUDCTL<1>) enables the auto-wake-up feature. When the auto-wake-up feature is enabled, the next falling edge on the RX/DT line will trigger an RCIF interrupt. The WUE bit will automatically clear after the rising RX/DT edge after triggering a falling edge. Receiving a RCIF interrupt after setting the WUE bit signals to the user that the wake-up event has occurred. See Figure 12-7 and Figure 12-8 for timing details of the auto-wake-up process.

#### 12.3.4.1 Special Considerations Using Auto-Wake-Up

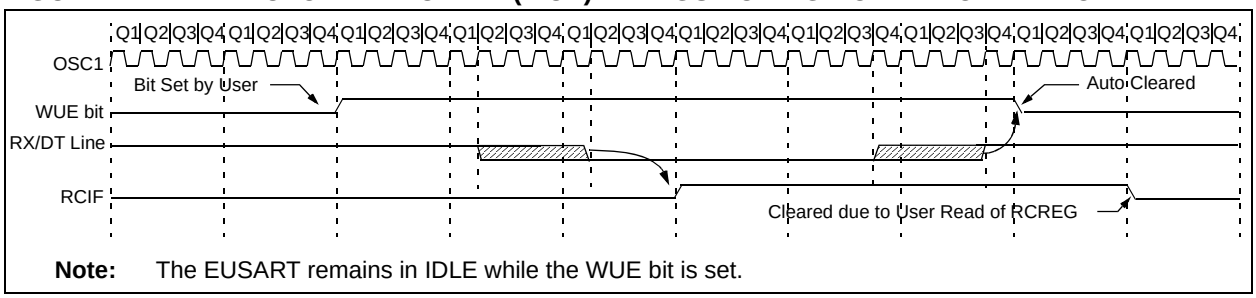
The auto-wake-up function is edge sensitive. To prevent data errors or framing errors, the data following the Break should be all '0's until the baud clock is stable. If the LP, XT or HS oscillators are used, the oscillator start-up time will affect the amount of time the application must wait before receiving valid data. Special care should be taken when using the Two-Speed Start-up or the Fail-Safe Clock Monitor because the application will start running from the internal oscillator before the primary oscillator is ready.

Because the auto-wake-up feature uses the RCIF flag to signify the wake-up event, the application should discard the data read from RCREG when servicing the RCIF flag after setting the WUE bit.

When entering Sleep with auto-wake-up enabled, the following procedure should be used.

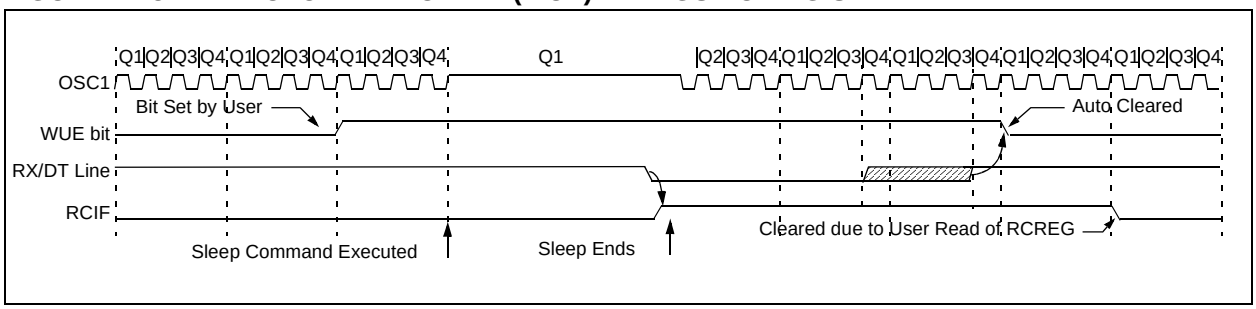
1. Clear all interrupt flags including RCIF.
2. Check RCIDL to ensure no receive is currently in progress.
3. No characters are being received so the WUE bit can be set.
4. Sleep.

**FIGURE 12-7: AUTO-WAKE-UP BIT (WUE) TIMINGS DURING NORMAL OPERATION**




**Note:** The EUSART remains in IDLE while the WUE bit is set.

**FIGURE 12-8: AUTO-WAKE-UP BIT (WUE) TIMINGS DURING SLEEP**



### 12.3.5 BREAK CHARACTER SEQUENCE

The EUSART module has the capability of sending the special Break character sequences that are required by the LIN bus standard. The Break character transmit consists of a Start bit, followed by 12 '0' bits and a Stop bit. The frame Break character is sent whenever the SENB and TXEN bits (TXSTA<3> and TXSTA<5>) are set, while the Transmit Shift register is loaded with data. Note that the value of data written to TXREG will be ignored and all '0's will be transmitted.

The SENB bit is automatically reset by hardware after the corresponding Stop bit is sent. This allows the user to preload the transmit FIFO with the next transmit byte following the Break character (typically, the Sync character in the LIN specification).

Note that the data value written to the TXREG for the Break character is ignored. The write simply serves the purpose of initiating the proper sequence.

The TRMT bit indicates when the transmit operation is active or IDLE, just as it does during normal transmission. See Figure 12-9 for the timing of the Break character sequence.

#### 12.3.5.1 Break and Sync Transmit Sequence

The following sequence will send a message frame header made up of a Break, followed by an auto-baud Sync byte. This sequence is typical of a LIN bus master.

1. Configure the EUSART for the desired mode.
2. Set the TXEN and SENB bits to setup the Break character.
3. Load the TXREG with a dummy character to initiate transmission (the value is ignored).
4. Write '55h' to TXREG to load the Sync character into the transmit FIFO buffer.
5. After the Break has been sent, the SENB bit is reset by hardware. The Sync character now transmits in the Pre-Configured mode.

When the TXREG becomes empty, as indicated by the TXIF, the next data byte can be written to TXREG.

### 12.3.6 RECEIVING A BREAK CHARACTER

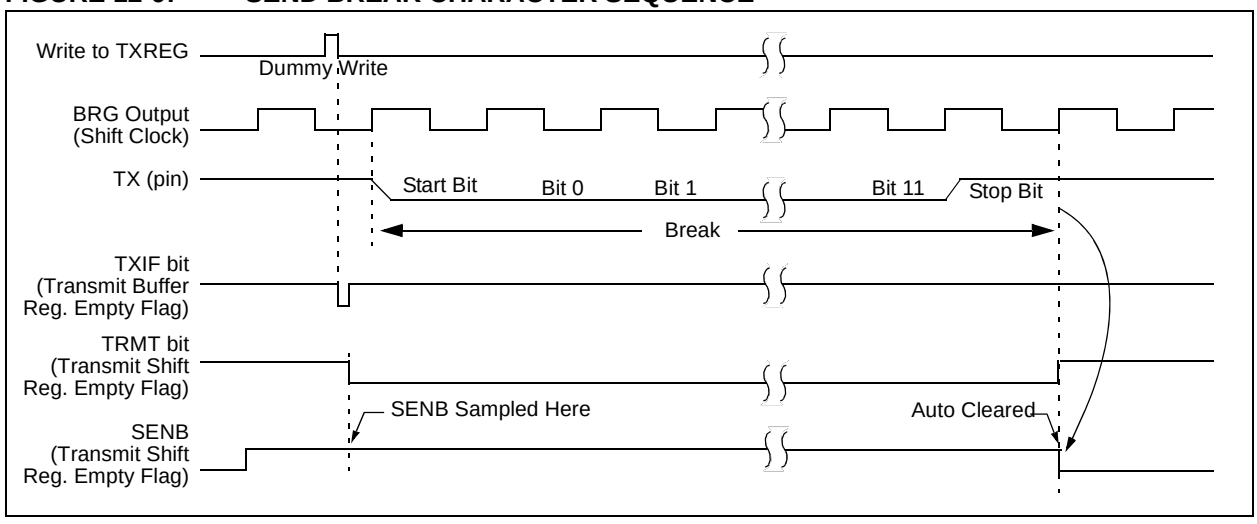
The EUSART module can receive a Break character in two ways.

The first method forces to configure the baud rate at a frequency of 9/13 the typical speed. This allows for the Stop bit transition to be at the correct sampling location (13 bits for Break versus Start bit and 8 data bits for typical data).

The second method uses the auto-wake-up feature described in **Section 12.3.4 "Auto-Wake-up on RX Pin Falling Edge"**. By enabling this feature, the EUSART will sample the next two transitions on RX/DT, cause an RCIF interrupt, and receive the next data byte followed by another interrupt.

Note that following a Break character, the user will typically want to enable the Auto-Baud Detect feature. For both methods, the user can set the ABD bit before placing the EUSART in its Sleep mode.

**FIGURE 12-9: SEND BREAK CHARACTER SEQUENCE**



## 12.4 EUSART Synchronous Master Mode

The Synchronous Master mode is entered by setting the CSRC bit (TXSTA<7>). In this mode, the data is transmitted in a half-duplex manner (i.e., transmission and reception do not occur at the same time). When transmitting data, the reception is inhibited and vice versa. Synchronous mode is entered by setting bit SYNC (TXSTA<4>). In addition, enable bit SPEN (RCSTA<7>) is set in order to configure the RB6/SCK/SCL and RB7/TX/CK or RB5/AN11/RX/DT I/O pins to CK (clock) and DT (data) lines, respectively.

The Master mode indicates that the processor transmits the master clock on the CK line. Clock polarity is selected with the SCKP bit (BAUDCTL<4>); setting SCKP sets the IDLE state on CK as high, while clearing the bit, sets the IDLE state low. This option is provided to support Microwire devices with this module.

### 12.4.1 EUSART SYNCHRONOUS MASTER TRANSMISSION

The EUSART transmitter block diagram is shown in Figure 12-2. The heart of the transmitter is the Transmit (serial) Shift Register (TSR). The shift register obtains its data from the read/write transmit buffer register TXREG. The TXREG register is loaded with data in software. The TSR register is not loaded until the last bit has been transmitted from the previous load. As soon as the last bit is transmitted, the TSR is loaded with new data from the TXREG (if available).

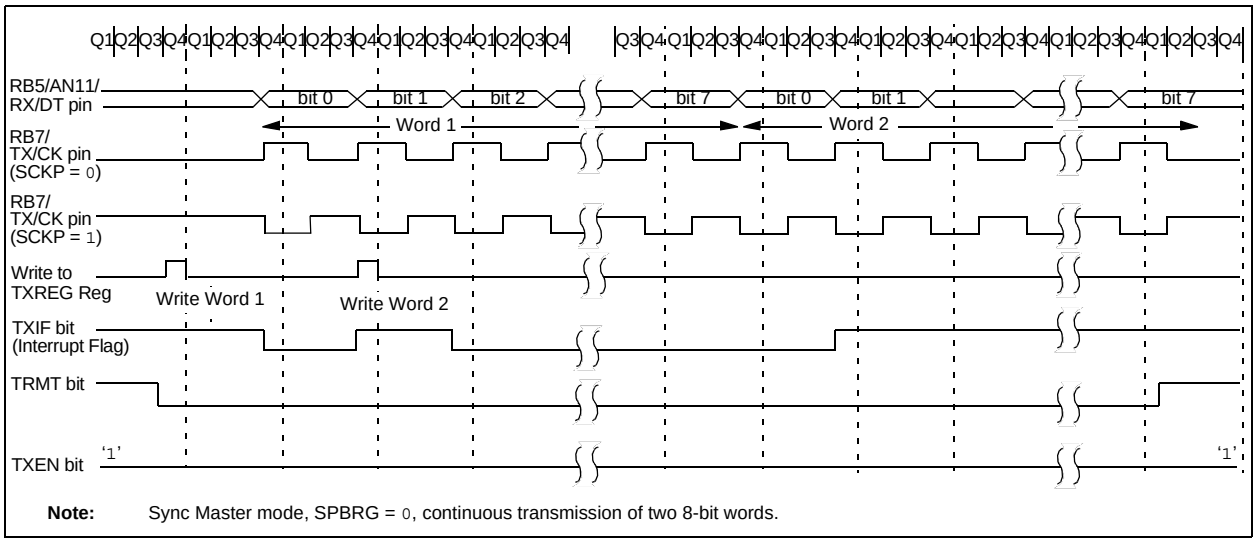
Once the TXREG register transfers the data to the TSR register (occurs in one TCYCLE), the TXREG is empty and interrupt bit TXIF (PIR1<4>) is set. The interrupt can be enabled/disabled by setting/clearing enable bit TXIE (PIE1<4>). Flag bit TXIF will be set, regardless of the state of enable bit TXIE, and cannot be cleared in software. It will reset only when new data is loaded into the TXREG register.

While flag bit TXIF indicates the status of the TXREG register, another bit, TRMT (TXSTA<1>), shows the status of the TSR register. TRMT is a read-only bit, which is set when the TSR is empty. No interrupt logic is tied to this bit, so the user has to poll this bit in order to determine if the TSR register is empty. The TSR is not mapped in data memory, so it is not available to the user.

To set up a Synchronous Master Transmission:

1. Initialize the SPBRGH:SPBRG registers for the appropriate baud rate. Set or clear the BRGH and BRG16 bits, as required, to achieve the desired baud rate.
2. Enable the synchronous master serial port by setting bits SYNC, SPEN and CSRC.
3. If interrupts are desired, set enable bit TXIE.
4. If 9-bit transmission is desired, set bit TX9.
5. Enable the transmission by setting bit TXEN.
6. If 9-bit transmission is selected, the ninth bit should be loaded in bit TX9D.
7. Start transmission by loading data to the TXREG register.
8. If using interrupts, ensure that the GIE and PEIE bits in the INTCON register (INTCON<7:6>) are set.

**FIGURE 12-10: SYNCHRONOUS TRANSMISSION**




**Note:** Sync Master mode, SPBRG = 0, continuous transmission of two 8-bit words.

**FIGURE 12-11: SYNCHRONOUS TRANSMISSION (THROUGH TXEN)**

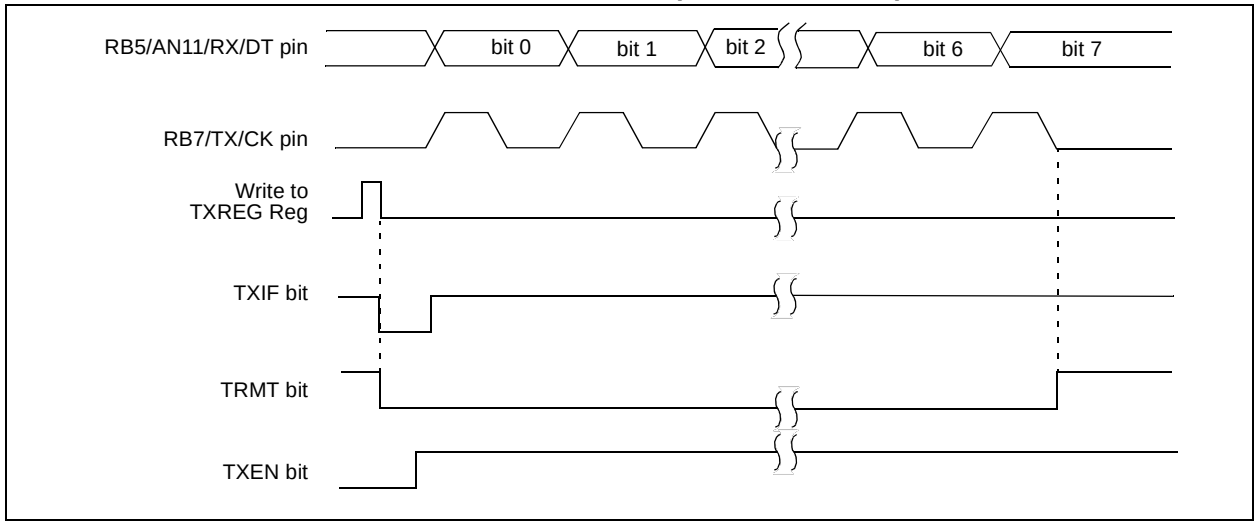



**TABLE 12-7: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER TRANSMISSION(1)**

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0Ch | PIR1 | — | ADIF | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | -000 0000 | -000 0000 |
| 18h | RCSTA | SPEN | RX9 | SREN | CREN | ADDEN | FERR | OERR | RX9D | 0000 000X | 0000 000X |
| 19h | TXREG | EUSART Transmit Data Register | | | | | | | | 0000 0000 | 0000 0000 |
| 1Ah | RCREG | EUSART Receive Data Register | | | | | | | | 0000 0000 | 0000 0000 |
| 86h | TRISB | TRISB7 | TRISB6 | TRISB5 | TRISB4 | — | — | — | — | 1111 ---- | 1111 ---- |
| 8Ch | PIE1 | — | ADIE | RCIE | TXIE | SSPIE | CCPIE | TMR2IE | TMR1IE | -000 0000 | -000 0000 |
| 98h | TXSTA | CSRC | TX9 | TXEN | SYNC | SENB | BRGH | TRMT | TX9D | 0000 0010 | 0000 0010 |
| 99h | SPBRG | BRG7 | BRG6 | BRG5 | BRG4 | BRG3 | BRG2 | BRG1 | BRG0 | 0000 0000 | 0000 0000 |
| 9Ah | SPBRGH | BRG15 | BRG14 | BRG13 | BRG12 | BRG11 | BRG10 | BRG9 | BRG8 | 0000 0000 | 0000 0000 |
| 9Bh | BAUDCTL | ABDOVF | RCIDL | — | SCKP | BRG16 | — | WUE | ABDEN | 01-0 0-00 | 01-0 0-00 |

**Legend:** x = unknown, – = unimplemented locations read as '0'. Shaded cells are not used for Asynchronous Reception.

**Note 1:** PIC16F687/PIC16F689/PIC16F690 only.

### 12.4.2 EUSART SYNCHRONOUS MASTER RECEPTION

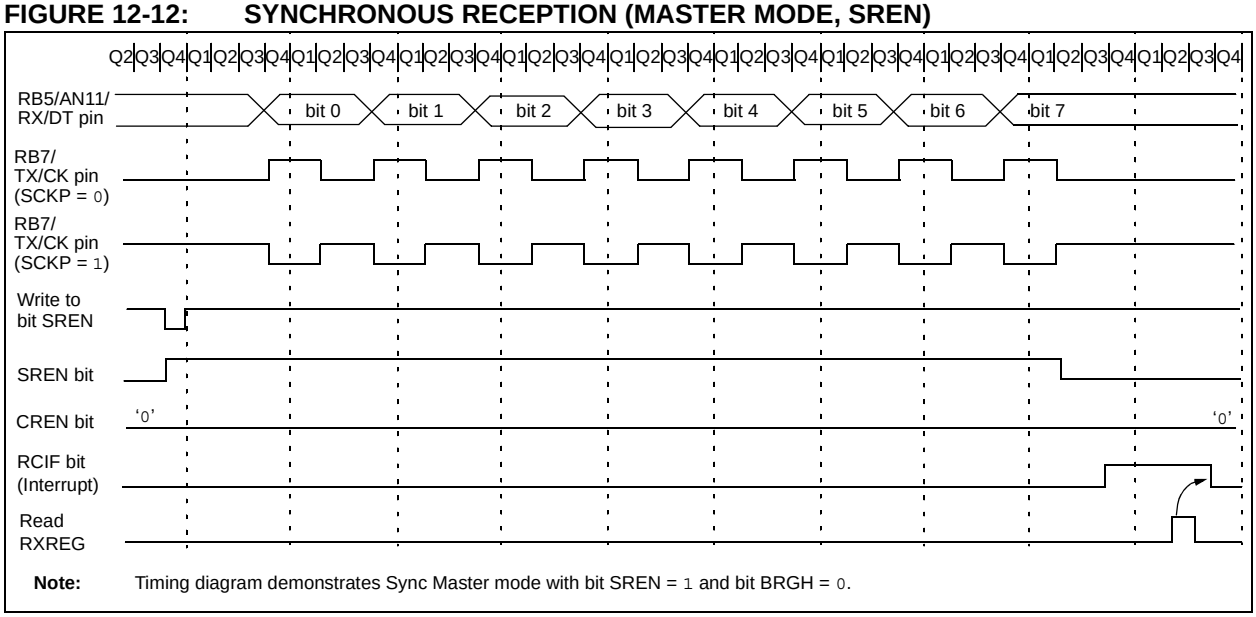
Once Synchronous mode is selected, reception is enabled by setting either the Single Receive Enable bit SREN (RCSTA<5>), or the Continuous Receive Enable bit, CREN (RCSTA<4>). Data is sampled on the RB5/AN11/RX/DT pin on the falling edge of the clock.

If enable bit SREN is set, only a single word is received. If enable bit CREN is set, the reception is continuous until CREN is cleared. If both bits are set, then CREN takes precedence.

To set up a Synchronous Master Reception:

1. Initialize the SPBRGH:SPBRG registers for the appropriate baud rate. Set or clear the BRGH and BRG16 bits, as required, to achieve the desired baud rate.
2. Enable the synchronous master serial port by setting bits SYNC, SPEN and CSRC.
3. Ensure bits CREN and SREN are clear.
4. If interrupts are desired, set enable bit RCIE.
5. If 9-bit reception is desired, set bit RX9.
6. If a single reception is required, set bit SREN. For continuous reception, set bit CREN.
7. Interrupt flag bit RCIF will be set when reception is complete and an interrupt will be generated if the enable bit RCIE was set.
8. Read the RCSTA register to get the 9th bit (if enabled) and determine if any error occurred during reception.
9. Read the 8-bit received data by reading the RCREG register.
10. If any error occurred, clear the error by clearing bit CREN.
11. If using interrupts, ensure that the GIE and PEIE bits in the INTCON register (INTCON<7:6>) are set.

**FIGURE 12-12: SYNCHRONOUS RECEPTION (MASTER MODE, SREN)**




**Note:** Timing diagram demonstrates Sync Master mode with bit SREN = 1 and bit BRGH = 0.

**TABLE 12-8: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER RECEPTION(1)**

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0Ch | PIR1 | — | ADIF | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | -000 0000 | -000 0000 |
| 18h | RCSTA | SPEN | RX9 | SREN | CREN | ADDEN | FERR | OERR | RX9D | 0000 000X | 0000 000X |
| 19h | TXREG | EUSART Transmit Data Register | | | | | | | | 0000 0000 | 0000 0000 |
| 1Ah | RCREG | EUSART Receive Data Register | | | | | | | | 0000 0000 | 0000 0000 |
| 86h | TRISB | TRISB7 | TRISB6 | TRISB5 | TRISB4 | — | — | — | — | 1111 ---- | 1111 ---- |
| 8Ch | PIE1 | — | ADIE | RCIE | TXIE | SSPIE | CCPIE | TMR2IE | TMR1IE | -000 0000 | -000 0000 |
| 98h | TXSTA | CSRC | TX9 | TXEN | SYNC | SENB | BRGH | TRMT | TX9D | 0000 0010 | 0000 0010 |
| 99h | SPBRG | BRG7 | BRG6 | BRG5 | BRG4 | BRG3 | BRG2 | BRG1 | BRG0 | 0000 0000 | 0000 0000 |
| 9Ah | SPBRGH | BRG15 | BRG14 | BRG13 | BRG12 | BRG11 | BRG10 | BRG9 | BRG8 | 0000 0000 | 0000 0000 |
| 9Bh | BAUDCTL | ABDOVF | RCIDL | — | SCKP | BRG16 | — | WUE | ABDEN | 01-0 0-00 | 01-0 0-00 |

**Legend:** x = unknown, – = unimplemented locations read as '0'. Shaded cells are not used for Asynchronous Reception.

**Note 1:** PIC16F687/PIC16F689/PIC16F690 only.

## 12.5 EUSART Synchronous Slave Mode

Synchronous Slave mode is entered by clearing bit CSRC (TXSTA<7>). This mode differs from the Synchronous Master mode in that the shift clock is supplied externally at the RB7/TX/CK pin (instead of being supplied internally in Master mode). This allows the device to transfer or receive data while in any Low-power mode.

### 12.5.1 EUSART SYNCHRONOUS SLAVE TRANSMIT

The operation of the Synchronous Master and Slave modes are identical, except in the case of the Sleep mode.

If two words are written to the TXREG and then the SLEEP instruction is executed, the following will occur:

a) The first word will immediately transfer to the TSR register.
b) The second word will remain in TXREG register.
c) Flag bit TXIF will not be set.
d) When the first word has been shifted out of TSR, the TXREG register will transfer the second word to the TSR and flag bit TXIF will now be set.
e) If enable bit TXIE is set, the interrupt will wake the chip from Sleep. If the global interrupt is enabled, the program will branch to the interrupt vector.

To set up a Synchronous Slave Transmission:

1. Enable the synchronous slave serial port by setting bits SYNC and SPEN and clearing bit CSRC.
2. Clear bits CREN and SREN.
3. If interrupts are desired, set enable bit TXIE.
4. If 9-bit transmission is desired, set bit TX9.
5. Enable the transmission by setting enable bit TXEN.
6. If 9-bit transmission is selected, the ninth bit should be loaded in bit TX9D.
7. Load data to TXREG register.
8. TXREG data will be transmitted synchronous to the master clock.
9. If using interrupts, ensure that the GIE and PEIE bits in the INTCON register (INTCON<7:6>) are set.

**TABLE 12-9: REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE TRANSMISSION(1)**

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0Ch | PIR1 | — | ADIF | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | -000 0000 | -000 0000 |
| 18h | RCSTA | SPEN | RX9 | SREN | CREN | ADDEN | FERR | OERR | RX9D | 0000 000X | 0000 000X |
| 19h | TXREG | EUSART Transmit Data Register | | | | | | | | 0000 0000 | 0000 0000 |
| 1Ah | RCREG | EUSART Receive Data Register | | | | | | | | 0000 0000 | 0000 0000 |
| 86h | TRISB | TRISB7 | TRISB6 | TRISB5 | TRISB4 | — | — | — | — | 1111 ---- | 1111 ---- |
| 8Ch | PIE1 | — | ADIE | RCIE | TXIE | SSPIE | CCPIE | TMR2IE | TMR1IE | -000 0000 | -000 0000 |
| 98h | TXSTA | CSRC | TX9 | TXEN | SYNC | SENB | BRGH | TRMT | TX9D | 0000 0010 | 0000 0010 |
| 99h | SPBRG | BRG7 | BRG6 | BRG5 | BRG4 | BRG3 | BRG2 | BRG1 | BRG0 | 0000 0000 | 0000 0000 |
| 9Ah | SPBRGH | BRG15 | BRG14 | BRG13 | BRG12 | BRG11 | BRG10 | BRG9 | BRG8 | 0000 0000 | 0000 0000 |
| 9Bh | BAUDCTL | ABDOVF | RCIDL | — | SCKP | BRG16 | — | WUE | ABDEN | 01-0 0-00 | 01-0 0-00 |

**Legend:** x = unknown, – = unimplemented locations read as '0'. Shaded cells are not used for Asynchronous Reception.

**Note 1:** PIC16F687/PIC16F689/PIC16F690 only.

### 12.5.2 EUSART SYNCHRONOUS SLAVE RECEPTION

The operation of the Synchronous Master and Slave modes is identical, except in the case of Sleep, or any IDLE mode and bit SREN, which is a “don’t care” in Slave mode.

If receive is enabled by setting the CREN bit prior to entering Sleep, then a word may be received. Once the word is received, the RSR register will transfer the data to the RCREG register; if the RCIE enable bit is set, the interrupt generated will wake the chip from Sleep. If the global interrupt is enabled, the program will branch to the interrupt vector.

To set up a Synchronous Slave Reception:

1. Enable the synchronous master serial port by setting bits SYNC and SPEN and clearing bit CSRC.
2. If interrupts are desired, set enable bit RCIE.
3. If 9-bit reception is desired, set bit RX9.
4. To enable reception, set enable bit CREN.
5. Flag bit RCIF will be set when reception is complete. An interrupt will be generated if enable bit RCIE was set.
6. Read the RCSTA register to get the 9th bit (if enabled) and determine if any error occurred during reception.
7. Read the 8-bit received data by reading the RCREG register.
8. If any error occurred, clear the error by clearing bit CREN.
9. If using interrupts, ensure that the GIE and PEIE bits in the INTCON register (INTCON<7:6>) are set.

**TABLE 12-10: REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE RECEPTION(1)**

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0Ch | PIR1 | — | ADIF | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | -000 0000 | -000 0000 |
| 18h | RCSTA | SPEN | RX9 | SREN | CREN | ADDEN | FERR | OERR | RX9D | 0000 000X | 0000 000X |
| 19h | TXREG | EUSART Transmit Register | | | | | | | | 0000 0000 | 0000 0000 |
| 1Ah | RCREG | EUSART Receive Register | | | | | | | | 0000 0000 | 0000 0000 |
| 86h | TRISB | TRISB7 | TRISB6 | TRISB5 | TRISB4 | — | — | — | — | 1111 ---- | 1111 ---- |
| 8Ch | PIE1 | — | ADIE | RCIE | TXIE | SSPIE | CCPIE | TMR2IE | TMR1IE | -000 0000 | -000 0000 |
| 98h | TXSTA | CSRC | TX9 | TXEN | SYNC | SENB | BRGH | TRMT | TX9D | 0000 0010 | 0000 0010 |
| 99h | SPBRG | BRG7 | BRG6 | BRG5 | BRG4 | BRG3 | BRG2 | BRG1 | BRG0 | 0000 0000 | 0000 0000 |
| 9Ah | SPBRGH | BRG15 | BRG14 | BRG13 | BRG12 | BRG11 | BRG10 | BRG9 | BRG8 | 0000 0000 | 0000 0000 |
| 9Bh | BAUDCTL | ABDOVF | RCIDL | — | SCKP | BRG16 | — | WUE | ABDEN | 01-0 0-00 | 01-0 0-00 |

**Legend:** x = unknown, – = unimplemented locations read as ‘0’. Shaded cells are not used for Asynchronous Reception.

**Note 1:** PIC16F687/PIC16F689/PIC16F690 only.

**NOTES:**

## 13.0 SSP MODULE OVERVIEW

The Synchronous Serial Port (SSP) module is a serial interface used to communicate with other peripheral or microcontroller devices. These peripheral devices may be serial EEPROMs, shift registers, display drivers, A/D converters, etc. The SSP module can operate in one of two modes:

- Serial Peripheral Interface (SPI™)
- Inter-Integrated Circuit ($I^2C$™)

Refer to Application Note AN578, "*Use of the SSP Module in the Multi-Master Environment*" (DS00578).

### 13.1 SPI Mode

This section contains register definitions and operational characteristics of the SPI module.

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

- Serial Data Out (SDO) – RC7/AN9/SDO
- Serial Data In (SDI) – RB4/AN10/SDI/SDA
- Serial Clock (SCK) – RB6/SCK/SCL

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

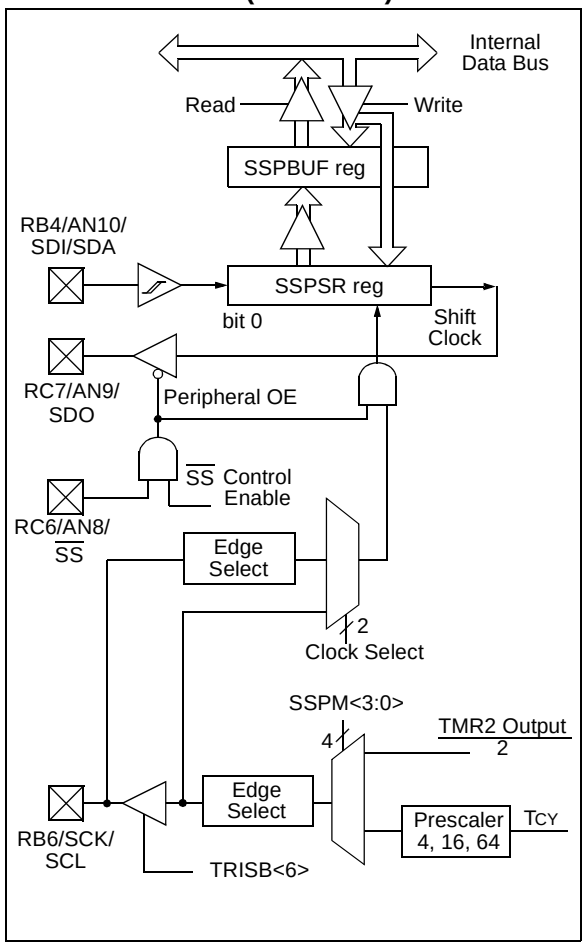
- Slave Select ($\overline{SS}$) – RC6/AN8/$\overline{SS}$

**Note 1:** When the SPI is in Slave mode with $\overline{SS}$ pin control enabled (SSPCON<3:0> = 0100), the SPI module will reset if the $\overline{SS}$ pin is set to $V_{DD}$.

**2:** If the SPI is used in Slave mode with CKE = 1, then the $\overline{SS}$ pin control must be enabled.

**3:** When the SPI is in Slave mode with $\overline{SS}$ pin control enabled (SSPCON<3:0> = 0100), the state of the $\overline{SS}$ pin can affect the state read back from the TRISC<4> bit. The peripheral OE signal from the SSP module into PORTC controls the state that is read back from the TRISC<4> bit (see **Section 17.0 "Electrical Specifications"** for information on PORTC). If read-write-modify instructions, such as BSF, are performed on the TRISC register while the $\overline{SS}$ pin is high, this will cause the TRISC<7> bit to be set, thus disabling the SDO output.

**FIGURE 13-1: SSP BLOCK DIAGRAM (SPI MODE)**



**REGISTER 13-1: SSPSTAT – SYNC SERIAL PORT STATUS REGISTER(1) (ADDRESS: 94h)**

| R/W-0 | R/W-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |
|---|---|---|---|---|---|---|---|
| SMP | CKE | D/$\overline{A}$ | P | S | R/$\overline{W}$ | UA | BF |
| bit 7 | | | | | | | bit 0 |

bit 7 **SMP**: SPI™ Data Input Sample Phase bit
SPI Master mode:
1 = Input data sampled at end of data output time
0 = Input data sampled at middle of data output time (Microwire)
SPI Slave mode:
SMP must be cleared when SPI is used in Slave mode
$I^2C$™ mode:
This bit must be maintained clear

bit 6 **CKE**: SPI Clock Edge Select bit
SPI mode, CKP = 0:
1 = Data transmitted on rising edge of SCK (Microwire alternate)
0 = Data transmitted on falling edge of SCK
SPI mode, CKP = 1:
1 = Data transmitted on falling edge of SCK (Microwire default)
0 = Data transmitted on rising edge of SCK
$I^2C$ mode:
This bit must be maintained clear

bit 5 **D/$\overline{A}$**: Data/$\overline{Address}$ bit ($I^2C$ mode only)
1 = Indicates that the last byte received or transmitted was data
0 = Indicates that the last byte received or transmitted was address

bit 4 **P**: Stop bit ($I^2C$ mode only)
This bit is cleared when the SSP module is disabled, or when the Start bit is detected last. SSPEN is cleared.
1 = Indicates that a Stop bit has been detected last (this bit is '0' on Reset)
0 = Stop bit was not detected last

bit 3 **S**: Start bit ($I^2C$ mode only)
This bit is cleared when the SSP module is disabled, or when the Stop bit is detected last. SSPEN is cleared.
1 = Indicates that a Start bit has been detected last (this bit is '0' on Reset)
0 = Start bit was not detected last

bit 2 **R/$\overline{W}$**: Read/$\overline{Write}$ bit Information ($I^2C$ mode only)
This bit holds the R/$\overline{W}$ bit information following the last address match. This bit is only valid from the address match to the next Start bit, Stop bit or $\overline{ACK}$ bit.
1 = Read
0 = Write

bit 1 **UA**: Update Address bit (10-bit $I^2C$ mode only)
1 = Indicates that the user needs to update the address in the SSPADD register
0 = Address does not need to be updated

bit 0 **BF**: Buffer Full Status bit
Receive (SPI and $I^2C$ modes):
1 = Receive complete, SSPBUF is full
0 = Receive not complete, SSPBUF is empty
Transmit ($I^2C$ mode only):
1 = Transmit in progress, SSPBUF is full
0 = Transmit complete, SSPBUF is empty

**Note 1:** PIC16F687/PIC16F689/PIC16F690 only.

| **Legend:** | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' | |
| - n = Value at POR | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |

**REGISTER 13-2: SSPCON – SYNC SERIAL PORT CONTROL REGISTER(1) (ADDRESS: 14h)**

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| WCOL | SSPOV | SSPEN | CKP | SSPM3(2) | SSPM2(2) | SSPM1(2) | SSPM0(2) |
| bit 7 | | | | | | | bit 0 |

bit 7 **WCOL**: Write Collision Detect bit
1 = The SSPBUF register is written while it is still transmitting the previous word (must be cleared in software)
0 = No collision

bit 6 **SSPOV**: Receive Overflow Indicator bit
In SPI™ mode:
1 = A new byte is received while the SSPBUF register is still holding the previous data. In case of overflow, the data in SSPSR is lost. Overflow can only occur in Slave mode. The user must read the SSPBUF, even if only transmitting data, to avoid setting overflow. In Master mode, the overflow bit is not set since each new reception (and transmission) is initiated by writing to the SSPBUF register.
0 = No overflow
In $I^2C$™ mode:
1 = A byte is received while the SSPBUF register is still holding the previous byte. SSPOV is a "don't care" in Transmit mode. SSPOV must be cleared in software in either mode.
0 = No overflow

bit 5 **SSPEN**: Synchronous Serial Port Enable bit
In SPI mode:
1 = Enables serial port and configures SCK, SDO and SDI as serial port pins
0 = Disables serial port and configures these pins as I/O port pins
In $I^2C$ mode:
1 = Enables the serial port and configures the SDA and SCL pins as serial port pins
0 = Disables serial port and configures these pins as I/O port pins
In both modes, when enabled, these pins must be properly configured as input or output.

bit 4 **CKP**: Clock Polarity Select bit
In SPI mode:
1 = Idle state for clock is a high level (Microwire default)
0 = Idle state for clock is a low level (Microwire alternate)
In $I^2C$ mode:
SCK release control
1 = Enable clock
0 = Holds clock low (clock stretch). (Used to ensure data setup time.)

bit 3-0 **SSPM<3:0>**: Synchronous Serial Port Mode Select bits
0000 = SPI Master mode, clock = FOSC/4
0001 = SPI Master mode, clock = FOSC/16
0010 = SPI Master mode, clock = FOSC/64
0011 = SPI Master mode, clock = TMR2 output/2
0100 = SPI Slave mode, clock = SCK pin. $\overline{SS}$ pin control enabled.
0101 = SPI Slave mode, clock = SCK pin. $\overline{SS}$ pin control disabled. $\overline{SS}$ can be used as I/O pin.
0110 = $I^2C$ Slave mode, 7-bit address
0111 = $I^2C$ Slave mode, 10-bit address
1000 = Reserved
1001 = Load SSPMSK register at SSPADD SFR address(2)
1010 = Reserved
1011 = $I^2C$ Firmware Controlled Master mode (slave IDLE)
1100 = Reserved
1101 = Reserved
1110 = $I^2C$ Slave mode, 7-bit address with Start and Stop bit interrupts enabled
1111 = $I^2C$ Slave mode, 10-bit address with Start and Stop bit interrupts enabled

**Note 1:** PIC16F687/PIC16F689/PIC16F690 only.
**2:** When this mode is selected, any reads or writes to the SSPADD SFR address actually accesses the SSPMSK register.

| **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 |

## 13.2 Operation

When initializing the SPI, several options need to be specified. This is done by programming the appropriate control bits (SSPCON<5:0> and SSPSTAT<7:6>). These control bits allow the following to be specified:

- Master mode (SCK is the clock output)
- Slave mode (SCK is the clock input)
- Clock Polarity (Idle state of SCK)
- Data Input Sample Phase (middle or end of data output time)
- Clock Edge (output data on rising/falling edge of SCK)
- Clock Rate (Master mode only)
- Slave Select mode (Slave mode only)

The SSP consists of a transmit/receive shift register (SSPSR) and a buffer register (SSPBUF). The SSPSR shifts the data in and out of the device, MSb first. The SSPBUF holds the data that was written to the SSPSR until the received data is ready. Once the eight bits of data have been received, that byte is moved to the SSPBUF register. Then, the Buffer Full Status bit, BF (SSPSTAT<0>), and the interrupt flag bit, SSPIF, are set. This double-buffering of the received data (SSPBUF) allows the next byte to start reception before reading the data that was just received. Any write to the SSPBUF register during transmission/reception of data will be ignored and the Write Collision Detect bit, WCOL (SSPCON<7>), will be set. User software must clear the WCOL bit so that it can be determined if the following write(s) to the SSPBUF register completed successfully.

When the application software is expecting to receive valid data, the SSPBUF should be read before the next byte of data to transfer is written to the SSPBUF. Buffer Full bit, BF (SSPSTAT<0>), indicates when SSPBUF has been loaded with the received data (transmission is complete). When the SSPBUF is read, the BF bit is cleared. This data may be irrelevant if the SPI is only a transmitter. Generally, the SSP interrupt is used to determine when the transmission/reception has completed. The SSPBUF must be read and/or written. If the interrupt method is not going to be used, then software polling can be done to ensure that a write collision does not occur. Example 13-1 shows the loading of the SSPBUF (SSPSR) for data transmission.

The SSPSR is not directly readable or writable and can only be accessed by addressing the SSPBUF register. Additionally, the SSP Status register (SSPSTAT) indicates the various status conditions.

**EXAMPLE 13-1: LOADING THE SSPBUF (SSPSR) REGISTER**

```
        BSF     STATUS,RP0      ;Bank 1
        BCF     STATUS,RP1      ;
LOOP    BTFSS   SSPSTAT, BF     ;Has data been received(transmit complete)?
        GOTO    LOOP            ;No
        BCF     STATUS,RP0      ;Bank 0
        MOVF    SSPBUF, W       ;WREG reg = contents of SSPBUF
        MOVWF   RXDATA          ;Save in user RAM, if data is meaningful
        MOVF    TXDATA, W       ;W reg = contents of TXDATA
        MOVWF   SSPBUF          ;New data to xmit
```

## 13.3 Enabling SPI I/O

To enable the serial port, SSP Enable bit, SSPEN (SSPCON<5>), must be set. To reset or reconfigure SPI mode, clear the SSPEN bit, re-initialize the SSPCON registers and then set the SSPEN bit. This configures the SDI, SDO, SCK and $\overline{SS}$ pins as serial port pins. For the pins to behave as the serial port function, some must have their data direction bits (in the TRISB and TRISC registers) appropriately programmed. That is:

- SDI is automatically controlled by the SPI module
- SDO must have TRISC<7> bit cleared
- SCK (Master mode) must have TRISB<6> bit cleared
- SCK (Slave mode) must have TRISB<6> bit set
- $\overline{SS}$ must have TRISC<6> bit set

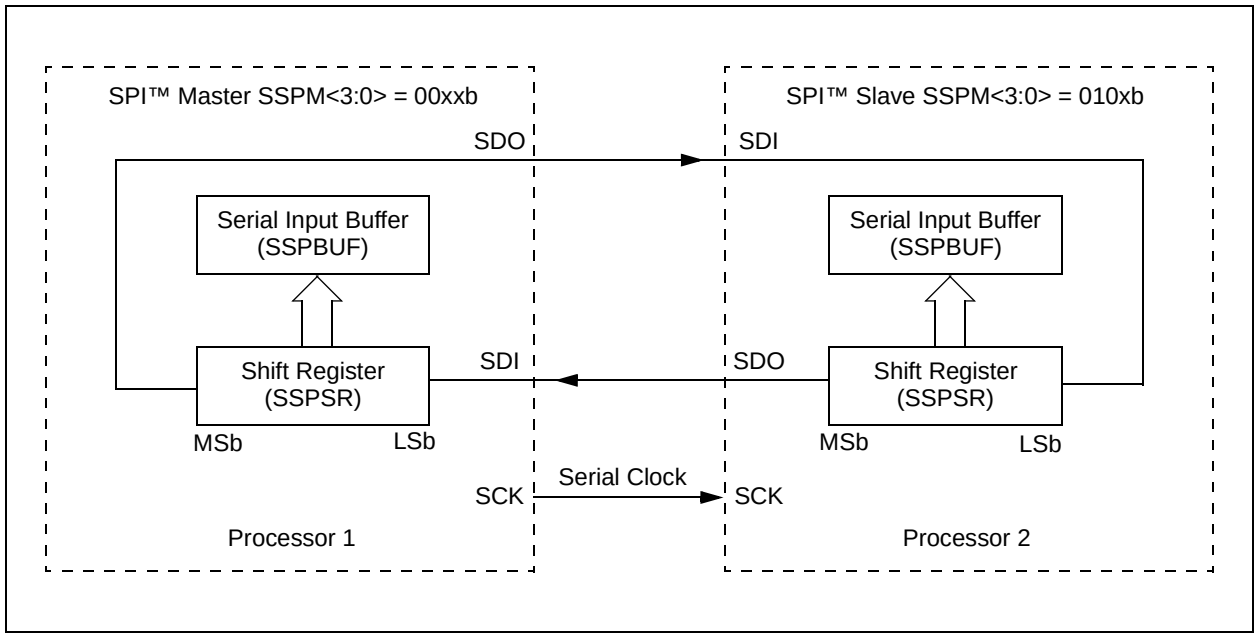
Any serial port function that is not desired may be overridden by programming the corresponding data direction (TRISB and TRISC) registers to the opposite value.

## 13.4 Typical Connection

Figure 13-2 shows a typical connection between two microcontrollers. The master controller (Processor 1) initiates the data transfer by sending the SCK signal. Data is shifted out of both shift registers on their programmed clock edge and latched on the opposite edge of the clock. Both processors should be programmed to the same Clock Polarity (CKP), then both controllers would send and receive data at the same time. Whether the data is meaningful (or dummy data) depends on the application software. This leads to three scenarios for data transmission:

- Master sends data – Slave sends dummy data
- Master sends data – Slave sends data
- Master sends dummy data – Slave sends data

**FIGURE 13-2: SPI™ MASTER/SLAVE CONNECTION**



## 13.5 Master Mode

The master can initiate the data transfer at any time because it controls the SCK. The master determines when the slave (Processor 2, Figure 13-2) is to broadcast data by the software protocol.

In Master mode, the data is transmitted/received as soon as the SSPBUF register is written to. If the SPI is only going to receive, the SDO output could be disabled (programmed as an input). The SSPSR register will continue to shift in the signal present on the SDI pin at the programmed clock rate. As each byte is received, it will be loaded into the SSPBUF register as if a normal received byte (interrupts and Status bits appropriately set). This could be useful in receiver applications as a “Line Activity Monitor” mode.

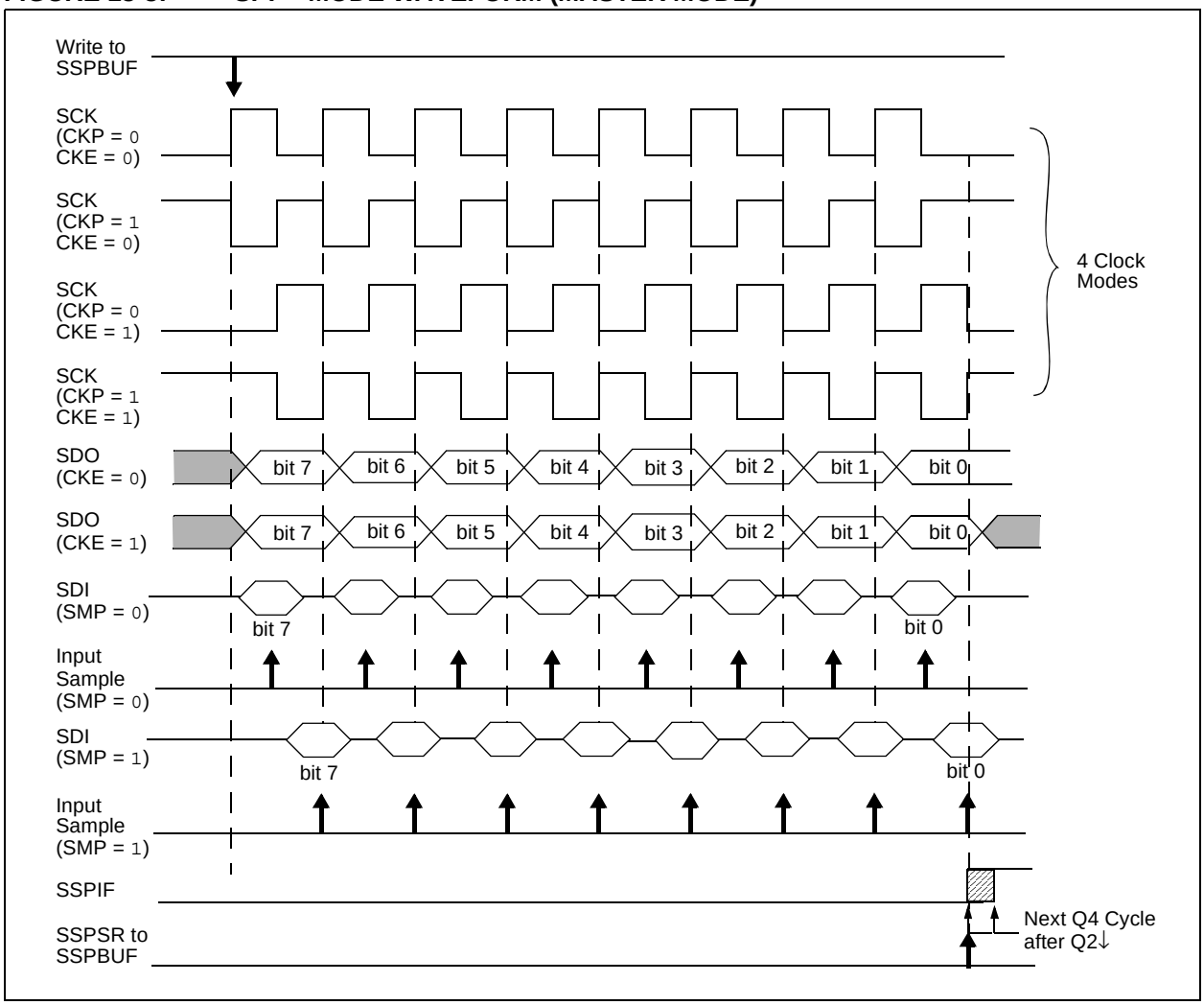
The clock polarity is selected by appropriately programming the CKP bit (SSPCON<4>). This then, would give waveforms for SPI communication as shown in Figure 13-3, Figure 13-5 and Figure 13-6, where the MSB is transmitted first. In Master mode, the SPI clock rate (bit rate) is user programmable to be one of the following:

- FOSC/4 (or TCY)
- FOSC/16 (or 4 • TCY)
- FOSC/64 (or 16 • TCY)
- Timer2 output/2

This allows a maximum data rate (at 40 MHz) of 10 Mbps.

Figure 13-3 shows the waveforms for Master mode. When the CKE bit is set, the SDO data is valid before there is a clock edge on SCK. The change of the input sample is shown based on the state of the SMP bit. The time when the SSPBUF is loaded with the received data is shown.

**FIGURE 13-3: SPI™ MODE WAVEFORM (MASTER MODE)**



## 13.6 Slave Mode

In Slave mode, the data is transmitted and received as the external clock pulses appear on SCK. When the last bit is latched, the SSPIF interrupt flag bit is set.

While in Slave mode, the external clock is supplied by the external clock source on the SCK pin. This external clock must meet the minimum high and low times as specified in the electrical specifications.

While in Sleep mode, the slave can transmit/receive data. When a byte is received, the device will wake-up from Sleep.

## 13.7 Slave Select Synchronization

The $\overline{SS}$ pin allows a Synchronous Slave mode. The SPI must be in Slave mode with $\overline{SS}$ pin control enabled (SSPCON<3:0> = 04h). The pin must not be driven low for the $\overline{SS}$ pin to function as an input. The data latch must be high. When the $\overline{SS}$ pin is low, transmission and reception are enabled and the SDO pin is driven. When the $\overline{SS}$ pin goes high, the SDO pin is no longer driven, even if in the middle of a transmitted byte, and becomes a floating output. External pull-up/pull-down resistors may be desirable, depending on the application.

| | |
|---|---|
| **Note 1:** | When the SPI is in Slave mode with $\overline{SS}$ pin control enabled (SSPCON<3:0> = 0100), the SPI module will reset if the $\overline{SS}$ pin is set to $V_{DD}$. |
| **2:** | If the SPI is used in Slave Mode with CKE set, then the $\overline{SS}$ pin control must be enabled. |

When the SPI module resets, the bit counter is forced to '0'. This can be done by either forcing the $\overline{SS}$ pin to a high level or clearing the SSPEN bit.

To emulate two-wire communication, the SDO pin can be connected to the SDI pin. When the SPI needs to operate as a receiver, the SDO pin can be configured as an input. This disables transmissions from the SDO. The SDI can always be left as an input (SDI function) since it cannot create a bus conflict.

**FIGURE 13-4: SLAVE SYNCHRONIZATION WAVEFORM**

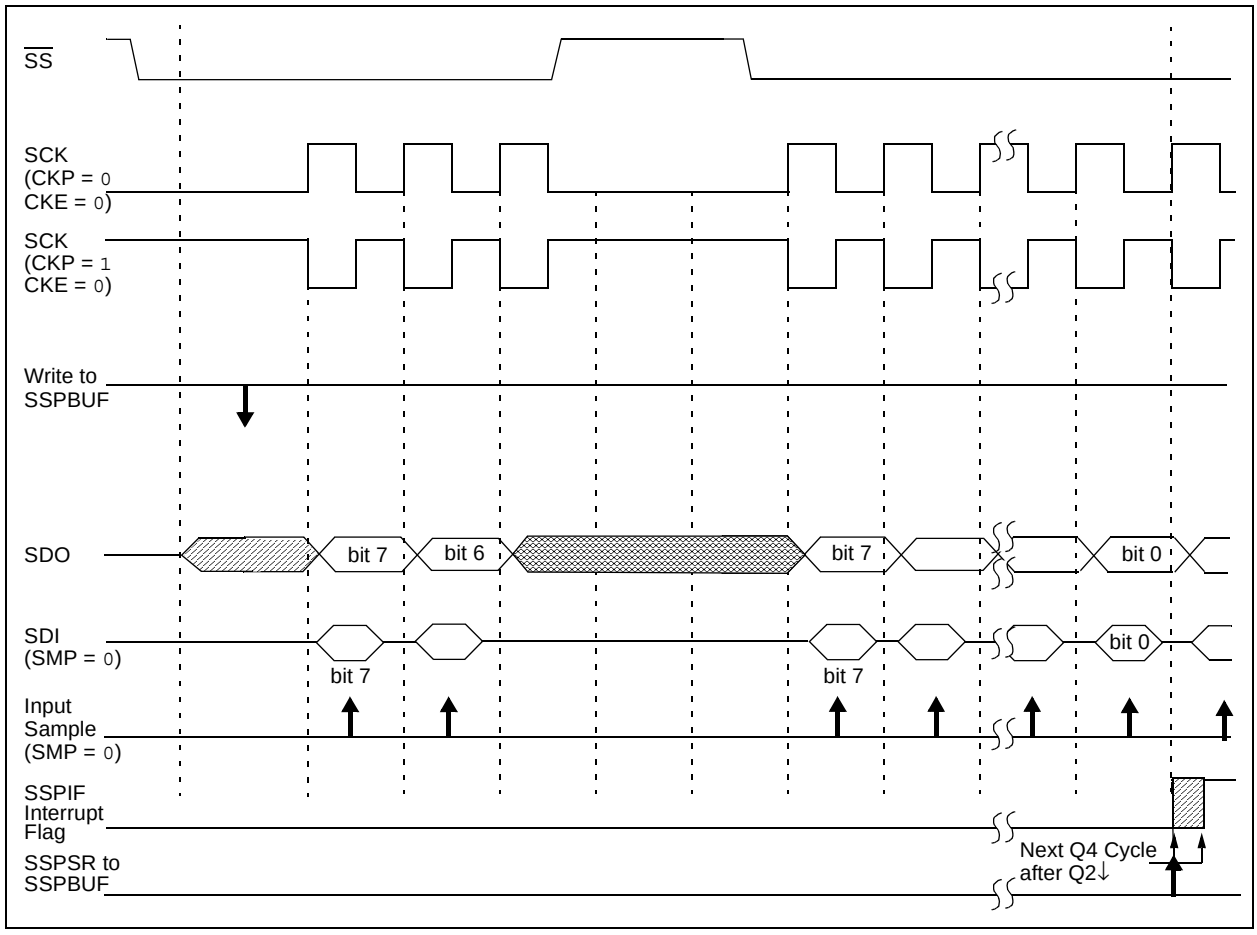


**FIGURE 13-5: SPI™ MODE WAVEFORM (SLAVE MODE WITH CKE = 0)**

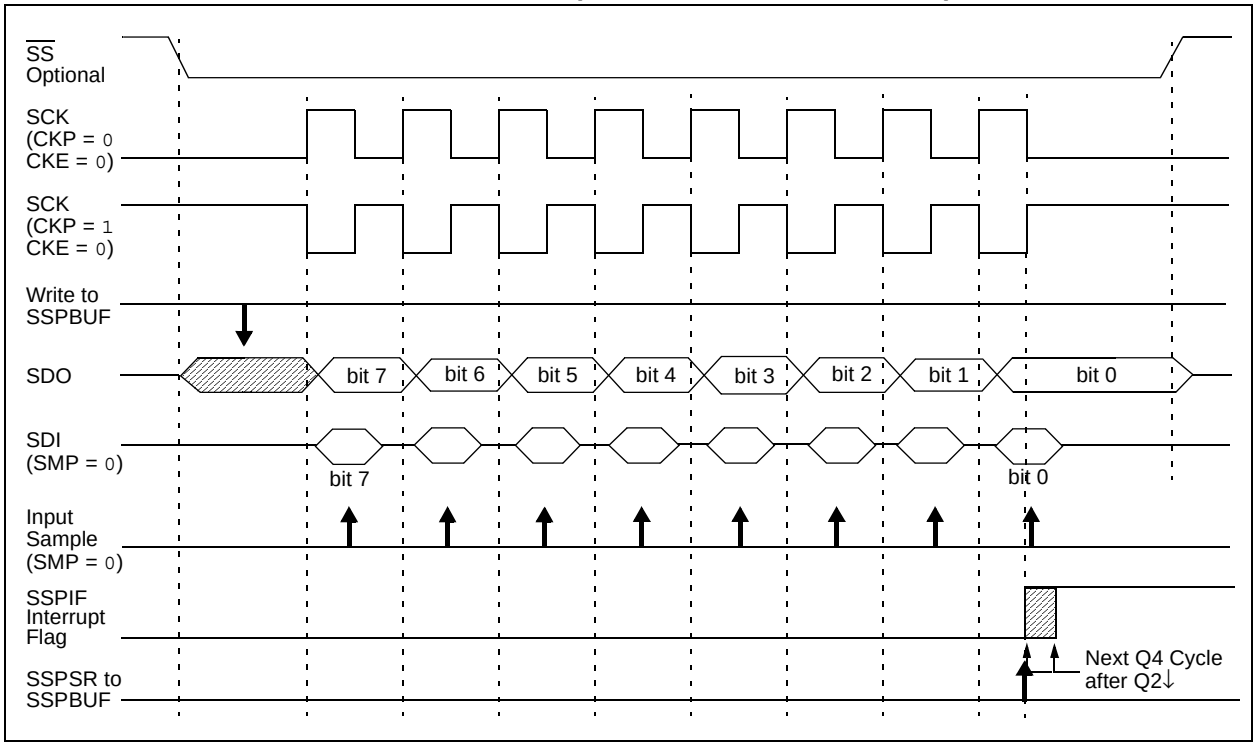
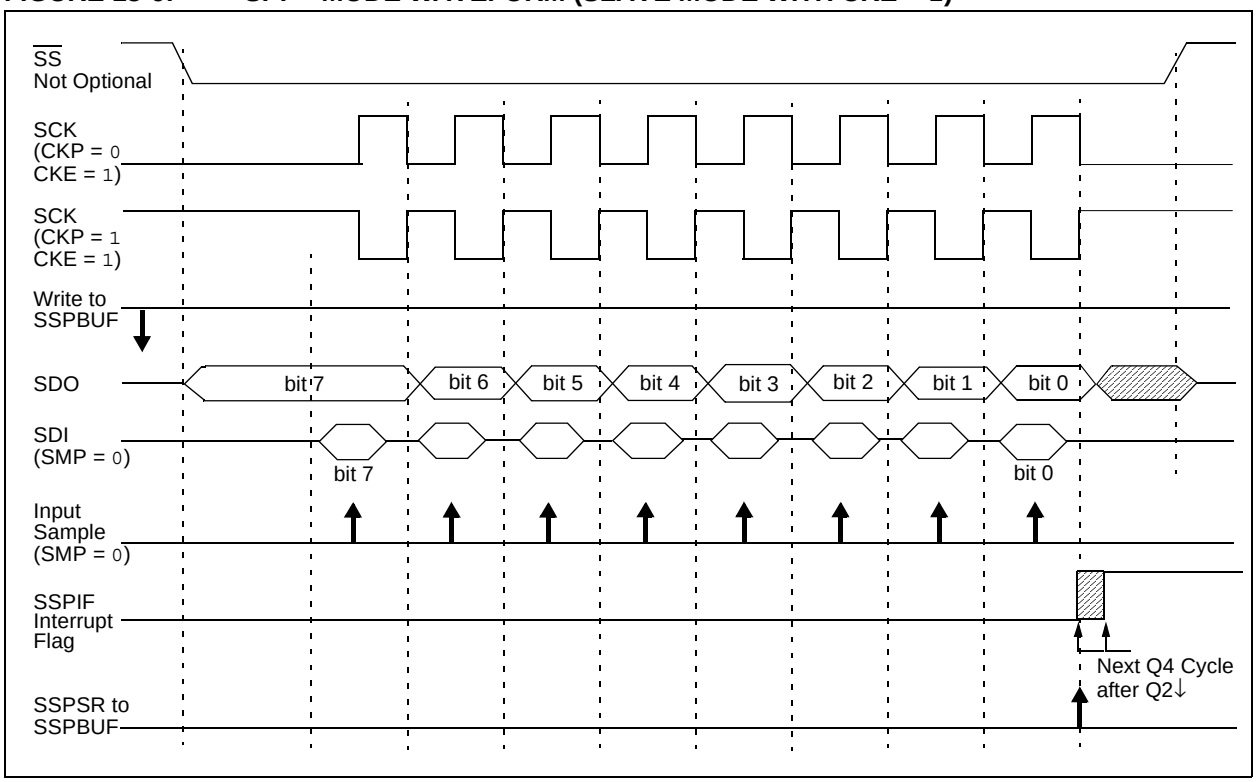



**FIGURE 13-6: SPI™ MODE WAVEFORM (SLAVE MODE WITH CKE = 1)**



## 13.8 Sleep Operation

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

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

## 13.9 Effects of a Reset

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

## 13.10 Bus Mode Compatibility

Table 13-1 shows the compatibility between the standard SPI modes and the states of the CKP and CKE control bits.

**TABLE 13-1: SPI™ BUS MODES**

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

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

**TABLE 13-2: REGISTERS ASSOCIATED WITH SPI™ OPERATION(1)**

| 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 | T0IE | INTE | RABIE | T0IF | INTF | RABIF | 0000 000x | 0000 000x |
| 0Ch | PIR1 | — | ADIF | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | -000 0000 | -000 0000 |
| 13h | SSPBUF | Synchronous Serial Port Receive Buffer/Transmit Register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 14h | SSPCON | WCOL | SSPOV | SSPEN | CKP | SSPM3 | SSPM2 | SSPM1 | SSPM0 | 0000 0000 | 0000 0000 |
| 86h/186h | TRISB | TRISB7 | TRISB6 | TRISB5 | TRISB4 | — | — | — | — | 1111 ---- | 1111 ---- |
| 87h/187h | TRISC | TRISC7 | TRISC6 | TRISC5 | TRISC4 | TRISC3 | TRISC2 | TRISC1 | TRISC0 | 1111 1111 | 1111 1111 |
| 8Ch | PIE1 | — | ADIE | RCIE | TXIE | SSPIE | CCP1IE | TMR2IE | TMR1IE | -000 0000 | -000 0000 |
| 94h | SSPSTAT | SMP | CKE | D/$\overline{A}$ | P | S | R/$\overline{W}$ | UA | BF | 0000 0000 | 0000 0000 |

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

**Note 1:** PIC16F687/PIC16F689/PIC16F690 only.

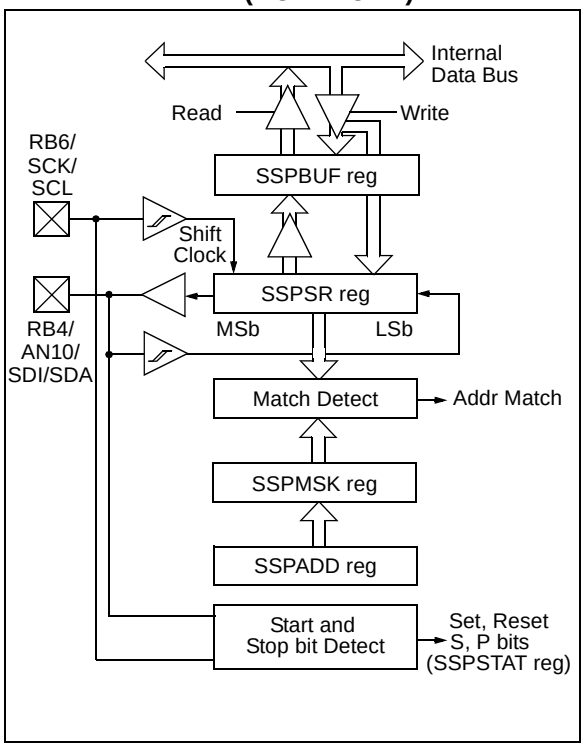
## 13.11 SSP I²C Operation

The SSP module in I²C mode, fully implements all slave functions, except general call support, and provides interrupts on Start and Stop bits in hardware to facilitate firmware implementations of the master functions. The SSP module implements the standard mode specifications, as well as 7-bit and 10-bit addressing.

Two pins are used for data transfer. These are the RB6/SCK/SCL pin, which is the clock (SCL), and the RB4/AN10/SDI/SDA pin, which is the data (SDA).

The SSP module functions are enabled by setting SSP enable bit SSPEN (SSPCON<5>).

**FIGURE 13-7: SSP BLOCK DIAGRAM (I²C™ MODE)**




The SSP module has six registers for the I²C operation, which are listed below.

- SSP Control register (SSPCON)
- SSP Status register (SSPSTAT)
- Serial Receive/Transmit Buffer (SSPBUF)
- SSP Shift register (SSPSR) – Not directly accessible
- SSP Address register (SSPADD)
- SSP Mask register (SSPMSK)

The SSPCON register allows control of the I²C operation. Four mode selection bits (SSPCON<3:0>) allow one of the following I²C modes to be selected:

- I²C Slave mode (7-bit address)
- I²C Slave mode (10-bit address)
- I²C Slave mode (7-bit address), with Start and Stop bit interrupts enabled to support Firmware Master mode
- I²C Slave mode (10-bit address), with Start and Stop bit interrupts enabled to support Firmware Master mode
- I²C Start and Stop bit interrupts enabled to support Firmware Master mode; Slave is idle

Selection of any I²C mode with the SSPEN bit set forces the SCL and SDA pins to be open drain, provided these pins are programmed to inputs by setting the appropriate TRISB bits. Pull-up resistors must be provided externally to the SCL and SDA pins for proper operation of the I²C module.

## 13.12 Slave Mode

In Slave mode, the SCL and SDA pins must be configured as inputs (TRISB<6,4> are set). The SSP module will override the input state with the output data when required (slave-transmitter).

When an address is matched, or the data transfer after an address match is received, the hardware automatically will generate the Acknowledge ($\overline{ACK}$) pulse, and then load the SSPBUF register with the received value currently in the SSPSR register.

There are certain conditions that will cause the SSP module not to give this $\overline{ACK}$ pulse. They include (either or both):

a) The buffer full bit BF (SSPSTAT<0>) was set before the transfer was received.

b) The overflow bit SSPOV (SSPCON<6>) was set before the transfer was received.

In this case, the SSPSR register value is not loaded into the SSPBUF, but bit SSPIF (PIR1<3>) is set. Table 13-3 shows the results of when a data transfer byte is received, given the status of bits BF and SSPOV. The shaded cells show the condition where user software did not properly clear the overflow condition. Flag bit BF is cleared by reading the SSPBUF register, while bit SSPOV is cleared through software.

The SCL clock input must have a minimum high and low for proper operation. For high and low times of the I²C specification, as well as the requirements of the SSP module, see **Section 17.0 "Electrical Specifications"**.

### 13.12.1 ADDRESSING

Once the SSP module has been enabled, it waits for a Start condition to occur. Following the Start condition, the 8-bits are shifted into the SSPSR register. All incoming bits are sampled with the rising edge of the clock (SCL) line. The value of register SSPSR<7:1> is compared to the value of the SSPADD register. The address is compared on the falling edge of the eighth clock (SCL) pulse. If the addresses match, and the BF and SSPOV bits are clear, the following events occur:

a) The SSPSR register value is loaded into the SSPBUF register.
b) The buffer full bit, BF is set.
c) An $\overline{\text{ACK}}$ pulse is generated.
d) SSP interrupt flag bit, SSPIF (PIR1<3>) is set (interrupt is generated if enabled) on the falling edge of the ninth SCL pulse.

In 10-bit Address mode, two address bytes need to be received by the slave (Figure 13-8). The five Most Significant bits (MSbs) of the first address byte specify if this is a 10-bit address. Bit R/$\overline{\text{W}}$ (SSPSTAT<2>) must specify a write so the slave device will receive the second address byte. For a 10-bit address, the first byte would equal '1111 0 A9 A8 0', where A9 and A8 are the two MSbs of the address.

The sequence of events for 10-bit address is as follows, with steps 7-9 for slave-transmitter:

1. Receive first (high) byte of address (bits SSPIF, BF and bit UA (SSPSTAT<1>) are set).
2. Update the SSPADD register with second (low) byte of address (clears bit UA and releases the SCL line).
3. Read the SSPBUF register (clears bit BF) and clear flag bit SSPIF.
4. Receive second (low) byte of address (bits SSPIF, BF and UA are set).
5. Update the SSPADD register with the first (high) byte of address; if match releases SCL line, this will clear bit UA.
6. Read the SSPBUF register (clears bit BF) and clear flag bit SSPIF.
7. Receive repeated Start condition.
8. Receive first (high) byte of address (bits SSPIF and BF are set).
9. Read the SSPBUF register (clears bit BF) and clear flag bit SSPIF.

**TABLE 13-3: DATA TRANSFER RECEIVED BYTE ACTIONS**

| Status Bits as Data Transfer is Received | | SSPSR → SSPBUF | Generate $\overline{\text{ACK}}$ Pulse | Set bit SSPIF (SSP Interrupt occurs if enabled) |
|---|---|---|---|---|
| BF | SSPOV | | | |
| 0 | 0 | Yes | Yes | Yes |
| 1 | 0 | No | No | Yes |
| 1 | 1 | No | No | Yes |
| 0 | 1 | No | No | Yes |

**Note:** Shaded cells show the conditions where the user software did not properly clear the overflow condition.

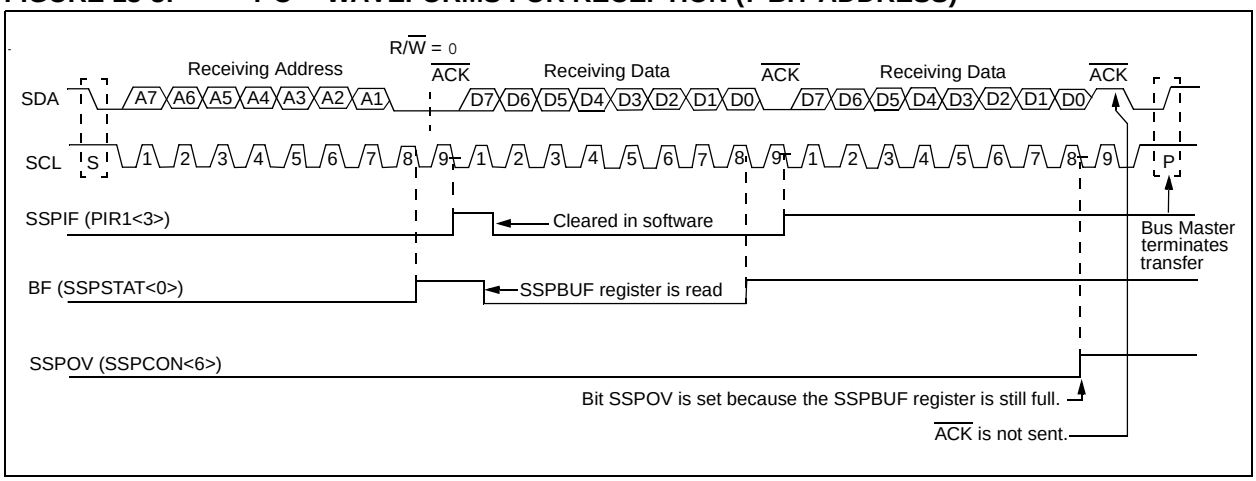
### 13.12.2 RECEPTION

When the R/$\overline{W}$ bit of the address byte is clear and an address match occurs, the R/$\overline{W}$ bit of the SSPSTAT register is cleared. The received address is loaded into the SSPBUF register.

When the address byte overflow condition exists, then no Acknowledge ($\overline{ACK}$) pulse is given. An overflow condition is defined as either bit BF (SSPSTAT<0>) is set, or bit SSPOV (SSPCON<6>) is set. This is an error condition due to the user's firmware.

An SSP interrupt is generated for each data transfer byte. Flag bit SSPIF (PIR1<3>) must be cleared in software. The SSPSTAT register is used to determine the status of the byte.

**FIGURE 13-8: I²C™ WAVEFORMS FOR RECEPTION (7-BIT ADDRESS)**



### 13.12.3 SSP MASK REGISTER

An SSP Mask (SSPMSK) register is available in $I^2C$ Slave mode as a mask for the value held in the SSPSR register during an address comparison operation. A zero ('0') bit in the SSPMSK register has the effect of making the corresponding bit in the SSPSR register a 'don't care'.

This register is reset to all '1's upon any Reset condition and, therefore, has no effect on standard SSP operation until written with a mask value.

This register must be initiated prior to setting SSPM<3:0> bits to select the $I^2C$ Slave mode (7-bit or 10-bit address).

This register can only be accessed when the appropriate mode is selected by bits (SSPM<3:0> of SSPCON).

The SSP Mask register is active during:

- 7-bit Address Mode: address compare of A<7:1>.
- 10-bit Address Mode: address compare of A<7:0> only. The SSP mask has no effect during the reception of the first (high) byte of the address.

**REGISTER 13-3: SSPMSK – SSP MASK REGISTER(1) (ADDRESS: 93h)**

| R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 |
|---|---|---|---|---|---|---|---|
| MSK7 | MSK6 | MSK5 | MSK4 | MSK3 | MSK2 | MSK1 | MSK0(2) |
| bit 7 | | | | | | | bit 0 |

bit 7-1 **MSK<7:1>:** Mask bits

1 = The received address bit n is compared to SSPADD<n> to detect $I^2C$ address match
0 = The received address bit n is not used to detect $I^2C$ address match

bit 0 **MSK<0>:** Mask bit for $I^2C$ Slave Mode, 10-bit Address(2)

$I^2C$ Slave Mode, 10-bit Address (SSPM<3:0> = 0111):
1 = The received address bit 0 is compared to SSPADD<0> to detect $I^2C$ address match
0 = The received address bit 0 is not used to detect $I^2C$ address match

**Note 1:** When SSPCON bits SSPM<3:0> = 1001, any reads or writes to the SSPADD SFR address are accessed through the SSPMSK register.

**2:** In all other SSP modes, this bit has no effect.

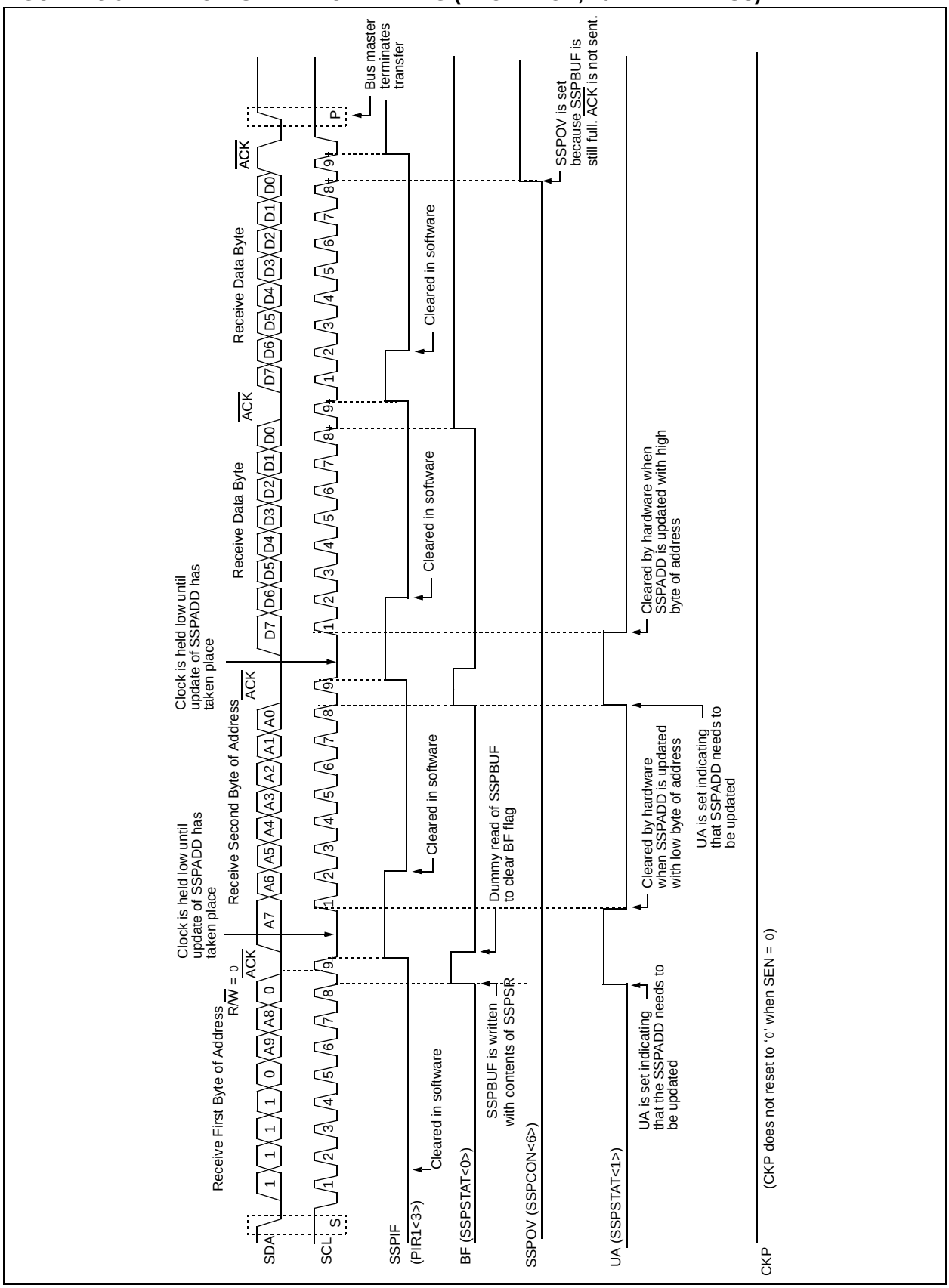
| **Legend:** | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' | |
| - n = Value at POR | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |

### 13.12.4 HALT ON ADDRESS DETECT

In some applications it is necessary to acknowledge multiple addresses or blocks of addresses. The Halt-on-Address-Detect feature allows software to check the address and perform validation.

Address Detect is enabled when the ADDEN bit of SSPCON1 register is set to '1'. The SSPIF flag and the CLKSTR bit are both set after the A1 (last bit of address) is clocked into the SSPSR and loaded into the SSPBUF, but before the address comparator result is read or the $\overline{ACK}$ pulse is generated. This allows the software to read the SSPBUF and validate the received address. If the address is determined to be valid, the software will copy the SSPBUF into SSPADD, thus setting the result of the comparator to true. The CLKSTR is then cleared ('0' written) the SSP engine is allowed to continue. Since the address compare is now true, an $\overline{ACK}$ pulse will be generated back to the master. If the software decides that the address is not to be acknowledged, the SSPADD is written with any value not equal to SSPBUF. This will set the compare to false and an $\overline{ACK}$ will be suppressed when CLKSTR is cleared.

**FIGURE 13-9: I²C™ SLAVE MODE TIMING (RECEPTION, 10-BIT ADDRESS)**



### 13.12.5 TRANSMISSION

When the R/$\overline{W}$ bit of the incoming address byte is set and an address match occurs, the R/$\overline{W}$ bit of the SSPSTAT register is set. The received address is loaded into the SSPBUF register. The $\overline{ACK}$ pulse will be sent on the ninth bit, and pin RB6/SCK/SCL is held low. The transmit data must be loaded into the SSPBUF register, which also loads the SSPSR register. Then, pin RB6/SCK/SCL should be enabled by setting bit CKP (SSPCON<4>). The master must monitor the SCL pin prior to asserting another clock pulse. The slave devices may be holding off the master by stretching the clock. The eight data bits are shifted out on the falling edge of the SCL input. This ensures that the SDA signal is valid during the SCL high time (Figure 13-10).

An SSP interrupt is generated for each data transfer byte. Flag bit SSPIF must be cleared in software, and the SSPSTAT register is used to determine the status of the byte. Flag bit SSPIF is set on the falling edge of the ninth clock pulse.

As a slave-transmitter, the $\overline{ACK}$ pulse from the master receiver is latched on the rising edge of the ninth SCL input pulse. If the SDA line was high (not $\overline{ACK}$), then the data transfer is complete. When the $\overline{ACK}$ is latched by the slave, the slave logic is reset (resets SSPSTAT register) and the slave then monitors for another occurrence of the Start bit. If the SDA line was low ($\overline{ACK}$), the transmit data must be loaded into the SSPBUF register, which also loads the SSPSR register. Then pin RB6/SCK/SCL should be enabled by setting bit CKP.

**FIGURE 13-10: I²C™ WAVEFORMS FOR TRANSMISSION (7-BIT ADDRESS)**

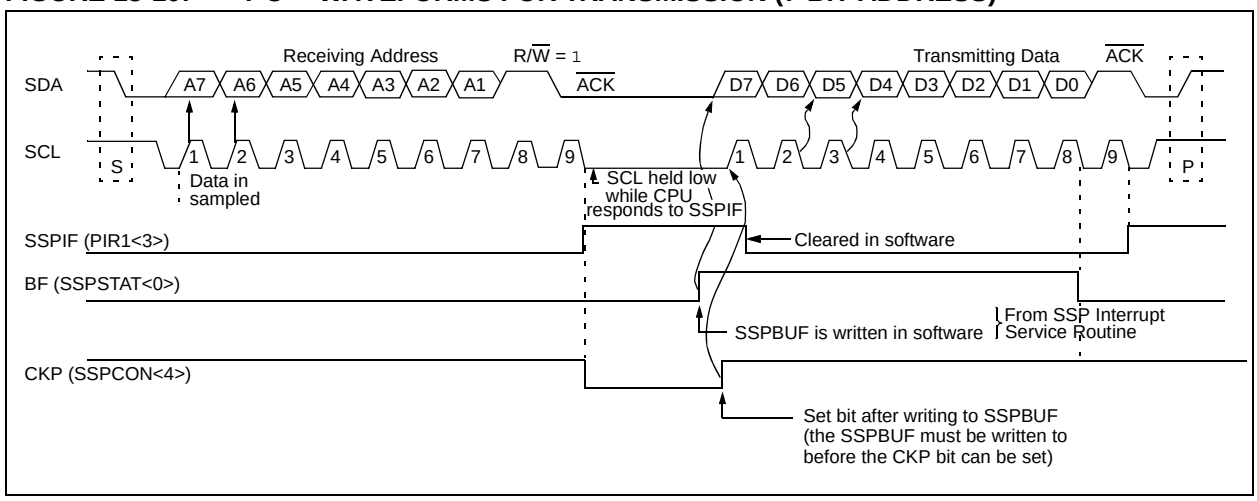
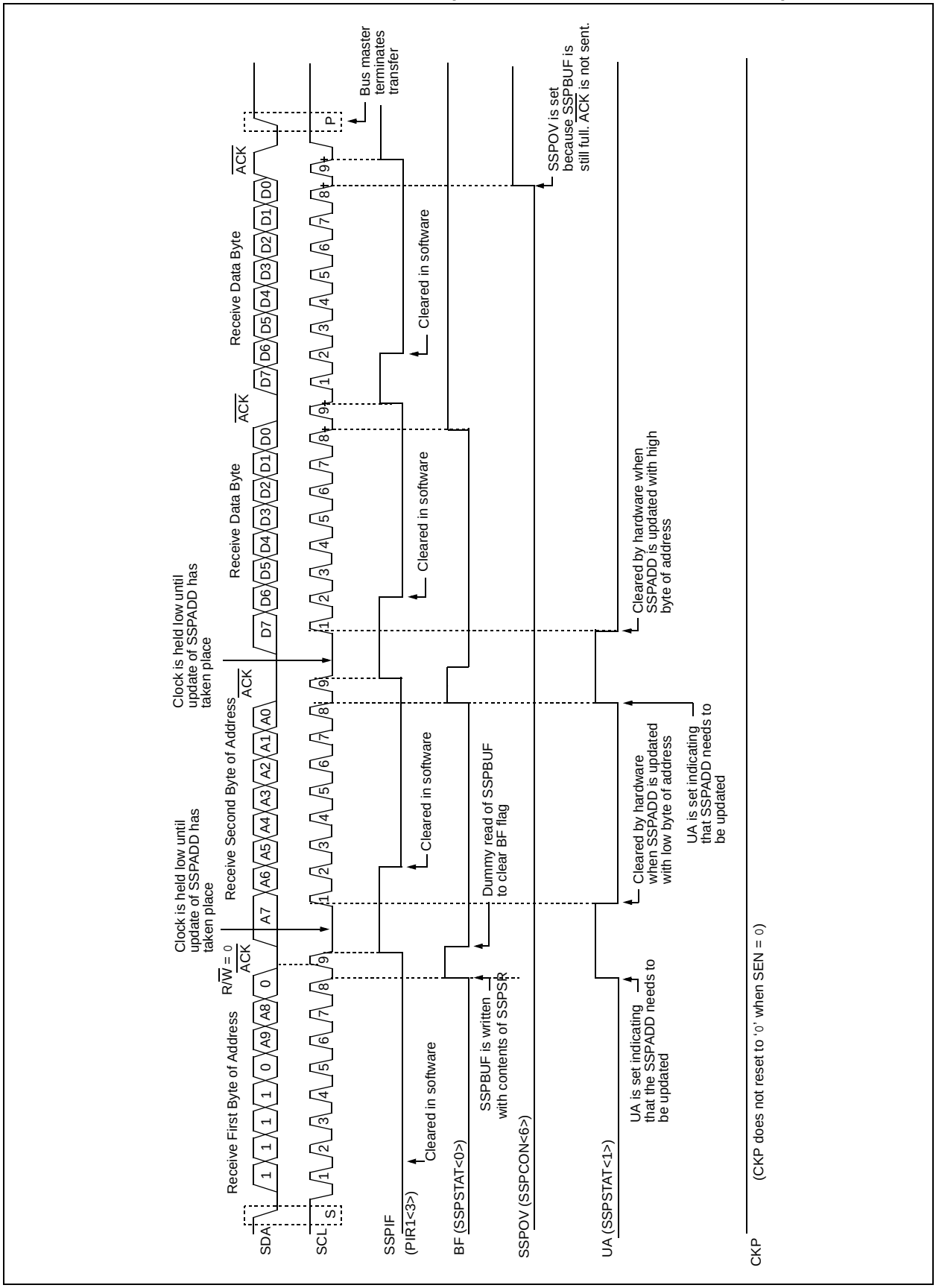


**FIGURE 13-11: $I^2C$™ SLAVE MODE TIMING (TRANSMISSION, 10-BIT ADDRESS)**



## 13.13 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 $I^2C$ 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 TRISB<6,4> bit(s). The output level is always low, irrespective of the value(s) in PORTB<6,4>. So when transmitting data, a '1' data bit must have the TRISB<4> bit set (input) and a '0' data bit must have the TRISB<4> bit cleared (output). The same scenario is true for the SCL line with the TRISB<6> bit. Pull-up resistors must be provided externally to the SCL and SDA pins for proper operation of the $I^2C$ module.

The following events will cause the 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 (SSPM<3:0> = 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.

## 13.14 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 $I^2C$ 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 TRISB<6,4>). 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 $\overline{\text{ACK}}$ pulse will be generated. If arbitration was lost during the data transfer stage, the device will need to re-transfer the data at a later time.

### 13.14.1 CLOCK SYNCHRONIZATION AND THE CKP BIT

When the CKP bit is cleared, the SCL output is forced to '0'; however, setting the CKP bit will not assert the SCL output low until the SCL output is already sampled low. Therefore, the CKP bit will not assert the SCL line until an external $I^2C$ master device has already asserted the SCL line. The SCL output will remain low until the CKP bit is set and all other devices on the $I^2C$ bus have deasserted SCL. This ensures that a write to the CKP bit will not violate the minimum high time requirement for SCL (see Figure 13-12).

**FIGURE 13-12: CLOCK SYNCHRONIZATION TIMING**

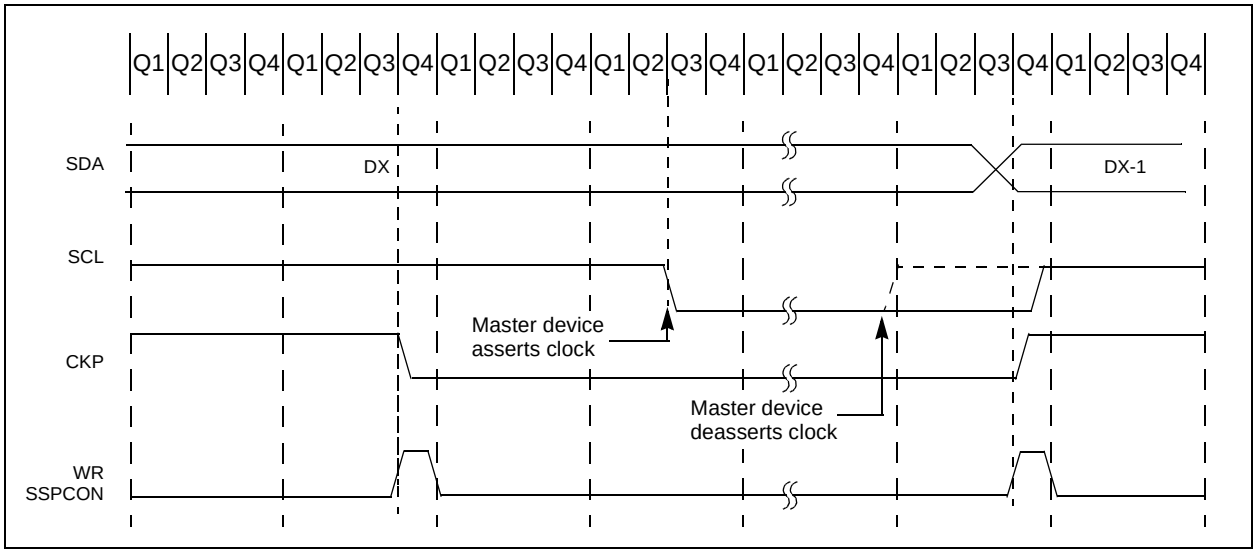



**TABLE 13-4: REGISTERS ASSOCIATED WITH I²C™ OPERATION(1)**

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0Bh/8Bh/ 10Bh/18Bh | INTCON | GIE | PEIE | T0IE | INTE | RABIE | T0IF | INTF | RABIF | 0000 000x | 0000 000x |
| 0Ch | PIR1 | — | ADIF | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | -000 0000 | -000 0000 |
| 13h | SSPBUF | Synchronous Serial Port Receive Buffer/Transmit Register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 14h | SSPCON | WCOL | SSPOV | SSPEN | CKP | SSPM3 | SSPM2 | SSPM1 | SSPM0 | 0000 0000 | 0000 0000 |
| 86h | TRISB | TRISB7 | TRISB6 | TRISB5 | TRISB4 | — | — | — | — | 1111 ---- | 1111 ---- |
| 93h | SSPMSK(2) | MSK7 | MSK6 | MSK5 | MSK4 | MSK3 | MSK2 | MSK1 | MSK0 | 1111 1111 | 1111 1111 |
| 94h | SSPSTAT | SMP(3) | CKE(3) | D/$\overline{A}$ | P | S | R/$\overline{W}$ | UA | BF | 0000 0000 | 0000 0000 |
| 8Ch | PIE1 | — | ADIE | RCIE | TXIE | SSPIE | CCP1IE | TMR2IF | TMR1IF | -000 0000 | -000 0000 |

**Legend:** – = Unimplemented locations, read as '0', u = unchanged, x = unknown. Shaded cells are not used by the SSP module.

**Note 1:** PIC16F687/PIC16F689/PIC16F690 only.

**2:** SSPMSK register (Register 13-3) can be accessed by reading or writing to SSPADD register with bits SSPM<3:0> = 1001. See Registers 13-2 and 13-3 for more details.

**3:** Maintain these bits clear.

## 14.0 SPECIAL FEATURES OF THE CPU

The PIC16F685/687/689/690 have a host of features intended to maximize system reliability, minimize cost through elimination of external components, provide power saving features and offer code protection.

These features are:

- Reset
  - Power-on Reset (POR)
  - Power-up Timer (PWRT)
  - Oscillator Start-up Timer (OST)
  - Brown-out Reset (BOR)
- Interrupts
- Watchdog Timer (WDT)
- Oscillator selection
- Sleep
- Code protection
- ID Locations
- In-Circuit Serial Programming

The PIC16F685/687/689/690 have 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 of 64 ms (nominal) on power-up only, designed to keep the part in Reset while the power supply stabilizes. There is also circuitry to reset the device if a brown-out occurs, which can use the Power-up Timer to provide at least a 64 ms Reset. With these three functions-on-chip, most applications need no external Reset circuitry.

The 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
- An interrupt

Several oscillator options are also made available to allow the part to fit the application. The INTOSC option saves system cost while the LP crystal option saves power. A set of configuration bits are used to select various options (see Register 14-1).

## 14.1 Configuration Bits

The configuration bits can be programmed (read as '0'), or left unprogrammed (read as '1') to select various device configurations as shown in Register 14-1. These bits are mapped in program memory location 2007h.

**Note:** Address 2007h is beyond the user program memory space. It belongs to the special configuration memory space (2000h-3FFFh), which can be accessed only during programming. See *"PIC12F6XX/16F6XX Memory Programming Specification"* (DS41204) for more information.

**REGISTER 14-1: CONFIG – CONFIGURATION WORD (ADDRESS: 2007h)**

| Reserved | Reserved | FCMEN | IESO | BOREN1(1) | BOREN0(1) | $\overline{CPD}$(2) | $\overline{CP}$(3) | MCLRE(4) | $\overline{PWRTE}$ | WDTE | FOSC2 | FOSC1 | FOSC0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| bit 13 | | | | | | | | | | | | | bit 0 |

bit 13-12 **Reserved:** Reserved bits. Do Not Use.

bit 11 **FCMEN:** Fail-Safe Clock Monitor Enabled bit
1 = Fail-Safe Clock Monitor is enabled
0 = Fail-Safe Clock Monitor is disabled

bit 10 **IESO:** Internal External Switchover bit
1 = Internal External Switchover mode is enabled
0 = Internal External Switchover mode is disabled

bit 9-8 **BOREN<1:0>:** Brown-out Reset Selection bits(1)
11 = BOR enabled
10 = BOR enabled during operation and disabled in Sleep
01 = BOR controlled by SBOREN bit (PCON<4>)
00 = BOR disabled

bit 7 **$\overline{CPD}$:** Data Code Protection bit(2)
1 = Data memory code protection is disabled
0 = Data memory code protection is enabled

bit 6 **$\overline{CP}$:** Code Protection bit(3)
1 = Program memory code protection is disabled
0 = Program memory code protection is enabled

bit 5 **MCLRE:** RA3/$\overline{MCLR}$/$V_{PP}$ pin function select bit(4)
1 = RA3/$\overline{MCLR}$/$V_{PP}$ pin function is $\overline{MCLR}$
0 = RA3/$\overline{MCLR}$/$V_{PP}$ pin function is digital input, $\overline{MCLR}$ internally tied to $V_{DD}$

bit 4 **$\overline{PWRTE}$:** Power-up Timer Enable bit
1 = PWRT disabled
0 = PWRT enabled

bit 3 **WDTE:** Watchdog Timer Enable bit
1 = WDT enabled
0 = WDT disabled and can be enabled by SWDTEN bit (WDTCON<0>)

bit 2-0 **FOSC<2:0>:** Oscillator Selection bits
111 = RC oscillator: CLKOUT function on RA4/AN3/$\overline{T1G}$/OSC2/CLKOUT pin, RC on RA5/T1CKI/OSC1/CLKIN
110 = RCIO oscillator: I/O function on RA4/AN3/$\overline{T1G}$/OSC2/CLKOUT pin, RC on RA5/T1CKI/OSC1/CLKIN
101 = INTOSC oscillator: CLKOUT function on RA4/AN3/$\overline{T1G}$/OSC2/CLKOUT pin, I/O function on RA5/T1CKI/OSC1/CLKIN
100 = INTOSCIO oscillator: I/O function on RA4/AN3/$\overline{T1G}$/OSC2/CLKOUT pin, I/O function on RA5/T1CKI/OSC1/CLKIN
011 = EC: I/O function on RA4/AN3/$\overline{T1G}$/OSC2/CLKOUT pin, CLKIN on RA5/T1CKI/OSC1/CLKIN
010 = HS oscillator: High-speed crystal/resonator on RA4/AN3/$\overline{T1G}$/OSC2/CLKOUT and RA5/T1CKI/OSC1/CLKIN
001 = XT oscillator: Crystal/resonator on RA4/AN3/$\overline{T1G}$/OSC2/CLKOUT and RA5/T1CKI/OSC1/CLKIN
000 = LP oscillator: Low-power crystal on RA4/AN3/$\overline{T1G}$/OSC2/CLKOUT and RA5/T1CKI/OSC1/CLKIN

**Note 1:** Enabling Brown-out Reset does not automatically enable Power-up Timer.
**2:** The entire data EEPROM will be erased when the code-protect is turned off.
**3:** The entire program memory will be erased when non code-protect is turned off.
**4:** When $\overline{MCLR}$ is asserted in INTOSC or RC mode, the internal clock oscillator is disabled.

| **Legend:** | | | |
|---|---|---|---|
| R = Readable | 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 |

## 14.2 Reset

The PIC16F685/687/689/690 differentiates between various kinds of Reset:

a) Power-on Reset (POR)
b) WDT Reset during normal operation
c) WDT Reset during Sleep
d) $\overline{\text{MCLR}}$ Reset during normal operation
e) $\overline{\text{MCLR}}$ Reset during Sleep
f) 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:

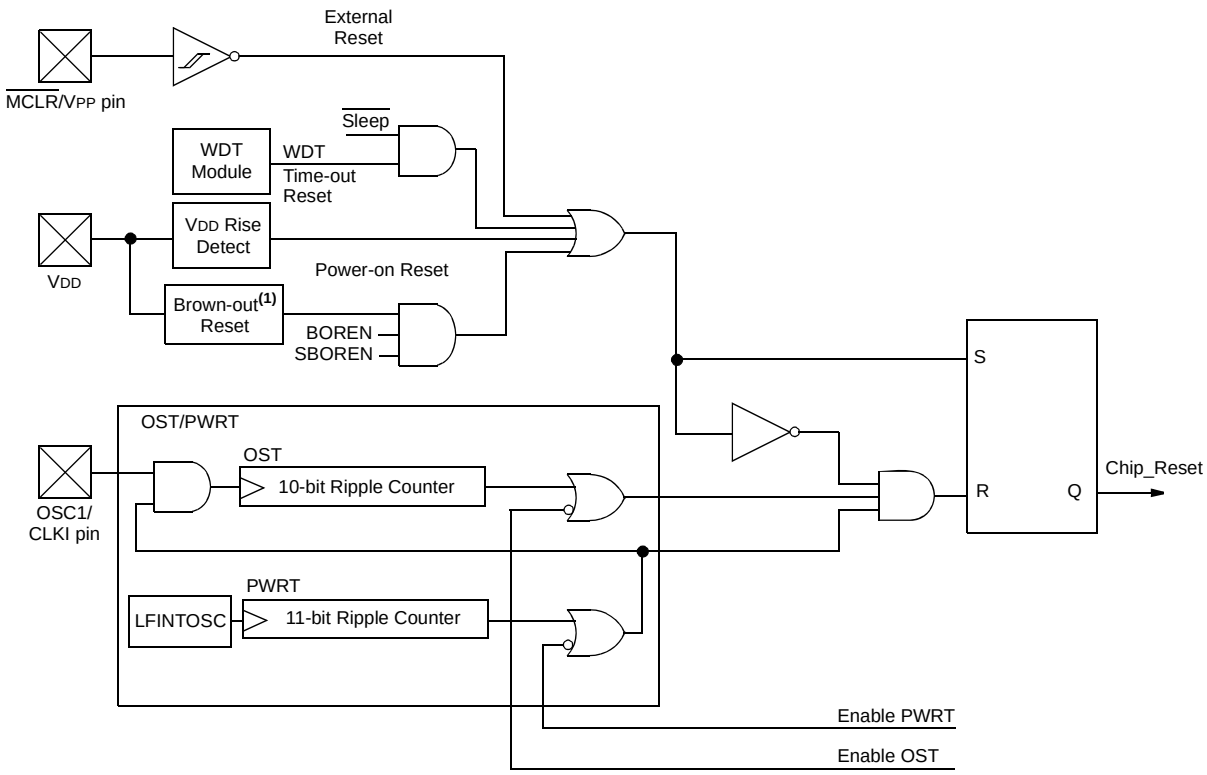
- Power-on Reset
- $\overline{\text{MCLR}}$ Reset
- $\overline{\text{MCLR}}$ Reset during Sleep
- WDT Reset
- Brown-out Reset (BOR)

They are not affected by a WDT wake-up since this is viewed as the resumption of normal operation. $\overline{\text{TO}}$ and $\overline{\text{PD}}$ bits are set or cleared differently in different Reset situations, as indicated in Table 14-2. These bits are used in software to determine the nature of the Reset. See Table 14-4 for a full description of Reset states of all registers.

A simplified block diagram of the On-Chip Reset Circuit is shown in Figure 14-1.

The $\overline{\text{MCLR}}$ Reset path has a noise filter to detect and ignore small pulses. See **Section 17.0 "Electrical Specifications"** for pulse-width specifications.

**FIGURE 14-1: SIMPLIFIED BLOCK DIAGRAM OF ON-CHIP RESET CIRCUIT**




**Note 1:** Refer to the Configuration Word register (Register 14-1).

### 14.2.1 POWER-ON RESET (POR)

The on-chip POR circuit holds the chip in Reset until VDD has reached a high enough level for proper operation. A maximum rise time for VDD is required. See **Section 17.0 "Electrical Specifications"** for details. If the BOR is enabled, the maximum rise time specification does not apply. The BOR circuitry will keep the device in Reset until VDD reaches VBOR (see **Section 14.2.4 "Brown-Out Reset (BOR)"**).

| | |
|---|---|
| **Note:** | The POR circuit does not produce an internal Reset when VDD declines. To re-enable the POR, VDD must reach Vss for a minimum of 100 µs. |

When the device starts normal operation (exits the Reset condition), device operating parameters (i.e., voltage, frequency, temperature, etc.) 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).

### 14.2.2 $\overline{\text{MCLR}}$

PIC16F685/687/689/690 has 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 early 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 14-2, is suggested.

An internal $\overline{\text{MCLR}}$ option is enabled by clearing the $\overline{\text{MCLRE}}$ bit in the Configuration Word register. When $\overline{\text{MCLRE}}$ = 0, the Reset signal to the chip is generated internally. When the $\overline{\text{MCLRE}}$ = 1, the RA3/$\overline{\text{MCLR}}$ pin becomes an external Reset input. In this mode, the RA3/$\overline{\text{MCLR}}$ pin has a weak pull-up to VDD.

### 14.2.3 POWER-UP TIMER (PWRT)

The Power-up Timer provides a fixed 64 ms (nominal) time-out on power-up only, from POR or Brown-out Reset. The Power-up Timer operates from the 31 kHz LFINTOSC oscillator. For more information, see **Section 3.4 "Internal Clock Modes"**. The chip is kept in Reset as long as PWRT is active. The PWRT delay allows the VDD to rise to an acceptable level. A configuration bit, $\overline{\text{PWRTE}}$, can disable (if set) or enable (if cleared or programmed) the Power-up Timer. The Power-up Timer should be enabled when Brown-out Reset is enabled, although it is not required.

The Power-up Timer delay will vary from chip-to-chip and vary due to:

- VDD variation
- Temperature variation
- Process variation

See DC parameters for details (**Section 17.0 "Electrical Specifications"**).

### 14.2.4 BROWN-OUT RESET (BOR)

The BOREN0 and BOREN1 bits in the Configuration Word register select one of four BOR modes. Two modes have been added to allow software or hardware control of the BOR enable. When BOREN<1:0> = 01, the SBOREN bit (PCON<4>) enables/disables the BOR allowing it to be controlled in software. By selecting BOREN<1:0>, the BOR is automatically disabled in Sleep to conserve power and enabled on wake-up. In this mode, the SBOREN bit is disabled. See Register 14-1 for the Configuration Word definition.

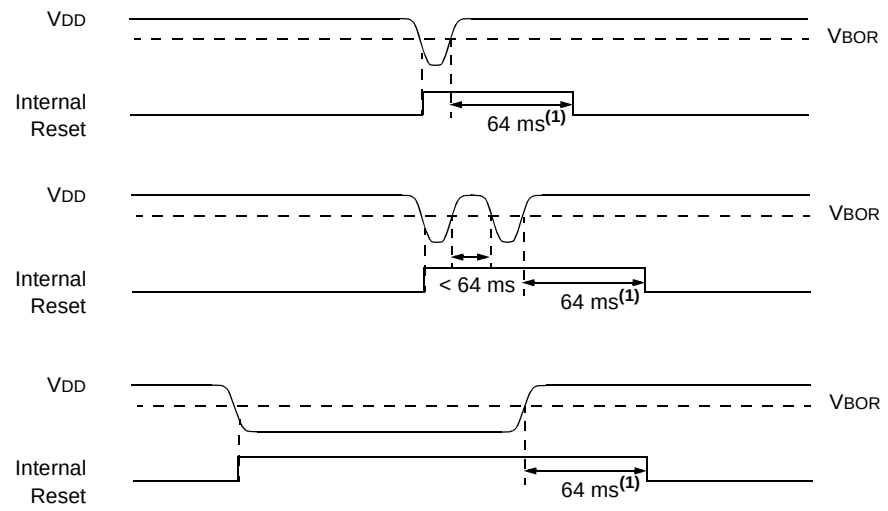
If VDD falls below VBOR for greater than parameter (TBOR) (see **Section 17.0 "Electrical Specifications"**), the Brown-out situation will reset the device. This will occur regardless of VDD slew rate. A Reset is not insured to occur if VDD falls below VBOR for less than parameter (TBOR).

On any Reset (Power-on, Brown-out Reset, Watchdog Timer, etc.), the chip will remain in Reset until VDD rises above VBOR (see Figure 14-2). The Power-up Timer will now be invoked, if enabled and will keep the chip in Reset an additional 64 ms.

| **Note:** | The Power-up Timer is enabled by the $\overline{\text{PWRTE}}$ bit in the Configuration Word register. |
|---|---|

If VDD drops below VBOR 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 VDD rises above VBOR, the Power-up Timer will execute a 64 ms Reset.

**FIGURE 14-2: BROWN-OUT SITUATIONS**




**Note 1:** 64 ms delay only if $\overline{\text{PWRTE}}$ bit is programmed to '0'.

### 14.2.5 TIME-OUT SEQUENCE

On power-up, the time-out sequence is as follows: first, PWRT time-out is invoked after POR has expired, then OST is activated after the PWRT time-out has expired. The total time-out will vary based on oscillator configuration and PWRTE bit status. For example, in EC mode with PWRTE bit erased (PWRT disabled), there will be no time-out at all. Figures 14-3, 14-4 and 14-5 depict time-out sequences. The device can execute code from the INTOSC while OST is active by enabling Two-Speed Start-up or Fail-Safe Monitor (see **Section 3.6.2 "Two-Speed Start-up Sequence"** and **Section 3.7 "Fail-Safe Clock Monitor"**).

Since the time-outs occur from the POR pulse, if MCLR is kept low long enough, the time-outs will expire. Then, bringing MCLR high will begin execution immediately (see Figure 14-4). This is useful for testing purposes or to synchronize more than one PIC16F685/687/689/690 device operating in parallel.

Table 14-5 shows the Reset conditions for some special registers, while Table 14-4 shows the Reset conditions for all the registers.

### 14.2.6 POWER CONTROL (PCON) REGISTER

The Power Control register PCON (address 8Eh) has two Status bits to indicate what type of Reset that last occurred.

Bit 0 is BOR (Brown-out Reset). BOR is unknown on Power-on Reset. It must then be set by the user and checked on subsequent Resets to see if BOR = 0, indicating that a Brown-out has occurred. The BOR Status bit is a "don't care" and is not necessarily predictable if the brown-out circuit is disabled (BOREN<1:0> = 00 in the Configuration Word register).

Bit 1 is POR (Power-on Reset). It is a '0' on Power-on Reset and unaffected otherwise. The user must write a '1' to this bit following a Power-on Reset. On a subsequent Reset, if POR is '0', it will indicate that a Power-on Reset has occurred (i.e., $V_{DD}$ may have gone too low).

For more information, see **Section 4.2.3 "Ultra Low-Power Wake-up"** and **Section 14.2.4 "Brown-Out Reset (BOR)"**.

**TABLE 14-1: TIME-OUT IN VARIOUS SITUATIONS**

| Oscillator Configuration | Power-up | | Brown-out Reset | | Wake-up from Sleep |
|---|---|---|---|---|---|
| | $\overline{PWRTE}$ = 0 | $\overline{PWRTE}$ = 1 | $\overline{PWRTE}$ = 0 | $\overline{PWRTE}$ = 1 | |
| XT, HS, LP | TPWRT + 1024 • TOSC | 1024 • TOSC | TPWRT + 1024 • TOSC | 1024 • TOSC | 1024 • TOSC |
| LP, T1OSCIN = 1 | TPWRT | — | TPWRT | — | — |
| RC, EC, INTOSC | TPWRT | — | TPWRT | — | — |

**TABLE 14-2: STATUS/PCON BITS AND THEIR SIGNIFICANCE**

| POR | BOR | $\overline{TO}$ | $\overline{PD}$ | Condition |
|---|---|---|---|---|
| 0 | u | 1 | 1 | Power-on Reset |
| 1 | 0 | 1 | 1 | Brown-out Reset |
| u | u | 0 | u | WDT Reset |
| u | u | 0 | 0 | WDT Wake-up |
| u | u | u | u | $\overline{MCLR}$ Reset during normal operation |
| u | u | 1 | 0 | $\overline{MCLR}$ Reset during Sleep |

**Legend:** u = unchanged, x = unknown

**TABLE 14-3: SUMMARY OF REGISTERS ASSOCIATED WITH BROWN-OUT**

| 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(1) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 03h/83h/ 103h/183h | STATUS | IRP | RP1 | RPO | $\overline{TO}$ | $\overline{PD}$ | Z | DC | C | 0001 1xxx | 000q quuu |
| 8Eh | PCON | — | — | ULPWUE | SBOREN | — | — | $\overline{POR}$ | $\overline{BOR}$ | --01 --qq | --0u --uu |

**Legend:** u = unchanged, x = unknown, – = unimplemented bit, reads as '0', q = value depends on condition. Shaded cells are not used by BOR.
**Note 1:** Other (non Power-up) Resets include MCLR Reset and Watchdog Timer Reset during normal operation.

**FIGURE 14-3: TIME-OUT SEQUENCE ON POWER-UP (DELAYED $\overline{MCLR}$): CASE 1**

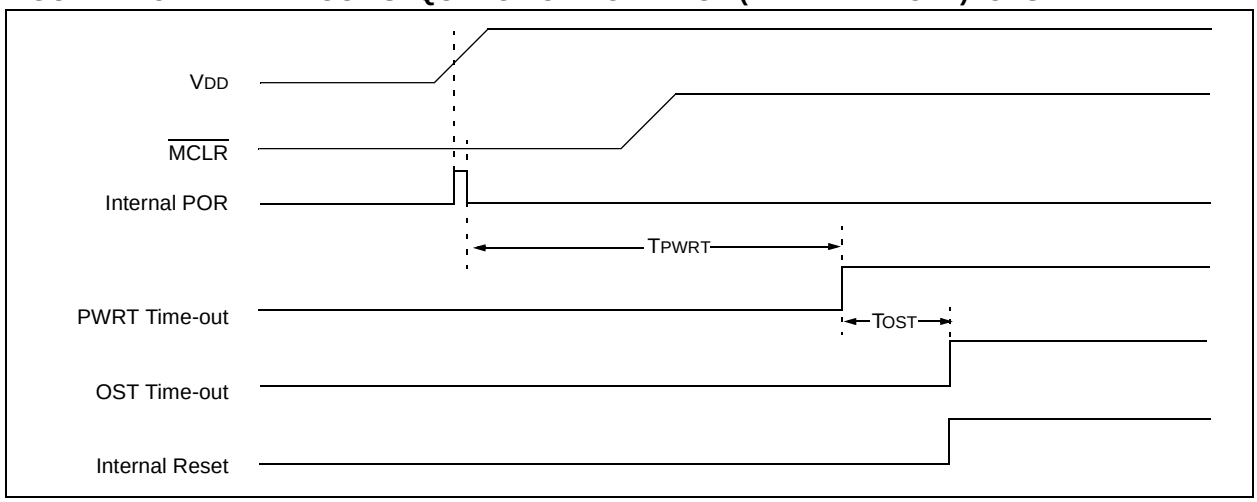



**FIGURE 14-4: TIME-OUT SEQUENCE ON POWER-UP (DELAYED $\overline{MCLR}$): CASE 2**

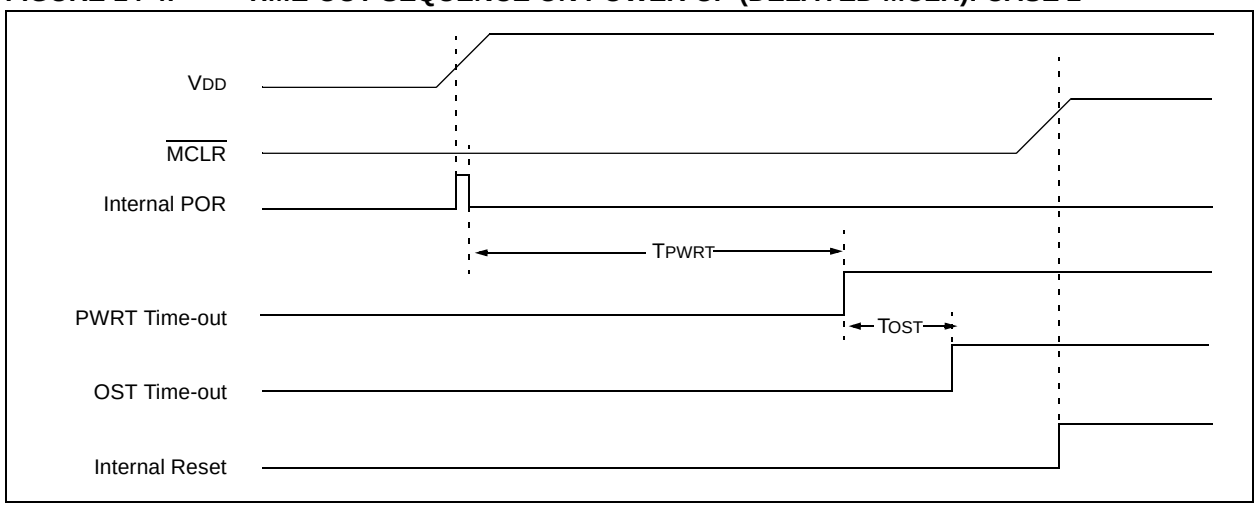



**FIGURE 14-5: TIME-OUT SEQUENCE ON POWER-UP ($\overline{MCLR}$ WITH VDD)**

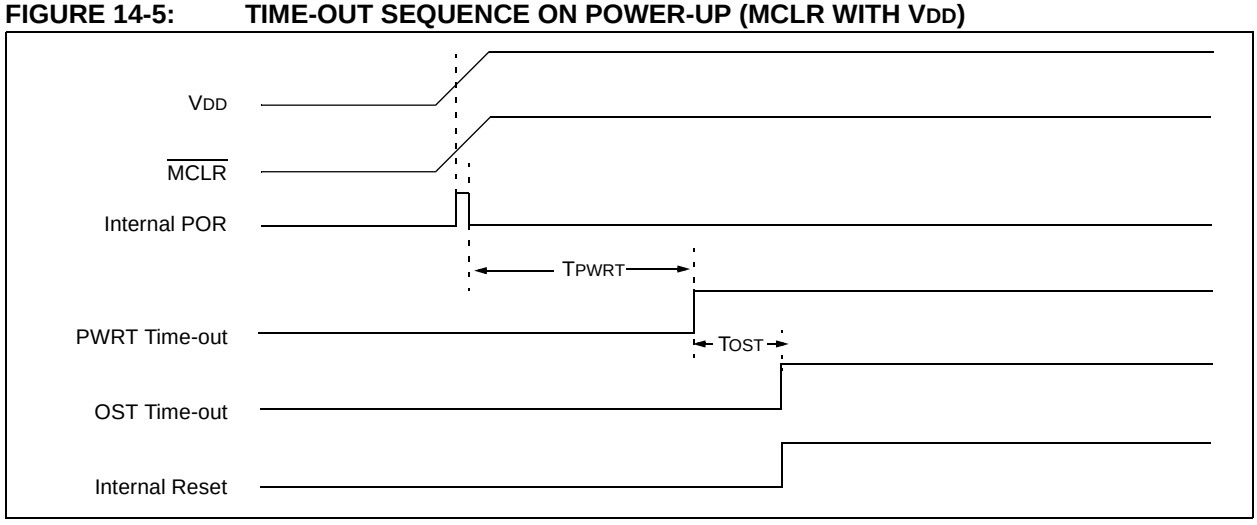


**TABLE 14-4: INITIALIZATION CONDITION FOR REGISTER**

| Register | Address | Power-on Reset | $\overline{\text{MCLR}}$ Reset<br>WDT Reset<br>Brown-out Reset(1) | Wake-up from Sleep through Interrupt<br>Wake-up from Sleep through WDT Time-out |
|---|---|---|---|---|
| W | — | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| INDF | 00h/80h/ 100h/180h | xxxx xxxx | xxxx xxxx | uuuu uuuu |
| TMR0 | 01h/101h | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| PCL | 02h/82h/ 102h/182h | 0000 0000 | 0000 0000 | PC + 1(3) |
| STATUS | 03h/83h/ 103h/183h | 0001 1xxx | 000q quuu(4) | uuuq quuu(4) |
| FSR | 04h/84h/ 104h184h | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| PORTA | 05h/105h | --xx xxxx | --00 0000 | --uu uuuu |
| PORTB | 06h/106h | xxxx ---- | 0000 ---- | uuuu ---- |
| PORTC | 07h/107h | xxxx xxxx | 0000 0000 | uuuu uuuu |
| PCLATH | 0Ah/8Ah/ 10Ah/18Ah | ---0 0000 | ---0 0000 | ---u uuuu |
| INTCON | 0Bh/8Bh/ 10Bh/18Bh | 0000 000x | 0000 000x | uuuu uuuu(2) |
| PIR1 | 0Ch | -000 0000 | -000 0000 | -uuu uuuu(2) |
| PIR2 | 0Dh | 0000 ---- | 0000 ---- | uuuu ----(2) |
| TMR1L | 0Eh | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| TMR1H | 0Fh | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| T1CON | 10h | 0000 0000 | uuuu uuuu | uuuu uuuu |
| TMR2 | 11h | 0000 0000 | 0000 0000 | uuuu uuuu |
| T2CON | 12h | -000 0000 | -000 0000 | -uuu uuuu |
| SSPBUF | 13h | xxxx xxxx | xxxx xxxx | uuuu uuuu |
| SSPCON | 14h | 0000 0000 | 0000 0000 | uuuu uuuu |
| CCPR1L | 15h | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| CCPR1H | 16h | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| CCP1CON | 17h | 0000 0000 | 0000 0000 | uuuu uuuu |
| RCSTA | 18h | 0000 000x | 0000 000x | uuuu uuuu |
| TXREG | 19h | 0000 0000 | 0000 0000 | uuuu uuuu |
| RCREG | 1Ah | 0000 0000 | 0000 0000 | uuuu uuuu |
| PWM1CON | 1Ch | 0000 0000 | 0000 0000 | uuuu uuuu |
| ECCPAS | 1Dh | 0000 0000 | 0000 0000 | uuuu uuuu |
| ADRESH | 1Eh | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| ADCON0 | 1Fh | 0000 0000 | 0000 0000 | uuuu uuuu |
| OPTION_REG | 81h/181h | 1111 1111 | 1111 1111 | uuuu uuuu |
| TRISA | 85h/185h | --11 1111 | --11 1111 | --uu uuuu |

**Legend:** u = unchanged, x = unknown, – = unimplemented bit, reads as '0', q = value depends on condition.

**Note 1:** If VDD goes too low, Power-on Reset will be activated and registers will be affected differently.

**2:** One or more bits in INTCON and/or PIR1 will be affected (to cause wake-up).

**3:** When the wake-up is due to an interrupt and the GIE bit is set, the PC is loaded with the interrupt vector (0004h).

**4:** See Table 14-5 for Reset value for specific condition.

**5:** If Reset was due to brown-out, then bit 0 = 0. All other Resets will cause bit 0 = u.

**6:** Accessible only when SSPM<3:0> = 1001.

**TABLE 14-4: INITIALIZATION CONDITION FOR REGISTER (CONTINUED)**

| Register | Address | Power-on Reset | $\overline{\text{MCLR}}$ Reset<br>WDT Reset (Continued)<br>Brown-out Reset(1) | Wake-up from Sleep through Interrupt<br>Wake-up from Sleep through WDT Time-out |
|---|---|---|---|---|
| TRISB | 86h/186h | 1111 ---- | 1111 ---- | uuuu ---- |
| TRISC | 87h/187h | 1111 1111 | 1111 1111 | uuuu uuuu |
| PIE1 | 8Ch | -000 0000 | -000 0000 | -uuu uuuu |
| PIE2 | 8Dh | 0000 ---- | 0000 ---- | uuuu uuuu |
| PCON | 8Eh | --01 --qq | --0u --uu(1, 5) | --uu --uu |
| OSCCON | 8Fh | -110 q000 | -110 x000 | -uuu uuuu |
| OSCTUNE | 90h | ---0 0000 | ---u uuuu | ---u uuuu |
| PR2 | 92h | 1111 1111 | 1111 1111 | 1111 1111 |
| SSPADD | 93h | 0000 0000 | 1111 1111 | uuuu uuuu |
| SSPMSK(6) | 93h | ---- ---- | 1111 1111 | uuuu uuuu |
| SSPSTAT | 94h | 0000 0000 | 1111 1111 | uuuu uuuu |
| WPUA | 95h | --11 -111 | --11 -111 | uuuu uuuu |
| IOCA | 96h | --00 0000 | --00 0000 | --uu uuuu |
| WDTCON | 97h | ---0 1000 | ---0 1000 | ---u uuuu |
| TXSTA | 98h | 0000 0010 | 0000 0010 | uuuu uuuu |
| SPBRG | 99h | 0000 0000 | 0000 0000 | uuuu uuuu |
| SPBRGH | 9Ah | 0000 0000 | 0000 0000 | uuuu uuuu |
| BAUDCTL | 9Bh | 01-0 0-00 | 01-0 0-00 | uu-u u-uu |
| ADRESL | 9Eh | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| ADCON1 | 9Fh | -000 ---- | -000 ---- | -uuu ---- |
| EEDAT | 10Ch | 0000 0000 | 0000 0000 | uuuu uuuu |
| EEADR | 10Dh | 0000 0000 | 0000 0000 | uuuu uuuu |
| EEDATH | 10Eh | --00 0000 | --00 0000 | --uu uuuu |
| EEADRH | 10Fh | ---- 0000 | ---- 0000 | ---- uuuu |
| WPUB | 115h | 1111 ---- | 1111 ---- | uuuu ---- |
| IOCB | 116h | 0000 ---- | 0000 ---- | uuuu ---- |
| VRCON | 118h | 0000 0000 | 0000 0000 | uuuu uuuu |
| CM1CON0 | 119h | 0000 -000 | 0000 -000 | uuuu -uuu |
| CM2CON0 | 11Ah | 0000 -000 | 0000 -000 | uuuu -uuu |
| CM2CON1 | 11Bh | 00-- --00 | 00-- --10 | uu-- --uu |
| ANSEL | 11Eh | 1111 1111 | 1111 1111 | uuuu uuuu |
| ANSELH | 11Fh | ---- 1111 | ---- 1111 | ---- uuuu |
| EECON1 | 18Ch | x--- x000 | u--- q000 | ---- uuuu |
| EECON2 | 18Dh | ---- ---- | ---- ---- | ---- ---- |
| PSTRCON | 19Dh | ---0 0001 | ---0 0001 | ---u uuuu |
| SRCON | 19EH | 0000 00-- | 0000 00-- | uuuu uu-- |

**Legend:** u = unchanged, x = unknown, – = unimplemented bit, reads as '0', q = value depends on condition.

**Note 1:** If VDD goes too low, Power-on Reset will be activated and registers will be affected differently.

**2:** One or more bits in INTCON and/or PIR1 will be affected (to cause wake-up).

**3:** When the wake-up is due to an interrupt and the GIE bit is set, the PC is loaded with the interrupt vector (0004h).

**4:** See Table 14-5 for Reset value for specific condition.

**5:** If Reset was due to brown-out, then bit 0 = 0. All other Resets will cause bit 0 = u.

**6:** Accessible only when SSPM<3:0> = 1001.

**TABLE 14-5: INITIALIZATION CONDITION FOR SPECIAL REGISTERS**

| Condition | Program Counter | Status Register | PCON Register |
|---|---|---|---|
| Power-on Reset | 000h | 0001 1xxx | --01 --0x |
| $\overline{\text{MCLR}}$ Reset during normal operation | 000h | 000u uuuu | --0u --uu |
| $\overline{\text{MCLR}}$ Reset during Sleep | 000h | 0001 0uuu | --0u --uu |
| WDT Reset | 000h | 0000 uuuu | --0u --uu |
| WDT Wake-up | PC + 1 | uuu0 0uuu | --uu --uu |
| Brown-out Reset | 000h | 0001 1uuu | --01 --10 |
| Interrupt Wake-up from Sleep | PC + 1(1) | uuu1 0uuu | --uu --uu |

**Legend:** u = unchanged, x = unknown, – = unimplemented bit, reads as '0'.

**Note 1:** When the wake-up is due to an interrupt and Global Interrupt Enable bit, GIE, is set, the PC is loaded with the interrupt vector (0004h) after execution of PC + 1.

## 14.3 Interrupts

The PIC16F685/687/689/690 have multiple sources of interrupt:

- External Interrupt RA2/INT
- TMR0 Overflow Interrupt
- PORTA/PORTB Change Interrupts
- 2 Comparator Interrupts
- A/D Interrupt
- Timer1 Overflow Interrupt
- Timer2 Match Interrupt
- EEPROM Data Write Interrupt
- Fail-Safe Clock Monitor Interrupt
- Enhanced CCP Interrupt
- EUSART Receive and Transmit interrupts

The Interrupt Control register (INTCON) and Peripheral Interrupt Request Register 1 (PIR1) record individual interrupt requests in flag bits. The INTCON register also has individual and global interrupt enable bits.

A Global Interrupt Enable bit, GIE (INTCON<7>), enables (if set) all unmasked interrupts, or disables (if cleared) all interrupts. Individual interrupts can be disabled through their corresponding enable bits in the INTCON, PIE1 and PIE2 registers, respectively. GIE is cleared on Reset.

The Return from Interrupt instruction, `RETFIE`, exits the interrupt routine, as well as sets the GIE bit, which re-enables unmasked interrupts.

The following interrupt flags are contained in the INTCON register:

- INT Pin Interrupt
- PORTA/PORTB Change Interrupts
- TMR0 Overflow Interrupt

The peripheral interrupt flags are contained in the special registers, PIR1 and PIR2. The corresponding interrupt enable bits are contained in special registers, PIE1 and PIE2.

The following interrupt flags are contained in the PIR1 register:

- A/D Interrupt
- EUSART Receive and Transmit Interrupts
- Timer1 Overflow Interrupt
- Synchronous Serial Port (SSP) Interrupt
- Enhanced CCP1 Interrupt
- Timer1 Overflow Interrupt
- Timer2 Match Interrupt

The following interrupt flags are contained in the PIR2 register:

- Fail-Safe Clock Monitor Interrupt
- 2 Comparator Interrupts
- EEPROM Data Write Interrupt

When an interrupt is serviced:

- The GIE is cleared to disable any further interrupt.
- The return address is pushed onto the stack.
- The PC is loaded with 0004h.

For external interrupt events, such as the INT pin, PORTA/PORTB change interrupts, the interrupt latency will be three or four instruction cycles. The exact latency depends upon when the interrupt event occurs (see Figure 14-7). The latency is the same for one or two-cycle instructions. Once in the Interrupt Service Routine, the source(s) of the interrupt can be determined by polling the interrupt flag bits. The interrupt flag bit(s) must be cleared in software before re-enabling interrupts to avoid multiple interrupt requests.

| | |
|---|---|
| **Note 1:** | Individual interrupt flag bits are set, regardless of the status of their corresponding mask bit or the GIE bit. |
| **2:** | When an instruction that clears the GIE bit is executed, any interrupts that were pending for execution in the next cycle are ignored. The interrupts, which were ignored, are still pending to be serviced when the GIE bit is set again. |

For additional information on Timer1, Timer2, comparators, A/D, data EEPROM, EUSART, SSP or Enhanced CCP modules, refer to the respective peripheral section.

### 14.3.1 RA2/INT INTERRUPT

External interrupt on RA2/INT pin is edge-triggered; either rising if the INTEDG bit (OPTION_REG<6>) is set, or falling, if the INTEDG bit is clear. When a valid edge appears on the RA2/INT pin, the INTF bit (INTCON<1>) is set. This interrupt can be disabled by clearing the INTE control bit (INTCON<4>). The INTF bit must be cleared in software in the Interrupt Service Routine before re-enabling this interrupt. The RA2/INT interrupt can wake-up the processor from Sleep, if the INTE bit was set prior to going into Sleep. The status of the GIE bit decides whether or not the processor branches to the interrupt vector following wake-up (0004h). See **Section 14.6 “Power-Down Mode (Sleep)”** for details on Sleep and Figure 14-9 for timing of wake-up from Sleep through RA2/INT interrupt.

| | |
|---|---|
| **Note:** | The ANSEL (11Eh) and CM2CON0 (11Ah) registers must be initialized to configure an analog channel as a digital input. Pins configured as analog inputs will read ‘0’. |

### 14.3.2 TMR0 INTERRUPT

An overflow (FFh → 00h) in the TMR0 register will set the T0IF (INTCON<2>) bit. The interrupt can be enabled/disabled by setting/clearing T0IE (INTCON<5>) bit. See **Section 5.0 “Timer0 Module”** for operation of the Timer0 module.

### 14.3.3 PORTA/PORTB INTERRUPT

An input change on PORTA or PORTB change sets the RABIF (INTCON<0>) bit. The interrupt can be enabled/disabled by setting/clearing the RABIE (INTCON<3>) bit. Plus, individual pins can be configured through the IOCA or IOCB registers.

| | |
|---|---|
| **Note:** | If a change on the I/O pin should occur when the read operation is being executed (start of the Q2 cycle), then the RABIF interrupt flag may not get set. See **Section 4.2.2 "Interrupt-on-change"** for more information. |

**FIGURE 14-6: INTERRUPT LOGIC**

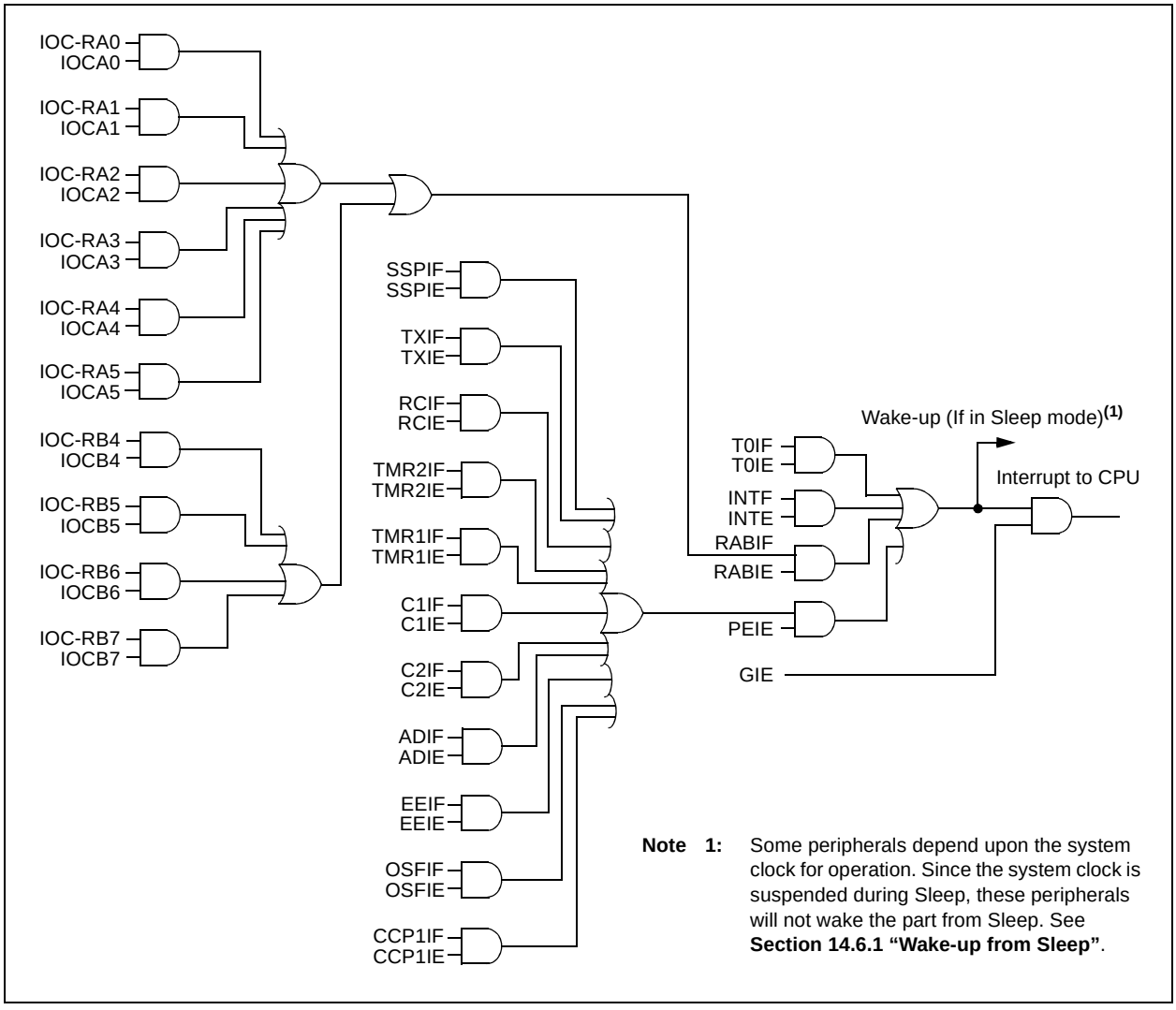
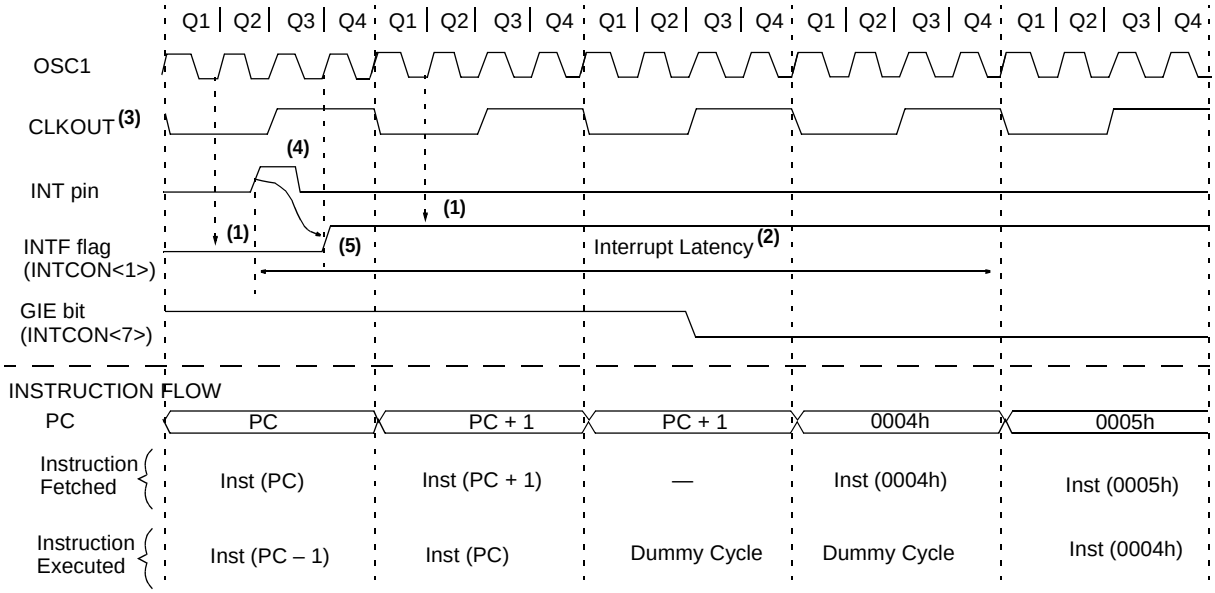


**FIGURE 14-7: INT PIN INTERRUPT TIMING**




**Note 1:** INTF flag is sampled here (every Q1).

**2:** Asynchronous interrupt latency = 3-4 TCY. Synchronous latency = 3 TCY, where TCY = instruction cycle time. Latency is the same whether Inst (PC) is a single cycle or a 2-cycle instruction.

**3:** CLKOUT is available only in INTOSC and RC Oscillator modes.

**4:** For minimum width of INT pulse, refer to AC specifications in **Section 17.0 "Electrical Specifications"**.

**5:** INTF is enabled to be set any time during the Q4-Q1 cycles.

**TABLE 14-6: SUMMARY OF INTERRUPT 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 | T0IE | INTE | RABIE | T0IF | INTF | RABIF | 0000 000x | 0000 000x |
| 0Ch | PIR1 | — | ADIF | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | -000 0000 | -000 0000 |
| 0Dh | PIR2 | OSFIF | C2IF | C1IF | EEIF | — | — | — | — | 0000 ---- | 0000 ---- |
| 8Ch | PIE1 | — | ADIE | RCIE | TXIE | SSPIE | CCP1IE | TMR2IE | TMR1IE | -000 0000 | -000 0000 |
| 8Dh | PIE2 | OSFIE | C2IE | C1IE | EEIE | — | — | — | — | 0000 ---- | 0000 ---- |

**Legend:** x = unknown, u = unchanged, – = unimplemented read as '0', q = value depends upon condition.
Shaded cells are not used by the interrupt module.

## 14.4 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 (e.g., W and Status registers). This must be implemented in software.

Since the upper 16 bytes of all GPR banks are common in the PIC16F685/687/689/690 (see Figures 2-1 and 2-2), temporary holding registers, W_TEMP and STATUS_TEMP, should be placed in here. These 16 locations do not require banking and therefore, make it easier to context save and restore. The same code shown in Example 14-1 can be used to:

- Store the W register
- Store the Status register
- Execute the ISR code
- Restore the Status (and Bank Select Bit register)
- Restore the W register

| **Note:** | The PIC16F685/687/689/690 normally does not require saving the PCLATH. However, if computed GOTO's are used in the ISR and the main code, the PCLATH must be saved and restored in the ISR. |
|---|---|

**EXAMPLE 14-1: SAVING STATUS AND W REGISTERS IN RAM**

```
MOVWF  W_TEMP           ;Copy W to TEMP register
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
:
:(ISR)                  ;Insert user code here
:
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
```

## 14.5 Watchdog Timer (WDT)

The new WDT is functionally compatible with previously designed WDT modules from other PICmicro® microcontrollers. Besides being backwards compatible, the WDT module has added capabilities to control a 16-bit prescaler through software. This allows the user to modify the prescale value for the WDT and TMR0 independently. Additionally, the WDT time-out value can be extended to 268 seconds because of the 16-bit prescaler. The WDT is cleared under certain conditions, which are described in Table 14-7.

### 14.5.1 WDT OSCILLATOR

The WDT derives its time base from the 31 kHz LFINTOSC oscillator, and on any Reset, the value of WDTCON is '---0 1000'. The resultant Reset value for WDTCON yields a nominal time base of 16 ms for the WDT. The new prescaler, that was added to the path between the LFINTOSC oscillator and the multiplexers, is used to divide the LFINTOSC oscillator by values between 32 and 65536. As a result of the combination of prescalers, a nominal range of 1 ms to 268s time out period for the WDT can be achieved. Figure 14-8 shows a block diagram of the WDT circuitry and where the new prescaler was designed into the circuit.

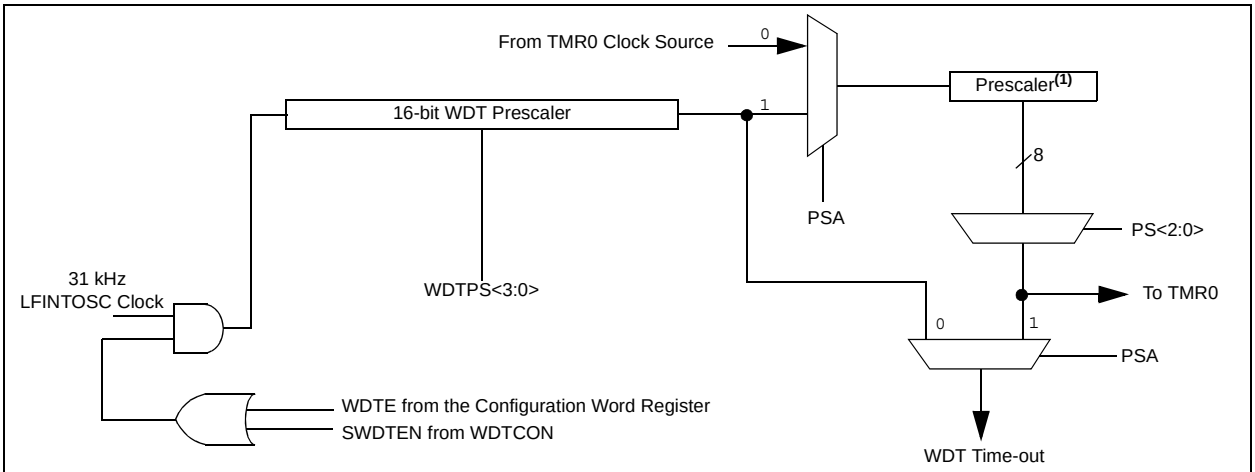
### 14.5.2 WDT CONTROL

When the WDTE bit (CONFIG<3>) is set, it enables the WDT and will continuously run. When the bit is clear, the WDT is disabled, but can be controlled through software in program memory and then SWDTEN bit (WDTCON<0>) has no effect. If WDTE is clear, the SWDTEN bit can be used to enable and disable the WDT through software in program memory.

The PSA<3> and PS<2:0> bits in the OPTION register (Register 2-2) have the same function as the WDT modules previously designed for PICmicro microcontrollers. See **Section 5.0 "Timer0 Module"** for more information about the OPTION register.

| **Note:** | When the Oscillator Start-up Timer (OST) is invoked, the WDT is held in Reset, because the WDT Ripple Counter is used by the OST to perform the oscillator delay count. When the OST count has expired, the WDT will begin counting (if enabled). |
|---|---|

**FIGURE 14-8: WATCHDOG TIMER BLOCK DIAGRAM**




**Note 1:** This is the shared Timer0/WDT prescaler. See **Section 5.4 "Prescaler"** for more information.

**TABLE 14-7: WDT STATUS**

| Conditions | WDT |
|---|---|
| WDTE = 0 | Cleared |
| CLRWDT Command | |
| Oscillator Fail Detected | |
| Exit Sleep + System Clock = T1OSC, EXTRC, INTOSC, EXTCLK | |
| Exit Sleep + System Clock = XT, HS, LP | Cleared until the end of OST |

**REGISTER 14-2: WDTCON – WATCHDOG TIMER CONTROL REGISTER (ADDRESS: 97h)**

| U-0 | U-0 | U-0 | R/W-0 | R/W-1 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | — | — | WDTPS3 | WDTPS2 | WDTPS1 | WDTPS0 | SWDTEN(1) |
| bit 7 | | | | | | | bit 0 |

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

bit 4-1 **WDTPS<3:0>:** Watchdog Timer Period Select bits

Bit Value = Prescale Rate

0000 = 1:32
0001 = 1:64
0010 = 1:128
0011 = 1:256
0100 = 1:512 (Reset value)
0101 = 1:1024
0110 = 1:2048
0111 = 1:4096
1000 = 1:8192
1001 = 1:16384
1010 = 1:32768
1011 = 1:65536
1100 = reserved
1101 = reserved
1110 = reserved
1111 = reserved

bit 0 **SWDTEN:** Software Enable or Disable the Watchdog Timer(1)

1 = WDT is turned on
0 = WDT is turned off (Reset value)

**Note 1:** If WDTE configuration bit = 1, then WDT is always enabled, irrespective of this control bit. If WDTE configuration bit = 0, then it is possible to turn WDT on/off with this control bit.

| **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 |

**TABLE 14-8: SUMMARY OF WATCHDOG TIMER REGISTERS**

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|
| 97h | WDTCON | — | — | — | WDTPS3 | WDTPS2 | WSTPS1 | WDTPS0 | SWDTEN |
| 81h/181h | OPTION_REG | $\overline{\text{RABPU}}$ | INTEDG | T0CS | T0SE | PSA | PS2 | PS1 | PS0 |
| 2007h(1) | CONFIG | $\overline{\text{CPD}}$ | $\overline{\text{CP}}$ | MCLRE | $\overline{\text{PWRTE}}$ | WDTE | FOSC2 | FOSC1 | FOSC0 |

**Legend:** Shaded cells are not used by the Watchdog Timer.

**Note 1:** See Register 14-1 for operation of all Configuration Word register bits.

## 14.6 Power-Down Mode (Sleep)

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

If the Watchdog Timer is enabled:

- WDT will be cleared but keeps running.
- $\overline{PD}$ bit in the Status register is cleared.
- $\overline{TO}$ bit is set.
- Oscillator driver is turned off.
- I/O ports maintain the status they had before SLEEP was executed (driving high, low or high-impedance).

For lowest current consumption in this mode, all I/O pins should be either at $V_{DD}$ or $V_{SS}$, with no external circuitry drawing current from the I/O pin and the comparators and $CV_{REF}$ should be disabled. I/O pins that are high-impedance inputs should be pulled 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 PORTA should be considered.

The $\overline{MCLR}$ pin must be at a logic high level.

| **Note:** | It should be noted that a Reset generated by a WDT time-out does not drive $\overline{MCLR}$ pin low. |
|---|---|

### 14.6.1 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 RA2/INT pin, PORTA change or a peripheral interrupt.

The first event will cause a device Reset. The two latter events are considered a continuation of program execution. 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. $\overline{TO}$ bit is cleared if WDT wake-up occurred.

The following peripheral interrupts can wake the device from Sleep:

1. TMR1 interrupt. Timer1 must be operating as an asynchronous counter.
2. ECCP Capture mode interrupt.
3. Special event trigger (Timer1 in Asynchronous mode using an external clock).
4. A/D conversion (when A/D clock source is RC).
5. EEPROM write operation completion.
6. Comparator output changes state.
7. Interrupt-on-change.
8. External Interrupt from INT pin.
9. EUSART Break detect, $I^2C$ slave.

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 prefetched. 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, 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.

| **Note:** | If the global interrupts are disabled (GIE is cleared), but any interrupt source has both its interrupt enable bit and the corresponding interrupt flag bits set, the device will immediately wake-up from Sleep. The SLEEP instruction is completely executed. |
|---|---|

The WDT is cleared when the device wakes up from Sleep, regardless of the source of wake-up.

### 14.6.2 WAKE-UP USING INTERRUPTS

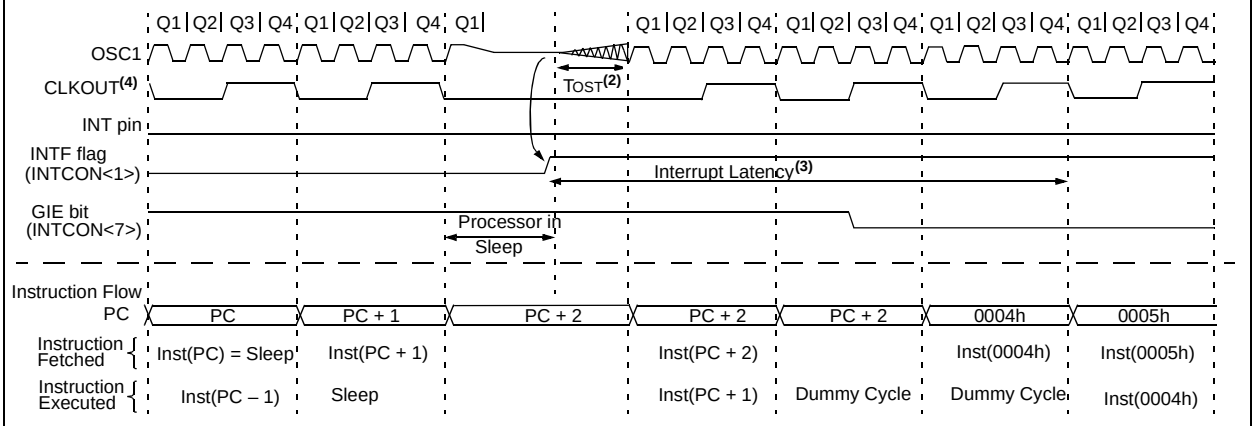
When global interrupts are disabled (GIE cleared) and any interrupt source has both its interrupt enable bit and interrupt flag bit set, one of the following will occur:

- If the interrupt occurs **before** the execution of a SLEEP instruction, the SLEEP instruction will complete as a NOP. Therefore, the WDT and WDT prescaler and postscaler (if enabled) will not be cleared, the $\overline{TO}$ bit will not be set and the $\overline{PD}$ bit 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 prescaler and postscaler (if enabled) 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.

**FIGURE 14-9: WAKE-UP FROM SLEEP THROUGH INTERRUPT**




**Note 1:** XT, HS or LP Oscillator mode assumed.

**2:** TOST = 1024 TOSC (drawing not to scale). This delay does not apply to EC and RC Oscillator modes.

**3:** GIE = 1 assumed. In this case after wake-up, the processor jumps to 0004h. If GIE = 0, execution will continue in-line.

**4:** CLKOUT is not available in XT, HS, LP or EC Oscillator modes, but shown here for timing reference.

## 14.7 Code Protection

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

| **Note:** | The entire data EEPROM and Flash program memory will be erased when the code protection is switched from on to off. See the "*PIC12F6XX/16F6XX Memory Programming Specification*" (DS41204) for more information. |
|---|---|

## 14.8 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 mode. Only the Least Significant 7 bits of the ID locations are used.

## 14.9 In-Circuit Serial Programming

The PIC16F685/687/689/690 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
- 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.

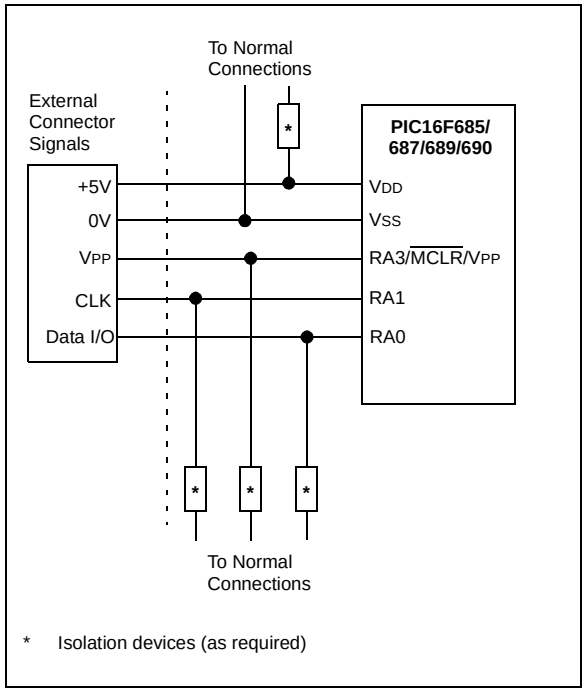
The device is placed into a Program/Verify mode by holding the RA0/AN0/C1IN+/ICSPDAT/ULPWU and RA1/AN1/C12IN-/VREF/ICSPCLK pins low, while raising the $\overline{\text{MCLR}}$ ($V_{PP}$) pin from $V_{IL}$ to $V_{IHH}$. See the "*PIC12F6XX/16F6XX Memory Programming Specification*" (DS41204) for more information. RA0 becomes the programming data and RA1 becomes the programming clock. Both RA0 and RA1 are Schmitt Trigger inputs in this mode.

After Reset, to place the device into Program/Verify mode, the Program Counter (PC) is at location 00h. A 6-bit command is then supplied to the device. Depending on the command, 14 bits of program data are then supplied to or from the device, depending on whether the command was a load or a read. For complete details of serial programming, please refer to the "*PIC12F6XX/16F6XX Memory Programming Specification*" (DS41204).

A typical In-Circuit Serial Programming connection is shown in Figure 14-10.

**FIGURE 14-10: TYPICAL IN-CIRCUIT SERIAL PROGRAMMING CONNECTION**



**NOTES:**

## 15.0 INSTRUCTION SET SUMMARY

The PIC16F685/687/689/690 instruction set is highly orthogonal and is comprised of three basic categories:

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

Each PIC16 instruction is a 14-bit word divided into an **opcode,** which specifies the instruction type and one or more **operands,** which further specify the operation of the instruction. The formats for each of the categories is presented in Figure 15-1, while the various opcode fields are summarized in Table 15-1.

Table 15-2 lists the instructions recognized by the MPASM™ assembler.

For **byte-oriented** instructions, 'f' represents a file register designator and 'd' represents a destination designator. The file register designator specifies which file register is to be used by the instruction.

The destination designator specifies where the result of the operation is to be placed. If 'd' is zero, the result is placed in the W register. If 'd' is one, the result is placed in the file register specified in the instruction.

For **bit-oriented** instructions, 'b' represents a bit field designator, which selects the bit affected by the operation, while 'f' represents the address of the file in which the bit is located.

For **literal and control** operations, 'k' represents an 8-bit or 11-bit constant, or literal value.

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

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

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

### 15.1 Read-Modify-Write Operations

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

For example, a CLRF PORTA instruction will read PORTA, clear all the data bits, then write the result back to PORTA. This example would have the unintended result of clearing the condition that set the RABIF flag.

**TABLE 15-1: 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.<br>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 |

**FIGURE 15-1: GENERAL FORMAT FOR INSTRUCTIONS**

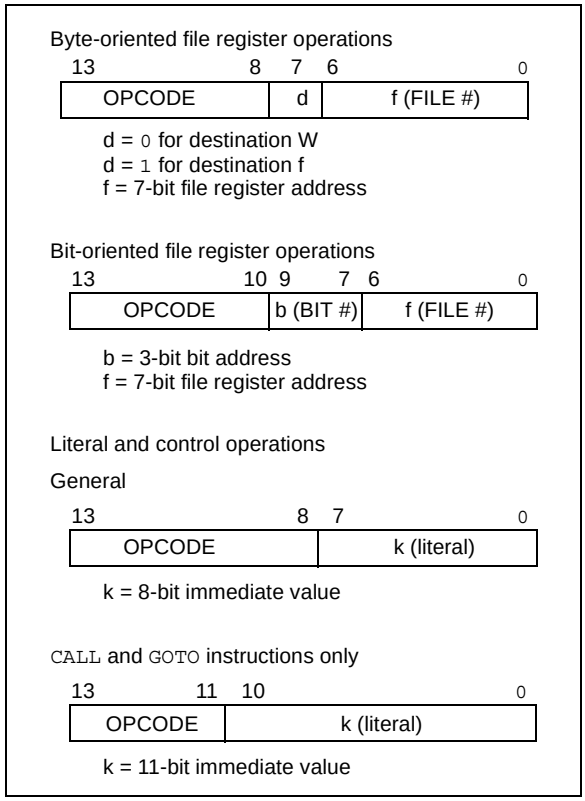


**TABLE 15-2: PIC16F685/687/689/690 INSTRUCTION SET**

| 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 | 0xxx | xxxx | 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 PORTA, 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 the Program Counter (PC) is modified, or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a `NOP`.

## 15.2 Instruction Descriptions

**ADDLW** — **Add literal and W**

| | |
|---|---|
| Syntax: | [ *label* ] ADDLW k |
| Operands: | 0 ≤ k ≤ 255 |
| Operation: | (W) + k → (W) |
| Status Affected: | C, DC, Z |
| Description: | The contents of the W register are added to the eight-bit literal 'k' and the result is placed in the W register. |

**ADDWF** — **Add W and f**

| | |
|---|---|
| Syntax: | [ *label* ] ADDWF f,d |
| Operands: | 0 ≤ f ≤ 127<br>d ∈ [0,1] |
| Operation: | (W) + (f) → (destination) |
| Status Affected: | C, DC, Z |
| Description: | Add 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'. |

**ANDLW** — **AND literal with W**

| | |
|---|---|
| Syntax: | [ *label* ] ANDLW k |
| Operands: | 0 ≤ k ≤ 255 |
| Operation: | (W) .AND. (k) → (W) |
| Status Affected: | Z |
| Description: | The contents of W register are AND'ed with the eight-bit literal 'k'. The result is placed in the W register. |

**ANDWF** — **AND W with f**

| | |
|---|---|
| Syntax: | [ *label* ] ANDWF f,d |
| Operands: | 0 ≤ f ≤ 127<br>d ∈ [0,1] |
| Operation: | (W) .AND. (f) → (destination) |
| Status Affected: | Z |
| Description: | AND 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'. |

**BCF** — **Bit Clear f**

| | |
|---|---|
| Syntax: | [ *label* ] BCF f,b |
| Operands: | 0 ≤ f ≤ 127<br>0 ≤ b ≤ 7 |
| Operation: | 0 → (f<b>) |
| Status Affected: | None |
| Description: | Bit 'b' in register 'f' is cleared. |

**BSF** — **Bit Set f**

| | |
|---|---|
| Syntax: | [ *label* ] BSF f,b |
| Operands: | 0 ≤ f ≤ 127<br>0 ≤ b ≤ 7 |
| Operation: | 1 → (f<b>) |
| Status Affected: | None |
| Description: | Bit 'b' in register 'f' is set. |

**BTFSC** — **Bit Test f, Skip if Clear**

| | |
|---|---|
| Syntax: | [ *label* ] BTFSC f,b |
| Operands: | 0 ≤ f ≤ 127<br>0 ≤ b ≤ 7 |
| Operation: | skip if (f<b>) = 0 |
| Status Affected: | None |
| Description: | If bit 'b' in register 'f' is '1', the next instruction is executed.<br>If bit 'b' in register 'f' is '0', the next instruction is discarded, and a NOP is executed instead, making this a two-cycle instruction. |

**BTFSS** **Bit Test f, Skip if Set**

| | |
|---|---|
| Syntax: | [ *label* ] BTFSS f,b |
| Operands: | 0 ≤ f ≤ 127<br>0 ≤ b < 7 |
| Operation: | skip if (f<b>) = 1 |
| Status Affected: | None |
| Description: | If bit 'b' in register 'f' is '0', the next instruction is executed.<br>If bit 'b' is '1', then the next instruction is discarded and a NOP is executed instead, making this a two-cycle instruction. |

**CALL** **Call Subroutine**

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

**CLRF** **Clear f**

| | |
|---|---|
| Syntax: | [ *label* ] CLRF f |
| Operands: | 0 ≤ f ≤ 127 |
| Operation: | 00h → (f)<br>1 → Z |
| Status Affected: | Z |
| Description: | The contents of register 'f' are cleared and the Z bit is set. |

**CLRW** **Clear W**

| | |
|---|---|
| Syntax: | [ *label* ] CLRW |
| Operands: | None |
| Operation: | 00h → (W)<br>1 → Z |
| Status Affected: | Z |
| Description: | W register is cleared. Zero bit (Z) is set. |

**CLRWDT** **Clear Watchdog Timer**

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

**COMF** **Complement f**

| | |
|---|---|
| Syntax: | [ *label* ] COMF f,d |
| Operands: | 0 ≤ f ≤ 127<br>d ∈ [0,1] |
| Operation: | ($\bar{f}$) → (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'. |

**DECF** **Decrement f**

| | |
|---|---|
| Syntax: | [ *label* ] DECF f,d |
| Operands: | 0 ≤ f ≤ 127<br>d ∈ [0,1] |
| Operation: | (f) - 1 → (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'. |

**DECFSZ** **Decrement f, Skip if 0**

Syntax: [ *label* ] DECFSZ f,d

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

Operation: (f) - 1 → (destination);
skip if result = 0

Status Affected: None

Description: The contents of register 'f' are decremented. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'.
If the result is '1', the next instruction is executed. If the result is '0', then a NOP is executed instead, making it a two-cycle instruction.

**GOTO** **Unconditional Branch**

Syntax: [ *label* ] GOTO k

Operands: $0 \le k \le 2047$

Operation: k → PC<10:0>
PCLATH<4:3> → PC<12:11>

Status Affected: None

Description: GOTO is an unconditional branch. The eleven-bit immediate value is loaded into PC bits <10:0>. The upper bits of PC are loaded from PCLATH<4:3>. GOTO is a two-cycle instruction.

**INCF** **Increment f**

Syntax: [ *label* ] INCF f,d

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

Operation: (f) + 1 → (destination)

Status Affected: Z

Description: The contents of register 'f' are incremented. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'.

**INCFSZ** **Increment f, Skip if 0**

Syntax: [ *label* ] INCFSZ f,d

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

Operation: (f) + 1 → (destination),
skip if result = 0

Status Affected: None

Description: The contents of register 'f' are incremented. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'.
If the result is '1', the next instruction is executed. If the result is '0', a NOP is executed instead, making it a two-cycle instruction.

**IORLW** **Inclusive OR literal with W**

Syntax: [ *label* ] IORLW k

Operands: $0 \le k \le 255$

Operation: (W) .OR. k → (W)

Status Affected: Z

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

**IORWF** **Inclusive OR W with f**

Syntax: [ *label* ] IORWF f,d

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

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

Status Affected: Z

Description: Inclusive OR the W register with register 'f'. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'.

**MOVF** **Move f**

Syntax: [ *label* ] MOVF f,d

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

Operation: (f) → (dest)

Status Affected: Z

Description: The contents of register 'f' is moved to a destination dependent upon the status of 'd'. If d = 0, destination is W register. If d = 1, the destination is file register 'f' itself. d = 1 is useful to test a file register since Status flag Z is affected.

Words: 1

Cycles: 1

Example
```
MOVF    FSR, 0
```
After Instruction
W = value in FSR register
Z = 1

**MOVWF** **Move W to f**

Syntax: [ *label* ] MOVWF f

Operands: $0 \le f \le 127$

Operation: (W) → (f)

Status Affected: None

Description: Move data from W register to register 'f'.

Words: 1

Cycles: 1

Example
```
MOVW    OPTION
F
```
Before Instruction
OPTION = 0xFF
W = 0x4F
After Instruction
OPTION = 0x4F
W = 0x4F

**MOVLW** **Move literal to W**

Syntax: [ *label* ] MOVLW k

Operands: $0 \le k \le 255$

Operation: k → (W)

Status Affected: None

Description: The eight-bit literal 'k' is loaded into W register. The "don't cares" will assemble as '0's.

Words: 1

Cycles: 1

Example
```
MOVLW   0x5A
```
After Instruction
W = 0x5A

**NOP** **No Operation**

Syntax: [ *label* ] NOP

Operands: None

Operation: No operation

Status Affected: None

Description: No operation.

Words: 1

Cycles: 1

Example
```
NOP
```

| RETFIE | Return from Interrupt |
|---|---|
| Syntax: | [ *label* ] RETFIE |
| Operands: | None |
| Operation: | TOS → PC,<br>1 → GIE |
| Status Affected: | None |
| Description: | Return from Interrupt. Stack is POPed and Top-of-Stack (TOS) is loaded in the PC. Interrupts are enabled by setting Global Interrupt Enable bit, GIE (INTCON<7>). This is a two-cycle instruction. |
| Words: | 1 |
| Cycles: | 2 |
| Example | `RETFIE`<br>After Interrupt<br>PC = TOS<br>GIE = `1` |

| RETLW | Return with literal in W |
|---|---|
| Syntax: | [ *label* ] RETLW k |
| Operands: | 0 ≤ k ≤ 255 |
| Operation: | k → (W);<br>TOS → PC |
| Status Affected: | None |
| Description: | The W register is loaded with the eight-bit literal 'k'. The program counter is loaded from the top of the stack (the return address). This is a two-cycle instruction. |
| Words: | 1 |
| Cycles: | 2 |
| Example | `CALL TABLE;W contains table`<br>;offset value<br>`TABLE` • ;W now has table value<br>•<br>•<br>ADDWF PC ;W = offset<br>RETLW k1 ;Begin table<br>RETLW k2 ;<br>•<br>•<br>•<br>RETLW kn ; End of table<br>Before Instruction<br>W = 0x07<br>After Instruction<br>W = value of k8 |

| 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. |

**RLF** **Rotate Left f through Carry**

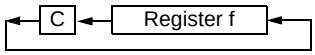
Syntax: [ *label* ] RLF f,d

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

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'.




Words: 1

Cycles: 1

Example

```
RLF     REG1,0
```

Before Instruction

```
REG1    =   1110 0110
C       =   0
```

After Instruction

```
REG1    =   1110 0110
W       =   1100 1100
C       =   1
```

**RRF** **Rotate Right f through Carry**

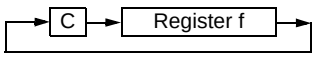
Syntax: [ *label* ] RRF f,d

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

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'.




**SLEEP** **Enter Sleep mode**

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.

**SUBLW** **Subtract W from literal**

Syntax: [ *label* ] SUBLW k

Operands: $0 \le k \le 255$

Operation: k - (W) → (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.

**SUBWF** **Subtract W from f**

Syntax: [ *label* ] SUBWF f,d

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

Operation: (f) - (W) → (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'.

| SWAPF | Swap Nibbles in f |
|---|---|
| Syntax: | [ *label* ] SWAPF f,d |
| Operands: | $0 \le f \le 127$<br>$d \in [0,1]$ |
| Operation: | (f<3:0>) → (destination<7:4>),<br>(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'. |

| XORLW | Exclusive OR literal with W |
|---|---|
| Syntax: | [ *label* ] XORLW k |
| Operands: | $0 \le k \le 255$ |
| 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. |

| XORWF | Exclusive OR W with f |
|---|---|
| Syntax: | [ *label* ] XORWF f,d |
| Operands: | $0 \le f \le 127$<br>$d \in [0,1]$ |
| 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'. |

**NOTES:**

## 16.0 DEVELOPMENT SUPPORT

The PICmicro® microcontrollers are supported with a full range of hardware and software development tools:

- Integrated Development Environment
  - MPLAB® IDE Software
- Assemblers/Compilers/Linkers
  - MPASM™ Assembler
  - MPLAB C17 and MPLAB C18 C Compilers
  - MPLINK™ Object Linker/ MPLIB™ Object Librarian
  - MPLAB C30 C Compiler
  - MPLAB ASM30 Assembler/Linker/Library
- Simulators
  - MPLAB SIM Software Simulator
  - MPLAB dsPIC30 Software Simulator
- Emulators
  - MPLAB ICE 2000 In-Circuit Emulator
  - MPLAB ICE 4000 In-Circuit Emulator
- In-Circuit Debugger
  - MPLAB ICD 2
- Device Programmers
  - PRO MATE® II Universal Device Programmer
  - PICSTART® Plus Development Programmer
  - MPLAB PM3 Device Programmer
- Low-Cost Demonstration Boards
  - PICDEM™ 1 Demonstration Board
  - PICDEM.net™ Demonstration Board
  - PICDEM 2 Plus Demonstration Board
  - PICDEM 3 Demonstration Board
  - PICDEM 4 Demonstration Board
  - PICDEM 17 Demonstration Board
  - PICDEM 18R Demonstration Board
  - PICDEM LIN Demonstration Board
  - PICDEM USB Demonstration Board
- Evaluation Kits
  - KEELOQ® Evaluation and Programming Tools
  - PICDEM MSC
  - microID® Developer Kits
  - CAN
  - PowerSmart® Developer Kits
  - Analog

### 16.1 MPLAB Integrated Development Environment Software

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

- An interface to debugging tools
  - simulator
  - programmer (sold separately)
  - emulator (sold separately)
  - in-circuit debugger (sold separately)
- A full-featured editor with color coded context
- A multiple project manager
- Customizable data windows with direct edit of contents
- High-level source code debugging
- Mouse over variable inspection
- Extensive on-line help

The MPLAB IDE allows you to:

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

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

### 16.2 MPASM Assembler

The MPASM assembler is a full-featured, universal macro assembler for all PICmicro 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

## 16.3 MPLAB C17 and MPLAB C18 C Compilers

The MPLAB C17 and MPLAB C18 Code Development Systems are complete ANSI C compilers for Microchip's PIC17CXXX and PIC18CXXX family of microcontrollers. 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.

## 16.4 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK object linker combines relocatable objects created by the MPASM assembler and the MPLAB C17 and MPLAB C18 C compilers. 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

## 16.5 MPLAB C30 C Compiler

The MPLAB C30 C compiler is a full-featured, ANSI compliant, optimizing compiler that translates standard ANSI C programs into dsPIC30F assembly language source. The compiler also supports many command line options and language extensions to take full advantage of the dsPIC30F device hardware capabilities and afford fine control of the compiler code generator.

MPLAB C30 is distributed with a complete ANSI C standard library. All library functions have been validated and conform to the ANSI C library standard. The library includes functions for string manipulation, dynamic memory allocation, data conversion, time-keeping and math functions (trigonometric, exponential and hyperbolic). The compiler provides symbolic information for high-level source debugging with the MPLAB IDE.

## 16.6 MPLAB ASM30 Assembler, Linker and Librarian

MPLAB ASM30 assembler produces relocatable machine code from symbolic assembly language for dsPIC30F devices. MPLAB C30 compiler uses the assembler to produce it's 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

## 16.7 MPLAB SIM Software Simulator

The MPLAB SIM software simulator allows code development in a PC hosted environment by simulating the PICmicro series microcontrollers on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a file, or user defined key press, to any pin. The execution can be performed in Single-Step, Execute Until Break or Trace mode.

The MPLAB SIM simulator fully supports symbolic debugging using the MPLAB C17 and MPLAB C18 C Compilers, as well as the MPASM assembler. The software simulator offers the flexibility to develop and debug code outside of the laboratory environment, making it an excellent, economical software development tool.

## 16.8 MPLAB SIM30 Software Simulator

The MPLAB SIM30 software simulator allows code development in a PC hosted environment by simulating the dsPIC30F series microcontrollers on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a file, or user defined key press, to any of the pins.

The MPLAB SIM30 simulator fully supports symbolic debugging using the MPLAB C30 C Compiler and MPLAB ASM30 assembler. The simulator runs in either a Command Line mode for automated tasks, or from MPLAB IDE. This high-speed simulator is designed to debug, analyze and optimize time intensive DSP routines.

## 16.9 MPLAB ICE 2000 High-Performance Universal In-Circuit Emulator

The MPLAB ICE 2000 universal in-circuit emulator is intended to provide the product development engineer with a complete microcontroller design tool set for PICmicro 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 universal architecture of the MPLAB ICE in-circuit emulator allows expansion to support new PICmicro 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.

## 16.10 MPLAB ICE 4000 High-Performance Universal In-Circuit Emulator

The MPLAB ICE 4000 universal in-circuit emulator is intended to provide the product development engineer with a complete microcontroller design tool set for high-end PICmicro microcontrollers. Software control of the MPLAB ICE in-circuit emulator is provided by the MPLAB Integrated Development Environment, which allows editing, building, downloading and source debugging from a single environment.

The MPLAB ICD 4000 is a premium emulator system, providing the features of MPLAB ICE 2000, but with increased emulation memory and high-speed performance for dsPIC30F and PIC18XXXX devices. Its advanced emulator features include complex triggering and timing, up to 2 Mb of emulation memory and the ability to view variables in real-time.

The MPLAB ICE 4000 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.

## 16.11 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 PICmicro MCUs and can be used to develop for these and other PICmicro microcontrollers. 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, 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 PICmicro devices.

## 16.12 PRO MATE II Universal Device Programmer

The PRO MATE II is a universal, CE compliant device programmer with programmable voltage verification at $V_{DDMIN}$ and $V_{DDMAX}$ for maximum reliability. It features an LCD display for instructions and error messages and a modular detachable socket assembly to support various package types. In Stand-Alone mode, the PRO MATE II device programmer can read, verify and program PICmicro devices without a PC connection. It can also set code protection in this mode.

## 16.13 MPLAB PM3 Device Programmer

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

## 16.14 PICSTART Plus Development Programmer

The PICSTART Plus development programmer is an easy-to-use, low-cost, prototype programmer. It connects to the PC via a COM (RS-232) port. MPLAB Integrated Development Environment software makes using the programmer simple and efficient. The PICSTART Plus development programmer supports most PICmicro devices up to 40 pins. Larger pin count devices, such as the PIC16C92X and PIC17C76X, may be supported with an adapter socket. The PICSTART Plus development programmer is CE compliant.

## 16.15 PICDEM 1 PICmicro Demonstration Board

The PICDEM 1 demonstration board demonstrates the capabilities of the PIC16C5X (PIC16C54 to PIC16C58A), PIC16C61, PIC16C62X, PIC16C71, PIC16C8X, PIC17C42, PIC17C43 and PIC17C44. All necessary hardware and software is included to run basic demo programs. The sample microcontrollers provided with the PICDEM 1 demonstration board can be programmed with a PRO MATE II device programmer or a PICSTART Plus development programmer. The PICDEM 1 demonstration board can be connected to the MPLAB ICE in-circuit emulator for testing. A prototype area extends the circuitry for additional application components. Features include an RS-232 interface, a potentiometer for simulated analog input, push button switches and eight LEDs.

## 16.16 PICDEM.net Internet/Ethernet Demonstration Board

The PICDEM.net demonstration board is an Internet/Ethernet demonstration board using the PIC18F452 microcontroller and TCP/IP firmware. The board supports any 40-pin DIP device that conforms to the standard pinout used by the PIC16F877 or PIC18C452. This kit features a user friendly TCP/IP stack, web server with HTML, a 24L256 Serial EEPROM for Xmodem download to web pages into Serial EEPROM, ICSP/MPLAB ICD 2 interface connector, an Ethernet interface, RS-232 interface and a 16 x 2 LCD display. Also included is the book and CD-ROM *"TCP/IP Lean, Web Servers for Embedded Systems,"* by Jeremy Bentham

## 16.17 PICDEM 2 Plus Demonstration Board

The PICDEM 2 Plus demonstration board supports many 18, 28 and 40-pin microcontrollers, including PIC16F87X and PIC18FXX2 devices. All the necessary hardware and software is included to run the demonstration programs. The sample microcontrollers provided with the PICDEM 2 demonstration board can be programmed with a PRO MATE II device programmer, PICSTART Plus development programmer, or MPLAB ICD 2 with a Universal Programmer Adapter. The MPLAB ICD 2 and MPLAB ICE in-circuit emulators may also be used with the PICDEM 2 demonstration board to test firmware. A prototype area extends the circuitry for additional application components. Some of the features include an RS-232 interface, a 2 x 16 LCD display, a piezo speaker, an on-board temperature sensor, four LEDs and sample PIC18F452 and PIC16F877 Flash microcontrollers.

## 16.18 PICDEM 3 PIC16C92X Demonstration Board

The PICDEM 3 demonstration board supports the PIC16C923 and PIC16C924 in the PLCC package. All the necessary hardware and software is included to run the demonstration programs.

## 16.19 PICDEM 4 8/14/18-Pin Demonstration Board

The PICDEM 4 can be used to demonstrate the capabilities of the 8, 14 and 18-pin PIC16XXXX and PIC18XXXX MCUs, including the PIC16F818/819, PIC16F87/88, PIC16F62XA and the PIC18F1320 family of microcontrollers. PICDEM 4 is intended to showcase the many features of these low pin count parts, including LIN and Motor Control using ECCP. Special provisions are made for low-power operation with the supercapacitor circuit and jumpers allow on-board hardware to be disabled to eliminate current draw in this mode. Included on the demo board are provisions for Crystal, RC or Canned Oscillator modes, a five volt regulator for use with a nine volt wall adapter or battery, DB-9 RS-232 interface, ICD connector for programming via ICSP and development with MPLAB ICD 2, 2 x 16 liquid crystal display, PCB footprints for H-Bridge motor driver, LIN transceiver and EEPROM. Also included are: header for expansion, eight LEDs, four potentiometers, three push buttons and a prototyping area. Included with the kit is a PIC16F627A and a PIC18F1320. Tutorial firmware is included along with the User's Guide.

## 16.20 PICDEM 17 Demonstration Board

The PICDEM 17 demonstration board is an evaluation board that demonstrates the capabilities of several Microchip microcontrollers, including PIC17C752, PIC17C756A, PIC17C762 and PIC17C766. A programmed sample is included. The PRO MATE II device programmer, or the PICSTART Plus development programmer, can be used to reprogram the device for user tailored application development. The PICDEM 17 demonstration board supports program download and execution from external on-board Flash memory. A generous prototype area is available for user hardware expansion.

## 16.21 PICDEM 18R PIC18C601/801 Demonstration Board

The PICDEM 18R demonstration board serves to assist development of the PIC18C601/801 family of Microchip microcontrollers. It provides hardware implementation of both 8-bit Multiplexed/Demultiplexed and 16-bit Memory modes. The board includes 2 Mb external Flash memory and 128 Kb SRAM memory, as well as serial EEPROM, allowing access to the wide range of memory types supported by the PIC18C601/801.

## 16.22 PICDEM LIN PIC16C43X Demonstration Board

The powerful LIN hardware and software kit includes a series of boards and three PICmicro microcontrollers. The small footprint PIC16C432 and PIC16C433 are used as slaves in the LIN communication and feature on-board LIN transceivers. A PIC16F874 Flash microcontroller serves as the master. All three microcontrollers are programmed with firmware to provide LIN bus communication.

## 16.23 PICkit™ 1 Flash Starter Kit

A complete “development system in a box”, the PICkit™ Flash Starter Kit includes a convenient multi-section board for programming, evaluation and development of 8/14-pin Flash PIC® microcontrollers. Powered via USB, the board operates under a simple Windows GUI. The PICkit 1 Starter Kit includes the User’s Guide (on CD ROM), PICkit 1 tutorial software and code for various applications. Also included are MPLAB® IDE (Integrated Development Environment) software, software and hardware “Tips ‘n Tricks for 8-pin Flash PIC® Microcontrollers” Handbook and a USB interface cable. Supports all current 8/14-pin Flash PIC microcontrollers, as well as many future planned devices.

## 16.24 PICDEM USB PIC16C7X5 Demonstration Board

The PICDEM USB Demonstration Board shows off the capabilities of the PIC16C745 and PIC16C765 USB microcontrollers. This board provides the basis for future USB products.

## 16.25 Evaluation and Programming Tools

In addition to the PICDEM series of circuits, Microchip has a line of evaluation kits and demonstration software for these products.

- KEELOQ evaluation and programming tools for Microchip’s HCS Secure Data Products
- CAN developers kit for automotive network applications
- Analog design boards and filter design software
- PowerSmart battery charging evaluation/ calibration kits
- IrDA® development kit
- microID development and rfLab™ development software
- SEEVAL® designer kit for memory evaluation and endurance calculations
- PICDEM MSC demo boards for Switching mode power supply, high-power IR driver, delta sigma ADC and flow rate sensor

Check the Microchip web page and the latest Product Selector Guide for the complete list of demonstration and evaluation kits.

**NOTES:**

## 17.0 ELECTRICAL SPECIFICATIONS

### Absolute Maximum Ratings(†)

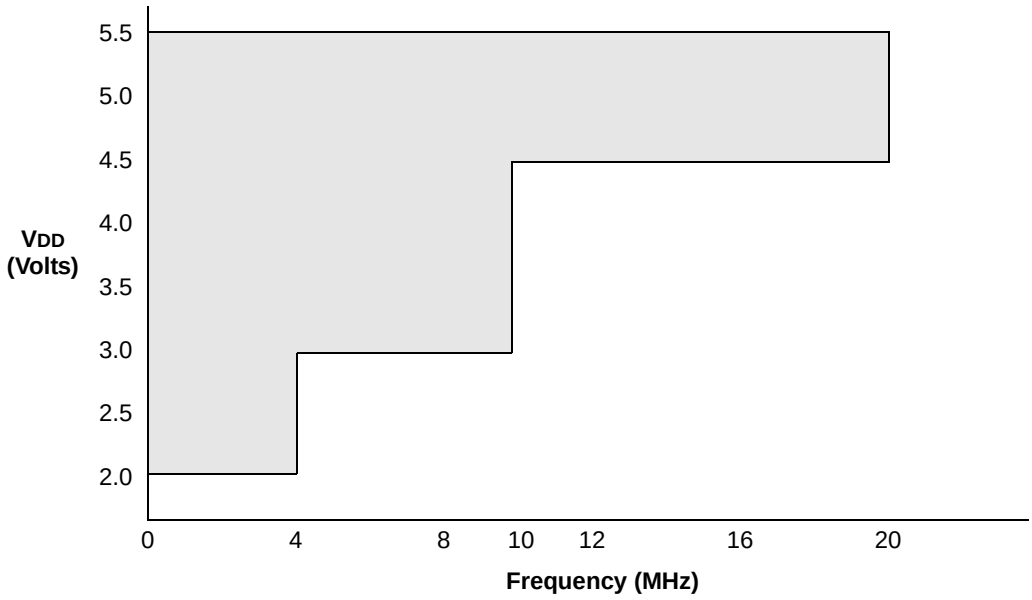
Ambient temperature under bias ..... -40° to +125°C
Storage temperature ..... -65°C to +150°C
Voltage on $V_{DD}$ with respect to $V_{SS}$ ..... -0.3V to +6.5V
Voltage on $\overline{MCLR}$ with respect to Vss ..... -0.3V to +13.5V
Voltage on all other pins with respect to $V_{SS}$ ..... -0.3V to ($V_{DD}$ + 0.3V)
Total power dissipation(1) ..... 800 mW
Maximum current out of $V_{SS}$ pin ..... 300 mA
Maximum current into $V_{DD}$ pin ..... 250 mA
Input clamp current, $I_{IK}$ ($V_I$ < 0 or $V_I$ > $V_{DD}$) ..... ± 20 mA
Output clamp current, $I_{OK}$ (Vo < 0 or Vo > $V_{DD}$) ..... ± 20 mA
Maximum output current sunk by any I/O pin ..... 25 mA
Maximum output current sourced by any I/O pin ..... 25 mA
Maximum current sunk by PORTA, PORTB and PORTC (combined) ..... 200 mA
Maximum current sourced PORTA, PORTB and PORTC (combined) ..... 200 mA

**Note 1:** Power dissipation is calculated as follows: $P_{DIS} = V_{DD} \times \{I_{DD} - \sum I_{OH}\} + \sum \{(V_{DD} - V_{OH}) \times I_{OH}\} + \sum(V_{OL} \times I_{OL})$.

† NOTICE: Stresses above those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

**Note:** Voltage spikes below $V_{SS}$ at the $\overline{MCLR}$ pin, inducing currents greater than 80 mA, may cause latch-up. Thus, a series resistor of 50-100 Ω should be used when applying a “low” level to the $\overline{MCLR}$ pin, rather than pulling this pin directly to $V_{SS}$.

**FIGURE 17-1: PIC16F685/687/689/690 VOLTAGE-FREQUENCY GRAPH, -40°C ≤ TA ≤ +125°C**




**Note 1:** The shaded region indicates the permissible combinations of voltage and frequency.

## 17.1 DC Characteristics: PIC16F685/687/689/690-I (Industrial) PIC16F685/687/689/690-E (Extended)

| DC CHARACTERISTICS | | | Standard Operating Conditions (unless otherwise stated)<br>Operating temperature -40°C ≤ TA ≤ +85°C for industrial<br>-40°C ≤ TA ≤ +125°C for extended | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Sym** | **Characteristic** | **Min** | **Typ†** | **Max** | **Units** | **Conditions** |
| | VDD | **Supply Voltage** | | | | | |
| D001 | | | 2.0 | — | 5.5 | V | FOSC < = 4 MHz |
| D001C | | | 3.0 | — | 5.5 | V | FOSC < = 10 MHz |
| D001D | | | 4.5 | — | 5.5 | V | FOSC < = 20 MHz |
| D002 | VDR | **RAM Data Retention Voltage(1)** | 1.5* | — | — | V | Device in Sleep mode |
| D003 | VPOR | **VDD Start Voltage** to ensure internal Power-on Reset signal | — | VSS | — | V | See **Section 14.2.1 "Power-On Reset (POR)"** for details. |
| D004 | SVDD | **VDD Rise Rate** to ensure internal Power-on Reset signal | 0.05* | — | — | V/ms | See **Section 14.2.1 "Power-On Reset (POR)"** for details. |
| D005 | VBOR | **VDD Voltage Required** to initiate a Brown-Out Reset | 2.025 | — | 2.175 | V | |

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** This is the limit to which VDD can be lowered in Sleep mode without losing RAM data.

## 17.2 DC Characteristics: PIC16F685/687/689/690-I (Industrial)

| DC CHARACTERISTICS | | Standard Operating Conditions (unless otherwise stated)<br>Operating temperature -40°C ≤ TA ≤ +85°C for industrial | | | | | |
|---|---|---|---|---|---|---|---|
| Param No. | Device Characteristics | Min | Typ† | Max | Units | Conditions | |
| | | | | | | VDD | Note |
| D010 | Supply Current (IDD)(1, 2) | — | 9 | TBD | µA | 2.0 | FOSC = 32 kHz<br>LP Oscillator mode |
| | | — | 18 | TBD | µA | 3.0 | |
| | | — | 35 | TBD | µA | 5.0 | |
| D011 | | — | 110 | TBD | µA | 2.0 | FOSC = 1 MHz<br>XT Oscillator mode |
| | | — | 190 | TBD | µA | 3.0 | |
| | | — | 330 | TBD | µA | 5.0 | |
| D012 | | — | 220 | TBD | µA | 2.0 | FOSC = 4 MHz<br>XT Oscillator mode |
| | | — | 370 | TBD | µA | 3.0 | |
| | | — | 0.6 | TBD | mA | 5.0 | |
| D013 | | — | 70 | TBD | µA | 2.0 | FOSC = 1 MHz<br>EC Oscillator mode |
| | | — | 140 | TBD | µA | 3.0 | |
| | | — | 260 | TBD | µA | 5.0 | |
| D014 | | — | 180 | TBD | µA | 2.0 | FOSC = 4 MHz<br>EC Oscillator mode |
| | | — | 320 | TBD | µA | 3.0 | |
| | | — | 580 | TBD | µA | 5.0 | |
| D015 | | — | TBD | TBD | µA | 2.0 | FOSC = 31 kHz<br>INTOSC mode |
| | | — | TBD | TBD | µA | 3.0 | |
| | | — | TBD | TBD | mA | 5.0 | |
| D016 | | — | 340 | TBD | µA | 2.0 | FOSC = 8 MHz<br>INTOSC mode |
| | | — | 500 | TBD | µA | 3.0 | |
| | | — | 0.8 | TBD | mA | 5.0 | |
| D017 | | — | 180 | TBD | µA | 2.0 | FOSC = 4 MHz<br>EXTRC mode |
| | | — | 320 | TBD | µA | 3.0 | |
| | | — | 580 | TBD | µA | 5.0 | |
| D018 | | — | 2.1 | TBD | mA | 4.5 | FOSC = 20 MHz<br>HS Oscillator mode |
| | | — | 2.4 | TBD | mA | 5.0 | |

**Legend:** TBD = To Be Determined

† Data in 'Typ' column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** 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; $\overline{MCLR}$ = VDD; WDT disabled.

**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.

**3:** The peripheral current is the sum of the base IDD or IPD and the additional current consumed when this peripheral is enabled. The peripheral ∆ current can be determined by subtracting the base IDD or IPD current from this limit. Max values should be used when calculating total current consumption.

**4:** 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.

## 17.2 DC Characteristics: PIC16F685/687/689/690-I (Industrial) (Continued)

| DC CHARACTERISTICS | | Standard Operating Conditions (unless otherwise stated)<br>Operating temperature -40°C ≤ TA ≤ +85°C for industrial | | | | | |
|---|---|---|---|---|---|---|---|
| Param No. | Device Characteristics | Min | Typ† | Max | Units | Conditions | |
| | | | | | | VDD | Note |
| D020 | **Power-down Base Current (IPD)(4)** | — | 0.1 | TBD | µA | 2.0 | WDT, BOR, Comparators, VREF and T1OSC disabled |
| | | — | 0.4 | TBD | µA | 3.0 | |
| | | — | 0.8 | TBD | µA | 5.0 | |
| D021 | | — | 0.3 | TBD | µA | 2.0 | WDT Current |
| | | — | 1.8 | TBD | µA | 3.0 | |
| | | — | 8.4 | TBD | µA | 5.0 | |
| D022 | | — | 58 | TBD | µA | 3.0 | BOR Current |
| | | — | 109 | TBD | µA | 5.0 | |
| D023 | | — | 3.3 | TBD | µA | 2.0 | Comparator Current(3) |
| | | — | 6.1 | TBD | µA | 3.0 | |
| | | — | 11.5 | TBD | µA | 5.0 | |
| D024 | | — | 58 | TBD | µA | 2.0 | CVREF Current |
| | | — | 85 | TBD | µA | 3.0 | |
| | | — | 138 | TBD | µA | 5.0 | |
| D025 | | — | 4.0 | TBD | µA | 2.0 | T1OSC Current |
| | | — | 4.6 | TBD | µA | 3.0 | |
| | | — | 6.0 | TBD | µA | 5.0 | |
| D026 | | — | 1.2 | TBD | nA | 3.0 | A/D Current |
| | | — | 2.2 | TBD | nA | 5.0 | |
| D027 | | — | TBD | TBD | µA | 3.0 | VP6 Current |
| | | — | TBD | TBD | µA | 5.0 | |

**Legend:** TBD = To Be Determined

† Data in 'Typ' column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** 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; $\overline{MCLR}$ = VDD; WDT disabled.

**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.

**3:** The peripheral current is the sum of the base IDD or IPD and the additional current consumed when this peripheral is enabled. The peripheral Δ current can be determined by subtracting the base IDD or IPD current from this limit. Max values should be used when calculating total current consumption.

**4:** 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.

## 17.3 DC Characteristics: PIC16F685/687/689/690-E (Extended)

| DC CHARACTERISTICS | | Standard Operating Conditions (unless otherwise stated)<br>Operating temperature -40°C ≤ TA ≤ +125°C for extended | | | | | |
|---|---|---|---|---|---|---|---|
| Param No. | Device Characteristics | Min | Typ† | Max | Units | Conditions | |
| | | | | | | VDD | Note |
| D010E | **Supply Current (IDD)** | — | 9 | TBD | µA | 2.0 | FOSC = 32 kHz<br>LP Oscillator mode |
| | | — | 18 | TBD | µA | 3.0 | |
| | | — | 35 | TBD | µA | 5.0 | |
| D011E | | — | 110 | TBD | µA | 2.0 | FOSC = 1 MHz<br>XT Oscillator mode |
| | | — | 190 | TBD | µA | 3.0 | |
| | | — | 330 | TBD | µA | 5.0 | |
| D012E | | — | 220 | TBD | µA | 2.0 | FOSC = 4 MHz<br>XT Oscillator mode |
| | | — | 370 | TBD | µA | 3.0 | |
| | | — | 0.6 | TBD | mA | 5.0 | |
| D013E | | — | 70 | TBD | µA | 2.0 | FOSC = 1 MHz<br>EC Oscillator mode |
| | | — | 140 | TBD | µA | 3.0 | |
| | | — | 260 | TBD | µA | 5.0 | |
| D014E | | — | 180 | TBD | µA | 2.0 | FOSC = 4 MHz<br>EC Oscillator mode |
| | | — | 320 | TBD | µA | 3.0 | |
| | | — | 580 | TBD | µA | 5.0 | |
| D015E | | — | TBD | TBD | µA | 2.0 | FOSC = 31 kHz<br>INTOSC mode |
| | | — | TBD | TBD | µA | 3.0 | |
| | | — | TBD | TBD | mA | 5.0 | |
| D016E | | — | 340 | TBD | µA | 2.0 | FOSC = 8 MHz<br>INTOSC mode |
| | | — | 500 | TBD | µA | 3.0 | |
| | | — | 0.8 | TBD | mA | 5.0 | |
| D017E | | — | 180 | TBD | µA | 2.0 | FOSC = 4 MHz<br>EXTRC mode |
| | | — | 320 | TBD | µA | 3.0 | |
| | | — | 580 | TBD | µA | 5.0 | |
| D018E | | — | 2.1 | TBD | mA | 4.5 | FOSC = 20 MHz<br>HS Oscillator mode |
| | | — | 2.4 | TBD | mA | 5.0 | |

**Legend:** TBD = To Be Determined

† Data in 'Typ' column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** 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; $\overline{MCLR}$ = VDD; WDT disabled.

**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.

**3:** The peripheral current is the sum of the base IDD or IPD and the additional current consumed when this peripheral is enabled. The peripheral ∆ current can be determined by subtracting the base IDD or IPD current from this limit. Max values should be used when calculating total current consumption.

**4:** 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.

## 17.3 DC Characteristics: PIC16F685/687/689/690-E (Extended)

| DC CHARACTERISTICS | | Standard Operating Conditions (unless otherwise stated)<br>Operating temperature -40°C ≤ TA ≤ +125°C for extended | | | | | |
|---|---|---|---|---|---|---|---|
| Param No. | Device Characteristics | Min | Typ† | Max | Units | Conditions | |
| | | | | | | VDD | Note |
| D020E | **Power-down Base Current (IPD)(4)** | — | 0.1 | TBD | µA | 2.0 | WDT, BOR, Comparators, VREF and T1OSC disabled |
| | | — | 0.4 | TBD | µA | 3.0 | |
| | | — | 0.8 | TBD | µA | 5.0 | |
| D021E | | — | 0.3 | TBD | µA | 2.0 | WDT Current |
| | | — | 1.8 | TBD | µA | 3.0 | |
| | | — | 8.4 | TBD | µA | 5.0 | |
| D022E | | — | 58 | TBD | µA | 3.0 | BOR Current |
| | | — | 109 | TBD | µA | 5.0 | |
| D023E | | — | 3.3 | TBD | µA | 2.0 | Comparator Current(3) |
| | | — | 6.1 | TBD | µA | 3.0 | |
| | | — | 11.5 | TBD | µA | 5.0 | |
| D024E | | — | 58 | TBD | µA | 2.0 | CVREF Current |
| | | — | 85 | TBD | µA | 3.0 | |
| | | — | 138 | TBD | µA | 5.0 | |
| D025E | | — | 4.0 | TBD | µA | 2.0 | T1OSC Current |
| | | — | 4.6 | TBD | µA | 3.0 | |
| | | — | 6.0 | TBD | µA | 5.0 | |
| D026E | | — | 1.2 | TBD | nA | 3.0 | A/D Current(3) |
| | | — | 2.2 | TBD | nA | 5.0 | |
| D027E | | — | TBD | TBD | µA | 3.0 | VP6 Current |
| | | — | TBD | TBD | µA | 5.0 | |

**Legend:** TBD = To Be Determined

† Data in 'Typ' column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** 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; $\overline{MCLR}$ = VDD; WDT disabled.

**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.

**3:** The peripheral current is the sum of the base IDD or IPD and the additional current consumed when this peripheral is enabled. The peripheral Δ current can be determined by subtracting the base IDD or IPD current from this limit. Max values should be used when calculating total current consumption.

**4:** 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.

## 17.4 DC Characteristics: PIC16F685/687/689/690-I (Industrial) PIC16F685/687/689/690-E (Extended)

| DC CHARACTERISTICS | | | Standard Operating Conditions (unless otherwise stated)<br>Operating temperature -40°C ≤ TA ≤ +85°C for industrial<br>-40°C ≤ TA ≤ +125°C for extended | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Sym** | **Characteristic** | **Min** | **Typ†** | **Max** | **Units** | **Conditions** |
| | VIL | **Input Low Voltage** | | | | | |
| | | I/O port: | | | | | |
| D030 | | with TTL buffer | Vss | — | 0.8 | V | 4.5V ≤ VDD ≤ 5.5V |
| D030A | | | Vss | — | 0.15 VDD | V | Otherwise |
| D031 | | with Schmitt Trigger buffer | Vss | — | 0.2 VDD | V | Entire range |
| D032 | | $\overline{MCLR}$, OSC1 (RC mode) | Vss | — | 0.2 VDD | V | |
| D033 | | OSC1 (XT and HS modes)(1) | Vss | — | 0.3 VDD | V | |
| D033A | | OSC1 (LP mode)(1) | Vss | — | 0.6 VDD – 1.0 | V | |
| D0033B | | OCS1 (ER mode)(1) | Vss | — | 0.1 VDD | V | |
| | VIH | **Input High Voltage** | | | | | |
| | | I/O port: | | — | | | |
| D040 | | with TTL buffer | 2.0 | — | VDD | V | 4.5V ≤ VDD ≤ 5.5V |
| D040A | | | (0.25 VDD + 0.8) | — | VDD | V | Otherwise |
| D041 | | with Schmitt Trigger buffer | 0.8 VDD | — | VDD | V | Entire range |
| D042 | | $\overline{MCLR}$, $\overline{PORTA}$ | 0.8 VDD | — | VDD | V | |
| D043 | | OSC1 (XT, HS and LP modes) | 0.7 VDD | — | VDD | V | **(Note 1)** |
| D043A | | OSC1 (ER mode) | 0.9 VDD | — | VDD | V | **(Note 1)** |
| D070 | IPUR | **PORTA Weak Pull-up Current**<br>**PORTB Weak Pull-up Current** | 50*<br>50* | 250<br>250 | 400*<br>400* | µA<br>µA | VDD = 5.0V, VPIN = VSS |
| | IIL | **Input Leakage Current(2)** | | | | | |
| D060 | | I/O port | — | — | ± 1 | µA | VSS ≤ VPIN ≤ VDD,<br>Pin at high-impedance |
| D061 | | $\overline{MCLR}$(3) | — | — | ± 5 | µA | VSS ≤ VPIN ≤ VDD |
| D063 | | OSC1 | — | — | ± 5 | µA | VSS ≤ VPIN ≤ VDD, XT, HS and LP osc configuration |
| | VOL | **Output Low Voltage** | | | | | |
| D080 | | I/O port | — | — | 0.6 | V | IOL = 8.5 mA, VDD = 4.5V |
| D083 | | OSC2/CLKOUT | — | — | 0.6 | V | IOL = 1.6 mA, VDD = 4.5V |

* These parameters are characterized but not tested.

† Data in 'Typ' column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** In RC oscillator configuration, the OSC1/CLKIN pin is a Schmitt Trigger input. It is not recommended to use an external clock in RC mode.

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

**3:** The leakage current on the $\overline{MCLR}$ pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.

**4:** See **Section 10.0 "Data EEPROM and Flash Program Memory Control"** for additional information.

## 17.4 DC Characteristics: PIC16F685/687/689/690-I (Industrial) PIC16F685/687/689/690-E (Extended) (Continued)

| DC CHARACTERISTICS | | | Standard Operating Conditions (unless otherwise stated)<br>Operating temperature -40°C ≤ TA ≤ +85°C for industrial<br>-40°C ≤ TA ≤ +125°C for extended | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Sym** | **Characteristic** | **Min** | **Typ†** | **Max** | **Units** | **Conditions** |
| | VOH | **Output High Voltage** | | | | | |
| D090 | | I/O port | VDD – 0.7 | — | — | V | IOH = -3.0 mA, VDD = 4.5V |
| D092 | | OSC2/CLKOUT | VDD – 0.7 | — | — | V | IOH = -1.3 mA, VDD = 4.5V |
| D100 | IULP | **Ultra Low-Power Wake-up Current** | — | 200 | — | nA | |
| | | **Capacitive Loading Specs on Output Pins** | | | | | |
| D100 | COSC2 | OSC2 pin | — | — | 15* | pF | In XT, HS and LP modes when external clock is used to drive OSC1 |
| D101 | CIO | All I/O pins | — | — | 50* | pF | |
| | | **Data EEPROM Memory** | | | | | |
| D120 | ED | Byte Endurance | 100K | 1M | — | E/W | -40°C ≤ TA ≤ +85°C |
| D120A | ED | Byte Endurance | 10K | 100K | — | E/W | +85°C ≤ TA ≤ +125°C |
| D121 | VDRW | VDD for Read/Write | VMIN | — | 5.5 | V | Using EECON1 to read/write VMIN = Minimum operating voltage |
| D122 | TDEW | Erase/Write Cycle Time | — | 5 | 6 | ms | |
| D123 | TRETD | Characteristic Retention | 40 | — | — | Year | Provided no other specifications are violated |
| D124 | TREF | Number of Total Erase/Write Cycles before Refresh(4) | 1M | 10M | — | E/W | -40°C ≤ TA ≤ +85°C |
| | | **Program Flash Memory** | | | | | |
| D130 | EP | Cell Endurance | 10K | 100K | — | E/W | -40°C ≤ TA ≤ +85°C |
| D130A | ED | Cell Endurance | 1K | 10K | — | E/W | +85°C ≤ TA ≤ +125°C |
| D131 | VPR | VDD for Read | VMIN | — | 5.5 | V | VMIN = Minimum operating voltage |
| D132 | VPEW | VDD for Erase/Write | 4.5 | — | 5.5 | V | |
| D133 | TPEW | Erase/Write cycle time | — | 2 | 2.5 | ms | |
| D134 | TRETD | Characteristic Retention | 40 | — | — | Year | Provided no other specifications are violated |

* These parameters are characterized but not tested.

† Data in 'Typ' column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** In RC oscillator configuration, the OSC1/CLKIN pin is a Schmitt Trigger input. It is not recommended to use an external clock in RC mode.

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

**3:** The leakage current on the $\overline{\text{MCLR}}$ pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.

**4:** See **Section 10.0 "Data EEPROM and Flash Program Memory Control"** for additional information.

## 17.5 Timing Parameter Symbology

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

1. TppS2ppS
2. TppS

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

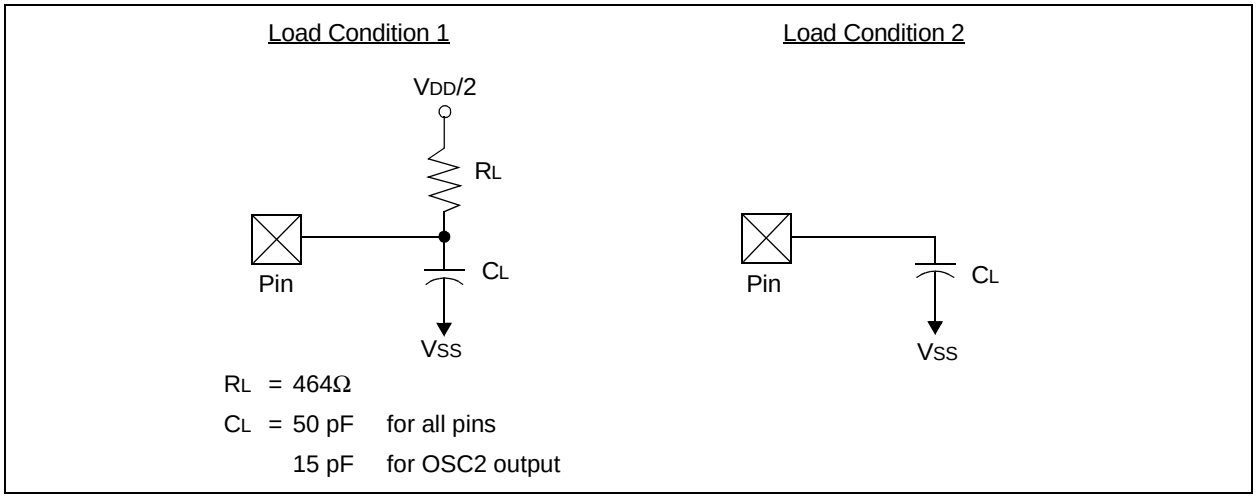
Lowercase letters (pp) and their meanings:

| **pp** | | | |
|---|---|---|---|
| cc | RC | 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 |

**FIGURE 17-2: LOAD CONDITIONS**



## 17.6 AC Characteristics: PIC16F685/687/689/690 (Industrial, Extended)

**FIGURE 17-3: EXTERNAL CLOCK TIMING**

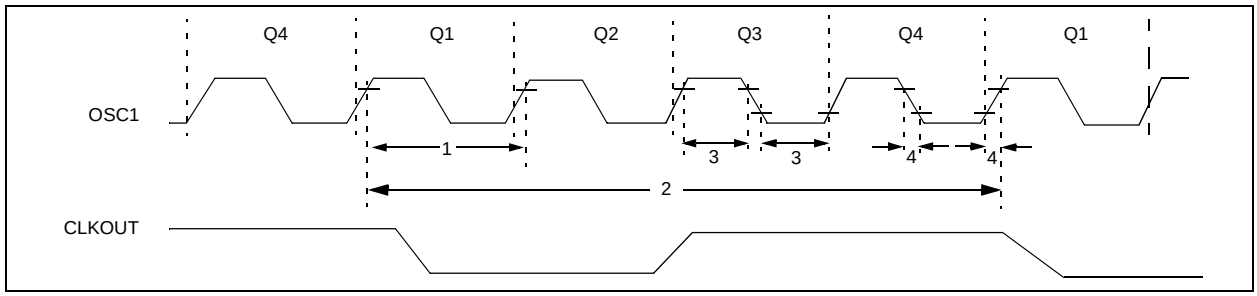



**TABLE 17-1: EXTERNAL CLOCK TIMING REQUIREMENTS**

**Standard Operating Conditions (unless otherwise stated)**
Operating Temperature -40°C ≤ TA ≤ +125°C

| Param No. | Sym | Characteristic | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|
| | FOSC | External CLKIN Frequency(1) | DC | — | 37 | kHz | LP Oscillator mode |
| | | | DC | — | 4 | MHz | XT Oscillator mode |
| | | | DC | — | 20 | MHz | HS Oscillator mode |
| | | | DC | — | 20 | MHz | EC Oscillator mode |
| | | Oscillator Frequency(1) | — | 8 | — | MHz | INTOSC Oscillator mode |
| | | | TBD | — | 4 | MHz | RC Oscillator mode |
| | | | — | 32 | — | kHz | LP Oscillator mode |
| | | | 0.1 | — | 4 | MHz | XT Oscillator mode |
| | | | 1 | — | 20 | MHz | HS Oscillator mode |
| 1 | TOSC | External CLKIN Period(1) | 27 | — | ∞ | µs | LP Oscillator mode |
| | | | 50 | — | ∞ | ns | HS Oscillator mode |
| | | | 50 | — | ∞ | ns | EC Oscillator mode |
| | | | 250 | — | ∞ | ns | XT Oscillator mode |
| | | Oscillator Period(1) | — | 31 | — | µs | LP Oscillator mode |
| | | | — | 125 | — | ns | INTOSC Oscillator mode |
| | | | 250 | — | TBD | ns | RC Oscillator mode |
| | | | 250 | — | 10,000 | ns | XT Oscillator mode |
| | | | 50 | — | 1,000 | ns | HS Oscillator mode |
| 2 | TCY | Instruction Cycle Time(1) | 200 | TCY | ∞ | ns | TCY = 4/FOSC |
| 3 | TosL, TosH | External CLKIN (OSC1) High | 2* | — | — | µs | LP oscillator, TOSC L/H duty cycle |
| | | External CLKIN Low | 20* | — | — | ns | HS oscillator, TOSC L/H duty cycle |
| | | | 100 * | — | — | ns | XT oscillator, TOSC L/H duty cycle |
| 4 | TosR, TosF | External CLKIN Rise | — | — | 50* | ns | LP oscillator |
| | | External CLKIN Fall | — | — | 25* | ns | XT oscillator |
| | | | — | — | 15* | ns | HS oscillator |

**Legend:** TBD = To Be Determined

* 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.

**Note 1:** Instruction cycle period (TCY) equals four times the input oscillator time base period. All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at 'min' values with an external clock applied to OSC1 pin. When an external clock input is used, the 'max' cycle time limit is 'DC' (no clock) for all devices.

**TABLE 17-2: PRECISION INTERNAL OSCILLATOR PARAMETERS**

| **Standard Operating Conditions (unless otherwise stated)**<br>Operating Temperature -40°C ≤ TA ≤ +125°C | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Param No.** | **Sym** | **Characteristic** | **Freq Tolerance** | **Min** | **Typ†** | **Max** | **Units** | **Conditions** |
| F10 | FOSC | Internal Calibrated INTOSC Frequency(1) | ±1% | — | 8.00 | TBD | MHz | VDD and Temperature TBD |
| | | | ±2% | — | 8.00 | TBD | MHz | 2.5V ≤ VDD ≤ 5.5V<br>0°C ≤ TA ≤ +85°C |
| | | | ±5% | — | 8.00 | TBD | MHz | 2.0V ≤ VDD ≤ 5.5V<br>-40°C ≤ TA ≤ +85°C (Ind.)<br>-40°C ≤ TA ≤ +125°C (Ext.) |
| F14 | TIOSCST | Oscillator Wake-up from Sleep Start-up Time* | — | — | TBD | TBD | µs | VDD = 2.0V, -40°C to +85°C |
| | | | — | — | TBD | TBD | µs | VDD = 3.0V, -40°C to +85°C |
| | | | — | — | TBD | TBD | µs | VDD = 5.0V, -40°C to +85°C |

**Legend:** TBD = To Be Determined

* These parameters are characterized but not tested.

† Data in 'Typ' column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** To ensure these oscillator frequency tolerances, VDD and VSS must be capacitively decoupled as close to the device as possible. 0.1 µF and 0.01 µF values in parallel are recommended.

**FIGURE 17-4: CLKOUT AND I/O TIMING**

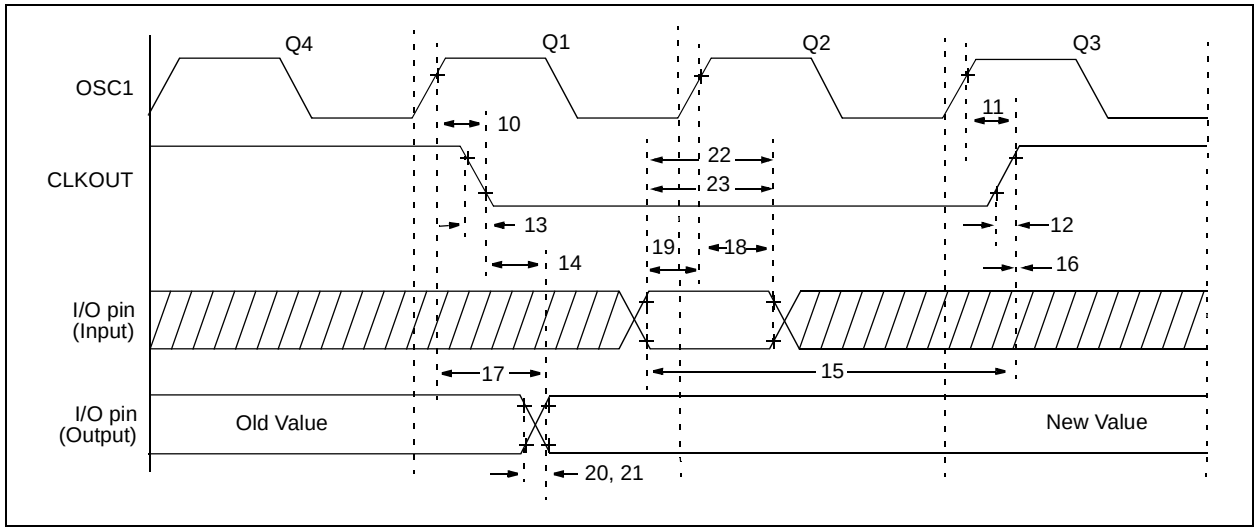



**TABLE 17-3: CLKOUT AND I/O TIMING REQUIREMENTS**

| **Standard Operating Conditions (unless otherwise stated)**<br>Operating Temperature -40°C ≤ TA ≤ +125°C | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Sym** | **Characteristic** | **Min** | **Typ†** | **Max** | **Units** | **Conditions** |
| 10 | TOSH2CKL | OSC1↑ to CLOUT↓ | — | 75 | 200 | ns | **(Note 1)** |
| 11 | TOSH2CKH | OSC1↑ to CLOUT↑ | — | 75 | 200 | ns | **(Note 1)** |
| 12 | TCKR | CLKOUT Rise Time | — | 35 | 100 | ns | **(Note 1)** |
| 13 | TCKF | CLKOUT Fall Time | — | 35 | 100 | ns | **(Note 1)** |
| 14 | TCKL2IOV | CLKOUT↓ to Port Out Valid | — | — | 20 | ns | **(Note 1)** |
| 15 | TIOV2CKH | Port In Valid before CLKOUT↑ | TOSC + 200 ns | — | — | ns | **(Note 1)** |
| 16 | TCKH2IOI | Port In Hold after CLKOUT↑ | 0 | — | — | ns | **(Note 1)** |
| 17 | TOSH2IOV | OSC1↑ (Q1 cycle) to Port Out Valid | — | 50 | 150* | ns | |
| | | | — | — | 300 | ns | |
| 18 | TOSH2IOI | OSC1↑ (Q2 cycle) to Port Input Invalid (I/O in hold time) | 100 | — | — | ns | |
| 19 | TIOV2OSH | Port Input Valid to OSC1↑ (I/O in setup time) | 0 | — | — | ns | |
| 20 | TIOR | Port Output Rise Time | — | 10 | 40 | ns | |
| 21 | TIOF | Port Output Fall Time | — | 10 | 40 | ns | |
| 22 | TINP | INT Pin High or Low Time | 25 | — | — | ns | |
| 23 | TRBP | PORTA change INT high or low time | TCY | — | — | ns | |

* These parameters are characterized but not tested.

† Data in 'Typ' column is at 5.0V, 25°C unless otherwise stated.

**Note 1:** Measurements are taken in RC mode where CLKOUT output is 4 x TOSC.

**FIGURE 17-5: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING**

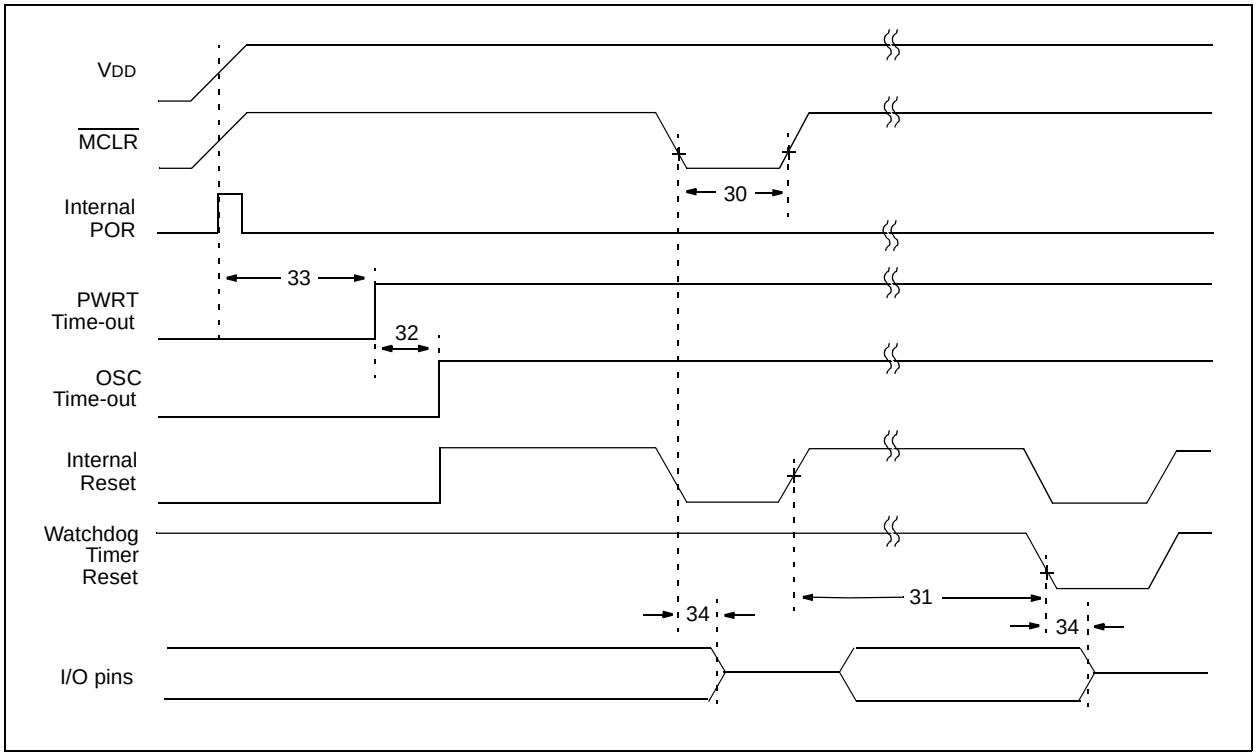
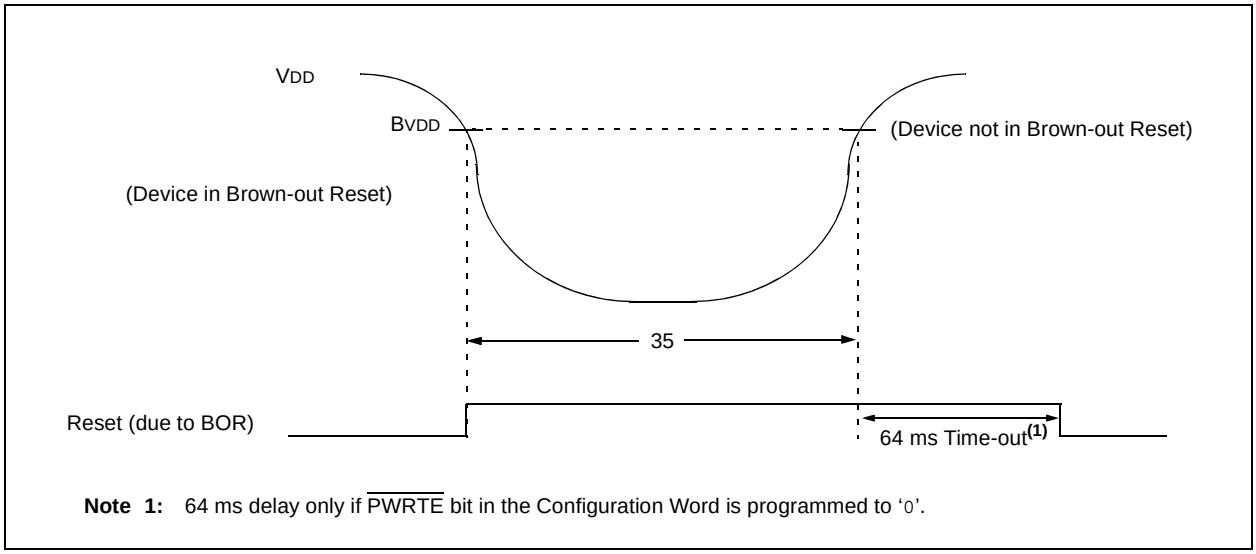



**FIGURE 17-6: BROWN-OUT RESET TIMING AND CHARACTERISTICS**




**Note 1:** 64 ms delay only if $\overline{\text{PWRTE}}$ bit in the Configuration Word is programmed to '0'.

**TABLE 17-4: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER, POWER-UP TIMER AND BROWN-OUT RESET REQUIREMENTS**

| **Standard Operating Conditions (unless otherwise stated)**<br>Operating Temperature-40°C ≤ TA ≤ +125°C | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Sym** | **Characteristic** | **Min** | **Typ†** | **Max** | **Units** | **Conditions** |
| 30 | TMCL | $\overline{\text{MCLR}}$ Pulse Width (low) | 2<br>11 | —<br>18 | —<br>24 | µs<br>ms | VDD = 5V, -40°C to +85°C<br>Extended temperature |
| 31 | TWDT | Watchdog Timer Time-out Period (No Prescaler) | 7<br>10 | 18<br>17 | 33<br>30 | ms<br>ms | VDD = 5V, -40°C to +85°C<br>Extended temperature |
| 32 | TOST | Oscillation Start-up Timer Period | — | 1024TOSC | — | — | TOSC = OSC1 period |
| 33* | TPWRT | Power-up Timer Period | 28*<br>TBD | 64<br>TBD | 132*<br>TBD | ms<br>ms | VDD = 5V, -40°C to +85°C<br>Extended Temperature |
| 34 | TIOZ | I/O High-impedance from $\overline{\text{MCLR}}$ Low or Watchdog Timer Reset | — | — | 2.0 | µs | |
| | BVHY | Brown-out Reset Hysteresis | — | 25 | — | mV | |
| | BVDD | Brown-out Reset Voltage | 2.025 | — | 2.175 | V | |
| 35 | TBOR | Brown-out Reset Pulse Width | 100* | — | — | µs | VDD ≤ BVDD (D005) |

**Legend:** TBD = To Be Determined

* 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.

**FIGURE 17-7: TIMER0 AND TIMER1 EXTERNAL CLOCK TIMINGS**

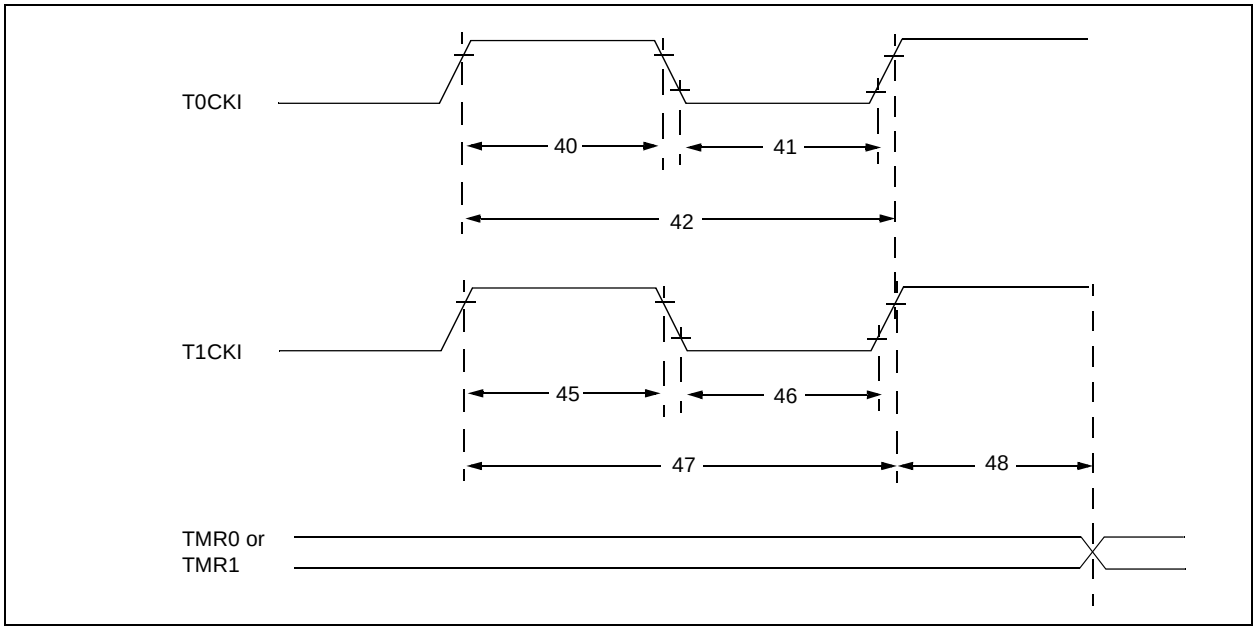



**TABLE 17-5: TIMER0 AND TIMER1 EXTERNAL CLOCK REQUIREMENTS**

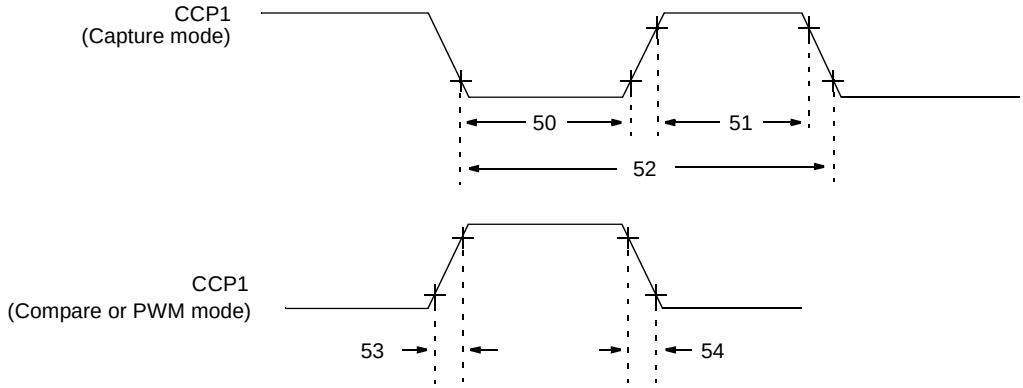
| Param No. | Sym | Characteristic | | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|---|
| 40* | TT0H | T0CKI High Pulse Width | No Prescaler | 0.5 TCY + 20 | — | — | ns | |
| | | | With Prescaler | 10 | — | — | ns | |
| 41* | TT0L | T0CKI Low Pulse Width | No Prescaler | 0.5 TCY + 20 | — | — | ns | |
| | | | With Prescaler | 10 | — | — | ns | |
| 42* | TT0P | T0CKI Period | | Greater of: 20 or TCY + 40 / N | — | — | ns | N = prescale value (2, 4, ..., 256) |
| 45* | TT1H | T1CKI High Time | Synchronous, No Prescaler | 0.5 TCY + 20 | — | — | ns | |
| | | | Synchronous, with Prescaler | 15 | — | — | ns | |
| | | | Asynchronous | 30 | — | — | ns | |
| 46* | TT1L | T1CKI Low Time | Synchronous, No Prescaler | 0.5 TCY + 20 | — | — | ns | |
| | | | Synchronous, with Prescaler | 15 | — | — | ns | |
| | | | Asynchronous | 30 | — | — | ns | |
| 47* | TT1P | T1CKI Input Period | Synchronous | Greater of: 30 or TCY + 40 / N | — | — | ns | N = prescale value (1, 2, 4, 8) |
| | | | Asynchronous | 60 | — | — | 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 | | 2 TOSC* | — | 7 TOSC* | — | |

* 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.

**FIGURE 17-8: CAPTURE/COMPARE/PWM+ TIMINGS (ECCP+)**




**Note:** Refer to Figure 17-2 for load conditions.

**TABLE 17-6: CAPTURE/COMPARE/PWM+ REQUIREMENTS (ECCP+)**

| **Standard Operating Conditions (unless otherwise stated)**<br>Operating Temperature -40°C ≤ TA ≤ +125°C | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Param No.** | **Symbol** | **Characteristic** | | **Min** | **Typ†** | **Max** | **Units** | **Conditions** |
| 50* | TccL | CCP1 Input Low Time | No Prescaler | 0.5TCY + 20 | — | — | ns | |
| | | | With Prescaler | 20 | — | — | ns | |
| 51* | TccH | CCP1 Input High Time | No Prescaler | 0.5TCY + 20 | — | — | ns | |
| | | | With Prescaler | 20 | — | — | ns | |
| 52* | TccP | CCP1 Input Period | | $\frac{3T_{CY} + 40}{N}$ | — | — | ns | N = prescale value (1, 4 or 16) |
| 53* | TccR | CCP1 Output Rise Time | | — | 10 | 25 | ns | |
| 54* | TccF | CCP1 Output Fall Time | | — | 10 | 25 | ns | |

* 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.

**TABLE 17-7: COMPARATOR SPECIFICATIONS**

| Comparator Specifications | | | Standard Operating Conditions (unless otherwise stated)<br>Operating temperature -40°C ≤ TA ≤ +125°C | | | | |
|---|---|---|---|---|---|---|---|
| **Param. No.** | **Sym** | **Characteristics** | **Min** | **Typ** | **Max** | **Units** | **Comments** |
| C01 | VOS | Input Offset Voltage | — | ± 5.0 | ± 10 | mV | |
| C02 | VCM | Input Common Mode Voltage | 0 | — | VDD - 1.5 | V | |
| C03 | CMRR | Common Mode Rejection Ratio | +55* | — | — | db | |
| C04 | TRT | Response Time(1) | — | 150 | 400* | ns | |
| C05 | TMC2COV | Comparator Mode Change to Output Valid | — | — | 10* | µs | |

* These parameters are characterized but not tested.

**Note 1:** Response time measured with one comparator input at (VDD - 1.5)/2 while the other input transitions from VSS to VDD – 1.5V.

**TABLE 17-8: COMPARATOR VOLTAGE REFERENCE (CVREF) SPECIFICATIONS**

| Comparator Voltage Reference Specifications | | | Standard Operating Conditions (unless otherwise stated)<br>Operating temperature -40°C ≤ TA ≤ +125°C | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Symbol** | **Characteristics** | **Min** | **Typ** | **Max** | **Units** | **Comments** |
| CV01 | CVRES | Resolution | VDD/24* | — | VDD/32* | LSb | |
| CV02 | | Absolute Accuracy | —<br>— | —<br>— | ±1/4*<br>±1/2* | LSb<br>LSb | Low Range (VRR = 1)<br>High Range (VRR = 0) |
| CV03 | | Unit Resistor Value (R) | — | 2K* | — | Ω | |
| CV04 | | R Ladder Settling Time(1) | — | — | 10* | µs | |
| CV05 | | VP6 Settling Time | TBD | TBD | TBD | | |

**Legend:** TBD = To Be Determined

* These parameters are characterized but not tested.

**Note 1:** Settling time measured while VRR = 1 and VR<3:0> transitions from 0000 to 1111.

**TABLE 17-9: VOLTAGE (VR) REFERENCE SPECIFICATIONS**

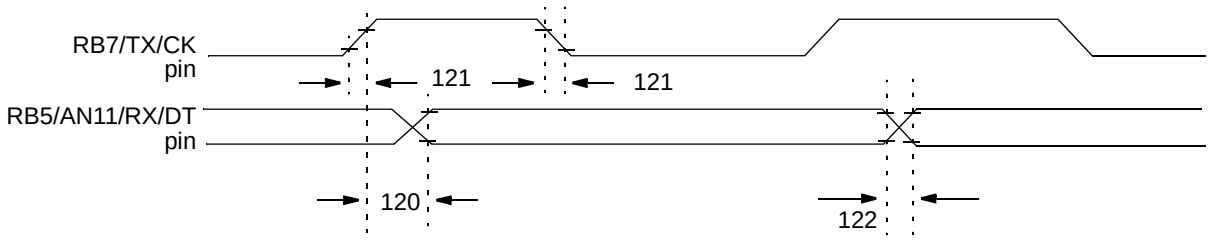
| VR Voltage Reference Specifications | | | Standard Operating Conditions (unless otherwise stated)<br>Operating temperature -40°C ≤ TA ≤ +125°C | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Symbol** | **Characteristics** | **Min** | **Typ** | **Max** | **Units** | **Comments** |
| VR01 | VROUT | VR voltage output | TBD | 0.6 | TBD | V | |
| VR02 | TCVOUT | Voltage drift temperature coefficient | — | 150 | TBD | ppm/°C | |
| VR03 | ΔVROUT/ ΔVDD | Voltage drift with respect to VDD regulation | — | 200 | — | µV/V | |
| VR04 | TSTABLE | Settling Time | — | 10 | 100* | µs | |

**Legend:** TBD = To Be Determined

* These parameters are characterized but not tested.

**FIGURE 17-9: EUSART SYNCHRONOUS TRANSMISSION (MASTER/SLAVE) TIMING**

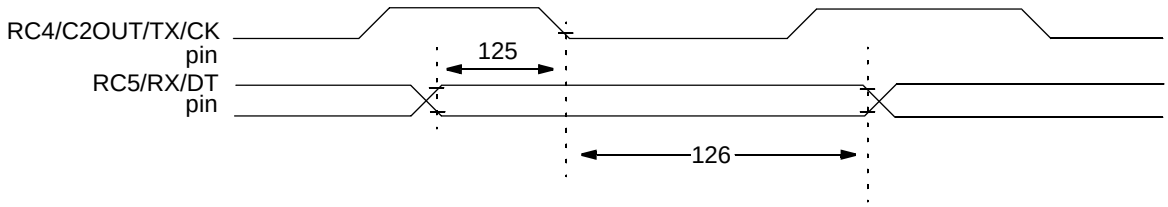



**Note:** Refer to Figure 17-2 for load conditions.

**TABLE 17-10: EUSART SYNCHRONOUS TRANSMISSION REQUIREMENTS**

| **Standard Operating Conditions (unless otherwise stated)**<br>Operating Temperature -40°C ≤ TA ≤ +125°C | | | | | | |
|---|---|---|---|---|---|---|
| **Param. No.** | **Symbol** | **Characteristic** | **Min** | **Max** | **Units** | **Conditions** |
| 120 | TCKH2DTV | SYNC XMIT (Master & Slave)<br>Clock high to data-out valid | — | 40 | ns | |
| 121 | TCKRF | Clock out rise time and fall time (Master mode) | — | 20 | ns | |
| 122 | TDTRF | Data-out rise time and fall time | — | 20 | ns | |

**FIGURE 17-10: EUSART SYNCHRONOUS RECEIVE (MASTER/SLAVE) TIMING**

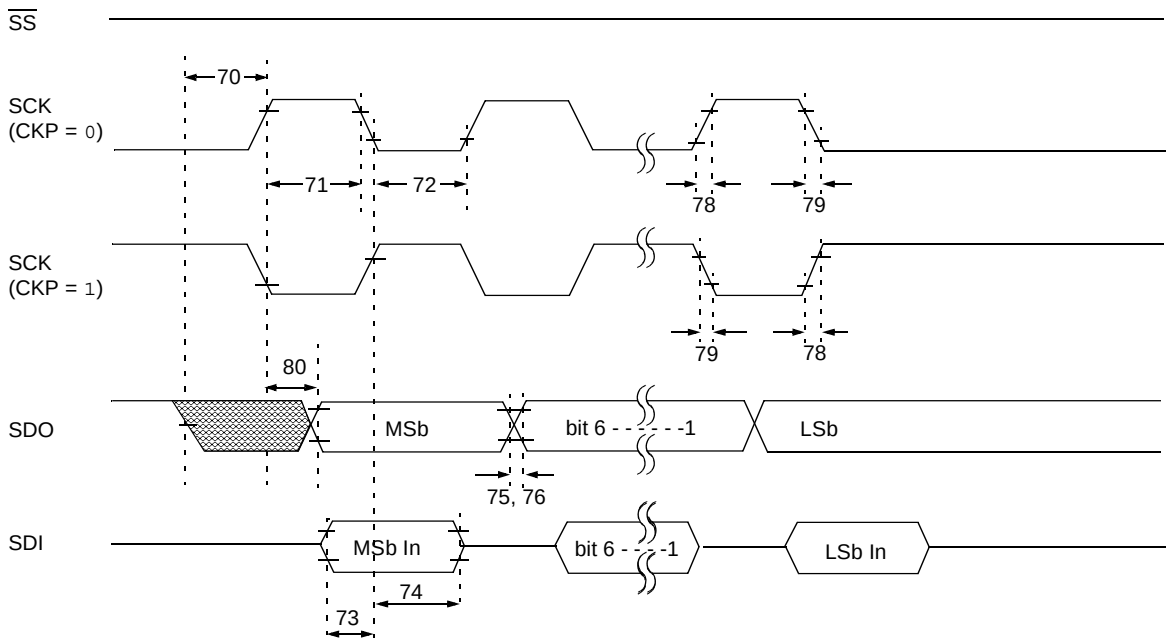



**Note:** Refer to Figure 17-2 for load conditions.

**TABLE 17-11: EUSART SYNCHRONOUS RECEIVE REQUIREMENTS**

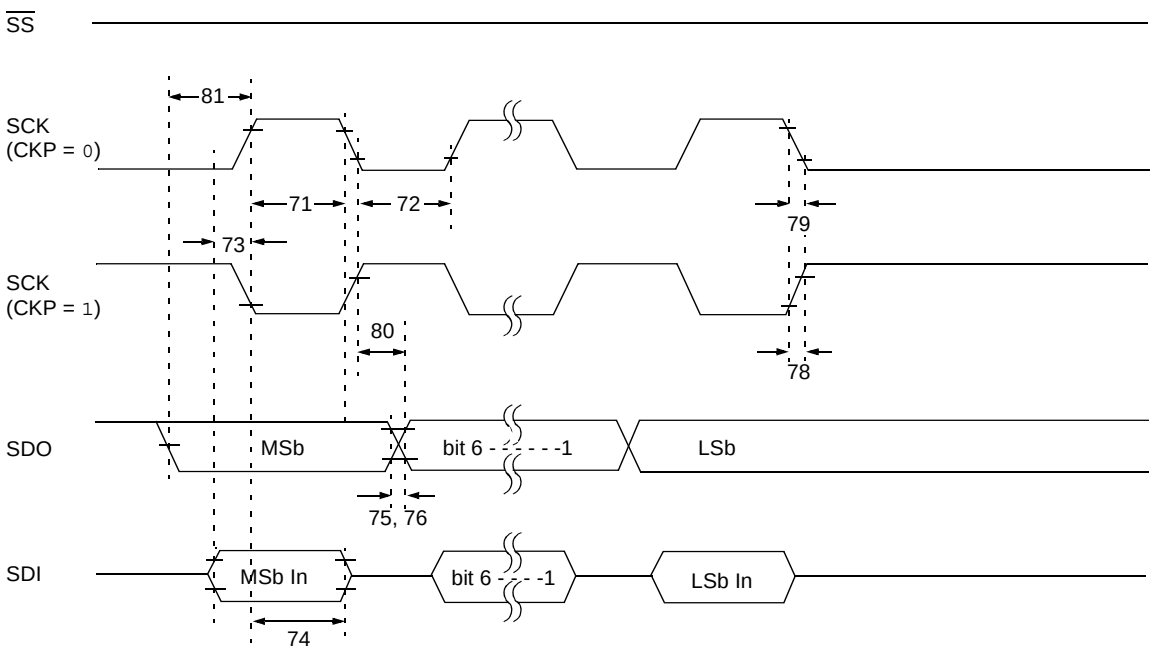
| **Standard Operating Conditions (unless otherwise stated)**<br>Operating Temperature -40°C ≤ TA ≤ +125°C | | | | | | |
|---|---|---|---|---|---|---|
| **Param. No.** | **Symbol** | **Characteristic** | **Min** | **Max** | **Units** | **Conditions** |
| 125 | TDTV2CKL | SYNC RCV (Master & Slave)<br>Data-hold before CK ↓ (DT hold time) | 10 | — | ns | |
| 126 | TCKL2DTL | Data-hold after CK ↓ (DT hold time) | 15 | — | ns | |

**FIGURE 17-11: SPI™ MASTER MODE TIMING (CKE = 0, SMP = 0)**




**Note:** Refer to Figure 17-2 for load conditions.

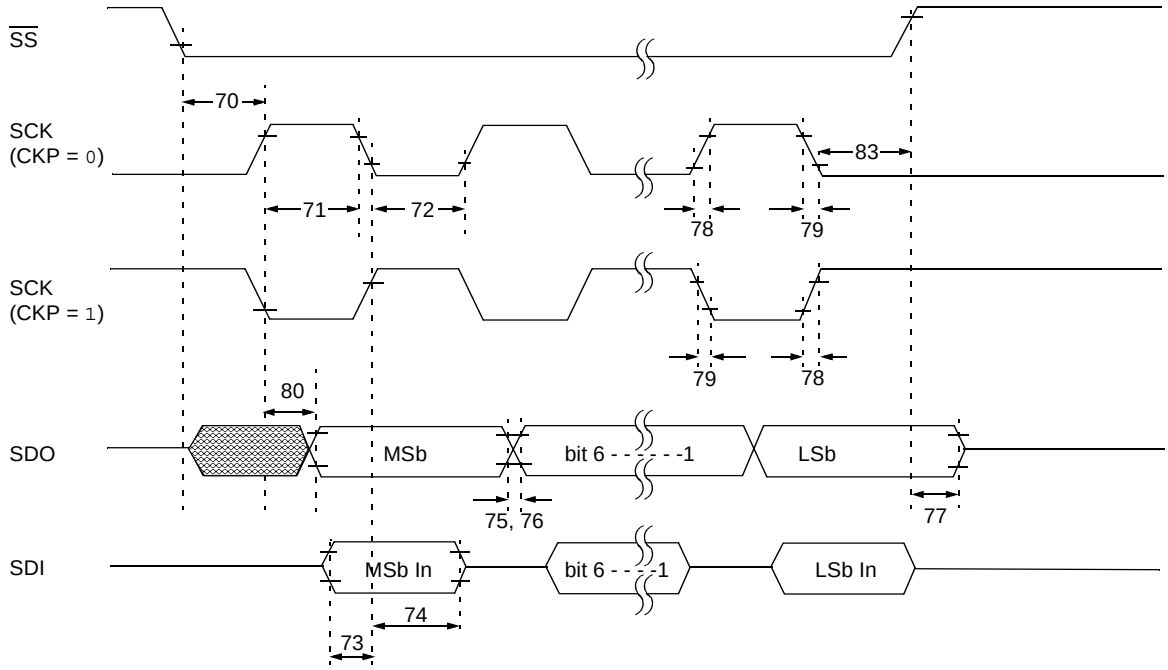
**FIGURE 17-12: SPI™ MASTER MODE TIMING (CKE = 1, SMP = 1)**




**Note:** Refer to Figure 17-2 for load conditions.

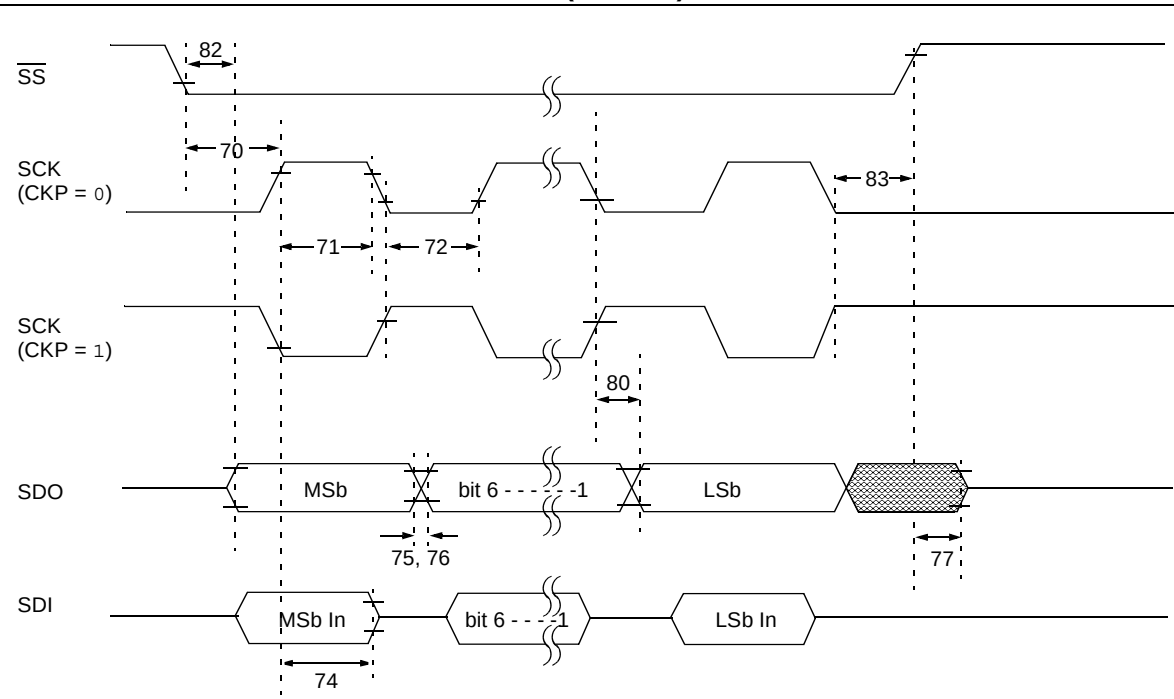
 

**FIGURE 17-13: SPI™ SLAVE MODE TIMING (CKE = 0)**




**Note:** Refer to Figure 17-2 for load conditions.

**FIGURE 17-14: SPI™ SLAVE MODE TIMING (CKE = 1)**




**Note:** Refer to Figure 17-2 for load conditions.

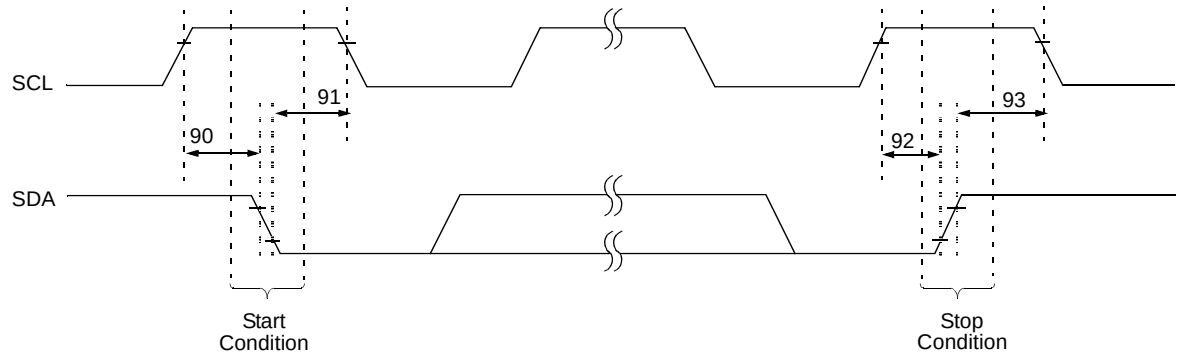
**TABLE 17-12: SPI™ MODE REQUIREMENTS**

| Param No. | Symbol | Characteristic | | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|---|
| 70* | TssL2scH, TssL2scL | $\overline{SS}\downarrow$ to SCK$\downarrow$ or SCK$\uparrow$ input | | Tcy | — | — | ns | |
| 71* | TscH | SCK input high time (Slave mode) | | Tcy + 20 | — | — | ns | |
| 72* | TscL | SCK input low time (Slave mode) | | Tcy + 20 | — | — | ns | |
| 73* | TdiV2scH, TdiV2scL | Setup time of SDI data input to SCK edge | | 100 | — | — | ns | |
| 74* | TscH2diL, TscL2diL | Hold time of SDI data input to SCK edge | | 100 | — | — | ns | |
| 75* | TdoR | SDO data output rise time | 3.0-5.5V | — | 10 | 25 | ns | |
| | | | 2.0-5.5V | — | 25 | 50 | ns | |
| 76* | TdoF | SDO data output fall time | | — | 10 | 25 | ns | |
| 77* | TssH2doZ | $\overline{SS}\uparrow$ to SDO output high-impedance | | 10 | — | 50 | ns | |
| 78* | TscR | SCK output rise time (Master mode) | 3.0-5.5V | — | 10 | 25 | ns | |
| | | | 2.0-5.5V | — | 25 | 50 | ns | |
| 79* | TscF | SCK output fall time (Master mode) | | — | 10 | 25 | ns | |
| 80* | TscH2doV, TscL2doV | SDO data output valid after SCK edge | 3.0-5.5V | — | — | 50 | ns | |
| | | | 2.0-5.5V | — | — | 145 | ns | |
| 81* | TdoV2scH, TdoV2scL | SDO data output setup to SCK edge | | Tcy | — | — | ns | |
| 82* | TssL2doV | SDO data output valid after $\overline{SS}\downarrow$ edge | | — | — | 50 | ns | |
| 83* | TscH2ssH, TscL2ssH | $\overline{SS}\uparrow$ after SCK edge | | 1.5Tcy + 40 | — | — | ns | |

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**FIGURE 17-15: $I^2C$™ BUS START/STOP BITS TIMING**




**Note**: Refer to Figure 17-2 for load conditions.

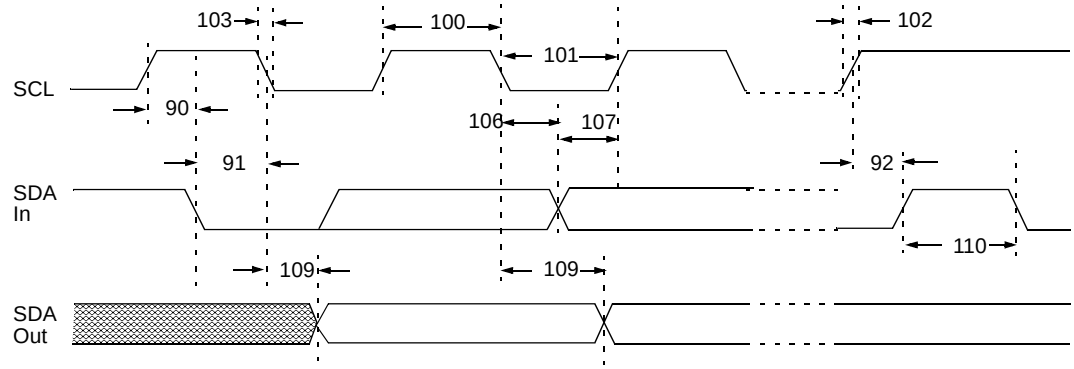
 

**TABLE 17-13: I²C™ BUS START/STOP BITS REQUIREMENTS**

| Param No. | Symbol | Characteristic | | Min | Typ | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|---|
| 90* | TSU:STA | Start condition Setup time | 100 kHz mode | 4700 | — | — | ns | Only relevant for Repeated Start condition |
| | | | 400 kHz mode | 600 | — | — | | |
| 91* | THD:STA | Start condition Hold time | 100 kHz mode | 4000 | — | — | ns | After this period, the first clock pulse is generated |
| | | | 400 kHz mode | 600 | — | — | | |
| 92* | TSU:STO | Stop condition Setup time | 100 kHz mode | 4700 | — | — | ns | |
| | | | 400 kHz mode | 600 | — | — | | |
| 93 | THD:STO | Stop condition Hold time | 100 kHz mode | 4000 | — | — | ns | |
| | | | 400 kHz mode | 600 | — | — | | |

* These parameters are characterized but not tested.

**FIGURE 17-16: I²C™ BUS DATA TIMING**




**Note:** Refer to Figure 17-2 for load conditions.

**TABLE 17-14: 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 | 1.5TCY | — | | |
| 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 | 1.5TCY | — | | |
| 102* | TR | SDA and SCL rise time | 100 kHz mode | — | 1000 | ns | |
| | | | 400 kHz mode | 20 + 0.1CB | 300 | ns | CB is specified to be from 10-400 pF |
| 103* | TF | SDA and SCL fall time | 100 kHz mode | — | 300 | ns | |
| | | | 400 kHz mode | 20 + 0.1CB | 300 | ns | CB 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 TR max. + TSU:DAT = 1000 + 250 = 1250 ns (according to the Standard mode I²C bus specification), before the SCL line is released.

**TABLE 17-15: PIC16F685/687/689/690 A/D CONVERTER CHARACTERISTICS**:

| **Standard Operating Conditions (unless otherwise stated)**<br>Operating Temperature -40°C ≤ TA ≤ +125°C | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Sym** | **Characteristic** | **Min** | **Typ†** | **Max** | **Units** | **Conditions** |
| A01 | NR | Resolution | — | — | 10 bits | bit | |
| A03 | EIL | Integral Error | — | — | ±1 | LSb | VREF = 5.0V |
| A04 | EDL | Differential Error | — | — | ±1 | LSb | No missing codes to 10 bits<br>VREF = 5.0V |
| A05 | EFS | Full-scale Range | 2.2* | — | 5.5* | V | |
| A06 | EOFF | Offset Error | — | — | ±1 | LSb | VREF = 5.0V |
| A07 | EGN | Gain Error | — | — | ±1 | LSb | VREF = 5.0V |
| A10 | — | Monotonicity | — | guaranteed(1) | — | — | VSS ≤ VAIN ≤ VREF+ |
| A20 | VREF | Reference Voltage | 2.0 | — | VDD + 0.3 | V | |
| A25 | VAIN | Analog Input Voltage | VSS | — | VREF | V | |
| A30 | ZAIN | Recommended Impedance of Analog Voltage Source | — | — | 10 | kΩ | |
| A50 | IREF | VREF Input Current*(2) | — | — | ±5 | µA | During VAIN acquisition. |
| | | | — | — | ±150 | µA | During A/D conversion cycle. |

* These parameters are characterized but not tested.

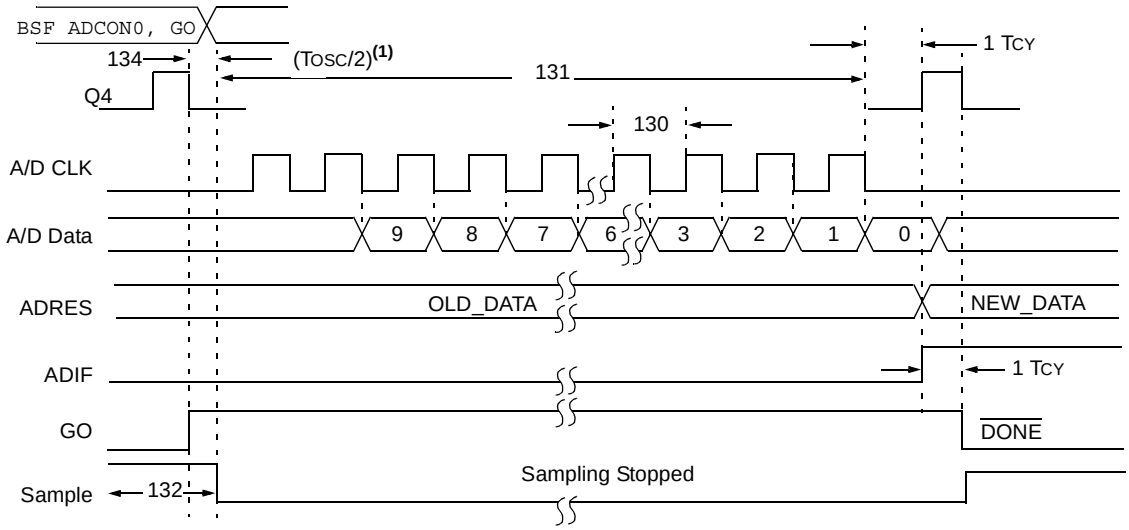
† Data in 'Typ' column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** The A/D conversion result never decreases with an increase in the input voltage and has no missing codes.

**2:** VREF current is from external VREF or VDD pin, whichever is selected as reference input.

**3:** When A/D is off, it will not consume any current other than leakage current. The power-down current specification includes any such leakage from the A/D module.

**FIGURE 17-17: PIC16F685/687/689/690 A/D CONVERSION TIMING (NORMAL MODE)**




**Note 1:** 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 17-16: PIC16F685/687/689/690 A/D CONVERSION REQUIREMENTS**

| **Standard Operating Conditions (unless otherwise stated)**<br>Operating Temperature -40°C ≤ TA ≤ +125°C | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Sym** | **Characteristic** | **Min** | **Typ†** | **Max** | **Units** | **Conditions** |
| 130 | TAD | A/D Clock Period | 1.5 | — | — | µs | TOSC-based, VREF ≥ 2.5V |
| | | | 3.0* | — | — | µs | TOSC-based, VREF full range |
| 130 | TAD | A/D Internal RC Oscillator Period | | | | | ADCS<1:0> = 11 (RC mode) |
| | | | 3.0* | 6.0 | 9.0* | µs | At VDD = 2.5V |
| | | | 2.0* | 4.0 | 6.0* | µs | At VDD = 5.0V |
| 131 | TCNV | Conversion Time (not including Acquisition Time)(1) | — | 11 | — | TAD | Set GO bit to new data in A/D Result register |
| 132 | TACQ | Acquisition Time | | 11.5 | — | 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., 4.1 mV @ 4.096V) from the last sampled voltage (as stored 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. |

* These parameters are characterized but not tested.

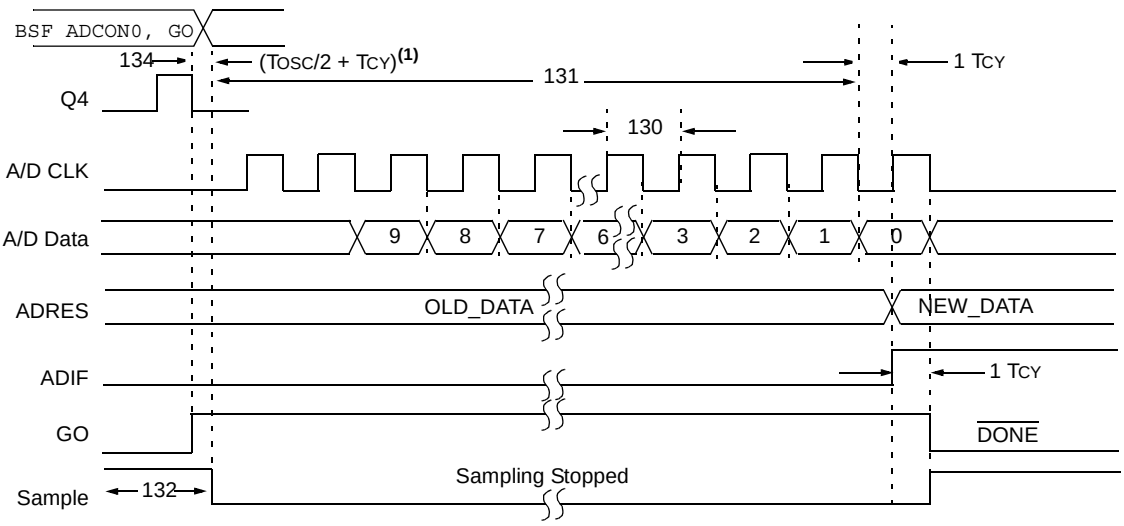
† Data in 'Typ' column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** ADRESH and ADRESL registers may be read on the following TCY cycle.

**2:** See Table 9-1 for minimum conditions.

**FIGURE 17-18: PIC16F685/687/689/690 A/D CONVERSION TIMING (SLEEP MODE)**




**Note 1:** 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 17-17: PIC16F685/687/689/690 A/D CONVERSION REQUIREMENTS (SLEEP MODE)**

| **Standard Operating Conditions (unless otherwise stated)**<br>Operating Temperature -40°C ≤ TA ≤ +125°C | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Sym** | **Characteristic** | **Min** | **Typ†** | **Max** | **Units** | **Conditions** |
| 130 | TAD | A/D Internal RC Oscillator Period | 3.0*<br>2.0* | 6.0<br>4.0 | 9.0*<br>6.0* | µs<br>µs | ADCS<1:0> = 11 (RC mode)<br>At VDD = 2.5V<br>At VDD = 5.0V |
| 131 | TCNV | Conversion Time (not including Acquisition Time)(1) | — | 11 | — | TAD | |
| 132 | TACQ | Acquisition Time | (2)<br>5* | 11.5<br>— | —<br>— | µs<br>µs | <br>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., 4.1 mV @ 4.096V) from the last sampled voltage (as stored on CHOLD). |
| 134 | TGO | Q4 to A/D Clock Start | — | TOSC/2 + TCY | — | — | 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. |

* These parameters are characterized but not tested.

† Data in 'Typ' column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** ADRES register may be read on the following TCY cycle.

**2:** See Table 9-1 for minimum conditions.

**NOTES:**

## 18.0 DC AND AC CHARACTERISTICS GRAPHS AND TABLES

Graphs are not available at this time.

**NOTES:**

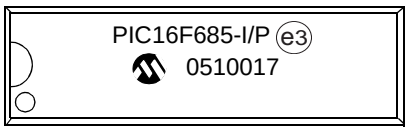
 

## 19.0 PACKAGING INFORMATION

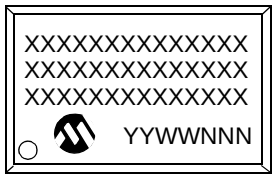
### 19.1 Package Marking Information

20-Lead PDIP

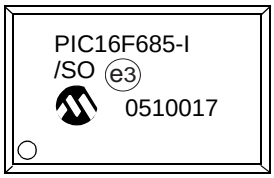

Example



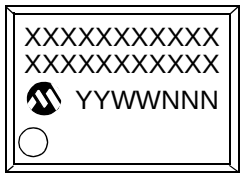

20-Lead SOIC (.300”)



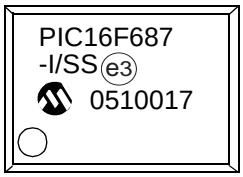

Example



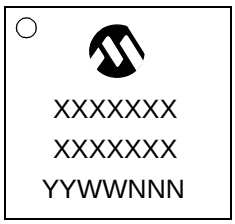

20-Lead SSOP



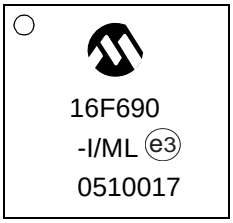

Example




20-Lead QFN




Example



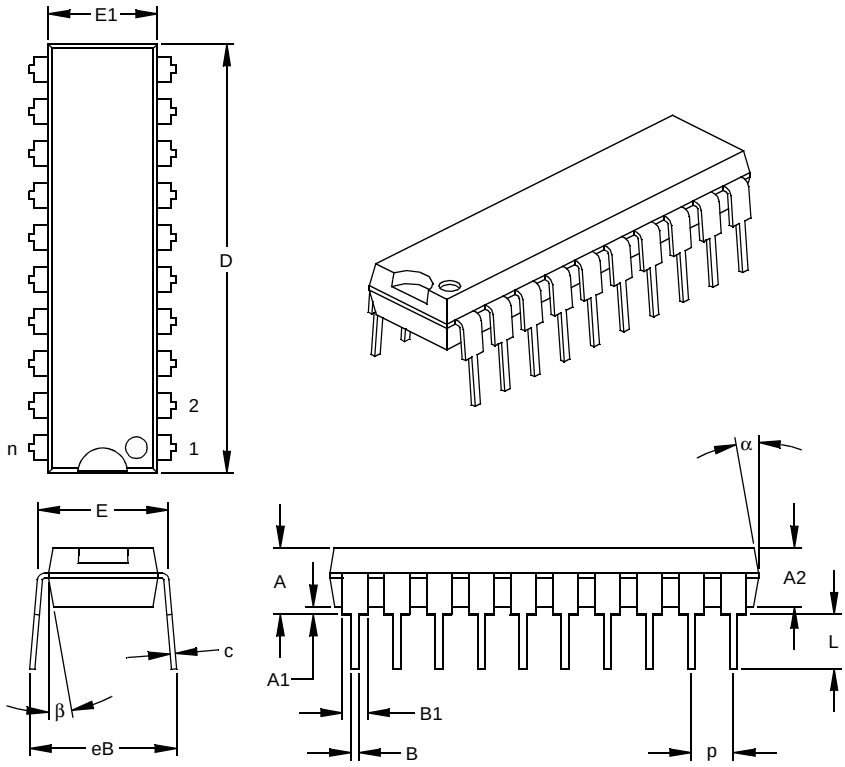

| | | |
|---|---|---|
| **Legend:** | XX...X | Customer-specific information |
| | Y | Year code (last digit of calendar year) |
| | YY | Year code (last 2 digits of calendar year) |
| | WW | Week code (week of January 1 is week ‘01’) |
| | NNN | Alphanumeric traceability code |
| | (e3) | Pb-free JEDEC 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.

## 19.2 Package Details

The following sections give the technical details of the packages.

### 20-Lead Plastic Dual In-line (P) – 300 mil Body (PDIP)



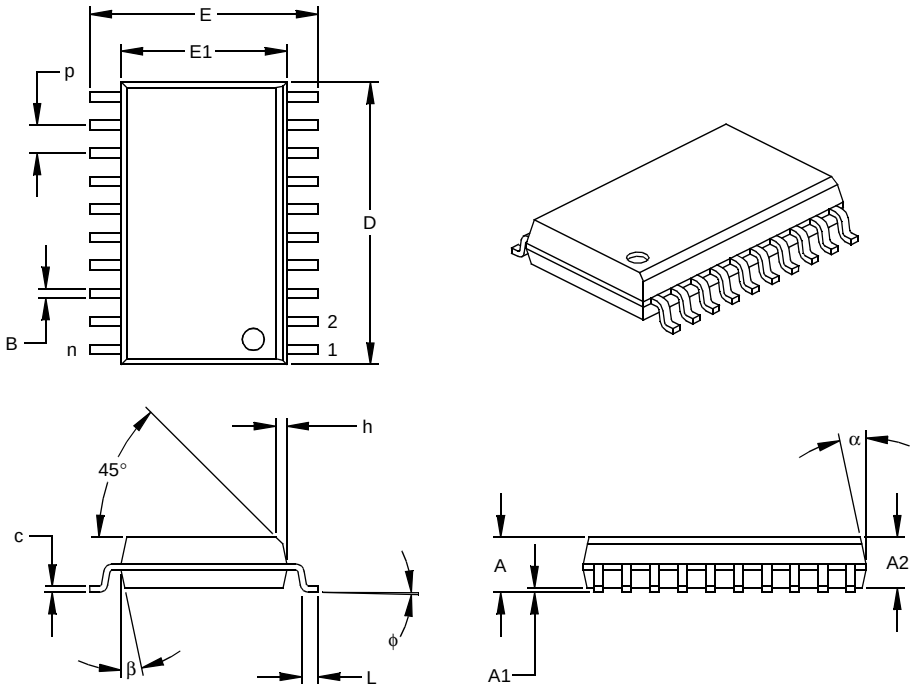

| Units | | INCHES* | | | MILLIMETERS | | |
|---|---|---|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX | MIN | NOM | MAX |
| Number of Pins | n | | 20 | | | 20 | |
| Pitch | p | | .100 | | | 2.54 | |
| Top to Seating Plane | A | .140 | .155 | .170 | 3.56 | 3.94 | 4.32 |
| Molded Package Thickness | A2 | .115 | .130 | .145 | 2.92 | 3.30 | 3.68 |
| Base to Seating Plane | A1 | .015 | | | 0.38 | | |
| Shoulder to Shoulder Width | E | .295 | .310 | .325 | 7.49 | 7.87 | 8.26 |
| Molded Package Width | E1 | .240 | .250 | .260 | 6.10 | 6.35 | 6.60 |
| Overall Length | D | 1.025 | 1.033 | 1.040 | 26.04 | 26.24 | 26.42 |
| Tip to Seating Plane | L | .120 | .130 | .140 | 3.05 | 3.30 | 3.56 |
| Lead Thickness | c | .008 | .012 | .015 | 0.20 | 0.29 | 0.38 |
| Upper Lead Width | B1 | .055 | .060 | .065 | 1.40 | 1.52 | 1.65 |
| Lower Lead Width | B | .014 | .018 | .022 | 0.36 | 0.46 | 0.56 |
| Overall Row Spacing § | eB | .310 | .370 | .430 | 7.87 | 9.40 | 10.92 |
| Mold Draft Angle Top | α | 5 | 10 | 15 | 5 | 10 | 15 |
| Mold Draft Angle Bottom | β | 5 | 10 | 15 | 5 | 10 | 15 |

* Controlling Parameter
§ Significant Characteristic

Notes:
Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010” (0.254mm) per side.
JEDEC Equivalent: MS-001
Drawing No. C04-019

## 20-Lead Plastic Small Outline (SO) – Wide, 300 mil Body (SOIC)

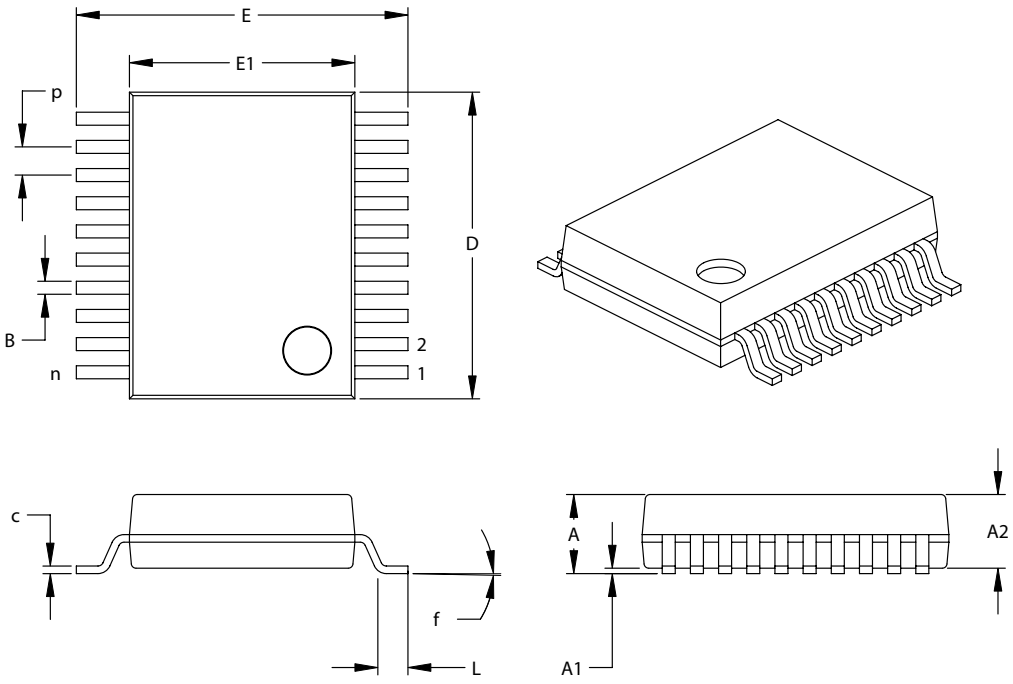



| Units | | INCHES* | | | MILLIMETERS | | |
|---|---|---|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX | MIN | NOM | MAX |
| Number of Pins | n | | 20 | | | 20 | |
| Pitch | p | | .050 | | | 1.27 | |
| Overall Height | A | .093 | .099 | .104 | 2.36 | 2.50 | 2.64 |
| Molded Package Thickness | A2 | .088 | .091 | .094 | 2.24 | 2.31 | 2.39 |
| Standoff § | A1 | .004 | .008 | .012 | 0.10 | 0.20 | 0.30 |
| Overall Width | E | .394 | .407 | .420 | 10.01 | 10.34 | 10.67 |
| Molded Package Width | E1 | .291 | .295 | .299 | 7.39 | 7.49 | 7.59 |
| Overall Length | D | .496 | .504 | .512 | 12.60 | 12.80 | 13.00 |
| Chamfer Distance | h | .010 | .020 | .029 | 0.25 | 0.50 | 0.74 |
| Foot Length | L | .016 | .033 | .050 | 0.41 | 0.84 | 1.27 |
| Foot Angle | ϕ | 0 | 4 | 8 | 0 | 4 | 8 |
| Lead Thickness | c | .009 | .011 | .013 | 0.23 | 0.28 | 0.33 |
| Lead Width | B | .014 | .017 | .020 | 0.36 | 0.42 | 0.51 |
| Mold Draft Angle Top | α | 0 | 12 | 15 | 0 | 12 | 15 |
| Mold Draft Angle Bottom | β | 0 | 12 | 15 | 0 | 12 | 15 |

* Controlling Parameter
§ Significant Characteristic

Notes:
Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010” (0.254mm) per side.
JEDEC Equivalent: MS-013
Drawing No. C04-094

## 20-Lead Plastic Shrink Small Outline (SS) – 209 mil Body, 5.30 mm (SSOP)




| Units | | INCHES | | | MILLIMETERS* | | |
|---|---|---|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX | MIN | NOM | MAX |
| Number of Pins | n | | 20 | | | 20 | |
| Pitch | p | | .026 | | | 0.65 | |
| Overall Height | A | - | - | .079 | - | - | 2.00 |
| Molded Package Thickness | A2 | .065 | .069 | .073 | 1.65 | 1.75 | 1.85 |
| Standoff | A1 | .002 | - | - | 0.05 | - | - |
| Overall Width | E | .291 | .307 | .323 | 7.40 | 7.80 | 8.20 |
| Molded Package Width | E1 | .197 | .209 | .220 | 5.00 | 5.30 | 5.60 |
| Overall Length | D | .272 | .283 | .289 | .295 | 7.20 | 7.50 |
| Foot Length | L | .022 | .030 | .037 | 0.55 | 0.75 | 0.95 |
| Lead Thickness | c | .004 | - | .010 | 0.09 | - | 0.25 |
| Foot Angle | f | 0° | 4° | 8° | 0° | 4° | 8° |
| Lead Width | B | .009 | - | .015 | 0.22 | - | 0.38 |

*Controlling Parameter

Notes:

Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" (0.254mm) per side.

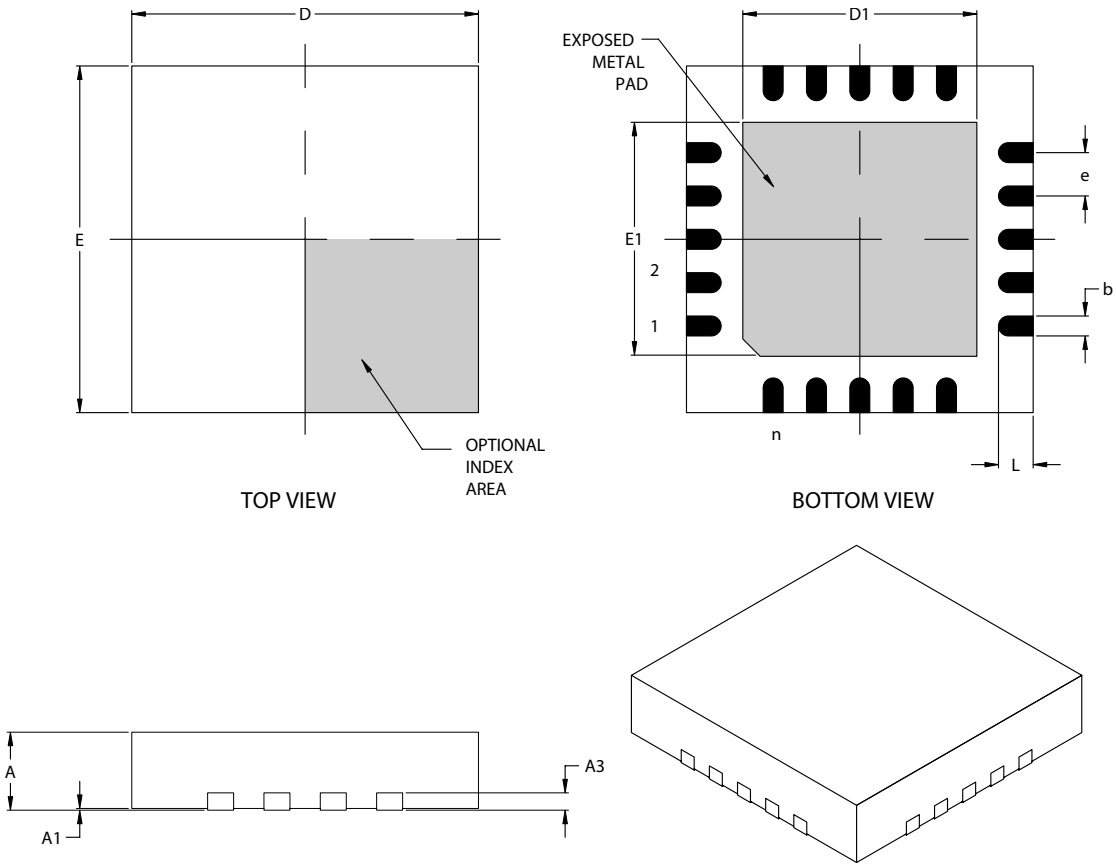
JEDEC Equivalent: MO-150

Drawing No. C04-072

Revised 11/03/03

## 20-Lead Plastic Quad Flat No Lead Package (ML) 4x4x0.9 mm Body (QFN) – Saw Singulated




| Units | | INCHES | | | MILLIMETERS* | | |
|---|---|---|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX | MIN | NOM | MAX |
| Number of Pins | n | | 20 | | | 20 | |
| Pitch | e | | .020 BSC | | | 0.50 BSC | |
| Overall Height | A | .031 | .035 | .039 | 0.80 | 0.90 | 1.00 |
| Standoff | A1 | .000 | .001 | .002 | 0.00 | 0.02 | 0.05 |
| Contact Thickness | A3 | | .008 REF | | | 0.20 REF | |
| Overall Width | E | .152 | .157 | .163 | 3.85 | 4.00 | 4.15 |
| Exposed Pad Width | E2 | .100 | .106 | .110 | 2.55 | 2.70 | 2.80 |
| Overall Length | D | .152 | .157 | .163 | 3.85 | 4.00 | 4.15 |
| Exposed Pad Length | D2 | .100 | .106 | .110 | 2.55 | 2.70 | 2.80 |
| Contact Width | b | .007 | .010 | .012 | 0.18 | 0.25 | 0.30 |
| Contact Length | L | .012 | .016 | .020 | 0.30 | 0.40 | 0.50 |

*Controlling Parameter

Notes:

JEDEC equivalent: Not Registered

Drawing No. C04-126

Revised 04-24-05

**NOTES:**

## APPENDIX A: DATA SHEET REVISION HISTORY

### Revision A

This is a new data sheet.

## APPENDIX B: MIGRATING FROM OTHER PICmicro® DEVICES

This discusses some of the issues in migrating from other PICmicro devices to the PIC16F6XX Family of devices.

### B.1 PIC16F676 to PIC16F685

**TABLE B-1: FEATURE COMPARISON**

| Feature | PIC16F676 | PIC16F685 |
|---|---|---|
| Max Operating Speed | 20 MHz | 20 MHz |
| Max Program Memory (Words) | 1024 | 4096 |
| SRAM (bytes) | 64 | 128 |
| A/D Resolution | 10-bit | 10-bit |
| Data EEPROM (Bytes) | 128 | 256 |
| Timers (8/16-bit) | 1/1 | 2/1 |
| Oscillator Modes | 8 | 8 |
| Brown-out Reset | Y | Y |
| Internal Pull-ups | RA0/1/2/4/5 | RA0/1/2/4/5, $\overline{\text{MCLR}}$ |
| Interrupt-on-change | RA0/1/2/3/4/5 | RA0/1/2/3/4/5 |
| Comparator | 1 | 2 |
| ECCP+ | N | Y |
| Ultra Low-Power Wake-Up | N | Y |
| Extended WDT | N | Y |
| Software Control Option of WDT/BOR | N | Y |
| INTOSC Frequencies | 4 MHz | 31 kHz-8 MHz |
| Clock Switching | N | Y |

**Note:** This device has been designed to perform to the parameters of its data sheet. It has been tested to an electrical specification designed to determine its conformance with these parameters. Due to process differences in the manufacture of this device, this device may have different performance characteristics than its earlier version. These differences may cause this device to perform differently in your application than the earlier version of this device.

**NOTES:**

## A



## B



## C

**R**



## U



## V



## W



## 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**

In addition, there is a Development Systems Information Line which lists the latest versions of Microchip's development systems software products. This line also provides information on how customers can receive currently available upgrade kits.

**The Development Systems Information Line numbers are:**

1-800-755-2345 – United States and most of Canada

1-480-792-7302 – Other International Locations

## READER RESPONSE

It is our intention to provide you with the best documentation possible to ensure successful use of your Microchip product. If you wish to provide your comments on organization, clarity, subject matter, and ways in which our documentation can better serve you, please FAX your comments to the Technical Publications Manager at (480) 792-4150.

Please list the following information, and use this outline to provide us with your comments about this document.

To: Technical Publications Manager Total Pages Sent _____

RE: Reader Response

From: Name _____

Company _____

Address _____

City / State / ZIP / Country _____

Telephone: (_____) _____ - _____ FAX: (_____) _____ - _____

Application (optional):

Would you like a reply? ___Y ___N

Device: PIC16F685/687/689/690 Literature Number: DS41262A

Questions:

1. What are the best features of this document?

2. How does this document meet your hardware and software development needs?

3. Do you find the organization of this document easy to follow? If not, why?

4. What additions to the document do you think would enhance the structure and subject?

5. What deletions from the document could be made without affecting the overall usefulness?

6. Is there any incorrect or misleading information (what and where)?

7. How would you improve this document?

## PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

| PART NO. | X | /XX | XXX |
|---|---|---|---|
| Device | Temperature Range | Package | Pattern |

| | | |
|---|---|---|
| **Device:** | PIC16F685[(1)], PIC16F687[(1)], PIC16F689[(1)], PIC16F690[(1)]; $V_{DD}$ range 4.2V to 5.5V | |
| **Temperature Range:** | I = -40°C to +85°C | (Industrial) |
| | E = -40°C to +125°C | (Extended) |
| **Package:** | ML = | QFN (Quad Flat, no lead) |
| | P = | PDIP |
| | SO = | SOIC |
| | SS = | SSOP |
| **Pattern:** | QTP, SQTP, Code or Special Requirements (blank otherwise) | |

**Examples:**

a) PIC16F685 - I/ML 301 = Industrial temp., QFN package, QTP pattern #301.

b) PIC16F689 - I/SO = Industrial temp., SOIC package.

c) PIC16F690T - T/E/SS = Extended temp., SSOP package.

**Note 1:** T = in tape and reel SSOP, SOIC and QFN packages only.

# WORLDWIDE SALES AND SERVICE

## AMERICAS

**Corporate Office**
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://support.microchip.com
Web Address:
www.microchip.com

**Atlanta**
Alpharetta, GA
Tel: 770-640-0034
Fax: 770-640-0307

**Boston**
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

**Chicago**
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

**Dallas**
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

**Detroit**
Farmington Hills, MI
Tel: 248-538-2250
Fax: 248-538-2260

**Kokomo**
Kokomo, IN
Tel: 765-864-8360
Fax: 765-864-8387

**Los Angeles**
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

**San Jose**
Mountain View, CA
Tel: 650-215-1444
Fax: 650-961-0286

**Toronto**
Mississauga, Ontario,
Canada
Tel: 905-673-0699
Fax: 905-673-6509

## ASIA/PACIFIC

**Australia - Sydney**
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

**China - Beijing**
Tel: 86-10-8528-2100
Fax: 86-10-8528-2104

**China - Chengdu**
Tel: 86-28-8676-6200
Fax: 86-28-8676-6599

**China - Fuzhou**
Tel: 86-591-8750-3506
Fax: 86-591-8750-3521

**China - Hong Kong SAR**
Tel: 852-2401-1200
Fax: 852-2401-3431

**China - Shanghai**
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

**China - Shenyang**
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

**China - Shenzhen**
Tel: 86-755-8203-2660
Fax: 86-755-8203-1760

**China - Shunde**
Tel: 86-757-2839-5507
Fax: 86-757-2839-5571

**China - Qingdao**
Tel: 86-532-502-7355
Fax: 86-532-502-7205

## ASIA/PACIFIC

**India - Bangalore**
Tel: 91-80-2229-0061
Fax: 91-80-2229-0062

**India - New Delhi**
Tel: 91-11-5160-8631
Fax: 91-11-5160-8632

**Japan - Kanagawa**
Tel: 81-45-471- 6166
Fax: 81-45-471-6122

**Korea - Seoul**
Tel: 82-2-554-7200
Fax: 82-2-558-5932 or
82-2-558-5934

**Singapore**
Tel: 65-6334-8870
Fax: 65-6334-8850

**Taiwan - Kaohsiung**
Tel: 886-7-536-4818
Fax: 886-7-536-4803

**Taiwan - Taipei**
Tel: 886-2-2500-6610
Fax: 886-2-2508-0102

**Taiwan - Hsinchu**
Tel: 886-3-572-9526
Fax: 886-3-572-6459

## EUROPE

**Austria - Weis**
Tel: 43-7242-2244-399
Fax: 43-7242-2244-393

**Denmark - Ballerup**
Tel: 45-4450-2828
Fax: 45-4485-2829

**France - Massy**
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

**Germany - Ismaning**
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

**Italy - Milan**
Tel: 39-0331-742611
Fax: 39-0331-466781

**Netherlands - Drunen**
Tel: 31-416-690399
Fax: 31-416-690340

**England - Berkshire**
Tel: 44-118-921-5869
Fax: 44-118-921-5820

03/01/05