# PIC16(L)F1703/7

## 14/20-Pin 8-Bit Advanced Analog Flash Microcontrollers

### Core Features

- C Compiler Optimized RISC Architecture
- Only 49 Instructions
- Operating Speed:
  - 0-32 MHz
  - 125 ns minimum instruction cycle
- Interrupt Capability
- 16-Level Deep Hardware Stack
- Up to Two 8-Bit Timers
- One 16-bit Timer
- Power-on Reset (POR)
- Power-up Timer (PWRT)
- Low-Power Brown-out Reset (LPBOR)
- Programmable Watchdog Timer (WDT) up to 256s
- Programmable Code Protection

### Memory

- Two Kwords Flash Program Memory
- 256 Bytes Data SRAM Memory
- Direct, Indirect and Relative Addressing modes
- High-Endurance Flash Data Memory (HEF)
  - 128 bytes of nonvolatile data storage
  - 100k erase/write cycles

### Operating Characteristics

- Operating Voltage Range:
  - 1.8V to 3.6V (PIC16LF1703/7)
  - 2.3V to 5.5V (PIC16F1703/7)
- Temperature Range:
  - Industrial: -40°C to 85°C
  - Extended: -40°C to 125°C

### Digital Peripherals

- Capture/Compare/PWM (CCP) module
- Serial Communications:
  - SPI, $I^2C$
- Up to 18 I/O Pins and One Input Pin:
  - Individually programmable weak pull-ups
  - Slew rate control
  - Interrupt-on-change with edge-select
- Peripheral Pin Select (PPS):
  - Enables pin mapping of digital I/O

### eXtreme Low-Power (XLP) Features

- Sleep mode: 50 nA @ 1.8V, typical
- Watchdog Timer: 500 nA @ 1.8V, typical
- Operating Current:
  - 8 uA @ 32 kHz, 1.8V, typical
  - 32 uA/MHz @ 1.8V, typical

### Intelligent Analog Peripherals

- Operational Amplifiers:
  - Two configurable rail-to-rail Op Amps
  - Selectable internal and external channels
  - 2 MHz gain bandwidth product
- 10-Bit Analog-to-Digital Converter (ADC):
  - Up to 12 external channels
  - Conversion available during Sleep
  - Temperature Indicator
- Zero-Cross Detector (ZCD):
  - Detect when AC signal on pin crosses ground

### Clocking Structure

- 16 MHz Internal Oscillator Block:
  - ±1% at calibration
  - Selectable frequency range from 0 to 32 MHz
- 31 kHz Low-Power Internal Oscillator

### Programming/Debug Features

- In-Circuit Debug Integrated On-Chip
- Emulation Header for Advanced Debug:
  - Provides trace, background debug and up to 32 hardware break points
- In-Circuit Serial Programming™ (ICSP™) via Two Pins

© 2013-2015 Microchip Technology Inc.

## PIC16(L)F170x Family Types

| Device | Data Sheet Index | Program Memory Flash (words) | Data SRAM (bytes) | High-Endurance Flash (bytes) | I/O's(2) | 10-bit ADC (ch) | 8-bit DAC | High-Speed/ Comparators | Op Amp | Zero Cross | Timers (8/16-bit) | CCP | PWM | COG | EUSART | MSSP ($I^2C$/SPI) | CLC | PPS | Debug(1) | XLP |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PIC16(L)F1703 | (3) | 2048 | 256 | 128 | 12 | 8 | 0 | 0 | 2 | 1 | 2/1 | 2 | 0 | 0 | 0 | 1 | 0 | Y | I/E | Y |
| PIC16(L)F1704 | (1) | 4096 | 512 | 128 | 12 | 8 | 1 | 2 | 2 | 1 | 4/1 | 2 | 2 | 1 | 1 | 1 | 3 | Y | I/E | Y |
| PIC16(L)F1705 | (2) | 8192 | 1024 | 128 | 12 | 8 | 1 | 2 | 2 | 1 | 4/1 | 2 | 2 | 1 | 1 | 1 | 3 | Y | I/E | Y |
| PIC16(L)F1707 | (3) | 2048 | 256 | 128 | 18 | 12 | 0 | 0 | 2 | 1 | 2/1 | 2 | 0 | 0 | 0 | 1 | 0 | Y | I/E | Y |
| PIC16(L)F1708 | (1) | 4096 | 512 | 128 | 18 | 12 | 1 | 2 | 2 | 1 | 4/1 | 2 | 2 | 1 | 1 | 1 | 3 | Y | I/E | Y |
| PIC16(L)F1709 | (2) | 8192 | 1024 | 128 | 18 | 12 | 1 | 2 | 2 | 1 | 4/1 | 2 | 2 | 1 | 1 | 1 | 3 | Y | I/E | Y |

**Note 1:** Debugging Methods: (I) – Integrated on Chip; (H) – using Debug Header; E – using Emulation Header.
**2:** One pin is input-only.

**Data Sheet Index:** (Unshaded devices are described in this document.)

**1:** DS40001715 PIC16(L)F1704/8 Data Sheet, 14/20-Pin Flash, 8-bit Microcontrollers.
**2:** DS40001729 PIC16(L)F1705/9 Data Sheet, 14/20-Pin Flash, 8-bit Microcontrollers.
**3:** DS40001722 PIC16(L)F1703/7 Data Sheet, 14/20-Pin Flash, 8-bit Microcontrollers

**Note:** For other small form-factor package availability and marking information, please visit **http://www.microchip.com/packaging** or contact your local sales office.


© 2013-2015 Microchip Technology Inc.

## Pin Diagrams

**FIGURE 1: 14-PIN DIAGRAM FOR PIC16(L)F1703**

**PDIP, SOIC, TSSOP**

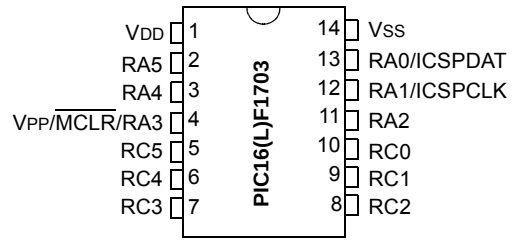



**FIGURE 2: 16-PIN PACKAGE DIAGRAM FOR PIC16(L)F1703**

**QFN**

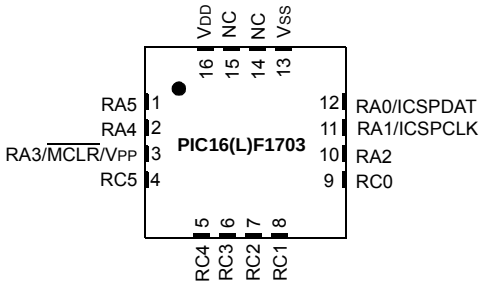


**FIGURE 3: 20-PIN PACKAGE DIAGRAM FOR PIC16(L)F1707**

**PDIP, SOIC, SSOP**

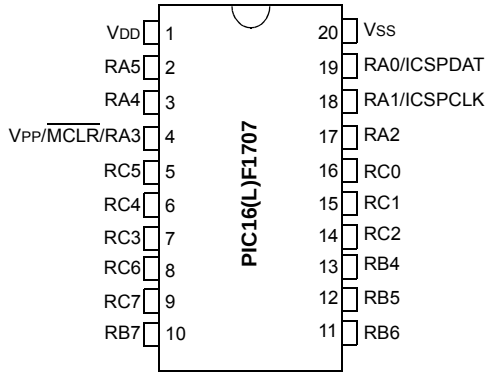
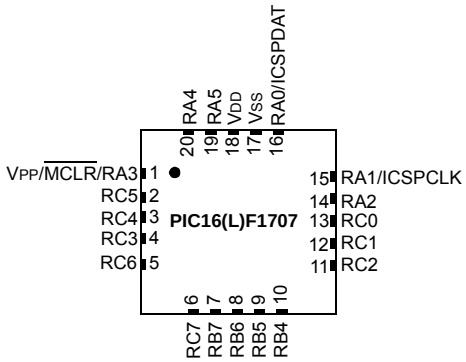



**FIGURE 4: 20-PIN PACKAGE DIAGRAM FOR PIC16(L)F1707**

**QFN**





© 2013-2015 Microchip Technology Inc.

**TABLE 1: 14/16-PIN ALLOCATION TABLE (PIC16(L)F1703)**

| I/O(2) | PDIP/SOIC/TSSOP | QFN | ADC | Reference | Op Amp | Zero Cross | Timers | CCP | MSSP | Interrupt | Pull-up | Basic |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RA0 | 13 | 12 | AN0 | VREF- | — | — | — | — | — | IOC | Y | ICSPDAT |
| RA1 | 12 | 11 | AN1 | VREF+ | — | — | — | — | — | IOC | Y | ICSPCLK |
| RA2 | 11 | 10 | AN2 | — | — | ZCD | T0CKI(1) | — | — | INT(1) IOC | Y | — |
| RA3 | 4 | 3 | — | — | — | — | — | — | — | IOC | Y | $\overline{\text{MCLR}}$ VPP |
| RA4 | 3 | 2 | AN3 | — | — | — | T1G(1) | — | — | IOC | Y | CLKOUT |
| RA5 | 2 | 1 | — | — | — | — | T1CKI(1) | — | — | IOC | Y | CLKIN |
| RC0 | 10 | 9 | AN4 | — | OPA1IN+ | — | — | — | SCK(1) SCL(3) | IOC | Y | — |
| RC1 | 9 | 8 | AN5 | — | OPA1IN- | — | — | — | SDI(1) SDA(3) | IOC | Y | — |
| RC2 | 8 | 7 | AN6 | — | OPA1OUT | — | — | — | — | IOC | Y | — |
| RC3 | 7 | 6 | AN7 | — | OPA2OUT | — | — | CCP2(1) | $\overline{\text{SS}}$(1) | IOC | Y | — |
| RC4 | 6 | 5 | — | — | OPA2IN- | — | — | — | — | IOC | Y | — |
| RC5 | 5 | 4 | — | — | OPA2IN+ | — | — | CCP1(1) | — | IOC | Y | — |
| VDD | 1 | 16 | — | — | — | — | — | — | — | — | — | VDD |
| VSS | 14 | 13 | — | — | — | — | — | — | — | — | — | VSS |
| OUT(2) | — | — | — | — | — | — | — | CCP1 | SDA(3) | — | — | — |
| | — | — | — | — | — | — | — | CCP2 | SCL(3) SCK | — | — | — |
| | — | — | — | — | — | — | — | — | SDO | — | — | — |

**Note 1:** Default peripheral input. Input can be moved to any other pin with the PPS input selection registers.

**2:** All pin digital outputs default to PORT latch data. Any pin can be selected as a peripheral digital output with the PPS output selection registers.

**3:** These peripheral functions are bidirectional. The output pin selections must be the same as the input pin selections.

**TABLE 2: 20-PIN ALLOCATION TABLE (PIC16(L)F1707)**

| I/O(2) | PDIP/SOIC/SSOP | QFN | ADC | Reference | Op Amp | Zero Cross | Timers | CCP | MSSP | Interrupt | Pull-up | Basic |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RA0 | 19 | 16 | AN0 | VREF- | — | — | — | — | — | IOC | Y | ICSPDAT |
| RA1 | 18 | 15 | AN1 | VREF+ | — | — | — | — | — | IOC | Y | ICSPCLK |
| RA2 | 17 | 14 | AN2 | — | — | ZCD | T0CKI(1) | — | — | INT(1) IOC | Y | — |
| RA3 | 4 | 1 | — | — | — | — | — | — | — | IOC | Y | $\overline{\text{MCLR}}$ VPP |
| RA4 | 3 | 20 | AN3 | — | — | — | T1G(1) | — | — | IOC | Y | CLKOUT |
| RA5 | 2 | 19 | — | — | — | — | T1CKI | — | — | IOC | Y | CLKIN |
| RB4 | 13 | 10 | AN10 | — | OPA1IN- | — | — | — | SCK(1) SDA(3) | IOC | Y | — |
| RB5 | 12 | 9 | AN11 | — | OPA1IN+ | — | — | — | — | IOC | Y | — |
| RB6 | 11 | 8 | — | — | — | — | — | — | SDI(1) SCL(3) | IOC | Y | — |
| RB7 | 10 | 7 | — | — | — | — | — | — | — | IOC | Y | — |
| RC0 | 16 | 13 | AN4 | — | — | — | — | — | — | IOC | Y | — |
| RC1 | 15 | 12 | AN5 | — | — | — | — | — | — | IOC | Y | — |
| RC2 | 14 | 11 | AN6 | — | OPA1OUT | — | — | — | — | IOC | Y | — |
| RC3 | 7 | 4 | AN7 | — | OPA2OUT | — | — | CCP2(1) | — | IOC | Y | — |
| RC4 | 6 | 3 | — | — | — | — | — | — | — | IOC | Y | — |
| RC5 | 5 | 2 | — | — | — | — | — | CCP1(1) | — | IOC | Y | — |
| RC6 | 8 | 5 | AN8 | — | OPA2IN- | — | — | — | $\overline{\text{SS}}$(1) | IOC | Y | — |
| RC7 | 9 | 6 | AN9 | — | OPA2IN+ | — | — | — | — | IOC | Y | — |
| VDD | 1 | 18 | — | — | — | — | — | — | — | — | — | VDD |
| VSS | 20 | 17 | — | — | — | — | — | — | — | — | — | VSS |
| OUT(2) | — | — | — | — | — | — | — | CPP1 | SDA(3) | — | — | — |
| | — | — | — | — | — | — | — | CPP2 | SCL(3) SCK | — | — | — |
| | — | — | — | — | — | — | — | — | SDO | — | — | — |

**Note 1:** Default peripheral input. Input can be moved to any other pin with the PPS input selection registers.

**2:** All pin digital outputs default to PORT latch data. Any pin can be selected as a peripheral digital output with the PPS output selection registers.

**3:** These peripheral functions are bidirectional. The output pin selections must be the same as the input pin selections.


© 2013-2015 Microchip Technology Inc.

## Table of Contents

1.0 Device Overview ..... 9
2.0 Enhanced Mid-Range CPU ..... 15
3.0 Memory Organization ..... 17
4.0 Device Configuration ..... 48
5.0 Resets ..... 53
6.0 Oscillator Module (with Fail-Safe Clock Monitor) ..... 61
7.0 Interrupts ..... 72
8.0 Power-Down Mode (Sleep) ..... 85
9.0 Watchdog Timer (WDT) ..... 89
10.0 Flash Program Memory Control ..... 94
11.0 I/O Ports ..... 110
12.0 Peripheral Pin Select (PPS) Module ..... 128
13.0 Interrupt-On-Change ..... 135
14.0 Fixed Voltage Reference (FVR) ..... 141
15.0 Temperature Indicator Module ..... 143
16.0 Analog-to-Digital Converter (ADC) Module ..... 145
17.0 Operational Amplifier (OPA) Modules ..... 159
18.0 Zero-Cross Detection (ZCD) Module ..... 162
19.0 Timer0 Module ..... 166
20.0 Timer1 Module with Gate Control ..... 169
21.0 Timer2 Module ..... 180
22.0 Capture/Compare/PWM Modules ..... 184
23.0 Master Synchronous Serial Port (MSSP) Module ..... 194
24.0 In-Circuit Serial Programming™ (ICSP™) ..... 249
25.0 Instruction Set Summary ..... 251
26.0 Electrical Specifications ..... 265
27.0 DC and AC Characteristics Graphs and Charts ..... 293
28.0 Development Support ..... 310
29.0 Packaging Information ..... 314
The Microchip Web Site ..... 335
Customer Change Notification Service ..... 335
Customer Support ..... 335
Product Identification System ..... 336

## 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**. 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., DS30000000A is version A of document DS30000000).

### 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.


© 2013-2015 Microchip Technology Inc.

## 1.0 DEVICE OVERVIEW

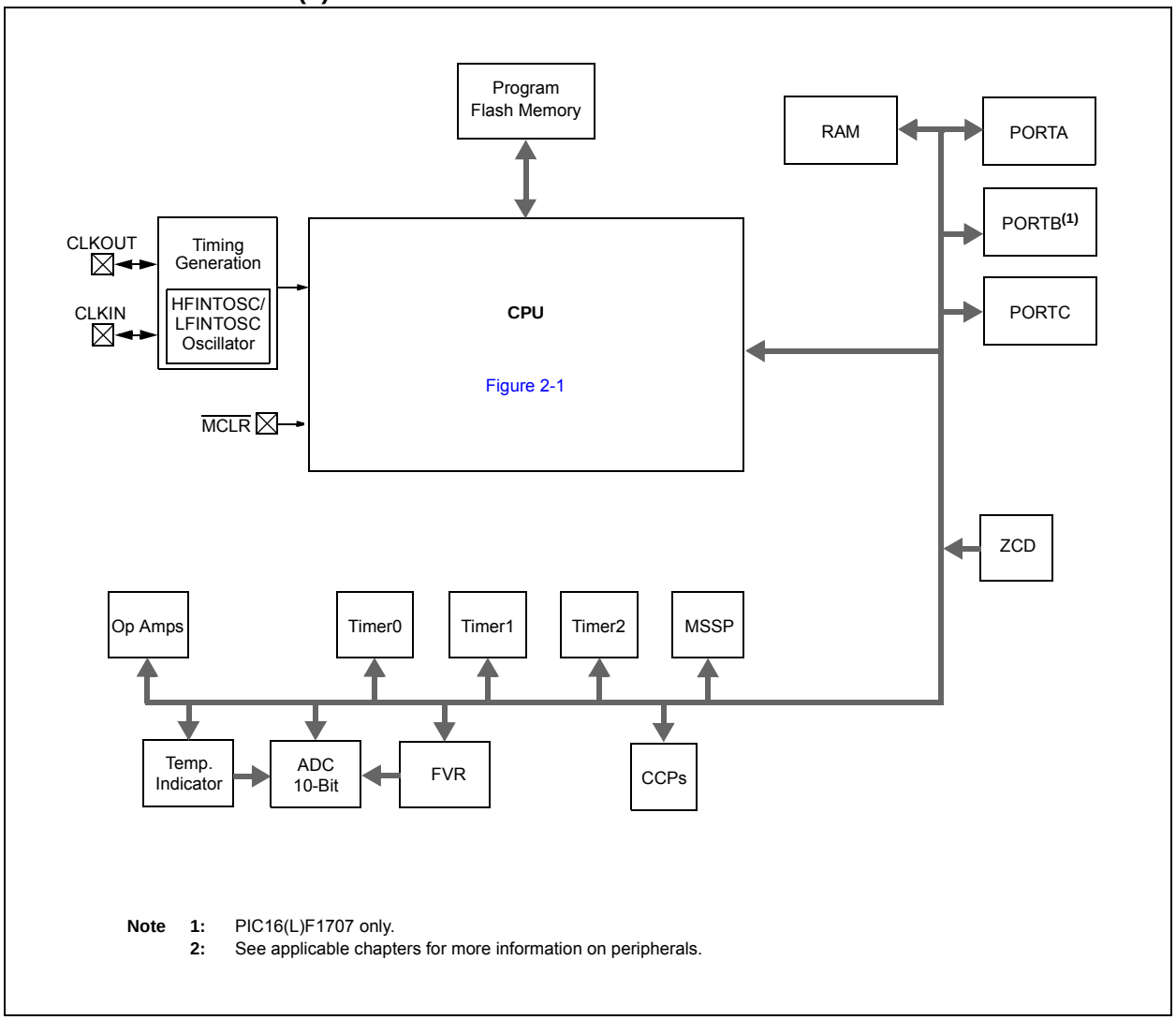
The PIC16(L)F1703/7 are described within this data sheet. They are available in 14-pin and 20-pin DIP packages and 16-pin and 20-pin QFN packages. Figure 1-1 shows a block diagram of the PIC16(L)F1703/7 devices. Table 1-2 shows the pinout descriptions.

Reference Table 1-1 for peripherals available per device.

**TABLE 1-1: DEVICE PERIPHERAL SUMMARY**

| Peripheral | | PIC16(L)F1703 | PIC16(L)F1707 |
|---|---|---|---|
| Analog-to-Digital Converter (ADC) | | ● | ● |
| Fixed Voltage Reference (FVR) | | ● | ● |
| Zero-Cross Detection (ZCD) | | ● | ● |
| Temperature Indicator | | ● | ● |
| Capture/Compare/PWM (CCP/ECCP) Modules | | | |
| | CCP1 | ● | ● |
| | CCP2 | ● | ● |
| Master Synchronous Serial Ports | | | |
| | MSSP | ● | ● |
| Op Amp | | | |
| | Op Amp 1 | ● | ● |
| | Op Amp 2 | ● | ● |
| Timers | | | |
| | Timer0 | ● | ● |
| | Timer1 | ● | ● |
| | Timer2 | ● | ● |

**FIGURE 1-1: PIC16(L)F1703/7 BLOCK DIAGRAM**




**Note 1:** PIC16(L)F1707 only.
**2:** See applicable chapters for more information on peripherals.


© 2013-2015 Microchip Technology Inc.

**TABLE 1-2: PIC16(L)F1703 PINOUT DESCRIPTION**

| Name | Function | Input Type | Output Type | Description |
|---|---|---|---|---|
| RA0/AN0/VREF-/ICSPDAT | RA0 | TTL/ST | CMOS | General purpose I/O. |
| | AN0 | AN | — | ADC Channel 0 input. |
| | VREF- | AN | — | ADC Negative Voltage Reference input. |
| | ICSPDAT | ST | CMOS | ICSP™ Data I/O. |
| RA1/AN1/VREF+/ICSPCLK | RA1 | TTL/ST | CMOS | General purpose I/O. |
| | AN1 | AN | — | ADC Channel 1 input. |
| | VREF+ | AN | — | ADC Voltage Reference input. |
| | ICSPCLK | ST | — | Serial Programming Clock. |
| RA2/AN2/ZCD/T0CKI(1)/INT(1) | RA2 | TTL/ST | CMOS | General purpose I/O. |
| | AN2 | AN | — | ADC Channel 2 input. |
| | ZCD | — | AN | Zero Cross Detection Current Source/Sink. |
| | T0CKI | TTL/ST | — | Timer0 clock input. |
| | INT | TTL/ST | — | External interrupt. |
| RA3/$\overline{MCLR}$/VPP | RA3 | TTL/ST | CMOS | General purpose I/O. |
| | $\overline{MCLR}$ | ST | — | Master Clear with internal pull-up. |
| | VPP | HV | — | Programming voltage. |
| RA4/AN3/T1G(1)/CLKOUT | RA4 | TTL/ST | CMOS | General purpose I/O. |
| | AN3 | AN | — | ADC Channel 3 input. |
| | T1G | TTL/ST | — | Timer1 gate input. |
| | CLKOUT | — | CMOS | FOSC/4 output. |
| RA5/T1CKI(1)/CLKIN | RA5 | TTL/ST | CMOS | General purpose I/O. |
| | T1CKI | TTL/ST | — | Timer1 clock input. |
| | CLKIN | TTL/ST | — | External clock input (EC mode). |
| RC0/AN4/OPA1IN+/SCK(1)/SCL(3) | RC0 | TTL/ST | CMOS | General purpose I/O. |
| | AN4 | AN | — | ADC Channel 4 input. |
| | OPA1IN+ | AN | — | Operational Amplifier 1 non-inverting input. |
| | SCK | ST | — | SPI clock. |
| | SCL | $I^2C$ | — | $I^2C$ clock. |
| RC1/AN5/OPA1IN-/SDI(1)/SDA(3) | RC1 | TTL/ST | CMOS | General purpose I/O. |
| | AN5 | AN | — | ADC Channel 5 input. |
| | OPA1IN- | AN | — | Operational Amplifier 1 inverting input. |
| | SDI | CMOS | — | SPI data input. |
| | SDA | $I^2C$ | — | $I^2C$ data input. |
| RC2/AN6/OPA1OUT | RC2 | TTL/ST | CMOS | General purpose I/O. |
| | AN6 | AN | — | ADC Channel 6 input. |
| | OPA1OUT | — | AN | Operational Amplifier 1 output. |

**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 $I^2C$ = Schmitt Trigger input with $I^2C$
HV = High Voltage

**Note 1:** Default peripheral input. Input can be moved to any other pin with the PPS input selection registers. See Register 12-1.

**2:** All pin outputs default to PORT latch data. Any pin can be selected as a digital peripheral output with the PPS output selection registers. See Register 12-3.

**3:** These $I^2C$ functions are bidirectional. The output pin selections must be the same as the input pin selections.

**TABLE 1-2: PIC16(L)F1703 PINOUT DESCRIPTION (CONTINUED)**

| Name | Function | Input Type | Output Type | Description |
|---|---|---|---|---|
| RC3/AN7/OPA2OUT/CCP2(1)/$\overline{SS}$(1) | RC3 | TTL/ST | CMOS | General purpose I/O. |
| | AN7 | AN | — | ADC Channel 7 input. |
| | OPA2OUT | — | AN | Operational Amplifier 2 output. |
| | CCP2 | TTL/ST | — | Capture/Compare/PWM2. |
| | $\overline{SS}$ | TTL/ST | — | Slave Select input. |
| RC4/OPA2IN- | RC4 | TTL/ST | CMOS | General purpose I/O. |
| | OPA2IN- | AN | — | Operational Amplifier 2 inverting input. |
| RC5/OPA2IN+/CCP1(1) | RC5 | TTL/ST | CMOS | General purpose I/O. |
| | OPA2IN+ | AN | — | Operational Amplifier 2 non-inverting input. |
| | CCP1 | TTL/ST | — | Capture/Compare/PWM1. |
| VDD | VDD | Power | — | Positive supply. |
| VSS | VSS | Power | — | Ground reference. |
| OUT(2) | CCP1 | — | CMOS | Capture/Compare/PWM1 output. |
| | CCP2 | — | CMOS | Capture/Compare/PWM2 output. |
| | SDA(3) | — | OD | $I^2C$ data input/output. |
| | SDO | — | CMOS | SPI data output. |
| | SCK | — | CMOS | SPI clock output. |
| | SCL(3) | — | OD | $I^2C$ clock output. |

**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 $I^2C$ = Schmitt Trigger input with $I^2C$
HV = High Voltage

**Note 1:** Default peripheral input. Input can be moved to any other pin with the PPS input selection registers. See Register 12-1.

**2:** All pin outputs default to PORT latch data. Any pin can be selected as a digital peripheral output with the PPS output selection registers. See Register 12-3.

**3:** These $I^2C$ functions are bidirectional. The output pin selections must be the same as the input pin selections.


© 2013-2015 Microchip Technology Inc.

**TABLE 1-3: PIC16(L)F1707 PINOUT DESCRIPTION**

| Name | Function | Input Type | Output Type | Description |
|---|---|---|---|---|
| RA0/AN0/VREF-/ICSPDAT | RA0 | TTL/ST | CMOS | General purpose I/O. |
| | AN0 | AN | — | ADC Channel 0 input. |
| | VREF- | AN | — | ADC Negative Voltage Reference input. |
| | ICSPDAT | ST | CMOS | ICSP™ Data I/O. |
| RA1/AN1/VREF+/ICSPCLK | RA1 | TTL/ST | CMOS | General purpose I/O. |
| | AN1 | AN | — | ADC Channel 1 input. |
| | VREF+ | AN | — | ADC Voltage Reference input. |
| | ICSPCLK | ST | — | Serial Programming Clock. |
| RA2/AN2/ZCD/ T0CKI(1)/INT(1) | RA2 | TTL/ST | CMOS | General purpose I/O. |
| | AN2 | AN | — | ADC Channel 2 input. |
| | ZCD | — | AN | Zero-Cross Detection Current Source/Sink. |
| | T0CKI | ST | — | Timer0 clock input. |
| | INT | ST | — | External interrupt. |
| RA3/$\overline{MCLR}$/VPP | RA3 | TTL/ST | CMOS | General purpose I/O. |
| | $\overline{MCLR}$ | ST | — | Master Clear with internal pull-up. |
| | VPP | HV | — | Programming voltage. |
| RA4/AN3/T1G(1)/CLKOUT | RA4 | TTL/ST | CMOS | General purpose I/O. |
| | AN3 | AN | — | ADC Channel 3 input. |
| | T1G | ST | — | Timer1 gate input. |
| | CLKOUT | — | CMOS | FOSC/4 output. |
| RA5/T1CKI/CLKIN | RA5 | TTL/ST | CMOS | General purpose I/O. |
| | T1CKI | ST | — | Timer1 clock input. |
| | CLKIN | ST | — | External clock input (EC mode). |
| RB4/AN10/OPA1IN-/SCK(1)/ SDA(3) | RB4 | TTL/ST | CMOS | General purpose I/O. |
| | AN10 | AN | — | ADC Channel 10 input. |
| | OPA1IN- | AN | — | Operational Amplifier 1 inverting input. |
| | SCK | ST | CMOS | SPI clock. |
| | SDA | $I^2C$ | OD | $I^2C$ data input/output. |
| RB5/AN11/OPA1IN+ | RB5 | TTL/ST | CMOS | General purpose I/O. |
| | AN11 | AN | — | ADC Channel 11 input. |
| | OPA1IN+ | AN | — | Operational Amplifier 1 non-inverting input. |
| RB6/SDI(1)/SCL(3) | RB6 | TTL/ST | CMOS | General purpose I/O. |
| | SDI | CMOS | — | SPI data input. |
| | SCL | $I^2C$ | OD | $I^2C$ clock. |
| RB7 | RB7 | TTL/ST | CMOS | General purpose I/O. |

**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 $I^2C$ = Schmitt Trigger input with $I^2C$
HV = High Voltage

**Note 1:** Default peripheral input. Input can be moved to any other pin with the PPS input selection registers. See Register 12-2.

**2:** All pin outputs default to PORT latch data. Any pin can be selected as a digital peripheral output with the PPS output selection registers. See Register 12-3.

**3:** These $I^2C$ functions are bidirectional. The output pin selections must be the same as the input pin selections.

**TABLE 1-3: PIC16(L)F1707 PINOUT DESCRIPTION (CONTINUED)**

| Name | Function | Input Type | Output Type | Description |
|---|---|---|---|---|
| RC0/AN4 | RC0 | TTL/ST | CMOS | General purpose I/O. |
| | AN4 | AN | — | ADC Channel 4 input. |
| RC1/AN5 | RC1 | TTL/ST | CMOS | General purpose I/O. |
| | AN5 | AN | — | ADC Channel 5 input. |
| RC2/AN6/OPA1OUT | RC2 | TTL/ST | CMOS | General purpose I/O. |
| | AN6 | AN | — | ADC Channel 6 input. |
| | OPA1OUT | — | AN | Operational Amplifier 1 output. |
| RC3/AN7/OPA2OUT/CCP2(1) | RC3 | TTL/ST | CMOS | General purpose I/O. |
| | AN7 | AN | — | ADC Channel 7 input. |
| | OPA2OUT | — | AN | Operational Amplifier 2 output. |
| | CCP2 | ST | CMOS | Capture/Compare/PWM2. |
| RC4 | RC4 | TTL/ST | CMOS | General purpose I/O. |
| RC5/CCP1(1) | RC5 | TTL/ST | CMOS | General purpose I/O. |
| | CCP1 | ST | CMOS | Capture/Compare/PWM1. |
| RC6/AN8/OPA2IN-/$\overline{SS}$(1) | RC6 | TTL/ST | CMOS | General purpose I/O. |
| | AN8 | AN | — | ADC Channel 8 input. |
| | OPA2IN- | AN | — | Operational Amplifier 2 inverting input. |
| | $\overline{SS}$ | ST | — | Slave Select input. |
| RC7/AN9/OPA2IN+ | RC7 | TTL/ST | CMOS | General purpose I/O. |
| | AN9 | AN | — | ADC Channel 9 input. |
| | OPA2IN+ | AN | — | Operational Amplifier 2 non-inverting input. |
| VDD | VDD | Power | — | Positive supply. |
| VSS | VSS | Power | — | Ground reference. |
| OUT(2) | CCP1 | — | CMOS | Capture/Compare/PWM1 output. |
| | CCP2 | — | CMOS | Capture/Compare/PWM2 output. |
| | SDA(3) | — | OD | $I^2C$ data input/output. |
| | SDO | — | CMOS | SPI data output. |
| | SCK | — | CMOS | SPI clock output. |
| | SCL(3) | $I^2C$ | OD | $I^2C$ clock output. |

**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 $I^2C$ = Schmitt Trigger input with $I^2C$
HV = High Voltage

**Note 1:** Default peripheral input. Input can be moved to any other pin with the PPS input selection registers. See Register 12-2.

**2:** All pin outputs default to PORT latch data. Any pin can be selected as a digital peripheral output with the PPS output selection registers. See Register 12-3.

**3:** These $I^2C$ functions are bidirectional. The output pin selections must be the same as the input pin selections.

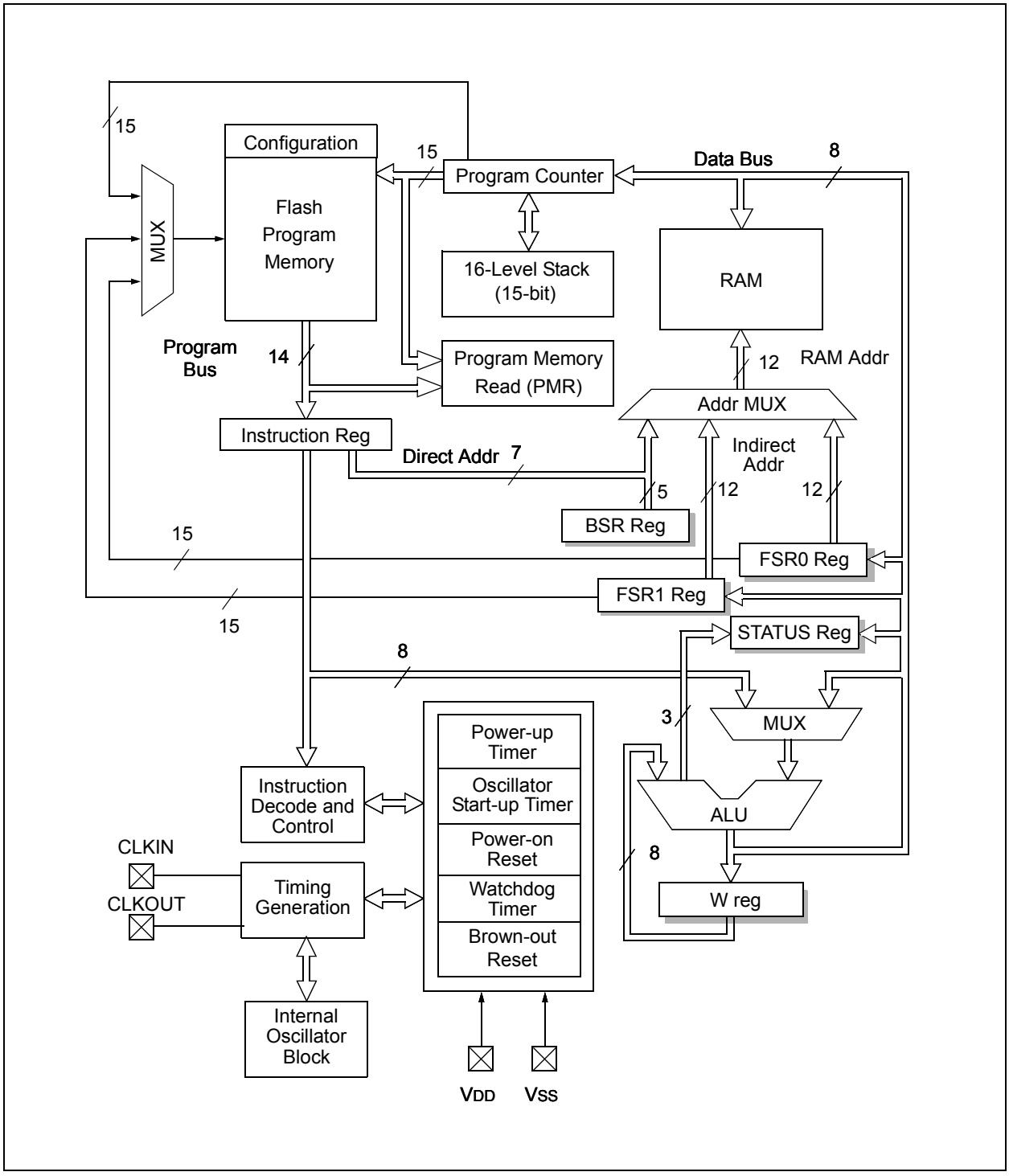

© 2013-2015 Microchip Technology Inc.

## 2.0 ENHANCED MID-RANGE CPU

This family of devices contain an enhanced mid-range 8-bit CPU core. The CPU has 49 instructions. Interrupt capability includes automatic context saving. The hardware stack is 16 levels deep and has Overflow and Underflow Reset capability. Direct, Indirect, and Relative addressing modes are available. Two File Select Registers (FSRs) provide the ability to read program and data memory.

- Automatic Interrupt Context Saving
- 16-level Stack with Overflow and Underflow
- File Select Registers
- Instruction Set

**FIGURE 2-1: CORE BLOCK DIAGRAM**



## 2.1 Automatic Interrupt Context Saving

During interrupts, certain registers are automatically saved in shadow registers and restored when returning from the interrupt. This saves stack space and user code. See **Section 7.5 "Automatic Context Saving"** for more information.

## 2.2 16-Level Stack with Overflow and Underflow

These devices have a hardware stack memory 15 bits wide and 16 words deep. A Stack Overflow or Underflow will set the appropriate bit (STKOVF or STKUNF) in the PCON register, and if enabled, will cause a software Reset. See **Section 3.6 "Stack"** for more details.

## 2.3 File Select Registers

There are two 16-bit File Select Registers (FSR). FSRs can access all file registers and program memory, which allows one Data Pointer for all memory. When an FSR points to program memory, there is one additional instruction cycle in instructions using INDF to allow the data to be fetched. General purpose memory can now also be addressed linearly, providing the ability to access contiguous data larger than 80 bytes. There are also new instructions to support the FSRs. See **Section 3.7 "Indirect Addressing"** for more details.

## 2.4 Instruction Set

There are 49 instructions for the enhanced mid-range CPU to support the features of the CPU. See **Section 25.0 "Instruction Set Summary"** for more details.


© 2013-2015 Microchip Technology Inc.

## 3.0 MEMORY ORGANIZATION

These devices contain the following types of memory:

- Program Memory
  - Configuration Words
  - Device ID
  - User ID
  - Flash Program Memory
- Data Memory
  - Core Registers
  - Special Function Registers
  - General Purpose RAM
  - Common RAM

**Note 1:** The method to access Flash memory through the PMCON registers is described in **Section 10.0 "Flash Program Memory Control"**.

The following features are associated with access and control of program memory and data memory:

- PCL and PCLATH
- Stack
- Indirect Addressing

### 3.1 Program Memory Organization

The enhanced mid-range core has a 15-bit program counter capable of addressing a 32K x 14 program memory space. Table 3-1 shows the memory sizes implemented for the PIC16(L)F1703/7 family. Accessing a location above these boundaries will cause a wrap-around within the implemented memory space. The Reset vector is at 0000h and the interrupt vector is at 0004h (see Figure 3-1).

### 3.2 High-Endurance Flash

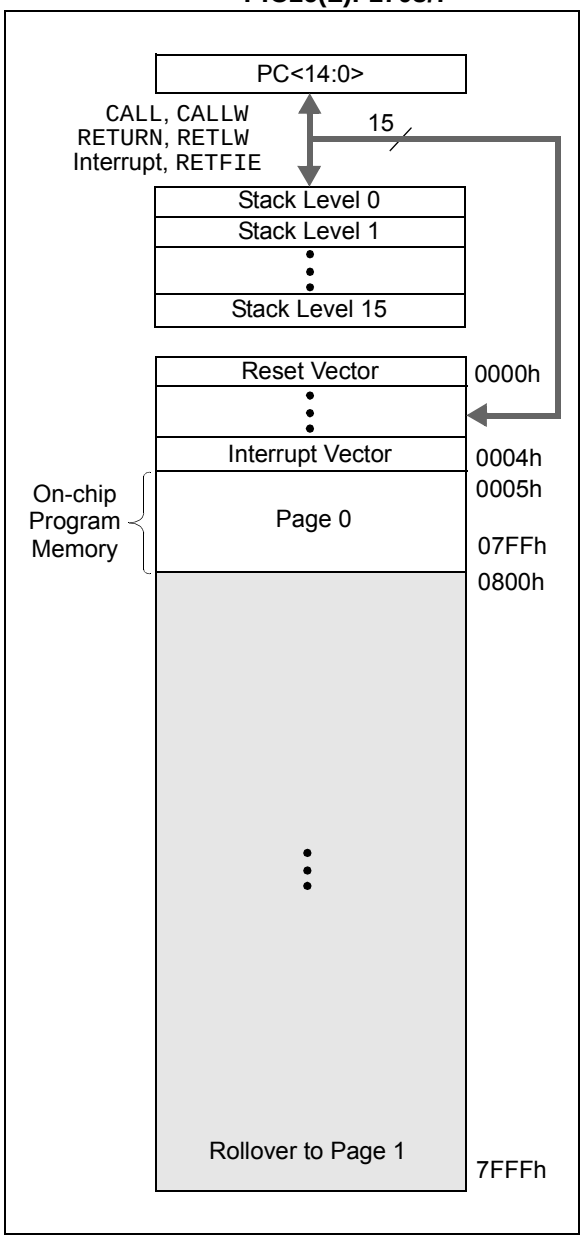
This device has a 128-byte section of high-endurance Program Flash Memory (PFM) in lieu of data EEPROM. This area is especially well suited for non-volatile data storage that is expected to be updated frequently over the life of the end product. See **Section 10.2 "Flash Program Memory Overview"** for more information on writing data to PFM. See **Section 3.2.1.2 "Indirect Read with FSR"** for more information about using the FSR registers to read byte data stored in PFM.

**TABLE 3-1: DEVICE SIZES AND ADDRESSES**

| Device | Program Memory Space (Words) | Last Program Memory Address | High-Endurance Flash Memory Address Range (1) |
|---|---|---|---|
| PIC16(L)F1703/7 | 2,048 | 07FFh | 0780h-07FFh |

**Note 1:** High-endurance Flash applies to low byte of each address in the range.

**FIGURE 3-1: PROGRAM MEMORY MAP AND STACK FOR PIC16(L)F1703/7**




### 3.2.1 READING PROGRAM MEMORY AS DATA

There are two methods of accessing constants in program memory. The first method is to use tables of RETLW instructions. The second method is to set an FSR to point to the program memory.

#### 3.2.1.1 RETLW Instruction

The RETLW instruction can be used to provide access to tables of constants. The recommended way to create such a table is shown in Example 3-1.

**EXAMPLE 3-1: RETLW INSTRUCTION**

```
constants
    BRW                 ;Add Index in W to
                        ;program counter to
                        ;select data
    RETLW DATA0         ;Index0 data
    RETLW DATA1         ;Index1 data
    RETLW DATA2
    RETLW DATA3


my_function
    ;… LOTS OF CODE…
    MOVLW       DATA_INDEX
    call constants
    ;… THE CONSTANT IS IN W
```

The BRW instruction makes this type of table very simple to implement. If your code must remain portable with previous generations of microcontrollers, then the BRW instruction is not available so the older table read method must be used.

#### 3.2.1.2 Indirect Read with FSR

The program memory can be accessed as data by setting bit 7 of the FSRxH register and reading the matching INDFx register. The MOVIW instruction will place the lower eight bits of the addressed word in the W register. Writes to the program memory cannot be performed via the INDF registers. Instructions that access the program memory via the FSR require one extra instruction cycle to complete. Example 3-2 demonstrates accessing the program memory via an FSR.

The high directive will set bit<7> if a label points to a location in program memory.


© 2013-2015 Microchip Technology Inc.

**EXAMPLE 3-2: ACCESSING PROGRAM MEMORY VIA FSR**

```
constants
    RETLW DATA0        ;Index0 data
    RETLW DATA1        ;Index1 data
    RETLW DATA2
    RETLW DATA3
my_function
    ;… LOTS OF CODE…
    MOVLW   LOW constants
    MOVWF   FSR1L
    MOVLW   HIGH constants
    MOVWF   FSR1H
    MOVIW   0[FSR1]
;THE PROGRAM MEMORY IS IN W
```

## 3.3 Data Memory Organization

The data memory is partitioned in 32 memory banks with 128 bytes in a bank. Each bank consists of (Figure 3-2):

- 12 core registers
- 20 Special Function Registers (SFR)
- Up to 80 bytes of General Purpose RAM (GPR)
- 16 bytes of common RAM

The active bank is selected by writing the bank number into the Bank Select Register (BSR). Unimplemented memory will read as '0'. All data memory can be accessed either directly (via instructions that use the file registers) or indirectly via the two File Select Registers (FSR). See **Section 3.7 "Indirect Addressing"** for more information.

Data memory uses a 12-bit address. The upper five bits of the address define the Bank address and the lower seven bits select the registers/RAM in that bank.

### 3.3.1 CORE REGISTERS

The core registers contain the registers that directly affect the basic operation. The core registers occupy the first 12 addresses of every data memory bank (addresses x00h/x08h through x0Bh/x8Bh). These registers are listed below in Table 3-2. For detailed information, see Table 3-9.

**TABLE 3-2: CORE REGISTERS**

| Addresses | BANKx |
|---|---|
| x00h or x80h | INDF0 |
| x01h or x81h | INDF1 |
| x02h or x82h | PCL |
| x03h or x83h | STATUS |
| x04h or x84h | FSR0L |
| x05h or x85h | FSR0H |
| x06h or x86h | FSR1L |
| x07h or x87h | FSR1H |
| x08h or x88h | BSR |
| x09h or x89h | WREG |
| x0Ah or x8Ah | PCLATH |
| x0Bh or x8Bh | INTCON |


© 2013-2015 Microchip Technology Inc.

#### 3.3.1.1 STATUS Register

The STATUS register, shown in Register 3-1, contains:

- the arithmetic status of the ALU
- the Reset status

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 (Refer to **Section 25.0 "Instruction Set Summary"**).

| **Note:** | The C and DC bits operate as $\overline{\text{Borrow}}$ and $\overline{\text{Digit Borrow}}$ out bits, respectively, in subtraction. |
|---|---|

## 3.4 Register Definitions: Status

**REGISTER 3-1: STATUS: STATUS REGISTER**

| U-0 | U-0 | U-0 | R-1/q | R-1/q | R/W-0/u | R/W-0/u | R/W-0/u |
|---|---|---|---|---|---|---|---|
| — | — | — | $\overline{TO}$ | $\overline{PD}$ | Z | DC(1) | C(1) |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| '1' = Bit is set | '0' = Bit is cleared | q = Value depends on condition |

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

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/$\overline{\text{Digit Borrow}}$ 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/$\overline{\text{Borrow}}$ bit(1) (`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 $\overline{\text{Borrow}}$, the polarity is reversed. A subtraction is executed by adding the two's complement of the second operand.

### 3.4.1 SPECIAL FUNCTION REGISTER

The Special Function Registers are registers used by the application to control the desired operation of peripheral functions in the device. The Special Function Registers occupy the 20 bytes after the core registers of every data memory bank (addresses x0Ch/x8Ch through x1Fh/x9Fh). The registers associated with the operation of the peripherals are described in the appropriate peripheral chapter of this data sheet.

### 3.4.2 GENERAL PURPOSE RAM

There are up to 80 bytes of GPR in each data memory bank. The Special Function Registers occupy the 20 bytes after the core registers of every data memory bank (addresses x0Ch/x8Ch through x1Fh/x9Fh).

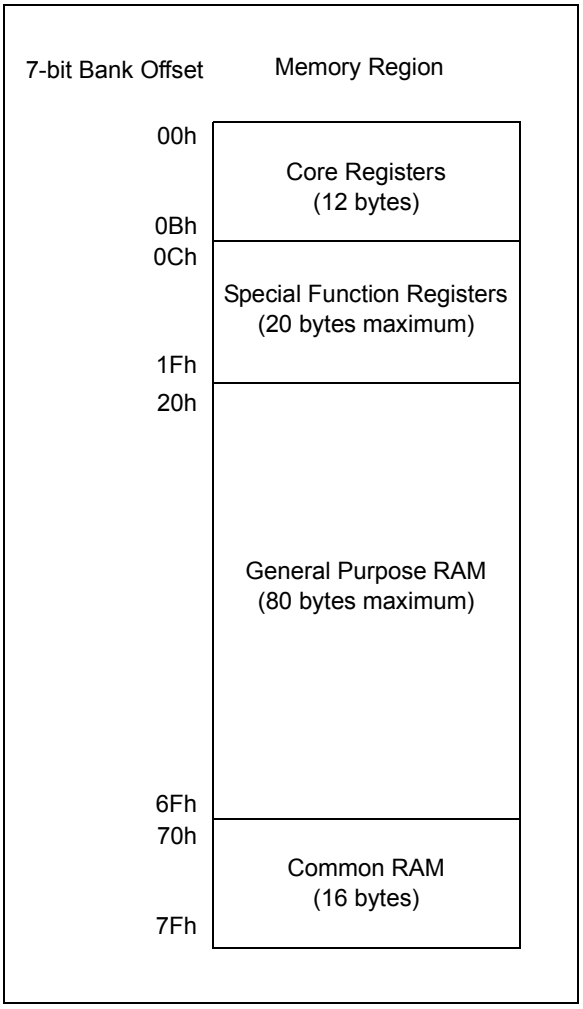
#### 3.4.2.1 Linear Access to GPR

The general purpose RAM can be accessed in a non-banked method via the FSRs. This can simplify access to large memory structures. See **Section 3.7.2 "Linear Data Memory"** for more information.

### 3.4.3 COMMON RAM

There are 16 bytes of common RAM accessible from all banks.

**FIGURE 3-2: BANKED MEMORY PARTITIONING**




### 3.4.4 DEVICE MEMORY MAPS

The memory maps for the device family are as shown in Tables 3-3 through 3-8.


© 2013-2015 Microchip Technology Inc.

**TABLE 3-3: PIC16(L)F1703/7 MEMORY MAP (BANKS 0-7)**

| BANK 0 | | BANK 1 | | BANK 2 | | BANK 3 | | BANK 4 | | BANK 5 | | BANK 6 | | BANK 7 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 000h | Core Registers (Table 3-2) | 080h | Core Registers (Table 3-2) | 100h | Core Registers (Table 3-2) | 180h | Core Registers (Table 3-2) | 200h | Core Registers (Table 3-2) | 280h | Core Registers (Table 3-2) | 300h | Core Registers (Table 3-2) | 380h | Core Registers (Table 3-2) |
| 00Bh | | 08Bh | | 10Bh | | 18Bh | | 20Bh | | 28Bh | | 30Bh | | 38Bh | |
| 00Ch | PORTA | 08Ch | TRISA | 10Ch | LATA | 18Ch | ANSELA | 20Ch | WPUA | 28Ch | ODCONA | 30Ch | SLRCONA | 38Ch | INLVLA |
| 00Dh | PORTB | 08Dh | TRISB | 10Dh | LATB | 18Dh | ANSELB | 20Dh | WPUB | 28Dh | ODCONB | 30Dh | SLRCONB | 38Dh | INLVLB |
| 00Eh | PORTC | 08Eh | TRISC | 10Eh | LATC | 18Eh | ANSELC | 20Eh | WPUC | 28Eh | ODCONC | 30Eh | SLRCONC | 38Eh | INLVLC |
| 00Fh | — | 08Fh | — | 10Fh | — | 18Fh | — | 20Fh | — | 28Fh | — | 30Fh | — | 38Fh | — |
| 010h | — | 090h | — | 110h | — | 190h | — | 210h | — | 290h | — | 310h | — | 390h | — |
| 011h | PIR1 | 091h | PIE1 | 111h | — | 191h | PMADR | 211h | SSP1BUF | 291h | CCPR1L | 311h | — | 391h | IOCAP |
| 012h | PIR2 | 092h | PIE2 | 112h | — | 191h | PMADRL | 212h | SSP1ADD | 292h | CCPR1H | 312h | — | 392h | IOCAN |
| 013h | PIR3 | 093h | PIE3 | 113h | — | 192h | PMADRH | 213h | SSP1MSK | 293h | CCP1CON | 313h | — | 393h | IOCAF |
| 014h | — | 094h | — | 114h | — | 193h | PMDATL | 214h | SSP1STAT | 294h | — | 314h | — | 394h | IOCBP |
| 015h | TMR0 | 095h | OPTION_REG | 115h | — | 194h | PMDATH | 215h | SSP1CON1 | 295h | — | 315h | — | 395h | IOCBN |
| 016h | TMR1L | 096h | PCON | 116h | BORCON | 195h | PMCON1 | 216h | SSP1CON1 | 296h | — | 316h | — | 396h | IOCBF |
| 017h | TMR1H | 097h | WDTCON | 117h | FVRCON | 196h | PMCON2 | 217h | SSP1CON2 | 297h | — | 317h | — | 397h | IOCCP |
| 018h | T1CON | 098h | OSCTUNE | 118h | — | 197h | — | 218h | SSP1CON3 | 298h | CCPR2 | 318h | — | 398h | IOCCN |
| 019h | T1GCON | 099h | OSCCON | 119h | — | 198h | — | 219h | — | 298h | CCPR2L | 319h | — | 399h | IOCCF |
| 01Ah | TMR2 | 09Ah | OSCSTAT | 11Ah | — | 199h | — | 21Ah | — | 299h | CCPR2H | 31Ah | — | 39Ah | — |
| 01Bh | PR2 | 09Bh | ADRESL | 11Bh | — | 19Ah | — | 21Bh | — | 29Ah | CCP2CON | 31Bh | — | 39Bh | — |
| 01Ch | T2CON | 09Ch | ADRESH | 11Ch | ZCD1CON | 19Bh | — | 21Ch | — | 29Bh | — | 31Ch | — | 39Ch | — |
| 01Dh | — | 09Dh | ADCON0 | 11Dh | — | 19Ch | — | 21Dh | — | 29Ch | — | 31Dh | — | 39Dh | — |
| 01Eh | — | 09Eh | ADCON1 | 11Eh | — | 19Dh | — | 21Eh | — | 29Dh | — | 31Eh | — | 39Eh | — |
| 01Fh | — | 09Fh | ADCON2 | 11Fh | — | 19Eh | — | 21Fh | — | 29Eh | — | 31Fh | — | 39Fh | — |
| 020h | General Purpose Register 80 Bytes | 0A0h | General Purpose Register 80 Bytes | 120h | General Purpose Register 80 Bytes | 19Fh | — | 220h | Unimplemented | 29Fh | — | 320h | Unimplemented | 3A0h | Unimplemented |
| 06Fh | | 0EFh | | 16Fh | | 1A0h | Unimplemented | 26Fh | | 2A0h | Unimplemented | 36Fh | | 3EFh | |
| 070h | Common RAM 70h - 7Fh | 0F0h | Accesses 70h - 7Fh | 170h | Accesses 70h - 7Fh | 1EFh | | 270h | Accesses 70h - 7Fh | 2EFh | | 370h | Accesses 70h - 7Fh | 3F0h | Accesses 70h - 7Fh |
| 07Fh | | 0FFh | | 17Fh | | 1F0h | Accesses 70h - 7Fh | 27Fh | | 2F0h | Accesses 70h - 7Fh | 37Fh | | 3FFh | |
| | | | | | | 1FFh | | | | 2FFh | | | | | |

© 2013-2015 Microchip Technology Inc.

**TABLE 3-4: PIC16(L)F1703/7 MEMORY MAP (BANKS 8-15)**

| BANK 8 | | BANK 9 | | BANK 10 | | BANK 11 | | BANK 12 | | BANK 13 | | BANK 14 | | BANK 15 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 400h | Core Registers (Table 3-2) | 480h | Core Registers (Table 3-2) | 500h | Core Registers (Table 3-2) | 580h | Core Registers (Table 3-2) | 600h | Core Registers (Table 3-2) | 680h | Core Registers (Table 3-2) | 700h | Core Registers (Table 3-2) | 780h | Core Registers (Table 3-2) |
| 40Bh | | 48Bh | | 50Bh | | 58Bh | | 60Bh | | 68Bh | | 70Bh | | 78Bh | |
| 40Ch | — | 48Ch | — | 50Ch | — | 58Ch | — | 60Ch | — | 68Ch | — | 70Ch | — | 78Ch | — |
| 40Dh | — | 48Dh | — | 50Dh | — | 58Dh | — | 60Dh | — | 68Dh | — | 70Dh | — | 78Dh | — |
| 40Eh | — | 48Eh | — | 50Eh | — | 58Eh | — | 60Eh | — | 68Eh | — | 70Eh | — | 78Eh | — |
| 40Fh | — | 48Fh | — | 50Fh | — | 58Fh | — | 60Fh | — | 68Fh | — | 70Fh | — | 78Fh | — |
| 410h | — | 490h | — | 510h | — | 590h | — | 610h | — | 690h | — | 710h | — | 790h | — |
| 411h | — | 491h | — | 511h | OPA1CON | 591h | — | 611h | — | 691h | — | 711h | — | 791h | — |
| 412h | — | 492h | — | 512h | — | 592h | — | 612h | — | 692h | — | 712h | — | 792h | — |
| 413h | — | 493h | — | 513h | — | 593h | — | 613h | — | 693h | — | 713h | — | 793h | — |
| 414h | — | 494h | — | 514h | — | 594h | — | 614h | — | 694h | — | 714h | — | 794h | — |
| 415h | — | 495h | — | 515h | OPA2CON | 595h | — | 615h | — | 695h | — | 715h | — | 795h | — |
| 416h | — | 496h | — | 516h | — | 596h | — | 616h | — | 696h | — | 716h | — | 796h | — |
| 417h | — | 497h | — | 517h | — | 597h | — | 617h | — | 697h | — | 717h | — | 797h | — |
| 418h | — | 498h | — | 518h | — | 598h | — | 618h | — | 698h | — | 718h | — | 798h | — |
| 419h | — | 499h | — | 519h | — | 599h | — | 619h | — | 699h | — | 719h | — | 799h | — |
| 41Ah | — | 49Ah | — | 51Ah | — | 59Ah | — | 61Ah | — | 69Ah | — | 71Ah | — | 79Ah | — |
| 41Bh | — | 49Bh | — | 51Bh | — | 59Bh | — | 61Bh | — | 69Bh | — | 71Bh | — | 79Bh | — |
| 41Ch | — | 49Ch | — | 51Ch | — | 59Ch | — | 61Ch | — | 69Ch | — | 71Ch | — | 79Ch | — |
| 41Dh | — | 49Dh | — | 51Dh | — | 59Dh | — | 61Dh | — | 69Dh | — | 71Dh | — | 79Dh | — |
| 41Eh | — | 49Eh | — | 51Eh | — | 59Eh | | 61Eh | — | 69Eh | — | 71Eh | — | 79Eh | — |
| 41Fh | — | 49Fh | — | 51Fh | — | 59Fh | | 61Fh | — | 69Fh | — | 71Fh | — | 79Fh | — |
| 420h | Unimplemented | 4A0h | Unimplemented | 520h | Unimplemented | 5A0h | Unimplemented | 620h | Unimplemented | 6A0h | Unimplemented | 720h | Unimplemented | 7A0h | Unimplemented |
| 46Fh | | 4EFh | | 56Fh | | 5EFh | | 66Fh | | 6EFh | | 76Fh | | 7EFh | |
| 470h | Accesses 70h - 7Fh | 4F0h | Accesses 70h - 7Fh | 570h | Accesses 70h - 7Fh | 5F0h | Accesses 70h - 7Fh | 670h | Accesses 70h - 7Fh | 6F0h | Accesses 70h - 7Fh | 770h | Accesses 70h - 7Fh | 7F0h | Accesses 70h - 7Fh |
| 47Fh | | 4FFh | | 57Fh | | 5FFh | | 67Fh | | 6FFh | | 77Fh | | 7FFh | |



© 2013-2015 Microchip Technology Inc.

**TABLE 3-5: PIC16(L)F1703/7 MEMORY MAP (BANKS 16-23)**

| BANK 16 | | BANK 17 | | BANK 18 | | BANK 19 | | BANK 20 | | BANK 21 | | BANK 22 | | BANK 23 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 800h | Core Registers (Table 3-2) | 880h | Core Registers (Table 3-2) | 900h | Core Registers (Table 3-2) | 980h | Core Registers (Table 3-2) | A00h | Core Registers (Table 3-2) | A80h | Core Registers (Table 3-2) | B00h | Core Registers (Table 3-2) | B80h | Core Registers (Table 3-2) |
| 80Bh | | 88Bh | | 90Bh | | 98Bh | | A0Bh | | A8Bh | | B0Bh | | B8Bh | |
| 80Ch | — | 88Ch | — | 90Ch | — | 98Ch | — | A0Ch | — | A8Ch | — | B0Ch | — | B8Ch | — |
| 80Dh | — | 88Dh | — | 90Dh | — | 98Dh | — | A0Dh | — | A8Dh | — | B0Dh | — | B8Dh | — |
| 80Eh | — | 88Eh | — | 90Eh | — | 98Eh | — | A0Eh | — | A8Eh | — | B0Eh | — | B8Eh | — |
| 80Fh | — | 88Fh | — | 90Fh | — | 98Fh | — | A0Fh | — | A8Fh | — | B0Fh | — | B8Fh | — |
| 810h | — | 890h | — | 910h | — | 990h | — | A10h | — | A90h | — | B10h | — | B90h | — |
| 811h | — | 891h | — | 911h | — | 991h | — | A11h | — | A91h | — | B11h | — | B91h | — |
| 812h | — | 892h | — | 912h | — | 992h | — | A12h | — | A92h | — | B12h | — | B92h | — |
| 813h | — | 893h | — | 913h | — | 993h | — | A13h | — | A93h | — | B13h | — | B93h | — |
| 814h | — | 894h | — | 914h | — | 994h | — | A14h | — | A94h | — | B14h | — | B94h | — |
| 815h | — | 895h | — | 915h | — | 995h | — | A15h | — | A95h | — | B15h | — | B95h | — |
| 816h | — | 896h | — | 916h | — | 996h | — | A16h | — | A96h | — | B16h | — | B96h | — |
| 817h | — | 897h | — | 917h | — | 997h | — | A17h | — | A97h | — | B17h | — | B97h | — |
| 818h | — | 898h | — | 918h | — | 998h | — | A18h | — | A98h | — | B18h | — | B98h | — |
| 819h | — | 899h | — | 919h | — | 999h | — | A19h | — | A99h | — | B19h | — | B99h | — |
| 81Ah | — | 89Ah | — | 91Ah | — | 99Ah | — | A1Ah | — | A9Ah | — | B1Ah | — | B9Ah | — |
| 81Bh | — | 89Bh | — | 91Bh | — | 99Bh | — | A1Bh | — | A9Bh | — | B1Bh | — | B9Bh | — |
| 81Ch | — | 89Ch | — | 91Ch | — | 99Ch | — | A1Ch | — | A9Ch | — | B1Ch | — | B9Ch | — |
| 81Dh | — | 89Dh | — | 91Dh | — | 99Dh | — | A1Dh | — | A9Dh | — | B1Dh | — | B9Dh | — |
| 81Eh | — | 89Eh | — | 91Eh | — | 99Eh | — | A1Eh | — | A9Eh | — | B1Eh | — | B9Eh | — |
| 81Fh | — | 89Fh | — | 91Fh | — | 99Fh | — | A1Fh | — | A9Fh | — | B1Fh | — | B9Fh | — |
| 820h | Unimplemented | 8A0h | Unimplemented | 920h | Unimplemented | 9A0h | Unimplemented | A20h | Unimplemented | AA0h | Unimplemented | B20h | Unimplemented | BA0h | Unimplemented |
| 86Fh | | 8EFh | | 96Fh | | 9EFh | | A6Fh | | AEFh | | B6Fh | | BEFh | |
| 870h | Accesses 70h - 7Fh | 8F0h | Accesses 70h - 7Fh | 970h | Accesses 70h - 7Fh | 9F0h | Accesses 70h - 7Fh | A70h | Accesses 70h - 7Fh | AF0h | Accesses 70h - 7Fh | B70h | Accesses 70h - 7Fh | BF0h | Accesses 70h - 7Fh |
| 87Fh | | 8FFh | | 97Fh | | 9FFh | | A7Fh | | AFFh | | B7Fh | | BFFh | |

© 2013-2015 Microchip Technology Inc.

**TABLE 3-6: PIC16(L)F1703/7 MEMORY MAP (BANKS 24-31)**

| BANK 24 | BANK 25 | BANK 26 | BANK 27 | BANK 28 | BANK 29 | BANK 30 | BANK 31 |
|---|---|---|---|---|---|---|---|
| C00h | C80h | D00h | D80h | E00h | E80h | F00h | F80h |
| Core Registers (Table 3-2) | Core Registers (Table 3-2) | Core Registers (Table 3-2) | Core Registers (Table 3-2) | Core Registers (Table 3-2) | Core Registers (Table 3-2) | Core Registers (Table 3-2) | Core Registers (Table 3-2) |
| C0Bh | C8Bh | D0Bh | D8Bh | E0Bh | E8Bh | F0Bh | F8Bh |
| C0Ch – C1Fh: — | C8Ch – C9Fh: — | D0Ch – D1Fh: — | D8Ch – D9Fh: — | E0Ch – E6Fh: See Table 3-7 for register mapping details | E8Ch – EEFh: See Table 3-7 for register mapping details | F0Ch – F6Fh: See Table 3-7 for register mapping details | F8Ch – FEFh: See Table 3-8 for register mapping details |
| C20h | CA0h | D20h | DA0h | E20h | EA0h | F20h | FA0h |
| Unimplemented | Unimplemented | Unimplemented | Unimplemented | | | | |
| C6Fh | CEFh | D6Fh | DEFh | E6Fh | EEFh | F6Fh | FEFh |
| C70h | CF0h | D70h | DF0h | E70h | EF0h | F70h | FF0h |
| Accesses 70h - 7Fh | Accesses 70h - 7Fh | Accesses 70h - 7Fh | Accesses 70h - 7Fh | Accesses 70h - 7Fh | Accesses 70h - 7Fh | Accesses 70h - 7Fh | Accesses 70h - 7Fh |
| C7Fh | CFFh | D7Fh | DFFh | E7Fh | EFFh | F7Fh | FFFh |



© 2013-2015 Microchip Technology Inc.

**TABLE 3-7: PIC16(L)F1703/7 MEMORY MAP (BANKS 28-31)**

| | Bank 28 | | Bank 29 | | Bank 30 |
|---|---|---|---|---|---|
| E0Ch | — | E8Ch | — | F0Ch | — |
| E0Dh | — | E8Dh | — | F0Dh | — |
| E0Eh | — | E8Eh | — | F0Eh | — |
| E0Fh | PPSLOCK | E8Fh | — | F0Fh | — |
| E10h | INTPPS | E90h | RA0PPS | F10h | — |
| E11h | T0CKIPPS | E91h | RA1PPS | F11h | — |
| E12h | T1CKIPPS | E92h | RA2PPS | F12h | — |
| E13h | T1GPPS | E93h | — | F13h | — |
| E14h | CCP1PPS | E94h | RA4PPS | F14h | — |
| E15h | CCP2PPS | E95h | RA5PPS | F15h | — |
| E16h | — | E96h | — | F16h | — |
| E17h | — | E97h | — | F17h | — |
| E18h | — | E98h | — | F18h | — |
| E19h | — | E99h | — | F19h | — |
| E1Ah | — | E9Ah | — | F1Ah | — |
| E1Bh | — | E9Bh | — | F1Bh | — |
| E1Ch | — | E9Ch | RB4PPS(1) | F1Ch | — |
| E1Dh | — | E9Dh | RB5PPS(1) | F1Dh | — |
| E1Eh | — | E9Eh | RB6PPS(1) | F1Eh | — |
| E1Fh | — | E9Fh | RB7PPS(1) | F1Fh | — |
| E20h | SSPCLKPPS | EA0h | RC0PPS | F20h | — |
| E21h | SSPDATPPS | EA1h | RC1PPS | F21h | — |
| E22h | SSPSSPPS | EA2h | RC2PPS | F22h | — |
| E23h | — | EA3h | RC3PPS | F23h | — |
| E24h | — | EA4h | RC4PPS | F24h | — |
| E25h | — | EA5h | RC5PPS | F25h | — |
| E26h | — | EA6h | RC6PPS(1) | F26h | — |
| E27h | — | EA7h | RC7PPS(1) | F27h | — |
| E28h | — | EA8h | — | F28h | — |
| E29h | — | EA9h | — | F29h | — |
| E2Ah | — | EAAh | — | F2Ah | — |
| E2Bh | — | EABh | — | F2Bh | — |
| E2Ch | — | EACh | — | F2Ch | — |
| E2Dh | — | EADh | — | F2Dh | — |
| E2Eh | — | EAEh | — | F2Eh | — |
| E2Fh | — | EAFh | — | F2Fh | — |
| E30h | — | EB0h | — | F30h | — |
| E31h | — | EB1h | — | F31h | — |
| E32h | — | EB2h | — | F32h | — |
| E33h | — | EB3h | — | F33h | — |
| E34h | — | EB4h | — | F34h | — |
| E35h | — | EB5h | — | F35h | — |
| E36h | — | EB6h | — | F36h | — |
| E37h | — | EB7h | — | F37h | — |
| E38h | — | EB8h | — | F38h | — |
| E39h | — | EB9h | — | F39h | — |
| E3Ah | — | EBAh | — | F3Ah | — |
| E3Bh | — | EBBh | — | F3Bh | — |
| E3Ch | — | EBCh | — | F3Ch | — |
| E3Dh | — | EBDh | — | F3Dh | — |
| E3Eh | — | EBEh | — | F3Eh | — |
| E3Fh | — | EBFh | — | F3Fh | — |
| E40h | — | EC0h | — | F40h | — |
| E6Fh | | EEFh | | F6Fh | |

Legend: = Unimplemented data memory locations, read as '0',

**Note 1:** Only available on PIC16(L)F1707 devices

**TABLE 3-8: PIC16(L)F1703/7 MEMORY MAP, BANK 31**

| Bank 31 | |
|---|---|
| F8Ch<br>FE3h | Unimplemented<br>Read as '0' |
| FE4h | STATUS_SHAD |
| FE5h | WREG_SHAD |
| FE6h | BSR_SHAD |
| FE7h | PCLATH_SHAD |
| FE8h | FSR0L_SHAD |
| FE9h | FSR0H_SHAD |
| FEAh | FSR1L_SHAD |
| FEBh | FSR1H_SHAD |
| FECh | — |
| FEDh | STKPTR |
| FEEh | TOSL |
| FEFh | TOSH |

Legend: = Unimplemented data memory locations, read as '0',


© 2013-2015 Microchip Technology Inc.

### 3.4.5 CORE FUNCTION REGISTERS SUMMARY

The Core Function registers listed in Table 3-9 can be addressed from any Bank.

**TABLE 3-9: CORE FUNCTION REGISTERS SUMMARY**

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Bank 0-31** | | | | | | | | | | | |
| x00h or x80h | INDF0 | Addressing this location uses contents of FSR0H/FSR0L to address data memory (not a physical register) | | | | | | | | xxxx xxxx | uuuu uuuu |
| x01h or x81h | INDF1 | Addressing this location uses contents of FSR1H/FSR1L to address data memory (not a physical register) | | | | | | | | xxxx xxxx | uuuu uuuu |
| x02h or x82h | PCL | Program Counter (PC) Least Significant Byte | | | | | | | | 0000 0000 | 0000 0000 |
| x03h or x83h | STATUS | — | — | — | $\overline{TO}$ | $\overline{PD}$ | Z | DC | C | ---1 1000 | ---q quuu |
| x04h or x84h | FSR0L | Indirect Data Memory Address 0 Low Pointer | | | | | | | | 0000 0000 | uuuu uuuu |
| x05h or x85h | FSR0H | Indirect Data Memory Address 0 High Pointer | | | | | | | | 0000 0000 | 0000 0000 |
| x06h or x86h | FSR1L | Indirect Data Memory Address 1 Low Pointer | | | | | | | | 0000 0000 | uuuu uuuu |
| x07h or x87h | FSR1H | Indirect Data Memory Address 1 High Pointer | | | | | | | | 0000 0000 | 0000 0000 |
| x08h or x88h | BSR | — | — | — | BSR4 | BSR3 | BSR2 | BSR1 | BSR0 | ---0 0000 | ---0 0000 |
| x09h or x89h | WREG | Working Register | | | | | | | | 0000 0000 | uuuu uuuu |
| x0Ah or x8Ah | PCLATH | — | Write Buffer for upper seven bits of the Program Counter | | | | | | | -000 0000 | -000 0000 |
| x0Bh or x8Bh | INTCON | GIE | PEIE | TMR0IE | INTE | IOCIE | TMR0IF | INTF | IOCIF | 0000 000x | 0000 000u |

**Legend:** x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, read as '0', r = reserved. Shaded locations are unimplemented, read as '0'.

**TABLE 3-10: SPECIAL FUNCTION REGISTER SUMMARY**

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Bank 0** | | | | | | | | | | | |
| 00Ch | PORTA | — | — | RA5 | RA4 | RA3 | RA2 | RA1 | RA0 | --xx xxxx | --uu uuuu |
| 00Dh | PORTB(3) | RB7 | RB6 | RB5 | RB4 | — | — | — | — | xxxx ---- | uuuu ---- |
| 00Eh | PORTC | RC7(3) | RC6(3) | RC5 | RC4 | RC3 | RC2 | RC1 | RC0 | xxxx xxxx | uuuu uuuu |
| 00Fh | — | Unimplemented | | | | | | | | — | — |
| 010h | — | Unimplemented | | | | | | | | — | — |
| 011h | PIR1 | TMR1GIF | ADIF | — | — | SSP1IF | CCP1IF | TMR2IF | TMR1IF | 00-- 0000 | 00-- 0000 |
| 012h | PIR2 | — | — | — | — | BCL1IF | — | — | CCP2IF | ---- 0--0 | ---- 0--0 |
| 013h | PIR3 | — | — | — | ZCDIF | — | — | — | — | ---0 ---- | ---0 ---- |
| 014h | — | Unimplemented | | | | | | | | — | — |
| 015h | TMR0 | Timer0 Module Register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 016h | TMR1L | Holding Register for the Least Significant Byte of the 16-bit TMR1 Register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 017h | TMR1H | Holding Register for the Most Significant Byte of the 16-bit TMR1 Register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 018h | T1CON | TMR1CS<1:0> | | T1CKPS<1:0> | | — | $\overline{\text{T1SYNC}}$ | — | TMR1ON | 0000 -0-0 | uuuu -u-u |
| 019h | T1GCON | TMR1GE | T1GPOL | T1GTM | T1GSPM | T1GGO/$\overline{\text{DONE}}$ | T1GVAL | T1GSS<1:0> | | 0000 0x00 | uuuu uxuu |
| 01Ah | TMR2 | Holding Register for the Least Significant Byte of the 16-bit TMR2 Register | | | | | | | | 0000 0000 | 0000 0000 |
| 01Bh | PR2 | Holding Register for the Most Significant Byte of the 16-bit TMR2 Register | | | | | | | | 1111 1111 | 1111 1111 |
| 01Ch | T2CON | — | T2OUTPS<3:0> | | | | TMR2ON | T2CKPS<1:0> | | -000 0000 | -000 0000 |
| 01Dh to 01Fh | — | Unimplemented | | | | | | | | — | — |

**Legend:** x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, read as '0', r = reserved. Shaded locations are unimplemented, read as '0'.

**Note 1:** Unimplemented, read as '1'.

**2:** PIC16(L)F1703 only.

**3:** PIC16(L)F1707 only.

**4:** Unimplemented on PIC16LF1703/7.

© 2013-2015 Microchip Technology Inc.

**TABLE 3-10: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)**

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR, BOR | Value on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Bank 1** | | | | | | | | | | | |
| 08Ch | TRISA | — | — | TRISA5 | TRISA4 | —(1) | TRISA2 | TRISA1 | TRISA0 | --11 1111 | --11 1111 |
| 08Dh | TRISB(3) | TRISB7 | TRISB6 | TRISB5 | TRISB4 | — | — | — | — | 1111 ---- | 1111 ---- |
| 08Eh | TRISC | TRISC7(3) | TRISC6(3) | TRISC5 | TRISC4 | TRISC3 | TRISC2 | TRISC1 | TRISC0 | 1111 1111 | 1111 1111 |
| 08Fh | — | Unimplemented | | | | | | | | — | — |
| 090h | — | Unimplemented | | | | | | | | — | — |
| 091h | PIE1 | TMR1GIE | ADIE | — | — | SSP1IE | CCP1IE | TMR2IE | TMR1IE | 00-- 0000 | 0000 0000 |
| 092h | PIE2 | — | — | — | — | BCL1IE | — | — | CCP2IE | ---- 0--0 | 000- 0000 |
| 093h | PIE3 | — | — | — | ZCDIE | — | — | — | — | ---0 ---- | ---0 ---- |
| 094h | — | Unimplemented | | | | | | | | — | — |
| 095h | OPTION_REG | $\overline{\text{WPUEN}}$ | INTEDG | TMR0CS | TMR0SE | PSA | PS<2:0> | | | 1111 1111 | 1111 1111 |
| 096h | PCON | STKOVF | STKUNF | — | $\overline{\text{RWDT}}$ | $\overline{\text{RMCLR}}$ | $\overline{\text{RI}}$ | $\overline{\text{POR}}$ | $\overline{\text{BOR}}$ | 00-1 11qq | qq-q qquu |
| 097h | WDTCON | — | — | WDTPS<4:0> | | | | | SWDTEN | --01 0110 | --01 0110 |
| 098h | OSCTUNE | — | — | TUN<5:0> | | | | | | --00 0000 | --00 0000 |
| 099h | OSCCON | SPLLEN | IRCF<3:0> | | | | — | SCS<1:0> | | 0011 1-00 | 0011 1-00 |
| 09Ah | OSCSTAT | — | PLLR | OSTS | HFIOFR | HFIOFL | MFIOFR | LFIOFR | HFIOFS | -qq0 0q0q | qqqq -q0q |
| 09Bh | ADRES | — | — | — | — | — | — | — | — | — | — |
| 09Bh | ADRESL | ADC Result Register Low | | | | | | | | xxxx xxxx | uuuu uuuu |
| 09Ch | ADRESH | ADC Result Register High | | | | | | | | xxxx xxxx | uuuu uuuu |
| 09Dh | ADCON0 | — | CHS<4:0> | | | | | GO/$\overline{\text{DONE}}$ | ADON | -000 0000 | -000 0000 |
| 09Eh | ADCON1 | ADFM | ADCS<2:0> | | | — | ADNREF | ADPREF<1:0> | | 0000 -000 | 0000 -000 |
| 09Fh | ADCON2 | TRIGSEL<3:0> | | | | — | — | — | — | 0000 ---- | 0000 ---- |

**Legend:** x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, read as '0', r = reserved. Shaded locations are unimplemented, read as '0'.

**Note 1:** Unimplemented, read as '1'.

**2:** PIC16(L)F1703 only.

**3:** PIC16(L)F1707 only.

**4:** Unimplemented on PIC16LF1703/7.

© 2013-2015 Microchip Technology Inc.

**TABLE 3-10: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)**

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR, BOR | Value on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Bank 2** | | | | | | | | | | | |
| 10Ch | LATA | — | — | LATA5 | LATA4 | — | LATA2 | LATA1 | LATA0 | --xx -xxx | --uu -uuu |
| 10Dh | LATB(3) | LATB7 | LATB6 | LATB5 | LATB4 | — | — | — | — | xxxx ---- | uuuu ---- |
| 10Eh | LATC | LATC7(3) | LATC6(3) | LATC5 | LATC4 | LATC3 | LATC2 | LATC1 | LATC0 | xxxx xxxx | uuuu uuuu |
| 10Fh | — | Unimplemented | | | | | | | | — | — |
| 110h | — | Unimplemented | | | | | | | | — | — |
| 111h | — | Unimplemented | | | | | | | | — | — |
| 112h | — | Unimplemented | | | | | | | | — | — |
| 113h | — | Unimplemented | | | | | | | | — | — |
| 114h | — | Unimplemented | | | | | | | | — | — |
| 115h | — | Unimplemented | | | | | | | | — | — |
| 116h | BORCON | SBOREN | BORFS | — | — | — | — | — | BORRDY | 10-- ---q | uu-- ---u |
| 117h | FVRCON | FVREN | FVRRDY | TSEN | TSRNG | CDAFVR<1:0> | | ADFVR<1:0> | | 0q00 0000 | 0q00 0000 |
| 118h | — | Unimplemented | | | | | | | | — | — |
| 119h | — | Unimplemented | | | | | | | | — | — |
| 11Ah | — | Unimplemented | | | | | | | | — | — |
| 11Bh | — | Unimplemented | | | | | | | | — | — |
| 11Ch | ZCD1CON | ZCD1EN | — | ZCD1OUT | ZCD1POL | — | — | ZCD1INTP | ZCD1INTN | 0-x0 --00 | 0-x0 --00 |
| 11Dh | — | Unimplemented | | | | | | | | — | — |
| 11Eh | — | Unimplemented | | | | | | | | — | — |
| 11Fh | — | Unimplemented | | | | | | | | — | — |

**Legend:** x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, read as '0', r = reserved.
Shaded locations are unimplemented, read as '0'.

**Note 1:** Unimplemented, read as '1'.
**2:** PIC16(L)F1703 only.
**3:** PIC16(L)F1707 only.
**4:** Unimplemented on PIC16LF1703/7.


© 2013-2015 Microchip Technology Inc.

**TABLE 3-10: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)**

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR, BOR | Value on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Bank 3** | | | | | | | | | | | |
| 18Ch | ANSELA | — | — | — | ANSA4 | — | ANSA2 | ANSA1 | ANSA0 | ---1 -111 | ---1 -111 |
| 18Dh | ANSELB(3) | — | — | ANSB5 | ANSB4 | — | — | — | — | --11 ---- | --11 ---- |
| 18Eh | ANSELC | ANSC7(3) | ANSC6(3) | ANSC5(2) | ANSC4(2) | ANSC3 | ANSC2 | ANSC1 | ANSC0 | 11-- 1111 | 11-- 1111 |
| 18Fh | — | Unimplemented | | | | | | | | — | — |
| 190h | — | Unimplemented | | | | | | | | — | — |
| 191h | PMADRL | Program Memory Address Register Low Byte | | | | | | | | 0000 0000 | 0000 0000 |
| 192h | PMADRH | —(1) | Program Memory Address Register High Byte | | | | | | | 1000 0000 | 1000 0000 |
| 193h | PMDATL | Program Memory Read Data Register Low Byte | | | | | | | | xxxx xxxx | uuuu uuuu |
| 194h | PMDATH | — | — | Program Memory Read Data Register High Byte | | | | | | --xx xxxx | --uu uuuu |
| 195h | PMCON1 | — | CFGS | LWLO | FREE | WRERR | WREN | WR | RD | -000 x000 | -000 q000 |
| 196h | PMCON2 | Program Memory Control Register 2 | | | | | | | | 0000 0000 | 0000 0000 |
| 197h | VREGCON(4) | — | — | — | — | — | — | VREGPM | Reserved | ---- --01 | ---- --01 |
| 198h | — | Unimplemented | | | | | | | | — | — |
| 199h | — | Unimplemented | | | | | | | | — | — |
| 19Ah | — | Unimplemented | | | | | | | | — | — |
| 19Bh | — | Unimplemented | | | | | | | | — | — |
| 19Ch | — | Unimplemented | | | | | | | | — | — |
| 19Dh | — | Unimplemented | | | | | | | | — | — |
| 19Eh | — | Unimplemented | | | | | | | | — | — |
| 19Fh | — | Unimplemented | | | | | | | | — | — |

**Legend:** x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, read as '0', r = reserved.
Shaded locations are unimplemented, read as '0'.

**Note 1:** Unimplemented, read as '1'.
**2:** PIC16(L)F1703 only.
**3:** PIC16(L)F1707 only.
**4:** Unimplemented on PIC16LF1703/7.

© 2013-2015 Microchip Technology Inc.

**TABLE 3-10: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)**

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR, BOR | Value on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Bank 4** | | | | | | | | | | | |
| 20Ch | WPUA | — | — | WPUA5 | WPUA4 | WPUA3 | WPUA2 | WPUA1 | WPUA0 | --11 1111 | --11 1111 |
| 20Dh | WPUB(3) | WPUB7 | WPUB6 | WPUB5 | WPUB4 | — | — | — | — | 1111 ---- | 1111 ---- |
| 20Eh | WPUC | WPUC7(3) | WPUC6(3) | WPUC5 | WPUC4 | WPUC3 | WPUC2 | WPUC1 | WPUC0 | 1111 1111 | 1111 1111 |
| 20Fh | — | Unimplemented | | | | | | | | — | — |
| 210h | — | Unimplemented | | | | | | | | — | — |
| 211h | SSP1BUF | Synchronous Serial Port Receive Buffer/Transmit Register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 212h | SSP1ADD | ADD<7:0> | | | | | | | | xxxx xxxx | uuuu uuuu |
| 213h | SSP1MSK | MSK<7:0> | | | | | | | | xxxx xxxx | uuuu uuuu |
| 214h | SSP1STAT | SMP | CKE | D/$\overline{A}$ | P | S | R/$\overline{W}$ | UA | BF | 0000 0000 | 0000 0000 |
| 215h | SSP1CON1 | WCOL | SSPOV | SSPEN | CKP | SSPM | | | | 0000 0000 | 0000 0000 |
| 216h | SSP1CON2 | GCEN | ACKSTAT | ACKDT | ACKEN | RCEN | PEN | RSEN | SEN | 0000 0000 | 0000 0000 |
| 217h | SSP1CON3 | ACKTIM | PCIE | SCIE | BOEN | SDAHT | SBCDE | AHEN | DHEN | 0000 0000 | 0000 0000 |
| 218h — 21Fh | — | Unimplemented | | | | | | | | — | — |

**Legend:** x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, read as '0', r = reserved. Shaded locations are unimplemented, read as '0'.

**Note 1:** Unimplemented, read as '1'.

**2:** PIC16(L)F1703 only.

**3:** PIC16(L)F1707 only.

**4:** Unimplemented on PIC16LF1703/7.



© 2013-2015 Microchip Technology Inc.

**TABLE 3-10: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)**

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR, BOR | Value on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Bank 5** | | | | | | | | | | | |
| 28Ch | ODCONA | — | — | ODA5 | ODA4 | — | ODA2 | ODA1 | ODA0 | --00 -000 | --00 -000 |
| 28Dh | ODCONB(3) | ODB7 | ODB6 | ODB5 | ODB4 | — | — | — | — | 0000 ---- | 0000 ---- |
| 28Eh | ODCONC | ODC7(3) | ODC6(3) | ODC5 | ODC4 | ODC3 | ODC2 | ODC1 | ODC0 | 0000 0000 | 0000 0000 |
| 28Fh | — | Unimplemented | | | | | | | | — | — |
| 290h | — | Unimplemented | | | | | | | | — | — |
| 291h | CCPR1L | Capture/Compare/PWM Register 1 (LSB) | | | | | | | | xxxx xxxx | uuuu uuuu |
| 292h | CCPR1H | Capture/Compare/PWM Register 1 (MSB) | | | | | | | | xxxx xxxx | uuuu uuuu |
| 293h | CCP1CON | — | — | DC1B<1:0> | | CCP1M<3:0> | | | | --00 0000 | --00 0000 |
| 294h — 297h | — | Unimplemented | | | | | | | | — | — |
| 298h | CCPR2L | Capture/Compare/PWM Register 2 (LSB) | | | | | | | | xxxx xxxx | uuuu uuuu |
| 299h | CCPR2H | Capture/Compare/PWM Register 2 (MSB) | | | | | | | | xxxx xxxx | uuuu uuuu |
| 29Ah | CCP2CON | — | — | DC2B<1:0> | | CCP2M<3:0> | | | | --00 0000 | --00 0000 |
| 29Bh — 29Fh | — | Unimplemented | | | | | | | | — | — |

**Legend:** x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, read as '0', r = reserved.
Shaded locations are unimplemented, read as '0'.

**Note 1:** Unimplemented, read as '1'.
**2:** PIC16(L)F1703 only.
**3:** PIC16(L)F1707 only.
**4:** Unimplemented on PIC16LF1703/7.

© 2013-2015 Microchip Technology Inc.

**TABLE 3-10: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)**

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR, BOR | Value on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Bank 6** | | | | | | | | | | | |
| 30Ch | SLRCONA | — | — | SLRA5 | SLRA4 | — | SLRA2 | SLRA1 | SLRA0 | --11 -111 | --11 -111 |
| 30Dh | SLRCONB(3) | SLRB7 | SLRB6 | SLRB5 | SLRB4 | — | — | — | — | 1111 ---- | 1111 ---- |
| 30Eh | SLRCONC | SLRC7(3) | SLRC6(3) | SLRC5 | SLRC4 | SLRC3 | SLRC2 | SLRC1 | SLRC0 | 1111 1111 | 1111 1111 |
| 30Fh — 31Fh | — | Unimplemented | | | | | | | | — | — |
| **Bank 7** | | | | | | | | | | | |
| 38Ch | INLVLA | — | — | INLVLA5 | INLVLA4 | INLVLA3 | INLVLA2 | INLVLA1 | INLVLA0 | --11 1111 | --11 1111 |
| 38Dh | INLVLB(3) | INLVLB7 | INLVLB6 | INLVLB5 | INLVLB4 | — | — | — | — | 1111 ---- | 1111 ---- |
| 38Eh | INLVLC | INLVLC7(3) | INLVLC6(3) | INLVLC5 | INLVLC4 | INLVLC3 | INLVLC2 | INLVLC1 | INLVLC0 | 1111 1111 | 1111 1111 |
| 38Fh | — | Unimplemented | | | | | | | | — | — |
| 390h | — | Unimplemented | | | | | | | | — | — |
| 391h | IOCAP | — | — | IOCAP5 | IOCAP4 | IOCAP3 | IOCAP2 | IOCAP1 | IOCAP0 | --00 0000 | --00 0000 |
| 392h | IOCAN | — | — | IOCAN5 | IOCAN4 | IOCAN3 | IOCAN2 | IOCAN1 | IOCAN0 | --00 0000 | --00 0000 |
| 393h | IOCAF | — | — | IOCAF5 | IOCAF4 | IOCAF3 | IOCAF2 | IOCAF1 | IOCAF0 | --00 0000 | --00 0000 |
| 394h | IOCBP(3) | IOCBP7 | IOCBP6 | IOCBP5 | IOCBP4 | — | — | — | — | 0000 ---- | 0000 ---- |
| 395h | IOCBN(3) | IOCBN7 | IOCBN6 | IOCBN5 | IOCBN4 | — | — | — | — | 0000 ---- | 0000 ---- |
| 396h | IOCBF(3) | IOCBF7 | IOCBF6 | IOCBF5 | IOCBF4 | — | — | — | — | 0000 ---- | 0000 ---- |
| 397h | IOCCP | IOCCP7(3) | IOCCP6(3) | IOCCP5 | IOCCP4 | IOCCP3 | IOCCP2 | IOCCP1 | IOCCP0 | 0000 0000 | 0000 0000 |
| 398h | IOCCN | IOCCN7(3) | IOCCN6(3) | IOCCN5 | IOCCN4 | IOCCN3 | IOCCN2 | IOCCN1 | IOCCN0 | 0000 0000 | 0000 0000 |
| 399h | IOCCF | IOCCF7(3) | IOCCF6(3) | IOCCF5 | IOCCF4 | IOCCF3 | IOCCF2 | IOCCF1 | IOCCF0 | 0000 0000 | 0000 0000 |
| 39Ah — 39Fh | — | Unimplemented | | | | | | | | — | — |
| **Bank 8** | | | | | | | | | | | |
| 40Ch — 41Fh | — | Unimplemented | | | | | | | | — | — |

**Legend:** x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, read as '0', r = reserved.
Shaded locations are unimplemented, read as '0'.

**Note 1:** Unimplemented, read as '1'.
**2:** PIC16(L)F1703 only.
**3:** PIC16(L)F1707 only.
**4:** Unimplemented on PIC16LF1703/7.


© 2013-2015 Microchip Technology Inc.

**TABLE 3-10: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)**

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR, BOR | Value on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Bank 9** | | | | | | | | | | | |
| 48Ch to 49Fh | — | Unimplemented | | | | | | | | — | — |
| **Bank 10** | | | | | | | | | | | |
| 50Ch — 510h | — | Unimplemented | | | | | | | | — | — |
| 511h | OPA1CON | OPA1EN | OPA1SP | — | OPA1UG | — | — | OPA1PCH<1:0> | | 00-0 --00 | 00-0 --00 |
| 512h — 514h | — | Unimplemented | | | | | | | | — | — |
| 515h | OPA2CON | OPA2EN | OPA2SP | — | OPA2UG | — | — | OPA2PCH<1:0> | | 00-0 --00 | 00-0 --00 |
| 516h — 51Fh | — | Unimplemented | | | | | | | | — | — |
| **Bank 11-27** | | | | | | | | | | | |
| 58Ch/ 59Fh — D8Ch/ D9Fh | — | Unimplemented | | | | | | | | — | — |

**Legend:** x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, read as '0', r = reserved.
Shaded locations are unimplemented, read as '0'.

**Note 1:** Unimplemented, read as '1'.
**2:** PIC16(L)F1703 only.
**3:** PIC16(L)F1707 only.
**4:** Unimplemented on PIC16LF1703/7.

© 2013-2015 Microchip Technology Inc.

**TABLE 3-10: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)**

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR, BOR | Value on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Bank 28** | | | | | | | | | | | |
| E0Ch — E0Eh | — | Unimplemented | | | | | | | | — | — |
| E0Fh | PPSLOCK | — | — | — | — | — | — | — | PPSLOCKED | ---- ---0 | ---- ---0 |
| E10h | INTPPS | — | — | — | INTPPS<4:0> | | | | | ---0 0010 | ---u uuuu |
| E11h | T0CKIPPS | — | — | — | T0CKIPPS<4:0> | | | | | ---0 0010 | ---u uuuu |
| E12h | T1CKIPPS | — | — | — | T1CKIPPS<4:0> | | | | | ---0 0101 | ---u uuuu |
| E13h | T1GPPS | — | — | — | T1GPPS<4:0> | | | | | ---0 0100 | ---u uuuu |
| E14h | CCP1PPS | — | — | — | CCP1PPS<4:0> | | | | | ---1 0101 | ---u uuuu |
| E15h | CCP2PPS | — | — | — | CCP2PPS<4:0> | | | | | ---1 0011 | ---u uuuu |
| E16h to E1Fh | — | Unimplemented | | | | | | | | — | — |
| E20h | SSPCLKPPS | — | — | — | SSPCLKPPS<4:0> | | | | | ---1 0000(3) | ---u uuuu |
| | | — | — | — | SSPCLKPPS<4:0> | | | | | ---0 1110(4) | ---u uuuu |
| E21h | SSPDATPPS | — | — | — | SSPDATPPS<4:0> | | | | | ---1 0001(3) | ---u uuuu |
| | | — | — | — | SSPDATPPS<4:0> | | | | | ---0 1100(4) | ---u uuuu |
| E22h | SSPSSPPS | — | — | — | SSPSSPPS<4:0> | | | | | ---1 0011(3) | ---u uuuu |
| | | — | — | — | SSPSSPPS<4:0> | | | | | ---1 0110(4) | ---u uuuu |
| E23h to E6Fh | — | Unimplemented | | | | | | | | — | — |

**Legend:** x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, read as '0', r = reserved.
Shaded locations are unimplemented, read as '0'.

**Note 1:** Unimplemented, read as '1'.
**2:** PIC16(L)F1703 only.
**3:** PIC16(L)F1707 only.
**4:** Unimplemented on PIC16LF1703/7.


© 2013-2015 Microchip Technology Inc.

**TABLE 3-10: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)**

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR, BOR | Value on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Bank 29** | | | | | | | | | | | |
| E8Ch — E8Fh | — | Unimplemented | | | | | | | | — | — |
| E90h | RA0PPS | — | — | — | RA0PPS<4:0> | | | | | ---0 0000 | ---u uuuu |
| E91h | RA1PPS | — | — | — | RA1PPS<4:0> | | | | | ---0 0000 | ---u uuuu |
| E92h | RA2PPS | — | — | — | RA2PPS<4:0> | | | | | ---0 0000 | ---u uuuu |
| E93h | — | Unimplemented | | | | | | | | — | — |
| E94h | RA4PPS | — | — | — | RA4PPS<4:0> | | | | | ---0 0000 | ---u uuuu |
| E95h | RA5PPS | — | — | — | RA5PPS<4:0> | | | | | ---0 0000 | ---u uuuu |
| E96h to E9Fh | — | Unimplemented | | | | | | | | — | — |
| EA0h | RC0PPS | — | — | — | RC0PPS<4:0> | | | | | ---0 0000 | ---u uuuu |
| EA1h | RC1PPS | — | — | — | RC1PPS<4:0> | | | | | ---0 0000 | ---u uuuu |
| EA2h | RC2PPS | — | — | — | RC2PPS<4:0> | | | | | ---0 0000 | ---u uuuu |
| EA3h | RC3PPS | — | — | — | RC3PPS<4:0> | | | | | ---0 0000 | ---u uuuu |
| EA4h | RC4PPS | — | — | — | RC4PPS<4:0> | | | | | ---0 0000 | ---u uuuu |
| EA5h | RC5PPS | — | — | — | RC5PPS<4:0> | | | | | ---0 0000 | ---u uuuu |
| EA6h — EEFh | — | Unimplemented | | | | | | | | — | — |
| **Bank 30** | | | | | | | | | | | |
| F0Ch — F6Fh | — | Unimplemented | | | | | | | | — | — |

**Legend:** x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, read as '0', r = reserved.
Shaded locations are unimplemented, read as '0'.

**Note 1:** Unimplemented, read as '1'.
**2:** PIC16(L)F1703 only.
**3:** PIC16(L)F1707 only.
**4:** Unimplemented on PIC16LF1703/7.

© 2013-2015 Microchip Technology Inc.

**TABLE 3-10: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)**

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR, BOR | Value on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Bank 31** | | | | | | | | | | | |
| F8Eh to FE3h | — | Unimplemented | | | | | | | | — | — |
| FE4h | STATUS_ SHAD | — | — | — | — | — | Z | DC | C | ---- -xxx | ---- -xxx |
| FE5h | WREG_ SHAD | Working Register Shadow | | | | | | | | xxxx xxxx | uuuu uuuu |
| FE6h | BSR_ SHAD | — | — | — | Bank Select Register Shadow | | | | | ---x xxxx | ---u uuuu |
| FE7h | PCLATH_ SHAD | — | Program Counter Latch High Register Shadow | | | | | | | -xxx xxxx | uuuu uuuu |
| FE8h | FSR0L_ SHAD | Indirect Data Memory Address 0 Low Pointer Shadow | | | | | | | | xxxx xxxx | uuuu uuuu |
| FE9h | FSR0H_ SHAD | Indirect Data Memory Address 0 High Pointer Shadow | | | | | | | | xxxx xxxx | uuuu uuuu |
| FEAh | FSR1L_ SHAD | Indirect Data Memory Address 1 Low Pointer Shadow | | | | | | | | xxxx xxxx | uuuu uuuu |
| FEBh | FSR1H_ SHAD | Indirect Data Memory Address 1 High Pointer Shadow | | | | | | | | xxxx xxxx | uuuu uuuu |
| FECh | — | Unimplemented | | | | | | | | — | — |
| FEDh | STKPTR | — | — | — | Current Stack Pointer | | | | | ---x xxxx | ---x xxxx |
| FEEh | TOSL | Top of Stack Low byte | | | | | | | | xxxx xxxx | uuuu uuuu |
| FEFh | TOSH | — | Top of Stack High byte | | | | | | | -xxx xxxx | -uuu uuuu |

**Legend:** x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, read as '0', r = reserved.
Shaded locations are unimplemented, read as '0'.

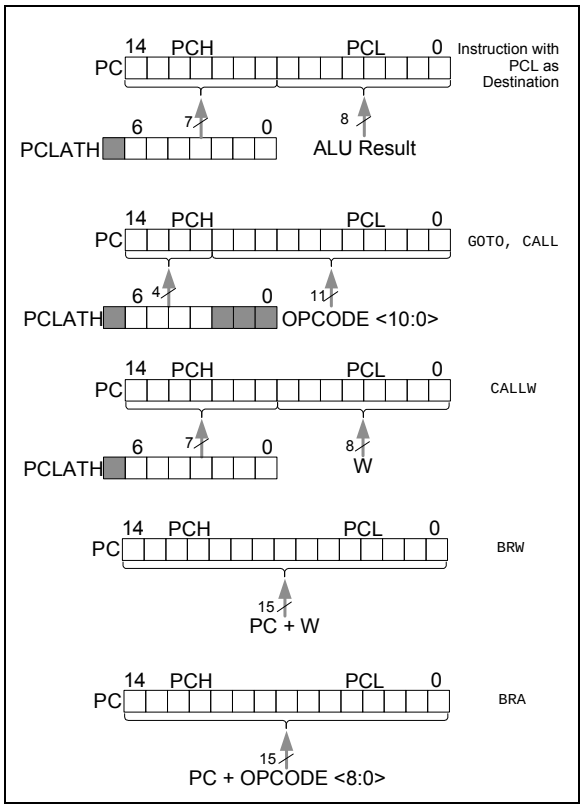
**Note 1:** Unimplemented, read as '1'.
**2:** PIC16(L)F1703 only.
**3:** PIC16(L)F1707 only.
**4:** Unimplemented on PIC16LF1703/7.


© 2013-2015 Microchip Technology Inc.

## 3.5 PCL and PCLATH

The Program Counter (PC) is 15 bits wide. The low byte comes from the PCL register, which is a readable and writable register. The high byte (PC<14:8>) is not directly readable or writable and comes from PCLATH. On any Reset, the PC is cleared. Figure 3-3 shows the five situations for the loading of the PC.

**FIGURE 3-3: LOADING OF PC IN DIFFERENT SITUATIONS**




### 3.5.1 MODIFYING PCL

Executing any instruction with the PCL register as the destination simultaneously causes the Program Counter PC<14:8> bits (PCH) to be replaced by the contents of the PCLATH register. This allows the entire contents of the program counter to be changed by writing the desired upper seven bits to the PCLATH register. When the lower eight bits are written to the PCL register, all 15 bits of the program counter will change to the values contained in the PCLATH register and those being written to the PCL register.

### 3.5.2 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 Application Note AN556, *"Implementing a Table Read"* (DS00556).

### 3.5.3 COMPUTED FUNCTION CALLS

A computed function CALL allows programs to maintain tables of functions and provide another way to execute state machines or look-up tables. When performing a table read using a computed function CALL, care should be exercised if the table location crosses a PCL memory boundary (each 256-byte block).

If using the CALL instruction, the PCH<2:0> and PCL registers are loaded with the operand of the CALL instruction. PCH<6:3> is loaded with PCLATH<6:3>.

The CALLW instruction enables computed calls by combining PCLATH and W to form the destination address. A computed CALLW is accomplished by loading the W register with the desired address and executing CALLW. The PCL register is loaded with the value of W and PCH is loaded with PCLATH.

### 3.5.4 BRANCHING

The branching instructions add an offset to the PC. This allows relocatable code and code that crosses page boundaries. There are two forms of branching, BRW and BRA. The PC will have incremented to fetch the next instruction in both cases. When using either branching instruction, a PCL memory boundary may be crossed.

If using BRW, load the W register with the desired unsigned address and execute BRW. The entire PC will be loaded with the address PC + 1 + W.

If using BRA, the entire PC will be loaded with PC + 1 +, the signed value of the operand of the BRA instruction.

## 3.6 Stack

All devices have a 16-level x 15-bit wide hardware stack (refer to Figure 3-1). The stack space is not part of either program or data space. The PC is PUSHed onto the stack when CALL or CALLW instructions are 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 if the STVREN bit is programmed to '0' (Configuration Words). This means that after the stack has been PUSHed sixteen times, the seventeenth PUSH overwrites the value that was stored from the first PUSH. The eighteenth PUSH overwrites the second PUSH (and so on). The STKOVF and STKUNF flag bits will be set on an Overflow/Underflow, regardless of whether the Reset is enabled.

| **Note:** | There are no instructions/mnemonics called PUSH or POP. These are actions that occur from the execution of the CALL, CALLW, RETURN, RETLW and RETFIE instructions or the vectoring to an interrupt address. |
|---|---|

### 3.6.1 ACCESSING THE STACK

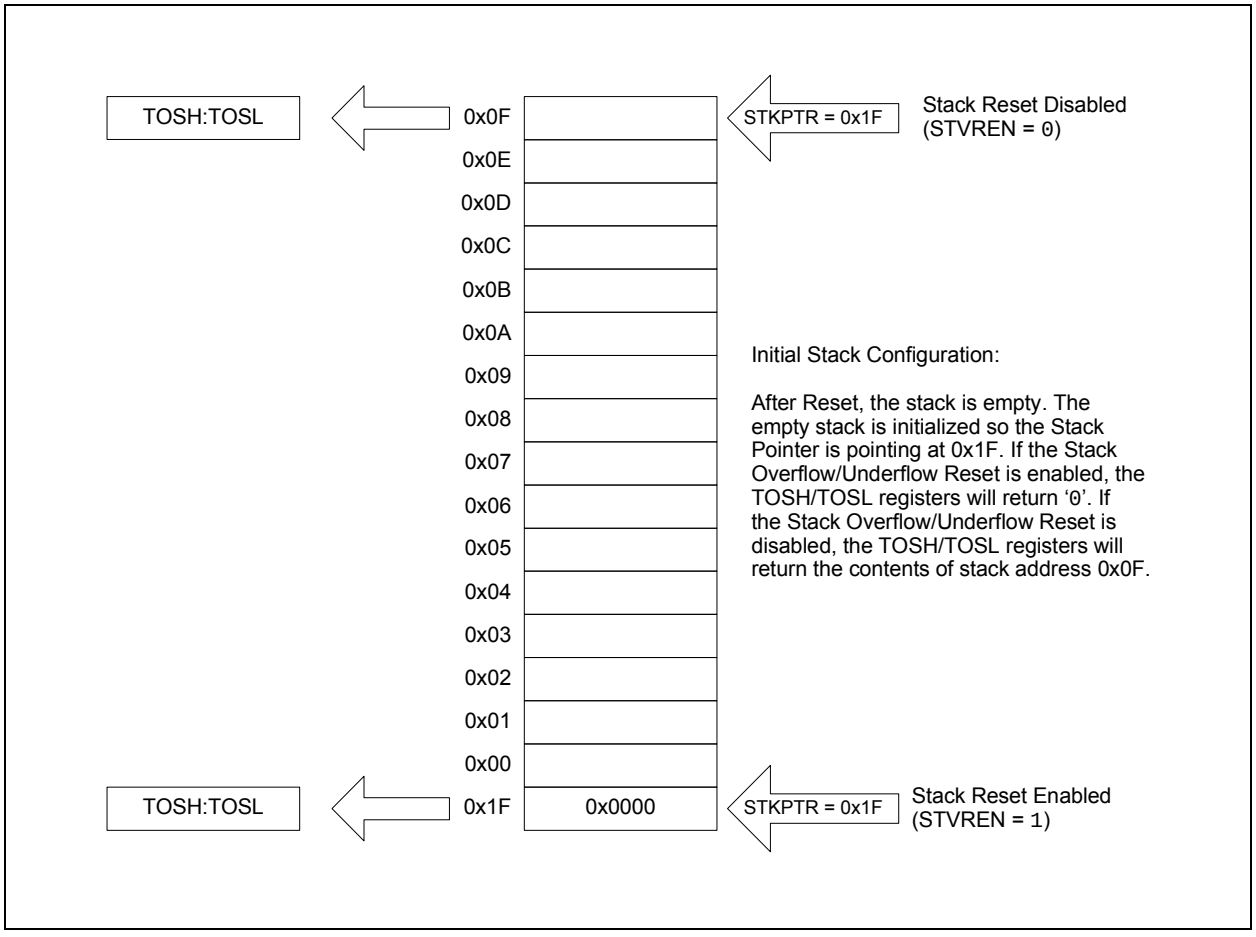
The stack is available through the TOSH, TOSL and STKPTR registers. STKPTR is the current value of the Stack Pointer. TOSH:TOSL register pair points to the TOP of the stack. Both registers are read/writable. TOS is split into TOSH and TOSL due to the 15-bit size of the PC. To access the stack, adjust the value of STKPTR, which will position TOSH:TOSL, then read/write to TOSH:TOSL. STKPTR is five bits to allow detection of overflow and underflow.

| **Note:** | Care should be taken when modifying the STKPTR while interrupts are enabled. |
|---|---|

During normal program operation, CALL, CALLW and Interrupts will increment STKPTR while RETLW, RETURN, and RETFIE will decrement STKPTR. At any time, STKPTR can be inspected to see how much stack is left. The STKPTR always points at the currently used place on the stack. Therefore, a CALL or CALLW will increment the STKPTR and then write the PC, and a return will unload the PC and then decrement the STKPTR.

Reference Figure 3-4 through Figure 3-7 for examples of accessing the stack.

**FIGURE 3-4: ACCESSING THE STACK EXAMPLE 1**





© 2013-2015 Microchip Technology Inc.

**FIGURE 3-5: ACCESSING THE STACK EXAMPLE 2**

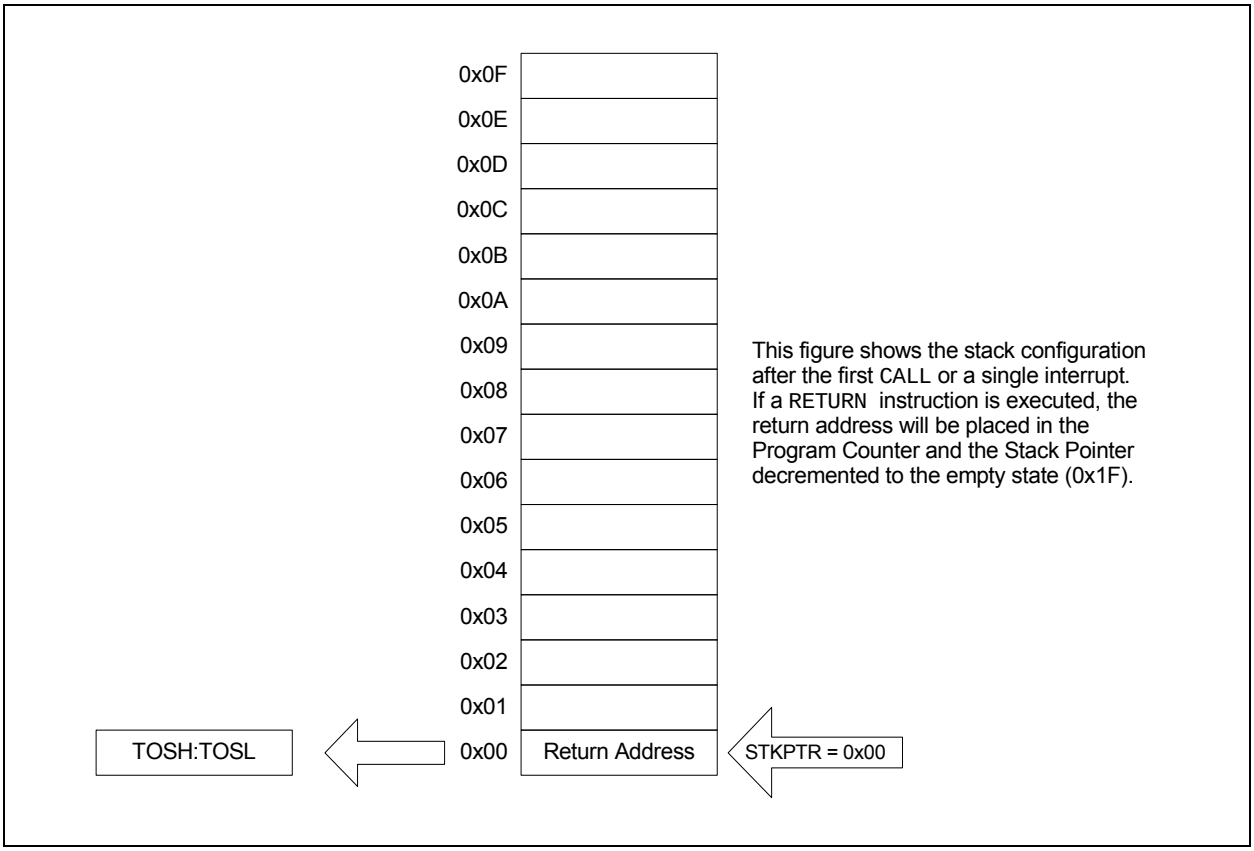



**FIGURE 3-6: ACCESSING THE STACK EXAMPLE 3**

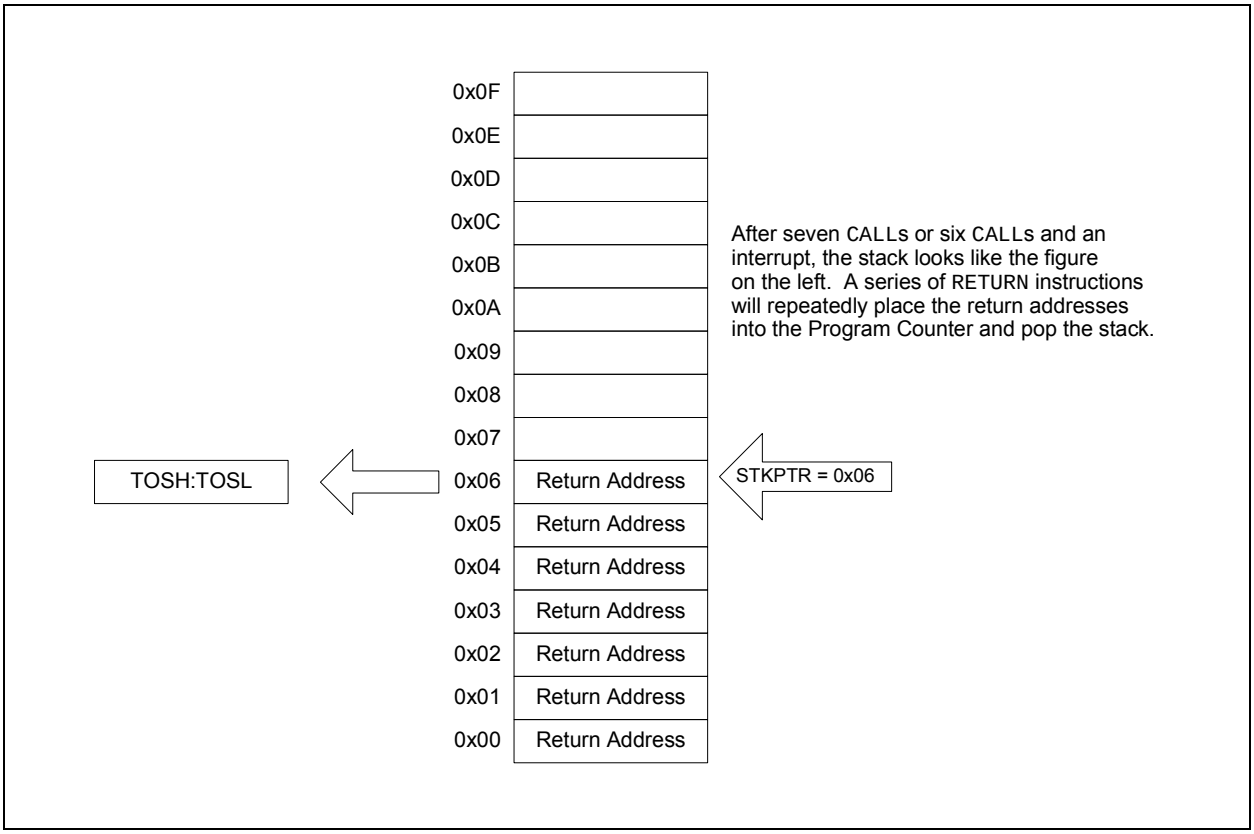
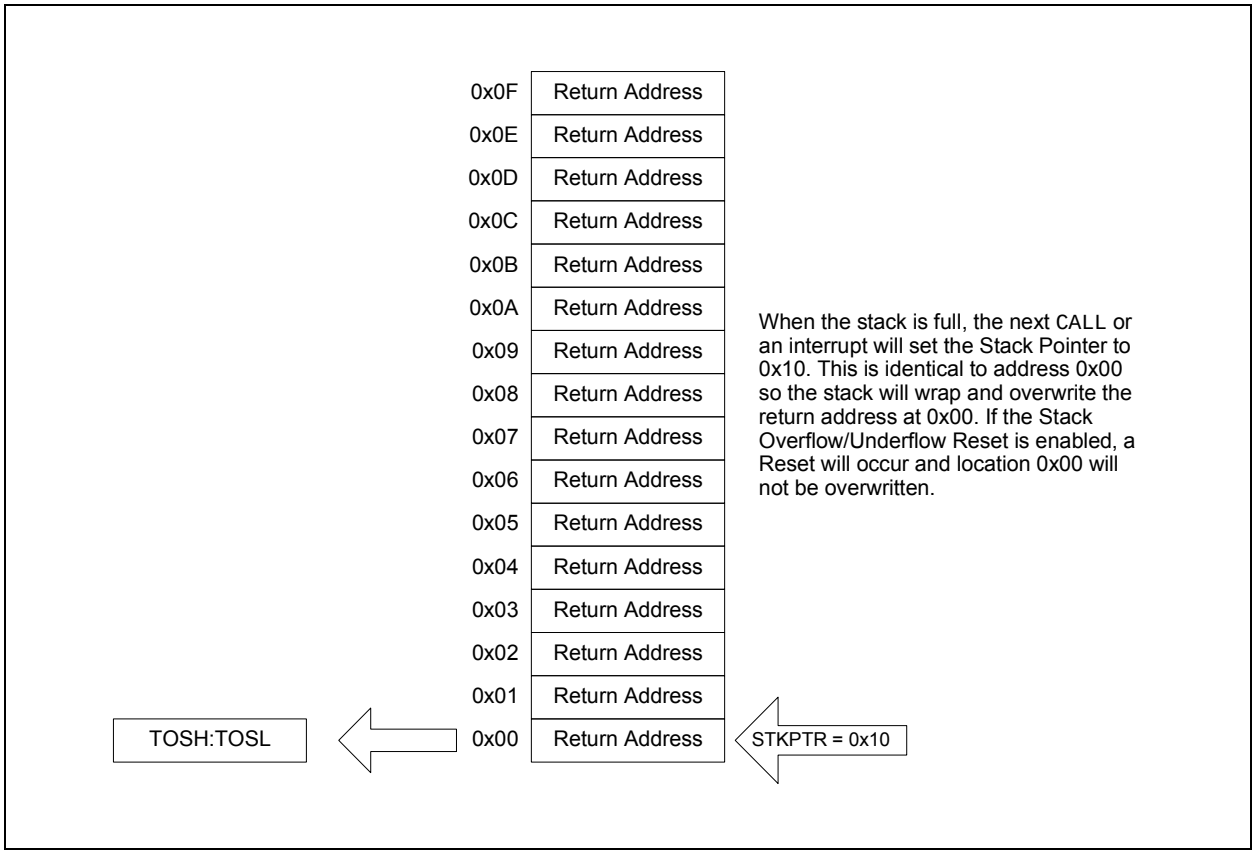


**FIGURE 3-7: ACCESSING THE STACK EXAMPLE 4**




### 3.6.2 OVERFLOW/UNDERFLOW RESET

If the STVREN bit in Configuration Words is programmed to '1', the device will be reset if the stack is PUSHed beyond the sixteenth level or POPed beyond the first level, setting the appropriate bits (STKOVF or STKUNF, respectively) in the PCON register.

## 3.7 Indirect Addressing

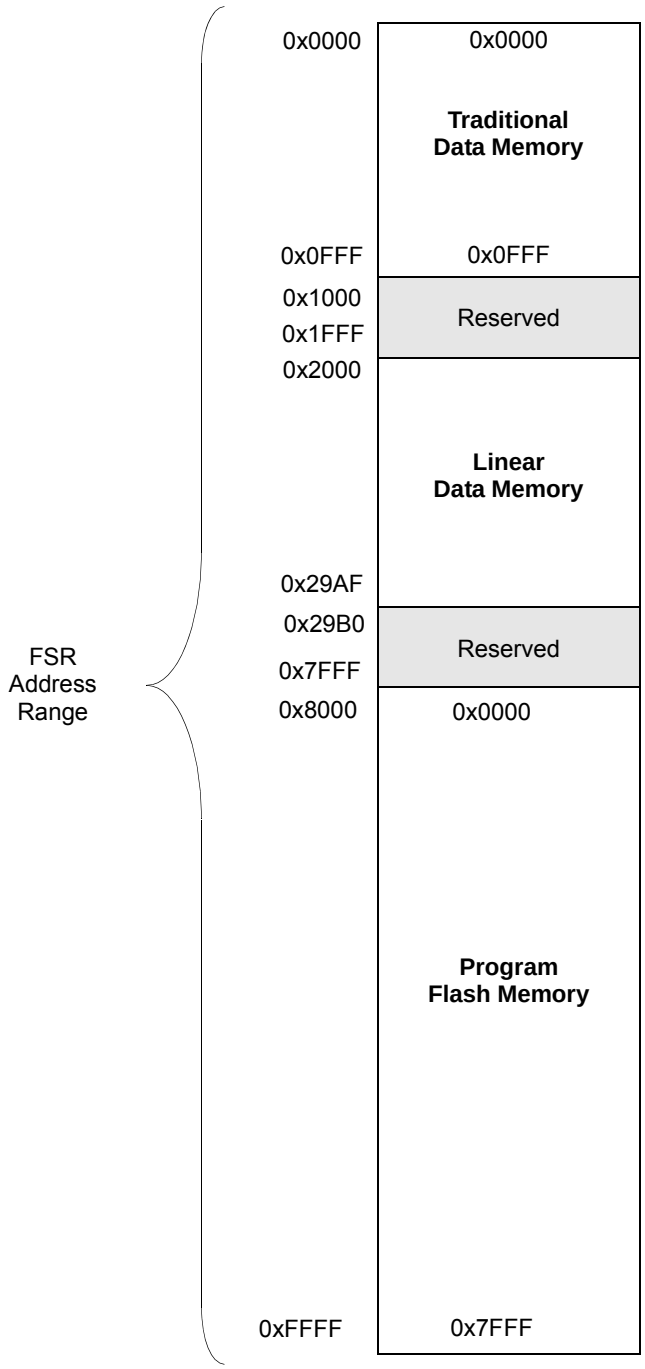
The INDFn registers are not physical registers. Any instruction that accesses an INDFn register actually accesses the register at the address specified by the File Select Registers (FSR). If the FSRn address specifies one of the two INDFn registers, the read will return '0' and the write will not occur (though Status bits may be affected). The FSRn register value is created by the pair FSRnH and FSRnL.

The FSR registers form a 16-bit address that allows an addressing space with 65536 locations. These locations are divided into three memory regions:

- Traditional Data Memory
- Linear Data Memory
- Program Flash Memory


© 2013-2015 Microchip Technology Inc.

**FIGURE 3-8: INDIRECT ADDRESSING**

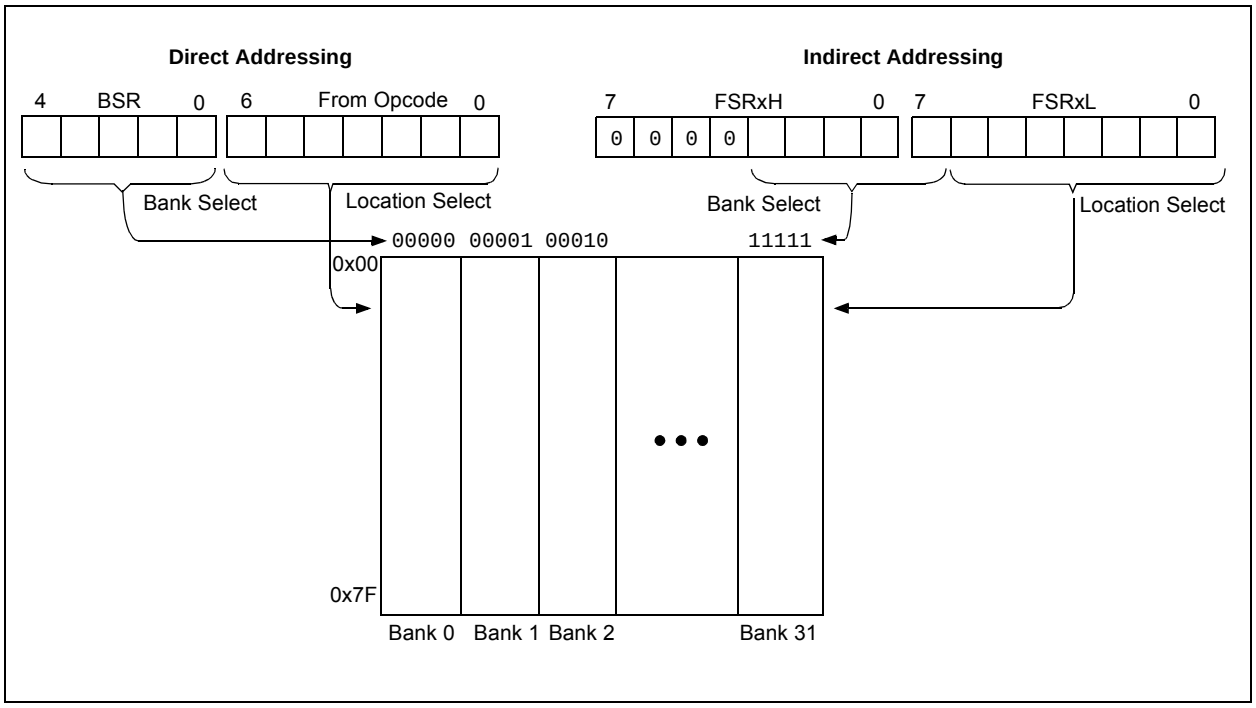



**Note:** Not all memory regions are completely implemented. Consult device memory tables for memory limits.

### 3.7.1 TRADITIONAL DATA MEMORY

The traditional data memory is a region from FSR address 0x000 to FSR address 0xFFF. The addresses correspond to the absolute addresses of all SFR, GPR and common registers.

**FIGURE 3-9: TRADITIONAL DATA MEMORY MAP**





© 2013-2015 Microchip Technology Inc.

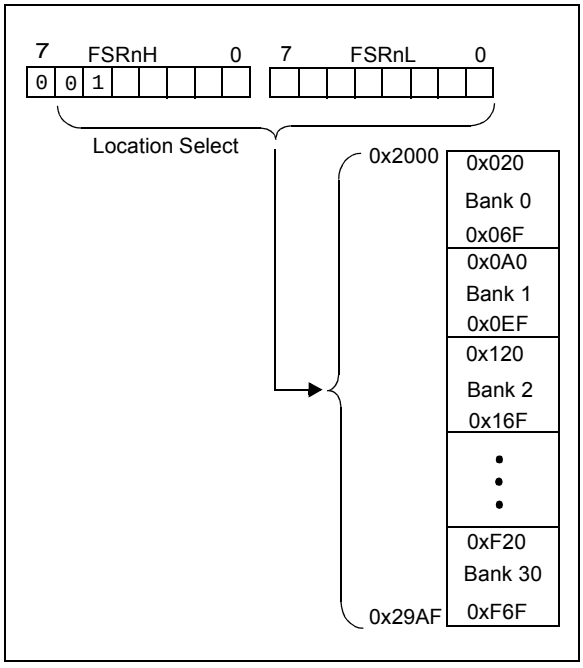
### 3.7.2 LINEAR DATA MEMORY

The linear data memory is the region from FSR address 0x2000 to FSR address 0x29AF. This region is a virtual region that points back to the 80-byte blocks of GPR memory in all the banks.

Unimplemented memory reads as 0x00. Use of the linear data memory region allows buffers to be larger than 80 bytes because incrementing the FSR beyond one bank will go directly to the GPR memory of the next bank.

The 16 bytes of common memory are not included in the linear data memory region.

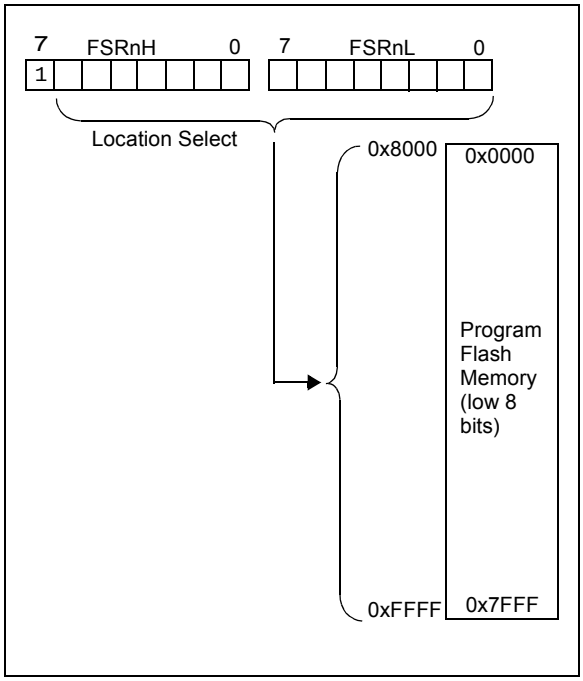
**FIGURE 3-10: LINEAR DATA MEMORY MAP**




### 3.7.3 PROGRAM FLASH MEMORY

To make constant data access easier, the entire program Flash memory is mapped to the upper half of the FSR address space. When the MSB of FSRnH is set, the lower 15 bits are the address in program memory which will be accessed through INDF. Only the lower eight bits of each memory location is accessible via INDF. Writing to the program Flash memory cannot be accomplished via the FSR/INDF interface. All instructions that access program Flash memory via the FSR/INDF interface will require one additional instruction cycle to complete.

**FIGURE 3-11: PROGRAM FLASH MEMORY MAP**



## 4.0 DEVICE CONFIGURATION

Device configuration consists of Configuration Words, Code Protection and Device ID.

### 4.1 Configuration Words

There are several Configuration Word bits that allow different oscillator and memory protection options. These are implemented as Configuration Word 1 at 8007h and Configuration Word 2 at 8008h.

| | |
|---|---|
| **Note:** | The $\overline{\text{DEBUG}}$ bit in Configuration Words is managed automatically by device development tools including debuggers and programmers. For normal device operation, this bit should be maintained as a '1'. |


© 2013-2015 Microchip Technology Inc.

## 4.2 Register Definitions: Configuration Words

**REGISTER 4-1: CONFIG1: CONFIGURATION WORD 1**

| U-1 | U-1 | R/P-1 | R/P-1 | R/P-1 | U-1 |
|---|---|---|---|---|---|
| — | — | $\overline{\text{CLKOUTEN}}$ | BOREN<1:0> | | — |
| bit 13 | | | | | bit 8 |

| R/P-1 | R/P-1 | R/P-1 | R/P-1 | R/P-1 | R/P-1 | R/P-1 | R/P-1 |
|---|---|---|---|---|---|---|---|
| $\overline{\text{CP}}$(1) | MCLRE | $\overline{\text{PWRTE}}$ | WDTE<1:0> | | — | FOSC<1:0> | |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | P = Programmable bit | U = Unimplemented bit, read as '1' |
| '0' = Bit is cleared | '1' = Bit is set | -n = Value when blank or after Bulk Erase |

bit 13-12 **Unimplemented:** Read as '1'

bit 11 **$\overline{\text{CLKOUTEN}}$:** Clock Out Enable bit
- 1 = CLKOUT function is disabled. I/O function on the CLKOUT pin.
- 0 = CLKOUT function is enabled on the CLKOUT pin

bit 10-9 **BOREN<1:0>:** Brown-out Reset Enable bits
- 11 = BOR enabled
- 10 = BOR enabled during operation and disabled in Sleep
- 01 = BOR controlled by SBOREN bit of the BORCON register
- 00 = BOR disabled

bit 8 **Unimplemented:** Read as '1'

bit 7 **$\overline{\text{CP}}$:** Code Protection bit(1)
- 1 = Program memory code protection is disabled
- 0 = Program memory code protection is enabled

bit 6 **MCLRE:** $\overline{\text{MCLR}}$/VPP Pin Function Select bit

If LVP bit = 1:

This bit is ignored.

If LVP bit = 0:
- 1 = $\overline{\text{MCLR}}$/VPP pin function is $\overline{\text{MCLR}}$; Weak pull-up enabled.
- 0 = $\overline{\text{MCLR}}$/VPP pin function is digital input; $\overline{\text{MCLR}}$ internally disabled; Weak pull-up under control of WPUE3 bit

bit 5 **$\overline{\text{PWRTE}}$:** Power-up Timer Enable bit
- 1 = PWRT disabled
- 0 = PWRT enabled

bit 4-3 **WDTE<1:0>:** Watchdog Timer Enable bit
- 11 = WDT enabled
- 10 = WDT enabled while running and disabled in Sleep
- 01 = WDT controlled by the SWDTEN bit in the WDTCON register
- 00 = WDT disabled

bit 2 **Unimplemented:** Read as '1'

bit 1-0 **FOSC<1:0>:** Oscillator Selection bits
- 11 = ECH: External Clock, High-Power mode (4-20 MHz): device clock supplied to CLKIN pin
- 10 = ECM: External Clock, Medium-Power mode (0.5-4 MHz): device clock supplied to CLKIN pin
- 01 = ECL: External Clock, Low-Power mode (0-0.5 MHz): device clock supplied to CLKIN pin
- 00 = INTOSC oscillator: I/O function on CLKIN pin

**Note 1:** The entire Flash program memory will be erased when the code protection is turned off during an erase. When a Bulk Erase Program Memory Command is executed, the entire program Flash memory and configuration memory will be erased.

**REGISTER 4-2: CONFIG2: CONFIGURATION WORD 2**

| R/P-1 | R/P-1 | R/P-1 | R/P-1 | R/P-1 | R/P-1 |
|---|---|---|---|---|---|
| LVP(1) | $\overline{\text{DEBUG}}$(2) | $\overline{\text{LPBOR}}$ | BORV(3) | STVREN | PLLEN |
| bit 13 | | | | | bit 8 |

| R/P-1 | U-1 | U-1 | U-1 | U-1 | R/P-1 | R/P-1 | R/P-1 |
|---|---|---|---|---|---|---|---|
| ZCDDIS | — | — | — | — | PPS1WAY | WRT<1:0> | |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | P = Programmable bit | U = Unimplemented bit, read as '1' |
| '0' = Bit is cleared | '1' = Bit is set | -n = Value when blank or after Bulk Erase |

bit 13 **LVP:** Low-Voltage Programming Enable bit(1)
1 = Low-voltage programming enabled
0 = High-voltage on $\overline{\text{MCLR}}$ must be used for programming

bit 12 **$\overline{\text{DEBUG}}$:** In-Circuit Debugger Mode bit(2)
1 = In-Circuit Debugger disabled, ICSPCLK and ICSPDAT are general purpose I/O pins
0 = In-Circuit Debugger enabled, ICSPCLK and ICSPDAT are dedicated to the debugger

bit 11 **$\overline{\text{LPBOR}}$:** Low-Power BOR Enable bit
1 = Low-Power Brown-out Reset is disabled
0 = Low-Power Brown-out Reset is enabled

bit 10 **BORV:** Brown-out Reset Voltage Selection bit(3)
1 = Brown-out Reset voltage ($V_{BOR}$), low trip point selected
0 = Brown-out Reset voltage ($V_{BOR}$), high trip point selected

bit 9 **STVREN:** Stack Overflow/Underflow Reset Enable bit
1 = Stack Overflow or Underflow will cause a Reset
0 = Stack Overflow or Underflow will not cause a Reset

bit 8 **PLLEN:** PLL Enable bit
1 = 4xPLL enabled
0 = 4xPLL disabled

bit 7 **ZCDDIS:** ZCD Disable bit
1 = ZCD disabled. ZCD can be enabled by setting the ZCDSEN bit of ZCDCON
0 = ZCD always enabled

bit 6-3 **Unimplemented:** Read as '1'

bit 2 **PPS1WAY**: PPSLOCK Bit One-Way Set Enable bit
1 = The PPSLOCK bit can only be set once after an unlocking sequence is executed; once PPSLOCK is set, all future changes to PPS registers are prevented
0 = The PPSLOCK bit can be set and cleared as needed (provided an unlocking sequence is executed)

bit 1-0 **WRT<1:0>:** Flash Memory Self-Write Protection bits
4 kW Flash memory
11 = Write protection off
10 = 000h to 1FFh write protected, 200h to FFFh may be modified by PMCON control
01 = 000h to 7FFh write protected, 800h to FFFh may be modified by PMCON control
00 = 000h to FFFh write protected, no addresses may be modified by PMCON control

**Note 1:** The LVP bit cannot be programmed to '0' when Programming mode is entered via LVP.
**2:** The $\overline{\text{DEBUG}}$ bit in Configuration Words is managed automatically by device development tools including debuggers and programmers. For normal device operation, this bit should be maintained as a '1'.
**3:** See $V_{BOR}$ parameter for specific trip point voltages.


© 2013-2015 Microchip Technology Inc.

## 4.3 Code Protection

Code protection allows the device to be protected from unauthorized access. Program memory protection is controlled independently. Internal access to the program memory is unaffected by any code protection setting.

### 4.3.1 PROGRAM MEMORY PROTECTION

The entire program memory space is protected from external reads and writes by the $\overline{CP}$ bit in Configuration Words. When $\overline{CP}$ = 0, external reads and writes of program memory are inhibited and a read will return all '0's. The CPU can continue to read program memory, regardless of the protection bit settings. Writing the program memory is dependent upon the write protection setting. See **Section 4.4 "Write Protection"** for more information.

## 4.4 Write Protection

Write protection allows the device to be protected from unintended self-writes. Applications, such as boot loader software, can be protected while allowing other regions of the program memory to be modified.

The WRT<1:0> bits in Configuration Words define the size of the program memory block that is protected.

## 4.5 User ID

Four memory locations (8000h-8003h) are designated as ID locations where the user can store checksum or other code identification numbers. These locations are readable and writable during normal execution. See **Section 10.4 "User ID, Device ID and Configuration Word Access"** for more information on accessing these memory locations. For more information on checksum calculation, see the *"PIC16(L)F170X Memory Programming Specification"* (DS41683).

## 4.6 Device ID and Revision ID

The 14-bit device ID word is located at 8006h and the 14-bit revision ID is located at 8005h. These locations are read-only and cannot be erased or modified. See **Section 10.4 "User ID, Device ID and Configuration Word Access"** for more information on accessing these memory locations.

Development tools, such as device programmers and debuggers, may be used to read the Device ID and Revision ID.

## 4.7 Register Definitions: Device and Revision

**REGISTER 4-3: DEVID: DEVICE ID REGISTER**

| R | R | R | R | R | R |
|---|---|---|---|---|---|
| DEV<13:8> | | | | | |
| bit 13 | | | | | bit 8 |

| R | R | R | R | R | R | R | R |
|---|---|---|---|---|---|---|---|
| DEV<7:0> | | | | | | | |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | | |
| '1' = Bit is set | '0' = Bit is cleared | |

bit 13-0 **DEV<13:0>:** Device ID bits

| Device | DEVID<13:0> Values |
|---|---|
| PIC16F1703 | 11 0000 0100 0011 (3043h) |
| PIC16LF1703 | 11 0000 0100 0101 (3045h) |
| PIC16F1707 | 11 0000 0100 0010 (3042h) |
| PIC16LF1707 | 11 0000 0100 0100 (3044h) |

**REGISTER 4-4: REVID: REVISION ID REGISTER**

| R | R | R | R | R | R |
|---|---|---|---|---|---|
| REV<13:8> | | | | | |
| bit 13 | | | | | bit 8 |

| R | R | R | R | R | R | R | R |
|---|---|---|---|---|---|---|---|
| REV<7:0> | | | | | | | |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | | |
| '1' = Bit is set | '0' = Bit is cleared | |

bit 13-0 **REV<13:0>:** Revision ID bits


© 2013-2015 Microchip Technology Inc.

## 5.0 RESETS

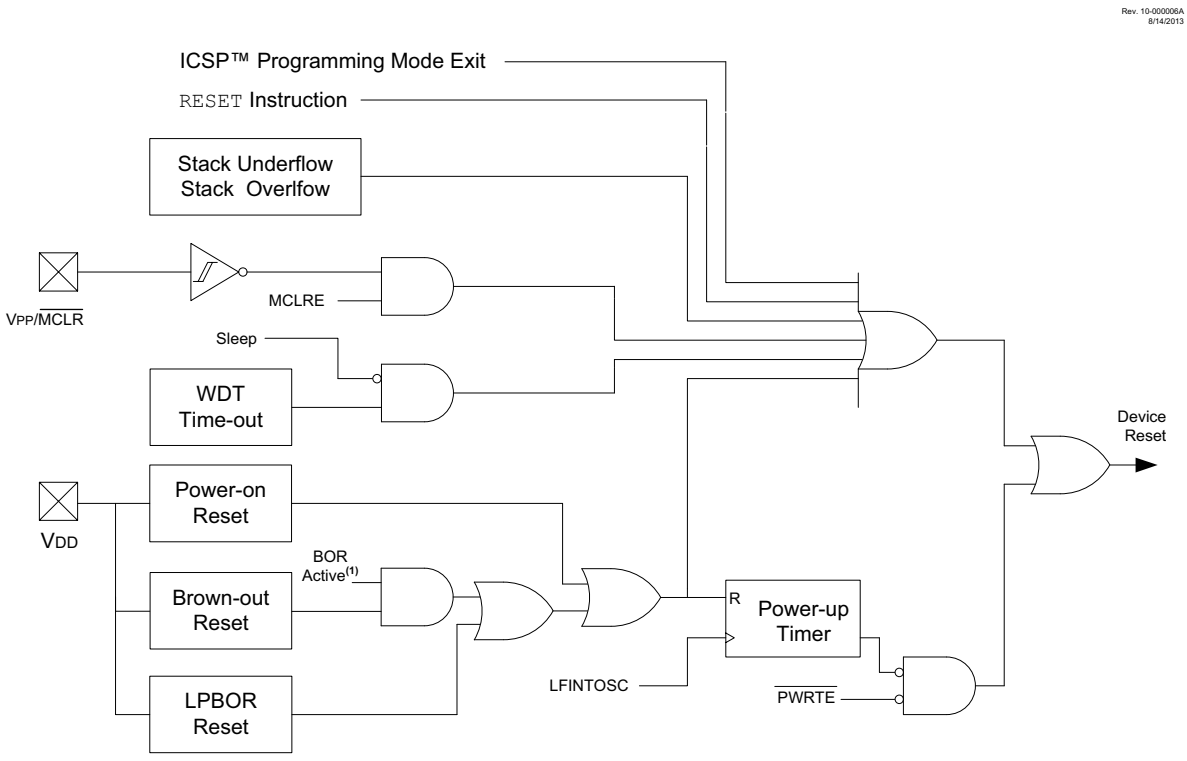
There are multiple ways to reset this device:

- Power-On Reset (POR)
- Brown-Out Reset (BOR)
- Low-Power Brown-Out Reset (LPBOR)
- $\overline{\text{MCLR}}$ Reset
- WDT Reset
- RESET instruction
- Stack Overflow
- Stack Underflow
- Programming mode exit

To allow VDD to stabilize, an optional Power-up Timer can be enabled to extend the Reset time after a BOR or POR event.

A simplified block diagram of the On-Chip Reset Circuit is shown in Figure 5-1.

**FIGURE 5-1: SIMPLIFIED BLOCK DIAGRAM OF ON-CHIP RESET CIRCUIT**




**Note 1:** See Table 5-1 for BOR active conditions.

## 5.1 Power-On Reset (POR)

The POR circuit holds the device in Reset until VDD has reached an acceptable level for minimum operation. Slow rising VDD, fast operating speeds or analog performance may require greater than minimum VDD. The PWRT, BOR or MCLR features can be used to extend the start-up period until all device operation conditions have been met.

### 5.1.1 POWER-UP TIMER (PWRT)

The Power-up Timer provides a nominal 64 ms time-out on POR or Brown-out Reset.

The device is held in Reset as long as PWRT is active. The PWRT delay allows additional time for the VDD to rise to an acceptable level. The Power-up Timer is enabled by clearing the PWRTE bit in Configuration Words.

The Power-up Timer starts after the release of the POR and BOR.

For additional information, refer to Application Note AN607, *"Power-up Trouble Shooting"* (DS00607).

## 5.2 Brown-Out Reset (BOR)

The BOR circuit holds the device in Reset when VDD reaches a selectable minimum level. Between the POR and BOR, complete voltage range coverage for execution protection can be implemented.

The Brown-out Reset module has four operating modes controlled by the BOREN<1:0> bits in Configuration Words. The four operating modes are:

- BOR is always on
- BOR is off when in Sleep
- BOR is controlled by software
- BOR is always off

Refer to Table 5-1 for more information.

The Brown-out Reset voltage level is selectable by configuring the BORV bit in Configuration Words.

A VDD noise rejection filter prevents the BOR from triggering on small events. If VDD falls below VBOR for a duration greater than parameter TBORDC, the device will reset. See Figure 5-2 for more information.

**TABLE 5-1: BOR OPERATING MODES**

| BOREN<1:0> | SBOREN | Device Mode | BOR Mode | Instruction Execution upon: Release of POR or Wake-up from Sleep |
|---|---|---|---|---|
| 11 | X | X | Active | Waits for BOR ready(1) (BORRDY = 1) |
| 10 | X | Awake | Active | Waits for BOR ready (BORRDY = 1) |
| | | Sleep | Disabled | |
| 01 | 1 | X | Active | Waits for BOR ready(1) (BORRDY = 1) |
| | 0 | X | Disabled | Begins immediately (BORRDY = x) |
| 00 | X | X | Disabled | |

**Note 1:** In these specific cases, "Release of POR" and "Wake-up from Sleep", there is no delay in start-up. The BOR ready flag, (BORRDY = 1), will be set before the CPU is ready to execute instructions because the BOR circuit is forced on by the BOREN<1:0> bits.

### 5.2.1 BOR IS ALWAYS ON

When the BOREN bits of Configuration Words are programmed to '11', the BOR is always on. The device start-up will be delayed until the BOR is ready and VDD is higher than the BOR threshold.

BOR protection is active during Sleep. The BOR does not delay wake-up from Sleep.

### 5.2.2 BOR IS OFF IN SLEEP

When the BOREN bits of Configuration Words are programmed to '10', the BOR is on, except in Sleep. The device start-up will be delayed until the BOR is ready and VDD is higher than the BOR threshold.

BOR protection is not active during Sleep. The device wake-up will be delayed until the BOR is ready.

### 5.2.3 BOR CONTROLLED BY SOFTWARE

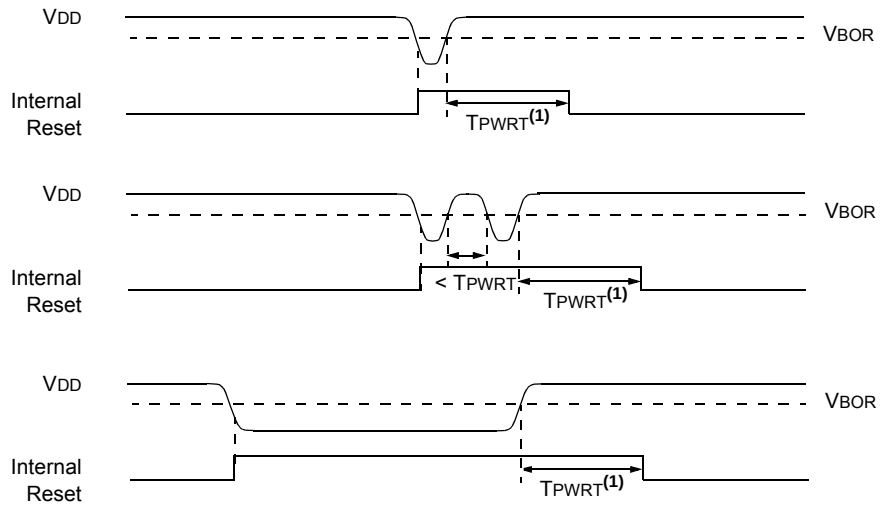
When the BOREN bits of Configuration Words are programmed to '01', the BOR is controlled by the SBOREN bit of the BORCON register. The device start-up is not delayed by the BOR ready condition or the VDD level.

BOR protection begins as soon as the BOR circuit is ready. The status of the BOR circuit is reflected in the BORRDY bit of the BORCON register.

BOR protection is unchanged by Sleep.


© 2013-2015 Microchip Technology Inc.

**FIGURE 5-2: BROWN-OUT SITUATIONS**




**Note 1:** TPWRT delay only if $\overline{\text{PWRTE}}$ bit is programmed to '0'.

## 5.3 Register Definitions: BOR Control

**REGISTER 5-1: BORCON: BROWN-OUT RESET CONTROL REGISTER**

| R/W-1/u | R/W-0/u | U-0 | U-0 | U-0 | U-0 | U-0 | R-q/u |
|---|---|---|---|---|---|---|---|
| SBOREN | BORFS(1) | — | — | — | — | — | BORRDY |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| '1' = Bit is set | '0' = Bit is cleared | q = Value depends on condition |

bit 7 **SBOREN:** Software Brown-out Reset Enable bit
If BOREN <1:0> in Configuration Words ≠ 01:
SBOREN is read/write, but has no effect on the BOR.
If BOREN <1:0> in Configuration Words = 01:
1 = BOR Enabled
0 = BOR Disabled

bit 6 **BORFS:** Brown-out Reset Fast Start bit(1)
If BOREN<1:0> = 11 (Always on) or BOREN<1:0> = 00 (Always off)
BORFS is Read/Write, but has no effect.
If BOREN <1:0> = 10 (Disabled in Sleep) or BOREN<1:0> = 01 (Under software control):
1 = Band gap is forced on always (covers sleep/wake-up/operating cases)
0 = Band gap operates normally, and may turn off

bit 5-1 **Unimplemented:** Read as '0'

bit 0 **BORRDY:** Brown-out Reset Circuit Ready Status bit
1 = The Brown-out Reset circuit is active
0 = The Brown-out Reset circuit is inactive

**Note 1:** BOREN<1:0> bits are located in Configuration Words.

## 5.4 Low-Power Brown-Out Reset (LPBOR)

The Low-Power Brown-Out Reset (LPBOR) is an essential part of the Reset subsystem. Refer to Figure 5-1 to see how the BOR interacts with other modules.

The LPBOR is used to monitor the external VDD pin. When too low of a voltage is detected, the device is held in Reset. When this occurs, a register bit (BOR) is changed to indicate that a BOR Reset has occurred. The same bit is set for both the BOR and the LPBOR. Refer to Register 5-2.

### 5.4.1 ENABLING LPBOR

The LPBOR is controlled by the LPBOR bit of Configuration Words. When the device is erased, the LPBOR module defaults to disabled.

#### 5.4.1.1 LPBOR Module Output

The output of the LPBOR module is a signal indicating whether or not a Reset is to be asserted. This signal is OR'd together with the Reset signal of the BOR module to provide the generic BOR signal, which goes to the PCON register and to the power control block.

## 5.5 MCLR

The MCLR is an optional external input that can reset the device. The MCLR function is controlled by the MCLRE bit of Configuration Words and the LVP bit of Configuration Words (Table 5-2).

**TABLE 5-2: MCLR CONFIGURATION**

| MCLRE | LVP | MCLR |
|---|---|---|
| 0 | 0 | Disabled |
| 1 | 0 | Enabled |
| x | 1 | Enabled |

### 5.5.1 MCLR ENABLED

When MCLR is enabled and the pin is held low, the device is held in Reset. The MCLR pin is connected to VDD through an internal weak pull-up.

The device has a noise filter in the MCLR Reset path. The filter will detect and ignore small pulses.

| **Note:** | A Reset does not drive the MCLR pin low. |
|---|---|

### 5.5.2 MCLR DISABLED

When MCLR is disabled, the pin functions as a general purpose input and the internal weak pull-up is under software control. See **Section 11.1 "PORTA Registers"** for more information.

## 5.6 Watchdog Timer (WDT) Reset

The Watchdog Timer generates a Reset if the firmware does not issue a CLRWDT instruction within the time-out period. The TO and PD bits in the STATUS register are changed to indicate the WDT Reset. See **Section 9.0 "Watchdog Timer (WDT)"** for more information.

## 5.7 RESET Instruction

A RESET instruction will cause a device Reset. The RI bit in the PCON register will be set to '0'. See Table 5-4 for default conditions after a RESET instruction has occurred.

## 5.8 Stack Overflow/Underflow Reset

The device can reset when the Stack Overflows or Underflows. The STKOVF or STKUNF bits of the PCON register indicate the Reset condition. These Resets are enabled by setting the STVREN bit in Configuration Words. See **Section 3.6.2 "Overflow/Underflow Reset"** for more information.

## 5.9 Programming Mode Exit

Upon exit of Programming mode, the device will behave as if a POR had just occurred.

## 5.10 Power-Up Timer

The Power-up Timer optionally delays device execution after a BOR or POR event. This timer is typically used to allow VDD to stabilize before allowing the device to start running.

The Power-up Timer is controlled by the PWRTE bit of Configuration Words.

## 5.11 Start-up Sequence

Upon the release of a POR or BOR, the following must occur before the device will begin executing:

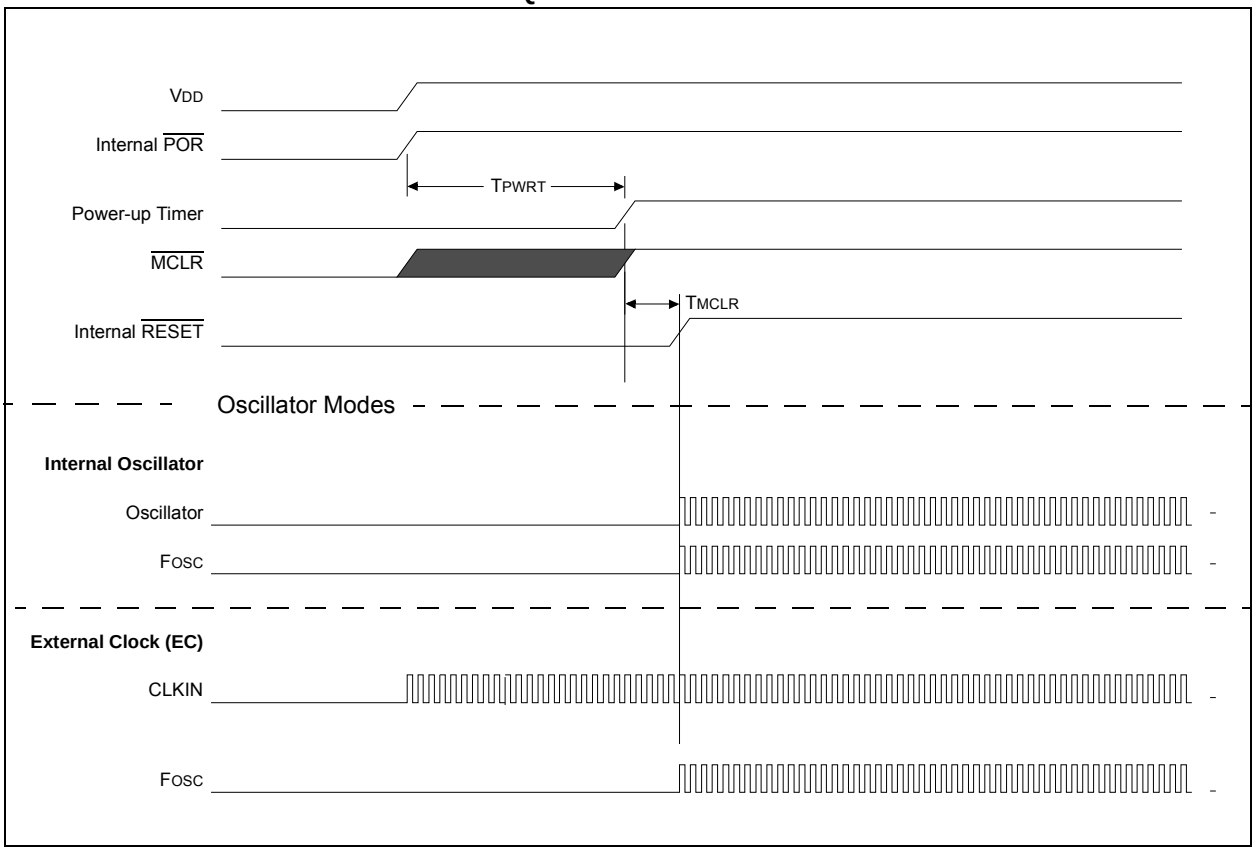
1. Power-up Timer runs to completion (if enabled).
2. Oscillator start-up timer runs to completion (if required for oscillator source).
3. MCLR must be released (if enabled).

The total time-out will vary based on oscillator configuration and Power-up Timer configuration. See **Section 6.0 "Oscillator Module (with Fail-Safe Clock Monitor)"** for more information.

The Power-up Timer and oscillator start-up timer run independently of MCLR Reset. If MCLR is kept low long enough, the Power-up Timer and oscillator start-up timer will expire. Upon bringing MCLR high, the device will begin execution after 10 FOSC cycles (see Figure 5-3). This is useful for testing purposes or to synchronize more than one device operating in parallel.


© 2013-2015 Microchip Technology Inc.

**FIGURE 5-3: RESET START-UP SEQUENCE**



## 5.12 Determining the Cause of a Reset

Upon any Reset, multiple bits in the STATUS and PCON register are updated to indicate the cause of the Reset. Table 5-3 and Table 5-4 show the Reset conditions of these registers.

**TABLE 5-3: RESET STATUS BITS AND THEIR SIGNIFICANCE**

| STKOVF | STKUNF | $\overline{\text{RWDT}}$ | $\overline{\text{RMCLR}}$ | $\overline{\text{RI}}$ | $\overline{\text{POR}}$ | $\overline{\text{BOR}}$ | $\overline{\text{TO}}$ | $\overline{\text{PD}}$ | Condition |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 1 | 0 | x | 1 | 1 | Power-on Reset |
| 0 | 0 | 1 | 1 | 1 | 0 | x | 0 | x | Illegal, $\overline{\text{TO}}$ is set on $\overline{\text{POR}}$ |
| 0 | 0 | 1 | 1 | 1 | 0 | x | x | 0 | Illegal, $\overline{\text{PD}}$ is set on $\overline{\text{POR}}$ |
| 0 | 0 | u | 1 | 1 | u | 0 | 1 | 1 | Brown-out Reset |
| u | u | 0 | u | u | u | u | 0 | u | WDT Reset |
| u | u | u | u | u | u | u | 0 | 0 | WDT Wake-up from Sleep |
| u | u | u | u | u | u | u | 1 | 0 | Interrupt Wake-up from Sleep |
| u | u | u | 0 | u | u | u | u | u | $\overline{\text{MCLR}}$ Reset during normal operation |
| u | u | u | 0 | u | u | u | 1 | 0 | $\overline{\text{MCLR}}$ Reset during Sleep |
| u | u | u | u | 0 | u | u | u | u | RESET Instruction Executed |
| 1 | u | u | u | u | u | u | u | u | Stack Overflow Reset (STVREN = 1) |
| u | 1 | u | u | u | u | u | u | u | Stack Underflow Reset (STVREN = 1) |

**TABLE 5-4: RESET CONDITION FOR SPECIAL REGISTERS**

| Condition | Program Counter | STATUS Register | PCON Register |
|---|---|---|---|
| Power-on Reset | 0000h | ---1 1000 | 00-- 110x |
| $\overline{\text{MCLR}}$ Reset during normal operation | 0000h | ---u uuuu | uu-- 0uuu |
| $\overline{\text{MCLR}}$ Reset during Sleep | 0000h | ---1 0uuu | uu-- 0uuu |
| WDT Reset | 0000h | ---0 uuuu | uu-- uuuu |
| WDT Wake-up from Sleep | PC + 1 | ---0 0uuu | uu-- uuuu |
| Brown-out Reset | 0000h | ---1 1uuu | 00-- 11u0 |
| Interrupt Wake-up from Sleep | PC + 1(1) | ---1 0uuu | uu-- uuuu |
| RESET Instruction Executed | 0000h | ---u uuuu | uu-- u0uu |
| Stack Overflow Reset (STVREN = 1) | 0000h | ---u uuuu | 1u-- uuuu |
| Stack Underflow Reset (STVREN = 1) | 0000h | ---u uuuu | u1-- uuuu |

**Legend:** u = unchanged, x = unknown, - = unimplemented bit, reads as '0'.

**Note 1:** When the wake-up is due to an interrupt and Global Enable bit (GIE) is set, the return address is pushed on the stack and PC is loaded with the interrupt vector (0004h) after execution of PC + 1.


© 2013-2015 Microchip Technology Inc.

## 5.13 Power Control (PCON) Register

The PCON register bits are shown in Register 5-2.

The Power Control (PCON) register contains flag bits to differentiate between a:

- Power-on Reset ($\overline{POR}$)
- Brown-out Reset ($\overline{BOR}$)
- Reset Instruction Reset ($\overline{RI}$)
- $\overline{MCLR}$ Reset ($\overline{RMCLR}$)
- Watchdog Timer Reset ($\overline{RWDT}$)
- Stack Underflow Reset (STKUNF)
- Stack Overflow Reset (STKOVF)

## 5.14 Register Definitions: Power Control

**REGISTER 5-2: PCON: POWER CONTROL REGISTER**

| R/W/HS-0/q | R/W/HS-0/q | U-0 | R/W/HC-1/q | R/W/HC-1/q | R/W/HC-1/q | R/W/HC-q/u | R/W/HC-q/u |
|---|---|---|---|---|---|---|---|
| STKOVF | STKUNF | — | $\overline{RWDT}$ | $\overline{RMCLR}$ | $\overline{RI}$ | $\overline{POR}$ | $\overline{BOR}$ |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| HC = Bit is cleared by hardware | | HS = Bit is set by hardware |
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as ‘0’ |
| u = Bit is unchanged | x = Bit is unknown | -m/n = Value at POR and BOR/Value at all other Resets |
| ‘1’ = Bit is set | ‘0’ = Bit is cleared | q = Value depends on condition |

bit 7 **STKOVF:** Stack Overflow Flag bit
1 = A Stack Overflow occurred
0 = A Stack Overflow has not occurred or cleared by firmware

bit 6 **STKUNF:** Stack Underflow Flag bit
1 = A Stack Underflow occurred
0 = A Stack Underflow has not occurred or cleared by firmware

bit 5 **Unimplemented:** Read as ‘0’

bit 4 **$\overline{RWDT}$:** Watchdog Timer Reset Flag bit
1 = A Watchdog Timer Reset has not occurred or set to ‘1’ by firmware
0 = A Watchdog Timer Reset has occurred (cleared by hardware)

bit 3 **$\overline{RMCLR}$:** $\overline{MCLR}$ Reset Flag bit
1 = A $\overline{MCLR}$ Reset has not occurred or set to ‘1’ by firmware
0 = A $\overline{MCLR}$ Reset has occurred (cleared by hardware)

bit 2 **$\overline{RI}$:** RESET Instruction Flag bit
1 = A RESET instruction has not been executed or set to ‘1’ by firmware
0 = A RESET instruction has been executed (cleared by hardware)

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 Power-on Reset or Brown-out Reset occurs)

**TABLE 5-5: SUMMARY OF REGISTERS ASSOCIATED WITH RESETS**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|
| BORCON | SBOREN | BORFS | — | — | — | — | — | BORRDY | 55 |
| PCON | STKOVF | STKUNF | — | $\overline{\text{RWDT}}$ | $\overline{\text{RMCLR}}$ | $\overline{\text{RI}}$ | $\overline{\text{POR}}$ | $\overline{\text{BOR}}$ | 59 |
| STATUS | — | — | — | $\overline{\text{TO}}$ | $\overline{\text{PD}}$ | Z | DC | C | 21 |
| WDTCON | — | — | WDTPS<4:0> | | | | | SWDTEN | 92 |

**Legend:** — = unimplemented location, read as '0'. Shaded cells are not used by Resets.


© 2013-2015 Microchip Technology Inc.

# 6.0 OSCILLATOR MODULE (WITH FAIL-SAFE CLOCK MONITOR)

## 6.1 Overview

The oscillator module has a limited variety of clock sources and selection features that allow it to be used in applications while maximizing performance and minimizing power consumption. Figure 6-1 illustrates a block diagram of the oscillator module.

Clock sources can be supplied from external oscillators, or one of two internal oscillators and PLL circuits, with a choice of speeds selectable via software. Additional clock features include:

- Selectable system clock source between external or internal sources via software.

The oscillator module can be configured in one of the following clock modes.

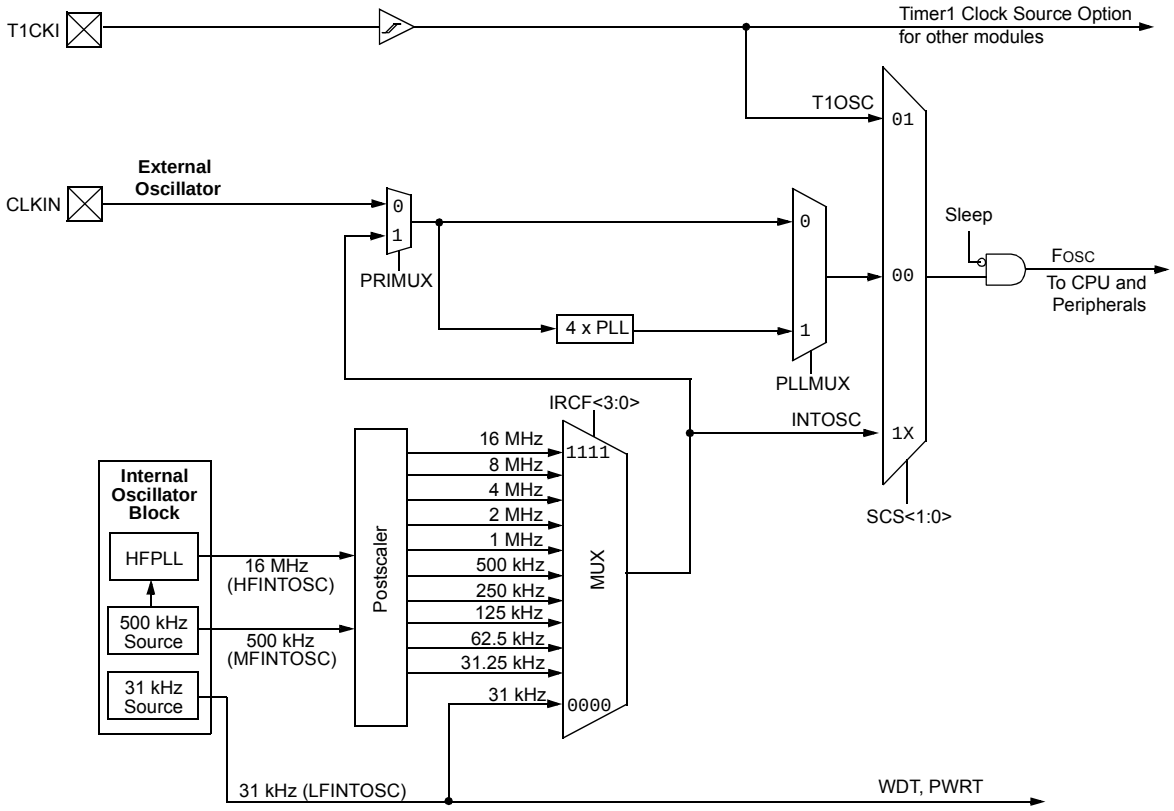
1. ECL – External Clock Low-Power mode (0 MHz to 0.5 MHz)
2. ECM – External Clock Medium Power mode (0.5 MHz to 4 MHz)
3. ECH – External Clock High-Power mode (4 MHz to 32 MHz)
4. INTOSC – Internal oscillator (31 kHz to 32 MHz)

Clock Source modes are selected by the FOSC<1:0> bits in the Configuration Words. The FOSC bits determine the type of oscillator that will be used when the device is first powered.

The ECH, ECM, and ECL clock modes rely on an external logic level signal as the device clock source. Each mode is optimized for a different frequency range.

The INTOSC internal oscillator block produces low, medium, and high-frequency clock sources, designated LFINTOSC, MFINTOSC and HFINTOSC. (see Internal Oscillator Block, Figure 6-1). A wide selection of device clock frequencies may be derived from these three clock sources.

**FIGURE 6-1: SIMPLIFIED PIC® MCU CLOCK SOURCE BLOCK DIAGRAM**




| Inputs | | | | Outputs | |
|---|---|---|---|---|---|
| **SCS** | **FOSC<1:0>** | **PLLEN or SPLLEN** | **IRCF** | **PRIMUX** | **PLLMUX** |
| =00 | =00 | 0 | x | 1 | 0 |
| | | 1 | =1110 | 1 | 1 |
| | | | ≠1110 | 1 | 0 |
| | ≠00 | 0 | x | 0 | 0 |
| | | 1 | x | 0 | 1 |
| ≠00 | X | X | X | X | X |


© 2013-2015 Microchip Technology Inc.

## 6.2 Clock Source Types

Clock sources can be classified as external or internal.

External clock sources rely on an external clock signal such as an oscillator module.

Internal clock sources are contained within the oscillator module. The internal oscillator block has two internal oscillators and a dedicated Phase-Lock Loop (HFPLL) that are used to generate three internal system clock sources: the 16 MHz High-Frequency Internal Oscillator (HFINTOSC), 500 kHz (MFINTOSC) and the 31 kHz Low-Frequency Internal Oscillator (LFINTOSC).

The system clock can be selected between external or internal clock sources via the System Clock Select (SCS) bits in the OSCCON register. See **Section 6.3 "Clock Switching"** for additional information.

### 6.2.1 EXTERNAL CLOCK SOURCES

An external clock source can be used as the device system clock by performing one of the following actions:

- Program the FOSC<1:0> bits in the Configuration Words to select an external clock source that will be used as the default system clock upon a device Reset.
- Write the SCS<1:0> bits in the OSCCON register to switch the system clock source to an external clock source determined by the value of the FOSC bits.

See **Section 6.3 "Clock Switching"**for more information.

#### 6.2.1.1 EC Mode

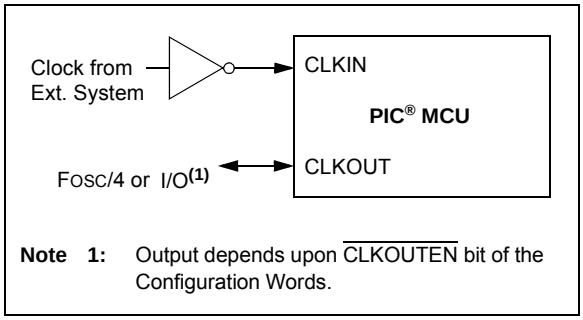
The External Clock (EC) mode allows an externally generated logic level signal to be the system clock source. When operating in this mode, an external clock source is connected to the CLKIN input. CLKOUT is available for general purpose I/O or CLKOUT. Figure 6-2 shows the pin connections for EC mode.

EC mode has three power modes to select from through Configuration Words:

- ECH – High power, 4-32 MHz
- ECM – Medium power, 0.5-4 MHz
- ECL – Low power, 0-0.5 MHz

There is no delay in operation after a Power-on Reset (POR) or wake-up from Sleep. Because the PIC® MCU 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 6-2: EXTERNAL CLOCK (EC) MODE OPERATION**




**Note 1:** Output depends upon $\overline{\text{CLKOUTEN}}$ bit of the Configuration Words.

#### 6.2.1.2 4x PLL

The oscillator module contains a 4x PLL that can be used with both external and internal clock sources to provide a system clock source. The input frequency for the 4x PLL must fall within specifications. See the PLL Clock Timing Specifications in Table 26-9.

The 4x PLL may be enabled for use by one of two methods:

1. Program the PLLEN bit in Configuration Words to a '1'.
2. Write the SPLLEN bit in the OSCCON register to a '1'. If the PLLEN bit in Configuration Words is programmed to a '1', then the value of SPLLEN is ignored.

### 6.2.2 INTERNAL CLOCK SOURCES

The device may be configured to use the internal oscillator block as the system clock by performing one of the following actions:

- Program the FOSC<1:0> bits in Configuration Words to select the INTOSC clock source, which will be used as the default system clock upon a device Reset.
- Write the SCS<1:0> bits in the OSCCON register to switch the system clock source to the internal oscillator during run-time. See **Section 6.3 "Clock Switching"** for more information.

In **INTOSC** mode, CLKIN is available for general purpose I/O. CLKOUT is available for general purpose I/O or CLKOUT.

The function of the CLKOUT pin is determined by the CLKOUTEN bit in Configuration Words.

The internal oscillator block has two independent oscillators and a dedicated Phase-Lock Loop, HFPLL that can produce one of three internal system clock sources.

1. The **HFINTOSC** (High-Frequency Internal Oscillator) is factory calibrated and operates at 16 MHz. The HFINTOSC source is generated from the 500 kHz MFINTOSC source and the dedicated Phase-Lock Loop, HFPLL. The frequency of the HFINTOSC can be user-adjusted via software using the OSCTUNE register (Register 6-3).
2. The **MFINTOSC** (Medium-Frequency Internal Oscillator) is factory calibrated and operates at 500 kHz. The frequency of the MFINTOSC can be user-adjusted via software using the OSCTUNE register (Register 6-3).
3. The **LFINTOSC** (Low-Frequency Internal Oscillator) is uncalibrated and operates at 31 kHz.

#### 6.2.2.1 HFINTOSC

The High-Frequency Internal Oscillator (HFINTOSC) is a factory calibrated 16 MHz internal clock source. The frequency of the HFINTOSC can be altered via software using the OSCTUNE register (Register 6-3).

The output of the HFINTOSC connects to a postscaler and multiplexer (see Figure 6-1). One of multiple frequencies derived from the HFINTOSC can be selected via software using the IRCF<3:0> bits of the OSCCON register. See **Section 6.2.2.7 "Internal Oscillator Clock Switch Timing"** for more information.

The HFINTOSC is enabled by:

- Configure the IRCF<3:0> bits of the OSCCON register for the desired HF frequency, and
- FOSC<1:0> = 00, or
- Set the System Clock Source (SCS) bits of the OSCCON register to '1x'

A fast start-up oscillator allows internal circuits to power up and stabilize before switching to HFINTOSC.

The High-Frequency Internal Oscillator Ready bit (HFIOFR) of the OSCSTAT register indicates when the HFINTOSC is running.

The High-Frequency Internal Oscillator Status Locked bit (HFIOFL) of the OSCSTAT register indicates when the HFINTOSC is running within 2% of its final value.

The High-Frequency Internal Oscillator Stable bit (HFIOFS) of the OSCSTAT register indicates when the HFINTOSC is running within 0.5% of its final value.

#### 6.2.2.2 MFINTOSC

The Medium-Frequency Internal Oscillator (MFINTOSC) is a factory calibrated 500 kHz internal clock source. The frequency of the MFINTOSC can be altered via software using the OSCTUNE register (Register 6-3).

The output of the MFINTOSC connects to a postscaler and multiplexer (see Figure 6-1). One of nine frequencies derived from the MFINTOSC can be selected via software using the IRCF<3:0> bits of the OSCCON register. See **Section 6.2.2.7 "Internal Oscillator Clock Switch Timing"** for more information.

The MFINTOSC is enabled by:

- Configure the IRCF<3:0> bits of the OSCCON register for the desired HF frequency, and
- FOSC<1:0> = 00, or
- Set the System Clock Source (SCS) bits of the OSCCON register to '1x'

The Medium-Frequency Internal Oscillator Ready bit (MFIOFR) of the OSCSTAT register indicates when the MFINTOSC is running.


© 2013-2015 Microchip Technology Inc.

#### 6.2.2.3 Internal Oscillator Frequency Adjustment

The 500 kHz internal oscillator is factory calibrated. This internal oscillator can be adjusted in software by writing to the OSCTUNE register (Register 6-3). Since the HFINTOSC and MFINTOSC clock sources are derived from the 500 kHz internal oscillator a change in the OSCTUNE register value will apply to both.

The default value of the OSCTUNE register is '0'. The value is a 6-bit two's complement number. A value of 1Fh will provide an adjustment to the maximum frequency. A value of 20h will provide an adjustment to the minimum frequency.

When the OSCTUNE register is modified, the oscillator frequency will begin shifting to the new frequency. 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.

#### 6.2.2.4 LFINTOSC

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

The output of the LFINTOSC connects to a multiplexer (see Figure 6-1). Select 31 kHz, via software, using the IRCF<3:0> bits of the OSCCON register. See **Section 6.2.2.7 "Internal Oscillator Clock Switch Timing"** for more information. 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<3:0> bits of the OSCCON register = 000) as the system clock source (SCS bits of the OSCCON register = 1x), or when any of the following are enabled:

- Configure the IRCF<3:0> bits of the OSCCON register for the desired LF frequency, and
- FOSC<1:0> = 00, or
- Set the System Clock Source (SCS) bits of the OSCCON register to '1x'

Peripherals that use the LFINTOSC are:

- Power-up Timer (PWRT)
- Watchdog Timer (WDT)
- Fail-Safe Clock Monitor (FSCM)

The Low-Frequency Internal Oscillator Ready bit (LFIOFR) of the OSCSTAT register indicates when the LFINTOSC is running.

#### 6.2.2.5 Internal Oscillator Frequency Selection

The system clock speed can be selected via software using the Internal Oscillator Frequency Select bits IRCF<3:0> of the OSCCON register.

The postscaled output of the 16 MHz HFINTOSC, 500 kHz MFINTOSC, and 31 kHz LFINTOSC connect to a multiplexer (see Figure 6-1). The Internal Oscillator Frequency Select bits IRCF<3:0> of the OSCCON register select the frequency output of the internal oscillators. One of the following frequencies can be selected via software:

- 32 MHz (requires 4x PLL)
- 16 MHz
- 8 MHz
- 4 MHz
- 2 MHz
- 1 MHz
- 500 kHz (default after Reset)
- 250 kHz
- 125 kHz
- 62.5 kHz
- 31.25 kHz
- 31 kHz (LFINTOSC)

| | |
|---|---|
| **Note:** | Following any Reset, the IRCF<3:0> bits of the OSCCON register are set to '0111' and the frequency selection is set to 500 kHz. The user can modify the IRCF bits to select a different frequency. |

The IRCF<3:0> bits of the OSCCON register allow duplicate selections for some frequencies. These duplicate choices can offer system design trade-offs. Lower power consumption can be obtained when changing oscillator sources for a given frequency. Faster transition times can be obtained between frequency changes that use the same oscillator source.

#### 6.2.2.6 32 MHz Internal Oscillator Frequency Selection

The Internal Oscillator Block can be used with the 4x PLL associated with the External Oscillator Block to produce a 32 MHz internal system clock source. The following settings are required to use the 32 MHz internal clock source:

- The FOSC bits in Configuration Words must be set to use the INTOSC source as the device system clock (FOSC<1:0> = 00).
- The SCS bits in the OSCCON register must be cleared to use the clock determined by FOSC<1:0> in Configuration Words (SCS<1:0> = 00).
- The IRCF bits in the OSCCON register must be set to the 8 MHz HFINTOSC set to use (IRCF<3:0> = 1110).
- The SPLLEN bit in the OSCCON register must be set to enable the 4x PLL, or the PLLEN bit of the Configuration Words must be programmed to a '1'.

| | |
|---|---|
| **Note:** | When using the PLLEN bit of the Configuration Words, the 4x PLL cannot be disabled by software and the SPLLEN option will not be available. |

The 4x PLL is not available for use with the internal oscillator when the SCS bits of the OSCCON register are set to '1x'. The SCS bits must be set to '00' to use the 4x PLL with the internal oscillator.

#### 6.2.2.7 Internal Oscillator Clock Switch Timing

When switching between the HFINTOSC, MFINTOSC and the LFINTOSC, the new oscillator may already be shut down to save power (see Figure 6-3). If this is the case, there is a delay after the IRCF<3:0> bits of the OSCCON register are modified before the frequency selection takes place. The OSCSTAT register will reflect the current active status of the HFINTOSC, MFINTOSC and LFINTOSC oscillators. The sequence of a frequency selection is as follows:

1. IRCF<3:0> bits of the OSCCON register are modified.
2. If the new clock is shut down, a clock start-up delay is started.
3. Clock switch circuitry waits for a falling edge of the current clock.
4. The current clock is held low and the clock switch circuitry waits for a rising edge in the new clock.
5. The new clock is now active.
6. The OSCSTAT register is updated as required.
7. Clock switch is complete.

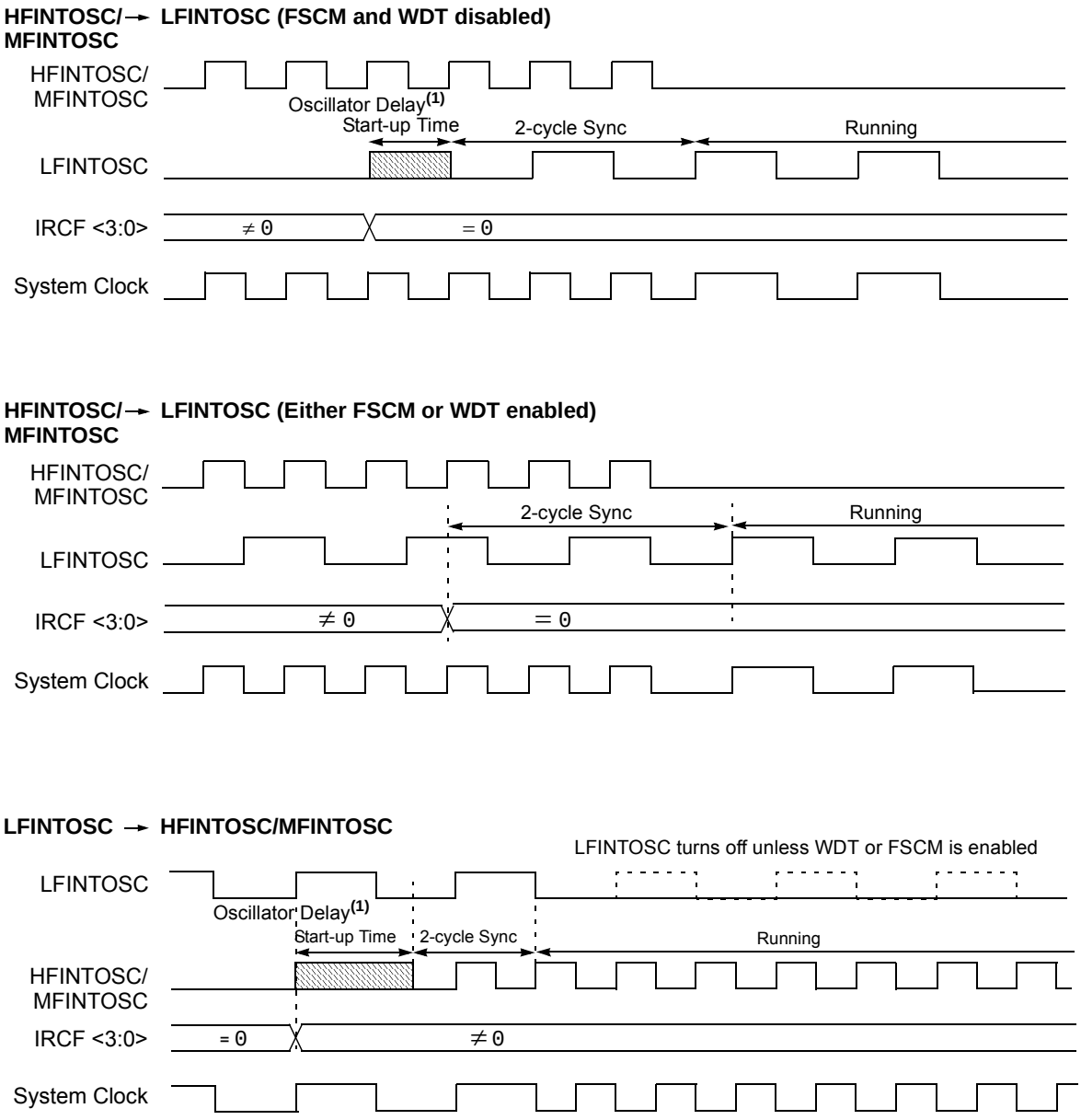
See Figure 6-3 for more details.

If the internal oscillator speed is switched between two clocks of the same source, there is no start-up delay before the new frequency is selected. Clock switching time delays are shown in Table 6-1.

Start-up delay specifications are located in the oscillator tables of **Section 26.0 "Electrical Specifications"**.


© 2013-2015 Microchip Technology Inc.

**FIGURE 6-3: INTERNAL OSCILLATOR SWITCH TIMING**




**Note 1:** See Table 6-1, Oscillator Switching Delays, for more information.

## 6.3 Clock Switching

The system clock source can be switched between external and internal clock sources via software using the System Clock Select (SCS) bits of the OSCCON register. The following clock sources can be selected using the SCS bits:

- Default system oscillator determined by FOSC bits in Configuration Words
- Timer1 32 kHz crystal oscillator
- Internal Oscillator Block (INTOSC)

### 6.3.1 SYSTEM CLOCK SELECT (SCS) BITS

The System Clock Select (SCS) bits of the OSCCON register select the system clock source that is used for the CPU and peripherals.

- When the SCS bits of the OSCCON register = 00, the system clock source is determined by the value of the FOSC<1:0> bits in the Configuration Words.
- When the SCS bits of the OSCCON register = 01, the system clock source is the secondary oscillator.
- When the SCS bits of the OSCCON register = 1x, the system clock source is chosen by the internal oscillator frequency selected by the IRCF<3:0> bits of the OSCCON register. After a Reset, the SCS bits of the OSCCON register are 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 bits of the OSCCON register. The user can monitor the OSTS bit of the OSCSTAT register to determine the current system clock source.

When switching between clock sources, a delay is required to allow the new clock to stabilize. These oscillator delays are shown in Table 6-1.

**TABLE 6-1: OSCILLATOR SWITCHING DELAYS**

| Switch From | Switch To | Frequency | Oscillator Delay |
|---|---|---|---|
| Sleep/POR | LFINTOSC(1)<br>MFINTOSC(1)<br>HFINTOSC(1) | 31 kHz<br>31.25 kHz-500 kHz<br>31.25 kHz-16 MHz | 2 cycles |
| Sleep/POR | EC | DC – 32 MHz | 2 cycles |
| LFINTOSC | EC | DC – 32 MHz | 1 cycle of each |
| Any clock source | MFINTOSC(1)<br>HFINTOSC(1) | 31.25 kHz-500 kHz<br>31.25 kHz-16 MHz | 2 μs (approx.) |
| Any clock source | LFINTOSC(1) | 31 kHz | 1 cycle of each |
| PLL inactive | PLL active | 16-32 MHz | 2 ms (approx.) |

**Note 1:** PLL inactive.


© 2013-2015 Microchip Technology Inc.

## 6.4 Register Definitions: Oscillator Control

**REGISTER 6-1: OSCCON: OSCILLATOR CONTROL REGISTER**

| R/W-0/0 | R/W-0/0 | R/W-1/1 | R/W-1/1 | R/W-1/1 | U-0 | R/W-0/0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| SPLLEN | IRCF<3:0> | | | | — | SCS<1:0> | |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| '1' = Bit is set | '0' = Bit is cleared | |

bit 7 **SPLLEN:** Software PLL Enable bit
If PLLEN in Configuration Words = 1:
SPLLEN bit is ignored. 4x PLL is always enabled (subject to oscillator requirements)
If PLLEN in Configuration Words = 0:
1 = 4x PLL Is enabled
0 = 4x PLL is disabled

bit 6-3 **IRCF<3:0>:** Internal Oscillator Frequency Select bits
1111 = 16 MHz HF
1110 = 8 MHz or 32 MHz HF(2)
1101 = 4 MHz HF
1100 = 2 MHz HF
1011 = 1 MHz HF
1010 = 500 kHz HF(1)
1001 = 250 kHz HF(1)
1000 = 125 kHz HF(1)
0111 = 500 kHz MF (default upon Reset)
0110 = 250 kHz MF
0101 = 125 kHz MF
0100 = 62.5 kHz MF
0011 = 31.25 kHz HF(1)
0010 = 31.25 kHz MF
000x = 31 kHz LF

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

bit 1-0 **SCS<1:0>:** System Clock Select bits
1x = Internal oscillator block
01 =Reserved
00 = Clock determined by FOSC<1:0> in Configuration Words

**Note 1:** Duplicate frequency derived from HFINTOSC.
**2:** 32 MHz when SPLLEN bit is set. Refer to **Section 6.2.2.6 "32 MHz Internal Oscillator Frequency Selection"**.

**REGISTER 6-2: OSCSTAT: OSCILLATOR STATUS REGISTER**

| U-0 | R-0/q | R-q/q | R-0/q | R-0/q | R-q/q | R-0/0 | R-0/q |
|---|---|---|---|---|---|---|---|
| — | PLLR | OSTS | HFIOFR | HFIOFL | MFIOFR | LFIOFR | HFIOFS |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as ‘0’ |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| ‘1’ = Bit is set | ‘0’ = Bit is cleared | q = Conditional |

bit 7 **Unimplemented:** Read as ‘0’

bit 6 **PLLR** 4x PLL Ready bit
1 = 4x PLL is ready
0 = 4x PLL is not ready

bit 5 **OSTS:** Oscillator Start-up Timer Status bit
1 = Running from the clock defined by the FOSC<1:0> bits of the Configuration Words
0 = Running from an internal oscillator (FOSC<1:0> = 00)

bit 4 **HFIOFR:** High-Frequency Internal Oscillator Ready bit
1 = HFINTOSC is ready
0 = HFINTOSC is not ready

bit 3 **HFIOFL:** High-Frequency Internal Oscillator Locked bit
1 = HFINTOSC is at least 2% accurate
0 = HFINTOSC is not 2% accurate

bit 2 **MFIOFR:** Medium-Frequency Internal Oscillator Ready bit
1 = MFINTOSC is ready
0 = MFINTOSC is not ready

bit 1 **LFIOFR:** Low-Frequency Internal Oscillator Ready bit
1 = LFINTOSC is ready
0 = LFINTOSC is not ready

bit 0 **HFIOFS:** High-Frequency Internal Oscillator Stable bit
1 = HFINTOSC is at least 0.5% accurate
0 = HFINTOSC is not 0.5% accurate


© 2013-2015 Microchip Technology Inc.

**REGISTER 6-3: OSCTUNE: OSCILLATOR TUNING REGISTER**

| U-0 | U-0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| — | — | TUN<5:0> | | | | | |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| '1' = Bit is set | '0' = Bit is cleared | |

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

bit 5-0 **TUN<5:0>:** Frequency Tuning bits

100000 = Minimum frequency
•
•
•
111111 =
000000 = Oscillator module is running at the factory-calibrated frequency
000001 =
•
•
•
011110 =
011111 = Maximum frequency

**TABLE 6-2: SUMMARY OF REGISTERS ASSOCIATED WITH CLOCK SOURCES**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|
| OSCCON | SPLLEN | IRCF<3:0> | | | | — | SCS<1:0> | | 69 |
| OSCSTAT | — | PLLR | OSTS | HFIOFR | HFIOFL | MFIOFR | LFIOFR | HFIOFS | 70 |
| OSCTUNE | — | — | TUN<5:0> | | | | | | 71 |
| PIR2 | — | — | — | — | BCL1IF | — | — | CCP2IF | 82 |
| PIE2 | — | — | — | — | BCL1IE | — | — | CCP2IE | 79 |

**Legend:** — = unimplemented location, read as '0'. Shaded cells are not used by clock sources.

**TABLE 6-3: SUMMARY OF CONFIGURATION BITS ASSOCIATED WITH CLOCK SOURCES**

| Name | Bits | Bit -/7 | Bit -/6 | Bit 13/5 | Bit 12/4 | Bit 11/3 | Bit 10/2 | Bit 9/1 | Bit 8/0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|---|
| CONFIG1 | 13:8 | — | — | — | — | $\overline{\text{CLKOUTEN}}$ | BOREN<1:0> | | — | 49 |
| | 7:0 | $\overline{\text{CP}}$ | MCLRE | $\overline{\text{PWRTE}}$ | WDTE<1:0> | | — | FOSC<1:0> | | |
| CONFIG2 | 13:8 | — | — | LVP | $\overline{\text{DEBUG}}$ | $\overline{\text{LPBOR}}$ | BORV | STVREN | PLLEN | 50 |
| | 7:0 | ZCDDIS | — | — | — | — | PPS1WAY | WRT<1:0> | | |

**Legend:** — = unimplemented location, read as '0'. Shaded cells are not used by clock sources.

## 7.0 INTERRUPTS

The interrupt feature allows certain events to preempt normal program flow. Firmware is used to determine the source of the interrupt and act accordingly. Some interrupts can be configured to wake the MCU from Sleep mode.

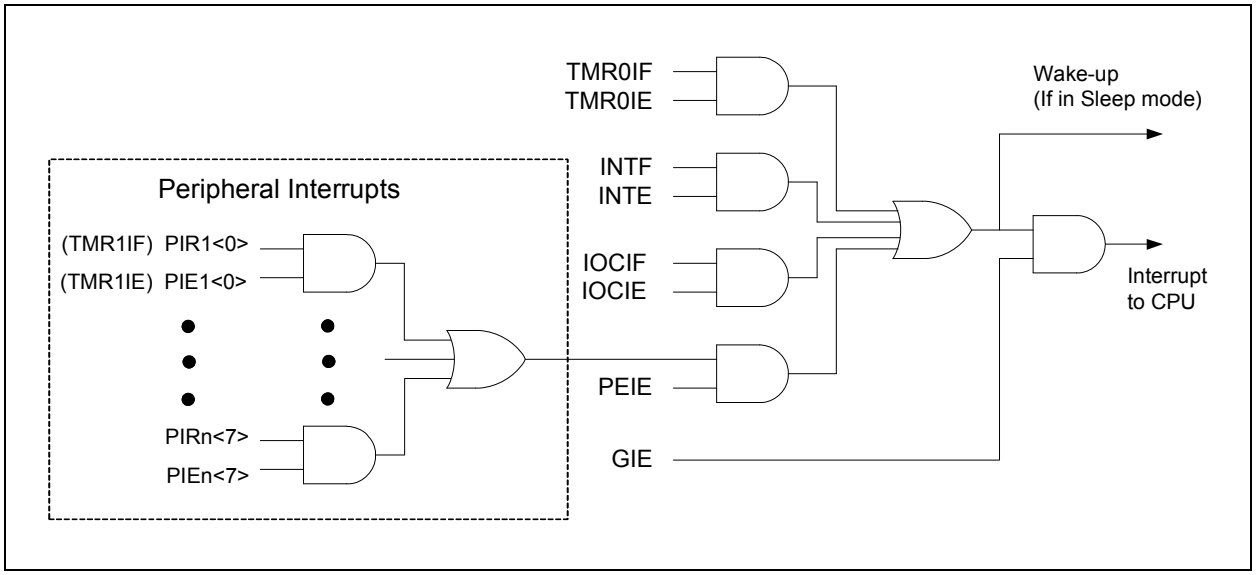
This chapter contains the following information for Interrupts:

- Operation
- Interrupt Latency
- Interrupts During Sleep
- INT Pin
- Automatic Context Saving

Many peripherals produce interrupts. Refer to the corresponding chapters for details.

A block diagram of the interrupt logic is shown in Figure 7-1.

**FIGURE 7-1: INTERRUPT LOGIC**





© 2013-2015 Microchip Technology Inc.

## 7.1 Operation

Interrupts are disabled upon any device Reset. They are enabled by setting the following bits:

- GIE bit of the INTCON register
- Interrupt Enable bit(s) for the specific interrupt event(s)
- PEIE bit of the INTCON register (if the Interrupt Enable bit of the interrupt event is contained in the PIE1 or PIE2 registers)

The INTCON, PIR1 and PIR2 registers record individual interrupts via interrupt flag bits. Interrupt flag bits will be set, regardless of the status of the GIE, PEIE and individual interrupt enable bits.

The following events happen when an interrupt event occurs while the GIE bit is set:

- Current prefetched instruction is flushed
- GIE bit is cleared
- Current Program Counter (PC) is pushed onto the stack
- Critical registers are automatically saved to the shadow registers (See **"Section 7.5 "Automatic Context Saving"**)
- PC is loaded with the interrupt vector 0004h

The firmware within the Interrupt Service Routine (ISR) should determine the source of the interrupt by polling the interrupt flag bits. The interrupt flag bits must be cleared before exiting the ISR to avoid repeated interrupts. Because the GIE bit is cleared, any interrupt that occurs while executing the ISR will be recorded through its interrupt flag, but will not cause the processor to redirect to the interrupt vector.

The `RETFIE` instruction exits the ISR by popping the previous address from the stack, restoring the saved context from the shadow registers and setting the GIE bit.

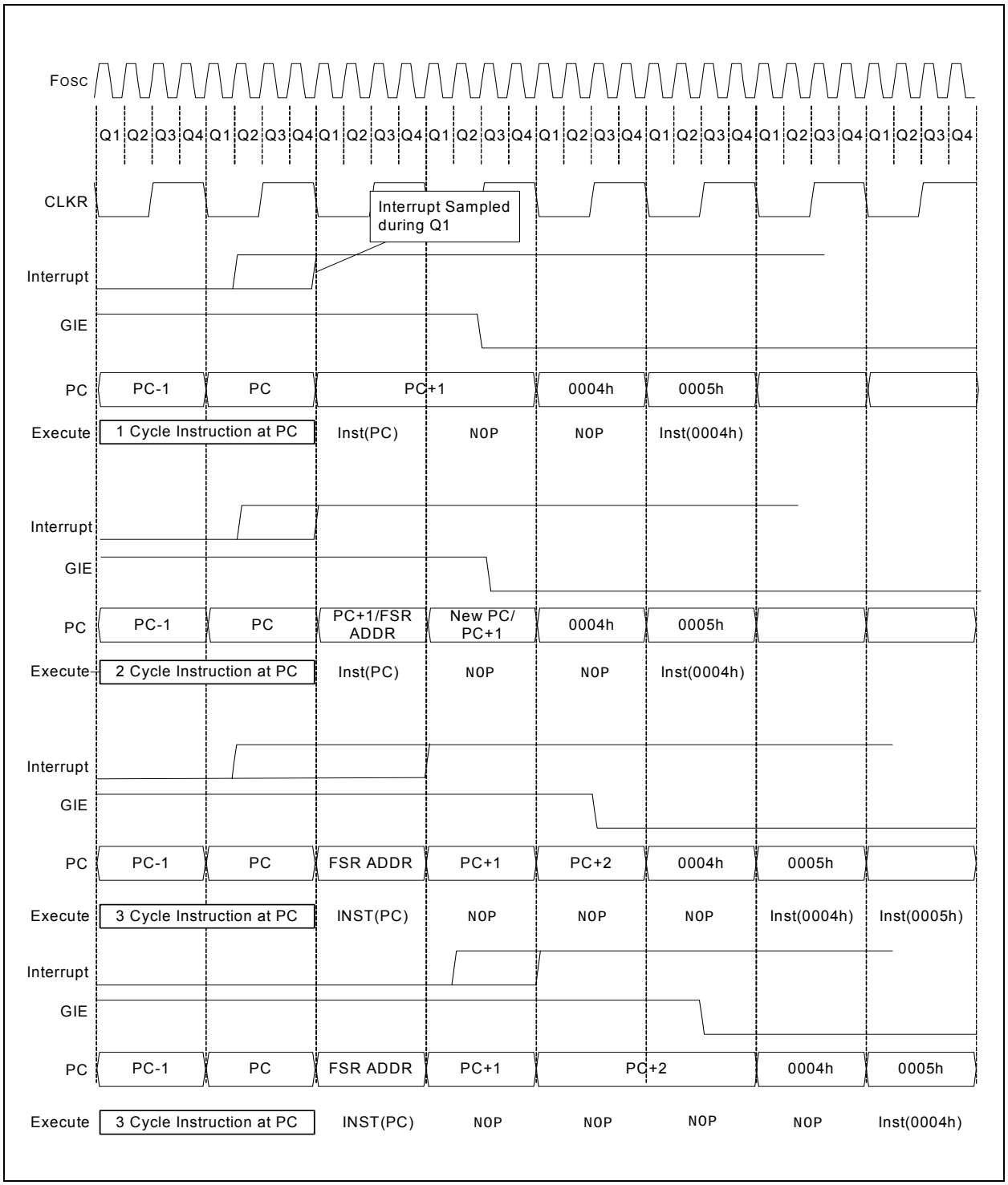
For additional information on a specific interrupt's operation, refer to its peripheral chapter.

| | |
|---|---|
| **Note 1:** | Individual interrupt flag bits are set, regardless of the state of any other enable bits. |
| **2:** | All interrupts will be ignored while the GIE bit is cleared. Any interrupt occurring while the GIE bit is clear will be serviced when the GIE bit is set again. |

## 7.2 Interrupt Latency

Interrupt latency is defined as the time from when the interrupt event occurs to the time code execution at the interrupt vector begins. The latency for synchronous interrupts is three or four instruction cycles. For asynchronous interrupts, the latency is three to five instruction cycles, depending on when the interrupt occurs. See Figure 7-2 and Figure 7-3 for more details.

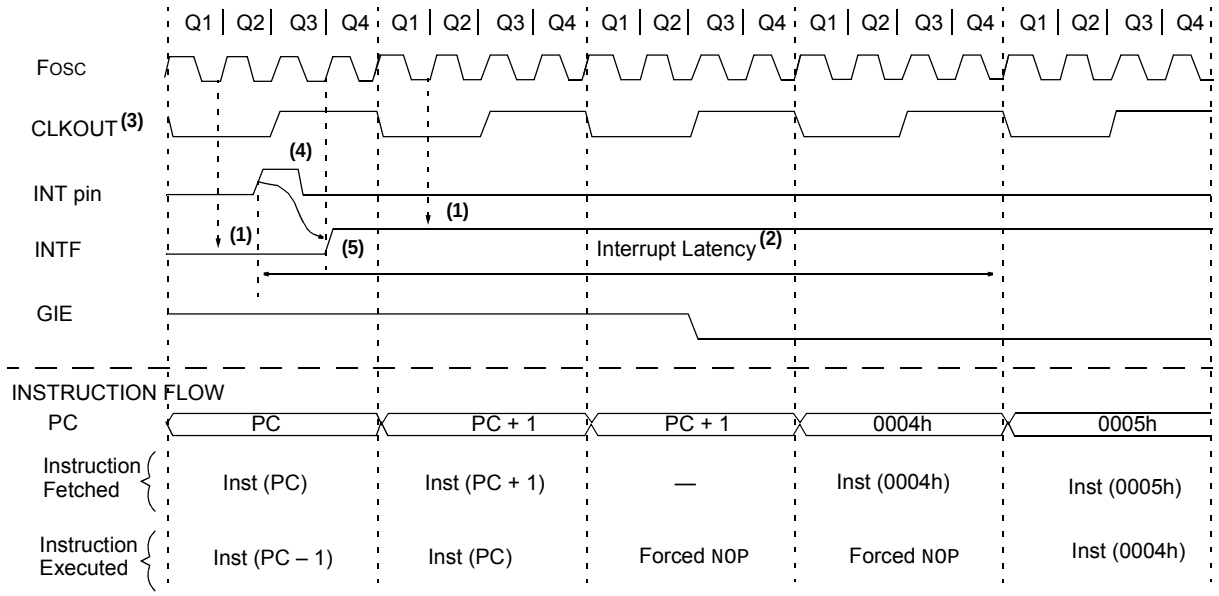
**FIGURE 7-2: INTERRUPT LATENCY**





© 2013-2015 Microchip Technology Inc.

**FIGURE 7-3: INT PIN INTERRUPT TIMING**




**Note 1:** INTF flag is sampled here (every Q1).

**2:** Asynchronous interrupt latency = 3-5 TCY. Synchronous latency = 3-4 TCY, where TCY = instruction cycle time. Latency is the same whether Inst (PC) is a single cycle or a 2-cycle instruction.

**3:** CLKOUT not available in all oscillator modes.

**4:** For minimum width of INT pulse, refer to AC specifications in **Section 26.0 "Electrical Specifications"**".

**5:** INTF is enabled to be set any time during the Q4-Q1 cycles.

## 7.3 Interrupts During Sleep

Some interrupts can be used to wake from Sleep. To wake from Sleep, the peripheral must be able to operate without the system clock. The interrupt source must have the appropriate Interrupt Enable bit(s) set prior to entering Sleep.

On waking from Sleep, if the GIE bit is also set, the processor will branch to the interrupt vector. Otherwise, the processor will continue executing instructions after the SLEEP instruction. The instruction directly after the SLEEP instruction will always be executed before branching to the ISR. Refer to **Section 8.0 "Power-Down Mode (Sleep)"** for more details.

## 7.4 INT Pin

The INT pin can be used to generate an asynchronous edge-triggered interrupt. This interrupt is enabled by setting the INTE bit of the INTCON register. The INTEDG bit of the OPTION_REG register determines on which edge the interrupt will occur. When the INTEDG bit is set, the rising edge will cause the interrupt. When the INTEDG bit is clear, the falling edge will cause the interrupt. The INTF bit of the INTCON register will be set when a valid edge appears on the INT pin. If the GIE and INTE bits are also set, the processor will redirect program execution to the interrupt vector.

## 7.5 Automatic Context Saving

Upon entering an interrupt, the return PC address is saved on the stack. Additionally, the following registers are automatically saved in the shadow registers:

- W register
- STATUS register (except for $\overline{TO}$ and $\overline{PD}$)
- BSR register
- FSR registers
- PCLATH register

Upon exiting the Interrupt Service Routine, these registers are automatically restored. Any modifications to these registers during the ISR will be lost. If modifications to any of these registers are desired, the corresponding shadow register should be modified and the value will be restored when exiting the ISR. The shadow registers are available in Bank 31 and are readable and writable. Depending on the user's application, other registers may also need to be saved.


© 2013-2015 Microchip Technology Inc.

## 7.6 Register Definitions: Interrupt Control

**REGISTER 7-1: INTCON: INTERRUPT CONTROL REGISTER**

| R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R-0/0 |
|---|---|---|---|---|---|---|---|
| GIE | PEIE | TMR0IE | INTE | IOCIE | TMR0IF | INTF | IOCIF(1) |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| '1' = Bit is set | '0' = Bit is cleared | |

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

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

bit 5 **TMR0IE:** Timer0 Overflow Interrupt Enable bit
1 = Enables the Timer0 interrupt
0 = Disables the Timer0 interrupt

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

bit 3 **IOCIE:** Interrupt-on-Change Enable bit
1 = Enables the interrupt-on-change
0 = Disables the interrupt-on-change

bit 2 **TMR0IF:** Timer0 Overflow Interrupt Flag bit
1 = TMR0 register has overflowed
0 = TMR0 register did not overflow

bit 1 **INTF:** INT External Interrupt Flag bit
1 = The INT external interrupt occurred
0 = The INT external interrupt did not occur

bit 0 **IOCIF:** Interrupt-on-Change Interrupt Flag bit(1)
1 = When at least one of the interrupt-on-change pins changed state
0 = None of the interrupt-on-change pins have changed state

**Note 1:** The IOCIF Flag bit is read-only and cleared when all the interrupt-on-change flags in the IOCxF registers have been cleared by software.

**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, of the INTCON register. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

**REGISTER 7-2: PIE1: PERIPHERAL INTERRUPT ENABLE REGISTER 1**

| R/W-0/0 | R/W-0/0 | U-0 | U-0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| TMR1GIE | ADIE | — | — | SSP1IE | CCP1IE | TMR2IE | TMR1IE |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| '1' = Bit is set | '0' = Bit is cleared | |

bit 7 **TMR1GIE:** Timer1 Gate Interrupt Enable bit
1 = Enables the Timer1 gate acquisition interrupt
0 = Disables the Timer1 gate acquisition interrupt

bit 6 **ADIE:** Analog-to-Digital Converter (ADC) Interrupt Enable bit
1 = Enables the ADC interrupt
0 = Disables the ADC interrupt

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

bit 3 **SSP1IE:** Synchronous Serial Port (MSSP) Interrupt Enable bit
1 = Enables the MSSP interrupt
0 = Disables the MSSP interrupt

bit 2 **CCP1IE:** CCP1 Interrupt Enable bit
1 = Enables the CCP1 interrupt
0 = Disables the CCP1 interrupt

bit 1 **TMR2IE:** TMR2 to PR2 Match Interrupt Enable bit
1 = Enables the Timer2 to PR2 match interrupt
0 = Disables the Timer2 to PR2 match interrupt

bit 0 **TMR1IE:** Timer1 Overflow Interrupt Enable bit
1 = Enables the Timer1 overflow interrupt
0 = Disables the Timer1 overflow interrupt

**Note:** Bit PEIE of the INTCON register must be set to enable any peripheral interrupt.


© 2013-2015 Microchip Technology Inc.

**REGISTER 7-3: PIE2: PERIPHERAL INTERRUPT ENABLE REGISTER 2**

| U-0 | U-0 | U-0 | U-0 | R/W-0/0 | U-0 | U-0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| — | — | — | — | BCL1IE | — | — | CCP2IE |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| '1' = Bit is set | '0' = Bit is cleared | |

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

bit 3 **BCL1IE:** MSSP Bus Collision Interrupt Enable bit
- 1 = Enables the MSSP Bus Collision Interrupt
- 0 = Disables the MSSP Bus Collision Interrupt

bit 2-1 **Unimplemented:** Read as '0'

bit 0 **CCP2IE:** CCP2 Interrupt Enable bit
- 1 = Enables the CCP2 interrupt
- 0 = Disables the CCP2 interrupt

**Note:** Bit PEIE of the INTCON register must be set to enable any peripheral interrupt.

**REGISTER 7-4: PIE3: PERIPHERAL INTERRUPT ENABLE REGISTER 3**

| U-0 | U-0 | U-0 | R/W-0/0 | U-0 | U-0 | U-0 | U-0 |
|---|---|---|---|---|---|---|---|
| — | — | — | ZCDIE | — | — | — | — |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as ‘0’ |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| ‘1’ = Bit is set | ‘0’ = Bit is cleared | |

bit 7-5 **Unimplemented**: Read as ‘0’

bit 4 **ZCDIE:** Zero-Cross Detection Interrupt Enable bit
1 = ZCD interrupt enabled
0 = ZCD interrupt disabled

bit 3-0 **Unimplemented**: Read as ‘0’

**Note:** Bit PEIE of the INTCON register must be set to enable any peripheral interrupt.


© 2013-2015 Microchip Technology Inc.

**REGISTER 7-5: PIR1: PERIPHERAL INTERRUPT REQUEST REGISTER 1**

| R/W-0/0 | R/W-0/0 | U-0 | U-0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| TMR1GIF | ADIF | — | — | SSP1IF | CCP1IF | TMR2IF | TMR1IF |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as ‘0’ |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| ‘1’ = Bit is set | ‘0’ = Bit is cleared | |

bit 7 **TMR1GIF:** Timer1 Gate Interrupt Flag bit
1 = Interrupt is pending
0 = Interrupt is not pending

bit 6 **ADIF:** Analog-to-Digital Converter (ADC) Interrupt Flag bit
1 = Interrupt is pending
0 = Interrupt is not pending

bit 5-4 **Unimplemented**: Read as ‘0’

bit 3 **SSP1IF:** Synchronous Serial Port (MSSP) Interrupt Flag bit
1 = Interrupt is pending
0 = Interrupt is not pending

bit 2 **CCP1IF:** CCP1 Interrupt Flag bit
1 = Interrupt is pending
0 = Interrupt is not pending

bit 1 **TMR2IF:** Timer2 to PR2 Interrupt Flag bit
1 = Interrupt is pending
0 = Interrupt is not pending

bit 0 **TMR1IF:** Timer1 Overflow Interrupt Flag bit
1 = Interrupt is pending
0 = Interrupt is not pending

**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, of the INTCON register. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

**REGISTER 7-6: PIR2: PERIPHERAL INTERRUPT REQUEST REGISTER 2**

| U-0 | U-0 | U-0 | U-0 | R/W-0/0 | U-0 | U-0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| — | — | — | — | BCL1IF | — | — | CCP2IF |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as ‘0’ |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| ‘1’ = Bit is set | ‘0’ = Bit is cleared | |

bit 7-4 **Unimplemented:** Read as ‘0’

bit 3 **BCL1IF:** MSSP Bus Collision Interrupt Flag bit
1 = Interrupt is pending
0 = Interrupt is not pending

bit 2-1 **Unimplemented:** Read as ‘0’

bit 0 **CCP2IF:** CCP2 Interrupt Flag bit
1 = Interrupt is pending
0 = Interrupt is not pending

**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, of the INTCON register. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.


© 2013-2015 Microchip Technology Inc.

**REGISTER 7-7: PIR3: PERIPHERAL INTERRUPT REQUEST REGISTER 3**

| U-0 | U-0 | U-0 | R/W-0/0 | U-0 | U-0 | U-0 | U-0 |
|---|---|---|---|---|---|---|---|
| — | — | — | ZCDIF | — | — | — | — |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as ‘0’ |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| ‘1’ = Bit is set | ‘0’ = Bit is cleared | |

bit 7-5 **Unimplemented**: Read as ‘0’

bit 4 **ZCDIF:** Zero-Cross Detection Interrupt Flag bit
1 = Interrupt is pending
0 = Interrupt is not pending

bit 3-0 **Unimplemented**: Read as ‘0’

**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, of the INTCON register. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

**TABLE 7-1: SUMMARY OF REGISTERS ASSOCIATED WITH INTERRUPTS**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|
| INTCON | GIE | PEIE | TMR0IE | INTE | IOCIE | TMR0IF | INTF | IOCIF | 77 |
| OPTION_REG | $\overline{\text{WPUEN}}$ | INTEDG | TMR0CS | TMR0SE | PSA | PS<2:0> | | | 168 |
| PIE1 | TMR1GIE | ADIE | — | — | SSP1IE | CCP1IE | TMR2IE | TMR1IE | 78 |
| PIE2 | — | — | — | — | BCL1IE | — | — | CCP2IE | 79 |
| PIE3 | — | — | — | ZCDIE | — | — | — | — | 80 |
| PIR1 | TMR1GIF | ADIF | — | — | SSP1IF | CCP1IF | TMR2IF | TMR1IF | 81 |
| PIR2 | — | — | — | — | BCL1IF | — | — | CCP2IF | 82 |
| PIR3 | — | — | — | ZCDIF | — | — | — | — | 83 |

**Legend:** — = unimplemented location, read as '0'. Shaded cells are not used by interrupts.


© 2013-2015 Microchip Technology Inc.

## 8.0 POWER-DOWN MODE (SLEEP)

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

Upon entering Sleep mode, the following conditions exist:

1. WDT will be cleared but keeps running, if enabled for operation during Sleep.
2. $\overline{PD}$ bit of the STATUS register is cleared.
3. $\overline{TO}$ bit of the STATUS register is set.
4. CPU clock is disabled.
5. 31 kHz LFINTOSC is unaffected and peripherals that operate from it may continue operation in Sleep.
6. Timer1 and peripherals that operate from Timer1 continue operation in Sleep when the Timer1 clock source selected is:
   - LFINTOSC
   - T1CKI
   - Secondary oscillator
7. ADC is unaffected, if the dedicated FRC oscillator is selected.
8. I/O ports maintain the status they had before SLEEP was executed (driving high, low or high-impedance).
9. Resets other than WDT are not affected by Sleep mode.

Refer to individual chapters for more details on peripheral operation during Sleep.

To minimize current consumption, the following conditions should be considered:

- I/O pins should not be floating
- External circuitry sinking current from I/O pins
- Internal circuitry sourcing current from I/O pins
- Current draw from pins with internal weak pull-ups
- Modules using 31 kHz LFINTOSC

I/O pins that are high-impedance inputs should be pulled to VDD or VSS externally to avoid switching currents caused by floating inputs.

Examples of internal circuitry that might be sourcing current include the FVR module. See **Section 14.0 "Fixed Voltage Reference (FVR)"** for more information on the FVR module.

### 8.1 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, if enabled
2. BOR Reset, if enabled
3. POR Reset
4. Watchdog Timer, if enabled
5. Any external interrupt
6. Interrupts by peripherals capable of running during Sleep (see individual peripheral for more information)

The first three events will cause a device Reset. The last three events are considered a continuation of program execution. To determine whether a device Reset or wake-up event occurred, refer to **Section 5.12 "Determining the Cause of a Reset"**.

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 enabled. Wake-up will occur regardless of the state of the GIE bit. If the GIE bit is disabled, the device continues execution at the instruction after the SLEEP instruction. If the GIE bit is enabled, the device executes the instruction after the SLEEP instruction, the device will then call the Interrupt Service Routine. In cases where the execution of the instruction following SLEEP is not desirable, the user should have a NOP after the SLEEP instruction.

The WDT is cleared when the device wakes up from Sleep, regardless of the source of wake-up.

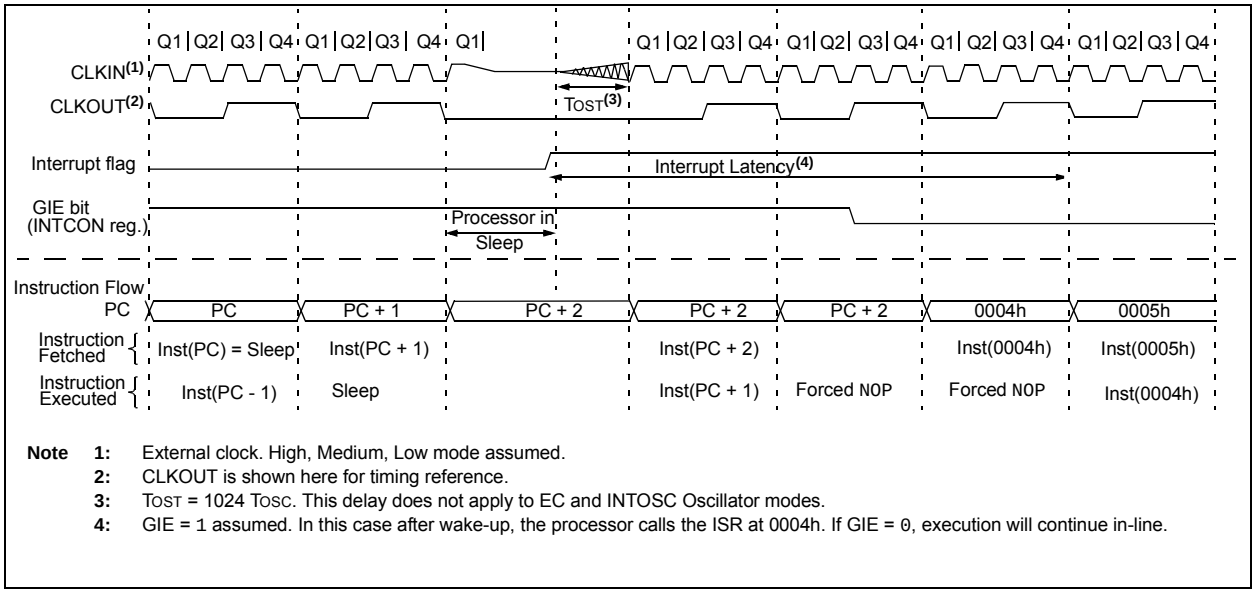
### 8.1.1 WAKE-UP USING INTERRUPTS

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
  - SLEEP instruction will execute as a NOP
  - WDT and WDT prescaler will not be cleared
  - $\overline{TO}$ bit of the STATUS register will not be set
  - $\overline{PD}$ bit of the STATUS register will not be cleared
- If the interrupt occurs **during or after** the execution of a SLEEP instruction
  - SLEEP instruction will be completely executed
  - Device will immediately wake-up from Sleep
  - WDT and WDT prescaler will be cleared
  - $\overline{TO}$ bit of the STATUS register will be set
  - $\overline{PD}$ bit of the STATUS register 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.

**FIGURE 8-1: WAKE-UP FROM SLEEP THROUGH INTERRUPT**




**Note 1:** External clock. High, Medium, Low mode assumed.
**2:** CLKOUT is shown here for timing reference.
**3:** TOST = 1024 TOSC. This delay does not apply to EC and INTOSC Oscillator modes.
**4:** GIE = 1 assumed. In this case after wake-up, the processor calls the ISR at 0004h. If GIE = 0, execution will continue in-line.


© 2013-2015 Microchip Technology Inc.

## 8.2 Low-Power Sleep Mode

The PIC16F1703/7 device contains an internal Low Dropout (LDO) voltage regulator, which allows the device I/O pins to operate at voltages up to 5.5V while the internal device logic operates at a lower voltage. The LDO and its associated reference circuitry must remain active when the device is in Sleep mode. The PIC16F1703/7 allows the user to optimize the operating current in Sleep, depending on the application requirements.

A Low-Power Sleep mode can be selected by setting the VREGPM bit of the VREGCON register. With this bit set, the LDO and reference circuitry are placed in a low-power state when the device is in Sleep.

### 8.2.1 SLEEP CURRENT VS. WAKE-UP TIME

In the default operating mode, the LDO and reference circuitry remain in the normal configuration while in Sleep. The device is able to exit Sleep mode quickly since all circuits remain active. In Low-Power Sleep mode, when waking up from Sleep, an extra delay time is required for these circuits to return to the normal configuration and stabilize.

The Low-Power Sleep mode is beneficial for applications that stay in Sleep mode for long periods of time. The normal mode is beneficial for applications that need to wake from Sleep quickly and frequently.

### 8.2.2 PERIPHERAL USAGE IN SLEEP

Some peripherals that can operate in Sleep mode will not operate properly with the Low-Power Sleep mode selected. The Low-Power Sleep mode is intended for use only with the following peripherals:

- Brown-Out Reset (BOR)
- Watchdog Timer (WDT)
- External interrupt pin/Interrupt-on-change pins
- Timer1 (with external clock source < 100 kHz)

| | |
|---|---|
| **Note:** | The PIC16LF1703/7 does not have a configurable Low-Power Sleep mode. PIC16LF1703/7 is an unregulated device and is always in the lowest power state when in Sleep, with no wake-up time penalty. This device has a lower maximum $V_{DD}$ and I/O voltage than the PIC16F1703/7. See **Section 26.0 "Electrical Specifications"** for more information. |

## 8.3 Register Definitions: Voltage Regulator Control

### REGISTER 8-1: VREGCON: VOLTAGE REGULATOR CONTROL REGISTER(1)

| U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | R/W-0/0 | R/W-1/1 |
|---|---|---|---|---|---|---|---|
| — | — | — | — | — | — | VREGPM | Reserved |
| bit 7 | | | | | | | bit 0 |

| Legend: | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| '1' = Bit is set | '0' = Bit is cleared | |

bit 7-2 **Unimplemented:** Read as '0'

bit 1 **VREGPM:** Voltage Regulator Power Mode Selection bit
- 1 = Low-Power Sleep mode enabled in Sleep(2)
  Draws lowest current in Sleep, slower wake-up
- 0 = Normal-Power mode enabled in Sleep(2)
  Draws higher current in Sleep, faster wake-up

bit 0 **Reserved:** Read as '1'. Maintain this bit set.

**Note 1:** PIC16F1703/7 only.
**2:** See **Section 26.0 "Electrical Specifications"**.

### TABLE 8-1: SUMMARY OF REGISTERS ASSOCIATED WITH POWER-DOWN MODE

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|
| INTCON | GIE | PEIE | TMR0IE | INTE | IOCIE | TMR0IF | INTF | IOCIF | 77 |
| IOCAP | — | — | IOCAP5 | IOCAP4 | IOCAP3 | IOCAP2 | IOCAP1 | IOCAP0 | 137 |
| IOCAN | — | — | IOCAN5 | IOCAN4 | IOCAN3 | IOCAN2 | IOCAN1 | IOCAN0 | 137 |
| IOCAF | — | — | IOCAF5 | IOCAF4 | IOCAF3 | IOCAF2 | IOCAF1 | IOCAF0 | 137 |
| IOCBP(1) | IOCBP7 | IOCBP6 | IOCBP5 | IOCBP4 | — | — | — | — | 138 |
| IOCBN(1) | IOCBN7 | IOCBN6 | IOCBN5 | IOCBN4 | — | — | — | — | 138 |
| IOCBF(1) | IOCBF7 | IOCBF6 | IOCBF5 | IOCBF4 | — | — | — | — | 138 |
| IOCCP | IOCCP7(1) | IOCCP6(1) | IOCCP5 | IOCCP4 | IOCCP3 | IOCCP2 | IOCCP1 | IOCCP0 | 139 |
| IOCCN | IOCCN7(1) | IOCCN6(1) | IOCCN5 | IOCCN4 | IOCCN3 | IOCCN2 | IOCCN1 | IOCCN0 | 139 |
| IOCCF | IOCCF7(1) | IOCCF6(1) | IOCCF5 | IOCCF4 | IOCCF3 | IOCCF2 | IOCCF1 | IOCCF0 | 139 |
| PIE1 | TMR1GIE | ADIE | — | — | SSP1IE | CCP1IE | TMR2IE | TMR1IE | 78 |
| PIE2 | — | — | — | — | BCL1IE | — | — | CCP2IE | 79 |
| PIE3 | — | — | — | ZCDIE | — | — | — | — | 80 |
| PIR1 | TMR1GIF | ADIF | — | — | SSP1IF | CCP1IF | TMR2IF | TMR1IF | 81 |
| PIR2 | — | — | — | — | BCL1IF | — | — | CCP2IF | 82 |
| PIR3 | — | — | — | ZCDIF | — | — | — | — | 83 |
| STATUS | — | — | — | $\overline{TO}$ | $\overline{PD}$ | Z | DC | C | 21 |
| VREGCON(2) | — | — | — | — | — | — | VREGPM | Reserved | 88 |
| WDTCON | — | — | WDTPS<4:0> | | | | | SWDTEN | 92 |

**Legend:** — = unimplemented location, read as '0'. Shaded cells are not used in Power-Down mode.
**Note 1:** PIC16(L)F1707 only.
**2:** PIC16F1703/7 only.


© 2013-2015 Microchip Technology Inc.

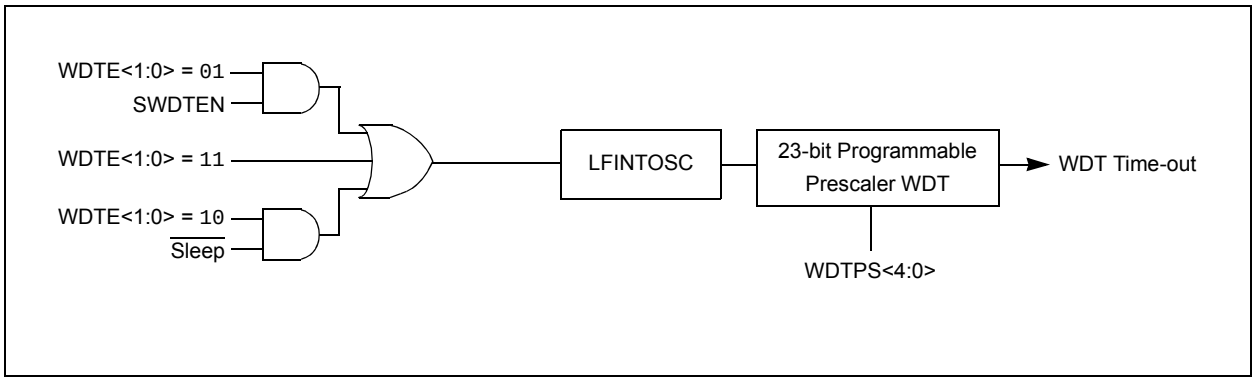
## 9.0 WATCHDOG TIMER (WDT)

The Watchdog Timer is a system timer that generates a Reset if the firmware does not issue a CLRWDT instruction within the time-out period. The Watchdog Timer is typically used to recover the system from unexpected events.

The WDT has the following features:

- Independent clock source
- Multiple operating modes
  - WDT is always on
  - WDT is off when in Sleep
  - WDT is controlled by software
  - WDT is always off
- Configurable time-out period is from 1 ms to 256 seconds (nominal)
- Multiple Reset conditions
- Operation during Sleep

**FIGURE 9-1: WATCHDOG TIMER BLOCK DIAGRAM**



## 9.1 Independent Clock Source

The WDT derives its time base from the 31 kHz LFINTOSC internal oscillator. Time intervals in this chapter are based on a nominal interval of 1 ms. See Table 26-8 for the LFINTOSC specification.

## 9.2 WDT Operating Modes

The Watchdog Timer module has four operating modes controlled by the WDTE<1:0> bits in Configuration Words. See Table 9-1.

### 9.2.1 WDT IS ALWAYS ON

When the WDTE bits of Configuration Words are set to '11', the WDT is always on.

WDT protection is active during Sleep.

### 9.2.2 WDT IS OFF IN SLEEP

When the WDTE bits of Configuration Words are set to '10', the WDT is on, except in Sleep.

WDT protection is not active during Sleep.

### 9.2.3 WDT CONTROLLED BY SOFTWARE

When the WDTE bits of Configuration Words are set to '01', the WDT is controlled by the SWDTEN bit of the WDTCON register.

WDT protection is unchanged by Sleep. See Table 9-1 for more details.

**TABLE 9-1: WDT OPERATING MODES**

| WDTE<1:0> | SWDTEN | Device Mode | WDT Mode |
|---|---|---|---|
| 11 | X | X | Active |
| 10 | X | Awake | Active |
| | | Sleep | Disabled |
| 01 | 1 | X | Active |
| | 0 | | Disabled |
| 00 | X | X | Disabled |

## 9.3 Time-Out Period

The WDTPS bits of the WDTCON register set the time-out period from 1 ms to 256 seconds (nominal). After a Reset, the default time-out period is two seconds.

## 9.4 Clearing the WDT

The WDT is cleared when any of the following conditions occur:

- Any Reset
- CLRWDT instruction is executed
- Device enters Sleep
- Device wakes up from Sleep
- Oscillator fail
- WDT is disabled
- Oscillator Start-up Timer (OST) is running

See Table 9-2 for more information.

## 9.5 Operation During Sleep

When the device enters Sleep, the WDT is cleared. If the WDT is enabled during Sleep, the WDT resumes counting.

When the device exits Sleep, the WDT is cleared again. The WDT remains clear until the OST, if enabled, completes. See **Section 6.0 "Oscillator Module (with Fail-Safe Clock Monitor)"** for more information on the OST.

When a WDT time-out occurs while the device is in Sleep, no Reset is generated. Instead, the device wakes up and resumes operation. The $\overline{TO}$ and $\overline{PD}$ bits in the STATUS register are changed to indicate the event. See STATUS Register (Register 3-1) for more information.


© 2013-2015 Microchip Technology Inc.

**TABLE 9-2: WDT CLEARING CONDITIONS**

| Conditions | WDT |
|---|---|
| WDTE<1:0> = 00 | Cleared |
| WDTE<1:0> = 01 and SWDTEN = 0 | |
| WDTE<1:0> = 10 and enter Sleep | |
| CLRWDT Command | |
| Oscillator Fail Detected | |
| Exit Sleep + System Clock = T1OSC, EXTRC, INTOSC, EXTCLK | |
| Change INTOSC divider (IRCF bits) | Unaffected |

## 9.6 Register Definitions: Watchdog Control

**REGISTER 9-1: WDTCON: WATCHDOG TIMER CONTROL REGISTER**

| U-0 | U-0 | R/W-0/0 | R/W-1/1 | R/W-0/0 | R/W-1/1 | R/W-1/1 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| — | — | WDTPS<4:0>(1) | | | | | SWDTEN |
| bit 7 | | | | | | | bit 0 |

| Legend: | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| u = Bit is unchanged | x = Bit is unknown | -m/n = Value at POR and BOR/Value at all other Resets |
| '1' = Bit is set | '0' = Bit is cleared | |

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

bit 5-1 **WDTPS<4:0>:** Watchdog Timer Period Select bits(1)

Bit Value = Prescale Rate

11111 = Reserved. Results in minimum interval (1:32)
•
•
•
10011 = Reserved. Results in minimum interval (1:32)

10010 = 1:8388608 ($2^{23}$) (Interval 256s nominal)
10001 = 1:4194304 ($2^{22}$) (Interval 128s nominal)
10000 = 1:2097152 ($2^{21}$) (Interval 64s nominal)
01111 = 1:1048576 ($2^{20}$) (Interval 32s nominal)
01110 = 1:524288 ($2^{19}$) (Interval 16s nominal)
01101 = 1:262144 ($2^{18}$) (Interval 8s nominal)
01100 = 1:131072 ($2^{17}$) (Interval 4s nominal)
01011 = 1:65536 (Interval 2s nominal) (Reset value)
01010 = 1:32768 (Interval 1s nominal)
01001 = 1:16384 (Interval 512 ms nominal)
01000 = 1:8192 (Interval 256 ms nominal)
00111 = 1:4096 (Interval 128 ms nominal)
00110 = 1:2048 (Interval 64 ms nominal)
00101 = 1:1024 (Interval 32 ms nominal)
00100 = 1:512 (Interval 16 ms nominal)
00011 = 1:256 (Interval 8 ms nominal)
00010 = 1:128 (Interval 4 ms nominal)
00001 = 1:64 (Interval 2 ms nominal)
00000 = 1:32 (Interval 1 ms nominal)

bit 0 **SWDTEN:** Software Enable/Disable for Watchdog Timer bit

If WDTE<1:0> = 1x:
This bit is ignored.
If WDTE<1:0> = 01:
1 = WDT is turned on
0 = WDT is turned off
If WDTE<1:0> = 00:
This bit is ignored.

**Note 1:** Times are approximate. WDT time is based on 31 kHz LFINTOSC.


© 2013-2015 Microchip Technology Inc.

**TABLE 9-3: SUMMARY OF REGISTERS ASSOCIATED WITH WATCHDOG TIMER**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|
| OSCCON | SPLLEN | IRCF<3:0> | | | | — | SCS<1:0> | | 69 |
| STATUS | — | — | — | $\overline{TO}$ | $\overline{PD}$ | Z | DC | C | 21 |
| WDTCON | — | — | WDTPS<4:0> | | | | | SWDTEN | 92 |

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

**TABLE 9-4: SUMMARY OF CONFIGURATION BITS ASSOCIATED WITH WATCHDOG TIMER**

| Name | Bits | Bit -/7 | Bit -/6 | Bit 13/5 | Bit 12/4 | Bit 11/3 | Bit 10/2 | Bit 9/1 | Bit 8/0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|---|
| CONFIG1 | 13:8 | — | — | — | — | $\overline{CLKOUTEN}$ | BOREN<1:0> | | — | 49 |
| | 7:0 | $\overline{CP}$ | MCLRE | $\overline{PWRTE}$ | WDTE<1:0> | | — | FOSC<1:0> | | |
| CONFIG2 | 13:8 | — | — | LVP | $\overline{DEBUG}$ | $\overline{LPBOR}$ | BORV | STVREN | PLLEN | 50 |
| | 7:0 | ZCDDIS | — | — | — | — | PPS1WAY | WRT<1:0> | | |

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

## 10.0 FLASH PROGRAM MEMORY CONTROL

The Flash program memory is readable and writable during normal operation over the full VDD range. Program memory is indirectly addressed using Special Function Registers (SFRs). The SFRs used to access program memory are:

- PMCON1
- PMCON2
- PMDATL
- PMDATH
- PMADRL
- PMADRH

When accessing the program memory, the PMDATH:PMDATL register pair forms a 2-byte word that holds the 14-bit data for read/write, and the PMADRH:PMADRL register pair forms a 2-byte word that holds the 15-bit address of the program memory location being read.

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 operating voltage range of the device.

The Flash program memory can be protected in two ways; by code protection ($\overline{CP}$ bit in Configuration Words) and write protection (WRT<1:0> bits in Configuration Words).

Code protection ($\overline{CP}$ = 0)**(1)**, disables access, reading and writing, to the Flash program memory via external device programmers. Code protection does not affect the self-write and erase functionality. Code protection can only be reset by a device programmer performing a Bulk Erase to the device, clearing all Flash program memory, Configuration bits and User IDs.

Write protection prohibits self-write and erase to a portion or all of the Flash program memory as defined by the bits WRT<1:0>. Write protection does not affect a device programmers ability to read, write or erase the device.

| | |
|---|---|
| **Note 1:** | Code protection of the entire Flash program memory array is enabled by clearing the $\overline{CP}$ bit of Configuration Words. |

## 10.1 PMADRL and PMADRH Registers

The PMADRH:PMADRL register pair can address up to a maximum of 32K words of program memory. When selecting a program address value, the MSB of the address is written to the PMADRH register and the LSB is written to the PMADRL register.

### 10.1.1 PMCON1 AND PMCON2 REGISTERS

PMCON1 is the control register for Flash program memory accesses.

Control bits RD and WR initiate read and write, respectively. These bits cannot be cleared, only set, in software. They are cleared by 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 occur. On power-up, the WREN bit is clear. The WRERR bit is set when a write operation is interrupted by a Reset during normal operation. In these situations, following Reset, the user can check the WRERR bit and execute the appropriate error handling routine.

The PMCON2 register is a write-only register. Attempting to read the PMCON2 register will return all '0's.

To enable writes to the program memory, a specific pattern (the unlock sequence), must be written to the PMCON2 register. The required unlock sequence prevents inadvertent writes to the program memory write latches and Flash program memory.

## 10.2 Flash Program Memory Overview

It is important to understand the Flash program memory structure for erase and programming operations. Flash program memory is arranged in rows. A row consists of a fixed number of 14-bit program memory words. A row is the minimum size that can be erased by user software.

After a row has been erased, the user can reprogram all or a portion of this row. Data to be written into the program memory row is written to 14-bit wide data write latches. These write latches are not directly accessible to the user, but may be loaded via sequential writes to the PMDATH:PMDATL register pair.

| | |
|---|---|
| **Note:** | If the user wants to modify only a portion of a previously programmed row, then the contents of the entire row must be read and saved in RAM prior to the erase. Then, new data and retained data can be written into the write latches to reprogram the row of Flash program memory. However, any unprogrammed locations can be written without first erasing the row. In this case, it is not necessary to save and rewrite the other previously programmed locations. |

See Table 10-1 for Erase Row size and the number of write latches for Flash program memory.


© 2013-2015 Microchip Technology Inc.

**TABLE 10-1: FLASH MEMORY ORGANIZATION BY DEVICE**

| Device | Row Erase (words) | Write Latches (words) |
|---|---|---|
| PIC16(L)F1703 | 16 | 16 |
| PIC16(L)F1707 | | |

### 10.2.1 READING THE FLASH PROGRAM MEMORY

To read a program memory location, the user must:

1. Write the desired address to the PMADRH:PMADRL register pair.
2. Clear the CFGS bit of the PMCON1 register.
3. Then, set control bit RD of the PMCON1 register.

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 PMCON1,RD`" instruction to be ignored. The data is available in the very next cycle, in the PMDATH:PMDATL register pair; therefore, it can be read as two bytes in the following instructions.

PMDATH:PMDATL register pair will hold this value until another read or until it is written to by the user.

| | |
|---|---|
| **Note:** | The two instructions following a program memory read are required to be `NOP`s. This prevents the user from executing a 2-cycle instruction on the next instruction after the RD bit is set. |

**FIGURE 10-1: FLASH PROGRAM MEMORY READ FLOWCHART**

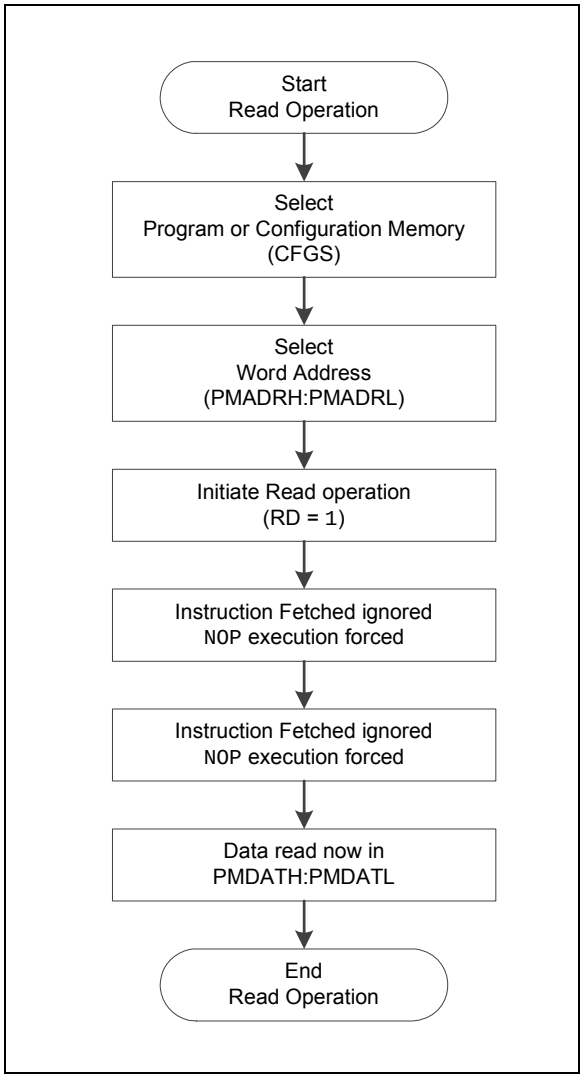
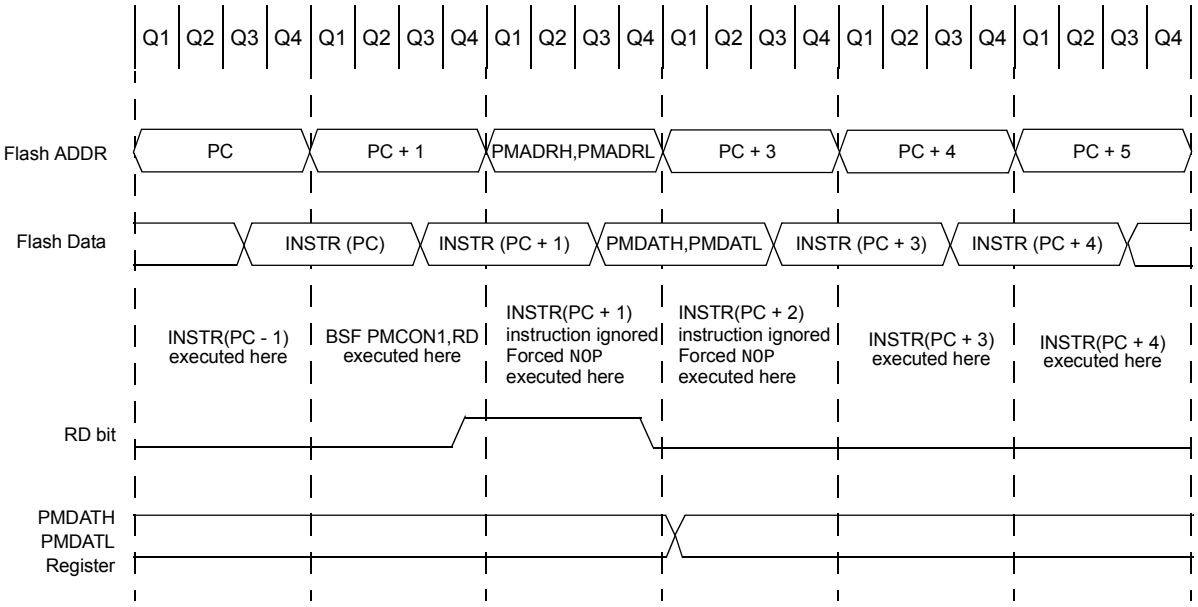


**FIGURE 10-2: FLASH PROGRAM MEMORY READ CYCLE EXECUTION**




**EXAMPLE 10-1: FLASH PROGRAM MEMORY READ**

```
* This code block will read 1 word of program
* memory at the memory address:
    PROG_ADDR_HI : PROG_ADDR_LO
*   data will be returned in the variables;
*   PROG_DATA_HI, PROG_DATA_LO

    BANKSEL  PMADRL           ; Select Bank for PMCON registers
    MOVLW    PROG_ADDR_LO     ;
    MOVWF    PMADRL           ; Store LSB of address
    MOVLW    PROG_ADDR_HI     ;
    MOVWF    PMADRH           ; Store MSB of address

    BCF      PMCON1,CFGS      ; Do not select Configuration Space
    BSF      PMCON1,RD        ; Initiate read
    NOP                       ; Ignored (Figure 10-1)
    NOP                       ; Ignored (Figure 10-1)

    MOVF     PMDATL,W         ; Get LSB of word
    MOVWF    PROG_DATA_LO     ; Store in user location
    MOVF     PMDATH,W         ; Get MSB of word
    MOVWF    PROG_DATA_HI     ; Store in user location
```


© 2013-2015 Microchip Technology Inc.

### 10.2.2 FLASH MEMORY UNLOCK SEQUENCE

The unlock sequence is a mechanism that protects the Flash program memory from unintended self-write programming or erasing. The sequence must be executed and completed without interruption to successfully complete any of the following operations:

- Row Erase
- Load program memory write latches
- Write of program memory write latches to program memory
- Write of program memory write latches to User IDs

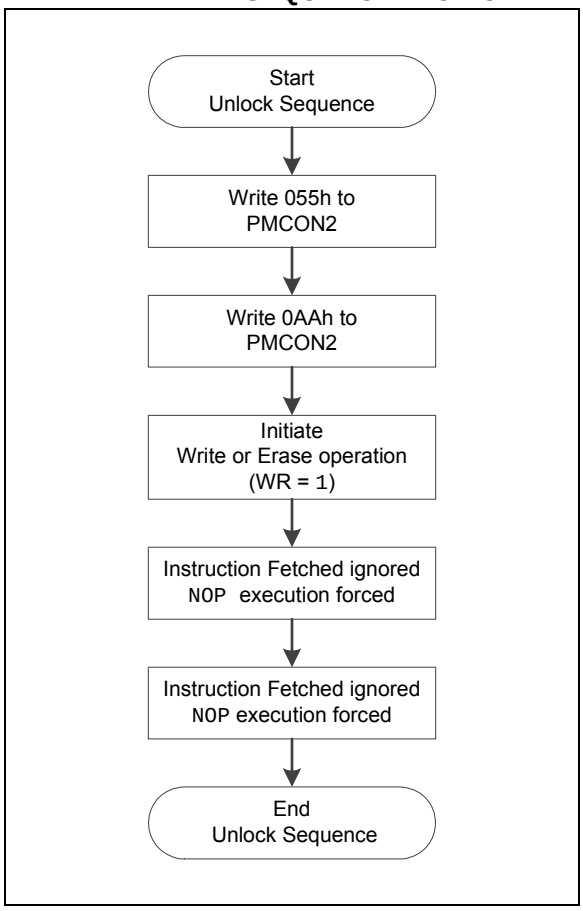
The unlock sequence consists of the following steps:

1. Write 55h to PMCON2
2. Write AAh to PMCON2
3. Set the WR bit in PMCON1
4. NOP instruction
5. NOP instruction

Once the WR bit is set, the processor will always force two NOP instructions. When an Erase Row or Program Row operation is being performed, the processor will stall internal operations (typical 2 ms), until the operation is complete and then resume with the next instruction. When the operation is loading the program memory write latches, the processor will always force the two NOP instructions and continue uninterrupted with the next instruction.

Since the unlock sequence must not be interrupted, global interrupts should be disabled prior to the unlock sequence and re-enabled after the unlock sequence is completed.

**FIGURE 10-3: FLASH PROGRAM MEMORY UNLOCK SEQUENCE FLOWCHART**



### 10.2.3 ERASING FLASH PROGRAM MEMORY

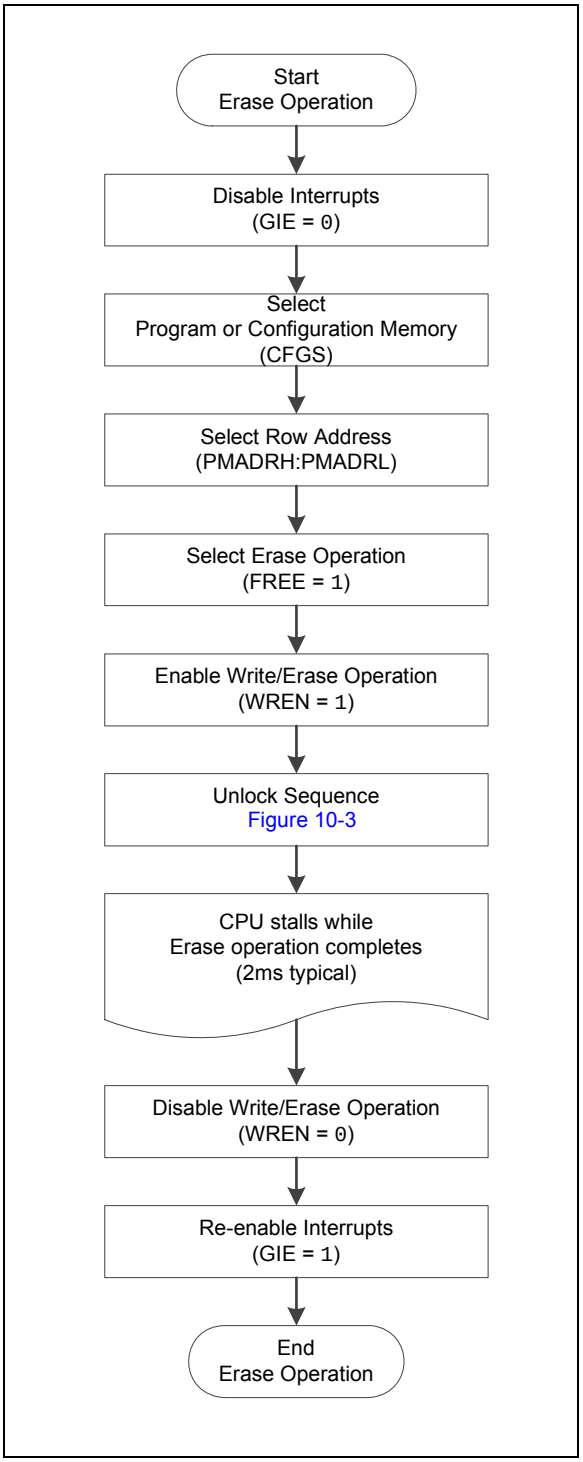
While executing code, program memory can only be erased by rows. To erase a row:

1. Load the PMADRH:PMADRL register pair with any address within the row to be erased.
2. Clear the CFGS bit of the PMCON1 register.
3. Set the FREE and WREN bits of the PMCON1 register.
4. Write 55h, then AAh, to PMCON2 (Flash programming unlock sequence).
5. Set control bit WR of the PMCON1 register to begin the erase operation.

See Example 10-2.

After the "`BSF PMCON1,WR`" instruction, the processor requires two cycles to set up the erase operation. The user must place two `NOP` instructions immediately following the WR bit set instruction. The processor will halt internal operations for the typical 2 ms erase time. This is not Sleep mode as the clocks and peripherals will continue to run. After the erase cycle, the processor will resume operation with the third instruction after the PMCON1 write instruction.

**FIGURE 10-4: FLASH PROGRAM MEMORY ERASE FLOWCHART**





© 2013-2015 Microchip Technology Inc.

**EXAMPLE 10-2: ERASING ONE ROW OF PROGRAM MEMORY**

```
; This row erase routine assumes the following:
; 1. A valid address within the erase row is loaded in ADDRH:ADDRL
; 2. ADDRH and ADDRL are located in shared data memory 0x70 - 0x7F (common RAM)


        BCF         INTCON,GIE      ; Disable ints so required sequences will execute properly
        BANKSEL     PMADRL
        MOVF        ADDRL,W         ; Load lower 8 bits of erase address boundary
        MOVWF       PMADRL
        MOVF        ADDRH,W         ; Load upper 6 bits of erase address boundary
        MOVWF       PMADRH
        BCF         PMCON1,CFGS     ; Not configuration space
        BSF         PMCON1,FREE     ; Specify an erase operation
        BSF         PMCON1,WREN     ; Enable writes

Required
Sequence
        MOVLW       55h             ; Start of required sequence to initiate erase
        MOVWF       PMCON2          ; Write 55h
        MOVLW       0AAh            ;
        MOVWF       PMCON2          ; Write AAh
        BSF         PMCON1,WR       ; Set WR bit to begin erase
        NOP                         ; NOP instructions are forced as processor starts
        NOP                         ; row erase of program memory.
                                    ;
                                    ; The processor stalls until the erase process is complete
                                    ; after erase processor continues with 3rd instruction

        BCF         PMCON1,WREN     ; Disable writes
        BSF         INTCON,GIE      ; Enable interrupts
```

### 10.2.4 WRITING TO FLASH PROGRAM MEMORY

Program memory is programmed using the following steps:

1. Load the address in PMADRH:PMADRL of the row to be programmed.
2. Load each write latch with data.
3. Initiate a programming operation.
4. Repeat steps 1 through 3 until all data is written.

Before writing to program memory, the word(s) to be written must be erased or previously unwritten. Program memory can only be erased one row at a time. No automatic erase occurs upon the initiation of the write.

Program memory can be written one or more words at a time. The maximum number of words written at one time is equal to the number of write latches. See Figure 10-5 (row writes to program memory with 16 write latches) for more details.

The write latches are aligned to the Flash row address boundary defined by the upper 10-bits of PMADRH:PMADRL, (PMADRH<6:0>:PMADRL<7:5>) with the lower five bits of PMADRL, (PMADRL<4:0>) determining the write latch being loaded. Write operations do not cross these boundaries. At the completion of a program memory write operation, the data in the write latches is reset to contain 0x3FFF.

The following steps should be completed to load the write latches and program a row of program memory. These steps are divided into two parts. First, each write latch is loaded with data from the PMDATH:PMDATL using the unlock sequence with LWLO = 1. When the last word to be loaded into the write latch is ready, the LWLO bit is cleared and the unlock sequence executed. This initiates the programming operation, writing all the latches into Flash program memory.

| | |
|---|---|
| **Note:** | The special unlock sequence is required to load a write latch with data or initiate a Flash programming operation. If the unlock sequence is interrupted, writing to the latches or program memory will not be initiated. |

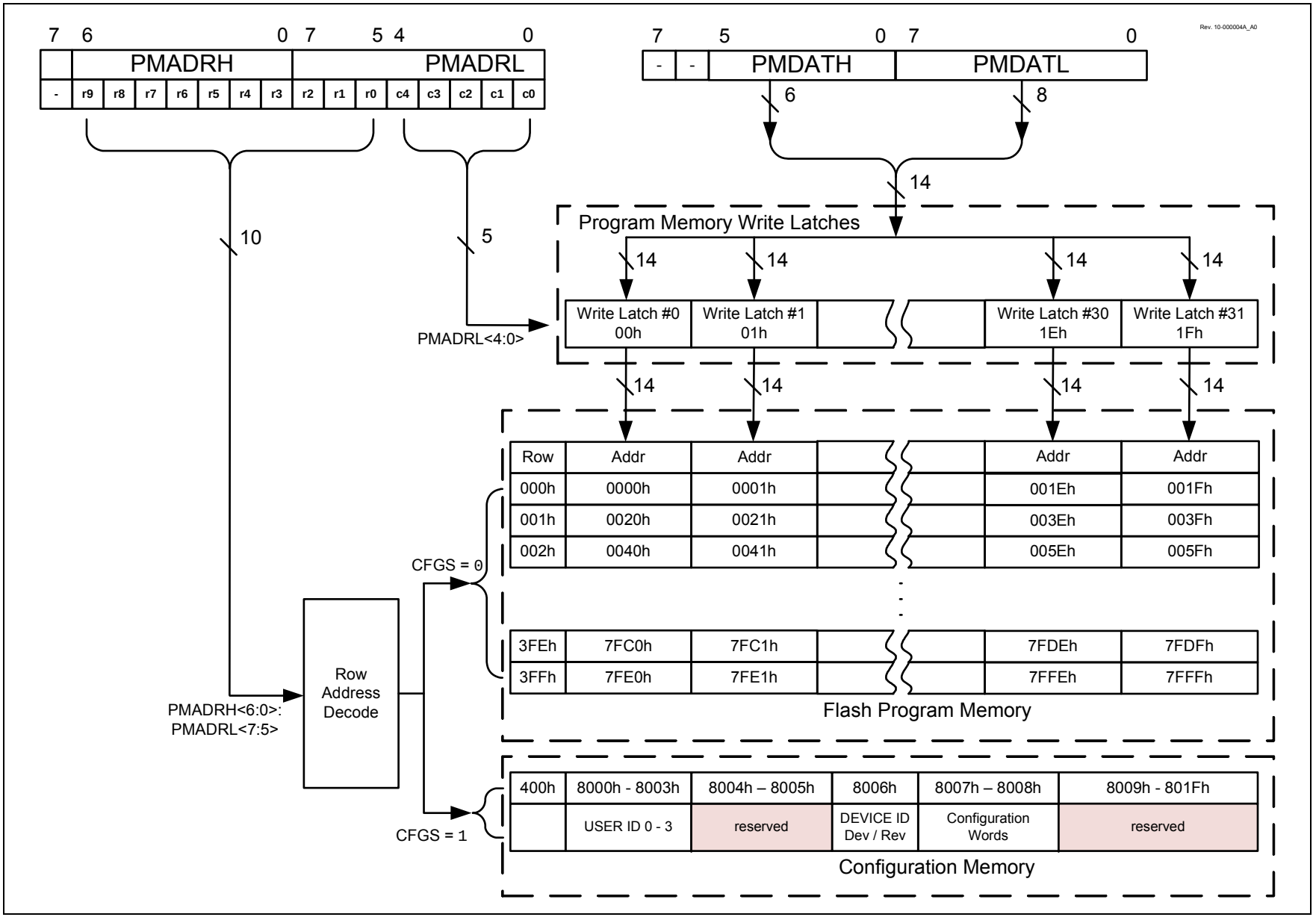
1. Set the WREN bit of the PMCON1 register.
2. Clear the CFGS bit of the PMCON1 register.
3. Set the LWLO bit of the PMCON1 register. When the LWLO bit of the PMCON1 register is '1', the write sequence will only load the write latches and will not initiate the write to Flash program memory.
4. Load the PMADRH:PMADRL register pair with the address of the location to be written.
5. Load the PMDATH:PMDATL register pair with the program memory data to be written.
6. Execute the unlock sequence (**Section 10.2.2 "Flash Memory Unlock Sequence"**). The write latch is now loaded.
7. Increment the PMADRH:PMADRL register pair to point to the next location.
8. Repeat steps 5 through 7 until all but the last write latch has been loaded.
9. Clear the LWLO bit of the PMCON1 register. When the LWLO bit of the PMCON1 register is '0', the write sequence will initiate the write to Flash program memory.
10. Load the PMDATH:PMDATL register pair with the program memory data to be written.
11. Execute the unlock sequence (**Section 10.2.2 "Flash Memory Unlock Sequence"**). The entire program memory latch content is now written to Flash program memory.

| | |
|---|---|
| **Note:** | The program memory write latches are reset to the blank state (0x3FFF) at the completion of every write or erase operation. As a result, it is not necessary to load all the program memory write latches. Unloaded latches will remain in the blank state. |

An example of the complete write sequence is shown in Example 10-3. The initial address is loaded into the PMADRH:PMADRL register pair; the data is loaded using indirect addressing.


© 2013-2015 Microchip Technology Inc.

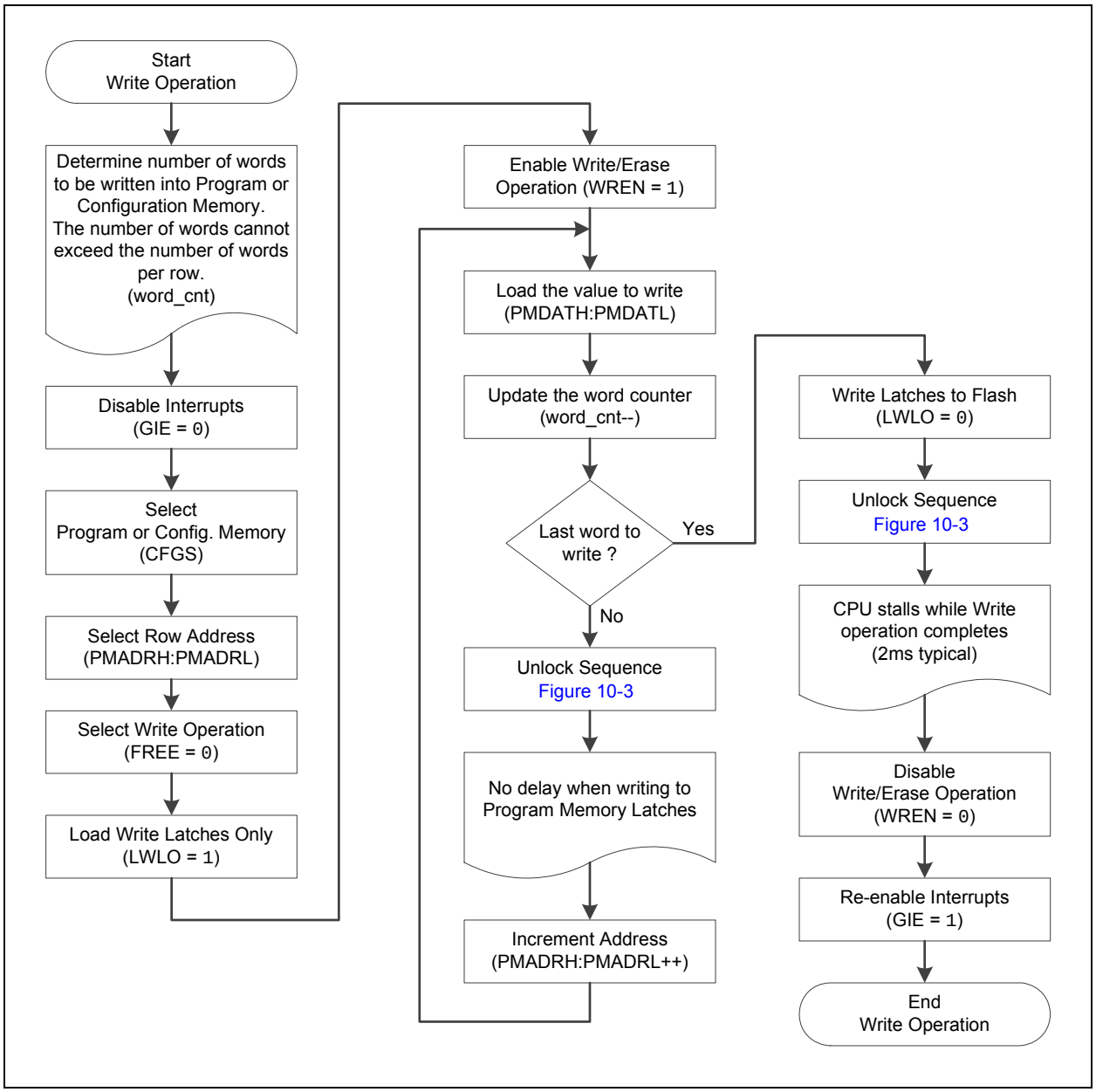
**FIGURE 10-5: BLOCK WRITES TO FLASH PROGRAM MEMORY WITH 16 WRITE LATCHES**





© 2013-2015 Microchip Technology Inc.

**FIGURE 10-6: FLASH PROGRAM MEMORY WRITE FLOWCHART**





© 2013-2015 Microchip Technology Inc.

**EXAMPLE 10-3: WRITING TO FLASH PROGRAM MEMORY**

```
; This write routine assumes the following:
; 1. 64 bytes of data are loaded, starting at the address in DATA_ADDR
; 2. Each word of data to be written is made up of two adjacent bytes in DATA_ADDR,
; stored in little endian format
; 3. A valid starting address (the least significant bits = 00000) is loaded in ADDRH:ADDRL
; 4. ADDRH and ADDRL are located in shared data memory 0x70 - 0x7F (common RAM)
;
        BCF         INTCON,GIE      ; Disable ints so required sequences will execute properly
        BANKSEL     PMADRH          ; Bank 3
        MOVF        ADDRH,W         ; Load initial address
        MOVWF       PMADRH          ;
        MOVF        ADDRL,W         ;
        MOVWF       PMADRL          ;
        MOVLW       LOW DATA_ADDR   ; Load initial data address
        MOVWF       FSR0L           ;
        MOVLW       HIGH DATA_ADDR  ; Load initial data address
        MOVWF       FSR0H           ;
        BCF         PMCON1,CFGS     ; Not configuration space
        BSF         PMCON1,WREN     ; Enable writes
        BSF         PMCON1,LWLO     ; Only Load Write Latches
LOOP
        MOVIW       FSR0++          ; Load first data byte into lower
        MOVWF       PMDATL          ;
        MOVIW       FSR0++          ; Load second data byte into upper
        MOVWF       PMDATH          ;

        MOVF        PMADRL,W        ; Check if lower bits of address are '00000'
        XORLW       0x1F            ; Check if we're on the last of 32 addresses
        ANDLW       0x1F            ;
        BTFSC       STATUS,Z        ; Exit if last of 32 words,
        GOTO        START_WRITE     ;

  Required Sequence
        MOVLW       55h             ; Start of required write sequence:
        MOVWF       PMCON2          ; Write 55h
        MOVLW       0AAh            ;
        MOVWF       PMCON2          ; Write AAh
        BSF         PMCON1,WR       ; Set WR bit to begin write
        NOP                         ; NOP instructions are forced as processor
                                    ; loads program memory write latches
        NOP                         ;

        INCF        PMADRL,F        ; Still loading latches Increment address
        GOTO        LOOP            ; Write next latches

START_WRITE
        BCF         PMCON1,LWLO     ; No more loading latches - Actually start Flash program
                                    ; memory write

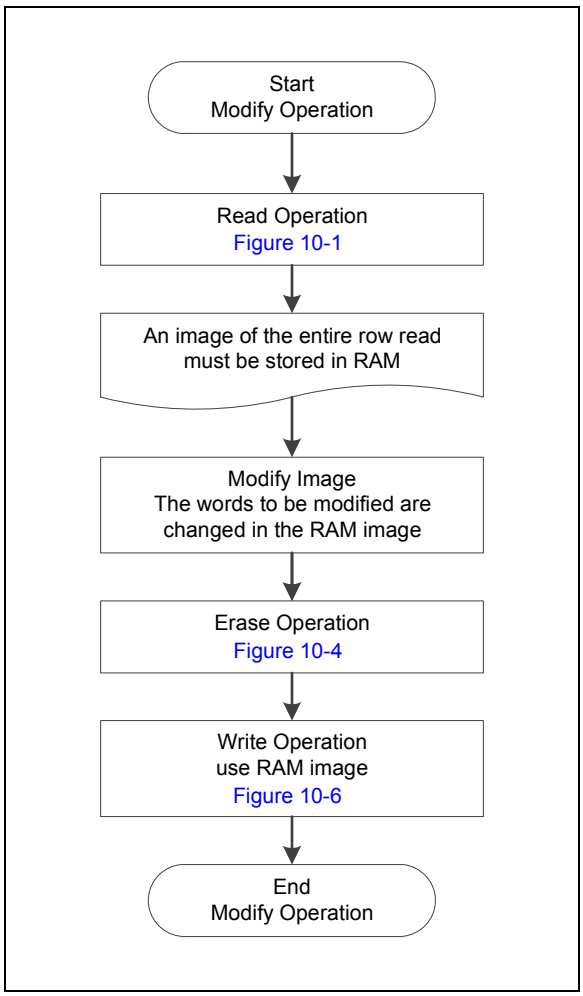
  Required Sequence
        MOVLW       55h             ; Start of required write sequence:
        MOVWF       PMCON2          ; Write 55h
        MOVLW       0AAh            ;
        MOVWF       PMCON2          ; Write AAh
        BSF         PMCON1,WR       ; Set WR bit to begin write
        NOP                         ; NOP instructions are forced as processor writes
                                    ; all the program memory write latches simultaneously
        NOP                         ; to program memory.
                                    ; After NOPs, the processor
                                    ; stalls until the self-write process in complete
                                    ; after write processor continues with 3rd instruction
        BCF         PMCON1,WREN     ; Disable writes
        BSF         INTCON,GIE      ; Enable interrupts
```

### 10.3 Modifying Flash Program Memory

When modifying existing data in a program memory row, and data within that row must be preserved, it must first be read and saved in a RAM image. Program memory is modified using the following steps:

1. Load the starting address of the row to be modified.
2. Read the existing data from the row into a RAM image.
3. Modify the RAM image to contain the new data to be written into program memory.
4. Load the starting address of the row to be rewritten.
5. Erase the program memory row.
6. Load the write latches with data from the RAM image.
7. Initiate a programming operation.

**FIGURE 10-7: FLASH PROGRAM MEMORY MODIFY FLOWCHART**





© 2013-2015 Microchip Technology Inc.

## 10.4 User ID, Device ID and Configuration Word Access

Instead of accessing program memory, the User ID's, Device ID/Revision ID and Configuration Words can be accessed when CFGS = 1 in the PMCON1 register. This is the region that would be pointed to by PC<15> = 1, but not all addresses are accessible. Different access may exist for reads and writes. Refer to Table 10-2.

When read access is initiated on an address outside the parameters listed in Table 10-2, the PMDATH:PMDATL register pair is cleared, reading back '0's.

**TABLE 10-2: USER ID, DEVICE ID AND CONFIGURATION WORD ACCESS (CFGS = 1)**

| Address | Function | Read Access | Write Access |
|---|---|---|---|
| 8000h-8003h | User IDs | Yes | Yes |
| 8005h-8006h | Device ID/Revision ID | Yes | No |
| 8007h-8008h | Configuration Words 1 and 2 | Yes | No |

**EXAMPLE 10-4: CONFIGURATION WORD AND DEVICE ID ACCESS**

```
* This code block will read 1 word of program memory at the memory address:
*   PROG_ADDR_LO (must be 00h-08h) data will be returned in the variables;
*   PROG_DATA_HI, PROG_DATA_LO

    BANKSEL  PMADRL           ; Select correct Bank
    MOVLW    PROG_ADDR_LO     ;
    MOVWF    PMADRL           ; Store LSB of address
    CLRF     PMADRH           ; Clear MSB of address

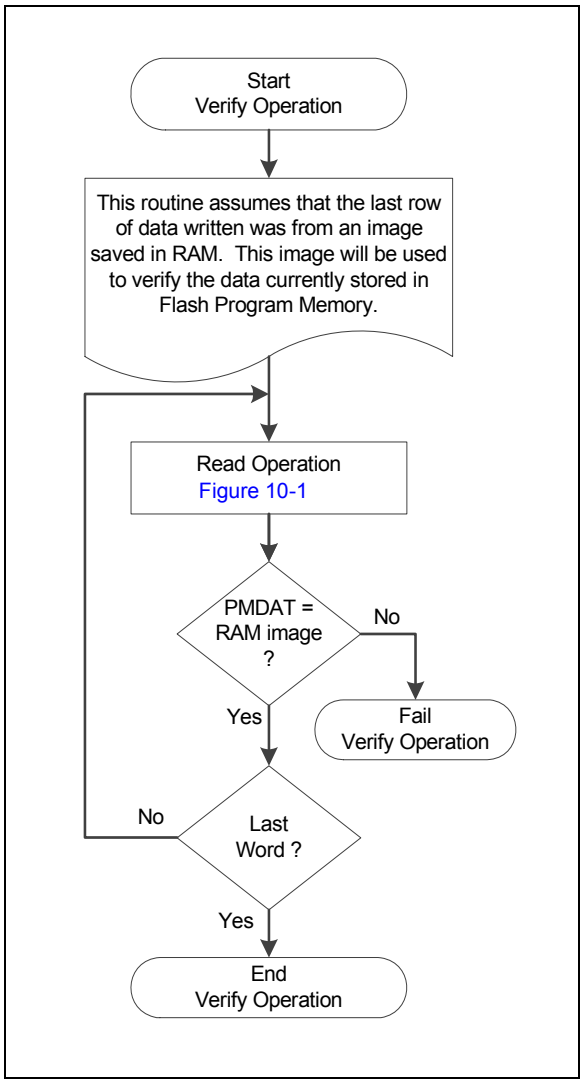
    BSF      PMCON1,CFGS      ; Select Configuration Space
    BCF      INTCON,GIE       ; Disable interrupts
    BSF      PMCON1,RD        ; Initiate read
    NOP                       ; Executed (See Figure 10-2)
    NOP                       ; Ignored (See Figure 10-2)
    BSF      INTCON,GIE       ; Restore interrupts

    MOVF     PMDATL,W         ; Get LSB of word
    MOVWF    PROG_DATA_LO     ; Store in user location
    MOVF     PMDATH,W         ; Get MSB of word
    MOVWF    PROG_DATA_HI     ; Store in user location
```

## 10.5 Write Verify

It is considered good programming practice to verify that program memory writes agree with the intended value. Since program memory is stored as a full page then the stored program memory contents are compared with the intended data stored in RAM after the last write is complete.

**FIGURE 10-8: FLASH PROGRAM MEMORY VERIFY FLOWCHART**





© 2013-2015 Microchip Technology Inc.

## 10.6 Register Definitions: Flash Program Memory Control

### REGISTER 10-1: PMDATL: PROGRAM MEMORY DATA LOW BYTE REGISTER

| R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u |
|---|---|---|---|---|---|---|---|
| PMDATL<7:0> | | | | | | | |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| '1' = Bit is set | '0' = Bit is cleared | |

bit 7-0 **PMDATL<7:0>**: Read/write value for Least Significant bits of program memory

### REGISTER 10-2: PMDATH: PROGRAM MEMORY DATA HIGH BYTE REGISTER

| U-0 | U-0 | R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u |
|---|---|---|---|---|---|---|---|
| — | — | PMDATH<13:8> | | | | | |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| '1' = Bit is set | '0' = Bit is cleared | |

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

bit 5-0 **PMDATH<13:8>**: Read/write value for Most Significant bits of program memory

### REGISTER 10-3: PMADRL: PROGRAM MEMORY ADDRESS LOW BYTE REGISTER

| R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| PMADRL<7:0> | | | | | | | |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| '1' = Bit is set | '0' = Bit is cleared | |

bit 7-0 **PMADRL<7:0>**: Specifies the Least Significant bits for program memory address

### REGISTER 10-4: PMADRH: PROGRAM MEMORY ADDRESS HIGH BYTE REGISTER

| U-1 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| —(1) | PMADRH<14:8> | | | | | | |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| '1' = Bit is set | '0' = Bit is cleared | |

bit 7 **Unimplemented:** Read as '1'

bit 6-0 **PMADRH<14:8>**: Specifies the Most Significant bits for program memory address

**Note 1:** Unimplemented, read as '1'.

**REGISTER 10-5: PMCON1: PROGRAM MEMORY CONTROL 1 REGISTER**

| U-1 | R/W-0/0 | R/W-0/0 | R/W/HC-0/0 | R/W/HC-x/q(2) | R/W-0/0 | R/S/HC-0/0 | R/S/HC-0/0 |
|---|---|---|---|---|---|---|---|
| —(1) | CFGS | LWLO(3) | FREE | WRERR | WREN | WR | RD |
| bit 7 | | | | | | | bit 0 |

| Legend: | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| S = Bit can only be set | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| '1' = Bit is set | '0' = Bit is cleared | HC = Bit is cleared by hardware |

bit 7 **Unimplemented:** Read as '1'

bit 6 **CFGS:** Configuration Select bit
- 1 = Access Configuration, User ID and Device ID Registers
- 0 = Access Flash program memory

bit 5 **LWLO:** Load Write Latches Only bit(3)
- 1 = Only the addressed program memory write latch is loaded/updated on the next WR command
- 0 = The addressed program memory write latch is loaded/updated and a write of all program memory write latches will be initiated on the next WR command

bit 4 **FREE:** Program Flash Erase Enable bit
- 1 = Performs an erase operation on the next WR command (hardware cleared upon completion)
- 0 = Performs a write operation on the next WR command

bit 3 **WRERR:** Program/Erase Error Flag bit
- 1 = Condition indicates an improper program or erase sequence attempt or termination (bit is set automatically on any set attempt (write '1') of the WR bit).
- 0 = The program or erase operation completed normally

bit 2 **WREN:** Program/Erase Enable bit
- 1 = Allows program/erase cycles
- 0 = Inhibits programming/erasing of program Flash

bit 1 **WR:** Write Control bit
- 1 = Initiates a program Flash program/erase operation.
  The operation is self-timed and the bit is cleared by hardware once operation is complete.
  The WR bit can only be set (not cleared) in software.
- 0 = Program/erase operation to the Flash is complete and inactive

bit 0 **RD:** Read Control bit
- 1 = Initiates a program Flash read. Read takes one cycle. RD is cleared in hardware. The RD bit can only be set (not cleared) in software.
- 0 = Does not initiate a program Flash read

**Note 1:** Unimplemented bit, read as '1'.
**2:** The WRERR bit is automatically set by hardware when a program memory write or erase operation is started (WR = 1).
**3:** The LWLO bit is ignored during a program memory erase operation (FREE = 1).


© 2013-2015 Microchip Technology Inc.

**REGISTER 10-6: PMCON2: PROGRAM MEMORY CONTROL 2 REGISTER**

| W-0/0 | W-0/0 | W-0/0 | W-0/0 | W-0/0 | W-0/0 | W-0/0 | W-0/0 |
|---|---|---|---|---|---|---|---|
| PMCON2<7:0> | | | | | | | |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| S = Bit can only be set | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| '1' = Bit is set | '0' = Bit is cleared | |

bit 7-0 **Flash Memory Unlock Pattern bits**

To unlock writes, a 55h must be written first, followed by an AAh, before setting the WR bit of the PMCON1 register. The value written to this register is used to unlock the writes. There are specific timing requirements on these writes.

**TABLE 10-3: SUMMARY OF REGISTERS ASSOCIATED WITH FLASH PROGRAM MEMORY**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|
| INTCON | GIE | PEIE | TMR0IE | INTE | IOCIE | TMR0IF | INTF | IOCIF | 77 |
| PMCON1 | —(1) | CFGS | LWLO | FREE | WRERR | WREN | WR | RD | 108 |
| PMCON2 | PMCON2<7:0> | | | | | | | | 109 |
| PMADRL | PMADR | | | | | | | | 107 |
| PMADRH | —(1) | PMADR<14:8> | | | | | | | 107 |
| PMDATL | PMDAT | | | | | | | | 107 |
| PMDATH | — | — | PMDAT<13:8> | | | | | | 107 |

**Legend:** — = unimplemented location, read as '0'. Shaded cells are not used by Flash program memory.
**Note 1:** Unimplemented, read as '1'.

**TABLE 10-4: SUMMARY OF CONFIGURATION BITS ASSOCIATED WITH FLASH PROGRAM MEMORY**

| Name | Bits | Bit -/7 | Bit -/6 | Bit 13/5 | Bit 12/4 | Bit 11/3 | Bit 10/2 | Bit 9/1 | Bit 8/0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|---|
| CONFIG1 | 13:8 | — | — | — | — | $\overline{\text{CLKOUTEN}}$ | BOREN<1:0> | | — | 49 |
| | 7:0 | $\overline{\text{CP}}$ | MCLRE | $\overline{\text{PWRTE}}$ | WDTE<1:0> | | — | FOSC<1:0> | | |
| CONFIG2 | 13:8 | — | — | LVP | $\overline{\text{DEBUG}}$ | $\overline{\text{LPBOR}}$ | BORV | STVREN | PLLEN | 50 |
| | 7:0 | ZCDDIS | — | — | — | — | PPS1WAY | WRT<1:0> | | |

**Legend:** — = unimplemented location, read as '0'. Shaded cells are not used by Flash program memory.

## 11.0 I/O PORTS

Each port has six standard registers for its operation. These registers are:

- TRISx registers (data direction)
- PORTx registers (reads the levels on the pins of the device)
- LATx registers (output latch)
- INLVLx (input level control)
- ODCONx registers (open drain)
- SLRCONx registers (slew rate

Some ports may have one or more of the following additional registers. These registers are:

- ANSELx (analog select)
- WPUx (weak pull-up)

In general, when a peripheral is enabled on a port pin, that pin cannot be used as a general purpose output. However, the pin can still be read.

**TABLE 11-1: PORT AVAILABILITY PER DEVICE**

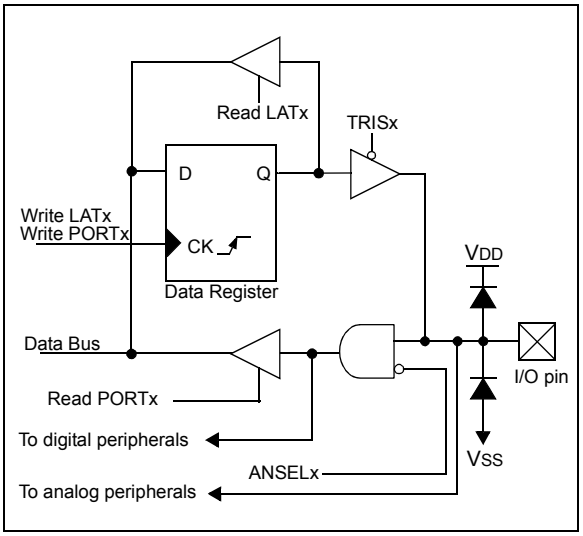
| Device | PORTA | PORTB | PORTC |
|---|---|---|---|
| PIC16(L)F1703 | • | | • |
| PIC16(L)F1707 | • | • | • |

The Data Latch (LATx registers) is useful for read-modify-write operations on the value that the I/O pins are driving.

A write operation to the LATx register has the same effect as a write to the corresponding PORTx register. A read of the LATx register reads of the values held in the I/O PORT latches, while a read of the PORTx register reads the actual I/O pin value.

Ports that support analog inputs have an associated ANSELx register. When an ANSEL bit is set, the digital input buffer associated with that bit is disabled. Disabling the input buffer prevents analog signal levels on the pin between a logic high and low from causing excessive current in the logic input circuitry. A simplified model of a generic I/O port, without the interfaces to other peripherals, is shown in Figure 11-1.

**FIGURE 11-1: GENERIC I/O PORT OPERATION**





© 2013-2015 Microchip Technology Inc.

## 11.1 PORTA Registers

### 11.1.1 DATA REGISTER

PORTA is a 6-bit wide, bidirectional port. The corresponding data direction register is TRISA (Register 11-2). Setting a TRISA bit (= 1) will make the corresponding PORTA pin an input (i.e., disable the output driver). Clearing a TRISA bit (= 0) will make the corresponding PORTA pin an output (i.e., enables output driver and puts 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 11-1 shows how to initialize PORTA.

Reading the PORTA register (Register 11-1) 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 (LATA).

### 11.1.2 DIRECTION CONTROL

The TRISA register (Register 11-2) controls the PORTA pin output drivers, even when they are being used as analog inputs. The user should ensure the bits in the TRISA register are maintained set when using them as analog inputs. I/O pins configured as analog inputs always read '0'.

### 11.1.3 OPEN DRAIN CONTROL

The ODCONA register (Register 11-6) controls the open-drain feature of the port. Open drain operation is independently selected for each pin. When an ODCONA bit is set, the corresponding port output becomes an open drain driver capable of sinking current only. When an ODCONA bit is cleared, the corresponding port output pin is the standard push-pull drive capable of sourcing and sinking current.

### 11.1.4 SLEW RATE CONTROL

The SLRCONA register (Register 11-7) controls the slew rate option for each port pin. Slew rate control is independently selectable for each port pin. When an SLRCONA bit is set, the corresponding port pin drive is slew rate limited. When an SLRCONA bit is cleared, The corresponding port pin drive slews at the maximum rate possible.

### 11.1.5 INPUT THRESHOLD CONTROL

The INLVLA register (Register 11-8) controls the input voltage threshold for each of the available PORTA input pins. A selection between the Schmitt Trigger CMOS or the TTL Compatible thresholds is available. The input threshold is important in determining the value of a read of the PORTA register and also the level at which an interrupt-on-change occurs, if that feature is enabled. See Table 26-4 for more information on threshold levels.

| | |
|---|---|
| **Note:** | Changing the input threshold selection should be performed while all peripheral modules are disabled. Changing the threshold level during the time a module is active may inadvertently generate a transition associated with an input pin, regardless of the actual voltage level on that pin. |

### 11.1.6 ANALOG CONTROL

The ANSELA register (Register 11-4) is used to configure the Input mode of an I/O pin to analog. Setting the appropriate ANSELA bit high will cause all digital reads on the pin to be read as '0' and allow analog functions on the pin to operate correctly.

The state of the ANSELA bits has no effect on digital output functions. A pin with TRIS clear and ANSEL set will still operate as a digital output, but the Input mode will be analog. This can cause unexpected behavior when executing read-modify-write instructions on the affected port.

| | |
|---|---|
| **Note:** | The ANSELA bits default to the Analog mode after Reset. To use any pins as digital general purpose or peripheral inputs, the corresponding ANSEL bits must be initialized to '0' by user software. |

**EXAMPLE 11-1: INITIALIZING PORTA**

```
; This code example illustrates
; initializing the PORTA register. The
; other ports are initialized in the same
; manner.

BANKSEL  PORTA        ;
CLRF     PORTA        ;Init PORTA
BANKSEL  LATA         ;Data Latch
CLRF     LATA         ;
BANKSEL  ANSELA       ;
CLRF     ANSELA       ;digital I/O
BANKSEL  TRISA        ;
MOVLW    B'00111000'  ;Set RA<5:3> as inputs
MOVWF    TRISA        ;and set RA<2:0> as
                      ;outputs
```

### 11.1.7 PORTA FUNCTIONS AND OUTPUT PRIORITIES

Each PORTA pin is multiplexed with other functions.

Each pin defaults to the PORT latch data after reset. Other functions are selected with the peripheral pin select logic. See **Section 12.0 "Peripheral Pin Select (PPS) Module"** for more information.

Analog input functions, such as ADC inputs, are not shown in the peripheral pin select lists. These inputs are active when the I/O pin is set for Analog mode using the ANSELA register. Digital output functions may continue to control the pin when it is in Analog mode.


© 2013-2015 Microchip Technology Inc.

## 11.2 Register Definitions: PORTA

**REGISTER 11-1: PORTA: PORTA REGISTER**

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

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| '1' = Bit is set | '0' = Bit is cleared | |

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

bit 5-0 **RA<5:0>**: PORTA I/O Value bits(1)
1 = Port pin is ≥ VIH
0 = Port pin is ≤ VIL

**Note 1:** Writes to PORTA are actually written to corresponding LATA register. Reads from PORTA register is return of actual I/O pin values.

**REGISTER 11-2: TRISA: PORTA TRI-STATE REGISTER**

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

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| '1' = Bit is set | '0' = Bit is cleared | |

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

bit 5-4 **TRISA<5:4>:** PORTA Tri-State Control bit
1 = PORTA pin configured as an input (tri-stated)
0 = PORTA pin configured as an output

bit 3 **Unimplemented:** Read as '1'

bit 2-0 **TRISA<2:0>:** PORTA Tri-State Control bit
1 = PORTA pin configured as an input (tri-stated)
0 = PORTA pin configured as an output

**Note 1:** Unimplemented, read as '1'.

**REGISTER 11-3: LATA: PORTA DATA LATCH REGISTER**

| U-0 | U-0 | R/W-x/u | R/W-x/u | U-0 | R/W-x/u | R/W-x/u | R/W-x/u |
|---|---|---|---|---|---|---|---|
| — | — | LATA5 | LATA4 | — | LATA2 | LATA1 | LATA0 |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| '1' = Bit is set | '0' = Bit is cleared | |

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

bit 5-4 **LATA<5:4>**: RA<5:4> Output Latch Value bits(1)

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

bit 2-0 **LATA<2:0>**: RA<2:0> Output Latch Value bits(1)

**Note 1:** Writes to PORTA are actually written to corresponding LATA register. Reads from PORTA register is return of actual I/O pin values.

**REGISTER 11-4: ANSELA: PORTA ANALOG SELECT REGISTER**

| U-0 | U-0 | U-0 | R/W-1/1 | U-0 | R/W-1/1 | R/W-1/1 | R/W-1/1 |
|---|---|---|---|---|---|---|---|
| — | — | — | ANSA4 | — | ANSA2 | ANSA1 | ANSA0 |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| '1' = Bit is set | '0' = Bit is cleared | |

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

bit 4 **ANSA4**: Analog Select between Analog or Digital Function on pin RA4
1 = Analog input. Pin is assigned as analog input(1). Digital input buffer disabled.
0 = Digital I/O. Pin is assigned to port or digital special function.

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

bit 2-0 **ANSA<2:0>**: Analog Select between Analog or Digital Function on pins RA<2:0>, respectively
1 = Analog input. Pin is assigned as analog input(1). Digital input buffer disabled.
0 = Digital I/O. Pin is assigned to port or digital special function.

**Note 1:** When setting a pin to an analog input, the corresponding TRIS bit must be set to Input mode in order to allow external control of the voltage on the pin.


© 2013-2015 Microchip Technology Inc.

**REGISTER 11-5: WPUA: WEAK PULL-UP PORTA REGISTER**

| U-0 | U-0 | R/W-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 |
|---|---|---|---|---|---|---|---|
| — | — | WPUA5 | WPUA4 | WPUA3 | WPUA2 | WPUA1 | WPUA0 |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| '1' = Bit is set | '0' = Bit is cleared | |

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

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

**Note 1:** Global $\overline{\text{WPUEN}}$ bit of the OPTION_REG register must be cleared for individual pull-ups to be enabled.
**2:** The weak pull-up device is automatically disabled if the pin is configured as an output.

**REGISTER 11-6: ODCONA: PORTA OPEN DRAIN CONTROL REGISTER**

| U-0 | U-0 | R/W-0/0 | R/W-0/0 | U-0 | R/W-0/0 | R/W-0/0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| — | — | ODA5 | ODA4 | — | ODA2 | ODA1 | ODA0 |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| '1' = Bit is set | '0' = Bit is cleared | |

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

bit 5-4 **ODA<5:4>:** PORTA Open Drain Enable bits
For RA<5:4> pins, respectively
1 = Port pin operates as open-drain drive (sink current only)
0 = Port pin operates as standard push-pull drive (source and sink current)

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

bit 2-0 **ODA<2:0>:** PORTA Open Drain Enable bits
For RA<2:0> pins, respectively
1 = Port pin operates as open-drain drive (sink current only)
0 = Port pin operates as standard push-pull drive (source and sink current)

**REGISTER 11-7: SLRCONA: PORTA SLEW RATE CONTROL REGISTER**

| U-0 | U-0 | R/W-1/1 | R/W-1/1 | U-0 | R/W-1/1 | R/W-1/1 | R/W-1/1 |
|---|---|---|---|---|---|---|---|
| — | — | SLRA5 | SLRA4 | — | SLRA2 | SLRA1 | SLRA0 |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as ‘0’ |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| ‘1’ = Bit is set | ‘0’ = Bit is cleared | |

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5-4 **SLRA<5:4>:** PORTA Slew Rate Enable bits
For RA<5:4> pins, respectively
1 = Port pin slew rate is limited
0 = Port pin slews at maximum rate

bit 3 **Unimplemented:** Read as ‘0’

bit 2-0 **SLRA<2:0>:** PORTA Slew Rate Enable bits
For RA<2:0> pins, respectively
1 = Port pin slew rate is limited
0 = Port pin slews at maximum rate

**REGISTER 11-8: INLVLA: PORTA INPUT LEVEL CONTROL REGISTER**

| U-0 | U-0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| — | — | INLVLA5 | INLVLA4 | INLVLA3 | INLVLA2 | INLVLA1 | INLVLA0 |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as ‘0’ |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| ‘1’ = Bit is set | ‘0’ = Bit is cleared | |

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5-0 **INLVLA<5:0>:** PORTA Input Level Select bits
For RA<5:0> pins, respectively
1 = ST input used for PORT reads and interrupt-on-change
0 = TTL input used for PORT reads and interrupt-on-change


© 2013-2015 Microchip Technology Inc.

**TABLE 11-2: SUMMARY OF REGISTERS ASSOCIATED WITH PORTA**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|
| ANSELA | — | — | — | ANSA4 | — | ANSA2 | ANSA1 | ANSA0 | 114 |
| INLVLA | — | — | INLVLA5 | INLVLA4 | INLVLA3 | INLVLA2 | INLVLA1 | INLVLA0 | 116 |
| LATA | — | — | LATA5 | LATA4 | — | LATA2 | LATA1 | LATA0 | 114 |
| ODCONA | — | — | ODA5 | ODA4 | — | ODA2 | ODA1 | ODA0 | 115 |
| OPTION_REG | WPUEN | INTEDG | TMR0CS | TMR0SE | PSA | PS<2:0> | | | 168 |
| PORTA | — | — | RA5 | RA4 | RA3 | RA2 | RA1 | RA0 | 113 |
| SLRCONA | — | — | SLRA5 | SLRA4 | — | SLRA2 | SLRA1 | SLRA0 | 116 |
| TRISA | — | — | TRISA5 | TRISA4 | —(1) | TRISA2 | TRISA1 | TRISA0 | 113 |
| WPUA | — | — | WPUA5 | WPUA4 | WPUA3 | WPUA2 | WPUA1 | WPUA0 | 115 |

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

**Note 1:** Unimplemented, read as '1'.

**TABLE 11-3: SUMMARY OF CONFIGURATION BITS ASSOCIATED WITH PORTA**

| Name | Bits | Bit -/7 | Bit -/6 | Bit 13/5 | Bit 12/4 | Bit 11/3 | Bit 10/2 | Bit 9/1 | Bit 8/0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|---|
| CONFIG1 | 13:8 | — | — | — | — | $\overline{\text{CLKOUTEN}}$ | BOREN<1:0> | | — | 49 |
| | 7:0 | $\overline{\text{CP}}$ | MCLRE | $\overline{\text{PWRTE}}$ | WDTE<1:0> | | — | FOSC<1:0> | | |
| CONFIG2 | 13:8 | — | — | LVP | $\overline{\text{DEBUG}}$ | $\overline{\text{LPBOR}}$ | BORV | STVREN | PLLEN | 50 |
| | 7:0 | ZCDDIS | — | — | — | — | PPS1WAY | WRT<1:0> | | |

**Legend:** — = unimplemented location, read as '0'. Shaded cells are not used by PORTA.

## 11.3 PORTB Registers (PIC16(L)F1707 only)

PORTB is a 4-bit wide, bidirectional port. The corresponding data direction register is TRISB (Register 11-10). 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., enable the output driver and put the contents of the output latch on the selected pin). Example 11-1 shows how to initialize an I/O port.

Reading the PORTB register (Register 11-9) 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 (LATB).

### 11.3.1 DIRECTION CONTROL

The TRISB register (Register 11-10) controls the PORTB pin output drivers, even when they are being used as analog inputs. The user should ensure the bits in the TRISB register are maintained set when using them as analog inputs. I/O pins configured as analog inputs always read '0'.

### 11.3.2 OPEN DRAIN CONTROL

The ODCONB register (Register 11-14) controls the open-drain feature of the port. Open drain operation is independently selected for each pin. When an ODCONB bit is set, the corresponding port output becomes an open drain driver capable of sinking current only. When an ODCONB bit is cleared, the corresponding port output pin is the standard push-pull drive capable of sourcing and sinking current.

### 11.3.3 SLEW RATE CONTROL

The SLRCONB register (Register 11-15) controls the slew rate option for each port pin. Slew rate control is independently selectable for each port pin. When an SLRCONB bit is set, the corresponding port pin drive is slew rate limited. When an SLRCONB bit is cleared, The corresponding port pin drive slews at the maximum rate possible.

### 11.3.4 INPUT THRESHOLD CONTROL

The INLVLB register (Register 11-16) controls the input voltage threshold for each of the available PORTB input pins. A selection between the Schmitt Trigger CMOS or the TTL Compatible thresholds is available. The input threshold is important in determining the value of a read of the PORTB register and also the level at which an interrupt-on-change occurs, if that feature is enabled. See Table 26-4 for more information on threshold levels.

| | |
|---|---|
| **Note:** | Changing the input threshold selection should be performed while all peripheral modules are disabled. Changing the threshold level during the time a module is active may inadvertently generate a transition associated with an input pin, regardless of the actual voltage level on that pin. |

### 11.3.5 ANALOG CONTROL

The ANSELB register (Register 11-12) is used to configure the Input mode of an I/O pin to analog. Setting the appropriate ANSELB bit high will cause all digital reads on the pin to be read as '0' and allow analog functions on the pin to operate correctly.

The state of the ANSELB bits has no effect on digital output functions. A pin with TRIS clear and ANSELB set will still operate as a digital output, but the Input mode will be analog. This can cause unexpected behavior when executing read-modify-write instructions on the affected port.

| | |
|---|---|
| **Note:** | The ANSELB bits default to the Analog mode after Reset. To use any pins as digital general purpose or peripheral inputs, the corresponding ANSEL bits must be initialized to '0' by user software. |

### 11.3.6 PORTB FUNCTIONS AND OUTPUT PRIORITIES

Each pin defaults to the PORT latch data after reset. Other functions are selected with the peripheral pin select logic. See **Section 12.0 "Peripheral Pin Select (PPS) Module"** for more information. Analog input functions, such as ADC and Op Amp inputs, are not shown in the peripheral pin select lists. These inputs are active when the I/O pin is set for Analog mode using the ANSELB register. Digital output functions may continue to control the pin when it is in Analog mode.


© 2013-2015 Microchip Technology Inc.

## 11.4 Register Definitions: PORTB

### REGISTER 11-9: PORTB: PORTB REGISTER

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

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as ‘0’ |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| ‘1’ = Bit is set | ‘0’ = Bit is cleared | |

bit 7-4 **RB<7:4>**: PORTB General Purpose I/O Pin bits[(1)]
1 = Port pin is ≥ VIH
0 = Port pin is ≤ VIL

bit 3-0 **Unimplemented:** Read as ‘0’

**Note 1:** Writes to PORTB are actually written to corresponding LATB register. Reads from PORTB register is return of actual I/O pin values.

### REGISTER 11-10: TRISB: PORTB TRI-STATE REGISTER

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

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as ‘0’ |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| ‘1’ = Bit is set | ‘0’ = Bit is cleared | |

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’

### REGISTER 11-11: LATB: PORTB DATA LATCH REGISTER

| R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u | U-0 | U-0 | U-0 | U-0 |
|---|---|---|---|---|---|---|---|
| LATB7 | LATB6 | LATB5 | LATB4 | — | — | — | — |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as ‘0’ |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| ‘1’ = Bit is set | ‘0’ = Bit is cleared | |

bit 7-4 **LATB<7:4>**: PORTB Output Latch Value bits[(1)]

bit 3-0 **Unimplemented:** Read as ‘0’

**Note 1:** Writes to PORTB are actually written to corresponding LATB register. Reads from PORTB register is return of actual I/O pin values.

**REGISTER 11-12: ANSELB: PORTB ANALOG SELECT REGISTER**

| U-0 | U-0 | R/W-1/1 | R/W-1/1 | U-0 | U-0 | U-0 | U-0 |
|---|---|---|---|---|---|---|---|
| — | — | ANSB5 | ANSB4 | — | — | — | — |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| '1' = Bit is set | '0' = Bit is cleared | |

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

bit 5-4 **ANSB<5:4>**: Analog Select between Analog or Digital Function on pins RB<5:4>, respectively
0 = Digital I/O. Pin is assigned to port or digital special function.
1 = Analog input. Pin is assigned as analog input(1). Digital input buffer disabled.

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

**Note 1:** When setting a pin to an analog input, the corresponding TRIS bit must be set to Input mode in order to allow external control of the voltage on the pin.

**REGISTER 11-13: WPUB: WEAK PULL-UP PORTB REGISTER**

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

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| '1' = Bit is set | '0' = Bit is cleared | |

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 $\overline{\text{WPUEN}}$ bit of the OPTION_REG register must be cleared for individual pull-ups to be enabled.
**2:** The weak pull-up device is automatically disabled if the pin is configured as an output.


© 2013-2015 Microchip Technology Inc.

**REGISTER 11-14: ODCONB: PORTB OPEN DRAIN CONTROL REGISTER**

| R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | U-0 | U-0 | U-0 | U-0 |
|---|---|---|---|---|---|---|---|
| ODB7 | ODB6 | ODB5 | ODB4 | — | — | — | — |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| '1' = Bit is set | '0' = Bit is cleared | |

bit 7-4 **ODB<7:4>:** PORTB Open Drain Enable bits
For RB<7:4> pins, respectively
1 = Port pin operates as open-drain drive (sink current only)
0 = Port pin operates as standard push-pull drive (source and sink current)

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

**REGISTER 11-15: SLRCONB: PORTB SLEW RATE CONTROL REGISTER**

| R/W-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 | U-0 | U-0 | U-0 | U-0 |
|---|---|---|---|---|---|---|---|
| SLRB7 | SLRB6 | SLRB5 | SLRB4 | — | — | — | — |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| '1' = Bit is set | '0' = Bit is cleared | |

bit 7-4 **SLRB<7:4>:** PORTB Slew Rate Enable bits
For RB<7:4> pins, respectively
1 = Port pin slew rate is limited
0 = Port pin slews at maximum rate

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

**REGISTER 11-16: INLVLB: PORTB INPUT LEVEL CONTROL REGISTER**

| R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | U-0 | U-0 | U-0 | U-0 |
|---|---|---|---|---|---|---|---|
| INLVLB7 | INLVLB6 | INLVLB5 | INLVLB4 | — | — | — | — |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| '1' = Bit is set | '0' = Bit is cleared | |

bit 7-4 **INLVLB<7:4>:** PORTB Input Level Select bits
For RB<7:4> pins, respectively
1 = ST input used for PORT reads and interrupt-on-change
0 = TTL input used for PORT reads and interrupt-on-change

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

**TABLE 11-4: SUMMARY OF REGISTERS ASSOCIATED WITH PORTB**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|
| ANSELB | — | — | ANSB5 | ANSB4 | — | — | — | — | 120 |
| INLVLB | INLVLB7 | INLVLB6 | INLVLB5 | INLVLB4 | — | — | — | — | 121 |
| LATB | LATB7 | LATB6 | LATB5 | LATB4 | — | — | — | — | 119 |
| ODCONB | ODB7 | ODB6 | ODB5 | ODB4 | — | — | — | — | 121 |
| PORTB | RB7 | RB6 | RB5 | RB4 | — | — | — | — | 119 |
| SLRCONB | SLRB7 | SLRB6 | SLRB5 | SLRB4 | — | — | — | — | 121 |
| TRISB | TRISB7 | TRISB6 | TRISB5 | TRISB4 | — | — | — | — | 121 |
| WPUB | WPUB7 | WPUB6 | WPUB5 | WPUB4 | — | — | — | — | 120 |

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


© 2013-2015 Microchip Technology Inc.

## 11.5 PORTC Registers

### 11.5.1 DATA REGISTER

PORTC is a 6-bit wide bidirectional port in the PIC16(L)F1703 device and 8-bit wide bidirectional port in the PIC16(L)F1707 device. The corresponding data direction register is TRISC (Register 11-18). 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., enable the output driver and put the contents of the output latch on the selected pin). Example 11-1 shows how to initialize an I/O port.

Reading the PORTC register (Register 11-17) 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 (LATC).

### 11.5.2 DIRECTION CONTROL

The TRISC register (Register 11-18) controls the PORTC pin output drivers, even when they are being used as analog inputs. The user should ensure the bits in the TRISC register are maintained set when using them as analog inputs. I/O pins configured as analog inputs always read '0'.

### 11.5.3 INPUT THRESHOLD CONTROL

The INLVLC register (Register 11-24) controls the input voltage threshold for each of the available PORTC input pins. A selection between the Schmitt Trigger CMOS or the TTL Compatible thresholds is available. The input threshold is important in determining the value of a read of the PORTC register and also the level at which an interrupt-on-change occurs, if that feature is enabled. See Table 26-4 for more information on threshold levels.

| | |
|---|---|
| **Note:** | Changing the input threshold selection should be performed while all peripheral modules are disabled. Changing the threshold level during the time a module is active may inadvertently generate a transition associated with an input pin, regardless of the actual voltage level on that pin. |

### 11.5.4 OPEN DRAIN CONTROL

The ODCONC register (Register 11-22) controls the open-drain feature of the port. Open drain operation is independently selected for each pin. When an ODCONC bit is set, the corresponding port output becomes an open drain driver capable of sinking current only. When an ODCONC bit is cleared, the corresponding port output pin is the standard push-pull drive capable of sourcing and sinking current.

### 11.5.5 SLEW RATE CONTROL

The SLRCONC register (Register 11-23) controls the slew rate option for each port pin. Slew rate control is independently selectable for each port pin. When an SLRCONC bit is set, the corresponding port pin drive is slew rate limited. When an SLRCONC bit is cleared, The corresponding port pin drive slews at the maximum rate possible.

### 11.5.6 ANALOG CONTROL

The ANSELC register (Register 11-20) is used to configure the Input mode of an I/O pin to analog. Setting the appropriate ANSELC bit high will cause all digital reads on the pin to be read as '0' and allow analog functions on the pin to operate correctly.

The state of the ANSELC bits has no effect on digital output functions. A pin with TRIS clear and ANSELC set will still operate as a digital output, but the Input mode will be analog. This can cause unexpected behavior when executing read-modify-write instructions on the affected port.

| | |
|---|---|
| **Note:** | The ANSELC bits default to the Analog mode after Reset. To use any pins as digital general purpose or peripheral inputs, the corresponding ANSEL bits must be initialized to '0' by user software. |

### 11.5.7 PORTC FUNCTIONS AND OUTPUT PRIORITIES

Each pin defaults to the PORT latch data after reset. Other functions are selected with the peripheral pin select logic. See **Section 12.0 "Peripheral Pin Select (PPS) Module"** for more information.

Analog input functions, such as ADC inputs, are not shown in the peripheral pin select lists. These inputs are active when the I/O pin is set for Analog mode using the ANSELC register. Digital output functions may continue to control the pin when it is in Analog mode.

## 11.6 Register Definitions: PORTC

### REGISTER 11-17: PORTC: PORTC REGISTER

| R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u |
|---|---|---|---|---|---|---|---|
| RC7(2) | RC6(2) | RC5 | RC4 | RC3 | RC2 | RC1 | RC0 |
| bit 7 | | | | | | | bit 0 |

| Legend: | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| '1' = Bit is set | '0' = Bit is cleared | |

bit 7-0 **RC<7:0>**: PORTC General Purpose I/O Pin bits(1, 2)
1 = Port pin is ≥ VIH
0 = Port pin is ≤ VIL

**Note 1:** Writes to PORTC are actually written to corresponding LATC register. Reads from PORTC register is return of actual I/O pin values.
**2:** RC<7:6> are available on PIC16(L)F1707 only.

### REGISTER 11-18: TRISC: PORTC TRI-STATE REGISTER

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

| Legend: | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| '1' = Bit is set | '0' = Bit is cleared | |

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

**Note 1:** TRISC<7:6> are available on PIC16(L)F1707 only.

### REGISTER 11-19: LATC: PORTC DATA LATCH REGISTER

| R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u |
|---|---|---|---|---|---|---|---|
| LATC7(1) | LATC6(1) | LATC5 | LATC4 | LATC3 | LATC2 | LATC1 | LATC0 |
| bit 7 | | | | | | | bit 0 |

| Legend: | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| '1' = Bit is set | '0' = Bit is cleared | |

bit 7-0 **LATC<7:0>**: PORTC Output Latch Value bits(1)

**Note 1:** LATC<7:6> are available on PIC16(L)F1707 only.


© 2013-2015 Microchip Technology Inc.

**REGISTER 11-20: ANSELC: PORTC ANALOG SELECT REGISTER**

| R/W-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 |
|---|---|---|---|---|---|---|---|
| ANSC7(2) | ANSC6(2) | ANSC5(3) | ANSC4(3) | ANSC3 | ANSC2 | ANSC1 | ANSC0 |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| '1' = Bit is set | '0' = Bit is cleared | |

bit 7-0 **ANSC<7:0>**: Analog Select between Analog or Digital Function on pins RC<7:0>, respectively(1)
0 = Digital I/O. Pin is assigned to port or digital special function.
1 = Analog input. Pin is assigned as analog input(1). Digital input buffer disabled.

**Note 1:** When setting a pin to an analog input, the corresponding TRIS bit must be set to Input mode in order to allow external control of the voltage on the pin.
**2:** ANSC<7:6> are available on PIC16(L)F1707 only.
**3:** ANSC<5:4> are available on PIC16(L)F1703 only.

**REGISTER 11-21: WPUC: WEAK PULL-UP PORTC REGISTER**

| R/W-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 |
|---|---|---|---|---|---|---|---|
| WPUC7(3) | WPUC6(3) | WPUC5 | WPUC4 | WPUC3 | WPUC2 | WPUC1 | WPUC0 |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| '1' = Bit is set | '0' = Bit is cleared | |

bit 7-0 **WPUC<7:0>**: Weak Pull-up Register bits(3)
1 = Pull-up enabled
0 = Pull-up disabled

**Note 1:** Global $\overline{\text{WPUEN}}$ bit of the OPTION_REG register must be cleared for individual pull-ups to be enabled.
**2:** The weak pull-up device is automatically disabled if the pin is configured as an output.
**3:** WPUC<7:6> are available on PIC16(L)F1707 only.

**REGISTER 11-22: ODCONC: PORTC OPEN DRAIN CONTROL REGISTER**

| R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| ODC7(1) | ODC6(1) | ODC5 | ODC4 | ODC3 | ODC2 | ODC1 | ODC0 |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| '1' = Bit is set | '0' = Bit is cleared | |

bit 7-0 **ODC<7:0>:** PORTC Open Drain Enable bits(1)
For RC<7:0> pins, respectively
1 = Port pin operates as open-drain drive (sink current only)
0 = Port pin operates as standard push-pull drive (source and sink current)

**Note 1:** ODC<7:6> are available on PIC16(L)F1707 only.

**REGISTER 11-23: SLRCONC: PORTC SLEW RATE CONTROL REGISTER**

| R/W-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 |
|---|---|---|---|---|---|---|---|
| SLRC7(1) | SLRC6(1) | SLRC5 | SLRC4 | SLRC3 | SLRC2 | SLRC1 | SLRC0 |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| '1' = Bit is set | '0' = Bit is cleared | |

bit 7-0 **SLRC<7:0>:** PORTC Slew Rate Enable bits(1)
For RC<7:0> pins, respectively
1 = Port pin slew rate is limited
0 = Port pin slews at maximum rate

**Note 1:** SLRC<7:6> are available on PIC16(L)F1707 only.

**REGISTER 11-24: INLVLC: PORTC INPUT LEVEL CONTROL REGISTER**

| R/W-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 |
|---|---|---|---|---|---|---|---|
| INLVLC7(1) | INLVLC6(1) | INLVLC5 | INLVLC4 | INLVLC3 | INLVLC2 | INLVLC1 | INLVLC0 |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| '1' = Bit is set | '0' = Bit is cleared | |

bit 7-0 **INLVLC<7:0>:** PORTC Input Level Select bits(1)
For RC<7:0> pins, respectively
1 = ST input used for PORT reads and interrupt-on-change
0 = TTL input used for PORT reads and interrupt-on-change

**Note 1:** INLVLC<7:6> are available on PIC16(L)F1707 only.


© 2013-2015 Microchip Technology Inc.

**TABLE 11-5: SUMMARY OF REGISTERS ASSOCIATED WITH PORTC**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|
| ANSELC | ANSC7(1) | ANSC6(1) | ANSC5(2) | ANSC4(2) | ANSC3 | ANSC2 | ANSC1 | ANSC0 | 125 |
| INLVLC | INLVLC7(1) | INLVLC6(1) | INLVLC5 | INLVLC4 | INLVLC3 | INLVLC2 | INLVLC1 | INLVLC0 | 126 |
| LATC | LATC7(1) | LATC6(1) | LATC5 | LATC4 | LATC3 | LATC2 | LATC1 | LATC0 | 124 |
| ODCONC | ODC7(1) | ODC6(1) | ODC5 | ODC4 | ODC3 | ODC2 | ODC1 | ODC0 | 126 |
| PORTC | RC7(1) | RC6(1) | RC5 | RC4 | RC3 | RC2 | RC1 | RC0 | 124 |
| SLRCONC | SLRC7(1) | SLRC6(1) | SLRC5 | SLRC4 | SLRC3 | SLRC2 | SLRC1 | SLRC0 | 126 |
| TRISC | TRISC7(1) | TRISC6(1) | TRISC5 | TRISC4 | TRISC3 | TRISC2 | TRISC1 | TRISC0 | 124 |
| WPUC | WPUC7(1) | WPUC6(1) | WPUC5 | WPUC4 | WPUC3 | WPUC2 | WPUC1 | WPUC0 | 125 |

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

**Note 1:** PIC16(L)F1707 only.

**2:** PIC16(L)F1703 only.

## 12.0 PERIPHERAL PIN SELECT (PPS) MODULE

The Peripheral Pin Select (PPS) module connects peripheral inputs and outputs to the device I/O pins. Only digital signals are included in the selections. All analog inputs and outputs remain fixed to their assigned pins. Input and output selections are independent as shown in the simplified block diagram Figure 12-1.

### 12.1 PPS Inputs

Each peripheral has a PPS register with which the inputs to the peripheral are selected. Inputs include the device pins.

Multiple peripherals can operate from the same source simultaneously. Port reads always return the pin level regardless of peripheral PPS selection. If a pin also has associated analog functions, the ANSEL bit for that pin must be cleared to enable the digital input buffer.

Although every peripheral has its own PPS input selection register, the selections are identical for every peripheral as shown in Register 12-1 for PIC16(L)F1703 devices and Register 12-2 for PIC16(L)F1707 devices.

| | |
|---|---|
| **Note:** | The notation “xxx” in the register name is a place holder for the peripheral identifier. For example, CCP1PPS. |

### 12.2 PPS Outputs

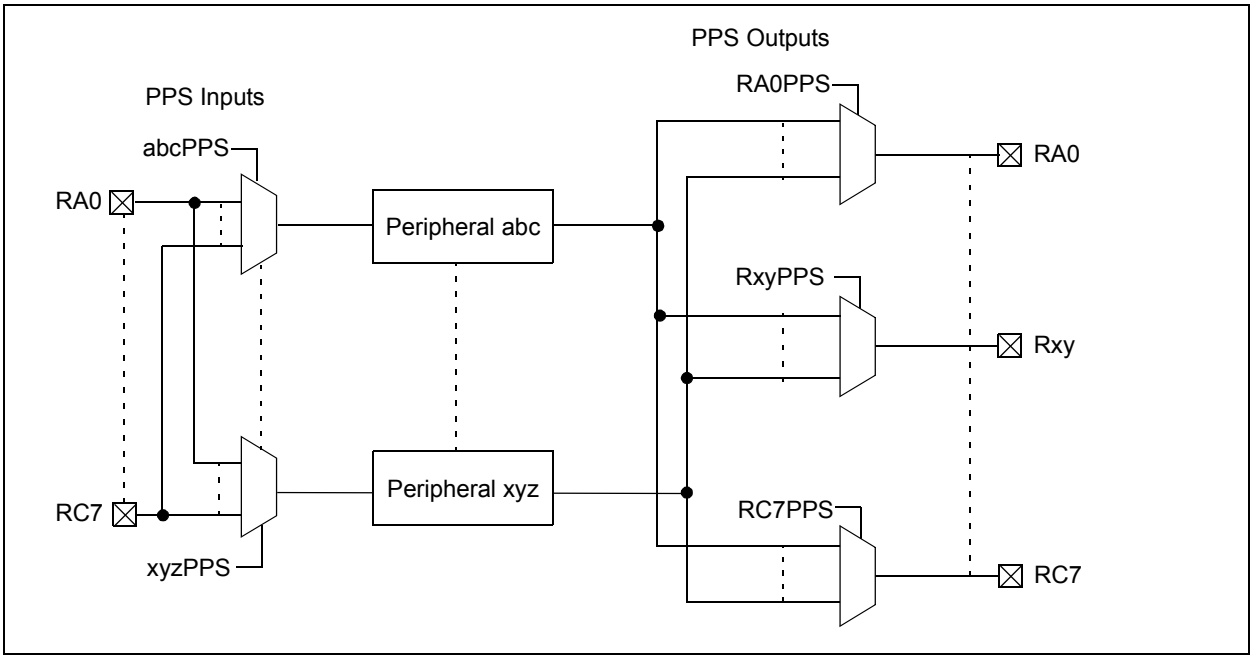
Each I/O pin has a PPS register with which the pin output source is selected. With few exceptions, the port TRIS control associated with that pin retains control over the pin output driver. Peripherals that control the pin output driver as part of the peripheral operation will override the TRIS control as needed. These peripherals include:

- MSSP ($I^2C$)

Although every pin has its own PPS peripheral selection register, the selections are identical for every pin as shown in Register 12-3.

| | |
|---|---|
| **Note:** | The notation “Rxy” is a place holder for the pin identifier. For example, RA0PPS. |

**FIGURE 12-1: SIMPLIFIED PPS BLOCK DIAGRAM**





© 2013-2015 Microchip Technology Inc.

## 12.3 Bidirectional Pins

PPS selections for peripherals with bidirectional signals on a single pin must be made so that the PPS input and PPS output select the same pin. Peripherals that have bidirectional signals include:

- MSSP ($I^2C$)

| **Note:** | The $I^2C$ default input pins are $I^2C$ and SMBus compatible and are the only pins on the device with this compatibility. |
|---|---|

## 12.4 PPS Lock

The PPS includes a mode in which all input and output selections can be locked to prevent inadvertent changes. PPS selections are locked by setting the PPSLOCKED bit of the PPSLOCK register. Setting and clearing this bit requires a special sequence as an extra precaution against inadvertent changes. Examples of setting and clearing the PPSLOCKED bit are shown in Example 12-1.

**EXAMPLE 12-1: PPS LOCK/UNLOCK SEQUENCE**

```
; suspend interrupts
    bcf     INTCON,GIE
;   BANKSEL PPSLOCK     ; set bank
; required sequence, next 5 instructions
    movlw   0x55
    movwf   PPSLOCK
    movlw   0xAA
    movwf   PPSLOCK
; Set PPSLOCKED bit to disable writes or
; Clear PPSLOCKED bit to enable writes
    bsf     PPSLOCK,PPSLOCKED
; restore interrupts
    bsf     INTCON,GIE
```

## 12.5 PPS Permanent Lock

The PPS can be permanently locked by setting the PPS1WAY Configuration bit. When this bit is set, the PPSLOCKED bit can only be cleared and set one time after a device Reset. This allows for clearing the PPSLOCKED bit so that the input and output selections can be made during initialization. When the PPSLOCKED bit is set after all selections have been made, it will remain set and cannot be cleared until after the next device Reset event.

## 12.6 Operation During Sleep

PPS input and output selections are unaffected by Sleep.

## 12.7 Effects of a Reset

A device Power-On-Reset (POR) clears all PPS input and output selections to their default values. All other Resets leave the selections unchanged. Default input selections are shown in pin allocation Table 1 and Table 2.

## 12.8 Register Definitions: PPS Input Selection

**REGISTER 12-1: xxxPPS: PERIPHERAL xxx INPUT SELECTION (PIC16(L)F1703)**

| U-0 | U-0 | U-0 | R/W-q/u | R/W-q/u | R/W-q/u | R/W-q/u | R/W-q/u |
|---|---|---|---|---|---|---|---|
| — | — | — | xxxPPS<4:0> | | | | |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as ‘0’ |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| ‘1’ = Bit is set | ‘0’ = Bit is cleared | q = value depends on peripheral |

bit 7-5 **Unimplemented:** Read as ‘0’

bit 4-0 **xxxPPS<4:0>:** Peripheral xxx Input Selection bits

11xxx = Reserved. Do not use.

1011x = Reserved. Do not use.
10101 = Peripheral input is RC5
10100 = Peripheral input is RC4
10011 = Peripheral input is RC3
10010 = Peripheral input is RC2
10001 = Peripheral input is RC1
10000 = Peripheral input is RC0

01xxx = Reserved. Do not use.

0011X = Reserved. Do not use.
00101 = Peripheral input is RA5
00100 = Peripheral input is RA4
00011 = Peripheral input is RA3
00010 = Peripheral input is RA2
00001 = Peripheral input is RA1
00000 = Peripheral input is RA0


© 2013-2015 Microchip Technology Inc.

**REGISTER 12-2: xxxPPS: PERIPHERAL xxx INPUT SELECTION (PIC16(L)F1707)**

| U-0 | U-0 | U-0 | R/W-q/u | R/W-q/u | R/W-q/u | R/W-q/u | R/W-q/u |
|---|---|---|---|---|---|---|---|
| — | — | — | xxxPPS<4:0> | | | | |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| '1' = Bit is set | '0' = Bit is cleared | q = value depends on peripheral |

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

bit 4-0 **xxxPPS<4:0>:** Peripheral xxx Input Selection bits

11xxx = Reserved. Do not use.

10111 = Peripheral input is RC7
10110 = Peripheral input is RC6
10101 = Peripheral input is RC5
10100 = Peripheral input is RC4
10011 = Peripheral input is RC3
10010 = Peripheral input is RC2
10001 = Peripheral input is RC1
10000 = Peripheral input is RC0

01111 = Peripheral input is RB7
01110 = Peripheral input is RB6
01101 = Peripheral input is RB5
01100 = Peripheral input is RB4

010xx = Reserved. Do not use.

0011X = Reserved. Do not use.
00101 = Peripheral input is RA5
00100 = Peripheral input is RA4
00011 = Peripheral input is RA3
00010 = Peripheral input is RA2
00001 = Peripheral input is RA1
00000 = Peripheral input is RA0

**REGISTER 12-3: RxyPPS: PIN Rxy OUTPUT SOURCE SELECTION REGISTER**

| U-0 | U-0 | U-0 | R/W-0/u | R/W-0/u | R/W-0/u | R/W-0/u | R/W-0/u |
|---|---|---|---|---|---|---|---|
| — | — | — | RxyPPS<4:0> | | | | |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| '1' = Bit is set | '0' = Bit is cleared | |

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

bit 4-0 **RxyPPS<4:0>:** Pin Rxy Output Source Selection bits

11xxx = Reserved
10111 = Rxy source is C2OUT
10110 = Rxy source is C1OUT
10101 = Reserved
10100 = Reserved
10011 = Reserved
10010 = Rxy source is SDO
10001 = Rxy source is SDA(1)
10000 = Rxy source is SCK/SCL(1)

01111 = Reserved
01110 = Reserved
01101 = Rxy source is CCP2
01100 = Rxy source is CCP1
01011 = Reserved
01010 = Reserved
01001 = Reserved
01000 = Reserved

00111 = Reserved
00110 = Reserved
00101 = Reserved
00100 = Reserved
00011 = Reserved
00010 = Reserved
00001 = Reserved
00000 = Rxy source is LATxy

**Note 1:** TRIS control is overridden by the peripheral as required.


© 2013-2015 Microchip Technology Inc.

**REGISTER 12-4: PPSLOCK: PPS LOCK REGISTER**

| U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| — | — | — | — | — | — | — | PPSLOCKED |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as ‘0’ |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| ‘1’ = Bit is set | ‘0’ = Bit is cleared | |

bit 7-1 **Unimplemented:** Read as ‘0’

bit 0 **PPSLOCKED:** PPS Locked bit
1= PPS is locked. PPS selections can not be changed.
0= PPS is not locked. PPS selections can be changed.

**TABLE 12-1: SUMMARY OF REGISTERS ASSOCIATED WITH THE PPS MODULE**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on page |
|---|---|---|---|---|---|---|---|---|---|
| PPSLOCK | — | — | — | — | — | — | — | PPSLOCKED | 133 |
| INTPPS | — | — | — | INTPPS<4:0> | | | | | 131 |
| T0CKIPPS | — | — | — | T0CKIPPS<4:0> | | | | | 131 |
| T1CKIPPS | — | — | — | T1CKIPPS<4:0> | | | | | 131 |
| T1GPPS | — | — | — | T1GPPS<4:0> | | | | | 131 |
| CCP1PPS | — | — | — | CCP1PPS<4:0> | | | | | 131 |
| CCP2PPS | — | — | — | CCP2PPS<4:0> | | | | | 131 |
| SSPCLKPPS | — | — | — | SSPCLKPPS<4:0> | | | | | 131 |
| SSPDATPPS | — | — | — | SSPDATPPS<4:0> | | | | | 131 |
| SSPSSPPS | — | — | — | SSPSSPPS<4:0> | | | | | 131 |
| RA0PPS | — | — | — | RA0PPS<4:0> | | | | | 132 |
| RA1PPS | — | — | — | RA1PPS<4:0> | | | | | 132 |
| RA2PPS | — | — | — | RA2PPS<4:0> | | | | | 132 |
| RA4PPS | — | — | — | RA4PPS<4:0> | | | | | 132 |
| RA5PPS | — | — | — | RA5PPS<4:0> | | | | | 132 |
| RB4PPS(1) | — | — | — | RB4PPS<4:0> | | | | | 132 |
| RB5PPS(1) | — | — | — | RB5PPS<4:0> | | | | | 132 |
| RB6PPS(1) | — | — | — | RB6PPS<4:0> | | | | | 132 |
| RB7PPS(1) | — | — | — | RB7PPS<4:0> | | | | | 132 |
| RC0PPS | — | — | — | RC0PPS<4:0> | | | | | 132 |
| RC1PPS | — | — | — | RC1PPS<4:0> | | | | | 132 |
| RC2PPS | — | — | — | RC2PPS<4:0> | | | | | 132 |
| RC3PPS | — | — | — | RC3PPS<4:0> | | | | | 132 |
| RC4PPS | — | — | — | RC4PPS<4:0> | | | | | 132 |
| RC5PPS | — | — | — | RC5PPS<4:0> | | | | | 132 |
| RC6PPS(1) | — | — | — | RC6PPS<4:0> | | | | | 132 |
| RC7PPS(1) | — | — | — | RC7PPS<4:0> | | | | | 132 |

**Legend:** — = unimplemented, read as '0'.

**Note 1:** PIC16(L)F1707 only.


© 2013-2015 Microchip Technology Inc.

## 13.0 INTERRUPT-ON-CHANGE

All pins on all ports can be configured to operate as Interrupt-On-Change (IOC) pins. An interrupt can be generated by detecting a signal that has either a rising edge or a falling edge. Any individual pin, or combination of pins, can be configured to generate an interrupt. The interrupt-on-change module has the following features:

- Interrupt-on-Change enable (Master Switch)
- Individual pin configuration
- Rising and falling edge detection
- Individual pin interrupt flags

Figure 13-1 is a block diagram of the IOC module.

### 13.1 Enabling the Module

To allow individual pins to generate an interrupt, the IOCIE bit of the INTCON register must be set. If the IOCIE bit is disabled, the edge detection on the pin will still occur, but an interrupt will not be generated.

### 13.2 Individual Pin Configuration

For each pin, a rising edge detector and a falling edge detector are present. To enable a pin to detect a rising edge, the associated bit of the IOCxP register is set. To enable a pin to detect a falling edge, the associated bit of the IOCxN register is set.

A pin can be configured to detect rising and falling edges simultaneously by setting the associated bits in both of the IOCxP and IOCxN registers.

### 13.3 Interrupt Flags

The bits located in the IOCxF registers are status flags that correspond to the interrupt-on-change pins of each port. If an expected edge is detected on an appropriately enabled pin, then the status flag for that pin will be set, and an interrupt will be generated if the IOCIE bit is set. The IOCIF bit of the INTCON register reflects the status of all IOCxF bits.

### 13.4 Clearing Interrupt Flags

The individual status flags, (IOCxF register bits), can be cleared by resetting them to zero. If another edge is detected during this clearing operation, the associated status flag will be set at the end of the sequence, regardless of the value actually being written.

In order to ensure that no detected edge is lost while clearing flags, only AND operations masking out known changed bits should be performed. The following sequence is an example of what should be performed.

**EXAMPLE 13-1: CLEARING INTERRUPT FLAGS (PORTA EXAMPLE)**

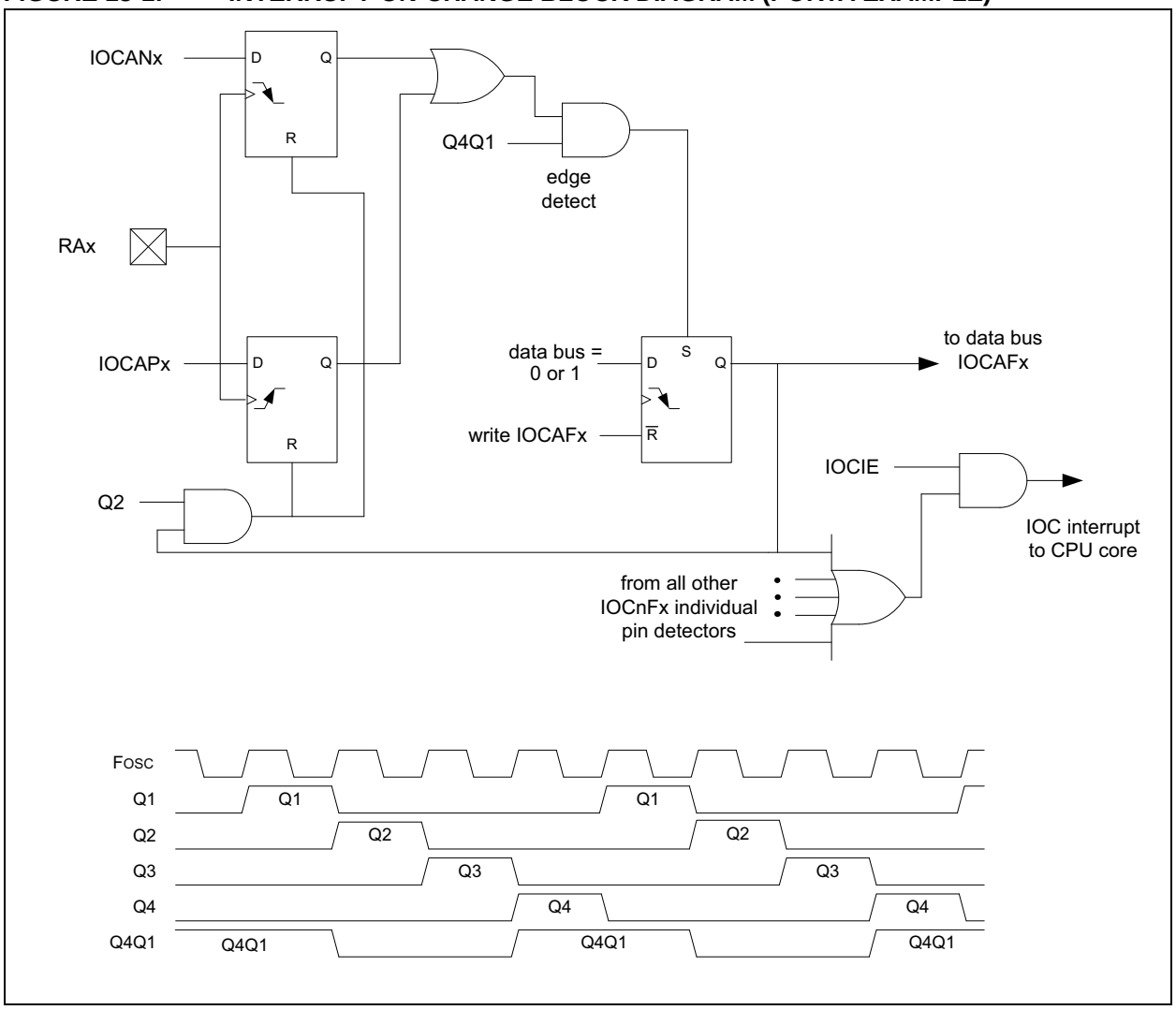
```
MOVLW  0xff
XORWF  IOCAF, W
ANDWF  IOCAF, F
```

### 13.5 Operation in Sleep

The interrupt-on-change interrupt sequence will wake the device from Sleep mode, if the IOCIE bit is set.

If an edge is detected while in Sleep mode, the affected IOCxF register will be updated prior to the first instruction executed out of Sleep.

**FIGURE 13-1: INTERRUPT-ON-CHANGE BLOCK DIAGRAM (PORTA EXAMPLE)**





© 2013-2015 Microchip Technology Inc.

## 13.6 Register Definitions: Interrupt-on-Change Control

### REGISTER 13-1: IOCAP: INTERRUPT-ON-CHANGE PORTA POSITIVE EDGE REGISTER

| U-0 | U-0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| — | — | IOCAP5 | IOCAP4 | IOCAP3 | IOCAP2 | IOCAP1 | IOCAP0 |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as ‘0’ |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| ‘1’ = Bit is set | ‘0’ = Bit is cleared | |

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5-0 **IOCAP<5:0>:** Interrupt-on-Change PORTA Positive Edge Enable bits
1 = Interrupt-on-Change enabled on the pin for a positive going edge. IOCAFx bit and IOCIF flag will be set upon detecting an edge.
0 = Interrupt-on-Change disabled for the associated pin.

### REGISTER 13-2: IOCAN: INTERRUPT-ON-CHANGE PORTA NEGATIVE EDGE REGISTER

| U-0 | U-0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| — | — | IOCAN5 | IOCAN4 | IOCAN3 | IOCAN2 | IOCAN1 | IOCAN0 |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as ‘0’ |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| ‘1’ = Bit is set | ‘0’ = Bit is cleared | |

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5-0 **IOCAN<5:0>:** Interrupt-on-Change PORTA Negative Edge Enable bits
1 = Interrupt-on-Change enabled on the pin for a negative going edge. IOCAFx bit and IOCIF flag will be set upon detecting an edge.
0 = Interrupt-on-Change disabled for the associated pin.

### REGISTER 13-3: IOCAF: INTERRUPT-ON-CHANGE PORTA FLAG REGISTER

| U-0 | U-0 | R/W/HS-0/0 | R/W/HS-0/0 | R/W/HS-0/0 | R/W/HS-0/0 | R/W/HS-0/0 | R/W/HS-0/0 |
|---|---|---|---|---|---|---|---|
| — | — | IOCAF5 | IOCAF4 | IOCAF3 | IOCAF2 | IOCAF1 | IOCAF0 |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as ‘0’ |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| ‘1’ = Bit is set | ‘0’ = Bit is cleared | HS - Bit is set in hardware |

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5-0 **IOCAF<5:0>:** Interrupt-on-Change PORTA Flag bits
1 = An enabled change was detected on the associated pin.
Set when IOCAPx = 1 and a rising edge was detected on RAx, or when IOCANx = 1 and a falling edge was detected on RAx.
0 = No change was detected, or the user cleared the detected change.

**REGISTER 13-4: IOCBP: INTERRUPT-ON-CHANGE PORTB POSITIVE EDGE REGISTER(1)**

| R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | U-0 | U-0 | U-0 | U-0 |
|---|---|---|---|---|---|---|---|
| IOCBP7 | IOCBP6 | IOCBP5 | IOCBP4 | — | — | — | — |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| '1' = Bit is set | '0' = Bit is cleared | |

bit 7-4 **IOCBP<7:4>:** Interrupt-on-Change PORTB Positive Edge Enable bits
1 = Interrupt-on-Change enabled on the pin for a positive going edge. IOCBFx bit and IOCIF flag will be set upon detecting an edge.
0 = Interrupt-on-Change disabled for the associated pin.

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

**Note 1:** PIC16(L)F1707 only.

**REGISTER 13-5: IOCBN: INTERRUPT-ON-CHANGE PORTB NEGATIVE EDGE REGISTER(1)**

| R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | U-0 | U-0 | U-0 | U-0 |
|---|---|---|---|---|---|---|---|
| IOCBN7 | IOCBN6 | IOCBN5 | IOCBN4 | — | — | — | — |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| '1' = Bit is set | '0' = Bit is cleared | |

bit 7-4 **IOCBN<7:4>:** Interrupt-on-Change PORTB Negative Edge Enable bits
1 = Interrupt-on-Change enabled on the pin for a negative going edge. IOCBFx bit and IOCIF flag will be set upon detecting an edge.
0 = Interrupt-on-Change disabled for the associated pin.

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

**Note 1:** PIC16(L)F1707 only.

**REGISTER 13-6: IOCBF: INTERRUPT-ON-CHANGE PORTB FLAG REGISTER(1)**

| R/W/HS-0/0 | R/W/HS-0/0 | R/W/HS-0/0 | R/W/HS-0/0 | U-0 | U-0 | U-0 | U-0 |
|---|---|---|---|---|---|---|---|
| IOCBF7 | IOCBF6 | IOCBF5 | IOCBF4 | — | — | — | — |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| '1' = Bit is set | '0' = Bit is cleared | HS - Bit is set in hardware |

bit 7-4 **IOCBF<7:4>:** Interrupt-on-Change PORTB Flag bits
1 = An enabled change was detected on the associated pin.
Set when IOCBPx = 1 and a rising edge was detected on RBx, or when IOCBNx = 1 and a falling edge was detected on RBx.
0 = No change was detected, or the user cleared the detected change.

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

**Note 1:** PIC16(L)F1707 only.


© 2013-2015 Microchip Technology Inc.

**REGISTER 13-7: IOCCP: INTERRUPT-ON-CHANGE PORTC POSITIVE EDGE REGISTER**

| R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| IOCCP7(1) | IOCCP6(1) | IOCCP5 | IOCCP4 | IOCCP3 | IOCCP2 | IOCCP1 | IOCCP0 |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| '1' = Bit is set | '0' = Bit is cleared | |

bit 7-0 **IOCCP<7:0>:** Interrupt-on-Change PORTC Positive Edge Enable bits
1 = Interrupt-on-Change enabled on the pin for a positive going edge. IOCCFx bit and IOCIF flag will be set upon detecting an edge.
0 = Interrupt-on-Change disabled for the associated pin.

**Note 1:** PIC16(L)F1707 only.

**REGISTER 13-8: IOCCN: INTERRUPT-ON-CHANGE PORTC NEGATIVE EDGE REGISTER**

| R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| IOCCN7(1) | IOCCN6(1) | IOCCN5 | IOCCN4 | IOCCN3 | IOCCN2 | IOCCN1 | IOCCN0 |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| '1' = Bit is set | '0' = Bit is cleared | |

bit 7-0 **IOCCN<7:0>:** Interrupt-on-Change PORTC Negative Edge Enable bits
1 = Interrupt-on-Change enabled on the pin for a negative going edge. IOCCFx bit and IOCIF flag will be set upon detecting an edge.
0 = Interrupt-on-Change disabled for the associated pin.

**Note 1:** PIC16(L)F1707 only.

**REGISTER 13-9: IOCCF: INTERRUPT-ON-CHANGE PORTC FLAG REGISTER**

| R/W/HS-0/0 | R/W/HS-0/0 | R/W/HS-0/0 | R/W/HS-0/0 | R/W/HS-0/0 | R/W/HS-0/0 | R/W/HS-0/0 | R/W/HS-0/0 |
|---|---|---|---|---|---|---|---|
| IOCCF7(1) | IOCCF6(1) | IOCCF5 | IOCCF4 | IOCCF3 | IOCCF2 | IOCCF1 | IOCCF0 |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| '1' = Bit is set | '0' = Bit is cleared | HS - Bit is set in hardware |

bit 7-0 **IOCCF<7:0>:** Interrupt-on-Change PORTC Flag bits
1 = An enabled change was detected on the associated pin.
Set when IOCCPx = 1 and a rising edge was detected on RCx, or when IOCCNx = 1 and a falling edge was detected on RCx.
0 = No change was detected, or the user cleared the detected change.

**Note 1:** PIC16(L)F1707 only.

**TABLE 13-1: SUMMARY OF REGISTERS ASSOCIATED WITH INTERRUPT-ON-CHANGE**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|
| ANSELA | — | — | — | ANSA4 | — | ANSA2 | ANSA1 | ANSA0 | 114 |
| ANSELB(1) | — | — | ANSB5 | ANSB4 | — | — | — | — | 120 |
| ANSELC | ANSC7(1) | ANSC6(1) | ANSC5(2) | ANSC4(2) | ANSC3 | ANSC2 | ANSC1 | ANSC0 | 125 |
| INTCON | GIE | PEIE | TMR0IE | INTE | IOCIE | TMR0IF | INTF | IOCIF | 77 |
| IOCAF | — | — | IOCAF5 | IOCAF4 | IOCAF3 | IOCAF2 | IOCAF1 | IOCAF0 | 137 |
| IOCAN | — | — | IOCAN5 | IOCAN4 | IOCAN3 | IOCAN2 | IOCAN1 | IOCAN0 | 137 |
| IOCAP | — | — | IOCAP5 | IOCAP4 | IOCAP3 | IOCAP2 | IOCAP1 | IOCAP0 | 137 |
| IOCBF(1) | IOCBF7 | IOCBF6 | IOCBF5 | IOCBF4 | — | — | — | — | 138 |
| IOCBN(1) | IOCBN7 | IOCBN6 | IOCBN5 | IOCBN4 | — | — | — | — | 138 |
| IOCBP(1) | IOCBP7 | IOCBP6 | IOCBP5 | IOCBP4 | — | — | — | — | 138 |
| IOCCF | IOCCF7(1) | IOCCF6(1) | IOCCF5 | IOCCF4 | IOCCF3 | IOCCF2 | IOCCF1 | IOCCF0 | 139 |
| IOCCN | IOCCN7(1) | IOCCN6(1) | IOCCN5 | IOCCN4 | IOCCN3 | IOCCN2 | IOCCN1 | IOCCN0 | 139 |
| IOCCP | IOCCP7(1) | IOCCP6(1) | IOCCP5 | IOCCP4 | IOCCP3 | IOCCP2 | IOCCP1 | IOCCP0 | 139 |
| TRISA | — | — | TRISA5 | TRISA4 | —(3) | TRISA2 | TRISA1 | TRISA0 | 113 |
| TRISB(1) | TRISB7 | TRISB6 | TRISB5 | TRISB4 | — | — | — | — | 119 |
| TRISC | TRISC7(1) | TRISC6(1) | TRISC5 | TRISC4 | TRISC3 | TRISC2 | TRISC1 | TRISC0 | 124 |

**Legend:** — = unimplemented location, read as '0'. Shaded cells are not used by interrupt-on-change.

**Note 1:** PIC16(L)F1707 only.

**2:** PIC16(L)F1703 only.

**3:** Unimplemented, read as '1'.


© 2013-2015 Microchip Technology Inc.

## 14.0 FIXED VOLTAGE REFERENCE (FVR)

The Fixed Voltage Reference, or FVR, is a stable voltage reference, independent of VDD, with 1.024V, 2.048V or 4.096V selectable output levels. The output of the FVR can be configured to supply a reference voltage to the following:

- ADC input channel
- ADC positive reference

The FVR can be enabled by setting the FVREN bit of the FVRCON register.

### 14.1 Independent Gain Amplifiers

The output of the FVR supplied to the ADC is routed through two independent programmable gain amplifiers. Each amplifier can be programmed for a gain of 1x, 2x or 4x, to produce the three possible voltage levels.

The ADFVR<1:0> bits of the FVRCON register are used to enable and configure the gain amplifier settings for the reference supplied to the ADC module. Reference **Section 16.0 "Analog-to-Digital Converter (ADC) Module"** for additional information.

The CDAFVR<1:0> bits of the FVRCON register are used to enable and configure the gain amplifier settings supplied to the Op Amp.

### 14.2 FVR Stabilization Period

When the Fixed Voltage Reference module is enabled, it requires time for the reference and amplifier circuits to stabilize. Once the circuits stabilize and are ready for use, the FVRRDY bit of the FVRCON register will be set. See Figure 27-62.

**FIGURE 14-1: VOLTAGE REFERENCE BLOCK DIAGRAM**

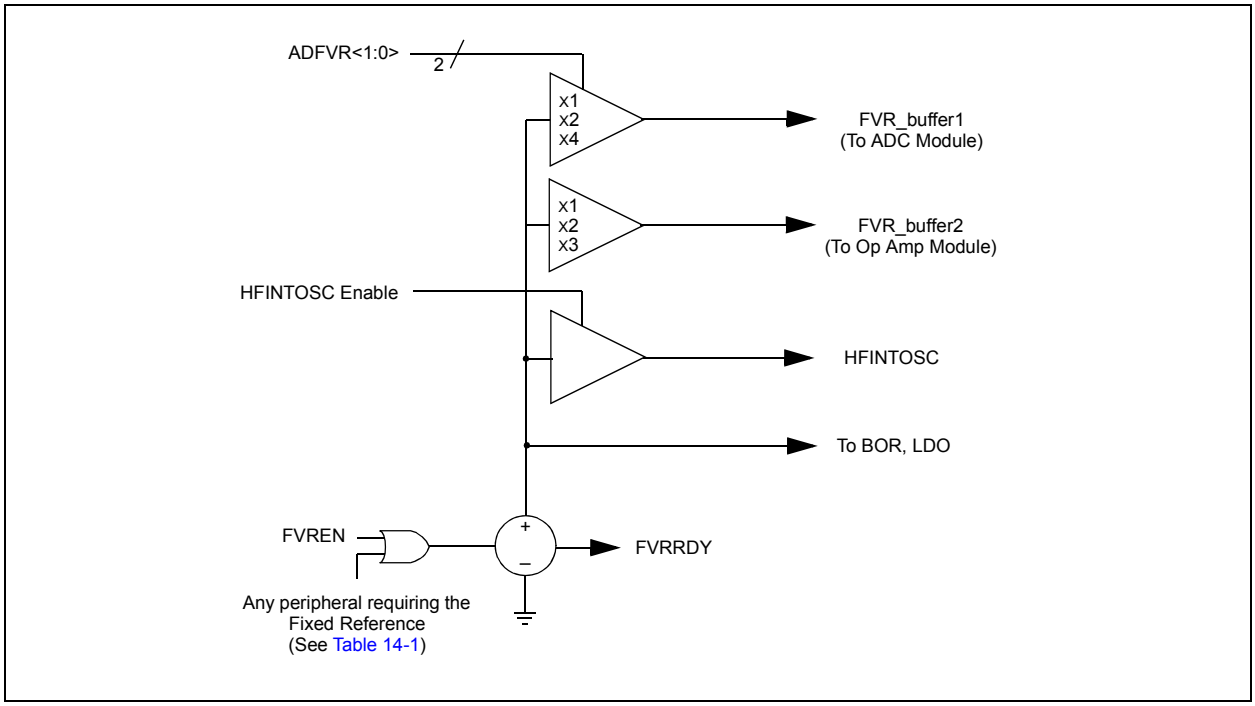



**TABLE 14-1: PERIPHERALS REQUIRING THE FIXED VOLTAGE REFERENCE (FVR)**

| Peripheral | Conditions | Description |
|---|---|---|
| HFINTOSC | FOSC<1:0> = 00 and IRCF<3:0> ≠ 000x | INTOSC is active and device is not in Sleep |
| BOR | BOREN<1:0> = 11 | BOR always enabled |
| | BOREN<1:0> = 10 and BORFS = 1 | BOR disabled in Sleep mode, BOR Fast Start enabled |
| | BOREN<1:0> = 01 and BORFS = 1 | BOR under software control, BOR Fast Start enabled |
| LDO | All PIC16F1703/7 devices, when VREGPM = 1 and not in Sleep | The device runs off of the ULP regulator when in Sleep mode |

## 14.3 Register Definitions: FVR Control

**REGISTER 14-1: FVRCON: FIXED VOLTAGE REFERENCE CONTROL REGISTER**

| R/W-0/0 | R-q/q | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| FVREN | FVRRDY(1) | TSEN(3) | TSRNG(3) | CDAFVR<1:0> | | ADFVR<1:0> | |
| bit 7 | | | | | | | bit 0 |

| Legend: | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| '1' = Bit is set | '0' = Bit is cleared | q = Value depends on condition |

bit 7 **FVREN:** Fixed Voltage Reference Enable bit
1 = Fixed Voltage Reference is enabled
0 = Fixed Voltage Reference is disabled

bit 6 **FVRRDY:** Fixed Voltage Reference Ready Flag bit(1)
1 = Fixed Voltage Reference output is ready for use
0 = Fixed Voltage Reference output is not ready or not enabled

bit 5 **TSEN:** Temperature Indicator Enable bit(3)
1 = Temperature Indicator is enabled
0 = Temperature Indicator is disabled

bit 4 **TSRNG:** Temperature Indicator Range Selection bit(3)
1 = $V_{OUT} = V_{DD} - 4V_T$ (High Range)
0 = $V_{OUT} = V_{DD} - 2V_T$ (Low Range)

bit 3-2 **CDAFVR<1:0>**: Op Amp Fixed Voltage Reference Selection bit
11 = Op Amp Fixed Voltage Reference Peripheral output is 4x (4.096V)(2)
10 = Op Amp Fixed Voltage Reference Peripheral output is 2x (2.048V)(2)
01 = Op Amp Fixed Voltage Reference Peripheral output is 1x (1.024V)
00 = Op Amp Fixed Voltage Reference Peripheral is off.

bit 1-0 **ADFVR<1:0>:** ADC FVR Buffer Gain Selection bit
11 = ADC FVR Buffer Gain is 4x, with output $V_{ADFVR} = 4x\ V_{FVR}$(2)
10 = ADC FVR Buffer Gain is 2x, with output $V_{ADFVR} = 2x\ V_{FVR}$(2)
01 = ADC FVR Buffer Gain is 1x, with output $V_{ADFVR} = 1x\ V_{FVR}$
00 = ADC FVR Buffer is off

**Note 1:** FVRRDY is always '1' on PIC16F1703/7 only.
**2:** Fixed Voltage Reference output cannot exceed $V_{DD}$.
**3:** See **Section 15.0 "Temperature Indicator Module"** for additional information.

**TABLE 14-2: SUMMARY OF REGISTERS ASSOCIATED WITH FIXED VOLTAGE REFERENCE**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on page |
|---|---|---|---|---|---|---|---|---|---|
| FVRCON | FVREN | FVRRDY | TSEN | TSRNG | CDAFVR<1:0> | | ADFVR<1:0> | | 142 |

**Legend:** Shaded cells are not used with the Fixed Voltage Reference.


© 2013-2015 Microchip Technology Inc.

## 15.0 TEMPERATURE INDICATOR MODULE

This family of devices is equipped with a temperature circuit designed to measure the operating temperature of the silicon die. The circuit's range of operating temperature falls between -40°C and +85°C. The output is a voltage that is proportional to the device temperature. The output of the temperature indicator is internally connected to the device ADC.

The circuit may be used as a temperature threshold detector or a more accurate temperature indicator, depending on the level of calibration performed. A one-point calibration allows the circuit to indicate a temperature closely surrounding that point. A two-point calibration allows the circuit to sense the entire range of temperature more accurately. Reference Application Note AN1333, "*Use and Calibration of the Internal Temperature Indicator*" (DS01333) for more details regarding the calibration process.

### 15.1 Circuit Operation

Figure 15-1 shows a simplified block diagram of the temperature circuit. The proportional voltage output is achieved by measuring the forward voltage drop across multiple silicon junctions.

Equation 15-1 describes the output characteristics of the temperature indicator.

**EQUATION 15-1: VOUT RANGES**

High Range: $V_{OUT} = V_{DD} - 4V_T$

Low Range: $V_{OUT} = V_{DD} - 2V_T$

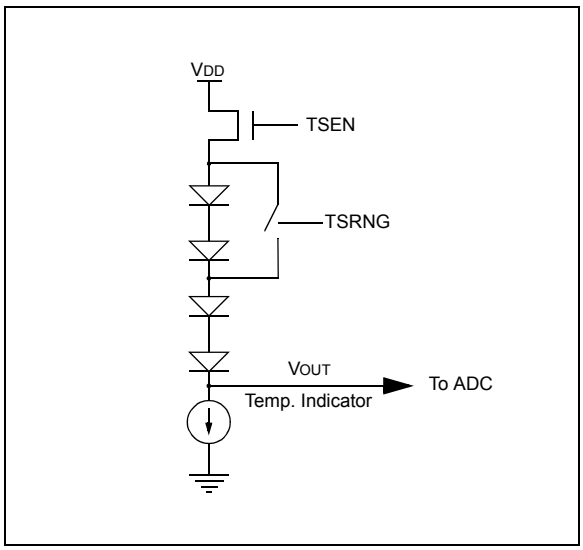
The temperature sense circuit is integrated with the Fixed Voltage Reference (FVR) module. See **Section 14.0 "Fixed Voltage Reference (FVR)"** for more information.

The circuit is enabled by setting the TSEN bit of the FVRCON register. When disabled, the circuit draws no current.

The circuit operates in either high or low range. The high range, selected by setting the TSRNG bit of the FVRCON register, provides a wider output voltage. This provides more resolution over the temperature range, but may be less consistent from part to part. This range requires a higher bias voltage to operate and thus, a higher VDD is needed.

The low range is selected by clearing the TSRNG bit of the FVRCON register. The low range generates a lower voltage drop and thus, a lower bias voltage is needed to operate the circuit. The low range is provided for low voltage operation.

**FIGURE 15-1: TEMPERATURE CIRCUIT DIAGRAM**




### 15.2 Minimum Operating VDD

When the temperature circuit is operated in low range, the device may be operated at any operating voltage that is within specifications.

When the temperature circuit is operated in high range, the device operating voltage, VDD, must be high enough to ensure that the temperature circuit is correctly biased.

Table 15-1 shows the recommended minimum VDD vs. range setting.

**TABLE 15-1: RECOMMENDED VDD VS. RANGE**

| Min. VDD, TSRNG = 1 | Min. VDD, TSRNG = 0 |
|---|---|
| 3.6V | 1.8V |

### 15.3 Temperature Output

The output of the circuit is measured using the internal Analog-to-Digital Converter. A channel is reserved for the temperature circuit output. Refer to **Section 16.0 "Analog-to-Digital Converter (ADC) Module"** for detailed information.

## 15.4 ADC Acquisition Time

To ensure accurate temperature measurements, the user must wait at least 200 μs after the ADC input multiplexer is connected to the temperature indicator output before the conversion is performed. In addition, the user must wait 200 μs between sequential conversions of the temperature indicator output.

**TABLE 15-2: SUMMARY OF REGISTERS ASSOCIATED WITH THE TEMPERATURE INDICATOR**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on page |
|---|---|---|---|---|---|---|---|---|---|
| FVRCON | FVREN | FVRRDY | TSEN | TSRNG | CDAFVR<1:0> | | ADFVR<1:0> | | 142 |

**Legend:** Shaded cells are unused by the temperature indicator module.


© 2013-2015 Microchip Technology Inc.

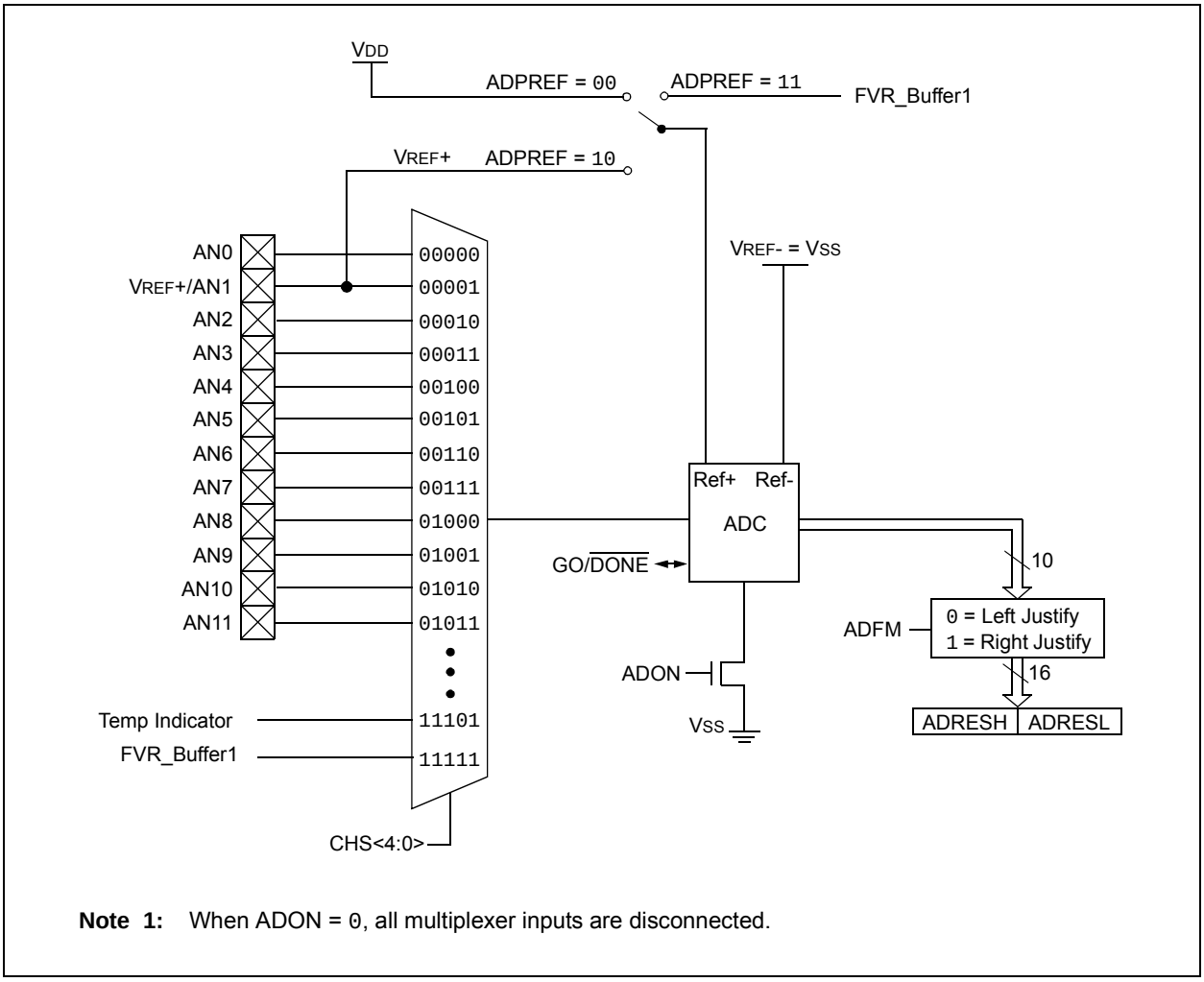
## 16.0 ANALOG-TO-DIGITAL CONVERTER (ADC) MODULE

The Analog-to-Digital Converter (ADC) allows conversion of an analog input signal to a 10-bit binary representation of that signal. This device uses analog inputs, which are multiplexed into a single sample and hold circuit. The output of the sample and hold is connected to the input of the converter. The converter generates a 10-bit binary result via successive approximation and stores the conversion result into the ADC result registers (ADRESH:ADRESL register pair). Figure 16-1 shows the block diagram of the ADC.

The ADC voltage reference is software selectable to be either internally generated or externally supplied.

The ADC can generate an interrupt upon completion of a conversion. This interrupt can be used to wake-up the device from Sleep.

**FIGURE 16-1: ADC BLOCK DIAGRAM**




**Note 1:** When ADON = 0, all multiplexer inputs are disconnected.

## 16.1 ADC Configuration

When configuring and using the ADC the following functions must be considered:

- Port configuration
- Channel selection
- ADC voltage reference selection
- ADC conversion clock source
- Interrupt control
- Result formatting

### 16.1.1 PORT CONFIGURATION

The ADC can be used to convert both analog and digital signals. When converting analog signals, the I/O pin should be configured for analog by setting the associated TRIS and ANSEL bits. Refer to **Section 11.0 "I/O Ports"** for more information.

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

### 16.1.2 CHANNEL SELECTION

There are up to 17 channel selections available:

- AN<13:8, 4:0> pins (PIC16(L)F1703 only)
- AN<21,13:0> pins (PIC16(L)F1707 only)
- Temperature Indicator
- FVR_buffer1

The CHS bits of the ADCON0 register (Register 16-1) determine which channel is connected to the sample and hold circuit.

When changing channels, a delay is required before starting the next conversion. Refer to **Section 16.2 "ADC Operation"** for more information.

### 16.1.3 ADC VOLTAGE REFERENCE

The ADPREF bits of the ADCON1 register provides control of the positive voltage reference. The positive voltage reference can be:

- $V_{REF}$+ pin
- $V_{DD}$
- FVR 2.048V
- FVR 4.096V (Not available on LF devices)
- $V_{SS}$

See **Section 16.0 "Analog-to-Digital Converter (ADC) Module"** for more details on the Fixed Voltage Reference.

### 16.1.4 CONVERSION CLOCK

The source of the conversion clock is software selectable via the ADCS bits of the ADCON1 register. There are seven possible clock options:

- $F_{OSC}$/2
- $F_{OSC}$/4
- $F_{OSC}$/8
- $F_{OSC}$/16
- $F_{OSC}$/32
- $F_{OSC}$/64
- FRC (internal RC oscillator)

The time to complete one bit conversion is defined as $T_{AD}$. One full 10-bit conversion requires 11.5 $T_{AD}$ periods as shown in Figure 16-2.

For correct conversion, the appropriate $T_{AD}$ specification must be met. Refer to Table 26-15 for more information. Table 16-1 gives examples of appropriate ADC clock selections.

| | |
|---|---|
| **Note:** | Unless using the FRC, any changes in the system clock frequency will change the ADC clock frequency, which may adversely affect the ADC result. |


© 2013-2015 Microchip Technology Inc.

**TABLE 16-1: ADC CLOCK PERIOD ($T_{AD}$) VS. DEVICE OPERATING FREQUENCIES**

| ADC Clock Period ($T_{AD}$) | | Device Frequency ($F_{OSC}$) | | | | | |
|---|---|---|---|---|---|---|---|
| ADC Clock Source | ADCS<2:0> | 32 MHz | 20 MHz | 16 MHz | 8 MHz | 4 MHz | 1 MHz |
| $F_{OSC}$/2 | 000 | 62.5ns(2) | 100 ns(2) | 125 ns(2) | 250 ns(2) | 500 ns(2) | 2.0 µs |
| $F_{OSC}$/4 | 100 | 125 ns(2) | 200 ns(2) | 250 ns(2) | 500 ns(2) | 1.0 µs | 4.0 µs |
| $F_{OSC}$/8 | 001 | 0.5 µs(2) | 400 ns(2) | 0.5 µs(2) | 1.0 µs | 2.0 µs | 8.0 µs(3) |
| $F_{OSC}$/16 | 101 | 800 ns | 800 ns | 1.0 µs | 2.0 µs | 4.0 µs | 16.0 µs(3) |
| $F_{OSC}$/32 | 010 | 1.0 µs | 1.6 µs | 2.0 µs | 4.0 µs | 8.0 µs(3) | 32.0 µs(2) |
| $F_{OSC}$/64 | 110 | 2.0 µs | 3.2 µs | 4.0 µs | 8.0 µs(3) | 16.0 µs(2) | 64.0 µs(2) |
| FRC | x11 | 1.0-6.0 µs(1,4) | 1.0-6.0 µs(1,4) | 1.0-6.0 µs(1,4) | 1.0-6.0 µs(1,4) | 1.0-6.0 µs(1,4) | 1.0-6.0 µs(1,4) |

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

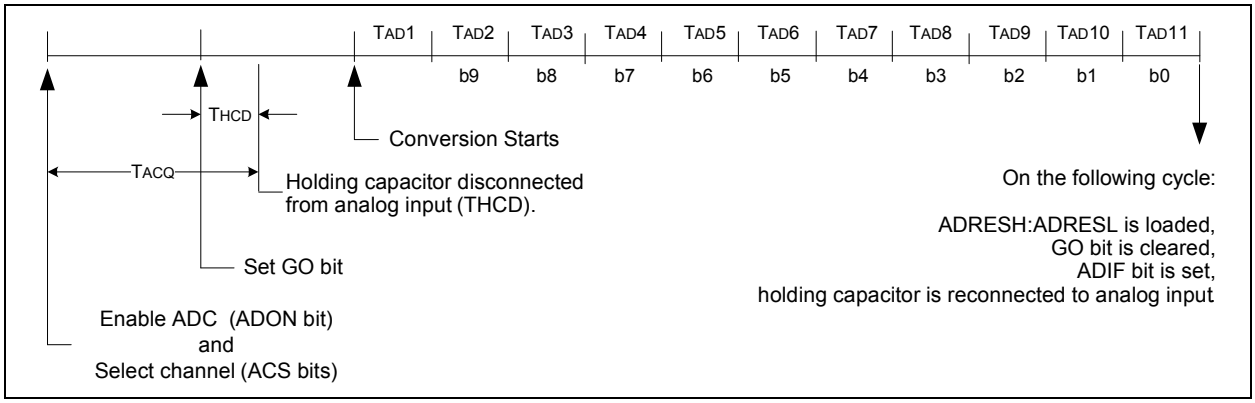
**Note 1:** See $T_{AD}$ parameter for FRC source typical $T_{AD}$ value.

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

**3:** Outside the recommended $T_{AD}$ time.

**4:** The ADC clock period ($T_{AD}$) and total ADC conversion time can be minimized when the ADC clock is derived from the system clock $F_{OSC}$. However, the FRC oscillator source must be used when conversions are to be performed with the device in Sleep mode.

**FIGURE 16-2: ANALOG-TO-DIGITAL CONVERSION $T_{AD}$ CYCLES**



### 16.1.5 INTERRUPTS

The ADC module allows for the ability to generate an interrupt upon completion of an Analog-to-Digital conversion. The ADC Interrupt Flag is the ADIF bit in the PIR1 register. The ADC Interrupt Enable is the ADIE bit in the PIE1 register. The ADIF bit must be cleared in software.

| | |
|---|---|
| **Note 1:** | The ADIF bit is set at the completion of every conversion, regardless of whether or not the ADC interrupt is enabled. |
| **2:** | The ADC operates during Sleep only when the FRC oscillator is selected. |

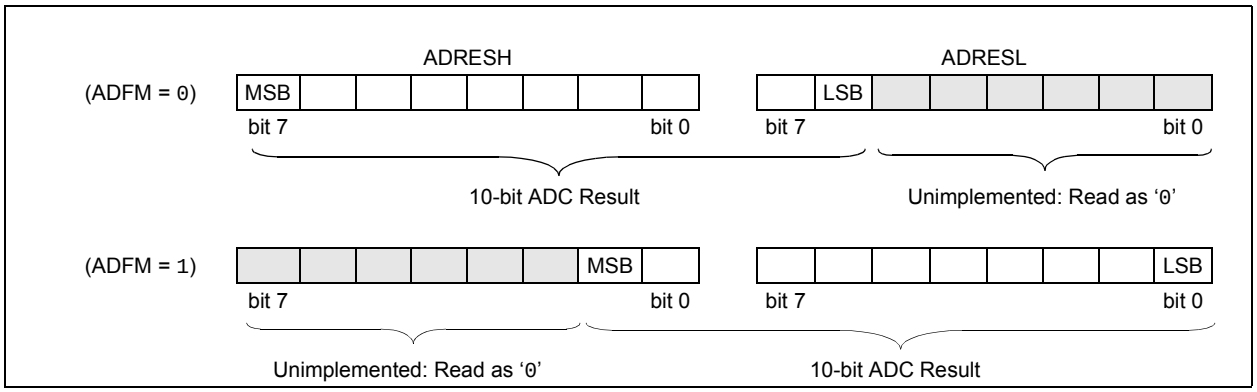
This interrupt can be generated while the device is operating or while in Sleep. If the device is in Sleep, the interrupt will wake-up the device. Upon waking from Sleep, the next instruction following the SLEEP instruction is always executed. If the user is attempting to wake-up from Sleep and resume in-line code execution, the ADIE bit of the PIE1 register and the PEIE bit of the INTCON register must both be set and the GIE bit of the INTCON register must be cleared. If all three of these bits are set, the execution will switch to the Interrupt Service Routine.

### 16.1.6 RESULT FORMATTING

The 10-bit ADC conversion result can be supplied in two formats, left justified or right justified. The ADFM bit of the ADCON1 register controls the output format.

Figure 16-3 shows the two output formats.

**FIGURE 16-3: 10-BIT ADC CONVERSION RESULT FORMAT**





© 2013-2015 Microchip Technology Inc.

## 16.2 ADC Operation

### 16.2.1 STARTING A CONVERSION

To enable the ADC module, the ADON bit of the ADCON0 register must be set to a '1'. Setting the GO/DONE bit of the ADCON0 register to a '1' will start the Analog-to-Digital conversion.

| | |
|---|---|
| **Note:** | The GO/DONE bit should not be set in the same instruction that turns on the ADC. Refer to **Section 16.2.6 "ADC Conversion Procedure"**. |

### 16.2.2 COMPLETION OF A CONVERSION

When the conversion is complete, the ADC module will:

- Clear the GO/DONE bit
- Set the ADIF Interrupt Flag bit
- Update the ADRESH and ADRESL registers with new conversion result

### 16.2.3 TERMINATING A CONVERSION

If a conversion must be terminated before completion, the GO/DONE bit can be cleared in software. The ADRESH and ADRESL registers will be updated with the partially complete Analog-to-Digital conversion sample. Incomplete bits will match the last bit converted.

| | |
|---|---|
| **Note:** | A device Reset forces all registers to their Reset state. Thus, the ADC module is turned off and any pending conversion is terminated. |

### 16.2.4 ADC OPERATION DURING SLEEP

The ADC module can operate during Sleep. This requires the ADC clock source to be set to the FRC option. When the FRC oscillator source is selected, the ADC waits one additional instruction before starting the conversion. This allows the SLEEP instruction to be executed, which can reduce system noise during the conversion. If the ADC interrupt is enabled, the device will wake-up from Sleep when the conversion completes. If the ADC interrupt is disabled, the ADC module is turned off after the conversion completes, although the ADON bit remains set.

When the ADC clock source is something other than FRC, a SLEEP instruction causes the present conversion to be aborted and the ADC module is turned off, although the ADON bit remains set.

### 16.2.5 AUTO-CONVERSION TRIGGER

The Auto-conversion Trigger allows periodic ADC measurements without software intervention. When a rising edge of the selected source occurs, the GO/DONE bit is set by hardware.

The Auto-conversion Trigger source is selected with the TRIGSEL<3:0> bits of the ADCON2 register.

Using the Auto-conversion Trigger does not assure proper ADC timing. It is the user's responsibility to ensure that the ADC timing requirements are met.

See Table 16-2 for auto-conversion sources.

**TABLE 16-2: AUTO-CONVERSION SOURCES**

| Source Peripheral | Signal Name |
|---|---|
| CCP1 | |
| CCP2 | |
| Timer0 | T0_overflow |
| Timer1 | T1_overflow |
| Timer2 | T2_match |

### 16.2.6 ADC CONVERSION PROCEDURE

This is an example procedure for using the ADC to perform an Analog-to-Digital conversion:

1. Configure Port:
   - Disable pin output driver (Refer to the TRIS register)
   - Configure pin as analog (Refer to the ANSEL register)
2. Configure the ADC module:
   - Select ADC conversion clock
   - Configure voltage reference
   - Select ADC input channel
   - Turn on ADC module
3. Configure ADC interrupt (optional):
   - Clear ADC interrupt flag
   - Enable ADC interrupt
   - Enable peripheral interrupt
   - Enable global interrupt(1)
4. Wait the required acquisition time(2).
5. Start conversion by setting the GO/$\overline{\text{DONE}}$ bit.
6. Wait for ADC conversion to complete by one of the following:
   - Polling the GO/$\overline{\text{DONE}}$ bit
   - Waiting for the ADC interrupt (interrupts enabled)
7. Read ADC Result.
8. Clear the ADC interrupt flag (required if interrupt is enabled).

**Note 1:** The global interrupt can be disabled if the user is attempting to wake-up from Sleep and resume in-line code execution.

**2:** Refer to **Section 16.4 "ADC Acquisition Requirements"**.

**EXAMPLE 16-1: ADC CONVERSION**

```
;This code block configures the ADC
;for polling, Vdd and Vss references, FRC
;oscillator and AN0 input.
;
;Conversion start & polling for completion
; are included.
;
BANKSEL   ADCON1       ;
MOVLW     B'11110000'  ;Right justify, FRC
                       ;oscillator
MOVWF     ADCON1       ;Vdd and Vss Vref
BANKSEL   TRISA        ;
BSF       TRISA,0      ;Set RA0 to input
BANKSEL   ANSEL        ;
BSF       ANSEL,0      ;Set RA0 to analog
BANKSEL   ADCON0       ;
MOVLW     B'00000001'  ;Select channel AN0
MOVWF     ADCON0       ;Turn ADC On
CALL      SampleTime   ;Acquisiton delay
BSF       ADCON0,ADGO  ;Start conversion
BTFSC     ADCON0,ADGO  ;Is conversion done?
GOTO      $-1          ;No, test again
BANKSEL   ADRESH       ;
MOVF      ADRESH,W     ;Read upper 2 bits
MOVWF     RESULTHI     ;store in GPR space
BANKSEL   ADRESL       ;
MOVF      ADRESL,W     ;Read lower 8 bits
MOVWF     RESULTLO     ;Store in GPR space
```


© 2013-2015 Microchip Technology Inc.

## 16.3 Register Definitions: ADC Control

### REGISTER 16-1: ADCON0: ADC CONTROL REGISTER 0

| U-0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| — | CHS<4:0> | | | | | GO/$\overline{\text{DONE}}$ | ADON |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| '1' = Bit is set | '0' = Bit is cleared | |

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

bit 6-2 **CHS<4:0>:** Analog Channel Select bits

11111 = FVR (Fixed Voltage Reference) Buffer 1 Output(1)
11110 = Reserved
11101 = Temperature Indicator(2)
11100 = Reserved. No channel connected.
•
•
•
01100 = Reserved. No channel connected.
01011 = AN11
01010 = AN10
01001 = AN9
01000 = AN8
00111 = AN7
00110 = AN6
00101 = AN5
00100 = AN4
00011 = AN3
00010 = AN2
00001 = AN1
00000 = AN0

bit 1 **GO/$\overline{\text{DONE}}$:** ADC Conversion Status bit

1 = ADC conversion cycle in progress. Setting this bit starts an ADC conversion cycle.
This bit is automatically cleared by hardware when the ADC conversion has completed.
0 = ADC conversion completed/not in progress

bit 0 **ADON:** ADC Enable bit

1 = ADC is enabled
0 = ADC is disabled and consumes no operating current

**Note 1:** See **Section 14.0 "Fixed Voltage Reference (FVR)"** for more information.
**2:** See **Section 15.0 "Temperature Indicator Module"** for more information.

**REGISTER 16-2: ADCON1: ADC CONTROL REGISTER 1**

| R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | U-0 | R/W-0/0 | R/W-0/0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| ADFM | ADCS<2:0> | | | — | ADNREF | ADPREF<1:0> | |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| '1' = Bit is set | '0' = Bit is cleared | |

bit 7 **ADFM:** ADC Result Format Select bit
- 1 = Right justified. Six Most Significant bits of ADRESH are set to '0' when the conversion result is loaded.
- 0 = Left justified. Six Least Significant bits of ADRESL are set to '0' when the conversion result is loaded.

bit 6-4 **ADCS<2:0>:** ADC Conversion Clock Select bits
- 111 = FRC (clock supplied from an internal RC oscillator)
- 110 = FOSC/64
- 101 = FOSC/16
- 100 = FOSC/4
- 011 = FRC (clock supplied from an internal RC oscillator)
- 010 = FOSC/32
- 001 = FOSC/8
- 000 = FOSC/2

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

bit 2 **ADNREF:** ADC Negative Voltage Reference Configuration bit
- 0 = VREF- is connected to VSS
- 1 = VREF- is connected to VREF- pin

bit 1-0 **ADPREF<1:0>:** ADC Positive Voltage Reference Configuration bits
- 11 = VREF+ is connected to internal FVR_Buffer1(1)
- 10 = VREF+ is connected to external VREF+ pin(1)
- 01 = Reserved
- 00 = VREF+ is connected to VDD

**Note 1:** When selecting the VREF+ pin as the source of the positive reference, be aware that a minimum voltage specification exists. See Table 26-15 for details.


© 2013-2015 Microchip Technology Inc.

**REGISTER 16-3: ADCON2: ADC CONTROL REGISTER 2**

| R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | U-0 | U-0 | U-0 | U-0 |
|---|---|---|---|---|---|---|---|
| TRIGSEL<3:0>(1) | | | | — | — | — | — |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| '1' = Bit is set | '0' = Bit is cleared | |

bit 7-4 **TRIGSEL<3:0>:** Auto-Conversion Trigger Selection bits(1)

0000 = No auto-conversion trigger selected
0001 = CCP1
0010 = CCP2
0011 = Timer0 – T0_overflow(2)
0100 = Timer1 – T1_overflow(2)
0101 = Timer2 – T2_match
0110 = Reserved
0111 = Reserved
1000 = Reserved
1001 = Reserved
1010 = Reserved
1011 = Reserved
1100 = Reserved
1101 = Reserved
1110 = Reserved
1111 = Reserved

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

**Note 1:** This is a rising edge sensitive input for all sources.
**2:** Signal also sets its corresponding interrupt flag.

**REGISTER 16-4: ADRESH: ADC RESULT REGISTER HIGH (ADRESH) ADFM = 0**

| R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u |
|---|---|---|---|---|---|---|---|
| ADRES<9:2> | | | | | | | |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as ‘0’ |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| ‘1’ = Bit is set | ‘0’ = Bit is cleared | |

bit 7-0 **ADRES<9:2>**: ADC Result Register bits
Upper eight bits of 10-bit conversion result

**REGISTER 16-5: ADRESL: ADC RESULT REGISTER LOW (ADRESL) ADFM = 0**

| R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u |
|---|---|---|---|---|---|---|---|
| ADRES<1:0> | | — | — | — | — | — | — |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as ‘0’ |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| ‘1’ = Bit is set | ‘0’ = Bit is cleared | |

bit 7-6 **ADRES<1:0>**: ADC Result Register bits
Lower two bits of 10-bit conversion result

bit 5-0 **Reserved**: Do not use.


© 2013-2015 Microchip Technology Inc.

**REGISTER 16-6: ADRESH: ADC RESULT REGISTER HIGH (ADRESH) ADFM = 1**

| R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u |
|---|---|---|---|---|---|---|---|
| — | — | — | — | — | — | ADRES<9:8> | |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as ‘0’ |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| ‘1’ = Bit is set | ‘0’ = Bit is cleared | |

bit 7-2 **Reserved**: Do not use.

bit 1-0 **ADRES<9:8>**: ADC Result Register bits
Upper two bits of 10-bit conversion result

**REGISTER 16-7: ADRESL: ADC RESULT REGISTER LOW (ADRESL) ADFM = 1**

| R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u |
|---|---|---|---|---|---|---|---|
| ADRES<7:0> | | | | | | | |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as ‘0’ |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| ‘1’ = Bit is set | ‘0’ = Bit is cleared | |

bit 7-0 **ADRES<7:0>**: ADC Result Register bits
Lower eight bits of 10-bit conversion result

## 16.4 ADC Acquisition Requirements

For the ADC to meet its specified accuracy, the charge holding capacitor (CHOLD) must be allowed to fully charge to the input channel voltage level. The Analog Input model is shown in Figure 16-4. The source impedance (RS) and the internal sampling switch (RSS) impedance directly affect the time required to charge the capacitor CHOLD. The sampling switch (RSS) impedance varies over the device voltage (VDD), refer to Figure 16-4. **The maximum recommended impedance for analog sources is 10 kΩ.** As the source impedance is decreased, the acquisition time may be decreased. After the analog input channel is selected (or changed), an ADC acquisition must be done before the conversion can be started. To calculate the minimum acquisition time, Equation 16-1 may be used. This equation assumes that 1/2 LSb error is used (1,024 steps for the ADC). The 1/2 LSb error is the maximum error allowed for the ADC to meet its specified resolution.

**EQUATION 16-1: ACQUISITION TIME EXAMPLE**

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

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

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

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

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

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

*Note: Where n = number of bits of the ADC.*

*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:*

$$T_{ACQ} = 2\mu s + 892ns + [(50°C - 25°C)(0.05\mu s/°C)]$$
$$= 4.62\mu s$$

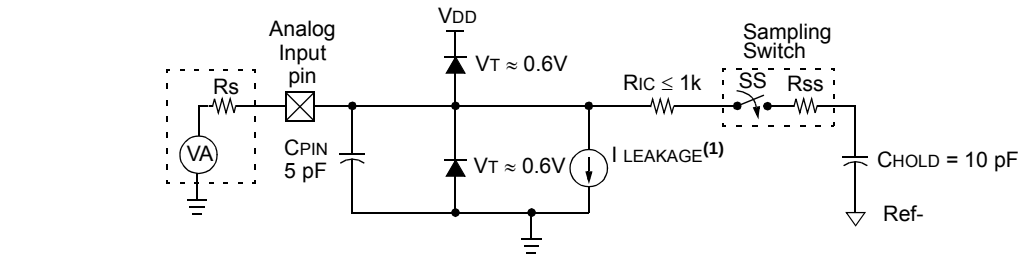
**Note 1:** The reference voltage (VREF) has no effect on the equation, since it cancels itself out.

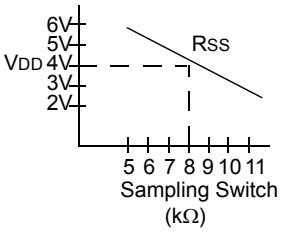
**2:** The charge holding capacitor (CHOLD) is not discharged after each conversion.

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


© 2013-2015 Microchip Technology Inc.

**FIGURE 16-4: ANALOG INPUT MODEL**




**Legend:**

| | | |
|---|---|---|
| CHOLD | = | Sample/Hold Capacitance |
| CPIN | = | Input Capacitance |
| I LEAKAGE | = | Leakage current at the pin due to various junctions |
| RIC | = | Interconnect Resistance |
| RSS | = | Resistance of Sampling Switch |
| SS | = | Sampling Switch |
| VT | = | Threshold Voltage |

**Note 1:** Refer to Table 26-4 (parameter D060).

**FIGURE 16-5: ADC TRANSFER FUNCTION**

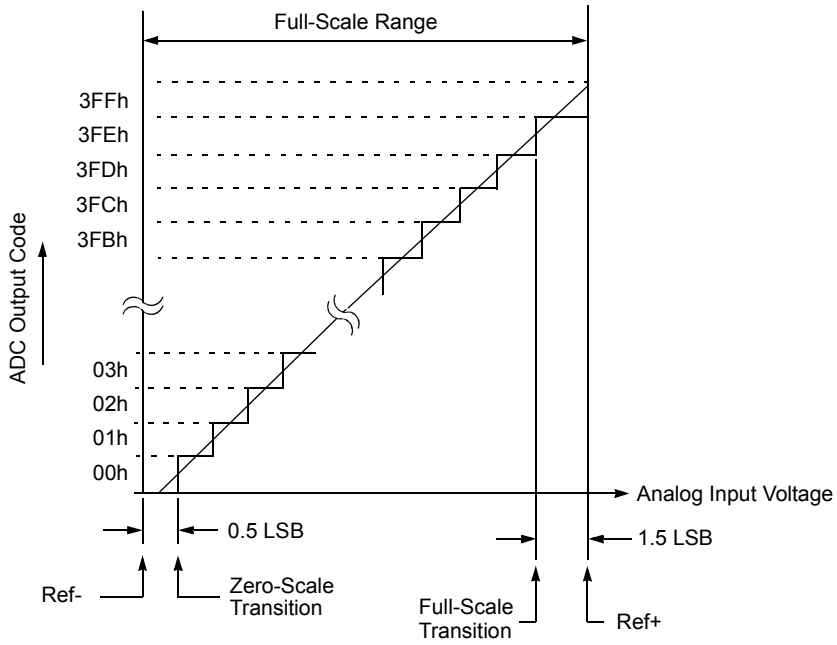


**TABLE 16-3: SUMMARY OF REGISTERS ASSOCIATED WITH ADC**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|
| ADCON0 | — | CHS<4:0> | | | | | GO/$\overline{\text{DONE}}$ | ADON | 151 |
| ADCON1 | ADFM | ADCS<2:0> | | | ADNREF<1:0> | | ADPREF<1:0> | | 152 |
| ADCON2 | TRIGSEL<3:0> | | | | — | — | — | — | 153 |
| ADRESH | ADRES<7:0> | | | | | | | | 154, 155 |
| ADRESL | ADRES<7:0> | | | | | | | | 154, 155 |
| ANSELA | — | — | — | ANSA4 | — | ANSA2 | ANSA1 | ANSA0 | 114 |
| ANSELB(1) | — | — | ANSB5 | ANSB4 | — | — | — | — | 120 |
| ANSELC | ANSC7(1) | ANSC6(1) | ANSC5(2) | ANSC4(2) | ANSC3 | ANSC2 | ANSC1 | ANSC0 | 125 |
| INTCON | GIE | PEIE | TMR0IE | INTE | IOCIE | TMR0IF | INTF | IOCIF | 81 |
| PIE1 | TMR1GIE | ADIE | — | — | SSP1IE | CCP1IE | TMR2IE | TMR1IE | 78 |
| PIR1 | TMR1GIF | ADIF | — | — | SSP1IF | CCP1IF | TMR2IF | TMR1IF | 81 |
| TRISA | — | — | TRISA5 | TRISA4 | —(3) | TRISA2 | TRISA1 | TRISA0 | 113 |
| TRISB(1) | TRISB7 | TRISB6 | TRISB5 | TRISB4 | — | — | — | — | 119 |
| TRISC | TRISC7(1) | TRISC6(1) | TRISC5 | TRISC4 | TRISC3 | TRISC2 | TRISC1 | TRISC0 | 124 |

**Legend:** x = unknown, u = unchanged, – = unimplemented read as '0', q = value depends on condition. Shaded cells are not used for the ADC module.

**Note 1:** PIC16(L)F1707 only.

**2:** PIC16(L)F1703 only.

**3:** Unimplemented, read as '1'.

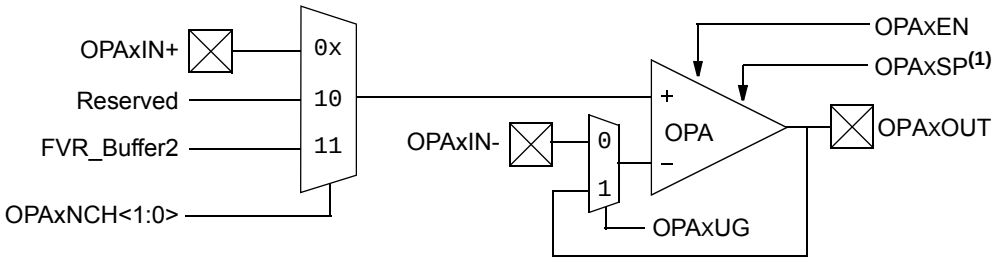

© 2013-2015 Microchip Technology Inc.

## 17.0 OPERATIONAL AMPLIFIER (OPA) MODULES

The Operational Amplifier (OPA) is a standard 3-terminal device requiring external feedback to operate. The OPA module has the following features:

- External connections to I/O ports
- Low leakage inputs
- Factory Calibrated Input Offset Voltage

**FIGURE 17-1: OPAx MODULE BLOCK DIAGRAM**




**Note 1:** The OPAxSP bit must be set. Low-Power mode is not supported.

## 17.1 OPA Module Performance

Common AC and DC performance specifications for the OPA module:

- Common-Mode Voltage Range
- Leakage Current
- Input Offset Voltage
- Open Loop Gain
- Gain Bandwidth Product

**Common-mode voltage range** is the specified voltage range for the OPA+ and OPA- inputs, for which the OPA module will perform to within its specifications. The OPA module is designed to operate with input voltages between VSS and VDD. Behavior for common-mode voltages greater than VDD, or below VSS, are not guaranteed.

**Leakage current** is a measure of the small source or sink currents on the OPA+ and OPA- inputs. To minimize the effect of leakage currents, the effective impedances connected to the OPA+ and OPA- inputs should be kept as small as possible and equal.

**Input offset voltage** is a measure of the voltage difference between the OPA+ and OPA- inputs in a closed loop circuit with the OPA in its linear region. The offset voltage will appear as a DC offset in the output equal to the input offset voltage, multiplied by the gain of the circuit. The input offset voltage is also affected by the common-mode voltage. The OPA is factory calibrated to minimize the input offset voltage of the module.

**Open loop gain** is the ratio of the output voltage to the differential input voltage, (OPA+) - (OPA-). The gain is greatest at DC and falls off with frequency.

**Gain Bandwidth Product** or GBWP is the frequency at which the open loop gain falls off to 0 dB.

### 17.1.1 OPA Module Control

The OPA module is enabled by setting the OPAxEN bit of the OPAxCON register. When enabled, the OPA forces the output driver of OPAxOUT pin into tri-state to prevent contention between the driver and the OPA output.

| | |
|---|---|
| **Note:** | When the OPA module is enabled, the OPAxOUT pin is driven by the Op Amp output, not by the PORT digital driver. Refer to Table 26-16 for the Op Amp output drive capability. |

### 17.1.2 UNITY GAIN MODE

The OPAxUG bit of the OPAxCON register selects the Unity Gain mode. When unity gain is selected, the OPA output is connected to the inverting input and the OPAxIN pin is relinquished, releasing the pin for general purpose input and output.

## 17.2 Effects of Reset

A device Reset forces all registers to their Reset state. This disables the OPA module.


© 2013-2015 Microchip Technology Inc.

## 17.3 Register Definitions: Op Amp Control

### REGISTER 17-1: OPAxCON: OPERATIONAL AMPLIFIERS (OPAx) CONTROL REGISTERS

| R/W-0/0 | R/W-0/0 | U-0 | R/W-0/0 | U-0 | U-0 | R/W-0/0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| OPAxEN | OPAxSP | — | OPAxUG | — | — | OPAxPCH<1:0> | |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| '1' = Bit is set | '0' = Bit is cleared | q = Value depends on condition |

bit 7 **OPAxEN:** Op Amp Enable bit
1 = Op Amp is enabled
0 = Op Amp is disabled and consumes no active power

bit 6 **OPAxSP:** Op Amp Speed/Power Select bit
1 = Op Amp operates in high GBWP mode
0 = Reserved. Do not use.

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

bit 4 **OPAxUG:** Op Amp Unity Gain Select bit
1 = OPA output is connected to inverting input. OPAxIN- pin is available for general purpose I/O.
0 = Inverting input is connected to the OPAxIN- pin

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

bit 1-0 **OPAxPCH<1:0>:** Non-inverting Channel Selection bits
11 = Non-inverting input connects to FVR_Buffer2 output
10 = Reserved
0x = Non-inverting input connects to OPAxIN+ pin

### TABLE 17-1: SUMMARY OF REGISTERS ASSOCIATED WITH OP AMPS

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|
| ANSELB(1) | — | — | ANSB5 | ANSB4 | — | — | — | — | 120 |
| ANSELC | ANSC7(1) | ANSC6(1) | ANSC5(2) | ANSC4(2) | ANSC3 | ANSC2 | ANSC1 | ANSC0 | 125 |
| FVRCON | FVREN | FVRRDY | TSEN | TSRNG | CDAFVR<1:0> | | ADFVR<1:0> | | 142 |
| OPA1CON | OPA1EN | OPA1SP | — | OPA1UG | — | — | OPA1PCH<1:0> | | 161 |
| OPA2CON | OPA2EN | OPA2SP | — | OPA2UG | — | — | OPA2PCH<1:0> | | 161 |
| TRISB(1) | TRISB7 | TRISB6 | TRISB5 | TRISB4 | — | — | — | — | 119 |
| TRISC | TRISC7(1) | TRISC6(1) | TRISC5 | TRISC4 | TRISC3 | TRISC2 | TRISC1 | TRISC0 | 124 |

**Legend:** — = unimplemented location, read as '0'. Shaded cells are not used by Op Amps.

**Note 1:** PIC16(L)F1707 only.
**2:** PIC16(L)F1703 only.
**3:** Unimplemented, read as '1'.

## 18.0 ZERO-CROSS DETECTION (ZCD) MODULE

The ZCD module detects when an A/C signal crosses through the ground potential. The actual zero crossing threshold is the zero crossing reference voltage, VREF, which is typically 0.75V above ground.

The connection to the signal to be detected is through a series current limiting resistor. The module applies a current source or sink to the ZCD pin to maintain a constant voltage on the pin, thereby preventing the pin voltage from forward biasing the ESD protection diodes. When the applied voltage is greater than the reference voltage, the module sinks current. When the applied voltage is less than the reference voltage, the module sources current. The current source and sink action keeps the pin voltage constant over the full range of the applied voltage. The ZCD module is shown in the simplified block diagram Figure 18-2.

The ZCD module is useful when monitoring an A/C waveform for, but not limited to, the following purposes:

- A/C period measurement
- Accurate long term time measurement
- Dimmer phase delayed drive
- Low EMI cycle switching

### 18.1 External Resistor Selection

The ZCD module requires a current limiting resistor in series with the external voltage source. The impedance and rating of this resistor depends on the external source peak voltage. Select a resistor value that will drop all of the peak voltage when the current through the resistor is nominally 300 µA. Refer to Equation 18-1 and Figure 18-1. Make sure that the ZCD I/O pin internal weak pull-up is disabled so it does not interfere with the current source and sink.

**EQUATION 18-1: EXTERNAL RESISTOR**

$$R_{SERIES} = \frac{V_{PEAK}}{3\times10^{-4}}$$

**FIGURE 18-1: EXTERNAL VOLTAGE**

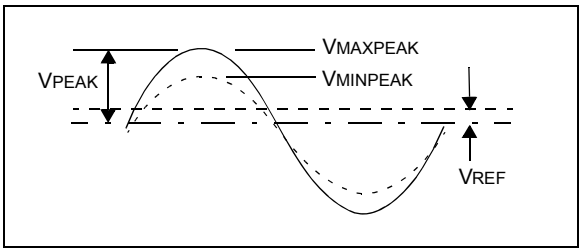
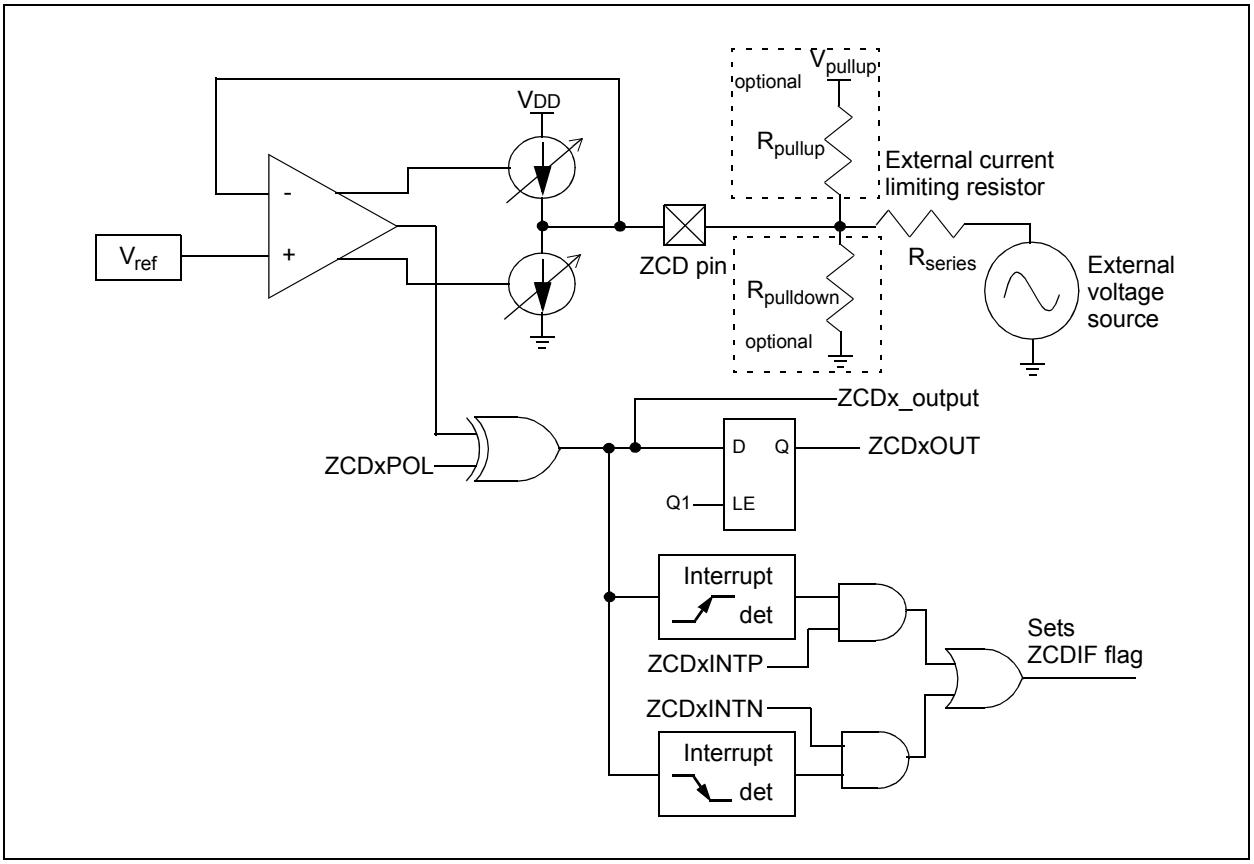



**FIGURE 18-2: SIMPLIFIED ZCD BLOCK DIAGRAM**





© 2013-2015 Microchip Technology Inc.

## 18.2 ZCD Logic Output

The ZCD module includes a Status bit, which can be read to determine whether the current source or sink is active. The ZCDxOUT bit of the ZCDCON register is set when the current sink is active, and cleared when the current source is active. The ZCDxOUT bit is affected by the polarity bit.

## 18.3 ZCD Logic Polarity

The ZCDxPOL bit of the ZCDxCON register inverts the ZCDxOUT bit relative to the current source and sink output. When the ZCDxPOL bit is set, a ZCDxOUT high indicates that the current source is active, and a low output indicates that the current sink is active.

The ZCDxPOL bit affects the ZCD interrupts. See **Section 18.4 "ZCD Interrupts"**.

## 18.4 ZCD Interrupts

An interrupt will be generated upon a change in the ZCD logic output when the appropriate interrupt enables are set. A rising edge detector and a falling edge detector are present in the ZCD for this purpose.

The ZCDIF bit of the PIR3 register will be set when either edge detector is triggered and its associated enable bit is set. The ZCDxINTP enables rising edge interrupts and the ZCDxINTN bit enables falling edge interrupts. Both are located in the ZCDxCON register.

To fully enable the interrupt, the following bits must be set:

- ZCDIE bit of the PIE3 register
- ZCDxINTP bit of the ZCDxCON register (for a rising edge detection)
- ZCDxINTN bit of the ZCDxCON register (for a falling edge detection)
- PEIE and GIE bits of the INTCON register

Changing the ZCDxPOL bit will cause an interrupt, regardless of the level of the ZCDxEN bit.

The ZCDIF bit of the PIR3 register must be cleared in software as part of the interrupt service. If another edge is detected while this flag is being cleared, the flag will still be set at the end of the sequence.

## 18.5 Correcting for VREF offset

The actual voltage at which the ZCD switches is the reference voltage at the non-inverting input of the ZCD Op Amp. For external voltage source waveforms other than square waves, this voltage offset from zero causes the zero-cross event to occur either too early or too late. When the waveform is varying relative to VSS, then the zero cross is detected too early as the waveform falls and too late as the waveform rises. When the waveform is varying relative to VDD, then the zero cross is detected too late as the waveform rises and too early as the waveform falls. The actual offset time can be determined for sinusoidal waveforms with the corresponding equations shown in Equation 18-2.

**EQUATION 18-2: ZCD EVENT OFFSET**

When External Voltage Source is relative to Vss:

$$T_{OFFSET} = \frac{\text{asin}\left(\frac{V_{REF}}{V_{PEAK}}\right)}{2\pi \bullet Freq}$$

When External Voltage Source is relative to VDD:

$$T_{OFFSET} = \frac{\text{asin}\left(\frac{V_{DD}-V_{REF}}{V_{PEAK}}\right)}{2\pi \bullet Freq}$$

This offset time can be compensated for by adding a pull-up or pull-down biasing resistor to the ZCD pin. A pull-up resistor is used when the external voltage source is varying relative to VSS. A pull-down resistor is used when the voltage is varying relative to VDD. The resistor adds a bias to the ZCD pin so that the target external voltage source must go to zero to pull the pin voltage to the VREF switching voltage. The pull-up or pull-down value can be determined with the equations shown in Equation 18-3 or Equation 18-4.

**EQUATION 18-3: ZCD PULL-UP/DOWN**

When External Signal is relative to Vss:

$$R_{PULLUP} = \frac{R_{SERIES}(V_{PULLUP} - V_{REF})}{V_{REF}}$$

When External Signal is relative to VDD:

$$R_{PULLDOWN} = \frac{R_{SERIES}(V_{REF})}{(V_{DD} - V_{REF})}$$

The pull-up and pull-down resistor values are significantly affected by small variations of VREF. Measuring VREF can be difficult, especially when the waveform is relative to VDD. However, by combining Equations 18-2 and 18-3, the resistor value can be determined from the time difference between the ZCDx_output high and low periods. Note that the time difference, ΔT, is 4*TOFFSET. The equation for determining the pull-up and pull-down resistor values from the high and low ZCDx_output periods is shown in Equation 18-4. The ZCDx_output signal can be directly observed on a pin by routing the ZCDx_output signal through one of the CLCs.

**EQUATION 18-4:**

$$R = R_{SERIES}\left(\frac{V_{BIAS}}{V_{PEAK}\left(\sin\left(\pi Freq\frac{(\Delta T)}{2}\right)\right)} - 1\right)$$

R is pull-up or pull-down resistor.

VBIAS is VPULLUP when R is pull-up or VDD when R is pull-down.

ΔT is the ZCDx_output high and low period difference.

## 18.6 Handling VPEAK variations

If the peak amplitude of the external voltage is expected to vary, the series resistor must be selected to keep the ZCD current source and sink below the design maximum range of ± 600 μA and above a reasonable minimum range. A general rule of thumb is that the maximum peak voltage can be no more than six times the minimum peak voltage. To ensure that the maximum current does not exceed ± 600 μA and the minimum is at least ± 100 μA, compute the series resistance as shown in Equation 18-5. The compensating pull-up for this series resistance can be determined with Equation 18-3 because the pull-up value is independent from the peak voltage.

**EQUATION 18-5: SERIES R FOR V RANGE**

$$R_{SERIES} = \frac{V_{MAXPEAK} + V_{MINPEAK}}{7\times10^{-4}}$$

## 18.7 Operation During Sleep

The ZCD current sources and interrupts are unaffected by Sleep.

## 18.8 Effects of a Reset

The ZCD circuit can be configured to default to the active or inactive state on Power-On-Reset (POR). When the ZCDDIS Configuration bit is cleared, the ZCD circuit will be active at POR. When the ZCDDIS Configuration bit is set, the ZCDxEN bit of the ZCDxCON register must be set to enable the ZCD module.


© 2013-2015 Microchip Technology Inc.

## 18.9 Register Definitions: ZCD Control

### REGISTER 18-1: ZCDxCON: ZERO CROSS DETECTION CONTROL REGISTER

| R/W-q/q | U-0 | R-x/x | R/W-0/0 | U-0 | U-0 | R/W-0/0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| ZCDxEN | — | ZCDxOUT | ZCDxPOL | — | — | ZCDxINTP | ZCDxINTN |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| '1' = Bit is set | '0' = Bit is cleared | q = value depends on configuration bits |

bit 7 **ZCDxEN:** Zero-Cross Detection Enable bit
1 = Zero-cross detect is enabled. ZCD pin is forced to output to source and sink current.
0 = Zero-cross detect is disabled. ZCD pin operates according to PPS and TRIS controls.

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

bit 5 **ZCDxOUT:** Zero-Cross Detection Logic Level bit
ZCDxPOL bit = 0:
1 = ZCD pin is sinking current
0 = ZCD pin is sourcing current
ZCDxPOL bit = 1:
1 = ZCD pin is sourcing current
0 = ZCD pin is sinking current

bit 4 **ZCDxPOL:** Zero-Cross Detection Logic Output Polarity bit
1 = ZCD logic output is inverted
0 = ZCD logic output is not inverted

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

bit 1 **ZCDxINTP:** Zero-Cross Positive Edge Interrupt Enable bit
1 = ZCDIF bit is set on low-to-high ZCDx_output transition
0 = ZCDIF bit is unaffected by low-to-high ZCDx_output transition

bit 0 **ZCDxINTN:** Zero-Cross Negative Edge Interrupt Enable bit
1 = ZCDIF bit is set on high-to-low ZCDx_output transition
0 = ZCDIF bit is unaffected by high-to-low ZCDx_output transition

### TABLE 18-1: SUMMARY OF REGISTERS ASSOCIATED WITH THE ZCD MODULE

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on page |
|---|---|---|---|---|---|---|---|---|---|
| PIE3 | — | — | — | ZCDIE | — | — | — | — | 80 |
| PIR3 | — | — | — | ZCDIF | — | — | — | — | 83 |
| ZCD1CON | ZCD1EN | — | ZCD1OUT | ZCD1POL | — | — | ZCD1INTP | ZCD1INTN | 165 |

**Legend:** — = unimplemented, read as '0'. Shaded cells are unused by the ZCD module.

### TABLE 18-2: SUMMARY OF CONFIGURATION BITS ASSOCIATED WITH THE ZCD MODULE

| Name | Bits | Bit -/7 | Bit -/6 | Bit 13/5 | Bit 12/4 | Bit 11/3 | Bit 10/2 | Bit 9/1 | Bit 8/0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|---|
| CONFIG2 | 13:8 | — | — | LVP | $\overline{\text{DEBUG}}$ | $\overline{\text{LPBOR}}$ | BORV | STVREN | PLLEN | 50 |
| | 7:0 | ZCDDIS | — | — | — | — | PPS1WAY | WRT<1:0> | | |

**Legend:** — = unimplemented location, read as '0'. Shaded cells are not used by ZCD module.

## 19.0 TIMER0 MODULE

The Timer0 module is an 8-bit timer/counter with the following features:

- 8-bit timer/counter register (TMR0)
- 8-bit prescaler (independent of Watchdog Timer)
- Programmable internal or external clock source
- Programmable external clock edge selection
- Interrupt on overflow
- TMR0 can be used to gate Timer1

Figure 19-1 is a block diagram of the Timer0 module.

## 19.1 Timer0 Operation

The Timer0 module can be used as either an 8-bit timer or an 8-bit counter.

### 19.1.1 8-BIT TIMER MODE

The Timer0 module will increment every instruction cycle, if used without a prescaler. 8-bit Timer mode is selected by clearing the TMR0CS bit of the OPTION_REG register.

When TMR0 is written, the increment is inhibited for two instruction cycles immediately following the write.

| | |
|---|---|
| **Note:** | The value written to the TMR0 register can be adjusted, in order to account for the two instruction cycle delay when TMR0 is written. |

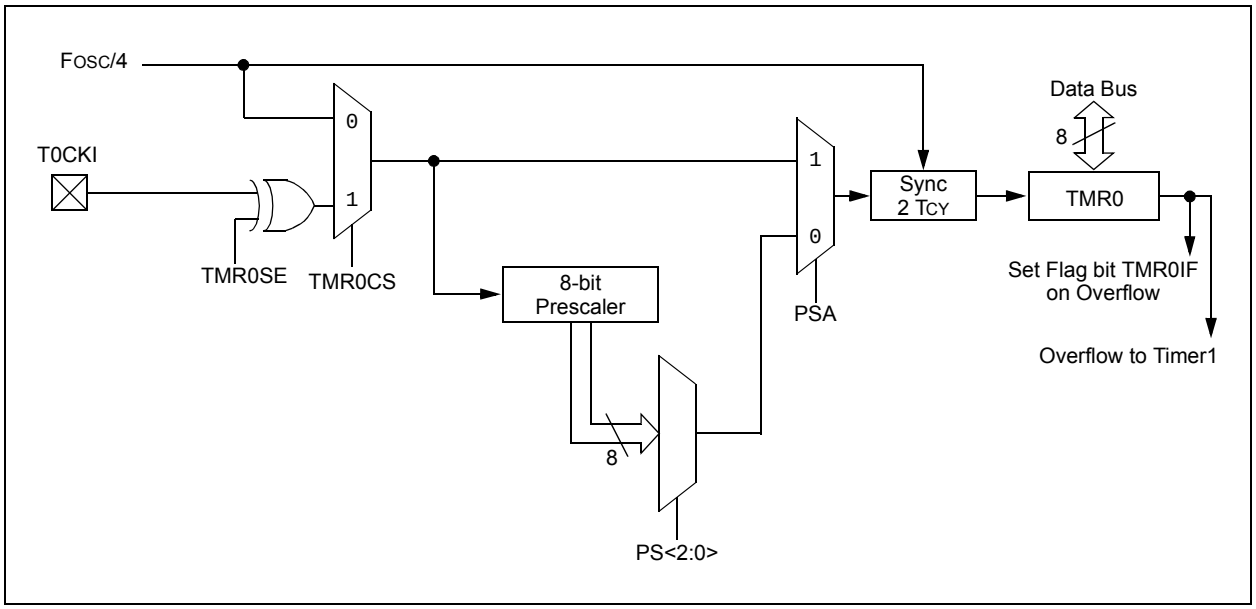
### 19.1.2 8-BIT COUNTER MODE

In 8-Bit Counter mode, the Timer0 module will increment on every rising or falling edge of the T0CKI pin.

8-Bit Counter mode using the T0CKI pin is selected by setting the TMR0CS bit in the OPTION_REG register to '1'.

The rising or falling transition of the incrementing edge for either input source is determined by the TMR0SE bit in the OPTION_REG register.

**FIGURE 19-1: BLOCK DIAGRAM OF THE TIMER0**





© 2013-2015 Microchip Technology Inc.

### 19.1.3 SOFTWARE PROGRAMMABLE PRESCALER

A software programmable prescaler is available for exclusive use with Timer0. The prescaler is enabled by clearing the PSA bit of the OPTION_REG register.

| | |
|---|---|
| **Note:** | The Watchdog Timer (WDT) uses its own independent prescaler. |

There are eight prescaler options for the Timer0 module ranging from 1:2 to 1:256. The prescale values are selectable via the PS<2:0> bits of the OPTION_REG register. In order to have a 1:1 prescaler value for the Timer0 module, the prescaler must be disabled by setting the PSA bit of the OPTION_REG register.

The prescaler is not readable or writable. All instructions writing to the TMR0 register will clear the prescaler.

### 19.1.4 TIMER0 INTERRUPT

Timer0 will generate an interrupt when the TMR0 register overflows from FFh to 00h. The TMR0IF interrupt flag bit of the INTCON register is set every time the TMR0 register overflows, regardless of whether or not the Timer0 interrupt is enabled. The TMR0IF bit can only be cleared in software. The Timer0 interrupt enable is the TMR0IE bit of the INTCON register.

| | |
|---|---|
| **Note:** | The Timer0 interrupt cannot wake the processor from Sleep since the timer is frozen during Sleep. |

### 19.1.5 8-BIT COUNTER MODE SYNCHRONIZATION

When in 8-Bit Counter mode, the incrementing edge on the T0CKI pin must be synchronized to the instruction clock. Synchronization can be accomplished by sampling the prescaler output on the Q2 and Q4 cycles of the instruction clock. The high and low periods of the external clocking source must meet the timing requirements as shown in Table 26-12.

### 19.1.6 OPERATION DURING SLEEP

Timer0 cannot operate while the processor is in Sleep mode. The contents of the TMR0 register will remain unchanged while the processor is in Sleep mode.

## 19.2 Register Definitions: Option Register

**REGISTER 19-1: OPTION_REG: OPTION REGISTER**

| R/W-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 |
|---|---|---|---|---|---|---|---|
| $\overline{\text{WPUEN}}$ | INTEDG | TMR0CS | TMR0SE | PSA | PS<2:0> | | |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| '1' = Bit is set | '0' = Bit is cleared | |

bit 7 **$\overline{\text{WPUEN}}$:** Weak Pull-Up Enable bit
1 = All weak pull-ups are disabled (except $\overline{\text{MCLR}}$, if it is enabled)
0 = Weak pull-ups are enabled by individual WPUx latch values

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

bit 5 **TMR0CS:** Timer0 Clock Source Select bit
1 = Transition on T0CKI pin
0 = Internal instruction cycle clock (FOSC/4)

bit 4 **TMR0SE:** Timer0 Source Edge Select bit
1 = Increment on high-to-low transition on T0CKI pin
0 = Increment on low-to-high transition on T0CKI pin

bit 3 **PSA:** Prescaler Assignment bit
1 = Prescaler is not assigned to the Timer0 module
0 = Prescaler is assigned to the Timer0 module

bit 2-0 **PS<2:0>:** Prescaler Rate Select bits

| Bit Value | Timer0 Rate |
|---|---|
| 000 | 1 : 2 |
| 001 | 1 : 4 |
| 010 | 1 : 8 |
| 011 | 1 : 16 |
| 100 | 1 : 32 |
| 101 | 1 : 64 |
| 110 | 1 : 128 |
| 111 | 1 : 256 |

**TABLE 19-1: SUMMARY OF REGISTERS ASSOCIATED WITH TIMER0**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|
| INTCON | GIE | PEIE | TMR0IE | INTE | IOCIE | TMR0IF | INTF | IOCIF | 77 |
| OPTION_REG | $\overline{\text{WPUEN}}$ | INTEDG | TMR0CS | TMR0SE | PSA | PS<2:0> | | | 168 |
| TMR0 | TMR0 | | | | | | | | 166* |
| TRISA | — | — | TRISA5 | TRISA4 | —(1) | TRISA2 | TRISA1 | TRISA0 | 113 |

**Legend:** — = Unimplemented location, read as '0'. Shaded cells are not used by the Timer0 module.
* Page provides register information.
**Note 1:** Unimplemented, read as '1'.


© 2013-2015 Microchip Technology Inc.

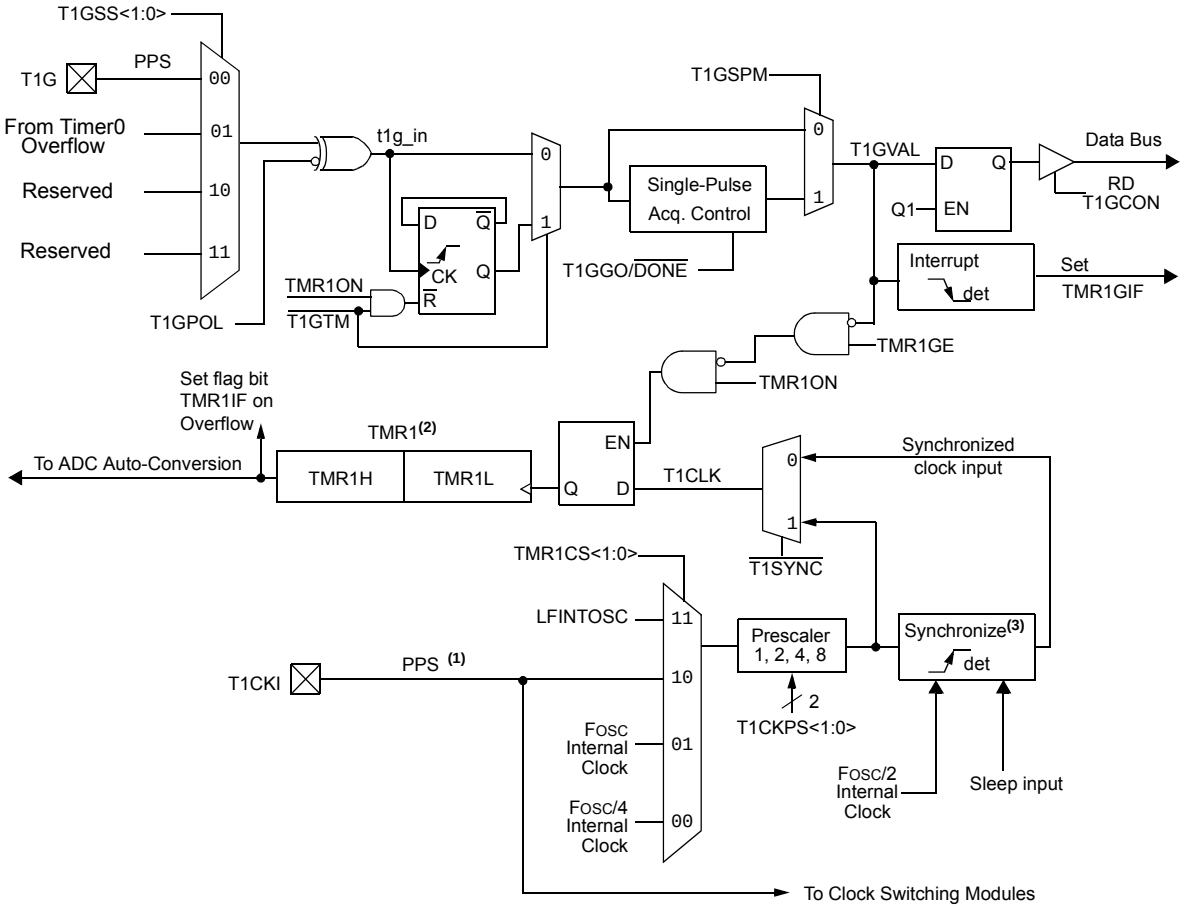
## 20.0 TIMER1 MODULE WITH GATE CONTROL

The Timer1 module is a 16-bit timer/counter with the following features:

- 16-bit timer/counter register pair (TMR1H:TMR1L)
- Programmable internal or external clock source
- 2-bit prescaler
- Dedicated 32 kHz oscillator circuit
- Multiple Timer1 gate (count enable) sources
- Interrupt on overflow
- Wake-up on overflow (external clock, Asynchronous mode only)
- Time base for the Capture/Compare function
- Auto-conversion Trigger (with CCP)
- Selectable Gate Source Polarity
- Gate Toggle mode
- Gate Single-pulse mode
- Gate Value Status
- Gate Event Interrupt

Figure 20-1 is a block diagram of the Timer1 module.

**FIGURE 20-1: TIMER1 BLOCK DIAGRAM**




**Note 1:** ST Buffer is high speed type when using T1CKI.
**2:** Timer1 register increments on rising edge.
**3:** Synchronize does not operate while in Sleep.

## 20.1 Timer1 Operation

The Timer1 module is a 16-bit incrementing counter which is accessed through the TMR1H:TMR1L register pair. Writes to TMR1H or TMR1L directly update the counter.

When used with an internal clock source, the module is a timer and increments on every instruction cycle. When used with an external clock source, the module can be used as either a timer or counter and increments on every selected edge of the external source.

Timer1 is enabled by configuring the TMR1ON and TMR1GE bits in the T1CON and T1GCON registers, respectively. Table 20-1 displays the Timer1 enable selections.

**TABLE 20-1: TIMER1 ENABLE SELECTIONS**

| TMR1ON | TMR1GE | Timer1 Operation |
|---|---|---|
| 0 | 0 | Off |
| 0 | 1 | Off |
| 1 | 0 | Always On |
| 1 | 1 | Count Enabled |

## 20.2 Clock Source Selection

The TMR1CS<1:0> bits of the T1CON register are used to select the clock source for Timer1. Table 20-2 displays the clock source selections.

### 20.2.1 INTERNAL CLOCK SOURCE

When the internal clock source is selected, the TMR1H:TMR1L register pair will increment on multiples of FOSC as determined by the Timer1 prescaler.

When the FOSC internal clock source is selected, the Timer1 register value will increment by four counts every instruction clock cycle. Due to this condition, a 2 LSB error in resolution will occur when reading the Timer1 value. To utilize the full resolution of Timer1, an asynchronous input signal must be used to gate the Timer1 clock input.

The following asynchronous sources may be used:

- Asynchronous event on the T1G pin to Timer1 gate

### 20.2.2 EXTERNAL CLOCK SOURCE

When the external clock source is selected, the Timer1 module may work as a timer or a counter.

When enabled to count, Timer1 is incremented on the rising edge of the external clock input T1CKI, which can be synchronized to the microcontroller system clock or can run asynchronously.

**Note:** In Counter mode, a falling edge must be registered by the counter prior to the first incrementing rising edge after any one or more of the following conditions:

- Timer1 enabled after POR
- Write to TMR1H or TMR1L
- Timer1 is disabled
- Timer1 is disabled (TMR1ON = 0) when T1CKI is high then Timer1 is enabled (TMR1ON=1) when T1CKI is low.

**TABLE 20-2: CLOCK SOURCE SELECTIONS**

| TMR1CS<1:0> | Clock Source |
|---|---|
| 11 | LFINTOSC |
| 10 | External Clocking on T1CKI Pin |
| 01 | System Clock (FOSC) |
| 00 | Instruction Clock (FOSC/4) |


© 2013-2015 Microchip Technology Inc.

## 20.3 Timer1 Prescaler

Timer1 has four prescaler options allowing 1, 2, 4 or 8 divisions of the clock input. The T1CKPS bits of the T1CON register 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.

## 20.4 Timer1 Operation in Asynchronous Counter Mode

If the control bit $\overline{\text{T1SYNC}}$ of the T1CON register is set, the external clock input is not synchronized. The timer increments asynchronously to the internal phase clocks. If the external clock source is selected then 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 20.4.1 "Reading and Writing Timer1 in Asynchronous Counter Mode"**).

| | |
|---|---|
| **Note:** | When switching from synchronous to asynchronous operation, it is possible to skip an increment. When switching from asynchronous to synchronous operation, it is possible to produce an additional increment. |

### 20.4.1 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 TMR1H:TMR1L register pair.

## 20.5 Timer1 Gate

Timer1 can be configured to count freely or the count can be enabled and disabled using Timer1 gate circuitry. This is also referred to as Timer1 Gate Enable.

Timer1 gate can also be driven by multiple selectable sources.

### 20.5.1 TIMER1 GATE ENABLE

The Timer1 Gate Enable mode is enabled by setting the TMR1GE bit of the T1GCON register. The polarity of the Timer1 Gate Enable mode is configured using the T1GPOL bit of the T1GCON register.

When Timer1 Gate Enable mode is enabled, Timer1 will increment on the rising edge of the Timer1 clock source. When Timer1 Gate Enable mode is disabled, no incrementing will occur and Timer1 will hold the current count. See Figure 20-3 for timing details.

**TABLE 20-3: TIMER1 GATE ENABLE SELECTIONS**

| T1CLK | T1GPOL | T1G | Timer1 Operation |
|---|---|---|---|
| ↑ | 0 | 0 | Counts |
| ↑ | 0 | 1 | Holds Count |
| ↑ | 1 | 0 | Holds Count |
| ↑ | 1 | 1 | Counts |

### 20.5.2 TIMER1 GATE SOURCE SELECTION

Timer1 gate source selections are shown in Table 20-4. Source selection is controlled by the T1GSS bits of the T1GCON register. The polarity for each available source is also selectable. Polarity selection is controlled by the T1GPOL bit of the T1GCON register.

**TABLE 20-4: TIMER1 GATE SOURCES**

| T1GSS | Timer1 Gate Source |
|---|---|
| 00 | Timer1 Gate Pin |
| 01 | Overflow of Timer0<br>(TMR0 increments from FFh to 00h) |
| 10 | Reserved |
| 11 | Reserved |

#### 20.5.2.1 T1G Pin Gate Operation

The T1G pin is one source for Timer1 gate control. It can be used to supply an external source to the Timer1 gate circuitry.

#### 20.5.2.2 Timer0 Overflow Gate Operation

When Timer0 increments from FFh to 00h, a low-to-high pulse will automatically be generated and internally supplied to the Timer1 gate circuitry.

### 20.5.3 TIMER1 GATE TOGGLE MODE

When Timer1 Gate Toggle mode is enabled, it is possible to measure the full-cycle length of a Timer1 gate signal, as opposed to the duration of a single level pulse.

The Timer1 gate source is routed through a flip-flop that changes state on every incrementing edge of the signal. See Figure 20-4 for timing details.

Timer1 Gate Toggle mode is enabled by setting the T1GTM bit of the T1GCON register. When the T1GTM bit is cleared, the flip-flop is cleared and held clear. This is necessary in order to control which edge is measured.

| **Note:** | Enabling Toggle mode at the same time as changing the gate polarity may result in indeterminate operation. |
|---|---|

### 20.5.4 TIMER1 GATE SINGLE-PULSE MODE

When Timer1 Gate Single-Pulse mode is enabled, it is possible to capture a single-pulse gate event. Timer1 Gate Single-Pulse mode is first enabled by setting the T1GSPM bit in the T1GCON register. Next, the T1GGO/$\overline{\text{DONE}}$ bit in the T1GCON register must be set. The Timer1 will be fully enabled on the next incrementing edge. On the next trailing edge of the pulse, the T1GGO/$\overline{\text{DONE}}$ bit will automatically be cleared. No other gate events will be allowed to increment Timer1 until the T1GGO/$\overline{\text{DONE}}$ bit is once again set in software. See Figure 20-5 for timing details.

If the Single-Pulse Gate mode is disabled by clearing the T1GSPM bit in the T1GCON register, the T1GGO/$\overline{\text{DONE}}$ bit should also be cleared.

Enabling the Toggle mode and the Single-Pulse mode simultaneously will permit both sections to work together. This allows the cycle times on the Timer1 gate source to be measured. See Figure 20-6 for timing details.

### 20.5.5 TIMER1 GATE VALUE STATUS

When Timer1 Gate Value Status is utilized, it is possible to read the most current level of the gate control value. The value is stored in the T1GVAL bit in the T1GCON register. The T1GVAL bit is valid even when the Timer1 gate is not enabled (TMR1GE bit is cleared).

### 20.5.6 TIMER1 GATE EVENT INTERRUPT

When Timer1 Gate Event Interrupt is enabled, it is possible to generate an interrupt upon the completion of a gate event. When the falling edge of T1GVAL occurs, the TMR1GIF flag bit in the PIR1 register will be set. If the TMR1GIE bit in the PIE1 register is set, then an interrupt will be recognized.

The TMR1GIF flag bit operates even when the Timer1 gate is not enabled (TMR1GE bit is cleared).


© 2013-2015 Microchip Technology Inc.

## 20.6 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 of the PIR1 register is set. To enable the interrupt on rollover, you must set these bits:

- TMR1ON bit of the T1CON register
- TMR1IE bit of the PIE1 register
- PEIE bit of the INTCON register
- GIE bit of the INTCON register

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

| **Note:** | The TMR1H:TMR1L register pair and the TMR1IF bit should be cleared before enabling interrupts. |
|---|---|

## 20.7 Timer1 Operation During Sleep

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

- TMR1ON bit of the T1CON register must be set
- TMR1IE bit of the PIE1 register must be set
- PEIE bit of the INTCON register must be set
- $\overline{\text{T1SYNC}}$ bit of the T1CON register must be set
- TMR1CS bits of the T1CON register must be configured

The device will wake-up on an overflow and execute the next instructions. If the GIE bit of the INTCON register is set, the device will call the Interrupt Service Routine.

Secondary oscillator will continue to operate in Sleep regardless of the $\overline{\text{T1SYNC}}$ bit setting.

## 20.8 CCP Capture/Compare Time Base

The CCP modules use the TMR1H:TMR1L register pair as the time base when operating in Capture or Compare mode.

In Capture mode, the value in the TMR1H:TMR1L register pair is copied into the CCPR1H:CCPR1L register pair on a configured event.

In Compare mode, an event is triggered when the value CCPR1H:CCPR1L register pair matches the value in the TMR1H:TMR1L register pair. This event can be an Auto-conversion Trigger.

For more information, see **Section 22.0 "Capture/Compare/PWM Modules"**.

## 20.9 CCP Auto-Conversion Trigger

When any of the CCP's are configured to trigger an auto-conversion, the trigger will clear the TMR1H:TMR1L register pair. This auto-conversion does not cause a Timer1 interrupt. The CCP module may still be configured to generate a CCP interrupt.

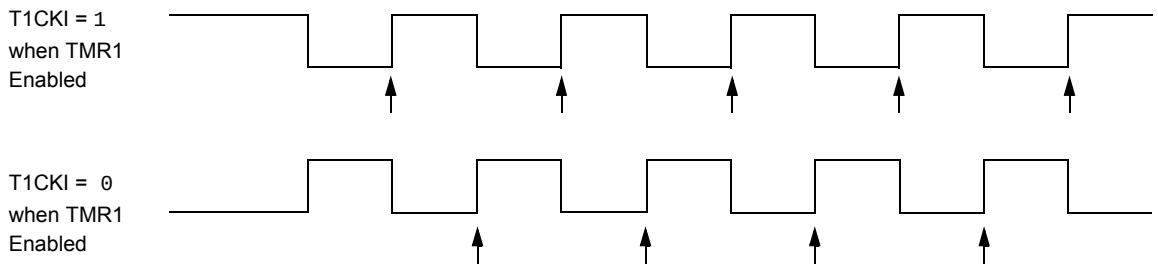
In this mode of operation, the CCPR1H:CCPR1L register pair becomes the period register for Timer1.

Timer1 should be synchronized and FOSC/4 should be selected as the clock source in order to utilize the Auto-conversion Trigger. Asynchronous operation of Timer1 can cause an Auto-conversion Trigger to be missed.

In the event that a write to TMR1H or TMR1L coincides with an Auto-conversion Trigger from the CCP, the write will take precedence.

For more information, see **Section 22.2.4 "Auto-Conversion Trigger"**.

**FIGURE 20-2: 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.

**FIGURE 20-3: TIMER1 GATE ENABLE MODE**

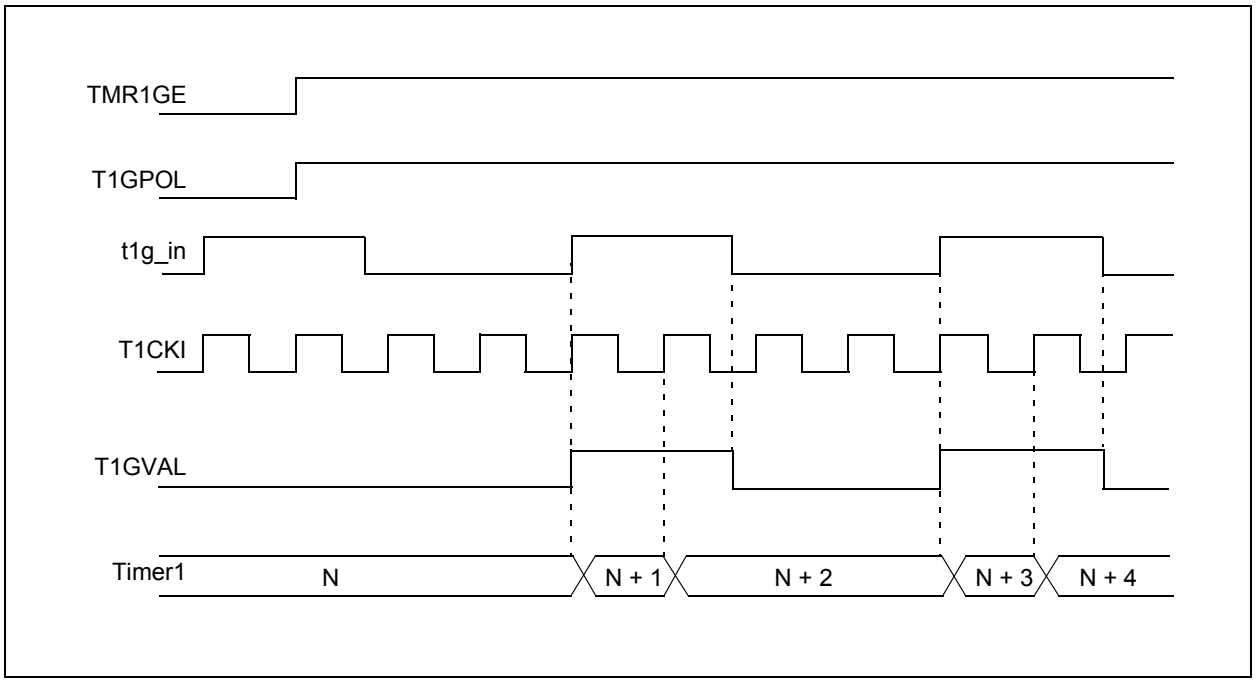
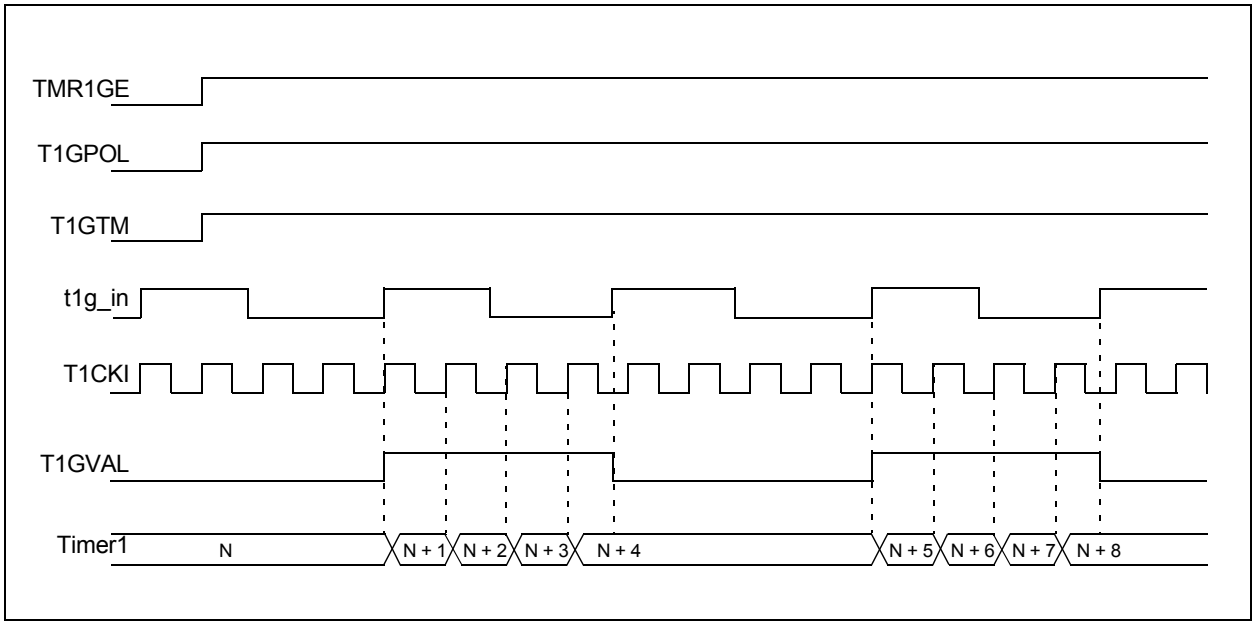



**FIGURE 20-4: TIMER1 GATE TOGGLE MODE**





© 2013-2015 Microchip Technology Inc.

**FIGURE 20-5: TIMER1 GATE SINGLE-PULSE MODE**

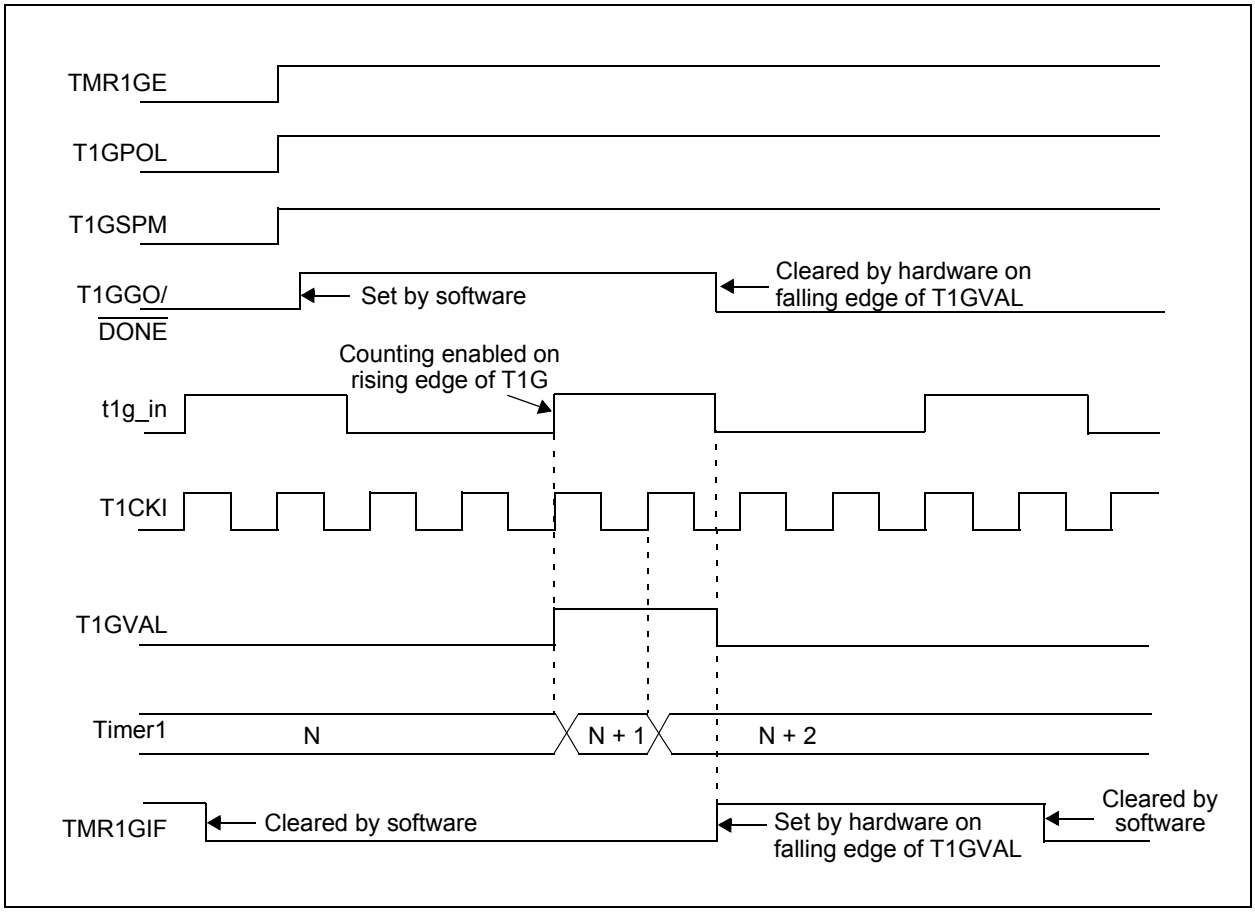
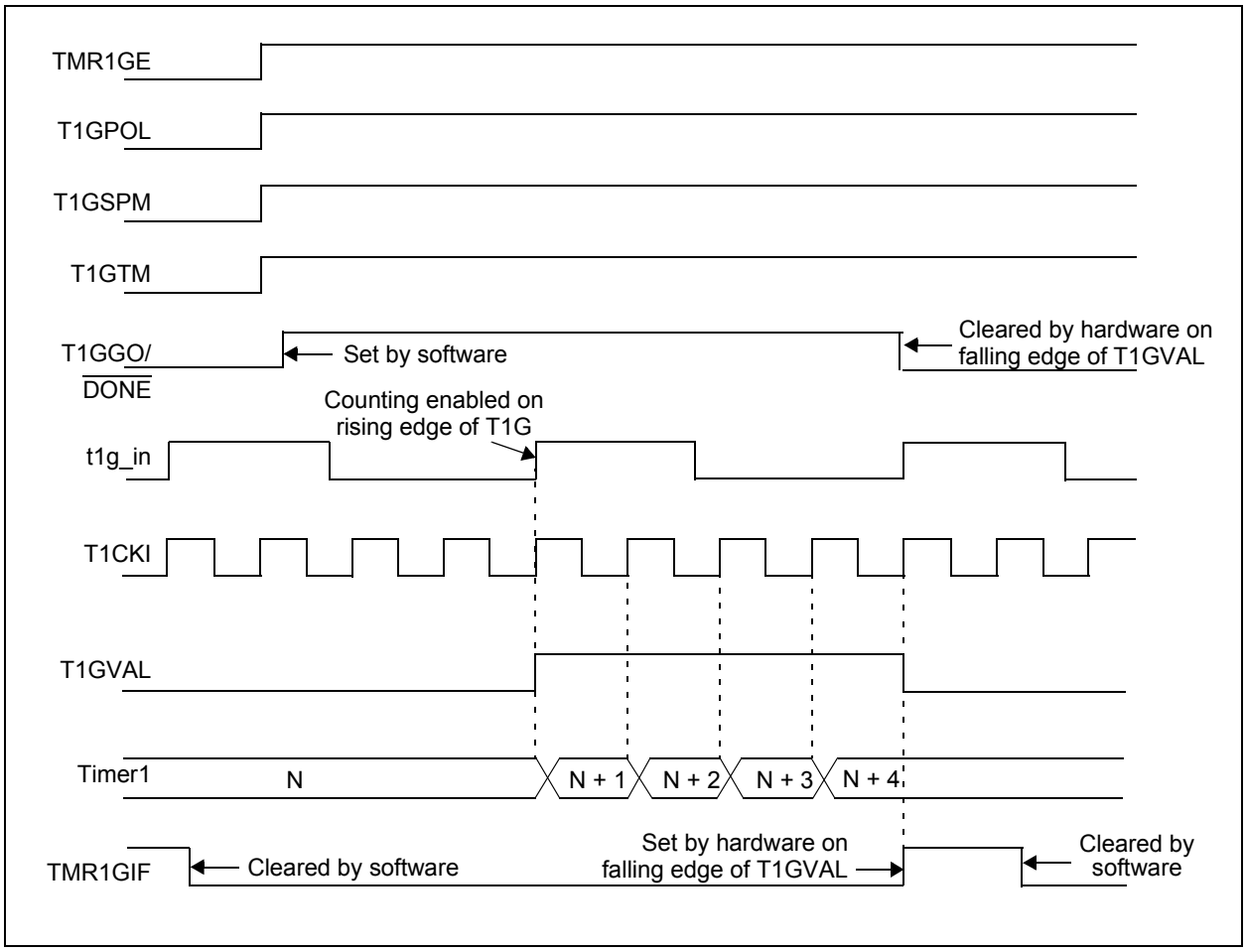


**FIGURE 20-6: TIMER1 GATE SINGLE-PULSE AND TOGGLE COMBINED MODE**





© 2013-2015 Microchip Technology Inc.

## 20.10 Register Definitions: Timer1 Control

**REGISTER 20-1: T1CON: TIMER1 CONTROL REGISTER**

| R/W-0/u | R/W-0/u | R/W-0/u | R/W-0/u | U-0 | R/W-0/u | U-0 | R/W-0/u |
|---|---|---|---|---|---|---|---|
| TMR1CS<1:0> | | T1CKPS<1:0> | | — | $\overline{\text{T1SYNC}}$ | — | TMR1ON |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| '1' = Bit is set | '0' = Bit is cleared | |

bit 7-6 **TMR1CS<1:0>:** Timer1 Clock Source Select bits
- 11 = LFINTOSC
- 10 = External clock from T1CKI pin (on rising edge)
- 01 = Timer1 clock source is system clock (FOSC)
- 00 = Timer1 clock source is instruction clock (FOSC/4)

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 **Unimplemented:** Read as '0'

bit 2 **$\overline{\text{T1SYNC}}$:** Timer1 Synchronization Control bit
- 1 = Do not synchronize asynchronous clock input
- 0 = Synchronize asynchronous clock input with system clock (FOSC)

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

bit 0 **TMR1ON:** Timer1 On bit
- 1 = Enables Timer1
- 0 = Stops Timer1 and clears Timer1 gate flip-flop

**REGISTER 20-2: T1GCON: TIMER1 GATE CONTROL REGISTER**

| R/W-0/u | R/W-0/u | R/W-0/u | R/W-0/u | R/W/HC-0/u | R-x/x | R/W-0/u | R/W-0/u |
|---|---|---|---|---|---|---|---|
| TMR1GE | T1GPOL | T1GTM | T1GSPM | T1GGO/$\overline{\text{DONE}}$ | T1GVAL | T1GSS<1:0> | |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| '1' = Bit is set | '0' = Bit is cleared | HC = Bit is cleared by hardware |

bit 7 **TMR1GE:** Timer1 Gate Enable bit

If TMR1ON = 0:
This bit is ignored
If TMR1ON = 1:
1 = Timer1 counting is controlled by the Timer1 gate function
0 = Timer1 counts regardless of Timer1 gate function

bit 6 **T1GPOL:** Timer1 Gate Polarity bit

1 = Timer1 gate is active-high (Timer1 counts when gate is high)
0 = Timer1 gate is active-low (Timer1 counts when gate is low)

bit 5 **T1GTM:** Timer1 Gate Toggle Mode bit

1 = Timer1 Gate Toggle mode is enabled
0 = Timer1 Gate Toggle mode is disabled and toggle flip-flop is cleared
Timer1 gate flip-flop toggles on every rising edge.

bit 4 **T1GSPM:** Timer1 Gate Single-Pulse Mode bit

1 = Timer1 Gate Single-Pulse mode is enabled and is controlling Timer1 gate
0 = Timer1 Gate Single-Pulse mode is disabled

bit 3 **T1GGO/$\overline{\text{DONE}}$:** Timer1 Gate Single-Pulse Acquisition Status bit

1 = Timer1 gate single-pulse acquisition is ready, waiting for an edge
0 = Timer1 gate single-pulse acquisition has completed or has not been started

bit 2 **T1GVAL:** Timer1 Gate Value Status bit

Indicates the current state of the Timer1 gate that could be provided to TMR1H:TMR1L
Unaffected by Timer1 Gate Enable (TMR1GE)

bit 1-0 **T1GSS<1:0>:** Timer1 Gate Source Select bits

11 = Reserved
10 = Reserved
01 = Timer0 overflow output
00 = Timer1 gate pin


© 2013-2015 Microchip Technology Inc.

**TABLE 20-5: SUMMARY OF REGISTERS ASSOCIATED WITH TIMER1**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|
| ANSELA | — | — | — | ANSA4 | — | ANSA2 | ANSA1 | ANSA0 | 114 |
| CCP1CON | — | — | DC1B<1:0> | | CCP1M<3:0> | | | | 193 |
| CCP2CON | — | — | DC2B<1:0> | | CCP2M<3:0> | | | | 193 |
| INTCON | GIE | PEIE | TMR0IE | INTE | IOCIE | TMR0IF | INTF | IOCIF | 77 |
| PIE1 | TMR1GIE | ADIE | — | — | SSP1IE | CCP1IE | TMR2IE | TMR1IE | 78 |
| PIR1 | TMR1GIF | ADIF | — | — | SSP1IF | CCP1IF | TMR2IF | TMR1IF | 81 |
| TMR1H | Holding Register for the Most Significant Byte of the 16-bit TMR1 Register | | | | | | | | 169* |
| TMR1L | Holding Register for the Least Significant Byte of the 16-bit TMR1 Register | | | | | | | | 169* |
| TRISA | — | — | TRISA5 | TRISA4 | —(1) | TRISA2 | TRISA1 | TRISA0 | 113 |
| T1CON | TMR1CS<1:0> | | T1CKPS<1:0> | | — | $\overline{\text{T1SYNC}}$ | — | TMR1ON | 177 |
| T1GCON | TMR1GE | T1GPOL | T1GTM | T1GSPM | T1GGO/$\overline{\text{DONE}}$ | T1GVAL | T1GSS<1:0> | | 178 |

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

* Page provides register information.

**Note 1:** Unimplemented, read as '1'.

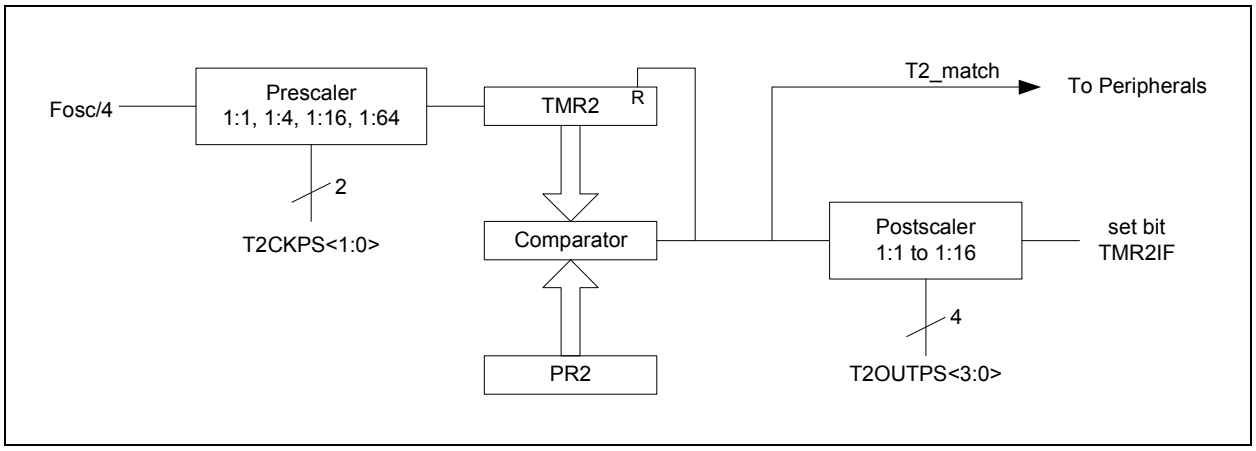
## 21.0 TIMER2 MODULE

The Timer2 module is an 8-bit timer that incorporates the following features:

- 8-bit Timer and Period registers (TMR2 and PR2, respectively)
- Readable and writable (both registers)
- Software programmable prescaler (1:1, 1:4, 1:16, and 1:64)
- Software programmable postscaler (1:1 to 1:16)
- Interrupt on TMR2 match with PR2, respectively
- Optional use as the shift clock for the MSSP module

See Figure 21-1 for a block diagram of Timer2.

**FIGURE 21-1: TIMER2 BLOCK DIAGRAM**





© 2013-2015 Microchip Technology Inc.

## 21.1 Timer2 Operation

The clock input to the Timer2 modules is the system instruction clock (FOSC/4).

TMR2 increments from 00h on each clock edge.

A 4-bit counter/prescaler on the clock input allows direct input, divide-by-4 and divide-by-16 prescale options. These options are selected by the prescaler control bits, T2CKPS<1:0> of the T2CON register. The value of TMR2 is compared to that of the Period register, PR2, on each clock cycle. When the two values match, the comparator generates a match signal as the timer output. This signal also resets the value of TMR2 to 00h on the next cycle and drives the output counter/postscaler (see **Section 21.2 "Timer2 Interrupt"**).

The TMR2 and PR2 registers are both directly readable and writable. The TMR2 register is cleared on any device Reset, whereas the PR2 register initializes to FFh. Both the prescaler and postscaler counters are cleared on the following events:

- a write to the TMR2 register
- a write to the T2CON register
- Power-on Reset (POR)
- Brown-out Reset (BOR)
- $\overline{\text{MCLR}}$ Reset
- Watchdog Timer (WDT) Reset
- Stack Overflow Reset
- Stack Underflow Reset
- RESET Instruction

| **Note:** | TMR2 is not cleared when T2CON is written. |
|---|---|

## 21.2 Timer2 Interrupt

Timer2 can also generate an optional device interrupt. The Timer2 output signal (TMR2-to-PR2 match) provides the input for the 4-bit counter/postscaler. This counter generates the TMR2 match interrupt flag which is latched in TMR2IF of the PIR1 register. The interrupt is enabled by setting the TMR2 Match Interrupt Enable bit, TMR2IE, of the PIE1 register.

A range of 16 postscale options (from 1:1 through 1:16 inclusive) can be selected with the postscaler control bits, T2OUTPS<3:0>, of the T2CON register.

## 21.3 Timer2 Output

The unscaled output of TMR2 is available primarily to the CCP modules, where it is used as a time base for operations in PWM mode.

Timer2 can be optionally used as the shift clock source for the MSSP module operating in SPI mode. Additional information is provided in **Section 23.0 "Master Synchronous Serial Port (MSSP) Module"**

## 21.4 Timer2 Operation During Sleep

The Timer2 timers cannot be operated while the processor is in Sleep mode. The contents of the TMR2 and PR2 registers will remain unchanged while the processor is in Sleep mode.

## 21.5 Register Definitions: Timer2 Control

**REGISTER 21-1: T2CON: TIMER2 CONTROL REGISTER**

| U-0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| — | T2OUTPS<3:0> | | | | TMR2ON | T2CKPS<1:0> | |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as ‘0’ |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| ‘1’ = Bit is set | ‘0’ = Bit is cleared | |

bit 7 **Unimplemented:** Read as ‘0’

bit 6-3 **T2OUTPS<3:0>:** Timer2 Output Postscaler Select bits
1111 = 1:16 Postscaler
1110 = 1:15 Postscaler
1101 = 1:14 Postscaler
1100 = 1:13 Postscaler
1011 = 1:12 Postscaler
1010 = 1:11 Postscaler
1001 = 1:10 Postscaler
1000 = 1:9 Postscaler
0111 = 1:8 Postscaler
0110 = 1:7 Postscaler
0101 = 1:6 Postscaler
0100 = 1:5 Postscaler
0011 = 1:4 Postscaler
0010 = 1:3 Postscaler
0001 = 1:2 Postscaler
0000 = 1:1 Postscaler

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
11 = Prescaler is 64
10 = Prescaler is 16
01 = Prescaler is 4
00 = Prescaler is 1


© 2013-2015 Microchip Technology Inc.

**TABLE 21-1: SUMMARY OF REGISTERS ASSOCIATED WITH TIMER2**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|
| CCP2CON | — | — | DC2B<1:0> | | CCP2M<3:0> | | | | 193 |
| INTCON | GIE | PEIE | TMR0IE | INTE | IOCIE | TMR0IF | INTF | IOCIF | 77 |
| PIE1 | TMR1GIE | ADIE | — | — | SSP1IE | CCP1IE | TMR2IE | TMR1IE | 78 |
| PIR1 | TMR1GIF | ADIF | — | — | SSP1IF | CCP1IF | TMR2IF | TMR1IF | 81 |
| PR2 | Timer2 Module Period Register | | | | | | | | 180* |
| T2CON | — | T2OUTPS<3:0> | | | | TMR2ON | T2CKPS<1:0> | | 182 |
| TMR2 | Holding Register for the 8-bit TMR2 Register | | | | | | | | 180* |

**Legend:** — = unimplemented location, read as '0'. Shaded cells are not used for Timer2 module.

* Page provides register information.

## 22.0 CAPTURE/COMPARE/PWM MODULES

The Capture/Compare/PWM module is a peripheral which allows the user to time and control different events, and to generate Pulse-Width Modulation (PWM) signals. In Capture mode, the peripheral allows the timing of the duration of an event. The Compare mode allows the user to trigger an external event when a predetermined amount of time has expired. The PWM mode can generate Pulse-Width Modulated signals of varying frequency and duty cycle.

This family of devices contains two standard Capture/Compare/PWM modules (CCP1 and CCP2).

The Capture and Compare functions are identical for all CCP modules.

**Note 1:** In devices with more than one CCP module, it is very important to pay close attention to the register names used. A number placed after the module acronym is used to distinguish between separate modules. For example, the CCP1CON and CCP2CON control the same operational aspects of two completely different CCP modules.

**2:** Throughout this section, generic references to a CCP module in any of its operating modes may be interpreted as being equally applicable to CCPx module. Register names, module signals, I/O pins, and bit names may use the generic designator 'x' to indicate the use of a numeral to distinguish a particular module, when required.


© 2013-2015 Microchip Technology Inc.

## 22.1 Capture Mode

The Capture mode function described in this section is available and identical for all CCP modules.

Capture mode makes use of the 16-bit Timer1 resource. When an event occurs on the CCPx pin, the 16-bit CCPRxH:CCPRxL register pair captures and stores the 16-bit value of the TMR1H:TMR1L register pair, respectively. An event is defined as one of the following and is configured by the CCPxM<3:0> bits of the CCPxCON register:

- Every falling edge
- Every rising edge
- Every 4th rising edge
- Every 16th rising edge

When a capture is made, the Interrupt Request Flag bit CCPxIF of the PIRx register is set. The interrupt flag must be cleared in software. If another capture occurs before the value in the CCPRxH, CCPRxL register pair is read, the old captured value is overwritten by the new captured value.

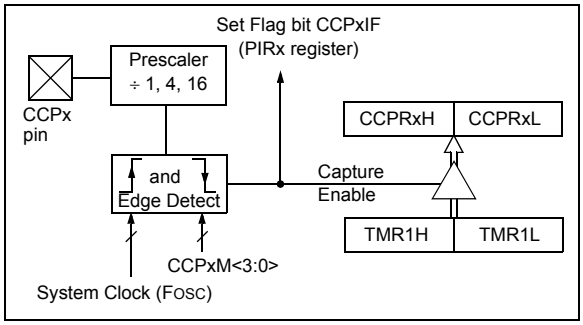
Figure 22-1 shows a simplified diagram of the capture operation.

### 22.1.1 CCP PIN CONFIGURATION

In Capture mode, the CCPx pin should be configured as an input by setting the associated TRIS control bit.

| **Note:** | If the CCPx pin is configured as an output, a write to the port can cause a capture condition. |
|---|---|

**FIGURE 22-1: CAPTURE MODE OPERATION BLOCK DIAGRAM**




### 22.1.2 TIMER1 MODE RESOURCE

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

See **Section 20.0 "Timer1 Module with Gate Control"** for more information on configuring Timer1.

### 22.1.3 SOFTWARE INTERRUPT MODE

When the Capture mode is changed, a false capture interrupt may be generated. The user should keep the CCPxIE interrupt enable bit of the PIEx register clear to avoid false interrupts. Additionally, the user should clear the CCPxIF interrupt flag bit of the PIRx register following any change in Operating mode.

| **Note:** | Clocking Timer1 from the system clock (FOSC) should not be used in Capture mode. In order for Capture mode to recognize the trigger event on the CCPx pin, Timer1 must be clocked from the instruction clock (FOSC/4) or from an external clock source. |
|---|---|

### 22.1.4 CCP PRESCALER

There are four prescaler settings specified by the CCPxM<3:0> bits of the CCPxCON register. Whenever the CCP module is turned off, or the CCP 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 does not clear the prescaler and may generate a false interrupt. To avoid this unexpected operation, turn the module off by clearing the CCPxCON register before changing the prescaler. Example 22-1 demonstrates the code to perform this function.

**EXAMPLE 22-1: CHANGING BETWEEN CAPTURE PRESCALERS**

```
BANKSEL CCPxCON     ;Set Bank bits to point
                    ;to CCPxCON
CLRF    CCPxCON     ;Turn CCP module off
MOVLW   NEW_CAPT_PS ;Load the W reg with
                    ;the new prescaler
                    ;move value and CCP ON
MOVWF   CCPxCON     ;Load CCPxCON with this
                    ;value
```

### 22.1.5 CAPTURE DURING SLEEP

Capture mode depends upon the Timer1 module for proper operation. There are two options for driving the Timer1 module in Capture mode. It can be driven by the instruction clock (FOSC/4), or by an external clock source.

When Timer1 is clocked by FOSC/4, Timer1 will not increment during Sleep. When the device wakes from Sleep, Timer1 will continue from its previous state.

Capture mode will operate during Sleep when Timer1 is clocked by an external clock source.


© 2013-2015 Microchip Technology Inc.

## 22.2 Compare Mode

The Compare mode function described in this section is available and identical for all CCP modules.

Compare mode makes use of the 16-bit Timer1 resource. The 16-bit value of the CCPRxH:CCPRxL register pair is constantly compared against the 16-bit value of the TMR1H:TMR1L register pair. When a match occurs, one of the following events can occur:

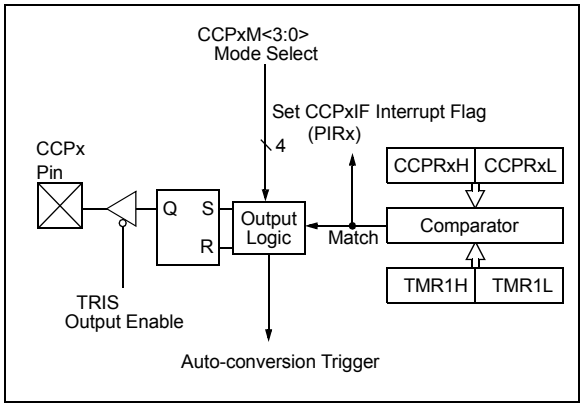
- Toggle the CCPx output
- Set the CCPx output
- Clear the CCPx output
- Generate an Auto-conversion Trigger
- Generate a Software Interrupt

The action on the pin is based on the value of the CCPxM<3:0> control bits of the CCPxCON register. At the same time, the interrupt flag CCPxIF bit is set.

All Compare modes can generate an interrupt.

Figure 22-2 shows a simplified diagram of the compare operation.

**FIGURE 22-2: COMPARE MODE OPERATION BLOCK DIAGRAM**




### 22.2.1 CCPX PIN CONFIGURATION

The user must configure the CCPx pin as an output by clearing the associated TRIS bit.

| | |
|---|---|
| **Note:** | Clearing the CCPxCON register will force the CCPx compare output latch to the default low level. This is not the PORT I/O data latch. |

### 22.2.2 TIMER1 MODE RESOURCE

In Compare mode, Timer1 must be running in either Timer mode or Synchronized Counter mode. The compare operation may not work in Asynchronous Counter mode.

See **Section 20.0 "Timer1 Module with Gate Control"** for more information on configuring Timer1.

| | |
|---|---|
| **Note:** | Clocking Timer1 from the system clock (FOSC) should not be used in Compare mode. In order for Compare mode to recognize the trigger event on the CCPx pin, TImer1 must be clocked from the instruction clock (FOSC/4) or from an external clock source. |

### 22.2.3 SOFTWARE INTERRUPT MODE

When Generate Software Interrupt mode is chosen (CCPxM<3:0> = `1010`), the CCPx module does not assert control of the CCPx pin (see the CCPxCON register).

### 22.2.4 AUTO-CONVERSION TRIGGER

When Auto-conversion Trigger mode is chosen (CCPxM<3:0> = `1011`), the CCPx module does the following:

- Resets Timer1
- Starts an ADC conversion if ADC is enabled

The CCPx module does not assert control of the CCPx pin in this mode.

The Auto-conversion Trigger output of the CCP occurs immediately upon a match between the TMR1H, TMR1L register pair and the CCPRxH, CCPRxL register pair. The TMR1H, TMR1L register pair is not reset until the next rising edge of the Timer1 clock. The Auto-conversion Trigger output starts an ADC conversion (if the ADC module is enabled). This allows the CCPRxH, CCPRxL register pair to effectively provide a 16-bit programmable period register for Timer1.

Refer to **Section 16.2.5 "Auto-Conversion Trigger"** for more information.

| | |
|---|---|
| **Note 1:** | The Auto-conversion Trigger from the CCP module does not set interrupt flag bit TMR1IF of the PIR1 register. |
| **2:** | Removing the match condition by changing the contents of the CCPRxH and CCPRxL register pair, between the clock edge that generates the Auto-conversion Trigger and the clock edge that generates the Timer1 Reset, will preclude the Reset from occurring. |

### 22.2.5 COMPARE DURING SLEEP

The Compare mode is dependent upon the system clock (FOSC) for proper operation. Since FOSC is shut down during Sleep mode, the Compare mode will not function properly during Sleep.


© 2013-2015 Microchip Technology Inc.

## 22.3 PWM Overview

Pulse-Width Modulation (PWM) is a scheme that provides power to a load by switching quickly between fully on and fully off states. The PWM signal resembles a square wave where the high portion of the signal is considered the on state and the low portion of the signal is considered the off state. The high portion, also known as the pulse width, can vary in time and is defined in steps. A larger number of steps applied, which lengthens the pulse width, also supplies more power to the load. Lowering the number of steps applied, which shortens the pulse width, supplies less power. The PWM period is defined as the duration of one complete cycle or the total amount of on and off time combined.

PWM resolution defines the maximum number of steps that can be present in a single PWM period. A higher resolution allows for more precise control of the pulse width time and in turn the power that is applied to the load.

The term duty cycle describes the proportion of the on time to the off time and is expressed in percentages, where 0% is fully off and 100% is fully on. A lower duty cycle corresponds to less power applied and a higher duty cycle corresponds to more power applied.

Figure 22-3 shows a typical waveform of the PWM signal.

### 22.3.1 STANDARD PWM OPERATION

The standard PWM function described in this section is available and identical for all CCP modules.

The standard PWM mode generates a Pulse-Width Modulation (PWM) signal on the CCPx pin with up to 10 bits of resolution. The period, duty cycle, and resolution are controlled by the following registers:

- PR2 registers
- T2CON registers
- CCPRxL registers
- CCPxCON registers

Figure 22-4 shows a simplified block diagram of PWM operation.

| **Note:** | The corresponding TRIS bit must be cleared to enable the PWM output on the CCPx pin. |
|---|---|

**FIGURE 22-3: CCP PWM OUTPUT SIGNAL**

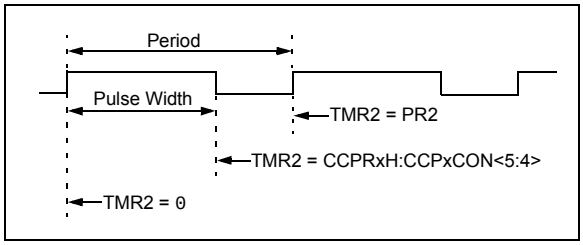
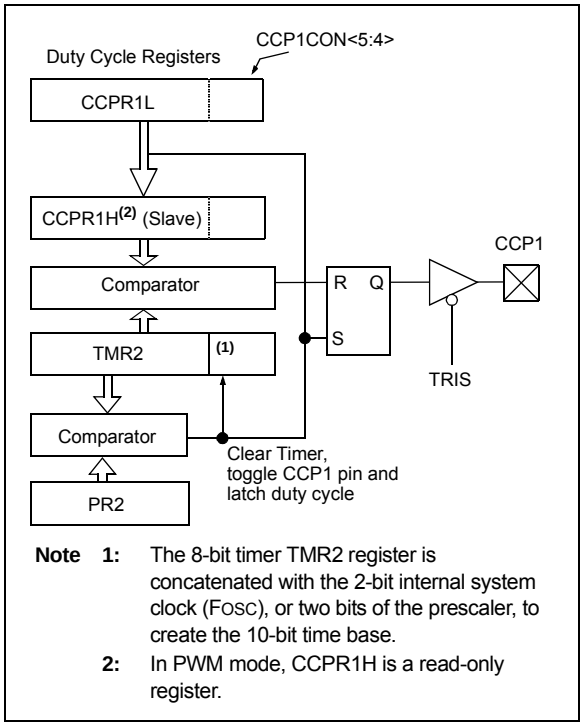



**FIGURE 22-4: SIMPLIFIED PWM BLOCK DIAGRAM**




**Note 1:** The 8-bit timer TMR2 register is concatenated with the 2-bit internal system clock (FOSC), or two bits of the prescaler, to create the 10-bit time base.

**2:** In PWM mode, CCPR1H is a read-only register.

### 22.3.2 SETUP FOR PWM OPERATION

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

1. Use the desired output pin RxyPPS control to select CCPx as the source and disable the CCPx pin output driver by setting the associated TRIS bit.
2. Load the PR2 register with the PWM period value.
3. Configure the CCP module for the PWM mode by loading the CCPxCON register with the appropriate values.
4. Load the CCPRxL register and the DCxBx bits of the CCPxCON register, with the PWM duty cycle value.
5. Configure and start Timer2:
   - Clear the TMR2IF interrupt flag bit of the PIRx register. See Note below.
   - Configure the T2CKPS bits of the T2CON register with the Timer prescale value.
   - Enable the Timer by setting the TMR2ON bit of the T2CON register.
6. Enable PWM output pin:
   - Wait until the Timer overflows and the TMR2IF bit of the PIR1 register is set. See Note below.
   - Enable the CCPx pin output driver by clearing the associated TRIS bit.

| **Note:** | In order to send a complete duty cycle and period on the first PWM output, the above steps must be included in the setup sequence. If it is not critical to start with a complete PWM signal on the first output, then step 6 may be ignored. |
|---|---|

### 22.3.3 TIMER2 TIMER RESOURCE

The PWM standard mode makes use of the 8-bit Timer2 timer resources to specify the PWM period.

### 22.3.4 PWM PERIOD

The PWM period is specified by the PR2 register of Timer2. The PWM period can be calculated using the formula of Equation 22-1.

**EQUATION 22-1: PWM PERIOD**

$$PWM\ Period = [(PR2) + 1] \bullet 4 \bullet T_{OSC} \bullet (TMR2\ Prescale\ Value)$$

**Note 1:** TOSC = 1/FOSC

When TMR2 is equal to PR2, the following three events occur on the next increment cycle:

- TMR2 is cleared
- The CCPx pin is set. (Exception: If the PWM duty cycle = 0%, the pin will not be set.)
- The PWM duty cycle is latched from CCPRxL into CCPRxH.

| **Note:** | The Timer postscaler (see **Section 21.1 "Timer2 Operation"**) is not used in the determination of the PWM frequency. |
|---|---|

### 22.3.5 PWM DUTY CYCLE

The PWM duty cycle is specified by writing a 10-bit value to multiple registers: CCPRxL register and DCxB<1:0> bits of the CCPxCON register. The CCPRxL contains the eight MSbs and the DCxB<1:0> bits of the CCPxCON register contain the two LSbs. CCPRxL and DCxB<1:0> bits of the CCPxCON register can be written to at any time. The duty cycle value is not latched into CCPRxH until after the period completes (i.e., a match between PR2 and TMR2 registers occurs). While using the PWM, the CCPRxH register is read-only.

Equation 22-2 is used to calculate the PWM pulse width.

Equation 22-3 is used to calculate the PWM duty cycle ratio.

**EQUATION 22-2: PULSE WIDTH**

$$Pulse\ Width = (CCPRxL{:}CCPxCON\langle 5{:}4\rangle) \bullet T_{OSC} \bullet (TMR2\ Prescale\ Value)$$

**EQUATION 22-3: DUTY CYCLE RATIO**

$$Duty\ Cycle\ Ratio = \frac{(CCPRxL{:}CCPxCON\langle 5{:}4\rangle)}{4(PR2 + 1)}$$

The CCPRxH register and a 2-bit internal latch are used to double buffer the PWM duty cycle. This double buffering is essential for glitchless PWM operation.

The 8-bit timer TMR2 register is concatenated with either the 2-bit internal system clock (FOSC), or two bits of the prescaler, to create the 10-bit time base. The system clock is used if the Timer2 prescaler is set to 1:1.

When the 10-bit time base matches the CCPRxH and 2-bit latch, then the CCPx pin is cleared (see Figure 22-4).


© 2013-2015 Microchip Technology Inc.

### 22.3.6 PWM RESOLUTION

The resolution determines the number of available duty cycles for a given period. For example, a 10-bit resolution will result in 1024 discrete duty cycles, whereas an 8-bit resolution will result in 256 discrete duty cycles.

The maximum PWM resolution is 10 bits when PR2 is 255. The resolution is a function of the PR2 register value as shown by Equation 22-4.

**EQUATION 22-4: PWM RESOLUTION**

$$Resolution = \frac{log[4(PR2 + 1)]}{log(2)} \ bits$$

**Note:** If the pulse width value is greater than the period the assigned PWM pin(s) will remain unchanged.

**TABLE 22-1: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS (FOSC = 20 MHz)**

| PWM Frequency | 1.22 kHz | 4.88 kHz | 19.53 kHz | 78.12 kHz | 156.3 kHz | 208.3 kHz |
|---|---|---|---|---|---|---|
| Timer Prescale | 16 | 4 | 1 | 1 | 1 | 1 |
| PR2 Value | 0xFF | 0xFF | 0xFF | 0x3F | 0x1F | 0x17 |
| Maximum Resolution (bits) | 10 | 10 | 10 | 8 | 7 | 6.6 |

**TABLE 22-2: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS (FOSC = 8 MHz)**

| PWM Frequency | 1.22 kHz | 4.90 kHz | 19.61 kHz | 76.92 kHz | 153.85 kHz | 200.0 kHz |
|---|---|---|---|---|---|---|
| Timer Prescale | 16 | 4 | 1 | 1 | 1 | 1 |
| PR2 Value | 0x65 | 0x65 | 0x65 | 0x19 | 0x0C | 0x09 |
| Maximum Resolution (bits) | 8 | 8 | 8 | 6 | 5 | 5 |

© 2013-2015 Microchip Technology Inc.

### 22.3.7 OPERATION IN SLEEP MODE

In Sleep mode, the TMR2 register will not increment and the state of the module will not change. If the CCPx pin is driving a value, it will continue to drive that value. When the device wakes up, TMR2 will continue from its previous state.

### 22.3.8 CHANGES IN SYSTEM CLOCK FREQUENCY

The PWM frequency is derived from the system clock frequency. Any changes in the system clock frequency will result in changes to the PWM frequency. See **Section 6.0 "Oscillator Module (with Fail-Safe Clock Monitor)"** for additional details.

### 22.3.9 EFFECTS OF RESET

Any Reset will force all ports to Input mode and the CCP registers to their Reset states.

**TABLE 22-3: SUMMARY OF REGISTERS ASSOCIATED WITH STANDARD PWM**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|
| CCP1CON | — | — | DC1B<1:0> | | CCP1M<3:0> | | | | 193 |
| CCPR1L | CCPR1L<7:0> | | | | | | | | 190* |
| INTCON | GIE | PEIE | TMR0IE | INTE | IOCIE | TMR0IF | INTF | IOCIF | 77 |
| PIE1 | TMR1GIE | ADIE | — | — | SSP1IE | CCP1IE | TMR2IE | TMR1IE | 78 |
| PIE2 | — | — | — | — | BCL1IE | — | — | CCP2IE | 79 |
| PIR1 | TMR1GIF | ADIF | — | — | SSP1IF | CCP1IF | TMR2IF | TMR1IF | 81 |
| PIR2 | — | — | — | — | BCL1IF | — | — | CCP2IF | 82 |
| PR2 | PR2 | | | | | | | | 180* |
| RxyPPS | — | — | — | RxyPPS<4:0> | | | | | 132 |
| T2CON | — | T2OUTPS<3:0> | | | | TMR2ON | T2CKPS<1:0> | | 182 |
| TMR2 | TMR2 | | | | | | | | 180 |

**Legend:** — = Unimplemented location, read as '0'. Shaded cells are not used by the PWM.
* Page provides register information.
**Note 1:** Unimplemented, read as '1'.


© 2013-2015 Microchip Technology Inc.

## 22.4 Register Definitions: CCP Control

### REGISTER 22-1: CCPxCON: CCPx CONTROL REGISTER

| U-0 | U-0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| — | — | DCxB<1:0> | | CCPxM<3:0> | | | |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as ‘0’ |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Reset |
| ‘1’ = Bit is set | ‘0’ = Bit is cleared | |

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5-4 **DCxB<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 CCPRxL.

bit 3-0 **CCPxM<3:0>:** CCPx Mode Select bits

11xx = PWM mode

1011 = Compare mode: Auto-conversion Trigger (sets CCPxIF bit), starts ADC conversion if TRIGSEL = CCPx (see Register 16-3)
1010 = Compare mode: generate software interrupt only
1001 = Compare mode: clear output on compare match (set CCPxIF)
1000 = Compare mode: set output on compare match (set CCPxIF)

0111 = Capture mode: every 16th rising edge
0110 = Capture mode: every 4th rising edge
0101 = Capture mode: every rising edge
0100 = Capture mode: every falling edge

0011 = Reserved
0010 = Compare mode: toggle output on match
0001 = Reserved
0000 = Capture/Compare/PWM off (resets CCPx module)

# 23.0 MASTER SYNCHRONOUS SERIAL PORT (MSSP) MODULE

## 23.1 MSSP Module Overview

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

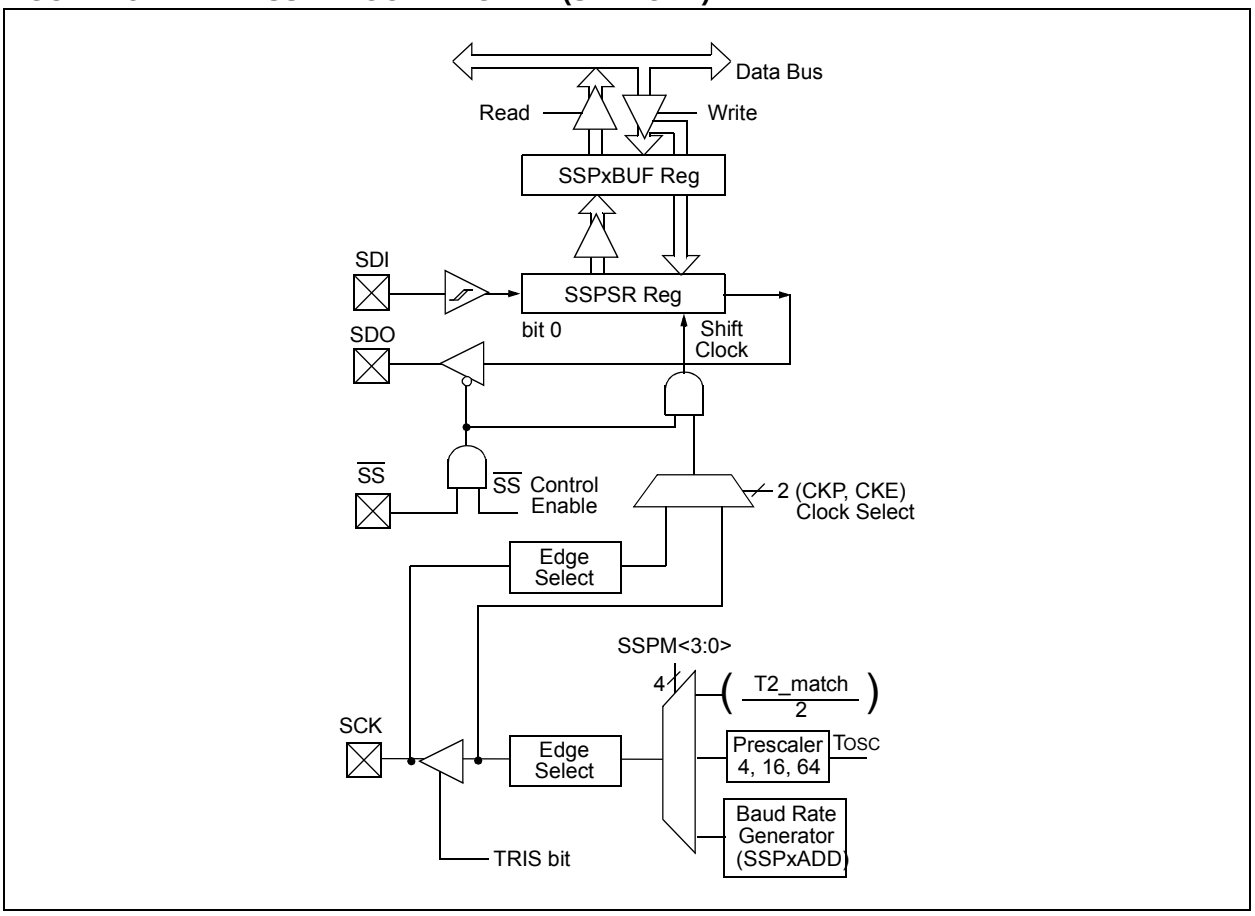
- Serial Peripheral Interface (SPI)
- Inter-Integrated Circuit ($I^2C$)

The SPI interface supports the following modes and features:

- Master mode
- Slave mode
- Clock Parity
- Slave Select Synchronization (Slave mode only)
- Daisy-chain connection of slave devices

Figure 23-1 is a block diagram of the SPI interface module.

**FIGURE 23-1: MSSP BLOCK DIAGRAM (SPI MODE)**





© 2013-2015 Microchip Technology Inc.

The I²C interface supports the following modes and features:

- Master mode
- Slave mode
- Byte NACKing (Slave mode)
- Limited multi-master support
- 7-bit and 10-bit addressing
- Start and Stop interrupts
- Interrupt masking
- Clock stretching
- Bus collision detection
- General call address matching
- Address masking
- Address Hold and Data Hold modes
- Selectable SDA hold times

Figure 23-2 is a block diagram of the I²C interface module in Master mode. Figure 23-3 is a diagram of the I²C interface module in Slave mode.

**FIGURE 23-2: MSSP BLOCK DIAGRAM (I²C MASTER MODE)**

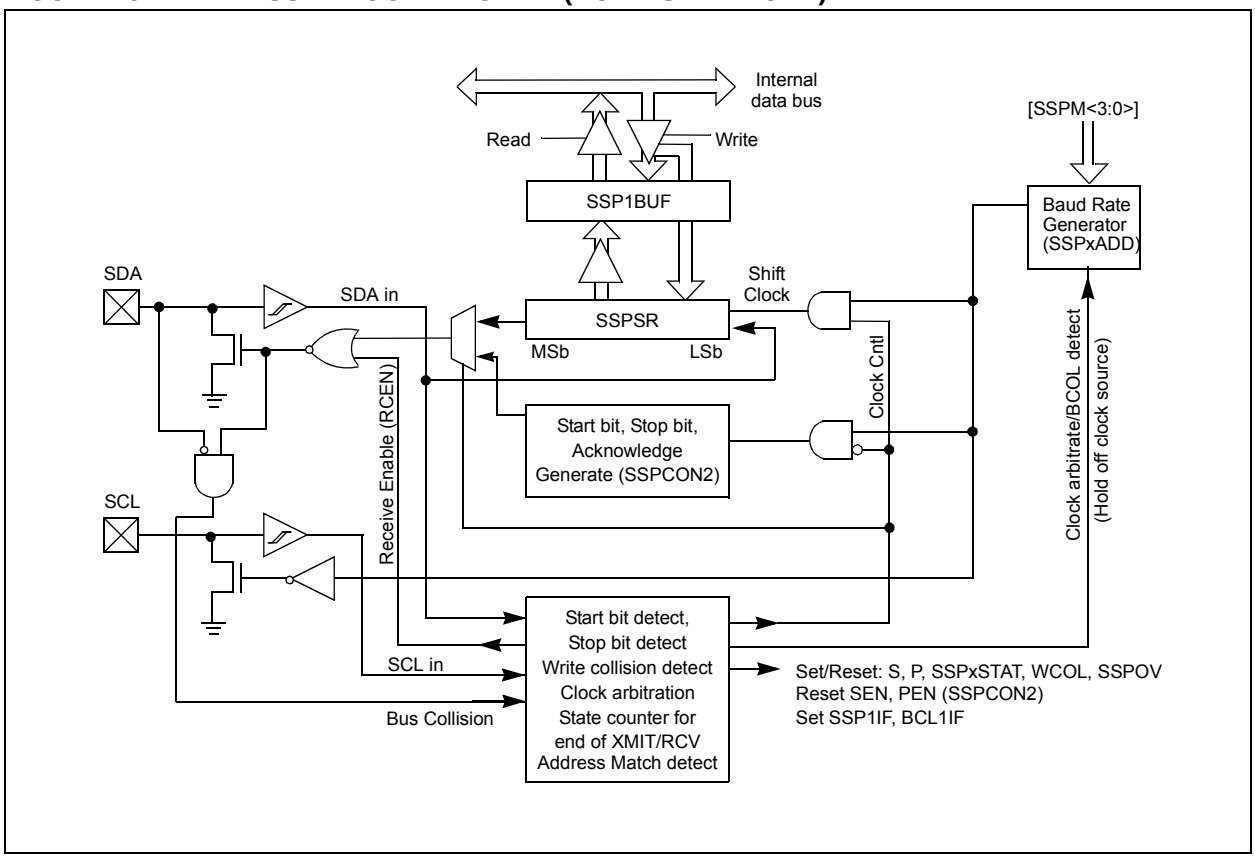
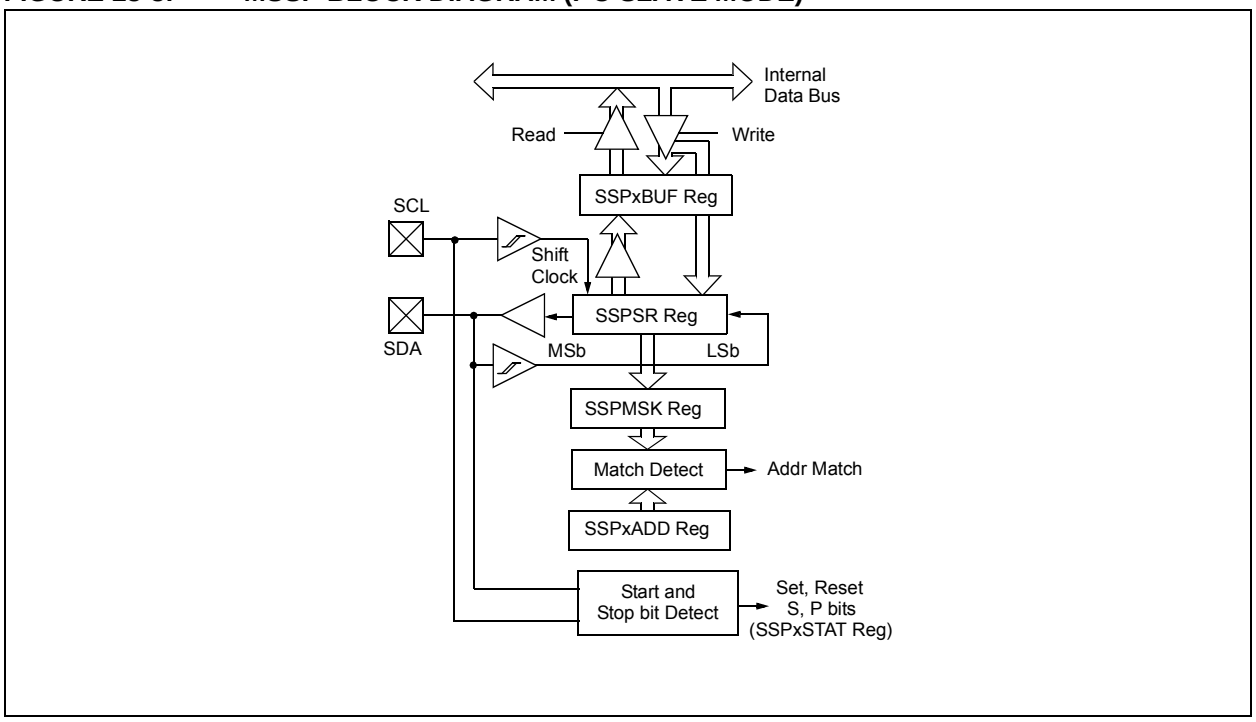


**FIGURE 23-3: MSSP BLOCK DIAGRAM ($I^2C$ SLAVE MODE)**





© 2013-2015 Microchip Technology Inc.

## 23.2 SPI Mode Overview

The Serial Peripheral Interface (SPI) bus is a synchronous serial data communication bus that operates in Full-Duplex mode. Devices communicate in a master/slave environment where the master device initiates the communication. A slave device is controlled through a Chip Select known as Slave Select.

The SPI bus specifies four signal connections:

- Serial Clock (SCK)
- Serial Data Out (SDO)
- Serial Data In (SDI)
- Slave Select ($\overline{SS}$)

Figure 23-1 shows the block diagram of the MSSP module when operating in SPI mode.

The SPI bus operates with a single master device and one or more slave devices. When multiple slave devices are used, an independent Slave Select connection is required from the master device to each slave device.

Figure 23-4 shows a typical connection between a master device and multiple slave devices.

The master selects only one slave at a time. Most slave devices have tri-state outputs so their output signal appears disconnected from the bus when they are not selected.

Transmissions involve two shift registers, eight bits in size, one in the master and one in the slave. With either the master or the slave device, data is always shifted out one bit at a time, with the Most Significant bit (MSb) shifted out first. At the same time, a new Least Significant bit (LSb) is shifted into the same register.

Figure 23-5 shows a typical connection between two processors configured as master and slave devices.

Data is shifted out of both shift registers on the programmed clock edge and latched on the opposite edge of the clock.

The master device transmits information out on its SDO output pin which is connected to, and received by, the slave's SDI input pin. The slave device transmits information out on its SDO output pin, which is connected to, and received by, the master's SDI input pin.

To begin communication, the master device first sends out the clock signal. Both the master and the slave devices should be configured for the same clock polarity.

The master device starts a transmission by sending out the MSb from its shift register. The slave device reads this bit from that same line and saves it into the LSb position of its shift register.

During each SPI clock cycle, a full-duplex data transmission occurs. This means that while the master device is sending out the MSb from its shift register (on its SDO pin) and the slave device is reading this bit and saving it as the LSb of its shift register, that the slave device is also sending out the MSb from its shift register (on its SDO pin) and the master device is reading this bit and saving it as the LSb of its shift register.

After eight bits have been shifted out, the master and slave have exchanged register values.

If there is more data to exchange, the shift registers are loaded with new data and the process repeats itself.

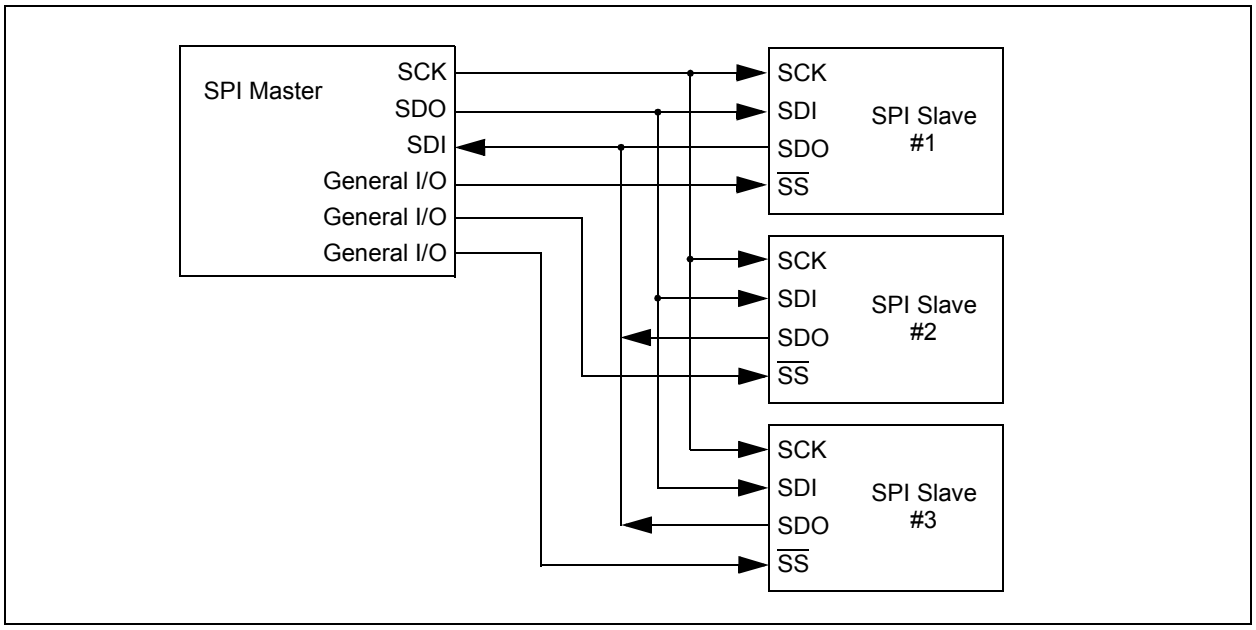
Whether the data is meaningful or not (dummy data), depends on the application software. This leads to three scenarios for data transmission:

- Master sends useful data and slave sends dummy data.
- Master sends useful data and slave sends useful data.
- Master sends dummy data and slave sends useful data.

Transmissions may involve any number of clock cycles. When there is no more data to be transmitted, the master stops sending the clock signal and it deselects the slave.

Every slave device connected to the bus that has not been selected through its slave select line must disregard the clock and transmission signals and must not transmit out any data of its own.

**FIGURE 23-4: SPI MASTER AND MULTIPLE SLAVE CONNECTION**




23.2.1 SPI MODE REGISTERS

The MSSP module has five registers for SPI mode operation. These are:

- MSSP STATUS register (SSPxSTAT)
- MSSP Control register 1 (SSPxCON1)
- MSSP Control register 3 (SSPxCON3)
- MSSP Data Buffer register (SSPxBUF)
- MSSP Address register (SSPxADD)
- MSSP Shift register (SSPSR)
  (Not directly accessible)

SSPxCON1 and SSPxSTAT are the control and STATUS registers in SPI mode operation. The SSPxCON1 register is readable and writable. The lower six bits of the SSPxSTAT are read-only. The upper two bits of the SSPxSTAT are read/write.

In one SPI master mode, SSPxADD can be loaded with a value used in the Baud Rate Generator. More information on the Baud Rate Generator is available in **Section 23.7 "Baud Rate Generator"**.

SSPSR is the shift register used for shifting data in and out. SSPxBUF provides indirect access to the SSPSR register. SSPxBUF is the buffer register to which data bytes are written, and from which data bytes are read.

In receive operations, SSPSR and SSPxBUF together create a buffered receiver. When SSPSR receives a complete byte, it is transferred to SSPxBUF and the SSPIF interrupt is set.

During transmission, the SSPxBUF is not buffered. A write to SSPxBUF will write to both SSPxBUF and SSPSR.


© 2013-2015 Microchip Technology Inc.

### 23.2.2 SPI MODE OPERATION

When initializing the SPI, several options need to be specified. This is done by programming the appropriate control bits (SSPxCON1<5:0> and SSPxSTAT<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)

To enable the serial port, SSP Enable bit, SSPEN of the SSPxCON1 register, must be set. To reset or reconfigure SPI mode, clear the SSPEN bit, re-initialize the SSPCONx 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 TRIS register) appropriately programmed as follows:

- SDI must have corresponding TRIS bit set
- SDO must have corresponding TRIS bit cleared
- SCK (Master mode) must have corresponding TRIS bit cleared
- SCK (Slave mode) must have corresponding TRIS bit set
- $\overline{SS}$ must have corresponding TRIS bit set

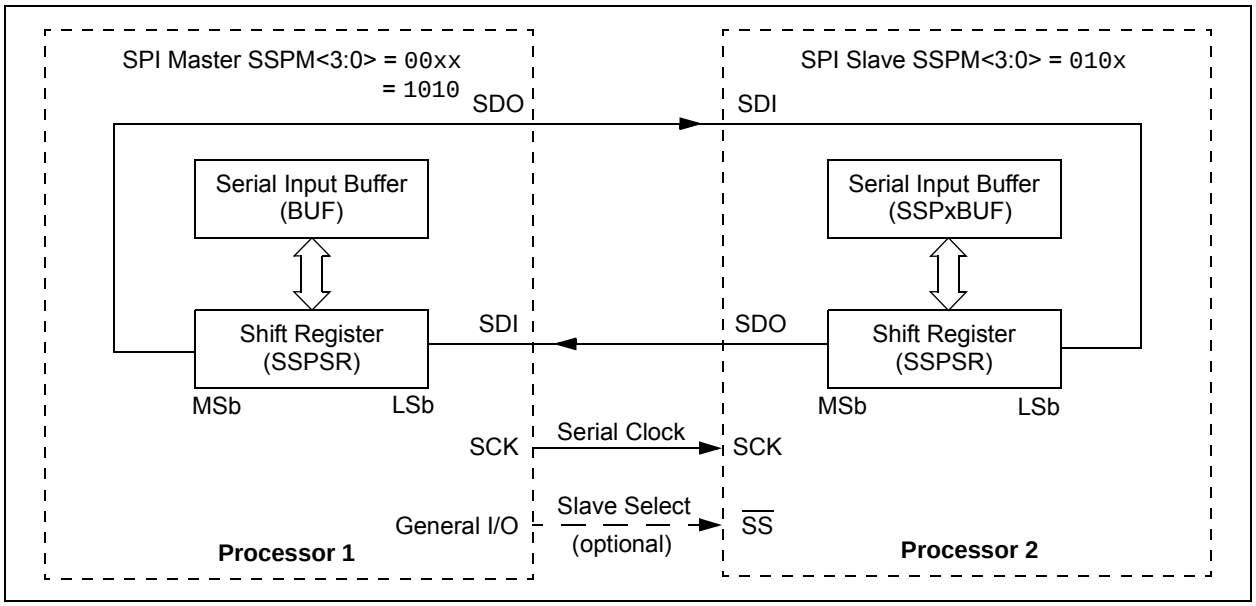
Any serial port function that is not desired may be overridden by programming the corresponding data direction (TRIS) register to the opposite value.

The MSSP consists of a transmit/receive shift register (SSPSR) and a buffer register (SSPxBUF). The SSPSR shifts the data in and out of the device, MSb first. The SSPxBUF 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 SSPxBUF register. Then, the Buffer Full Detect bit, BF of the SSPxSTAT register, and the interrupt flag bit, SSPIF, are set. This double-buffering of the received data (SSPxBUF) allows the next byte to start reception before reading the data that was just received. Any write to the SSPxBUF register during transmission/reception of data will be ignored and the write collision detect bit WCOL of the SSPxCON1 register, will be set. User software must clear the WCOL bit to allow the following write(s) to the SSPxBUF register to complete successfully.

When the application software is expecting to receive valid data, the SSPxBUF should be read before the next byte of data to transfer is written to the SSPxBUF. The Buffer Full bit, BF of the SSPxSTAT register, indicates when SSPxBUF has been loaded with the received data (transmission is complete). When the SSPxBUF is read, the BF bit is cleared. This data may be irrelevant if the SPI is only a transmitter. Generally, the MSSP interrupt is used to determine when the transmission/reception has completed. 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.

The SSPSR is not directly readable or writable and can only be accessed by addressing the SSPxBUF register. Additionally, the SSPxSTAT register indicates the various Status conditions.

**FIGURE 23-5: SPI MASTER/SLAVE CONNECTION**





© 2013-2015 Microchip Technology Inc.

### 23.2.3 SPI MASTER MODE

The master can initiate the data transfer at any time because it controls the SCK line. The master determines when the slave (Processor 2, Figure 23-5) is to broadcast data by the software protocol.

In Master mode, the data is transmitted/received as soon as the SSPxBUF 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 SSPxBUF register as if a normal received byte (interrupts and Status bits appropriately set).

The clock polarity is selected by appropriately programming the CKP bit of the SSPxCON1 register and the CKE bit of the SSPxSTAT register. This then, would give waveforms for SPI communication as shown in Figure 23-6, Figure 23-8, Figure 23-9 and Figure 23-10, where the MSB is transmitted first. In Master mode, the SPI clock rate (bit rate) is user programmable to be one of the following:

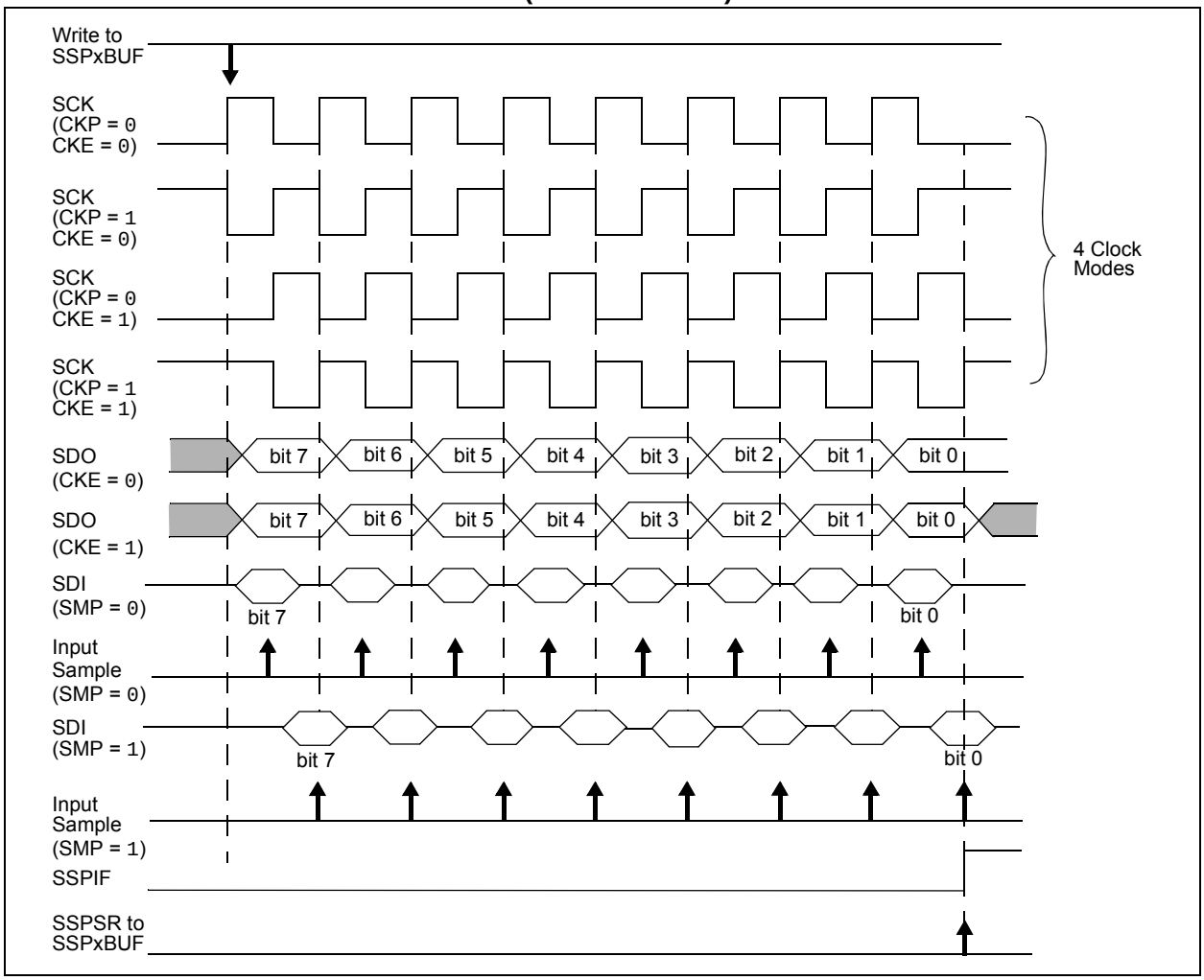
- FOSC/4 (or TCY)
- FOSC/16 (or 4 * TCY)
- FOSC/64 (or 16 * TCY)
- Timer2 output/2
- FOSC/(4 * (SSPxADD + 1))

Figure 23-6 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 SSPxBUF is loaded with the received data is shown.

| **Note:** | In Master mode, the clock signal output to the SCK pin is also the clock signal input to the peripheral. The pin selected for output with the RxyPPS register must also be selected as the peripheral input with the SSPCLKPPS register. |
|---|---|

**FIGURE 23-6: SPI MODE WAVEFORM (MASTER MODE)**



### 23.2.4 SPI SLAVE MODE

In Slave mode, the data is transmitted and received as external clock pulses appear on SCK. When the last bit is latched, the SSPIF interrupt flag bit is set.

Before enabling the module in SPI Slave mode, the clock line must match the proper Idle state. The clock line can be observed by reading the SCK pin. The Idle state is determined by the CKP bit of the SSPxCON1 register.

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. The shift register is clocked from the SCK pin input and when a byte is received, the device will generate an interrupt. If enabled, the device will wake-up from Sleep.

#### 23.2.4.1 Daisy-Chain Configuration

The SPI bus can sometimes be connected in a daisy-chain configuration. The first slave output is connected to the second slave input, the second slave output is connected to the third slave input, and so on. The final slave output is connected to the master input. Each slave sends out, during a second group of clock pulses, an exact copy of what was received during the first group of clock pulses. The whole chain acts as one large communication shift register. The daisy-chain feature only requires a single Slave Select line from the master device.

Figure 23-7 shows the block diagram of a typical daisy-chain connection when operating in SPI mode.

In a daisy-chain configuration, only the most recent byte on the bus is required by the slave. Setting the BOEN bit of the SSPxCON3 register will enable writes to the SSPxBUF register, even if the previous byte has not been read. This allows the software to ignore data that may not apply to it.

### 23.2.5 SLAVE SELECT SYNCHRONIZATION

The Slave Select can also be used to synchronize communication. The Slave Select line is held high until the master device is ready to communicate. When the Slave Select line is pulled low, the slave knows that a new transmission is starting.

If the slave fails to receive the communication properly, it will be reset at the end of the transmission, when the Slave Select line returns to a high state. The slave is then ready to receive a new transmission when the Slave Select line is pulled low again. If the Slave Select line is not used, there is a risk that the slave will eventually become out of sync with the master. If the slave misses a bit, it will always be one bit off in future transmissions. Use of the Slave Select line allows the slave and master to align themselves at the beginning of each transmission.

The $\overline{SS}$ pin allows a Synchronous Slave mode. The SPI must be in Slave mode with $\overline{SS}$ pin control enabled (SSPxCON1<3:0> = 0100).

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 (SSPxCON1<3:0> = 0100), the SPI module will reset if the $\overline{SS}$ pin is set to VDD. |
| **2:** | When the SPI is used in Slave mode with CKE set; the user must enable $\overline{SS}$ pin control. |
| **3:** | While operated in SPI Slave mode the SMP bit of the SSPxSTAT register must remain clear. |

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.


© 2013-2015 Microchip Technology Inc.

**FIGURE 23-7: SPI DAISY-CHAIN CONNECTION**

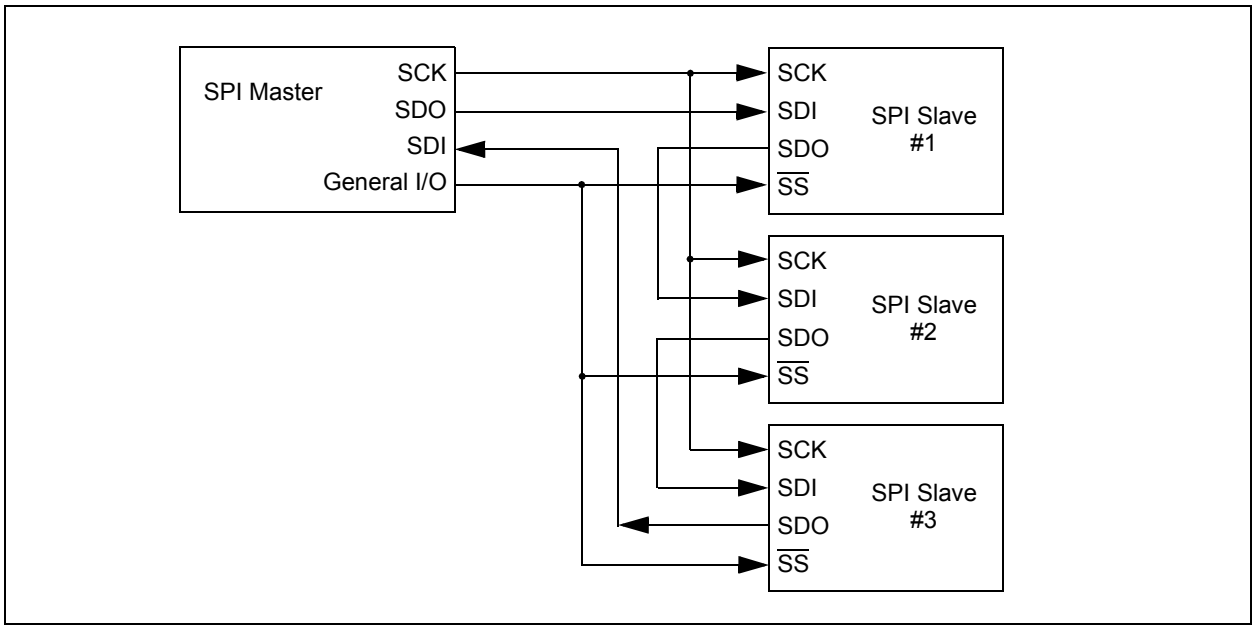



**FIGURE 23-8: SLAVE SELECT SYNCHRONOUS WAVEFORM**

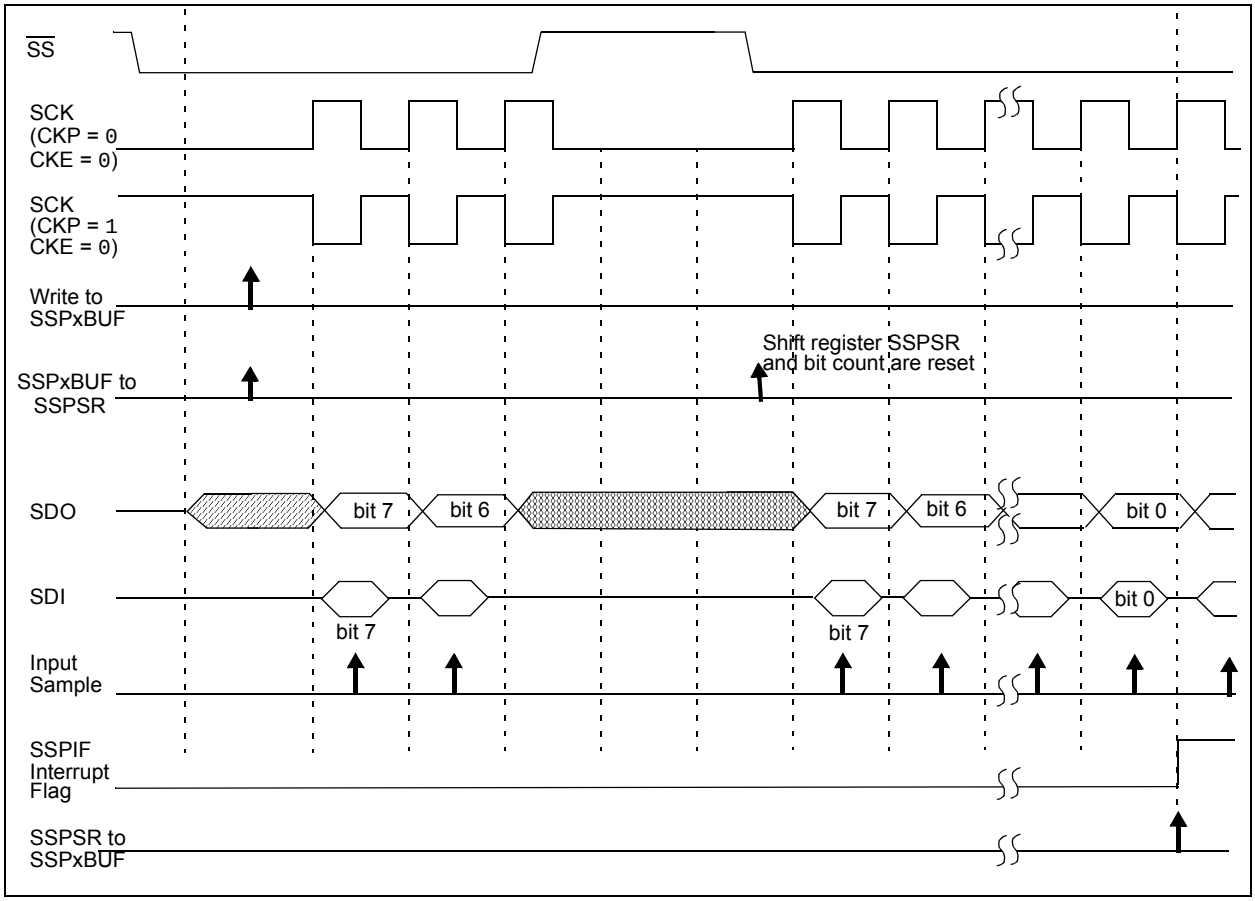


**FIGURE 23-9: SPI MODE WAVEFORM (SLAVE MODE WITH CKE = 0)**

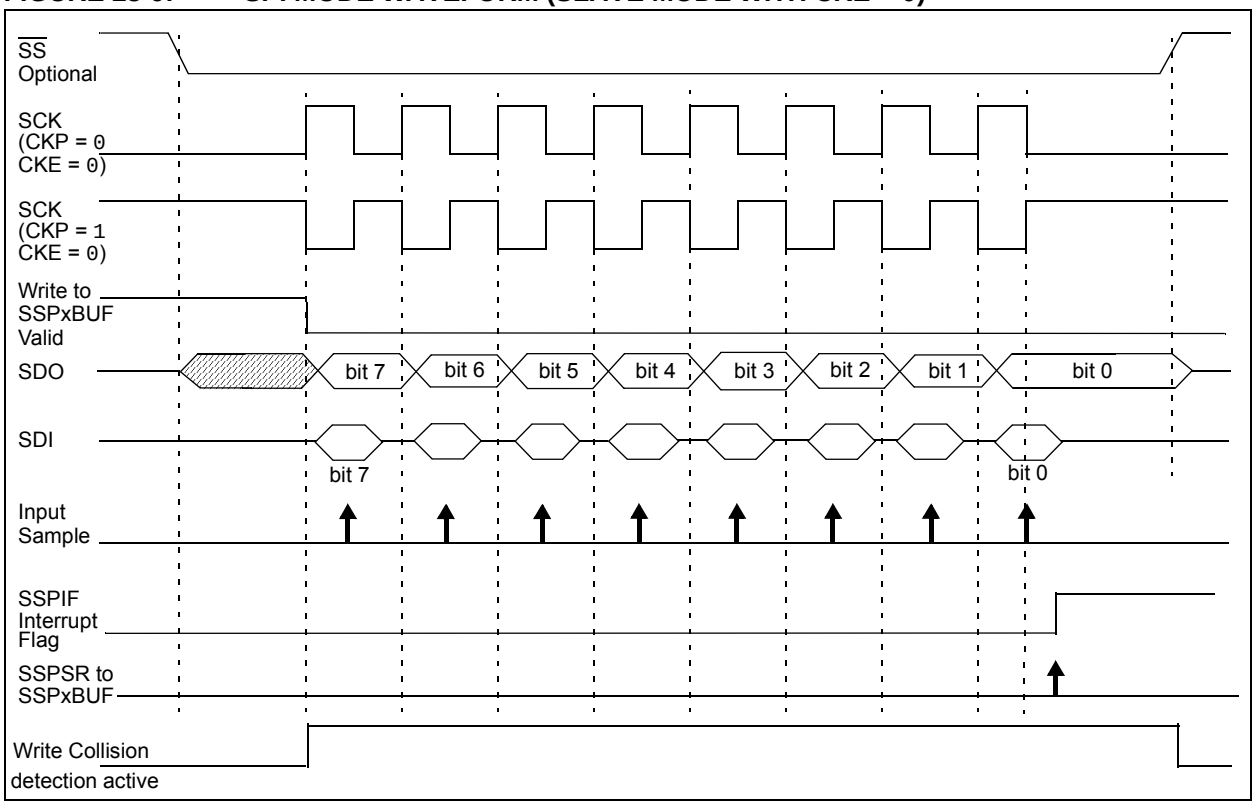
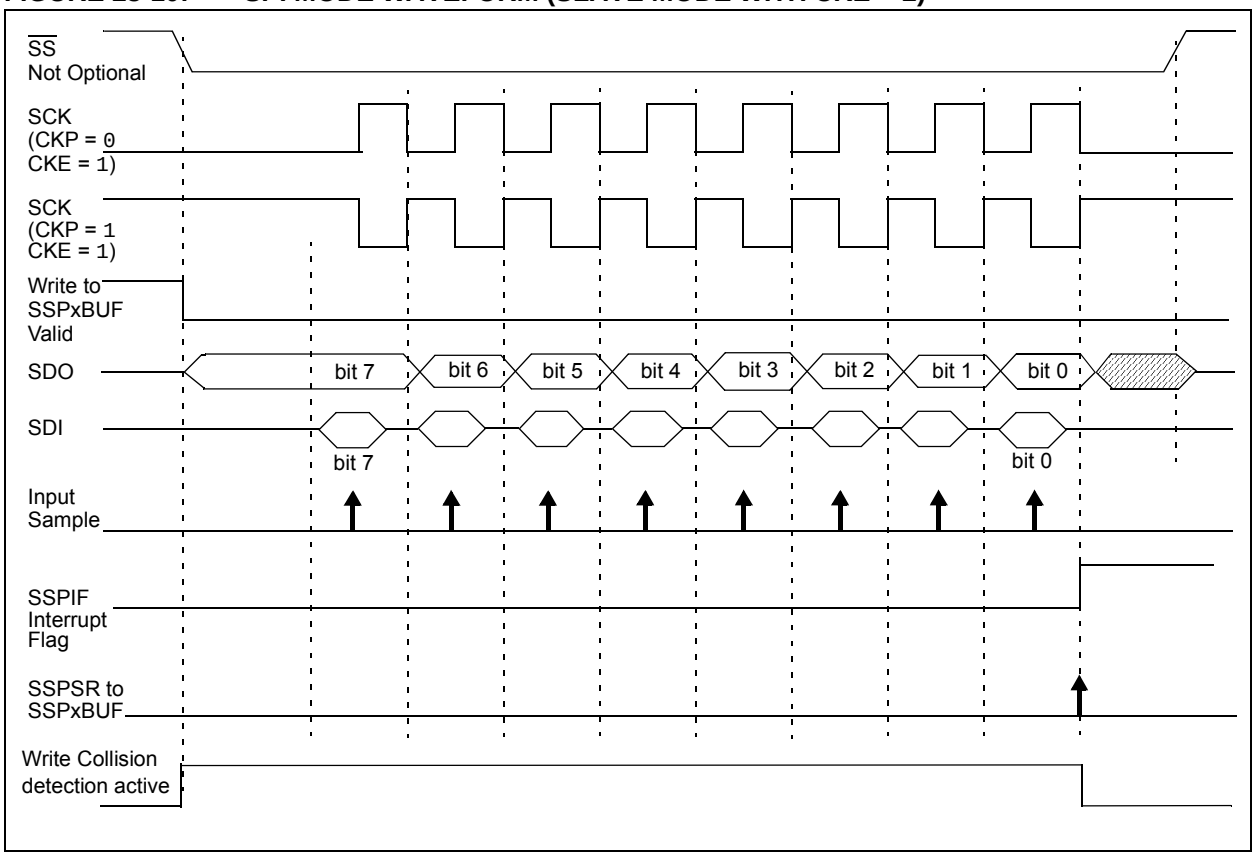



**FIGURE 23-10: SPI MODE WAVEFORM (SLAVE MODE WITH CKE = 1)**





© 2013-2015 Microchip Technology Inc.

### 23.2.6 SPI OPERATION IN SLEEP MODE

In SPI Master mode, module clocks may be operating at a different speed than when in Full-Power mode; in the case of the Sleep mode, all clocks are halted.

Special care must be taken by the user when the MSSP clock is much faster than the system clock.

In Slave mode, when MSSP interrupts are enabled, after the master completes sending data, an MSSP interrupt will wake the controller from Sleep.

If an exit from Sleep mode is not desired, MSSP interrupts should be disabled.

In SPI Master mode, when the Sleep mode is selected, all module clocks are halted and the transmission/reception will remain in that state until the device wakes. After the device returns to Run mode, the module will resume transmitting and receiving data.

In SPI 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 eight bits have been received, the MSSP interrupt flag bit will be set and if enabled, will wake the device.

**TABLE 23-1: SUMMARY OF REGISTERS ASSOCIATED WITH SPI OPERATION**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|
| ANSELA | — | — | — | ANSA4 | — | ANSA2 | ANSA1 | ANSA0 | 114 |
| ANSELC | ANSC7(2) | ANSC6(2) | ANSC5(3) | ANSC4(3) | ANSC3 | ANSC2 | ANSC1 | ANSC0 | 125 |
| INTCON | GIE | PEIE | TMR0IE | INTE | IOCIE | TMR0IF | INTF | IOCIF | 77 |
| PIE1 | TMR1GIE | ADIE | — | — | SSP1IE | CCP1IE | TMR2IE | TMR1IE | 78 |
| PIR1 | TMR1GIF | ADIF | — | — | SSP1IF | CCP1IF | TMR2IF | TMR1IF | 81 |
| RxyPPS | — | — | — | RxyPPS<4:0> | | | | | 132 |
| SSPCLKPPS | — | — | — | SSPCLKPPS<4:0> | | | | | 130, 131 |
| SSPDATPPS | — | — | — | SSPDATPPS<4:0> | | | | | 130, 131 |
| SSPSSPPS | — | — | — | SSPSSPPS<4:0> | | | | | 130, 131 |
| SSP1BUF | BUF<7:0> | | | | | | | | 198* |
| SSP1CON1 | WCOL | SSPOV | SSPEN | CKP | SSPM<3:0> | | | | 244 |
| SSP1CON3 | ACKTIM | PCIE | SCIE | BOEN | SDAHT | SBCDE | AHEN | DHEN | 246 |
| SSP1STAT | SMP | CKE | D/$\overline{A}$ | P | S | R/$\overline{W}$ | UA | BF | 242 |
| TRISA | — | — | TRISA5 | TRISA4 | —(1) | TRISA2 | TRISA1 | TRISA0 | 113 |
| TRISB(2) | TRISB7 | TRISB6 | TRISB5 | TRISB4 | — | — | — | — | 119 |
| TRISC | TRISC7(2) | TRISC6(2) | TRISC5 | TRISC4 | TRISC3 | TRISC2 | TRISC1 | TRISC0 | 124 |

**Legend:** — = Unimplemented location, read as '0'. Shaded cells are not used by the MSSP in SPI mode.

* Page provides register information.

**Note 1:** Unimplemented, read as '1'.

**2:** PIC16(L)F1707 only.

**3:** PIC16(L)F1703 only.

## 23.3 I²C MODE OVERVIEW

The Inter-Integrated Circuit (I²C) bus is a multi-master serial data communication bus. Devices communicate in a master/slave environment where the master devices initiate the communication. A slave device is controlled through addressing.

The I²C bus specifies two signal connections:

- Serial Clock (SCL)
- Serial Data (SDA)

Figure 23-11 shows the block diagram of the MSSP module when operating in I²C mode.

Both the SCL and SDA connections are bidirectional open-drain lines, each requiring pull-up resistors for the supply voltage. Pulling the line to ground is considered a logical zero and letting the line float is considered a logical one.

Figure 23-11 shows a typical connection between two processors configured as master and slave devices.

The I²C bus can operate with one or more master devices and one or more slave devices.

There are four potential modes of operation for a given device:

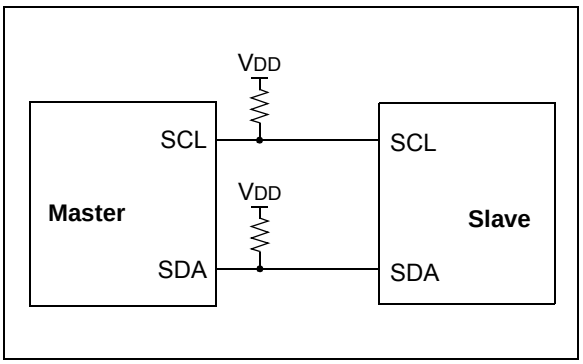
- Master Transmit mode
  (master is transmitting data to a slave)
- Master Receive mode
  (master is receiving data from a slave)
- Slave Transmit mode
  (slave is transmitting data to a master)
- Slave Receive mode
  (slave is receiving data from the master)

To begin communication, a master device starts out in Master Transmit mode. The master device sends out a Start bit followed by the address byte of the slave it intends to communicate with. This is followed by a single Read/Write bit, which determines whether the master intends to transmit to or receive data from the slave device.

If the requested slave exists on the bus, it will respond with an Acknowledge bit, otherwise known as an $\overline{\text{ACK}}$. The master then continues in either Transmit mode or Receive mode and the slave continues in the complement, either in Receive mode or Transmit mode, respectively.

A Start bit is indicated by a high-to-low transition of the SDA line while the SCL line is held high. Address and data bytes are sent out, Most Significant bit (MSb) first. The Read/Write bit is sent out as a logical one when the master intends to read data from the slave, and is sent out as a logical zero when it intends to write data to the slave.

**FIGURE 23-11: I²C MASTER/ SLAVE CONNECTION**




The Acknowledge bit ($\overline{\text{ACK}}$) is an active-low signal, which holds the SDA line low to indicate to the transmitter that the slave device has received the transmitted data and is ready to receive more.

The transition of a data bit is always performed while the SCL line is held low. Transitions that occur while the SCL line is held high are used to indicate Start and Stop bits.

If the master intends to write to the slave, then it repeatedly sends out a byte of data, with the slave responding after each byte with an $\overline{\text{ACK}}$ bit. In this example, the master device is in Master Transmit mode and the slave is in Slave Receive mode.

If the master intends to read from the slave, then it repeatedly receives a byte of data from the slave, and responds after each byte with an $\overline{\text{ACK}}$ bit. In this example, the master device is in Master Receive mode and the slave is Slave Transmit mode.

On the last byte of data communicated, the master device may end the transmission by sending a Stop bit. If the master device is in Receive mode, it sends the Stop bit in place of the last $\overline{\text{ACK}}$ bit. A Stop bit is indicated by a low-to-high transition of the SDA line while the SCL line is held high.

In some cases, the master may want to maintain control of the bus and re-initiate another transmission. If so, the master device may send another Start bit in place of the Stop bit or last $\overline{\text{ACK}}$ bit when it is in receive mode.

The I²C bus specifies three message protocols;

- Single message where a master writes data to a slave.
- Single message where a master reads data from a slave.
- Combined message where a master initiates a minimum of two writes, or two reads, or a combination of writes and reads, to one or more slaves.


© 2013-2015 Microchip Technology Inc.

When one device is transmitting a logical one, or letting the line float, and a second device is transmitting a logical zero, or holding the line low, the first device can detect that the line is not a logical one. This detection, when used on the SCL line, is called clock stretching. Clock stretching gives slave devices a mechanism to control the flow of data. When this detection is used on the SDA line, it is called arbitration. Arbitration ensures that there is only one master device communicating at any single time.

### 23.3.1 CLOCK STRETCHING

When a slave device has not completed processing data, it can delay the transfer of more data through the process of clock stretching. An addressed slave device may hold the SCL clock line low after receiving or sending a bit, indicating that it is not yet ready to continue. The master that is communicating with the slave will attempt to raise the SCL line in order to transfer the next bit, but will detect that the clock line has not yet been released. Because the SCL connection is open-drain, the slave has the ability to hold that line low until it is ready to continue communicating.

Clock stretching allows receivers that cannot keep up with a transmitter to control the flow of incoming data.

### 23.3.2 ARBITRATION

Each master device must monitor the bus for Start and Stop bits. If the device detects that the bus is busy, it cannot begin a new message until the bus returns to an Idle state.

However, two master devices may try to initiate a transmission on or about the same time. When this occurs, the process of arbitration begins. Each transmitter checks the level of the SDA data line and compares it to the level that it expects to find. The first transmitter to observe that the two levels do not match, loses arbitration, and must stop transmitting on the SDA line.

For example, if one transmitter holds the SDA line to a logical one (lets it float) and a second transmitter holds it to a logical zero (pulls it low), the result is that the SDA line will be low. The first transmitter then observes that the level of the line is different than expected and concludes that another transmitter is communicating.

The first transmitter to notice this difference is the one that loses arbitration and must stop driving the SDA line. If this transmitter is also a master device, it also must stop driving the SCL line. It then can monitor the lines for a Stop condition before trying to reissue its transmission. In the meantime, the other device that has not noticed any difference between the expected and actual levels on the SDA line continues with its original transmission. It can do so without any complications, because so far, the transmission appears exactly as expected with no other transmitter disturbing the message.

Slave Transmit mode can also be arbitrated, when a master addresses multiple slaves, but this is less common.

If two master devices are sending a message to two different slave devices at the address stage, the master sending the lower slave address always wins arbitration. When two master devices send messages to the same slave address, and addresses can sometimes refer to multiple slaves, the arbitration process must continue into the data stage.

Arbitration usually occurs very rarely, but it is a necessary process for proper multi-master support.

## 23.4 I²C MODE OPERATION

All MSSP I²C communication is byte oriented and shifted out MSb first. Six SFR registers and two interrupt flags interface the module with the PIC® microcontroller and user software. Two pins, SDA and SCL, are exercised by the module to communicate with other external I²C devices.

### 23.4.1 BYTE FORMAT

All communication in I²C is done in 9-bit segments. A byte is sent from a master to a slave or vice-versa, followed by an Acknowledge bit sent back. After the eighth falling edge of the SCL line, the device outputting data on the SDA changes that pin to an input and reads in an acknowledge value on the next clock pulse.

The clock signal, SCL, is provided by the master. Data is valid to change while the SCL signal is low, and sampled on the rising edge of the clock. Changes on the SDA line while the SCL line is high define special conditions on the bus, explained below.

### 23.4.2 DEFINITION OF I²C TERMINOLOGY

There is language and terminology in the description of I²C communication that have definitions specific to I²C. That word usage is defined below and may be used in the rest of this document without explanation. This table was adapted from the Philips I²C specification.

### 23.4.3 SDA AND SCL PINS

Selection of any I²C mode with the SSPEN bit set, forces the SCL and SDA pins to be open-drain. These pins should be set by the user to inputs by setting the appropriate TRIS bits.

**Note 1:** Data is tied to output zero when an I²C mode is enabled.

**2:** Any device pin can be selected for SDA and SCL functions with the PPS peripheral. These functions are bidirectional. The SDA input is selected with the SSPDATPPS registers. The SCL input is selected with the SSPCLKPPS registers. Outputs are selected with the RxyPPS registers. It is the user's responsibility to make the selections so that both the input and the output for each function is on the same pin.

### 23.4.4 SDA HOLD TIME

The hold time of the SDA pin is selected by the SDAHT bit of the SSPxCON3 register. Hold time is the time SDA is held valid after the falling edge of SCL. Setting the SDAHT bit selects a longer 300 ns minimum hold time and may help on buses with large capacitance.

**TABLE 23-2: I²C BUS TERMS**

| TERM | Description |
|---|---|
| Transmitter | The device which shifts data out onto the bus. |
| Receiver | The device which shifts data in from the bus. |
| Master | The device that initiates a transfer, generates clock signals and terminates a transfer. |
| Slave | The device addressed by the master. |
| Multi-master | A bus with more than one device that can initiate data transfers. |
| Arbitration | Procedure to ensure that only one master at a time controls the bus. Winning arbitration ensures that the message is not corrupted. |
| Synchronization | Procedure to synchronize the clocks of two or more devices on the bus. |
| Idle | No master is controlling the bus, and both SDA and SCL lines are high. |
| Active | Any time one or more master devices are controlling the bus. |
| Addressed Slave | Slave device that has received a matching address and is actively being clocked by a master. |
| Matching Address | Address byte that is clocked into a slave that matches the value stored in SSPxADD. |
| Write Request | Slave receives a matching address with R/$\overline{W}$ bit clear, and is ready to clock in data. |
| Read Request | Master sends an address byte with the R/$\overline{W}$ bit set, indicating that it wishes to clock data out of the Slave. This data is the next and all following bytes until a Restart or Stop. |
| Clock Stretching | When a device on the bus hold SCL low to stall communication. |
| Bus Collision | Any time the SDA line is sampled low by the module while it is outputting and expected high state. |


© 2013-2015 Microchip Technology Inc.

### 23.4.5 START CONDITION

The $I^2C$ specification defines a Start condition as a transition of SDA from a high to a low state while SCL line is high. A Start condition is always generated by the master and signifies the transition of the bus from an Idle to an Active state. Figure 23-12 shows wave forms for Start and Stop conditions.

A bus collision can occur on a Start condition if the module samples the SDA line low before asserting it low. This does not conform to the $I^2C$ Specification that states no bus collision can occur on a Start.

### 23.4.6 STOP CONDITION

A Stop condition is a transition of the SDA line from low-to-high state while the SCL line is high.

**Note:** At least one SCL low time must appear before a Stop is valid, therefore, if the SDA line goes low then high again while the SCL line stays high, only the Start condition is detected.

### 23.4.7 RESTART CONDITION

A Restart is valid any time that a Stop would be valid. A master can issue a Restart if it wishes to hold the bus after terminating the current transfer. A Restart has the same effect on the slave that a Start would, resetting all slave logic and preparing it to clock in an address. The master may want to address the same or another slave. Figure 23-13 shows the wave form for a Restart condition.

In 10-bit Addressing Slave mode a Restart is required for the master to clock data out of the addressed slave. Once a slave has been fully addressed, matching both high and low address bytes, the master can issue a Restart and the high address byte with the R/$\overline{W}$ bit set. The slave logic will then hold the clock and prepare to clock out data.

After a full match with R/$\overline{W}$ clear in 10-bit mode, a prior match flag is set and maintained until a Stop condition, a high address with R/$\overline{W}$ clear, or high address match fails.

### 23.4.8 START/STOP CONDITION INTERRUPT MASKING

The SCIE and PCIE bits of the SSPxCON3 register can enable the generation of an interrupt in Slave modes that do not typically support this function. Slave modes where interrupt on Start and Stop detect are already enabled, these bits will have no effect.

**FIGURE 23-12: $I^2C$ START AND STOP CONDITIONS**

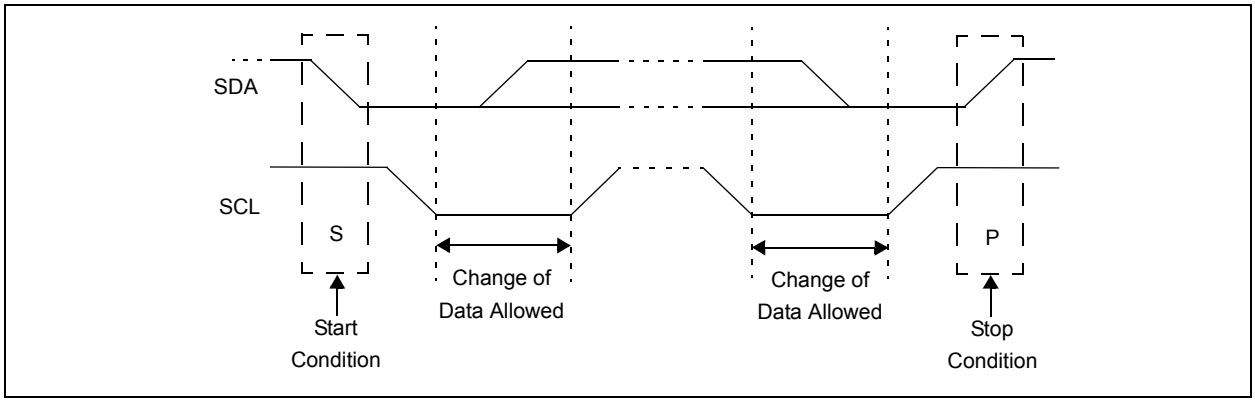
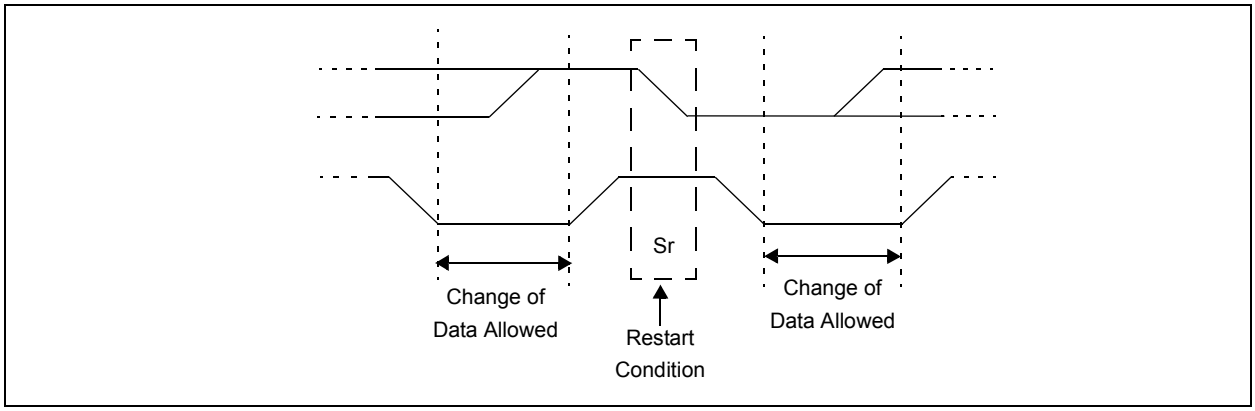



**FIGURE 23-13: $I^2C$ RESTART CONDITION**



#### 23.4.9 ACKNOWLEDGE SEQUENCE

The ninth SCL pulse for any transferred byte in $I^2C$ is dedicated as an Acknowledge. It allows receiving devices to respond back to the transmitter by pulling the SDA line low. The transmitter must release control of the line during this time to shift in the response. The Acknowledge ($\overline{ACK}$) is an active-low signal, pulling the SDA line low indicates to the transmitter that the device has received the transmitted data and is ready to receive more.

The result of an $\overline{ACK}$ is placed in the ACKSTAT bit of the SSPCON2 register.

Slave software, when the AHEN and DHEN bits are set, allow the user to set the $\overline{ACK}$ value sent back to the transmitter. The ACKDT bit of the SSPCON2 register is set/cleared to determine the response.

Slave hardware will generate an $\overline{ACK}$ response if the AHEN and DHEN bits of the SSPxCON3 register are clear.

There are certain conditions where an $\overline{ACK}$ will not be sent by the slave. If the BF bit of the SSPxSTAT register or the SSPOV bit of the SSPxCON1 register are set when a byte is received.

When the module is addressed, after the eighth falling edge of SCL on the bus, the ACKTIM bit of the SSPxCON3 register is set. The ACKTIM bit indicates the acknowledge time of the active bus. The ACKTIM Status bit is only active when the AHEN bit or DHEN bit is enabled.

## 23.5 $I^2C$ SLAVE MODE OPERATION

The MSSP Slave mode operates in one of four modes selected by the SSPM bits of SSPxCON1 register. The modes can be divided into 7-bit and 10-bit Addressing mode. 10-bit Addressing modes operate the same as 7-bit with some additional overhead for handling the larger addresses.

Modes with Start and Stop bit interrupts operate the same as the other modes with SSPIF additionally getting set upon detection of a Start, Restart, or Stop condition.

#### 23.5.1 SLAVE MODE ADDRESSES

The SSPxADD register (Register 23-6) contains the Slave mode address. The first byte received after a Start or Restart condition is compared against the value stored in this register. If the byte matches, the value is loaded into the SSPxBUF register and an interrupt is generated. If the value does not match, the module goes idle and no indication is given to the software that anything happened.

The SSP Mask register (Register 23-5) affects the address matching process. See **Section 23.5.9 "SSP Mask Register"** for more information.

##### 23.5.1.1 $I^2C$ Slave 7-bit Addressing Mode

In 7-bit Addressing mode, the LSb of the received data byte is ignored when determining if there is an address match.

##### 23.5.1.2 $I^2C$ Slave 10-bit Addressing Mode

In 10-bit Addressing mode, the first received byte is compared to the binary value of '1 1 1 1 0 A9 A8 0'. A9 and A8 are the two MSb's of the 10-bit address and stored in bits 2 and 1 of the SSPxADD register.

After the acknowledge of the high byte the UA bit is set and SCL is held low until the user updates SSPxADD with the low address. The low address byte is clocked in and all eight bits are compared to the low address value in SSPxADD. Even if there is not an address match; SSPIF and UA are set, and SCL is held low until SSPxADD is updated to receive a high byte again. When SSPxADD is updated the UA bit is cleared. This ensures the module is ready to receive the high address byte on the next communication.

A high and low address match as a write request is required at the start of all 10-bit addressing communication. A transmission can be initiated by issuing a Restart once the slave is addressed, and clocking in the high address with the R/$\overline{W}$ bit set. The slave hardware will then acknowledge the read request and prepare to clock out data. This is only valid for a slave after it has received a complete high and low address byte match.


© 2013-2015 Microchip Technology Inc.

### 23.5.2 SLAVE RECEPTION

When the R/$\overline{W}$ bit of a matching received address byte is clear, the R/$\overline{W}$ bit of the SSPxSTAT register is cleared. The received address is loaded into the SSPxBUF register and acknowledged.

When the overflow condition exists for a received address, then not Acknowledge is given. An overflow condition is defined as either bit BF of the SSPxSTAT register is set, or bit SSPOV of the SSPxCON1 register is set. The BOEN bit of the SSPxCON3 register modifies this operation. For more information see Register 23-4.

An MSSP interrupt is generated for each transferred data byte. Flag bit, SSPIF, must be cleared by software.

When the SEN bit of the SSPCON2 register is set, SCL will be held low (clock stretch) following each received byte. The clock must be released by setting the CKP bit of the SSPxCON1 register, except sometimes in 10-bit mode. See **Section 23.5.6.2 "10-bit Addressing Mode"** for more detail.

#### 23.5.2.1 7-bit Addressing Reception

This section describes a standard sequence of events for the MSSP module configured as an $I^2C$ slave in 7-bit Addressing mode. Figure 23-14 and Figure 23-15 is used as a visual reference for this description.

This is a step by step process of what typically must be done to accomplish $I^2C$ communication.

1. Start bit detected.
2. S bit of SSPxSTAT is set; SSPIF is set if interrupt on Start detect is enabled.
3. Matching address with R/$\overline{W}$ bit clear is received.
4. The slave pulls SDA low sending an $\overline{ACK}$ to the master, and sets SSPIF bit.
5. Software clears the SSPIF bit.
6. Software reads received address from SSPxBUF clearing the BF flag.
7. If SEN = 1; Slave software sets CKP bit to release the SCL line.
8. The master clocks out a data byte.
9. Slave drives SDA low sending an $\overline{ACK}$ to the master, and sets SSPIF bit.
10. Software clears SSPIF.
11. Software reads the received byte from SSPxBUF clearing BF.
12. Steps 8-12 are repeated for all received bytes from the master.
13. Master sends Stop condition, setting P bit of SSPxSTAT, and the bus goes idle.

#### 23.5.2.2 7-bit Reception with AHEN and DHEN

Slave device reception with AHEN and DHEN set operate the same as without these options with extra interrupts and clock stretching added after the eighth falling edge of SCL. These additional interrupts allow the slave software to decide whether it wants to $\overline{ACK}$ the receive address or data byte, rather than the hardware. This functionality adds support for PMBus™ that was not present on previous versions of this module.

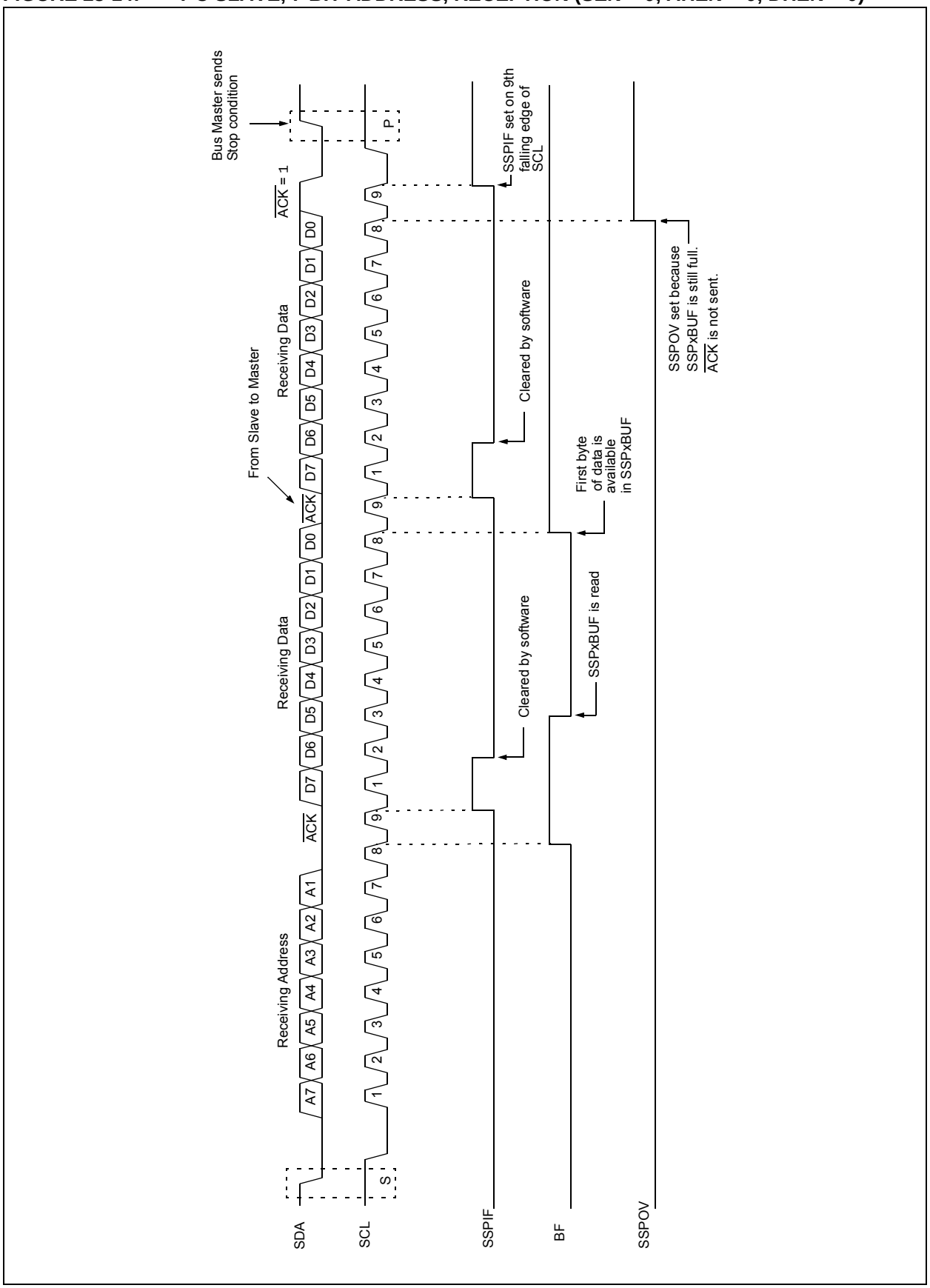
This list describes the steps that need to be taken by slave software to use these options for $I^2C$ communication. Figure 23-16 displays a module using both address and data holding. Figure 23-17 includes the operation with the SEN bit of the SSPCON2 register set.

1. S bit of SSPxSTAT is set; SSPIF is set if interrupt on Start detect is enabled.
2. Matching address with R/$\overline{W}$ bit clear is clocked in. SSPIF is set and CKP cleared after the eighth falling edge of SCL.
3. Slave clears the SSPIF.
4. Slave can look at the ACKTIM bit of the SSPxCON3 register to determine if the SSPIF was after or before the $\overline{ACK}$.
5. Slave reads the address value from SSPxBUF, clearing the BF flag.
6. Slave sets $\overline{ACK}$ value clocked out to the master by setting ACKDT.
7. Slave releases the clock by setting CKP.
8. SSPIF is set after an $\overline{ACK}$, not after a NACK.
9. If SEN = 1 the slave hardware will stretch the clock after the $\overline{ACK}$.
10. Slave clears SSPIF.

**Note:** SSPIF is still set after the ninth falling edge of SCL even if there is no clock stretching and BF has been cleared. Only if NACK is sent to master is SSPIF not set

11. SSPIF set and CKP cleared after eighth falling edge of SCL for a received data byte.
12. Slave looks at ACKTIM bit of SSPxCON3 to determine the source of the interrupt.
13. Slave reads the received data from SSPxBUF clearing BF.
14. Steps 7-14 are the same for each received data byte.
15. Communication is ended by either the slave sending an $\overline{ACK}$ = 1, or the master sending a Stop condition. If a Stop is sent and Interrupt on Stop Detect is disabled, the slave will only know by polling the P bit of the SSTSTAT register.

**FIGURE 23-14: I²C SLAVE, 7-BIT ADDRESS, RECEPTION (SEN = 0, AHEN = 0, DHEN = 0)**





© 2013-2015 Microchip Technology Inc.

**FIGURE 23-15: I²C SLAVE, 7-BIT ADDRESS, RECEPTION (SEN = 1, AHEN = 0, DHEN = 0)**

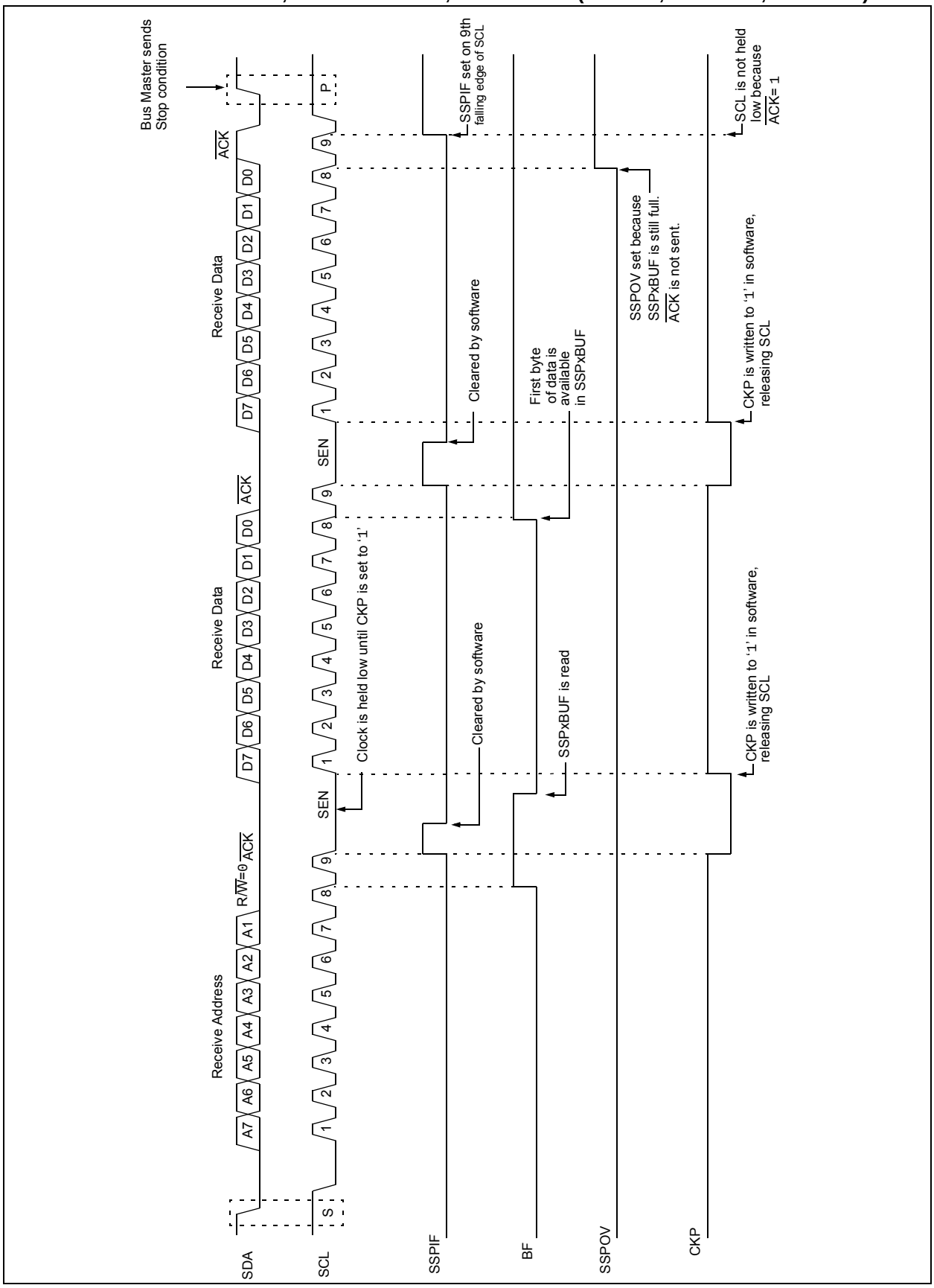
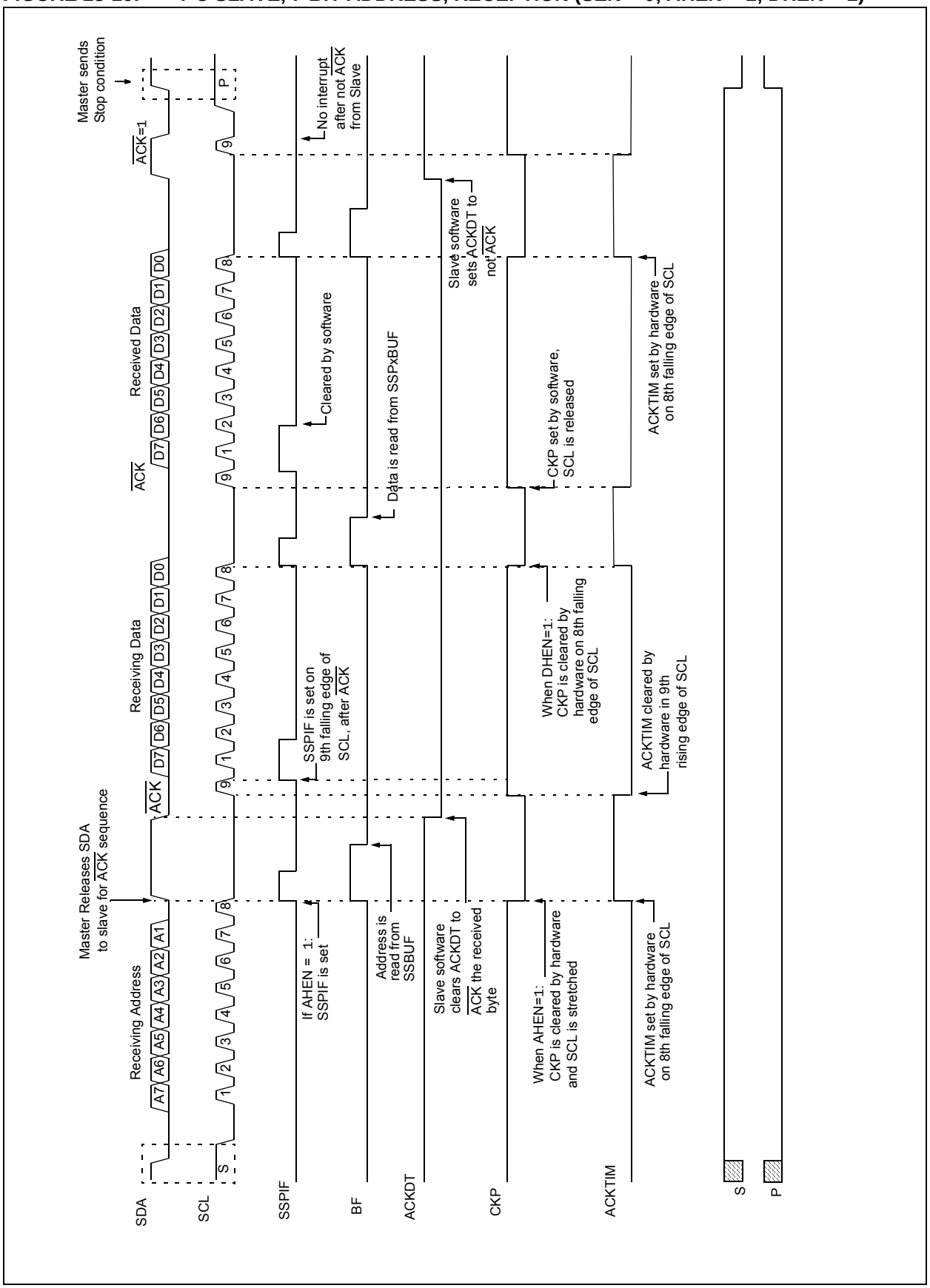


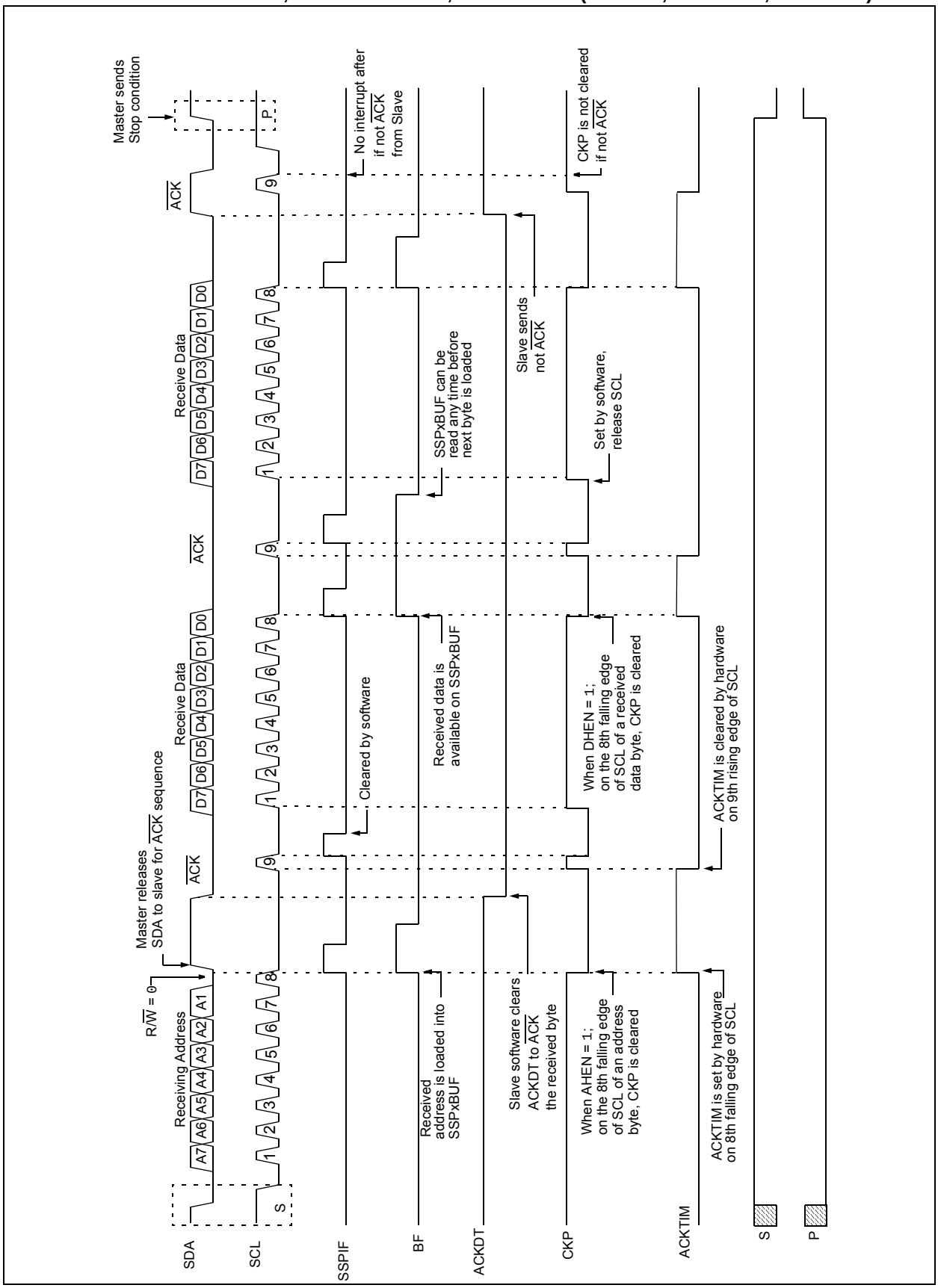
**FIGURE 23-16: I²C SLAVE, 7-BIT ADDRESS, RECEPTION (SEN = 0, AHEN = 1, DHEN = 1)**





© 2013-2015 Microchip Technology Inc.

**FIGURE 23-17: I²C SLAVE, 7-BIT ADDRESS, RECEPTION (SEN = 1, AHEN = 1, DHEN = 1)**



### 23.5.3 SLAVE 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 SSPxSTAT register is set. The received address is loaded into the SSPxBUF register, and an $\overline{ACK}$ pulse is sent by the slave on the ninth bit.

Following the $\overline{ACK}$, slave hardware clears the CKP bit and the SCL pin is held low (see **Section 23.5.6 "Clock Stretching"** for more detail). By stretching the clock, the master will be unable to assert another clock pulse until the slave is done preparing the transmit data.

The transmit data must be loaded into the SSPxBUF register which also loads the SSPSR register. Then the SCL pin should be released by setting the CKP bit of the SSPxCON1 register. 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.

The $\overline{ACK}$ pulse from the master-receiver is latched on the rising edge of the ninth SCL input pulse. This $\overline{ACK}$ value is copied to the ACKSTAT bit of the SSPCON2 register. If ACKSTAT is set (not $\overline{ACK}$), then the data transfer is complete. In this case, when the not $\overline{ACK}$ is latched by the slave, the slave goes idle and waits for another occurrence of the Start bit. If the SDA line was low ($\overline{ACK}$), the next transmit data must be loaded into the SSPxBUF register. Again, the SCL pin must be released by setting bit CKP.

An MSSP interrupt is generated for each data transfer byte. The SSPIF bit must be cleared by software and the SSPxSTAT register is used to determine the status of the byte. The SSPIF bit is set on the falling edge of the ninth clock pulse.

#### 23.5.3.1 Slave Mode Bus Collision

A slave receives a Read request and begins shifting data out on the SDA line. If a bus collision is detected and the SBCDE bit of the SSPxCON3 register is set, the BCLIF bit of the PIR register is set. Once a bus collision is detected, the slave goes idle and waits to be addressed again. User software can use the BCLIF bit to handle a slave bus collision.

#### 23.5.3.2 7-bit Transmission

A master device can transmit a read request to a slave, and then clock data out of the slave. The list below outlines what software for a slave will need to do to accomplish a standard transmission. Figure 23-18 can be used as a reference to this list.

1. Master sends a Start condition on SDA and SCL.
2. S bit of SSPxSTAT is set; SSPIF is set if interrupt on Start detect is enabled.
3. Matching address with R/$\overline{W}$ bit set is received by the Slave setting SSPIF bit.
4. Slave hardware generates an $\overline{ACK}$ and sets SSPIF.
5. SSPIF bit is cleared by user.
6. Software reads the received address from SSPxBUF, clearing BF.
7. R/$\overline{W}$ is set so CKP was automatically cleared after the $\overline{ACK}$.
8. The slave software loads the transmit data into SSPxBUF.
9. CKP bit is set releasing SCL, allowing the master to clock the data out of the slave.
10. SSPIF is set after the $\overline{ACK}$ response from the master is loaded into the ACKSTAT register.
11. SSPIF bit is cleared.
12. The slave software checks the ACKSTAT bit to see if the master wants to clock out more data.

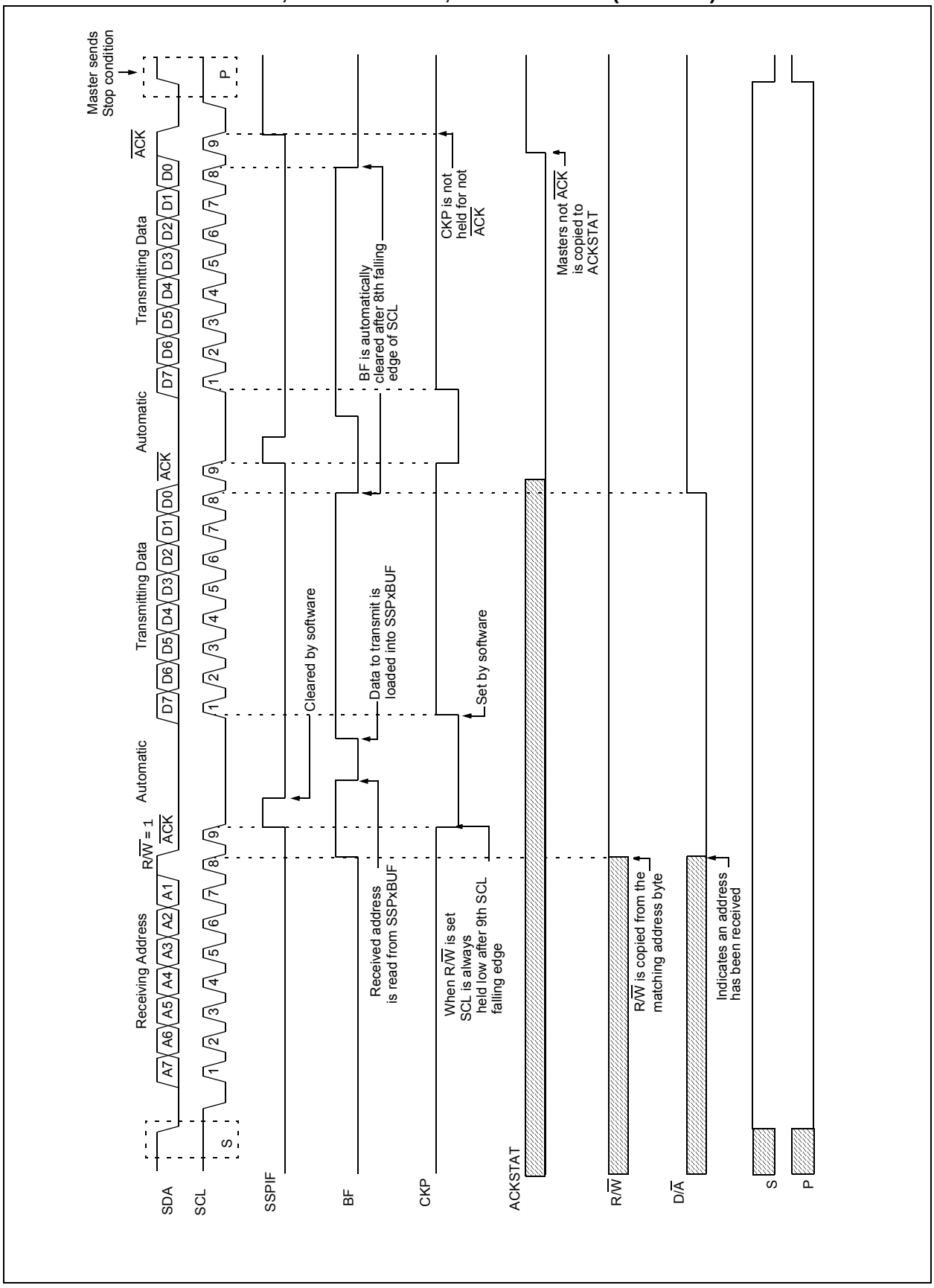
**Note 1:** If the master $\overline{ACK}$s the clock will be stretched.

**2:** ACKSTAT is the only bit updated on the rising edge of SCL (9th) rather than the falling.

13. Steps 9-13 are repeated for each transmitted byte.
14. If the master sends a not $\overline{ACK}$; the clock is not held, but SSPIF is still set.
15. The master sends a Restart condition or a Stop.
16. The slave is no longer addressed.


© 2013-2015 Microchip Technology Inc.

**FIGURE 23-18: I²C SLAVE, 7-BIT ADDRESS, TRANSMISSION (AHEN = 0)**



#### 23.5.3.3 7-bit Transmission with Address Hold Enabled

Setting the AHEN bit of the SSPxCON3 register enables additional clock stretching and interrupt generation after the eighth falling edge of a received matching address. Once a matching address has been clocked in, CKP is cleared and the SSPIF interrupt is set.

Figure 23-19 displays a standard waveform of a 7-bit address slave transmission with AHEN enabled.

1. Bus starts Idle.
2. Master sends Start condition; the S bit of SSPxSTAT is set; SSPIF is set if interrupt on Start detect is enabled.
3. Master sends matching address with R/$\overline{W}$ bit set. After the eighth falling edge of the SCL line the CKP bit is cleared and SSPIF interrupt is generated.
4. Slave software clears SSPIF.
5. Slave software reads ACKTIM bit of SSPxCON3 register, and R/$\overline{W}$ and D/$\overline{A}$ of the SSPxSTAT register to determine the source of the interrupt.
6. Slave reads the address value from the SSPxBUF register clearing the BF bit.
7. Slave software decides from this information if it wishes to $\overline{ACK}$ or not $\overline{ACK}$ and sets the ACKDT bit of the SSPCON2 register accordingly.
8. Slave sets the CKP bit releasing SCL.
9. Master clocks in the $\overline{ACK}$ value from the slave.
10. Slave hardware automatically clears the CKP bit and sets SSPIF after the $\overline{ACK}$ if the R/$\overline{W}$ bit is set.
11. Slave software clears SSPIF.
12. Slave loads value to transmit to the master into SSPxBUF setting the BF bit.

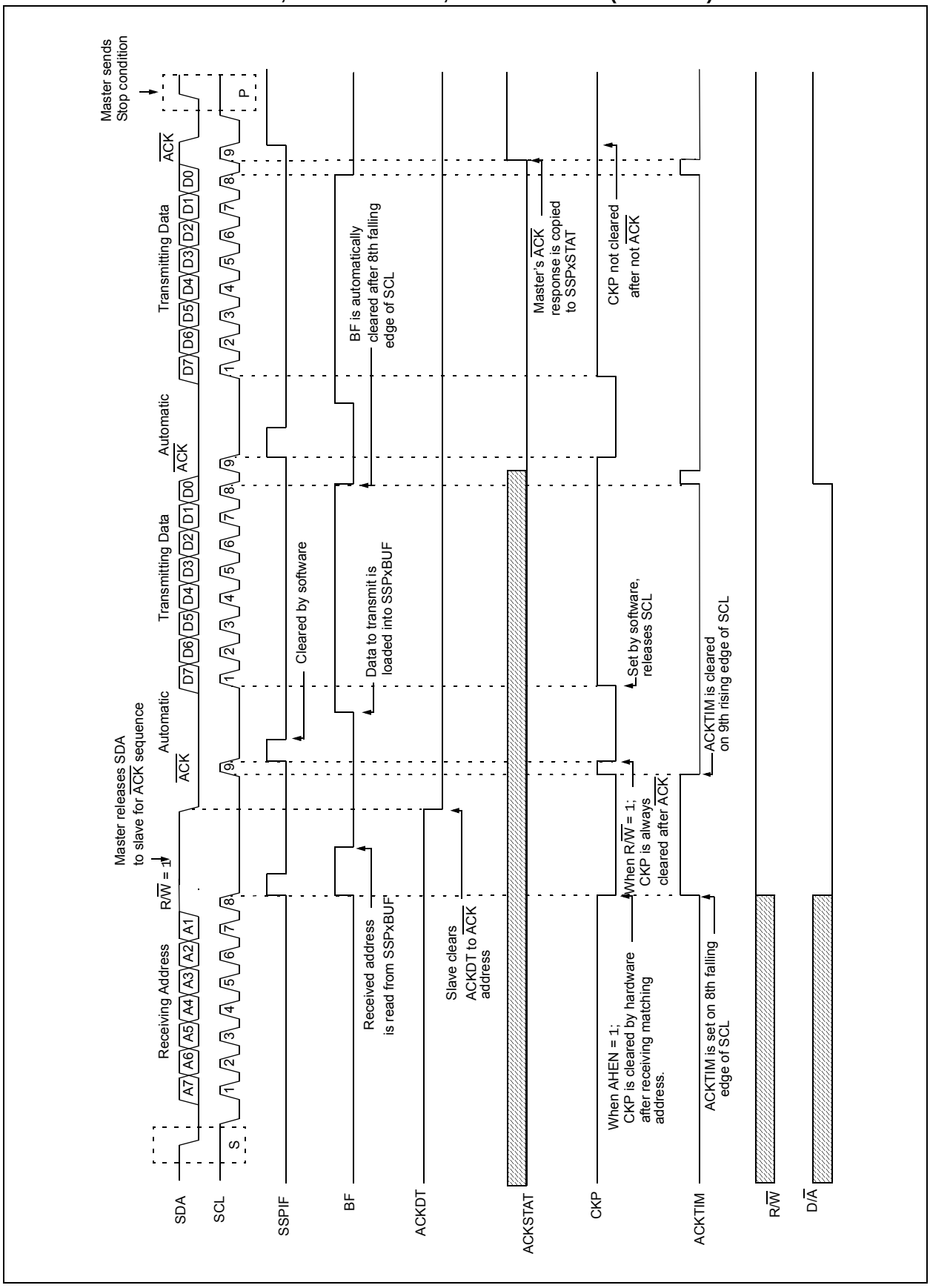
**Note:** SSPxBUF cannot be loaded until after the $\overline{ACK}$.

13. Slave sets the CKP bit releasing the clock.
14. Master clocks out the data from the slave and sends an $\overline{ACK}$ value on the ninth SCL pulse.
15. Slave hardware copies the $\overline{ACK}$ value into the ACKSTAT bit of the SSPCON2 register.
16. Steps 10-15 are repeated for each byte transmitted to the master from the slave.
17. If the master sends a not $\overline{ACK}$ the slave releases the bus allowing the master to send a Stop and end the communication.

**Note:** Master must send a not $\overline{ACK}$ on the last byte to ensure that the slave releases the SCL line to receive a Stop.


© 2013-2015 Microchip Technology Inc.

**FIGURE 23-19: I²C SLAVE, 7-BIT ADDRESS, TRANSMISSION (AHEN = 1)**



### 23.5.4 SLAVE MODE 10-BIT ADDRESS RECEPTION

This section describes a standard sequence of events for the MSSP module configured as an $I^2C$ slave in 10-bit Addressing mode.

Figure 23-20 is used as a visual reference for this description.

This is a step by step process of what must be done by slave software to accomplish $I^2C$ communication.

1. Bus starts Idle.
2. Master sends Start condition; S bit of SSPxSTAT is set; SSPIF is set if interrupt on Start detect is enabled.
3. Master sends matching high address with R/$\overline{W}$ bit clear; UA bit of the SSPxSTAT register is set.
4. Slave sends $\overline{ACK}$ and SSPIF is set.
5. Software clears the SSPIF bit.
6. Software reads received address from SSPxBUF clearing the BF flag.
7. Slave loads low address into SSPxADD, releasing SCL.
8. Master sends matching low address byte to the slave; UA bit is set.

**Note:** Updates to the SSPxADD register are not allowed until after the $\overline{ACK}$ sequence.

9. Slave sends $\overline{ACK}$ and SSPIF is set.

**Note:** If the low address does not match, SSPIF and UA are still set so that the slave software can set SSPxADD back to the high address. BF is not set because there is no match. CKP is unaffected.

10. Slave clears SSPIF.
11. Slave reads the received matching address from SSPxBUF clearing BF.
12. Slave loads high address into SSPxADD.
13. Master clocks a data byte to the slave and clocks out the slaves $\overline{ACK}$ on the ninth SCL pulse; SSPIF is set.
14. If SEN bit of SSPCON2 is set, CKP is cleared by hardware and the clock is stretched.
15. Slave clears SSPIF.
16. Slave reads the received byte from SSPxBUF clearing BF.
17. If SEN is set the slave sets CKP to release the SCL.
18. Steps 13-17 repeat for each received byte.
19. Master sends Stop to end the transmission.

### 23.5.5 10-BIT ADDRESSING WITH ADDRESS OR DATA HOLD

Reception using 10-bit addressing with AHEN or DHEN set is the same as with 7-bit modes. The only difference is the need to update the SSPxADD register using the UA bit. All functionality, specifically when the CKP bit is cleared and SCL line is held low are the same. Figure 23-21 can be used as a reference of a slave in 10-bit addressing with AHEN set.

Figure 23-22 shows a standard waveform for a slave transmitter in 10-bit Addressing mode.


© 2013-2015 Microchip Technology Inc.

**FIGURE 23-20: I²C SLAVE, 10-BIT ADDRESS, RECEPTION (SEN = 1, AHEN = 0, DHEN = 0)**

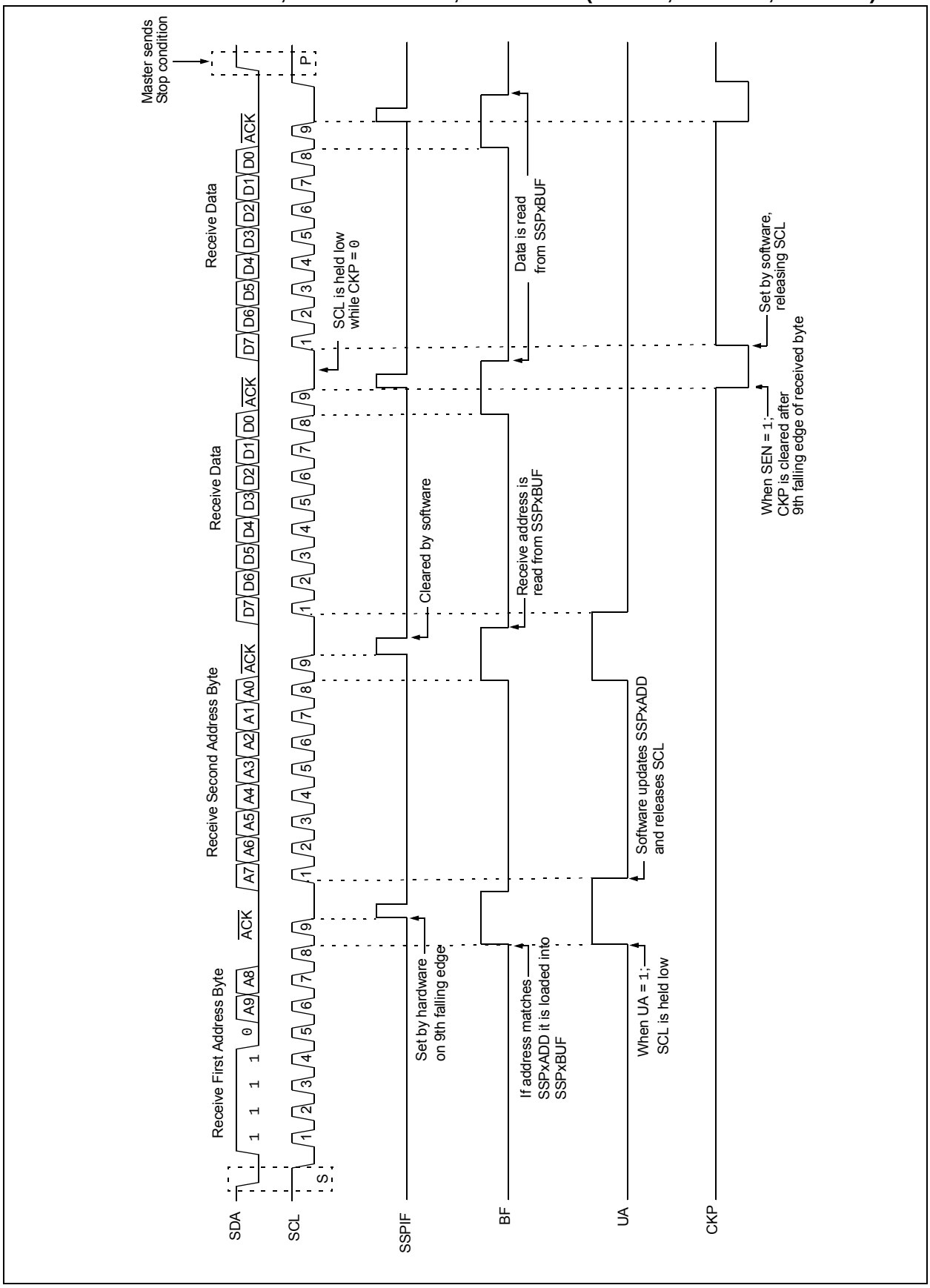
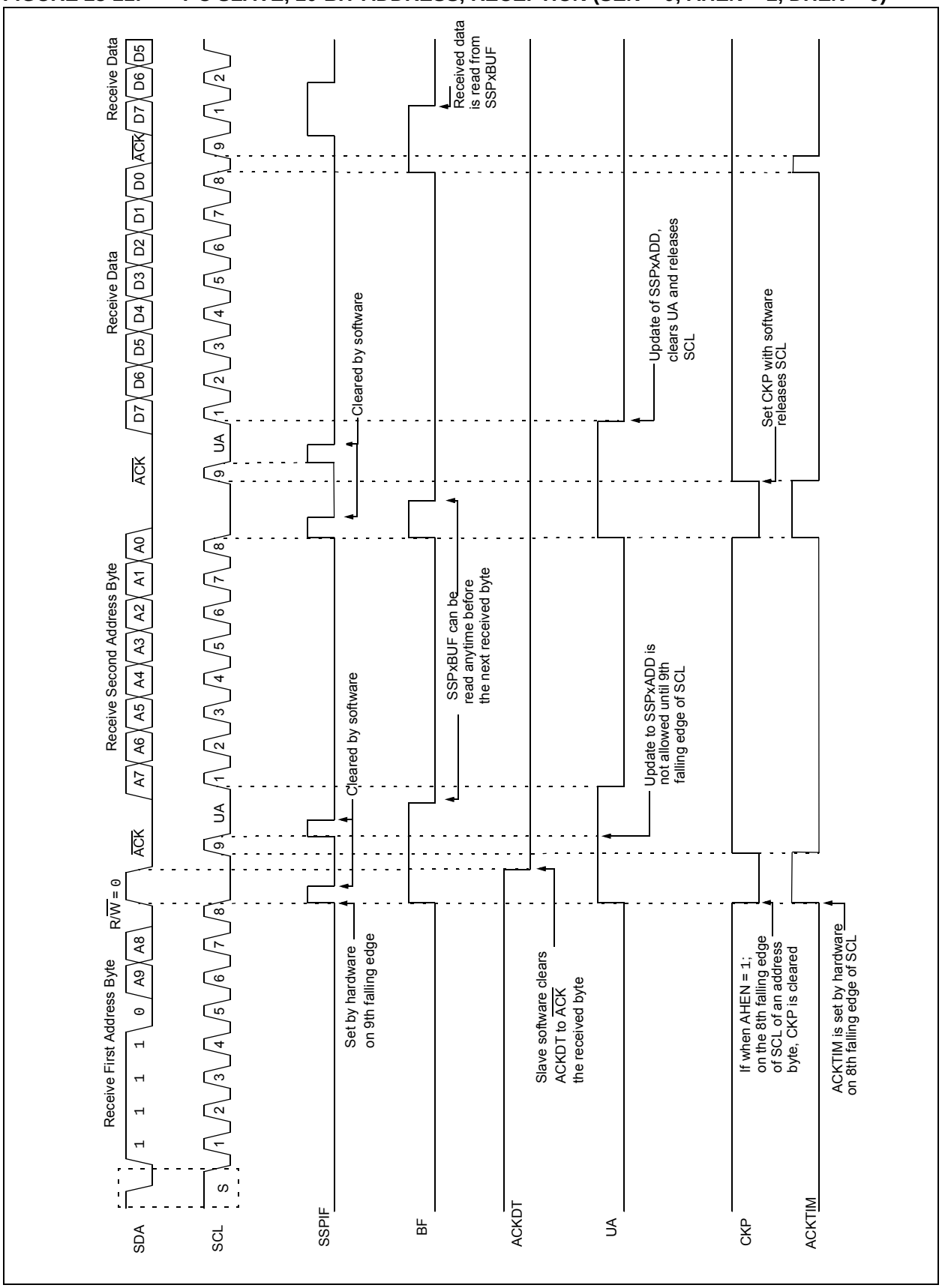


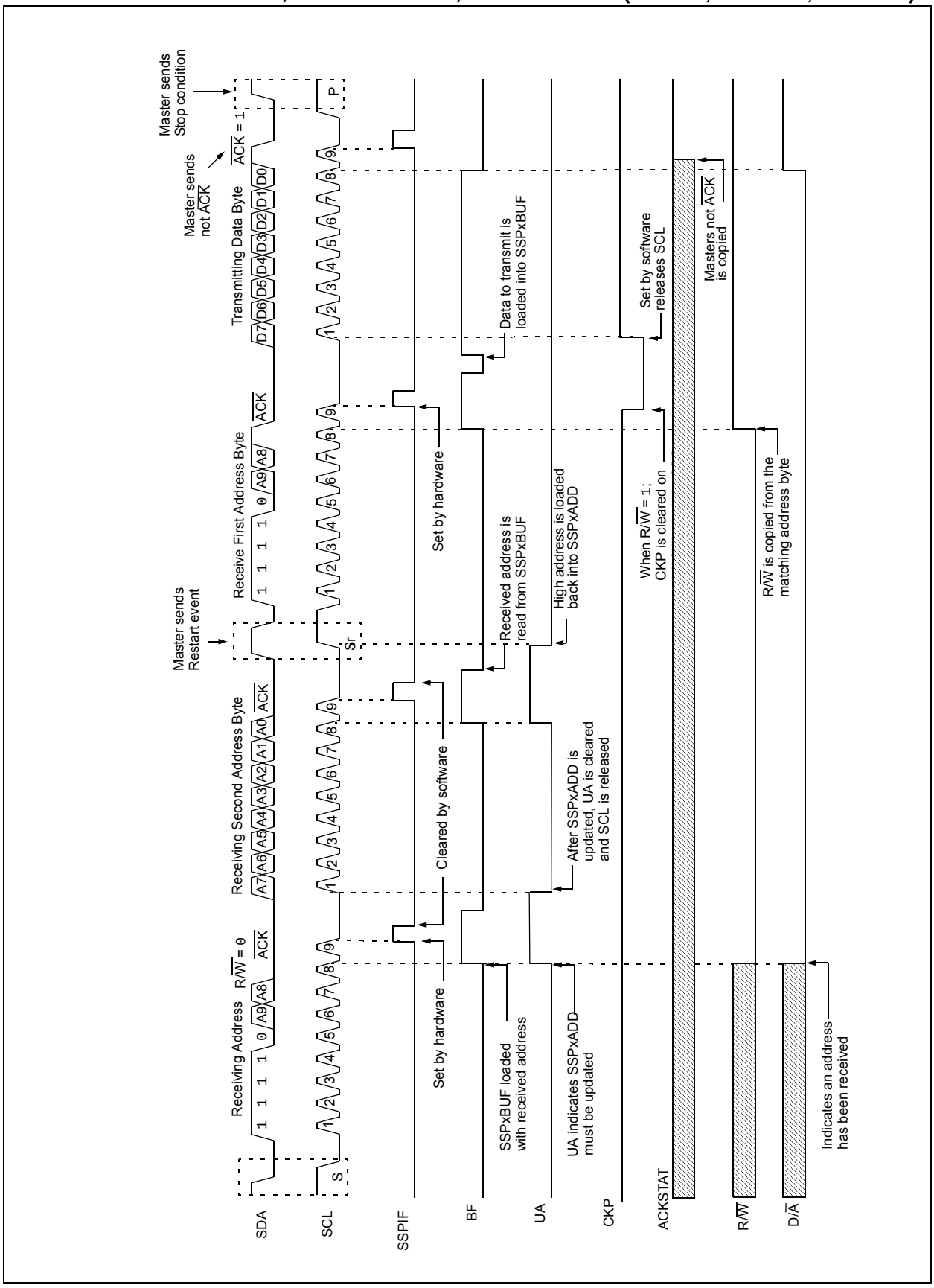
**FIGURE 23-21: I²C SLAVE, 10-BIT ADDRESS, RECEPTION (SEN = 0, AHEN = 1, DHEN = 0)**





© 2013-2015 Microchip Technology Inc.

**FIGURE 23-22: I²C SLAVE, 10-BIT ADDRESS, TRANSMISSION (SEN = 0, AHEN = 0, DHEN = 0)**



### 23.5.6 CLOCK STRETCHING

Clock stretching occurs when a device on the bus holds the SCL line low, effectively pausing communication. The slave may stretch the clock to allow more time to handle data or prepare a response for the master device. A master device is not concerned with stretching as anytime it is active on the bus and not transferring data it is stretching. Any stretching done by a slave is invisible to the master software and handled by the hardware that generates SCL.

The CKP bit of the SSPxCON1 register is used to control stretching in software. Any time the CKP bit is cleared, the module will wait for the SCL line to go low and then hold it. Setting CKP will release SCL and allow more communication.

#### 23.5.6.1 Normal Clock Stretching

Following an $\overline{ACK}$ if the R/$\overline{W}$ bit of SSPxSTAT is set, a read request, the slave hardware will clear CKP. This allows the slave time to update SSPxBUF with data to transfer to the master. If the SEN bit of SSPCON2 is set, the slave hardware will always stretch the clock after the $\overline{ACK}$ sequence. Once the slave is ready; CKP is set by software and communication resumes.

**Note 1:** The BF bit has no effect on if the clock will be stretched or not. This is different than previous versions of the module that would not stretch the clock, clear CKP, if SSPxBUF was read before the ninth falling edge of SCL.

**2:** Previous versions of the module did not stretch the clock for a transmission if SSPxBUF was loaded before the ninth falling edge of SCL. It is now always cleared for read requests.

#### 23.5.6.2 10-bit Addressing Mode

In 10-bit Addressing mode, when the UA bit is set the clock is always stretched. This is the only time the SCL is stretched without CKP being cleared. SCL is released immediately after a write to SSPxADD.

**Note:** Previous versions of the module did not stretch the clock if the second address byte did not match.

#### 23.5.6.3 Byte NACKing

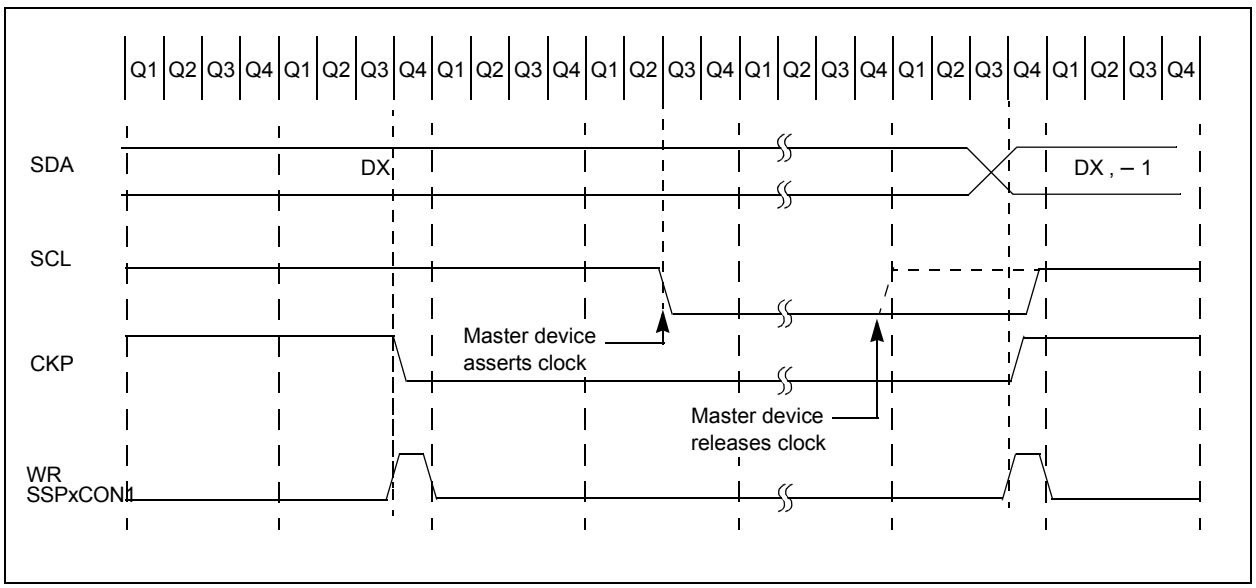
When AHEN bit of SSPxCON3 is set; CKP is cleared by hardware after the eighth falling edge of SCL for a received matching address byte. When DHEN bit of SSPxCON3 is set; CKP is cleared after the eighth falling edge of SCL for received data.

Stretching after the eighth falling edge of SCL allows the slave to look at the received address or data and decide if it wants to ACK the received data.

### 23.5.7 CLOCK SYNCHRONIZATION AND THE CKP BIT

Any time the CKP bit is cleared, the module will wait for the SCL line to go low and then hold it. However, clearing 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 released SCL. This ensures that a write to the CKP bit will not violate the minimum high time requirement for SCL (see Figure 23-23).

**FIGURE 23-23: CLOCK SYNCHRONIZATION TIMING**





© 2013-2015 Microchip Technology Inc.

### 23.5.8 GENERAL CALL ADDRESS SUPPORT

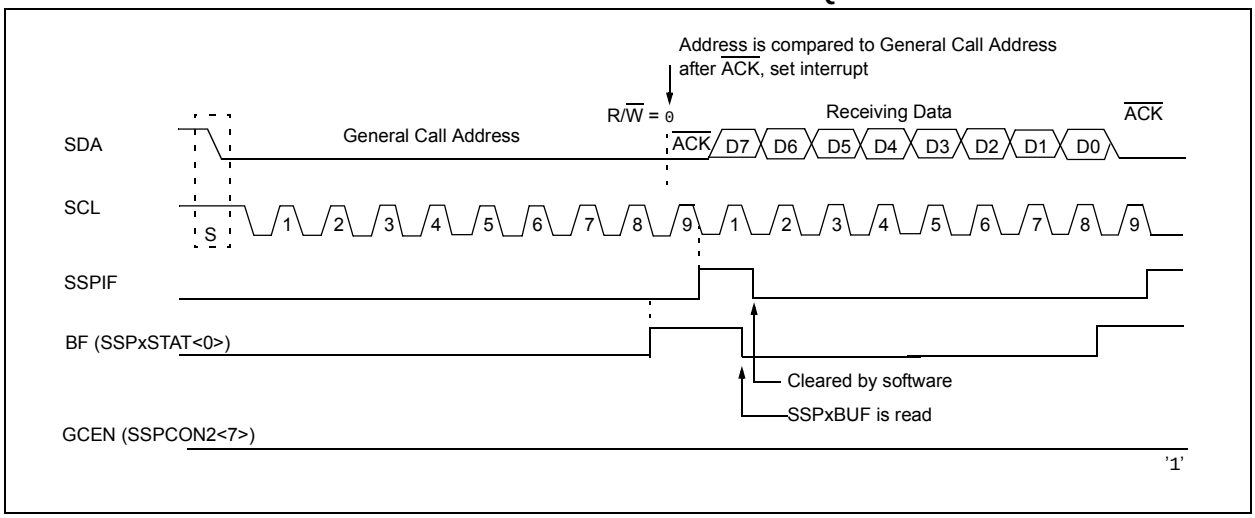
The addressing procedure for the I²C bus is such that the first byte after the Start condition usually determines which device will be the slave addressed by the master device. The exception is the general call address which can address all devices. When this address is used, all devices should, in theory, respond with an acknowledge.

The general call address is a reserved address in the I²C protocol, defined as address 0x00. When the GCEN bit of the SSPCON2 register is set, the slave module will automatically ACK the reception of this address regardless of the value stored in SSPxADD. After the slave clocks in an address of all zeros with the R/W bit clear, an interrupt is generated and slave software can read SSPxBUF and respond. Figure 23-24 shows a general call reception sequence.

In 10-bit Address mode, the UA bit will not be set on the reception of the general call address. The slave will prepare to receive the second byte as data, just as it would in 7-bit mode.

If the AHEN bit of the SSPxCON3 register is set, just as with any other address reception, the slave hardware will stretch the clock after the eighth falling edge of SCL. The slave must then set its ACKDT value and release the clock with communication progressing as it would normally.

**FIGURE 23-24: SLAVE MODE GENERAL CALL ADDRESS SEQUENCE**




### 23.5.9 SSP MASK REGISTER

An SSP Mask (SSPMSK) register (Register 23-5) is available in I²C 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 of the received address 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.

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.

## 23.6 I²C Master Mode

Master mode is enabled by setting and clearing the appropriate SSPM bits in the SSPxCON1 register and by setting the SSPEN bit. In Master mode, the SDA and SCK pins must be configured as inputs. The MSSP peripheral hardware will override the output driver TRIS controls when necessary to drive the pins low.

Master mode of operation is supported by 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 MSSP module is disabled. Control of the I²C bus may be taken when the P bit is set, or the bus is Idle.

In Firmware Controlled Master mode, user code conducts all I²C bus operations based on Start and Stop bit condition detection. Start and Stop condition detection is the only active circuitry in this mode. All other communication is done by the user software directly manipulating the SDA and SCL lines.

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

- Start condition detected
- Stop condition detected
- Data transfer byte transmitted/received
- Acknowledge transmitted/received
- Repeated Start generated

**Note 1:** The MSSP module, when configured in I²C Master mode, does not allow queuing of events. For instance, the user is not allowed to initiate a Start condition and immediately write the SSPxBUF register to initiate transmission before the Start condition is complete. In this case, the SSPxBUF will not be written to and the WCOL bit will be set, indicating that a write to the SSPxBUF did not occur

**2:** When in Master mode, Start/Stop detection is masked and an interrupt is generated when the SEN/PEN bit is cleared and the generation is complete.

### 23.6.1 I²C MASTER MODE OPERATION

The master device generates all of the serial clock pulses and the Start and Stop conditions. A transfer is ended with a Stop condition or with a Repeated Start condition. Since the Repeated Start condition is also the beginning of the next serial transfer, the I²C bus will not be released.

In Master Transmitter mode, serial data is output through SDA, while SCL outputs the serial clock. The first byte transmitted contains the slave address of the receiving device (7 bits) and the Read/Write (R/$\overline{W}$) bit. In this case, the R/$\overline{W}$ bit will be logic '0'. Serial data is transmitted eight bits at a time. After each byte is transmitted, an Acknowledge bit is received. Start and Stop conditions are output to indicate the beginning and the end of a serial transfer.

In Master Receive mode, the first byte transmitted contains the slave address of the transmitting device (7 bits) and the R/$\overline{W}$ bit. In this case, the R/$\overline{W}$ bit will be logic '1'. Thus, the first byte transmitted is a 7-bit slave address followed by a '1' to indicate the receive bit. Serial data is received via SDA, while SCL outputs the serial clock. Serial data is received eight bits at a time. After each byte is received, an Acknowledge bit is transmitted. Start and Stop conditions indicate the beginning and end of transmission.

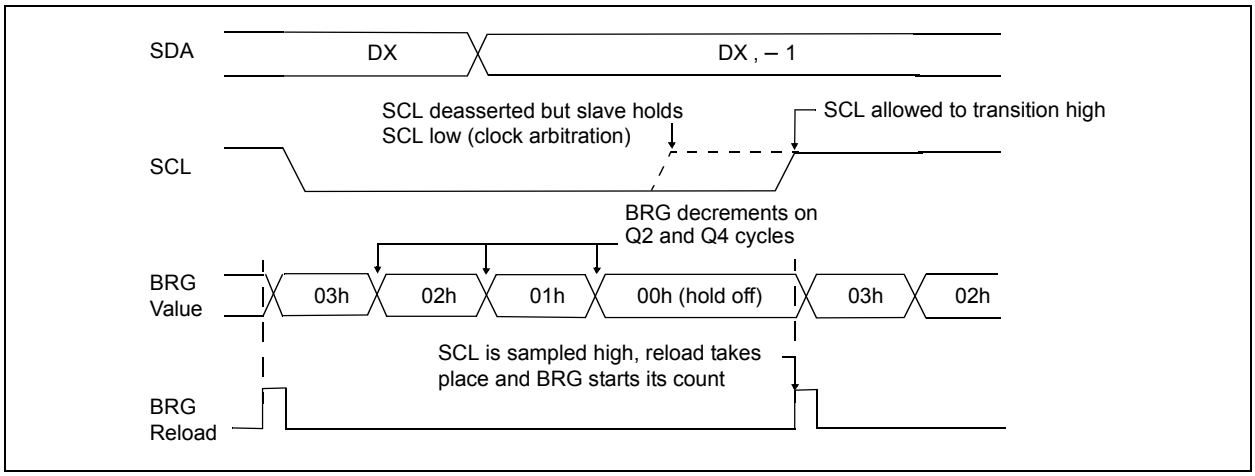
A Baud Rate Generator is used to set the clock frequency output on SCL. See **Section 23.7 "Baud Rate Generator"** for more detail.


© 2013-2015 Microchip Technology Inc.

### 23.6.2 CLOCK ARBITRATION

Clock arbitration occurs when the master, during any receive, transmit or Repeated Start/Stop condition, releases the SCL pin (SCL allowed to float high). When the SCL pin is allowed to float high, the Baud Rate Generator (BRG) is suspended from counting until the SCL pin is actually sampled high. When the SCL pin is sampled high, the Baud Rate Generator is reloaded with the contents of SSPxADD<7:0> and begins counting. This ensures that the SCL high time will always be at least one BRG rollover count in the event that the clock is held low by an external device (Figure 23-25).

**FIGURE 23-25: BAUD RATE GENERATOR TIMING WITH CLOCK ARBITRATION**




### 23.6.3 WCOL STATUS FLAG

If the user writes the SSPxBUF when a Start, Restart, Stop, Receive or Transmit sequence is in progress, the WCOL is set and the contents of the buffer are unchanged (the write does not occur). Any time the WCOL bit is set it indicates that an action on SSPxBUF was attempted while the module was not idle.

| | |
|---|---|
| **Note:** | Because queuing of events is not allowed, writing to the lower five bits of SSPCON2 is disabled until the Start condition is complete. |

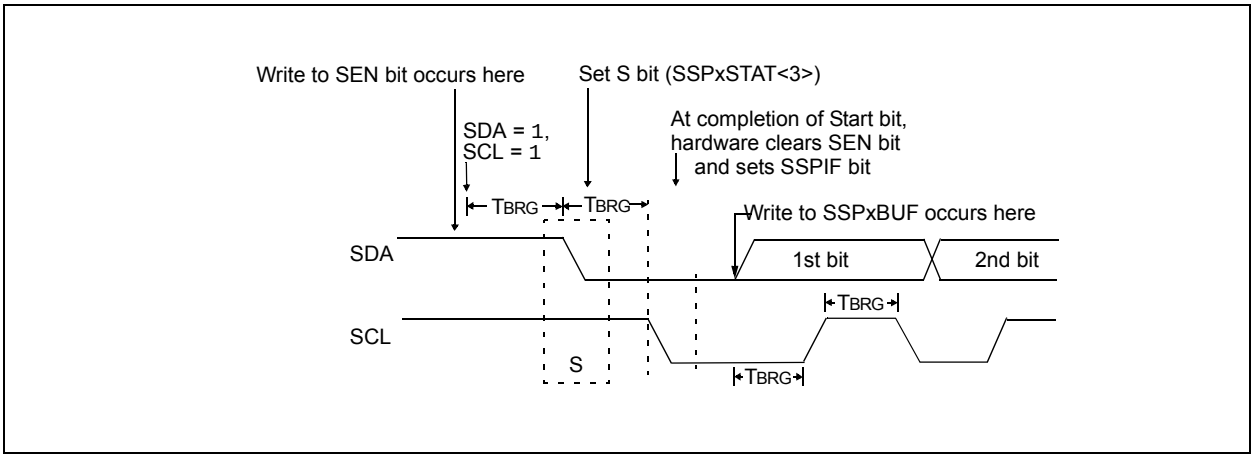
### 23.6.4 I²C MASTER MODE START CONDITION TIMING

To initiate a Start condition (Figure 23-26), the user sets the Start Enable bit, SEN bit of the SSPCON2 register. If the SDA and SCL pins are sampled high, the Baud Rate Generator is reloaded with the contents of SSPxADD<7:0> and starts its count. If SCL and SDA are both sampled high when the Baud Rate Generator times out (TBRG), the SDA pin is driven low. The action of the SDA being driven low while SCL is high is the Start condition and causes the S bit of the SSPxSTAT1 register to be set. Following this, the Baud Rate Generator is reloaded with the contents of SSPxADD<7:0> and resumes its count. When the Baud Rate Generator times out (TBRG), the SEN bit of the SSPCON2 register will be automatically cleared by hardware; the Baud Rate Generator is suspended, leaving the SDA line held low and the Start condition is complete.

**Note 1:** If at the beginning of the Start condition, the SDA and SCL pins are already sampled low, or if during the Start condition, the SCL line is sampled low before the SDA line is driven low, a bus collision occurs, the Bus Collision Interrupt Flag, BCLIF, is set, the Start condition is aborted and the I²C module is reset into its Idle state.

**2:** The Philips I²C specification states that a bus collision cannot occur on a Start.

**FIGURE 23-26: FIRST START BIT TIMING**





© 2013-2015 Microchip Technology Inc.

### 23.6.5 I²C MASTER MODE REPEATED START CONDITION TIMING

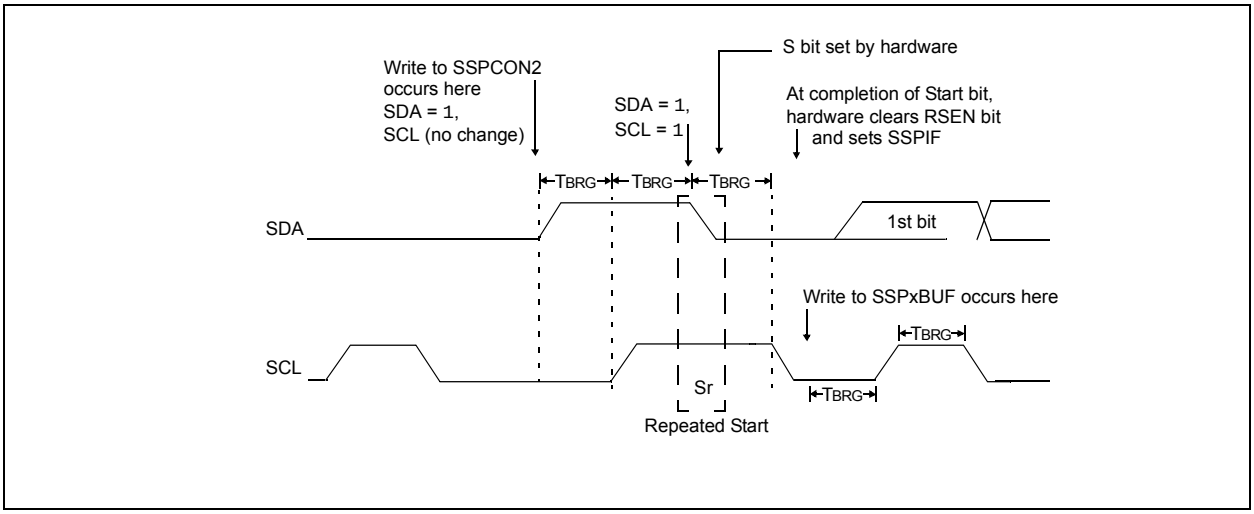
A Repeated Start condition (Figure 23-27) occurs when the RSEN bit of the SSPCON2 register is programmed high and the master state machine is no longer active. When the RSEN bit is set, the SCL pin is asserted low. When the SCL pin is sampled low, the Baud Rate Generator is loaded and begins counting. The SDA pin is released (brought high) for one Baud Rate Generator count (TBRG). When the Baud Rate Generator times out, if SDA is sampled high, the SCL pin will be deasserted (brought high). When SCL is sampled high, the Baud Rate Generator is reloaded and begins counting. SDA and SCL must be sampled high for one TBRG. This action is then followed by assertion of the SDA pin (SDA = 0) for one TBRG while SCL is high. SCL is asserted low. Following this, the RSEN bit of the SSPCON2 register will be automatically cleared and the Baud Rate Generator will not be reloaded, leaving the SDA pin held low. As soon as a Start condition is detected on the SDA and SCL pins, the S bit of the SSPxSTAT register will be set. The SSPIF bit will not be set until the Baud Rate Generator has timed out.

**Note 1:** If RSEN is programmed while any other event is in progress, it will not take effect.

**2:** A bus collision during the Repeated Start condition occurs if:

- SDA is sampled low when SCL goes from low-to-high.
- SCL goes low before SDA is asserted low. This may indicate that another master is attempting to transmit a data '1'.

**FIGURE 23-27: REPEATED START CONDITION WAVEFORM**



### 23.6.6 I²C MASTER MODE TRANSMISSION

Transmission of a data byte, a 7-bit address or the other half of a 10-bit address is accomplished by simply writing a value to the SSPxBUF register. This action will set the Buffer Full flag bit, BF, and allow the Baud Rate Generator to begin counting and start the next transmission. Each bit of address/data will be shifted out onto the SDA pin after the falling edge of SCL is asserted. SCL is held low for one Baud Rate Generator rollover count (TBRG). Data should be valid before SCL is released high. When the SCL pin is released high, it is held that way for TBRG. The data on the SDA pin must remain stable for that duration and some hold time after the next falling edge of SCL. After the eighth bit is shifted out (the falling edge of the eighth clock), the BF flag is cleared and the master releases SDA. This allows the slave device being addressed to respond with an $\overline{ACK}$ bit during the ninth bit time if an address match occurred, or if data was received properly. The status of $\overline{ACK}$ is written into the ACKSTAT bit on the rising edge of the ninth clock. If the master receives an Acknowledge, the Acknowledge Status bit, ACKSTAT, is cleared. If not, the bit is set. After the ninth clock, the SSPIF bit is set and the master clock (Baud Rate Generator) is suspended until the next data byte is loaded into the SSPxBUF, leaving SCL low and SDA unchanged (Figure 23-28).

After the write to the SSPxBUF, each bit of the address will be shifted out on the falling edge of SCL until all seven address bits and the R/$\overline{W}$ bit are completed. On the falling edge of the eighth clock, the master will release the SDA pin, allowing the slave to respond with an Acknowledge. On the falling edge of the ninth clock, the master will sample the SDA pin to see if the address was recognized by a slave. The status of the $\overline{ACK}$ bit is loaded into the ACKSTAT Status bit of the SSPCON2 register. Following the falling edge of the ninth clock transmission of the address, the SSPIF is set, the BF flag is cleared and the Baud Rate Generator is turned off until another write to the SSPxBUF takes place, holding SCL low and allowing SDA to float.

#### 23.6.6.1 BF Status Flag

In Transmit mode, the BF bit of the SSPxSTAT register is set when the CPU writes to SSPxBUF and is cleared when all eight bits are shifted out.

#### 23.6.6.2 WCOL Status Flag

If the user writes the SSPxBUF when a transmit is already in progress (i.e., SSPSR is still shifting out a data byte), the WCOL bit is set and the contents of the buffer are unchanged (the write does not occur).

WCOL must be cleared by software before the next transmission.

#### 23.6.6.3 ACKSTAT Status Flag

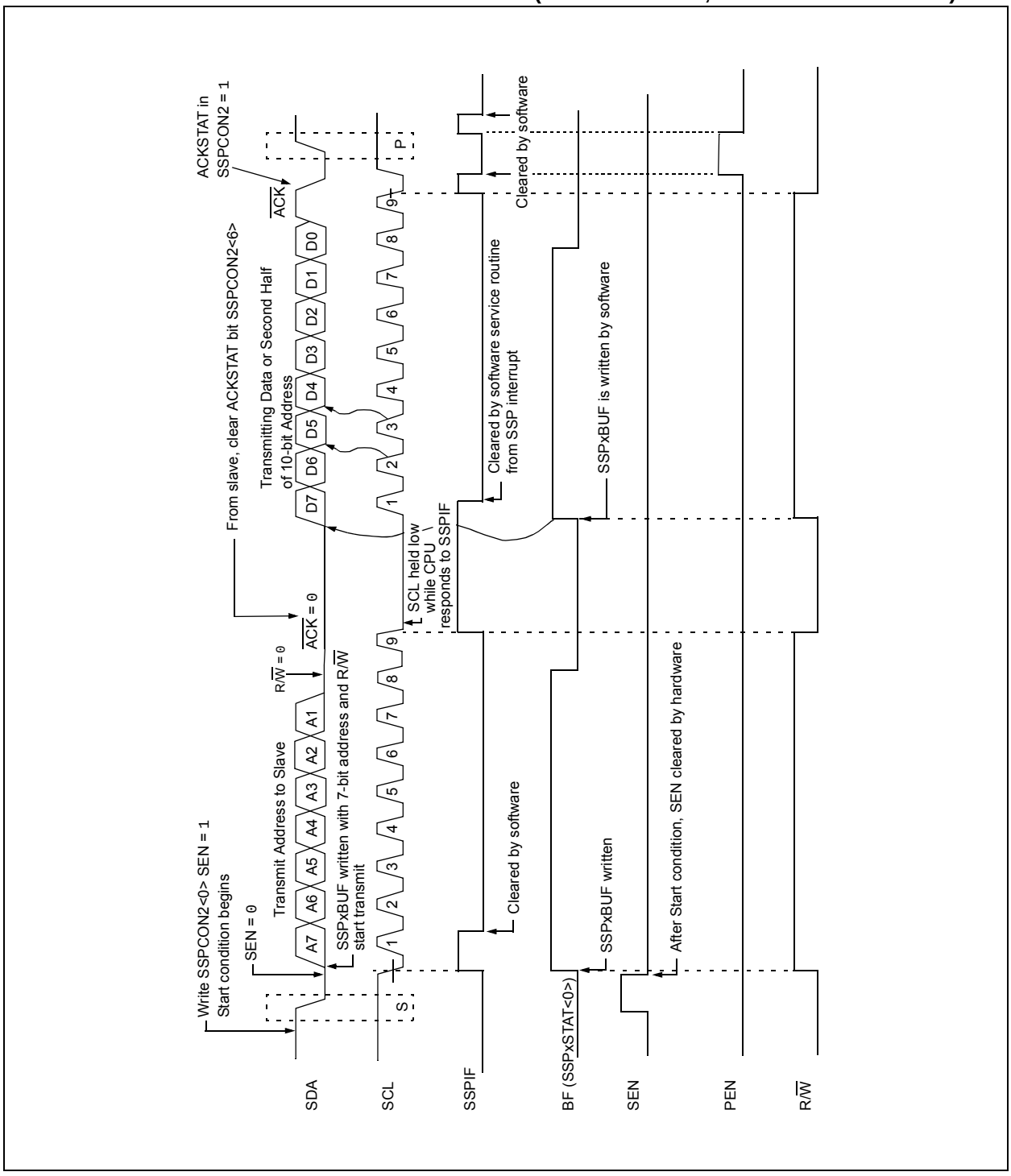
In Transmit mode, the ACKSTAT bit of the SSPCON2 register is cleared when the slave has sent an Acknowledge ($\overline{ACK}$ = 0) and is set when the slave does not Acknowledge ($\overline{ACK}$ = 1). A slave sends an Acknowledge when it has recognized its address (including a general call), or when the slave has properly received its data.

#### 23.6.6.4 Typical transmit sequence:

1. The user generates a Start condition by setting the SEN bit of the SSPCON2 register.
2. SSPIF is set by hardware on completion of the Start.
3. SSPIF is cleared by software.
4. The MSSP module will wait the required start time before any other operation takes place.
5. The user loads the SSPxBUF with the slave address to transmit.
6. Address is shifted out the SDA pin until all eight bits are transmitted. Transmission begins as soon as SSPxBUF is written to.
7. The MSSP module shifts in the $\overline{ACK}$ bit from the slave device and writes its value into the ACKSTAT bit of the SSPCON2 register.
8. The MSSP module generates an interrupt at the end of the ninth clock cycle by setting the SSPIF bit.
9. The user loads the SSPxBUF with eight bits of data.
10. Data is shifted out the SDA pin until all eight bits are transmitted.
11. The MSSP module shifts in the $\overline{ACK}$ bit from the slave device and writes its value into the ACKSTAT bit of the SSPCON2 register.
12. Steps 8-11 are repeated for all transmitted data bytes.
13. The user generates a Stop or Restart condition by setting the PEN or RSEN bits of the SSPCON2 register. Interrupt is generated once the Stop/Restart condition is complete.


© 2013-2015 Microchip Technology Inc.

**FIGURE 23-28: I²C MASTER MODE WAVEFORM (TRANSMISSION, 7 OR 10-BIT ADDRESS)**



### 23.6.7 I²C MASTER MODE RECEPTION

Master mode reception (Figure 23-29) is enabled by programming the Receive Enable bit, RCEN bit of the SSP1CON2 register.

| **Note:** | The MSSP module must be in an Idle state before the RCEN bit is set or the RCEN bit will be disregarded. |
|---|---|

The Baud Rate Generator begins counting and on each rollover, the state of the SCL pin changes (high-to-low/low-to-high) and data is shifted into the SSPSR. After the falling edge of the eighth clock, the receive enable flag is automatically cleared, the contents of the SSPSR are loaded into the SSPxBUF, the BF flag bit is set, the SSPIF flag bit is set and the Baud Rate Generator is suspended from counting, holding SCL low. The MSSP is now in Idle state awaiting the next command. When the buffer is read by the CPU, the BF flag bit is automatically cleared. The user can then send an Acknowledge bit at the end of reception by setting the Acknowledge Sequence Enable, ACKEN bit of the SSPCON2 register.

#### 23.6.7.1 BF Status Flag

In receive operation, the BF bit is set when an address or data byte is loaded into SSPxBUF from SSPSR. It is cleared when the SSPxBUF register is read.

#### 23.6.7.2 SSPOV Status Flag

In receive operation, the SSPOV bit is set when eight bits are received into the SSPSR and the BF flag bit is already set from a previous reception.

#### 23.6.7.3 WCOL Status Flag

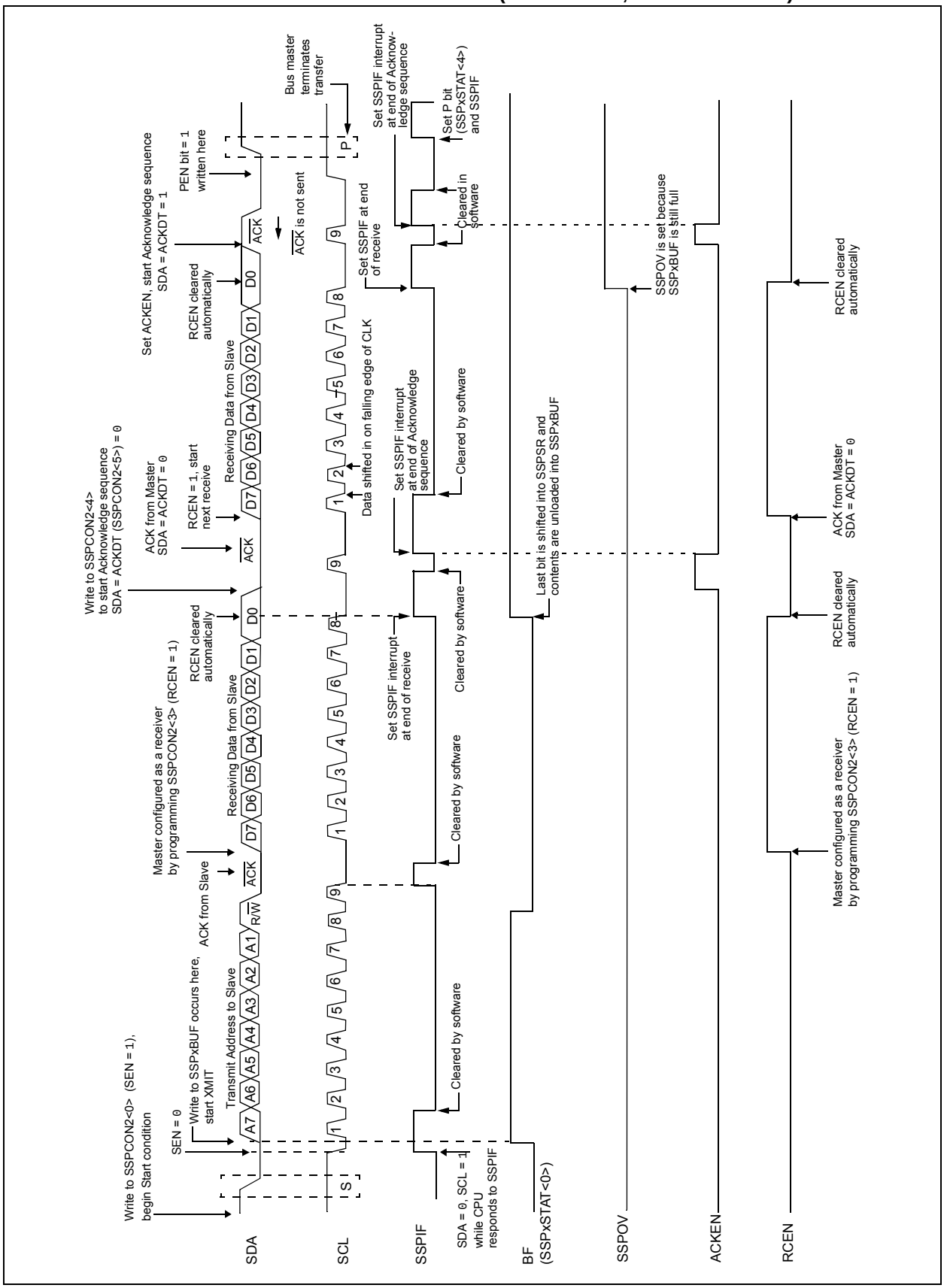
If the user writes the SSPxBUF when a receive is already in progress (i.e., SSPSR is still shifting in a data byte), the WCOL bit is set and the contents of the buffer are unchanged (the write does not occur).

#### 23.6.7.4 Typical Receive Sequence:

1. The user generates a Start condition by setting the SEN bit of the SSPCON2 register.
2. SSPIF is set by hardware on completion of the Start.
3. SSPIF is cleared by software.
4. User writes SSPxBUF with the slave address to transmit and the R/$\overline{W}$ bit set.
5. Address is shifted out the SDA pin until all eight bits are transmitted. Transmission begins as soon as SSPxBUF is written to.
6. The MSSP module shifts in the $\overline{ACK}$ bit from the slave device and writes its value into the ACKSTAT bit of the SSPCON2 register.
7. The MSSP module generates an interrupt at the end of the ninth clock cycle by setting the SSPIF bit.
8. User sets the RCEN bit of the SSPCON2 register and the master clocks in a byte from the slave.
9. After the eighth falling edge of SCL, SSPIF and BF are set.
10. Master clears SSPIF and reads the received byte from SSPUF, clears BF.
11. Master sets $\overline{ACK}$ value sent to slave in ACKDT bit of the SSPCON2 register and initiates the $\overline{ACK}$ by setting the ACKEN bit.
12. Master's $\overline{ACK}$ is clocked out to the slave and SSPIF is set.
13. User clears SSPIF.
14. Steps 8-13 are repeated for each received byte from the slave.
15. Master sends a not $\overline{ACK}$ or Stop to end communication.


© 2013-2015 Microchip Technology Inc.

**FIGURE 23-29: I²C MASTER MODE WAVEFORM (RECEPTION, 7-BIT ADDRESS)**



### 23.6.8 ACKNOWLEDGE SEQUENCE TIMING

An Acknowledge sequence is enabled by setting the Acknowledge Sequence Enable bit, ACKEN bit of the SSPCON2 register. When this bit is set, the SCL pin is pulled low and the contents of the Acknowledge data bit are presented on the SDA pin. If the user wishes to generate an Acknowledge, then the ACKDT bit should be cleared. If not, the user should set the ACKDT bit before starting an Acknowledge sequence. The Baud Rate Generator then counts for one rollover period (TBRG) and the SCL pin is deasserted (pulled high). When the SCL pin is sampled high (clock arbitration), the Baud Rate Generator counts for TBRG. The SCL pin is then pulled low. Following this, the ACKEN bit is automatically cleared, the Baud Rate Generator is turned off and the MSSP module then goes into Idle mode (Figure 23-30).

#### 23.6.8.1 WCOL Status Flag

If the user writes the SSPxBUF when an Acknowledge sequence is in progress, then WCOL bit is set and the contents of the buffer are unchanged (the write does not occur).

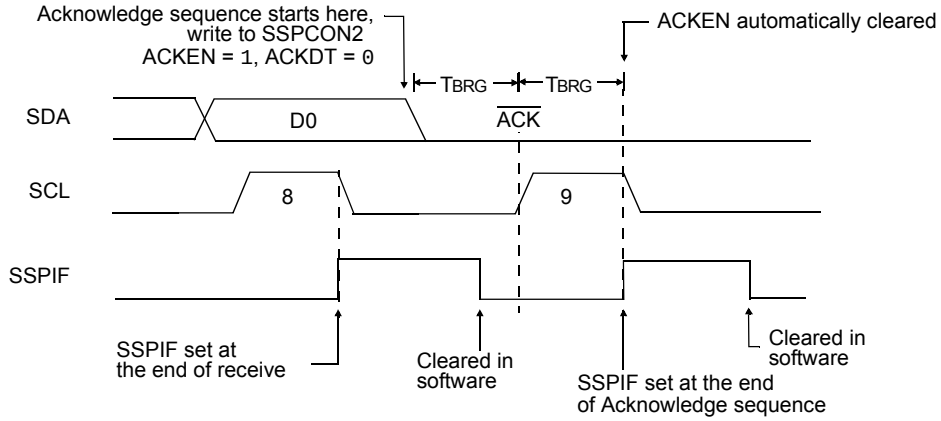
### 23.6.9 STOP CONDITION TIMING

A Stop bit is asserted on the SDA pin at the end of a receive/transmit by setting the Stop Sequence Enable bit, PEN bit of the SSPCON2 register. At the end of a receive/transmit, the SCL line is held low after the falling edge of the ninth clock. When the PEN bit is set, the master will assert the SDA line low. When the SDA line is sampled low, the Baud Rate Generator is reloaded and counts down to '0'. When the Baud Rate Generator times out, the SCL pin will be brought high and one TBRG (Baud Rate Generator rollover count) later, the SDA pin will be deasserted. When the SDA pin is sampled high while SCL is high, the P bit of the SSPxSTAT register is set. A TBRG later, the PEN bit is cleared and the SSPIF bit is set (Figure 23-31).

#### 23.6.9.1 WCOL Status Flag

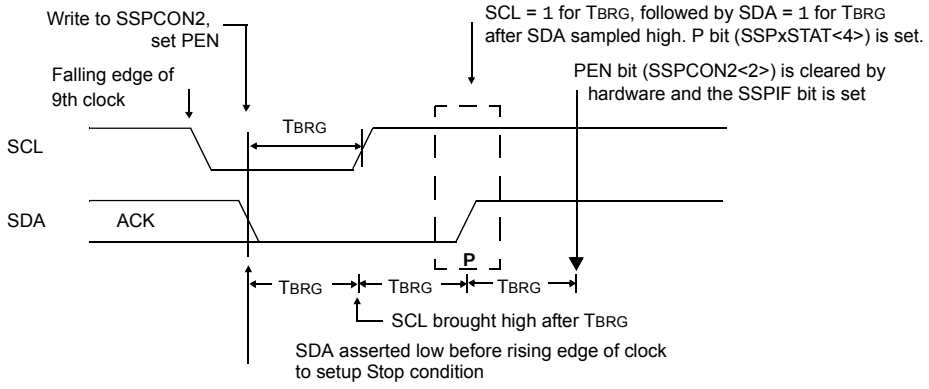
If the user writes the SSPxBUF when a Stop sequence is in progress, then the WCOL bit is set and the contents of the buffer are unchanged (the write does not occur).

**FIGURE 23-30: ACKNOWLEDGE SEQUENCE WAVEFORM**




**Note:** TBRG = one Baud Rate Generator period.

**FIGURE 23-31: STOP CONDITION RECEIVE OR TRANSMIT MODE**




**Note:** TBRG = one Baud Rate Generator period.


© 2013-2015 Microchip Technology Inc.

### 23.6.10 SLEEP OPERATION

While in Sleep mode, the I²C slave module can receive addresses or data and when an address match or complete byte transfer occurs, wake the processor from Sleep (if the MSSP interrupt is enabled).

### 23.6.11 EFFECTS OF A RESET

A Reset disables the MSSP module and terminates the current transfer.

### 23.6.12 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 MSSP module is disabled. Control of the I²C bus may be taken when the P bit of the SSPxSTAT register is set, or the bus is Idle, with 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 for arbitration to see if the signal level is the expected output level. This check is performed by hardware with the result placed in the BCLIF bit.

The states where arbitration can be lost are:

- Address Transfer
- Data Transfer
- A Start Condition
- A Repeated Start Condition
- An Acknowledge Condition

### 23.6.13 MULTI -MASTER COMMUNICATION, BUS COLLISION AND BUS ARBITRATION

Multi-Master mode support is achieved by bus arbitration. When the master outputs address/data bits onto the SDA pin, arbitration takes place when the master outputs a '1' on SDA, by letting SDA float high and another master asserts a '0'. When the SCL pin floats high, data should be stable. If the expected data on SDA is a '1' and the data sampled on the SDA pin is '0', then a bus collision has taken place. The master will set the Bus Collision Interrupt Flag, BCLIF and reset the I²C port to its Idle state (Figure 23-32).

If a transmit was in progress when the bus collision occurred, the transmission is halted, the BF flag is cleared, the SDA and SCL lines are deasserted and the SSPxBUF can be written to. When the user services the bus collision Interrupt Service Routine and if the I²C bus is free, the user can resume communication by asserting a Start condition.

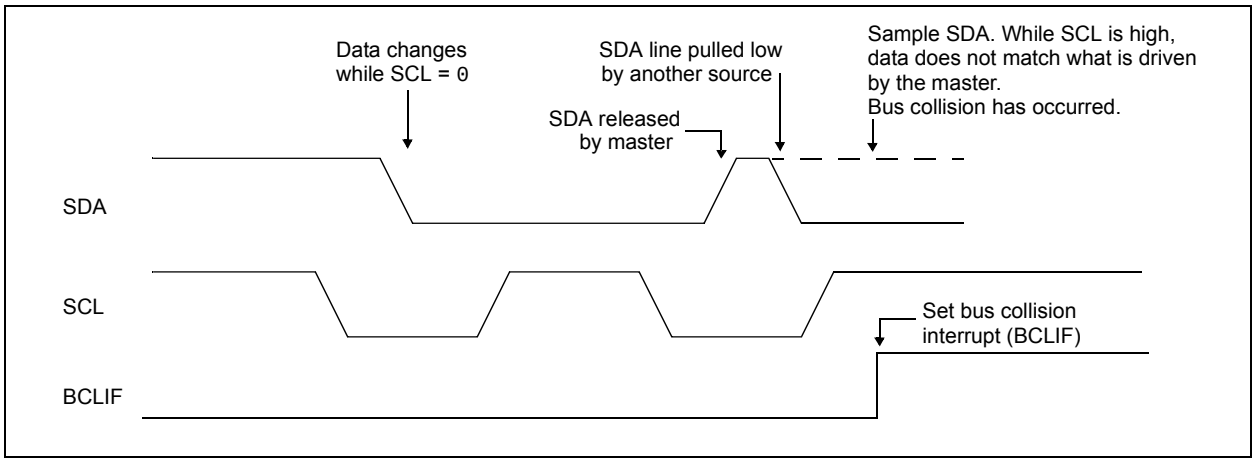
If a Start, Repeated Start, Stop or Acknowledge condition was in progress when the bus collision occurred, the condition is aborted, the SDA and SCL lines are deasserted and the respective control bits in the SSPCON2 register are cleared. When the user services the bus collision Interrupt Service Routine and if the I²C bus is free, the user can resume communication by asserting a Start condition.

The master will continue to monitor the SDA and SCL pins. If a Stop condition occurs, the SSPIF bit will be set.

A write to the SSPxBUF will start the transmission of data at the first data bit, regardless of where the transmitter left off when the bus collision occurred.

In Multi-Master mode, the interrupt generation on the detection of Start and Stop conditions allows the determination of when the bus is free. Control of the I²C bus can be taken when the P bit is set in the SSPxSTAT register, or the bus is Idle and the S and P bits are cleared.

**FIGURE 23-32: BUS COLLISION TIMING FOR TRANSMIT AND ACKNOWLEDGE**



#### 23.6.13.1 Bus Collision During a Start Condition

During a Start condition, a bus collision occurs if:

a) SDA or SCL are sampled low at the beginning of the Start condition (Figure 23-33).

b) SCL is sampled low before SDA is asserted low (Figure 23-34).

During a Start condition, both the SDA and the SCL pins are monitored.

If the SDA pin is already low, or the SCL pin is already low, then all of the following occur:

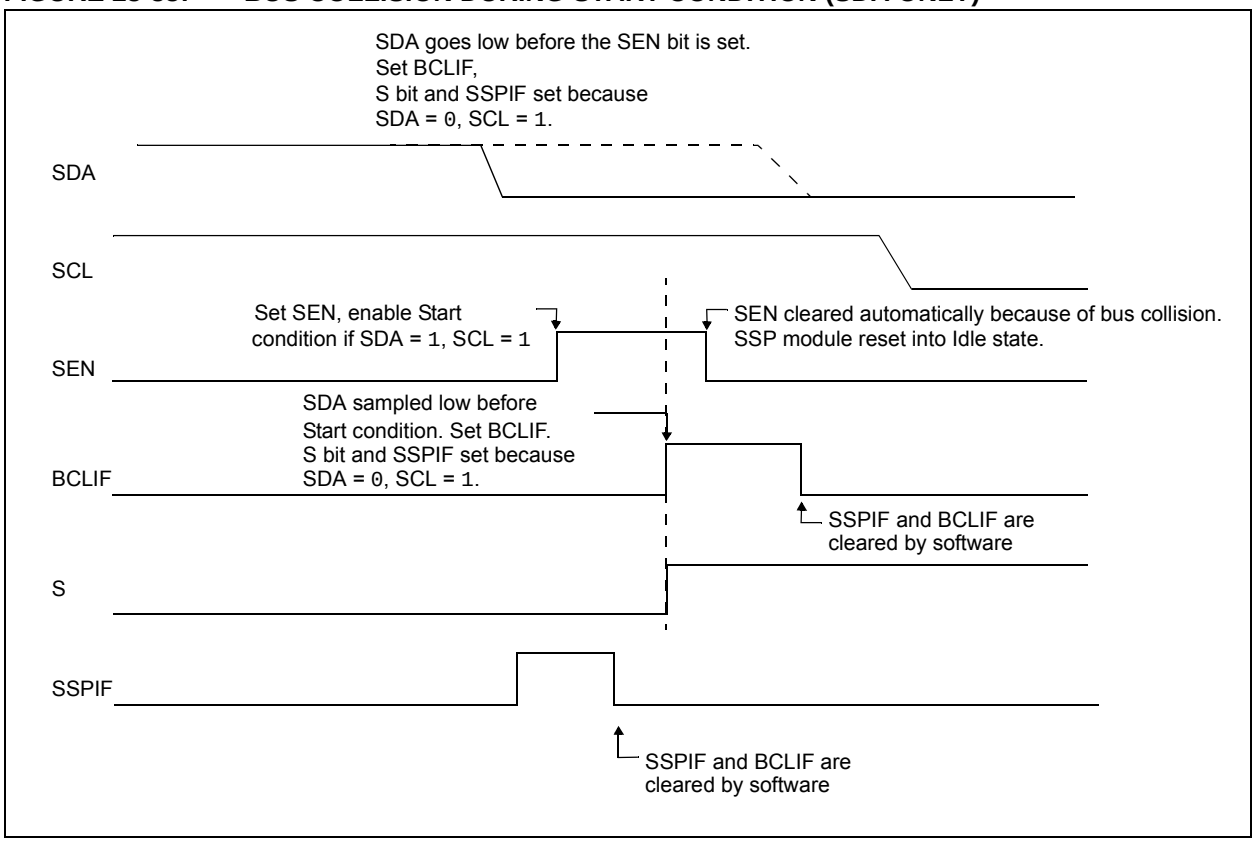
- the Start condition is aborted,
- the BCLIF flag is set and
- the MSSP module is reset to its Idle state (Figure 23-33).

The Start condition begins with the SDA and SCL pins deasserted. When the SDA pin is sampled high, the Baud Rate Generator is loaded and counts down. If the SCL pin is sampled low while SDA is high, a bus collision occurs because it is assumed that another master is attempting to drive a data '1' during the Start condition.

If the SDA pin is sampled low during this count, the BRG is reset and the SDA line is asserted early (Figure 23-35). If, however, a '1' is sampled on the SDA pin, the SDA pin is asserted low at the end of the BRG count. The Baud Rate Generator is then reloaded and counts down to zero; if the SCL pin is sampled as '0' during this time, a bus collision does not occur. At the end of the BRG count, the SCL pin is asserted low.

**Note:** The reason that bus collision is not a factor during a Start condition is that no two bus masters can assert a Start condition at the exact same time. Therefore, one master will always assert SDA before the other. This condition does not cause a bus collision because the two masters must be allowed to arbitrate the first address following the Start condition. If the address is the same, arbitration must be allowed to continue into the data portion, Repeated Start or Stop conditions.

**FIGURE 23-33: BUS COLLISION DURING START CONDITION (SDA ONLY)**





© 2013-2015 Microchip Technology Inc.

**FIGURE 23-34: BUS COLLISION DURING START CONDITION (SCL = 0)**

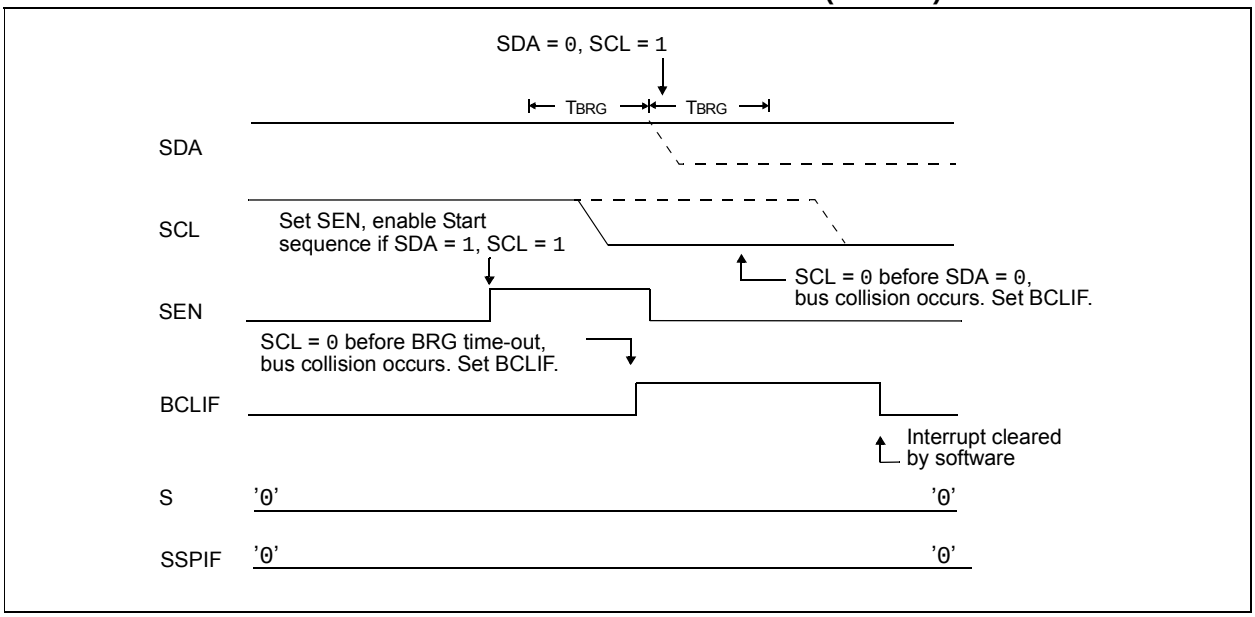
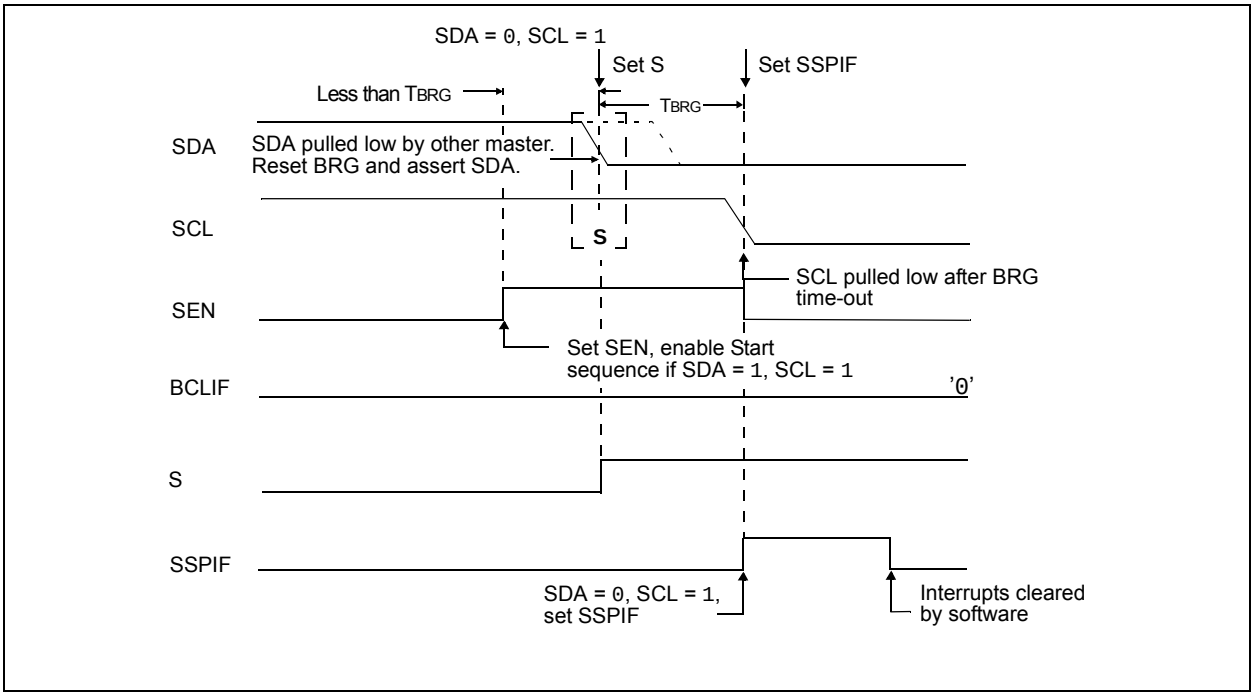



**FIGURE 23-35: BRG RESET DUE TO SDA ARBITRATION DURING START CONDITION**



### 23.6.13.2 Bus Collision During a Repeated Start Condition

During a Repeated Start condition, a bus collision occurs if:

a) A low level is sampled on SDA when SCL goes from low level to high level (Case 1).

b) SCL goes low before SDA is asserted low, indicating that another master is attempting to transmit a data '1' (Case 2).

When the user releases SDA and the pin is allowed to float high, the BRG is loaded with SSPxADD and counts down to zero. The SCL pin is then deasserted and when sampled high, the SDA pin is sampled.

If SDA is low, a bus collision has occurred (i.e., another master is attempting to transmit a data '0', Figure 23-36). If SDA is sampled high, the BRG is reloaded and begins counting. If SDA goes from high-to-low before the BRG times out, no bus collision occurs because no two masters can assert SDA at exactly the same time.

If SCL goes from high-to-low before the BRG times out and SDA has not already been asserted, a bus collision occurs. In this case, another master is attempting to transmit a data '1' during the Repeated Start condition, see Figure 23-37.

If, at the end of the BRG time-out, both SCL and SDA are still high, the SDA pin is driven low and the BRG is reloaded and begins counting. At the end of the count, regardless of the status of the SCL pin, the SCL pin is driven low and the Repeated Start condition is complete.

**FIGURE 23-36: BUS COLLISION DURING A REPEATED START CONDITION (CASE 1)**

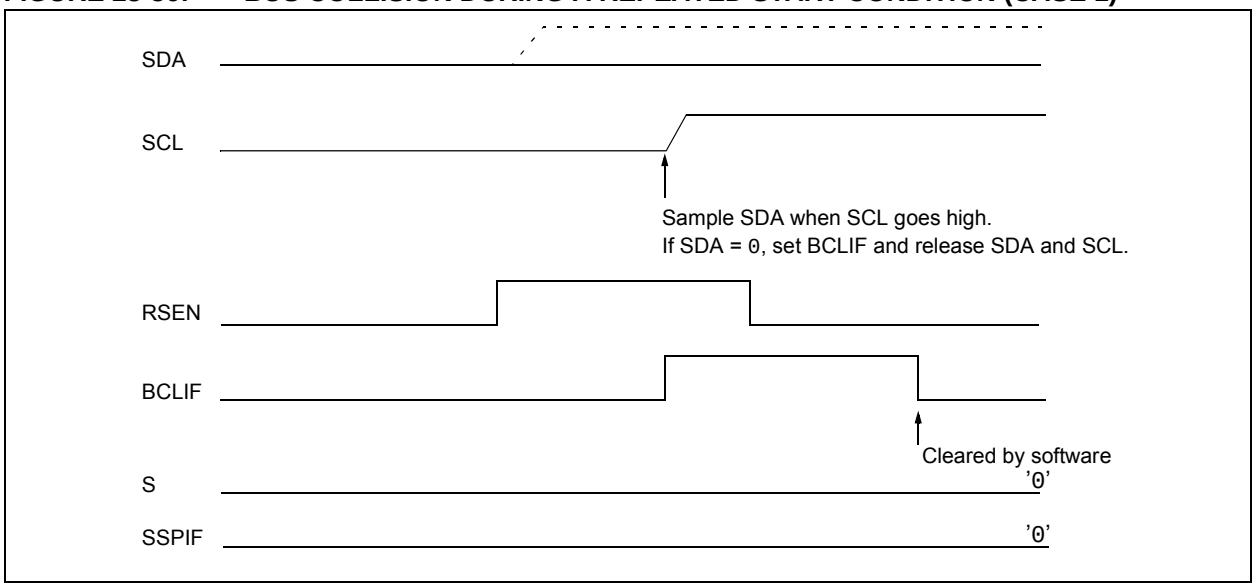
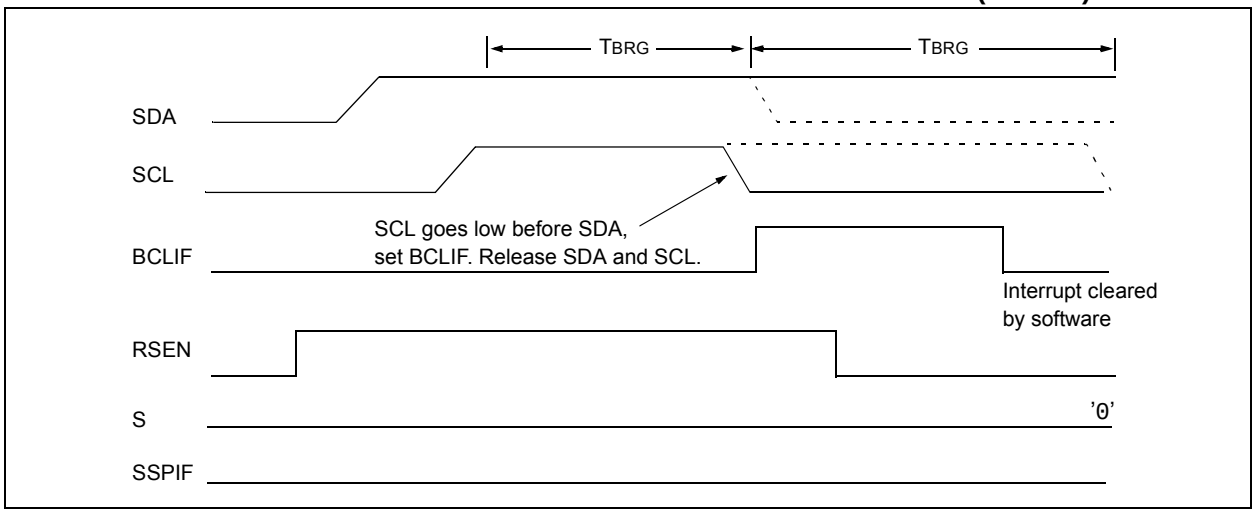



**FIGURE 23-37: BUS COLLISION DURING REPEATED START CONDITION (CASE 2)**





© 2013-2015 Microchip Technology Inc.

### 23.6.13.3 Bus Collision During a Stop Condition

Bus collision occurs during a Stop condition if:

a) After the SDA pin has been deasserted and allowed to float high, SDA is sampled low after the BRG has timed out (Case 1).

b) After the SCL pin is deasserted, SCL is sampled low before SDA goes high (Case 2).

The Stop condition begins with SDA asserted low. When SDA is sampled low, the SCL pin is allowed to float. When the pin is sampled high (clock arbitration), the Baud Rate Generator is loaded with SSPxADD and counts down to zero. After the BRG times out, SDA is sampled. If SDA is sampled low, a bus collision has occurred. This is due to another master attempting to drive a data '0' (Figure 23-38). If the SCL pin is sampled low before SDA is allowed to float high, a bus collision occurs. This is another case of another master attempting to drive a data '0' (Figure 23-39).

**FIGURE 23-38: BUS COLLISION DURING A STOP CONDITION (CASE 1)**

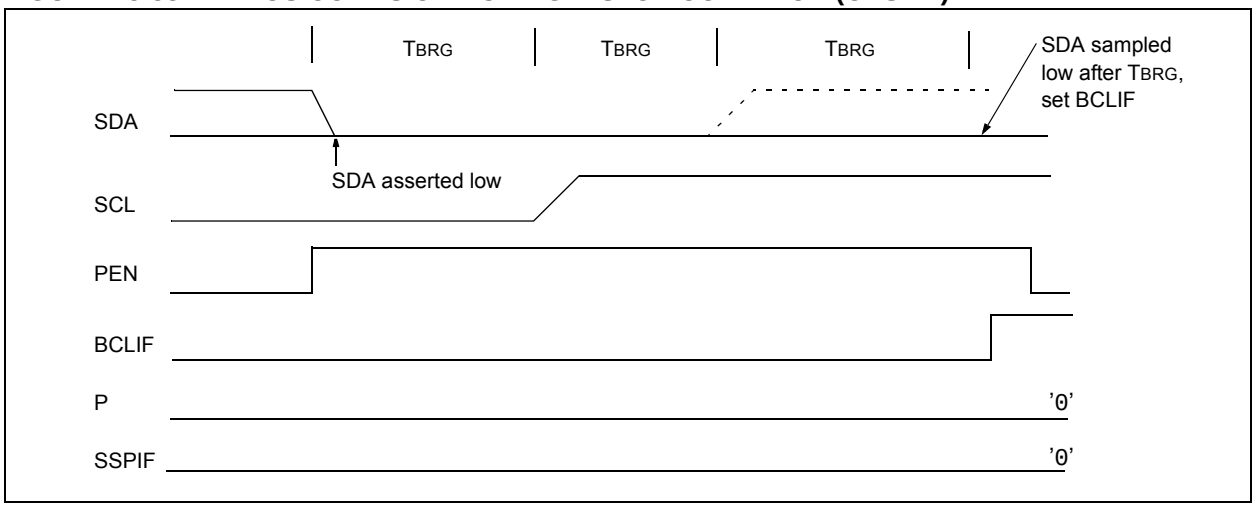



**FIGURE 23-39: BUS COLLISION DURING A STOP CONDITION (CASE 2)**

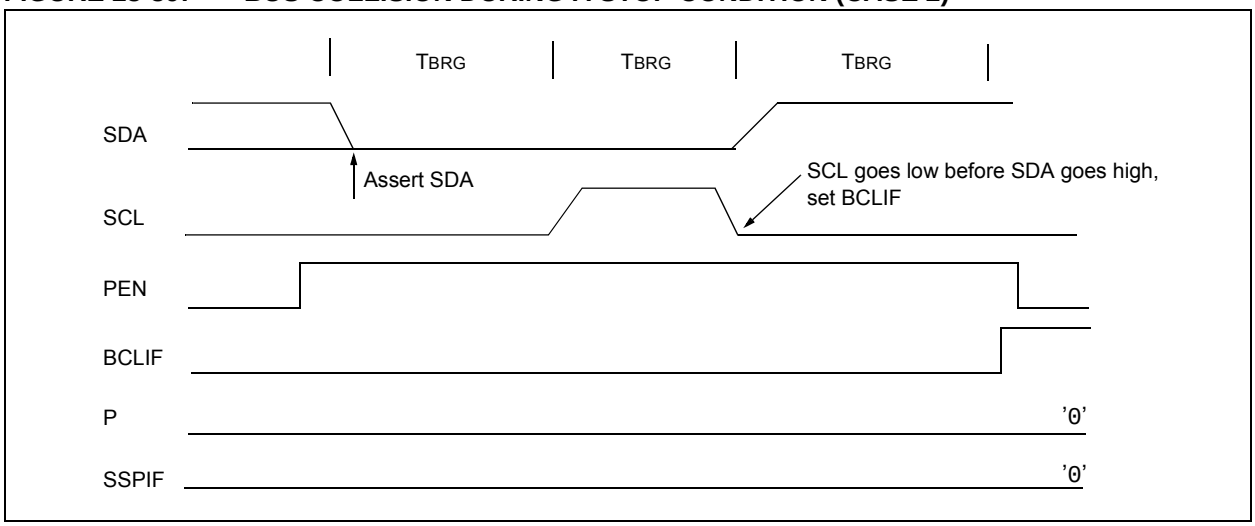


**TABLE 23-3: SUMMARY OF REGISTERS ASSOCIATED WITH $I^2C$ OPERATION**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Reset Values on Page: |
|---|---|---|---|---|---|---|---|---|---|
| ANSELA | — | — | — | ANSA4 | — | ANSA2 | ANSA1 | ANSA0 | 114 |
| ANSELB(1) | — | — | ANSB5 | ANSB4 | — | — | — | — | 120 |
| ANSELC | ANSC7(1) | ANSC6(1) | ANSC5(2) | ANSC4(2) | ANSC3 | ANSC2 | ANSC1 | ANSC0 | 125 |
| INTCON | GIE | PEIE | TMR0IE | INTE | IOCIE | TMR0IF | INTF | IOCIF | 77 |
| PIE1 | TMR1GIE | ADIE | — | — | SSP1IE | CCP1IE | TMR2IE | TMR1IE | 78 |
| PIE2 | — | — | — | — | BCL1IE | — | — | CCP2IE | 79 |
| PIR1 | TMR1GIF | ADIF | — | — | SSP1IF | CCP1IF | TMR2IF | TMR1IF | 81 |
| PIR2 | — | — | — | — | BCL1IF | — | — | CCP2IF | 82 |
| RxyPPS | — | — | — | RxyPPS<4:0> | | | | | 132 |
| SSPCLKPPS | — | — | — | SSPCLKPPS<4:0> | | | | | 130, 131 |
| SSPDATPPS | — | — | — | SSPDATPPS<4:0> | | | | | 130, 131 |
| SSPSSPPS | — | — | — | SSPSSPPS<4:0> | | | | | 130, 131 |
| SSP1ADD | ADD<7:0> | | | | | | | | 248 |
| SSP1BUF | Synchronous Serial Port Receive Buffer/Transmit Register | | | | | | | | 198* |
| SSP1CON1 | WCOL | SSPOV | SSPEN | CKP | SSPM<3:0> | | | | 244 |
| SSP1CON2 | GCEN | ACKSTAT | ACKDT | ACKEN | RCEN | PEN | RSEN | SEN | 245 |
| SSP1CON3 | ACKTIM | PCIE | SCIE | BOEN | SDAHT | SBCDE | AHEN | DHEN | 246 |
| SSP1MSK | MSK<7:0> | | | | | | | | 248 |
| SSP1STAT | SMP | CKE | D/$\overline{A}$ | P | S | R/$\overline{W}$ | UA | BF | 242 |
| TRISA | — | — | TRISA5 | TRISA4 | —(3) | TRISA2 | TRISA1 | TRISA0 | 113 |
| TRISB(1) | TRISB7 | TRISB6 | TRISB5 | TRISB4 | — | — | — | — | 119 |
| TRISC | TRISC7(1) | TRISC6(1) | TRISC5 | TRISC4 | TRISC3 | TRISC2 | TRISC1 | TRISC0 | 124 |

**Legend:** — = unimplemented location, read as '0'. Shaded cells are not used by the MSSP module in $I^2C$ mode.

* Page provides register information.

**Note 1:** PIC16(L)F1707 only.

**2:** PIC16(L)F1703 only.

**3:** Unimplemented, read as '1'.


© 2013-2015 Microchip Technology Inc.

## 23.7 BAUD RATE GENERATOR

The MSSP module has a Baud Rate Generator available for clock generation in both $I^2C$ and SPI Master modes. The Baud Rate Generator (BRG) reload value is placed in the SSPxADD register (Register 23-6). When a write occurs to SSPxBUF, the Baud Rate Generator will automatically begin counting down.

Once the given operation is complete, the internal clock will automatically stop counting and the clock pin will remain in its last state.

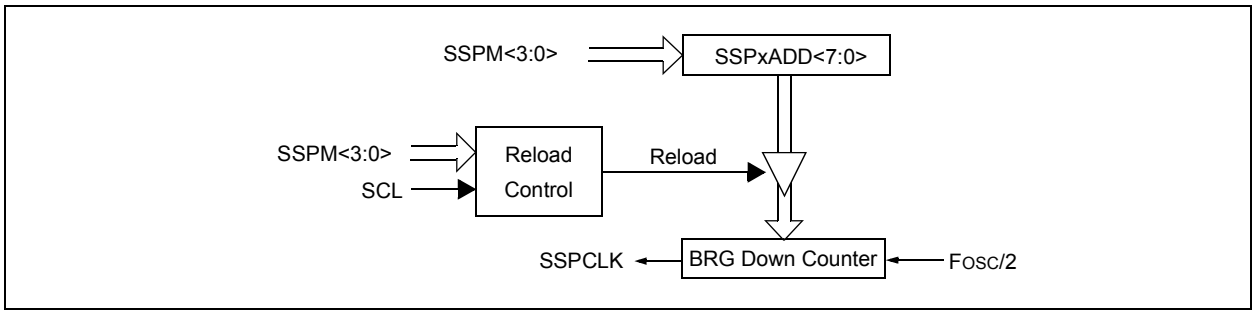
An internal signal “Reload” in Figure 23-40 triggers the value from SSPxADD to be loaded into the BRG counter. This occurs twice for each oscillation of the module clock line. The logic dictating when the reload signal is asserted depends on the mode the MSSP is being operated in.

Table 23-4 demonstrates clock rates based on instruction cycles and the BRG value loaded into SSPxADD.

**EQUATION 23-1:**

$$F_{CLOCK} = \frac{F_{OSC}}{(SSPxADD + 1)(4)}$$

**FIGURE 23-40: BAUD RATE GENERATOR BLOCK DIAGRAM**




**Note:** Values of 0x00, 0x01 and 0x02 are not valid for SSPxADD when used as a Baud Rate Generator for $I^2C$. This is an implementation limitation.

**TABLE 23-4: MSSP CLOCK RATE W/BRG**

| FOSC | FCY | BRG Value | FCLOCK (2 Rollovers of BRG) |
|---|---|---|---|
| 32 MHz | 8 MHz | 13h | 400 kHz |
| 32 MHz | 8 MHz | 19h | 308 kHz |
| 32 MHz | 8 MHz | 4Fh | 100 kHz |
| 16 MHz | 4 MHz | 09h | 400 kHz |
| 16 MHz | 4 MHz | 0Ch | 308 kHz |
| 16 MHz | 4 MHz | 27h | 100 kHz |
| 4 MHz | 1 MHz | 09h | 100 kHz |

**Note:** Refer to the I/O port electrical specifications in Table 26-4 to ensure the system is designed to support IOL requirements.

## 23.8 Register Definitions: MSSP Control

### REGISTER 23-1: SSP1STAT: SSP STATUS REGISTER

| R/W-0/0 | R/W-0/0 | R-0/0 | R-0/0 | R-0/0 | R-0/0 | R-0/0 | R-0/0 |
|---|---|---|---|---|---|---|---|
| SMP | CKE | D/$\overline{A}$ | P | S | R/$\overline{W}$ | UA | BF |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| '1' = Bit is set | '0' = Bit is cleared | |

bit 7 **SMP:** SPI Data Input Sample bit

SPI Master mode:
1 = Input data sampled at end of data output time
0 = Input data sampled at middle of data output time

SPI Slave mode:
SMP must be cleared when SPI is used in Slave mode

In $I^2C$ Master or Slave mode:
1 = Slew rate control disabled for Standard Speed mode (100 kHz and 1 MHz)
0 = Slew rate control enabled for High-Speed mode (400 kHz)

bit 6 **CKE:** SPI Clock Edge Select bit (SPI mode only)

In SPI Master or Slave mode:
1 = Transmit occurs on transition from active to Idle clock state
0 = Transmit occurs on transition from Idle to active clock state

In $I^2C$™ mode only:
1 = Enable input logic so that thresholds are compliant with SMBus specification
0 = Disable SMBus specific inputs

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 MSSP module is disabled, 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 MSSP module is disabled, 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 not ACK bit.

In $I^2C$ Slave mode:
1 = Read
0 = Write

In $I^2C$ Master mode:
1 = Transmit is in progress
0 = Transmit is not in progress
OR-ing this bit with SEN, RSEN, PEN, RCEN or ACKEN will indicate if the MSSP is in Idle mode.

bit 1 **UA:** Update Address bit (10-bit $I^2C$ mode only)
1 = Indicates that the user needs to update the address in the SSPxADD register
0 = Address does not need to be updated


© 2013-2015 Microchip Technology Inc.

**REGISTER 23-1: SSP1STAT: SSP STATUS REGISTER (CONTINUED)**

bit 0 **BF:** Buffer Full Status bit

Receive (SPI and $I^2C$ modes):
1 = Receive complete, SSPxBUF is full
0 = Receive not complete, SSPxBUF is empty

Transmit ($I^2C$ mode only):
1 = Data transmit in progress (does not include the $\overline{ACK}$ and Stop bits), SSPxBUF is full
0 = Data transmit complete (does not include the $\overline{ACK}$ and Stop bits), SSPxBUF is empty

## REGISTER 23-2: SSP1CON1: SSP CONTROL REGISTER 1

| R/C/HS-0/0 | R/C/HS-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| WCOL | SSPOV(1) | SSPEN | CKP | SSPM<3:0> | | | |
| bit 7 | | | | | | | bit 0 |

| Legend: | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' | |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets | |
| '1' = Bit is set | '0' = Bit is cleared | HS = Bit is set by hardware | C = User cleared |

bit 7 **WCOL:** Write Collision Detect bit
Master mode:
1 = A write to the SSPxBUF register was attempted while the I²C conditions were not valid for a transmission to be started
0 = No collision
Slave mode:
1 = The SSPxBUF 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(1)
In SPI mode:
1 = A new byte is received while the SSPxBUF register is still holding the previous data. In case of overflow, the data in SSPSR is lost. Overflow can only occur in Slave mode. In Slave mode, the user must read the SSPxBUF, 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 SSPxBUF register (must be cleared in software).
0 = No overflow
In I²C mode:
1 = A byte is received while the SSPxBUF register is still holding the previous byte. SSPOV is a "don't care" in Transmit mode (must be cleared in software).
0 = No overflow

bit 5 **SSPEN:** Synchronous Serial Port Enable bit
In both modes, when enabled, these pins must be properly configured as input or output
In SPI mode:
1 = Enables serial port and configures SCK, SDO, SDI and $\overline{SS}$ as the source of the serial port pins(2)
0 = Disables serial port and configures these pins as I/O port pins
In I²C mode:
1 = Enables the serial port and configures the SDA and SCL pins as the source of the serial port pins(3)
0 = Disables serial port and configures these pins as I/O port pins

bit 4 **CKP:** Clock Polarity Select bit
In SPI mode:
1 = Idle state for clock is a high level
0 = Idle state for clock is a low level
In I²C Slave mode:
SCL release control
1 = Enable clock
0 = Holds clock low (clock stretch). (Used to ensure data setup time.)
In I²C Master mode:
Unused in this mode

bit 3-0 **SSPM<3:0>:** Synchronous Serial Port Mode Select bits
1111 = I²C Slave mode, 10-bit address with Start and Stop bit interrupts enabled
1110 = I²C Slave mode, 7-bit address with Start and Stop bit interrupts enabled
1101 = Reserved
1100 = Reserved
1011 = I²C firmware controlled Master mode (slave idle)
1010 = SPI Master mode, clock = FOSC/(4 * (SSPxADD+1))(5)
1001 = Reserved
1000 = I²C Master mode, clock = FOSC / (4 * (SSPxADD+1))(4)
0111 = I²C Slave mode, 10-bit address
0110 = I²C Slave mode, 7-bit address
0101 = SPI Slave mode, clock = SCK pin, $\overline{SS}$ pin control disabled, $\overline{SS}$ can be used as I/O pin
0100 = SPI Slave mode, clock = SCK pin, $\overline{SS}$ pin control enabled
0011 = SPI Master mode, clock = T2_match/2
0010 = SPI Master mode, clock = FOSC/64
0001 = SPI Master mode, clock = FOSC/16
0000 = SPI Master mode, clock = FOSC/4

**Note 1:** In Master mode, the overflow bit is not set since each new reception (and transmission) is initiated by writing to the SSPxBUF register.
**2:** When enabled, these pins must be properly configured as input or output. Use SSPSSPPS, SSPCLKPPS, SSPDATPPS, and RxyPPS to select the pins.
**3:** When enabled, the SDA and SCL pins must be configured as inputs. Use SSPCLKPPS, SSPDATPPS, and RxyPPS to select the pins.
**4:** SSPxADD values of 0, 1 or 2 are not supported for I²C mode.
**5:** SSPxADD value of '0' is not supported. Use SSPM = 0000 instead.


© 2013-2015 Microchip Technology Inc.

**REGISTER 23-3: SSP1CON2: SSP CONTROL REGISTER 2(1)**

| R/W-0/0 | R-0/0 | R/W-0/0 | R/S/HS-0/0 | R/S/HS-0/0 | R/S/HS-0/0 | R/S/HS-0/0 | R/W/HS-0/0 |
|---|---|---|---|---|---|---|---|
| GCEN | ACKSTAT | ACKDT | ACKEN | RCEN | PEN | RSEN | SEN |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| '1' = Bit is set | '0' = Bit is cleared | HC = Cleared by hardware S = User set |

bit 7 **GCEN:** General Call Enable bit (in $I^2C$ Slave mode only)
1 = Enable interrupt when a general call address (0x00 or 00h) is received in the SSPSR
0 = General call address disabled

bit 6 **ACKSTAT:** Acknowledge Status bit (in $I^2C$ mode only)
1 = Acknowledge was not received
0 = Acknowledge was received

bit 5 **ACKDT:** Acknowledge Data bit (in $I^2C$ mode only)
In Receive mode:
Value transmitted when the user initiates an Acknowledge sequence at the end of a receive
1 = Not Acknowledge
0 = Acknowledge

bit 4 **ACKEN:** Acknowledge Sequence Enable bit (in $I^2C$ Master mode only)
In Master Receive mode:
1 = Initiate Acknowledge sequence on SDA and SCL pins, and transmit ACKDT data bit. Automatically cleared by hardware.
0 = Acknowledge sequence idle

bit 3 **RCEN:** Receive Enable bit (in $I^2C$ Master mode only)
1 = Enables Receive mode for $I^2C$
0 = Receive idle

bit 2 **PEN:** Stop Condition Enable bit (in $I^2C$ Master mode only)
SCKMSSP Release Control:
1 = Initiate Stop condition on SDA and SCL pins. Automatically cleared by hardware.
0 = Stop condition Idle

bit 1 **RSEN:** Repeated Start Condition Enable bit (in $I^2C$ Master mode only)
1 = Initiate Repeated Start condition on SDA and SCL pins. Automatically cleared by hardware.
0 = Repeated Start condition Idle

bit 0 **SEN:** Start Condition Enable/Stretch Enable bit
In Master mode:
1 = Initiate Start condition on SDA and SCL pins. Automatically cleared by hardware.
0 = Start condition Idle
In Slave mode:
1 = Clock stretching is enabled for both slave transmit and slave receive (stretch enabled)
0 = Clock stretching is disabled

**Note 1:** For bits ACKEN, RCEN, PEN, RSEN, SEN: If the $I^2C$ module is not in the Idle mode, this bit may not be set (no spooling) and the SSPxBUF may not be written (or writes to the SSPxBUF are disabled).

**REGISTER 23-4: SSP1CON3: SSP CONTROL REGISTER 3**

| R-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| ACKTIM(3) | PCIE | SCIE | BOEN | SDAHT | SBCDE | AHEN | DHEN |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| '1' = Bit is set | '0' = Bit is cleared | |

bit 7 **ACKTIM:** Acknowledge Time Status bit ($I^2C$ mode only)(3)
1 = Indicates the $I^2C$ bus is in an Acknowledge sequence, set on eighth falling edge of SCL clock
0 = Not an Acknowledge sequence, cleared on ninth rising edge of SCL clock

bit 6 **PCIE**: Stop Condition Interrupt Enable bit ($I^2C$ mode only)
1 = Enable interrupt on detection of Stop condition
0 = Stop detection interrupts are disabled(2)

bit 5 **SCIE**: Start Condition Interrupt Enable bit ($I^2C$ mode only)
1 = Enable interrupt on detection of Start or Restart conditions
0 = Start detection interrupts are disabled(2)

bit 4 **BOEN:** Buffer Overwrite Enable bit
In SPI Slave mode:(1)
1 = SSPxBUF updates every time that a new data byte is shifted in ignoring the BF bit
0 = If new byte is received with BF bit of the SSPxSTAT register already set, SSPOV bit of the SSPxCON1 register is set, and the buffer is not updated
In $I^2C$ Master mode and SPI Master mode:
This bit is ignored.
In $I^2C$ Slave mode:
1 = SSPxBUF is updated and $\overline{ACK}$ is generated for a received address/data byte, ignoring the state of the SSPOV bit only if the BF bit = 0.
0 = SSPxBUF is only updated when SSPOV is clear

bit 3 **SDAHT:** SDA Hold Time Selection bit ($I^2C$ mode only)
1 = Minimum of 300 ns hold time on SDA after the falling edge of SCL
0 = Minimum of 100 ns hold time on SDA after the falling edge of SCL

bit 2 **SBCDE:** Slave Mode Bus Collision Detect Enable bit ($I^2C$ Slave mode only)
If, on the rising edge of SCL, SDA is sampled low when the module is outputting a high state, the BCL1IF bit of the PIR2 register is set, and bus goes idle
1 = Enable slave bus collision interrupts
0 = Slave bus collision interrupts are disabled

bit 1 **AHEN:** Address Hold Enable bit ($I^2C$ Slave mode only)
1 = Following the eighth falling edge of SCL for a matching received address byte; CKP bit of the SSPxCON1 register will be cleared and the SCL will be held low.
0 = Address holding is disabled

bit 0 **DHEN:** Data Hold Enable bit ($I^2C$ Slave mode only)
1 = Following the eighth falling edge of SCL for a received data byte; slave hardware clears the CKP bit of the SSPxCON1 register and SCL is held low.
0 = Data holding is disabled


© 2013-2015 Microchip Technology Inc.

**REGISTER 23-4: SSP1CON3: SSP CONTROL REGISTER 3 (CONTINUED)**

**Note 1:** For daisy-chained SPI operation; allows the user to ignore all but the last received byte. SSPOV is still set when a new byte is received and BF = 1, but hardware continues to write the most recent byte to SSPxBUF.

**2:** This bit has no effect in Slave modes that Start and Stop condition detection is explicitly listed as enabled.

**3:** The ACKTIM Status bit is only active when the AHEN bit or DHEN bit is set.

**REGISTER 23-5: SSP1MSK: SSP MASK REGISTER**

| R/W-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 |
|---|---|---|---|---|---|---|---|
| MSK<7:0> | | | | | | | |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| '1' = Bit is set | '0' = Bit is cleared | |

bit 7-1 **MSK<7:1>:** Mask bits
1 = The received address bit n is compared to SSPxADD<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
$I^2C$ Slave mode, 10-bit address (SSPM<3:0> = 0111 or 1111):
1 = The received address bit 0 is compared to SSPxADD<0> to detect $I^2C$ address match
0 = The received address bit 0 is not used to detect $I^2C$ address match
$I^2C$ Slave mode, 7-bit address, the bit is ignored

**REGISTER 23-6: SSPxADD: MSSP ADDRESS AND BAUD RATE REGISTER ($I^2C$ MODE)**

| R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| ADD<7:0> | | | | | | | |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| '1' = Bit is set | '0' = Bit is cleared | |

**Master mode:**

bit 7-0 **ADD<7:0>:** Baud Rate Clock Divider bits
SCL pin clock period = ((ADD<7:0> + 1) *4)/FOSC

**10-Bit Slave mode – Most Significant Address Byte:**

bit 7-3 **Not used:** Unused for Most Significant Address Byte. Bit state of this register is a "don't care". Bit pattern sent by master is fixed by $I^2C$ specification and must be equal to '11110'. However, those bits are compared by hardware and are not affected by the value in this register.

bit 2-1 **ADD<2:1>:** Two Most Significant bits of 10-bit address

bit 0 **Not used:** Unused in this mode. Bit state is a "don't care".

**10-Bit Slave mode – Least Significant Address Byte:**

bit 7-0 **ADD<7:0>:** Eight Least Significant bits of 10-bit address

**7-Bit Slave mode:**

bit 7-1 **ADD<7:1>:** 7-bit address

bit 0 **Not used:** Unused in this mode. Bit state is a "don't care".


© 2013-2015 Microchip Technology Inc.

## 24.0 IN-CIRCUIT SERIAL PROGRAMMING™ (ICSP™)

ICSP™ programming allows customers to manufacture circuit boards with unprogrammed devices. Programming can be done after the assembly process, allowing the device to be programmed with the most recent firmware or a custom firmware. Five pins are needed for ICSP™ programming:

- ICSPCLK
- ICSPDAT
- $\overline{MCLR}/V_{PP}$
- $V_{DD}$
- $V_{SS}$

In Program/Verify mode the program memory, user IDs and the Configuration Words are programmed through serial communications. The ICSPDAT pin is a bidirectional I/O used for transferring the serial data and the ICSPCLK pin is the clock input. For more information on ICSP™ refer to the **"***PIC16(L)F170X Memory Programming Specification***"** (DS41683).

### 24.1 High-Voltage Programming Entry Mode

The device is placed into High-Voltage Programming Entry mode by holding the ICSPCLK and ICSPDAT pins low then raising the voltage on $\overline{MCLR}/V_{PP}$ to $V_{IHH}$.

### 24.2 Low-Voltage Programming Entry Mode

The Low-Voltage Programming Entry mode allows the PIC® Flash MCUs to be programmed using $V_{DD}$ only, without high voltage. When the LVP bit of Configuration Words is set to '1', the low-voltage ICSP programming entry is enabled. To disable the Low-Voltage ICSP mode, the LVP bit must be programmed to '0'.

Entry into the Low-Voltage Programming Entry mode requires the following steps:

1. $\overline{MCLR}$ is brought to $V_{IL}$.
2. A 32-bit key sequence is presented on ICSPDAT, while clocking ICSPCLK.

Once the key sequence is complete, $\overline{MCLR}$ must be held at $V_{IL}$ for as long as Program/Verify mode is to be maintained.

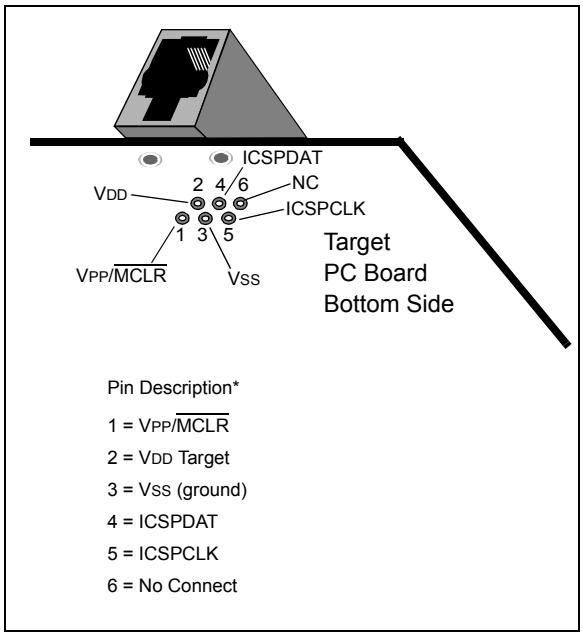
If low-voltage programming is enabled (LVP = 1), the $\overline{MCLR}$ Reset function is automatically enabled and cannot be disabled. See **Section 5.5 "MCLR"** for more information.

The LVP bit can only be reprogrammed to '0' by using the High-Voltage Programming mode.

### 24.3 Common Programming Interfaces

Connection to a target device is typically done through an ICSP™ header. A commonly found connector on development tools is the RJ-11 in the 6P6C (6-pin, 6-connector) configuration. See Figure 24-1.

**FIGURE 24-1: ICD RJ-11 STYLE CONNECTOR INTERFACE**

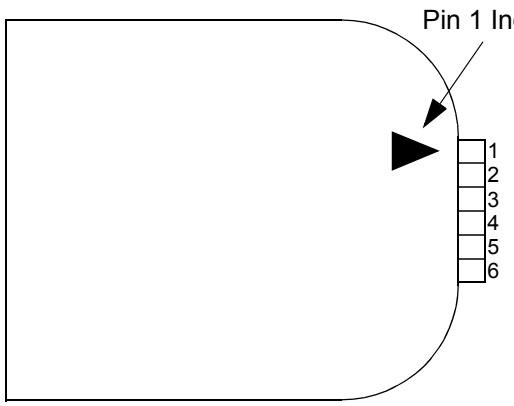



Another connector often found in use with the PICkit™ programmers is a standard 6-pin header with 0.1 inch spacing. Refer to Figure 24-2.

For additional interface recommendations, refer to your specific device programmer manual prior to PCB design.

It is recommended that isolation devices be used to separate the programming pins from other circuitry. The type of isolation is highly dependent on the specific application and may include devices such as resistors, diodes, or even jumpers. See Figure 24-3 for more information.

**FIGURE 24-2: PICkit™ PROGRAMMER STYLE CONNECTOR INTERFACE**




Pin Description*

1 = VPP/$\overline{\text{MCLR}}$

2 = VDD Target

3 = VSS (ground)

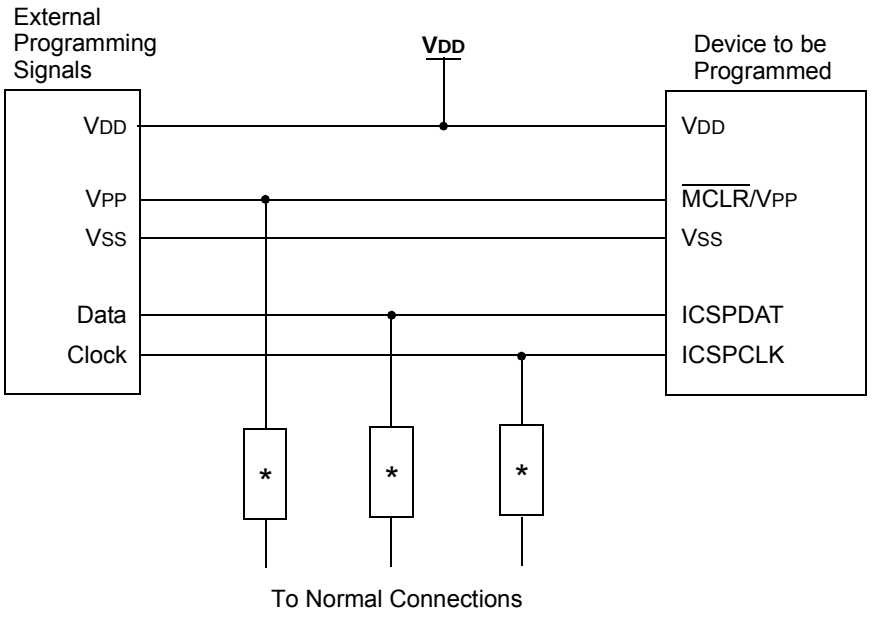
4 = ICSPDAT

5 = ICSPCLK

6 = No Connect

* The 6-pin header (0.100" spacing) accepts 0.025" square pins.

**FIGURE 24-3: TYPICAL CONNECTION FOR ICSP™ PROGRAMMING**




* Isolation devices (as required).


© 2013-2015 Microchip Technology Inc.

## 25.0 INSTRUCTION SET SUMMARY

Each instruction is a 14-bit word containing the operation code (opcode) and all required operands. The opcodes are broken into three broad categories.

- Byte Oriented
- Bit Oriented
- Literal and Control

The literal and control category contains the most varied instruction word format.

Table 25-3 lists the instructions recognized by the MPASM™ assembler.

All instructions are executed within a single instruction cycle, with the following exceptions, which may take two or three cycles:

- Subroutine takes two cycles (CALL, CALLW)
- Returns from interrupts or subroutines take two cycles (RETURN, RETLW, RETFIE)
- Program branching takes two cycles (GOTO, BRA, BRW, BTFSS, BTFSC, DECFSZ, INCSFZ)
- One additional instruction cycle will be used when any instruction references an indirect file register and the file select register is pointing to program memory.

One instruction cycle consists of 4 oscillator cycles; for an oscillator frequency of 4 MHz, this gives a nominal instruction execution rate of 1 MHz.

All instruction examples use the format '0xhh' to represent a hexadecimal number, where 'h' signifies a hexadecimal digit.

### 25.1 Read-Modify-Write Operations

Any instruction that specifies a file register as part of the instruction performs a Read-Modify-Write (R-M-W) 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.

**TABLE 25-1: OPCODE FIELD DESCRIPTIONS**

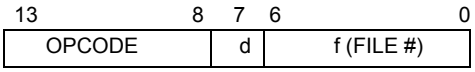
| Field | Description |
|---|---|
| f | Register file address (0x00 to 0x7F) |
| W | Working register (accumulator) |
| b | Bit address within an 8-bit file register |
| k | Literal field, constant data or label |
| x | Don't care location (= 0 or 1).<br>The assembler will generate code with x = 0.<br>It 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. |
| n | FSR or INDF number. (0-1) |
| mm | Pre-post increment-decrement mode selection |

**TABLE 25-2: ABBREVIATION DESCRIPTIONS**

| Field | Description |
|---|---|
| PC | Program Counter |
| $\overline{TO}$ | Time-Out bit |
| C | Carry bit |
| DC | Digit Carry bit |
| Z | Zero bit |
| $\overline{PD}$ | Power-Down bit |

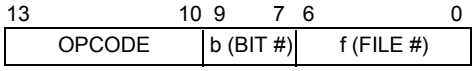
**FIGURE 25-1: GENERAL FORMAT FOR INSTRUCTIONS**

**Byte-oriented file register operations**




d = 0 for destination W
d = 1 for destination f
f = 7-bit file register address

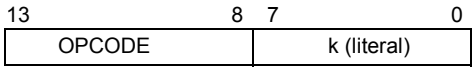
**Bit-oriented file register operations**




b = 3-bit bit address
f = 7-bit file register address

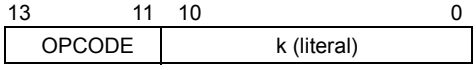
**Literal and control operations**

General



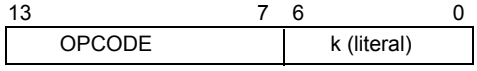

k = 8-bit immediate value

CALL and GOTO instructions only



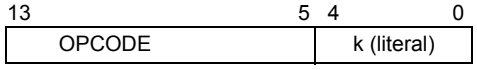

k = 11-bit immediate value

MOVLP instruction only



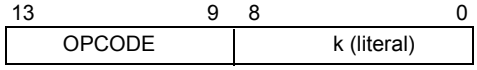

k = 7-bit immediate value

MOVLB instruction only



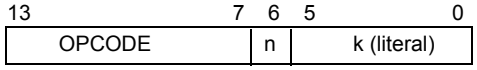

k = 5-bit immediate value

BRA instruction only



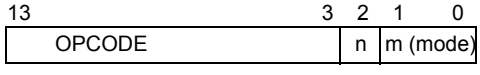

k = 9-bit immediate value

FSR Offset instructions



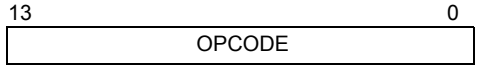

n = appropriate FSR
k = 6-bit immediate value

FSR Increment instructions




n = appropriate FSR
m = 2-bit mode value

OPCODE only





© 2013-2015 Microchip Technology Inc.

**TABLE 25-3: PIC16(L)F1703/7 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 | 2 |
| ADDWFC | f, d | Add with Carry W and f | 1 | 11 | 1101 | dfff | ffff | C, DC, Z | 2 |
| ANDWF | f, d | AND W with f | 1 | 00 | 0101 | dfff | ffff | Z | 2 |
| ASRF | f, d | Arithmetic Right Shift | 1 | 11 | 0111 | dfff | ffff | C, Z | 2 |
| LSLF | f, d | Logical Left Shift | 1 | 11 | 0101 | dfff | ffff | C, Z | 2 |
| LSRF | f, d | Logical Right Shift | 1 | 11 | 0110 | dfff | ffff | C, Z | 2 |
| CLRF | f | Clear f | 1 | 00 | 0001 | lfff | ffff | Z | 2 |
| CLRW | – | Clear W | 1 | 00 | 0001 | 0000 | 00xx | Z | |
| COMF | f, d | Complement f | 1 | 00 | 1001 | dfff | ffff | Z | 2 |
| DECF | f, d | Decrement f | 1 | 00 | 0011 | dfff | ffff | Z | 2 |
| INCF | f, d | Increment f | 1 | 00 | 1010 | dfff | ffff | Z | 2 |
| IORWF | f, d | Inclusive OR W with f | 1 | 00 | 0100 | dfff | ffff | Z | 2 |
| MOVF | f, d | Move f | 1 | 00 | 1000 | dfff | ffff | Z | 2 |
| MOVWF | f | Move W to f | 1 | 00 | 0000 | 1fff | ffff | | 2 |
| RLF | f, d | Rotate Left f through Carry | 1 | 00 | 1101 | dfff | ffff | C | 2 |
| RRF | f, d | Rotate Right f through Carry | 1 | 00 | 1100 | dfff | ffff | C | 2 |
| SUBWF | f, d | Subtract W from f | 1 | 00 | 0010 | dfff | ffff | C, DC, Z | 2 |
| SUBWFB | f, d | Subtract with Borrow W from f | 1 | 11 | 1011 | dfff | ffff | C, DC, Z | 2 |
| SWAPF | f, d | Swap nibbles in f | 1 | 00 | 1110 | dfff | ffff | | 2 |
| XORWF | f, d | Exclusive OR W with f | 1 | 00 | 0110 | dfff | ffff | Z | 2 |
| **BYTE ORIENTED SKIP OPERATIONS** | | | | | | | | | |
| DECFSZ | f, d | Decrement f, Skip if 0 | 1(2) | 00 | 1011 | dfff | ffff | | 1, 2 |
| INCFSZ | f, d | Increment f, Skip if 0 | 1(2) | 00 | 1111 | dfff | ffff | | 1, 2 |
| **BIT-ORIENTED FILE REGISTER OPERATIONS** | | | | | | | | | |
| BCF | f, b | Bit Clear f | 1 | 01 | 00bb | bfff | ffff | | 2 |
| BSF | f, b | Bit Set f | 1 | 01 | 01bb | bfff | ffff | | 2 |
| **BIT-ORIENTED SKIP OPERATIONS** | | | | | | | | | |
| BTFSC | f, b | Bit Test f, Skip if Clear | 1 (2) | 01 | 10bb | bfff | ffff | | 1, 2 |
| BTFSS | f, b | Bit Test f, Skip if Set | 1 (2) | 01 | 11bb | bfff | ffff | | 1, 2 |
| **LITERAL OPERATIONS** | | | | | | | | | |
| ADDLW | k | Add literal and W | 1 | 11 | 1110 | kkkk | kkkk | C, DC, Z | |
| ANDLW | k | AND literal with W | 1 | 11 | 1001 | kkkk | kkkk | Z | |
| IORLW | k | Inclusive OR literal with W | 1 | 11 | 1000 | kkkk | kkkk | Z | |
| MOVLB | k | Move literal to BSR | 1 | 00 | 0000 | 001k | kkkk | | |
| MOVLP | k | Move literal to PCLATH | 1 | 11 | 0001 | 1kkk | kkkk | | |
| MOVLW | k | Move literal to W | 1 | 11 | 0000 | kkkk | kkkk | | |
| SUBLW | k | Subtract W from literal | 1 | 11 | 1100 | kkkk | kkkk | C, DC, Z | |
| XORLW | k | Exclusive OR literal with W | 1 | 11 | 1010 | kkkk | kkkk | Z | |

**Note 1:** 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.

**2:** If this instruction addresses an INDF register and the MSb of the corresponding FSR is set, this instruction will require one additional instruction cycle.

**TABLE 25-3: PIC16(L)F1703/7 INSTRUCTION SET (CONTINUED)**

| Mnemonic, Operands | | Description | Cycles | 14-Bit Opcode MSb | LSb | Status Affected | Notes |
|---|---|---|---|---|---|---|---|
| **CONTROL OPERATIONS** | | | | | | | |
| BRA | k | Relative Branch | 2 | 11 001k kkkk kkkk | | | |
| BRW | – | Relative Branch with W | 2 | 00 0000 0000 1011 | | | |
| CALL | k | Call Subroutine | 2 | 10 0kkk kkkk kkkk | | | |
| CALLW | – | Call Subroutine with W | 2 | 00 0000 0000 1010 | | | |
| GOTO | k | Go to address | 2 | 10 1kkk kkkk kkkk | | | |
| RETFIE | k | Return from interrupt | 2 | 00 0000 0000 1001 | | | |
| RETLW | k | Return with literal in W | 2 | 11 0100 kkkk kkkk | | | |
| RETURN | – | Return from Subroutine | 2 | 00 0000 0000 1000 | | | |
| **INHERENT OPERATIONS** | | | | | | | |
| CLRWDT | – | Clear Watchdog Timer | 1 | 00 0000 0110 0100 | | $\overline{TO}$, $\overline{PD}$ | |
| NOP | – | No Operation | 1 | 00 0000 0000 0000 | | | |
| OPTION | – | Load OPTION_REG register with W | 1 | 00 0000 0110 0010 | | | |
| RESET | – | Software device Reset | 1 | 00 0000 0000 0001 | | | |
| SLEEP | – | Go into Standby mode | 1 | 00 0000 0110 0011 | | $\overline{TO}$, $\overline{PD}$ | |
| TRIS | f | Load TRIS register with W | 1 | 00 0000 0110 0fff | | | |
| **C-COMPILER OPTIMIZED** | | | | | | | |
| ADDFSR | n, k | Add Literal k to FSRn | 1 | 11 0001 0nkk kkkk | | | |
| MOVIW | n mm | Move Indirect FSRn to W with pre/post inc/dec modifier, mm | 1 | 00 0000 0001 0nmm | | Z | **2, 3** |
| | k[n] | Move INDFn to W, Indexed Indirect. | 1 | 11 1111 0nkk kkkk | | Z | **2** |
| MOVWI | n mm | Move W to Indirect FSRn with pre/post inc/dec modifier, mm | 1 | 00 0000 0001 1nmm | | | **2, 3** |
| | k[n] | Move W to INDFn, Indexed Indirect. | 1 | 11 1111 1nkk kkkk | | | **2** |

**Note 1:** 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.

**2:** If this instruction addresses an INDF register and the MSb of the corresponding FSR is set, this instruction will require one additional instruction cycle.

**3:** See Table in the MOVIW and MOVWI instruction descriptions.


© 2013-2015 Microchip Technology Inc.

## 25.2 Instruction Descriptions

**ADDFSR — Add Literal to FSRn**

| | |
|---|---|
| Syntax: | [ *label* ] ADDFSR FSRn, k |
| Operands: | -32 ≤ k ≤ 31<br>n ∈ [ 0, 1] |
| Operation: | FSR(n) + k → FSR(n) |
| Status Affected: | None |
| Description: | The signed 6-bit literal 'k' is added to the contents of the FSRnH:FSRnL register pair.<br><br>FSRn is limited to the range 0000h-FFFFh. Moving beyond these bounds will cause the FSR to wrap-around. |

**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 8-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'. |

**ADDWFC — ADD W and CARRY bit to f**

| | |
|---|---|
| Syntax: | [ *label* ] ADDWFC f {,d} |
| Operands: | 0 ≤ f ≤ 127<br>d ∈ [0,1] |
| Operation: | (W) + (f) + (C) → dest |
| Status Affected: | C, DC, Z |
| Description: | Add W, the Carry flag and data memory location 'f'. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed in data memory location 'f'. |

**ANDLW — AND literal with W**

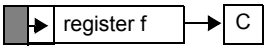
| | |
|---|---|
| Syntax: | [ *label* ] ANDLW k |
| Operands: | 0 ≤ k ≤ 255 |
| Operation: | (W) .AND. (k) → (W) |
| Status Affected: | Z |
| Description: | The contents of W register are AND'ed with the 8-bit literal 'k'. The result is placed in the W register. |

**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'. |

**ASRF — Arithmetic Right Shift**

| | |
|---|---|
| Syntax: | [ *label* ] ASRF f {,d} |
| Operands: | 0 ≤ f ≤ 127<br>d ∈ [0,1] |
| Operation: | (f<7>)→ dest<7><br>(f<7:1>) → dest<6:0>,<br>(f<0>) → C, |
| Status Affected: | C, Z |
| Description: | The contents of register 'f' are shifted one bit to the right through the Carry flag. The MSb remains unchanged. If 'd' is '0', the result is placed in W. 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. |

**BRA** **Relative Branch**

| | |
|---|---|
| Syntax: | [ *label* ] BRA label<br>[ *label* ] BRA $+k |
| Operands: | -256 ≤ label - PC + 1 ≤ 255<br>-256 ≤ k ≤ 255 |
| Operation: | (PC) + 1 + k → PC |
| Status Affected: | None |
| Description: | Add the signed 9-bit literal 'k' to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be PC + 1 + k. This instruction is a 2-cycle instruction. This branch has a limited range. |

**BRW** **Relative Branch with W**

| | |
|---|---|
| Syntax: | [ *label* ] BRW |
| Operands: | None |
| Operation: | (PC) + (W) → PC |
| Status Affected: | None |
| Description: | Add the contents of W (unsigned) to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be PC + 1 + (W). This instruction is a 2-cycle instruction. |

**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 2-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 2-cycle instruction. |


© 2013-2015 Microchip Technology Inc.

**CALL** **Call Subroutine**

| | |
|---|---|
| Syntax: | [ *label* ] CALL k |
| Operands: | $0 \le k \le 2047$ |
| Operation: | (PC)+ 1→ TOS,<br>k → PC<10:0>,<br>(PCLATH<6:3>) → PC<14:11> |
| Status Affected: | None |
| Description: | Call Subroutine. First, return address (PC + 1) is pushed onto the stack. The 11-bit immediate address is loaded into PC bits <10:0>. The upper bits of the PC are loaded from PCLATH. CALL is a 2-cycle instruction. |

**CALLW** **Subroutine Call With W**

| | |
|---|---|
| Syntax: | [ *label* ] CALLW |
| Operands: | None |
| Operation: | (PC) +1 → TOS,<br>(W) → PC<7:0>,<br>(PCLATH<6:0>) → PC<14:8> |
| Status Affected: | None |
| Description: | Subroutine call with W. First, the return address (PC + 1) is pushed onto the return stack. Then, the contents of W is loaded into PC<7:0>, and the contents of PCLATH into PC<14:8>. CALLW is a 2-cycle instruction. |

**CLRF** **Clear f**

| | |
|---|---|
| Syntax: | [ *label* ] CLRF f |
| Operands: | $0 \le f \le 127$ |
| 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. Status bits $\overline{TO}$ and $\overline{PD}$ are set. |

**COMF** **Complement f**

| | |
|---|---|
| Syntax: | [ *label* ] COMF f,d |
| Operands: | $0 \le f \le 127$<br>$d \in [0,1]$ |
| 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: | $0 \le f \le 127$<br>$d \in [0,1]$ |
| 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: | $0 \le f \le 127$<br>$d \in [0,1]$ |
| Operation: | (f) - 1 → (destination);<br>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'.<br>If the result is '1', the next instruction is executed. If the result is '0', then a NOP is executed instead, making it a 2-cycle instruction. |

**GOTO** **Unconditional Branch**

| | |
|---|---|
| Syntax: | [ *label* ] GOTO k |
| Operands: | $0 \le k \le 2047$ |
| Operation: | k → PC<10:0><br>PCLATH<6:3> → PC<14:11> |
| Status Affected: | None |
| Description: | GOTO is an unconditional branch. The 11-bit immediate value is loaded into PC bits <10:0>. The upper bits of PC are loaded from PCLATH<4:3>. GOTO is a 2-cycle instruction. |

**INCF** **Increment f**

| | |
|---|---|
| Syntax: | [ *label* ] INCF f,d |
| Operands: | $0 \le f \le 127$<br>$d \in [0,1]$ |
| 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: | $0 \le f \le 127$<br>$d \in [0,1]$ |
| Operation: | (f) + 1 → (destination),<br>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'.<br>If the result is '1', the next instruction is executed. If the result is '0', a NOP is executed instead, making it a 2-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 8-bit literal 'k'. The result is placed in the W register. |

**IORWF** **Inclusive OR W with f**

| | |
|---|---|
| Syntax: | [ *label* ] IORWF f,d |
| Operands: | $0 \le f \le 127$<br>$d \in [0,1]$ |
| 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'. |


© 2013-2015 Microchip Technology Inc.

**LSLF** **Logical Left Shift**

Syntax: [ *label* ] LSLF f {,d}

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

Operation: (f<7>) → C
(f<6:0>) → dest<7:1>
0 → dest<0>

Status Affected: C, Z

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




**LSRF** **Logical Right Shift**

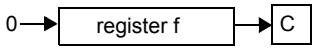
Syntax: [ *label* ] LSRF f {,d}

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

Operation: 0 → dest<7>
(f<7:1>) → dest<6:0>,
(f<0>) → C,

Status Affected: C, Z

Description: The contents of register 'f' are shifted one bit to the right through the Carry flag. A '0' is shifted into the MSb. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is stored 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

**MOVIW** **Move INDFn to W**

Syntax: [ *label* ] MOVIW ++FSRn
[ *label* ] MOVIW --FSRn
[ *label* ] MOVIW FSRn++
[ *label* ] MOVIW FSRn--
[ *label* ] MOVIW k[FSRn]

Operands: n ∈ [0,1]
mm ∈ [00,01, 10, 11]
-32 ≤ k ≤ 31

Operation: INDFn → W
Effective address is determined by
- FSR + 1 (preincrement)
- FSR - 1 (predecrement)
- FSR + k (relative offset)

After the Move, the FSR value will be either:
- FSR + 1 (all increments)
- FSR - 1 (all decrements)
- Unchanged

Status Affected: Z

| Mode | Syntax | mm |
|---|---|---|
| Preincrement | ++FSRn | 00 |
| Predecrement | --FSRn | 01 |
| Postincrement | FSRn++ | 10 |
| Postdecrement | FSRn-- | 11 |

Description: This instruction is used to move data between W and one of the indirect registers (INDFn). Before/after this move, the pointer (FSRn) is updated by pre/post incrementing/decrementing it.

**Note:** The INDFn registers are not physical registers. Any instruction that accesses an INDFn register actually accesses the register at the address specified by the FSRn.

FSRn is limited to the range 0000h - FFFFh. Incrementing/decrementing it beyond these bounds will cause it to wrap-around.

**MOVLB** **Move literal to BSR**

Syntax: [ *label* ] MOVLB k

Operands: 0 ≤ k ≤ 31

Operation: k → BSR

Status Affected: None

Description: The 5-bit literal 'k' is loaded into the Bank Select Register (BSR).

**MOVLP** **Move literal to PCLATH**

Syntax: [ *label* ] MOVLP k

Operands: 0 ≤ k ≤ 127

Operation: k → PCLATH

Status Affected: None

Description: The 7-bit literal 'k' is loaded into the PCLATH register.

**MOVLW** **Move literal to W**

Syntax: [ *label* ] MOVLW k

Operands: 0 ≤ k ≤ 255

Operation: k → (W)

Status Affected: None

Description: The 8-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

**MOVWF** **Move W to f**

Syntax: [ *label* ] MOVWF f

Operands: 0 ≤ f ≤ 127

Operation: (W) → (f)

Status Affected: None

Description: Move data from W register to register 'f'.

Words: 1

Cycles: 1

Example:
```
MOVWF   OPTION_REG
```
Before Instruction
OPTION_REG = 0xFF
W = 0x4F
After Instruction
OPTION_REG = 0x4F
W = 0x4F


© 2013-2015 Microchip Technology Inc.

**MOVWI** **Move W to INDFn**

Syntax: [ *label* ] MOVWI ++FSRn
[ *label* ] MOVWI --FSRn
[ *label* ] MOVWI FSRn++
[ *label* ] MOVWI FSRn--
[ *label* ] MOVWI k[FSRn]

Operands: $n \in [0,1]$
$mm \in [00,01, 10, 11]$
$-32 \leq k \leq 31$

Operation: W → INDFn
Effective address is determined by
- FSR + 1 (preincrement)
- FSR - 1 (predecrement)
- FSR + k (relative offset)

After the Move, the FSR value will be either:
- FSR + 1 (all increments)
- FSR - 1 (all decrements)

Unchanged

Status Affected: None

| Mode | Syntax | mm |
|---|---|---|
| Preincrement | ++FSRn | 00 |
| Predecrement | --FSRn | 01 |
| Postincrement | FSRn++ | 10 |
| Postdecrement | FSRn-- | 11 |

Description: This instruction is used to move data between W and one of the indirect registers (INDFn). Before/after this move, the pointer (FSRn) is updated by pre/post incrementing/decrementing it.

**Note:** The INDFn registers are not physical registers. Any instruction that accesses an INDFn register actually accesses the register at the address specified by the FSRn.

FSRn is limited to the range 0000h-FFFFh. Incrementing/decrementing it beyond these bounds will cause it to wrap-around.

The increment/decrement operation on FSRn WILL NOT affect any Status bits.

**NOP** **No Operation**

Syntax: [ *label* ] NOP

Operands: None

Operation: No operation

Status Affected: None

Description: No operation.

Words: 1

Cycles: 1

Example:
```
NOP
```

**OPTION** **Load OPTION_REG Register with W**

Syntax: [ *label* ] OPTION

Operands: None

Operation: (W) → OPTION_REG

Status Affected: None

Description: Move data from W register to OPTION_REG register.

Words: 1

Cycles: 1

Example:
```
OPTION
```
Before Instruction
OPTION_REG = 0xFF
W = 0x4F
After Instruction
OPTION_REG = 0x4F
W = 0x4F

**RESET** **Software Reset**

Syntax: [ *label* ] RESET

Operands: None

Operation: Execute a device Reset. Resets the $\overline{RI}$ flag of the PCON register.

Status Affected: None

Description: This instruction provides a way to execute a hardware Reset by software.

**RETFIE** **Return from Interrupt**

Syntax: [ *label* ] RETFIE k

Operands: None

Operation: TOS → PC,
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 2-cycle instruction.

Words: 1

Cycles: 2

Example:

```
RETFIE
```

After Interrupt
PC = TOS
GIE = 1

**RETLW** **Return with literal in W**

Syntax: [ *label* ] RETLW k

Operands: 0 ≤ k ≤ 255

Operation: k → (W);
TOS → PC

Status Affected: None

Description: The W register is loaded with the 8-bit literal 'k'. The program counter is loaded from the top of the stack (the return address). This is a 2-cycle instruction.

Words: 1

Cycles: 2

Example:

```
CALL TABLE;W contains table
    ;offset value
•   ;W now has table value
•
•
ADDWF PC  ;W = offset
RETLW k1  ;Begin table
RETLW k2  ;
•
•
•
RETLW kn  ; End of table
```

Before Instruction
W = 0x07
After Instruction
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 2-cycle instruction.

**RLF** **Rotate Left f through Carry**

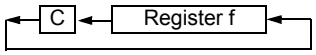
Syntax: [ *label* ] RLF f,d

Operands: 0 ≤ f ≤ 127
d ∈ [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


© 2013-2015 Microchip Technology Inc.

**RRF** **Rotate Right f through Carry**

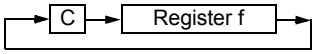
Syntax: [ *label* ] RRF f,d

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 8-bit literal 'k'. The result is placed in the W register.

| | |
|---|---|
| C = 0 | W > k |
| C = 1 | W ≤ k |
| DC = 0 | W<3:0> > k<3:0> |
| DC = 1 | W<3:0> ≤ k<3:0> |

**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.

| | |
|---|---|
| C = 0 | W > f |
| C = 1 | W ≤ f |
| DC = 0 | W<3:0> > f<3:0> |
| DC = 1 | W<3:0> ≤ f<3:0> |

**SUBWFB** **Subtract W from f with Borrow**

Syntax: SUBWFB f {,d}

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

Operation: (f) – (W) – ($\overline{B}$) → dest

Status Affected: C, DC, Z

Description: Subtract W and the BORROW flag (CARRY) from register 'f' (2's complement method). If 'd' is '0', the result is stored in W. 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'. |

**TRIS** **Load TRIS Register with W**

| | |
|---|---|
| Syntax: | [ *label* ] TRIS f |
| Operands: | $5 \le f \le 7$ |
| Operation: | (W) → TRIS register 'f' |
| Status Affected: | None |
| Description: | Move data from W register to TRIS register.<br>When 'f' = 5, TRISA is loaded.<br>When 'f' = 6, TRISB is loaded.<br>When 'f' = 7, TRISC is loaded. |

**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 8-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'. |


© 2013-2015 Microchip Technology Inc.

# 26.0 ELECTRICAL SPECIFICATIONS

## 26.1 Absolute Maximum Ratings(†)

Ambient temperature under bias ..... -40°C to +125°C
Storage temperature ..... -65°C to +150°C
Voltage on pins with respect to $V_{SS}$
- on $V_{DD}$ pin
  - PIC16F1703/7 ..... -0.3V to +6.5V
  - PIC16LF1703/7 ..... -0.3V to +4.0V
- on $\overline{MCLR}$ pin ..... -0.3V to +9.0V
- on all other pins ..... -0.3V to ($V_{DD}$ + 0.3V)

Maximum current
- on $V_{SS}$ pin(1)
  - -40°C ≤ $T_A$ ≤ +85°C ..... 250 mA
  - +85°C ≤ $T_A$ ≤ +125°C ..... 85 mA
- on $V_{DD}$ pin(1)
  - -40°C ≤ $T_A$ ≤ +85°C ..... 250 mA
  - +85°C ≤ $T_A$ ≤ +125°C ..... 85 mA
- sunk by any I/O pin ..... 50 mA
- sourced by any I/O pin ..... 50 mA
- sourced by any Op Amp output pin ..... 100 mA

Clamp current, $I_K$ ($V_{PIN}$ < 0 or $V_{PIN}$ > $V_{DD}$) ..... ±20 mA
Total power dissipation(2) ..... 800 mW

**Note 1:** Maximum current rating requires even load distribution across I/O pins. Maximum current rating may be limited by the device package power dissipation characterizations, see Table 26-6 to calculate device specifications.

**2:** 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 above maximum rating conditions for extended periods may affect device reliability.

## 26.2 Standard Operating Conditions

The standard operating conditions for any device are defined as:

Operating Voltage: $V_{DDMIN} \leq V_{DD} \leq V_{DDMAX}$
Operating Temperature: $T_{A\_MIN} \leq T_A \leq T_{A\_MAX}$

**$V_{DD}$ — Operating Supply Voltage(1)**

PIC16LF1703/7

$V_{DDMIN}$ (Fosc ≤ 16 MHz) ..... +1.8V
$V_{DDMIN}$ (Fosc > 16 MHz) ..... +2.5V
$V_{DDMAX}$ ..... +3.6V

PIC16F1703/7

$V_{DDMIN}$ (Fosc ≤ 16 MHz) ..... +2.3V
$V_{DDMIN}$ (Fosc > 16 MHz) ..... +2.5V
$V_{DDMAX}$ ..... +5.5V

**$T_A$ — Operating Ambient Temperature Range**

Industrial Temperature

$T_{A\_MIN}$ ..... -40°C
$T_{A\_MAX}$ ..... +85°C

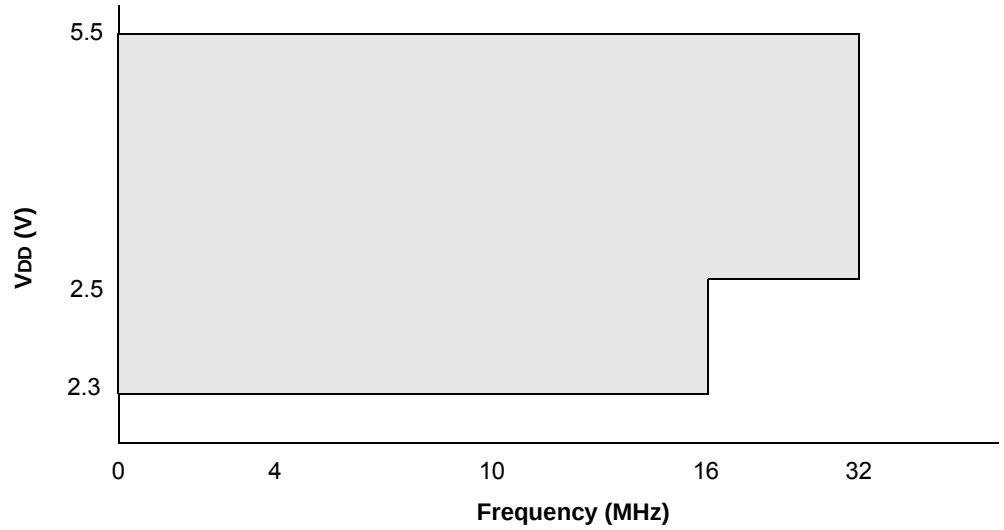
Extended Temperature

$T_{A\_MIN}$ ..... -40°C
$T_{A\_MAX}$ ..... +125°C

**Note 1:** See Parameter D001, DS Characteristics: Supply Voltage.

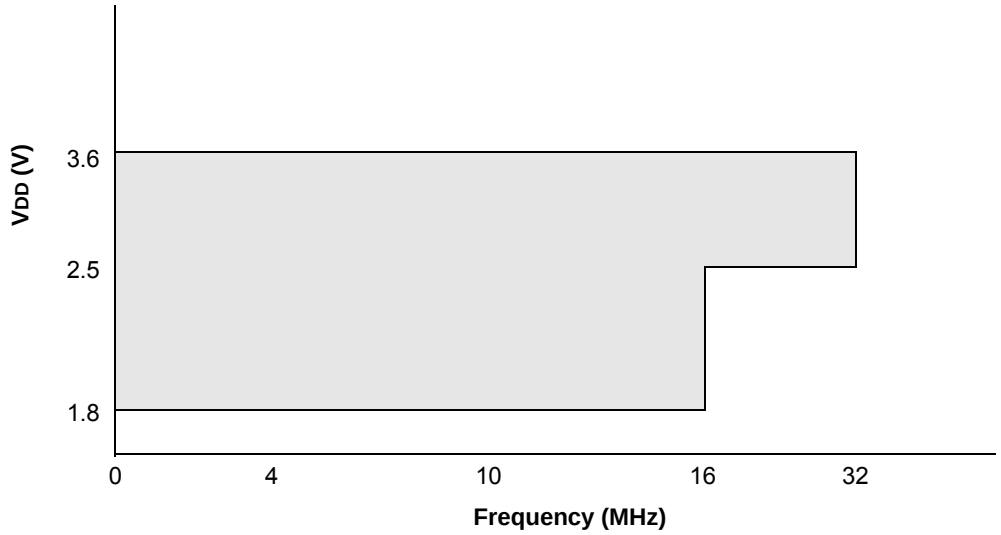

© 2013-2015 Microchip Technology Inc.

**FIGURE 26-1: VOLTAGE FREQUENCY GRAPH, -40°C ≤ TA ≤ +125°C, PIC16F1703/7 ONLY**




**Note 1:** The shaded region indicates the permissible combinations of voltage and frequency.
**2:** Refer to Table 26-7 for each Oscillator mode's supported frequencies.

**FIGURE 26-2: VOLTAGE FREQUENCY GRAPH, -40°C ≤ TA ≤ +125°C, PIC16LF1703/7 ONLY**




**Note 1:** The shaded region indicates the permissible combinations of voltage and frequency.
**2:** Refer to Table 26-7 for each Oscillator mode's supported frequencies.

## 26.3 DC Characteristics

### TABLE 26-1: SUPPLY VOLTAGE

| PIC16LF1703/7 | | | | Standard Operating Conditions (unless otherwise stated) | | | | |
|---|---|---|---|---|---|---|---|---|
| PIC16F1703/7 | | | | | | | | |
| Param. No. | Sym. | Characteristic | | Min. | Typ† | Max. | Units | Conditions |
| D001 | VDD | **Supply Voltage** | | | | | | |
| | | | | VDDMIN<br>1.8<br>2.5 | —<br>—<br>— | VDDMAX<br>3.6<br>3.6 | V<br>V<br>V | FOSC ≤ 16 MHz<br>FOSC ≤ 32 MHz (**Note 2**) |
| D001 | | PIC16F1703/7 | | 2.3<br>2.5 | —<br>— | 5.5<br>5.5 | V<br>V | FOSC ≤ 16 MHz:<br>FOSC ≤ 32 MHz (**Note 2**) |
| D002* | VDR | **RAM Data Retention Voltage(1)** | | | | | | |
| | | | | 1.5 | — | — | V | Device in Sleep mode |
| D002* | | | | 1.7 | — | — | V | Device in Sleep mode |
| D002A* | VPOR | **Power-on Reset Release Voltage(3)** | | | | | | |
| | | | | — | 1.6 | — | V | |
| D002A* | | | | — | 1.6 | — | V | |
| D002B* | VPORR* | **Power-on Reset Rearm Voltage(3)** | | | | | | |
| | | | | — | 0.8 | — | V | |
| D002B* | | | | — | 1.5 | — | V | |
| D003 | VFVR | **Fixed Voltage Reference Voltage** | 1x (1.024 typical) | -4 | — | +4 | % | VDD ≥ 2.5V, -40°C to +85°C |
| | | | 2x (2.048 typical) | -4 | — | +4 | % | VDD ≥ 2.5V, -40°C to +85°C |
| | | | 4x (4.096 typical) | -5 | — | +5 | % | VDD ≥ 4.75V, -40°C to +85°C |
| D004* | SVDD | **VDD Rise Rate(2)** | | 0.05 | — | — | V/ms | Ensures that the Power-on Reset signal is released properly. |

* These parameters are characterized but not tested.

† Data in "Typ" column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

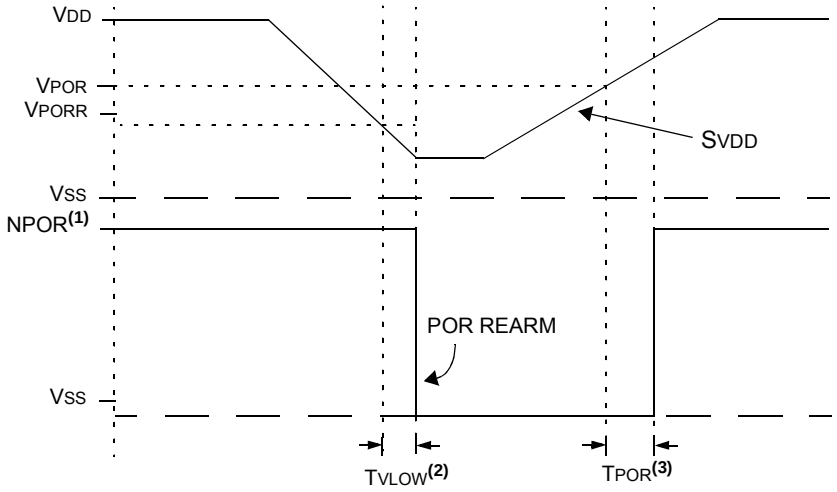
**Note 1:** This is the limit to which VDD can be lowered in Sleep mode without losing RAM data.

**2:** PLL required for 32 MHz operation.

**3:** See Figure 26-3: POR and POR Rearm with Slow Rising VDD.


© 2013-2015 Microchip Technology Inc.

**FIGURE 26-3: POR AND POR REARM WITH SLOW RISING VDD**




**Note 1:** When NPOR is low, the device is held in Reset.
**2:** TPOR 1 μs typical.
**3:** TVLOW 2.7 μs typical.

**TABLE 26-2: SUPPLY CURRENT ($I_{DD}$)(1,2)**

| PIC16LF1703/7 | | Standard Operating Conditions (unless otherwise stated) | | | | | |
|---|---|---|---|---|---|---|---|
| PIC16F1703/7 | | | | | | | |
| Param No. | Device Characteristics | Min. | Typ† | Max. | Units | Conditions | |
| | | | | | | $V_{DD}$ | Note |
| D009 | LDO Regulator | — | 75 | — | µA | — | High-Power mode, normal operation |
| | | — | 15 | — | µA | — | Sleep, VREGCON<1> = 0 |
| | | — | 0.3 | — | µA | — | Sleep, VREGCON<1> = 1 |
| D014 | | — | 120 | 180 | µA | 1.8 | $F_{OSC}$ = 4 MHz, External Clock (ECM), Medium-Power mode |
| | | — | 210 | 300 | µA | 3.0 | |
| D014 | | — | 185 | 280 | µA | 2.3 | $F_{OSC}$ = 4 MHz, External Clock (ECM), Medium-Power mode |
| | | — | 245 | 350 | µA | 3.0 | |
| | | — | 305 | 420 | µA | 5.0 | |
| D015 | | — | 1.6 | 2.3 | mA | 3.0 | $F_{OSC}$ = 32 MHz, External Clock (ECH), High-Power mode **(Note 3)** |
| | | — | 2 | 2.6 | mA | 3.6 | |
| D015 | | — | 2 | 2.3 | mA | 3.0 | $F_{OSC}$ = 32 MHz, External Clock (ECH), High-Power mode **(Note 3)** |
| | | — | 2 | 2.5 | mA | 5.0 | |
| D017 | | — | 115 | 170 | µA | 1.8 | $F_{OSC}$ = 500 kHz, MFINTOSC mode |
| | | — | 135 | 200 | µA | 3.0 | |
| D017 | | — | 150 | 200 | µA | 2.3 | $F_{OSC}$ = 500 kHz, MFINTOSC mode |
| | | — | 170 | 220 | µA | 3.0 | |
| | | — | 215 | 280 | µA | 5.0 | |
| D019 | | — | 0.7 | 0.9 | mA | 1.8 | $F_{OSC}$ = 16 MHz, HFINTOSC mode |
| | | — | 1.2 | 1.4 | mA | 3.0 | |
| D019 | | — | 0.9 | 1.1 | mA | 2.3 | $F_{OSC}$ = 16 MHz, HFINTOSC mode |
| | | — | 1.1 | 1.5 | mA | 3.0 | |
| | | — | 1.2 | 1.6 | mA | 5.0 | |
| D020 | | — | 2 | 2.5 | mA | 3.0 | $F_{OSC}$ = 32 MHz, HFINTOSC mode **(Note 3)** |
| | | — | 2.4 | 3.0 | mA | 3.6 | |
| D020 | | — | 2 | 2.6 | mA | 3.0 | $F_{OSC}$ = 32 MHz, HFINTOSC mode **(Note 3)** |
| | | — | 2.2 | 2.8 | mA | 5.0 | |

† Data in "Typ" column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** The test conditions for all $I_{DD}$ measurements in active operation mode are: all I/O pins tri-stated, pulled to $V_{DD}$; $\overline{MCLR}$ = $V_{DD}$; 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:** 8 MHz clock with 4x PLL enabled.


© 2013-2015 Microchip Technology Inc.

**TABLE 26-3: POWER-DOWN CURRENTS ($I_{PD}$)(1,2)**

| PIC16LF1703/7 | | | Operating Conditions: (unless otherwise stated)<br>Low-Power Sleep Mode | | | | | |
|---|---|---|---|---|---|---|---|---|
| PIC16F1703/7 | | | Low-Power Sleep Mode, VREGPM = 1 | | | | | |
| Param No. | Device Characteristics | Min. | Typ† | Max. +85°C | Max. +125°C | Units | Conditions | |
| | | | | | | | $V_{DD}$ | Note |
| D022 | Base $I_{PD}$ | — | 0.05 | 1.0 | 8.0 | µA | 1.8 | WDT, BOR and FVR disabled, all Peripherals Inactive |
| | | — | 0.08 | 2.0 | 9.0 | µA | 3.0 | |
| D022 | Base $I_{PD}$ | — | 0.3 | 3 | 11 | µA | 2.3 | WDT, BOR and FVR disabled, all Peripherals Inactive, Low-Power Sleep mode VREGPM = 1 |
| | | — | 0.4 | 4 | 12 | µA | 3.0 | |
| | | — | 0.5 | 6 | 15 | µA | 5.0 | |
| D022A | Base $I_{PD}$ | — | 9.8 | 16 | 18 | µA | 2.3 | WDT, BOR and FVR disabled, all Peripherals inactive, Normal-Power Sleep mode |
| | | — | 10.3 | 18 | 20 | µA | 3.0 | |
| | | — | 11.5 | 21 | 26 | µA | 5.0 | |
| D024 | | — | 0.5 | 6 | 14 | µA | 1.8 | WDT Current |
| | | — | 0.8 | 7 | 17 | µA | 3.0 | |
| D024 | | — | 0.8 | 6 | 15 | µA | 2.3 | WDT Current |
| | | — | 0.9 | 7 | 20 | µA | 3.0 | |
| | | — | 1.0 | 8 | 22 | µA | 5.0 | |
| D025 | | — | 15 | 28 | 30 | µA | 1.8 | FVR Current |
| | | — | 18 | 30 | 33 | µA | 3.0 | |
| D025 | | — | 18 | 33 | 35 | µA | 2.3 | FVR Current |
| | | — | 19 | 35 | 37 | µA | 3.0 | |
| | | — | 20 | 37 | 39 | µA | 5.0 | |
| D026 | | — | 7.5 | 25 | 28 | µA | 3.0 | BOR Current |
| D026 | | — | 10 | 25 | 28 | µA | 3.0 | BOR Current |
| | | — | 12 | 28 | 31 | µA | 5.0 | |
| D027 | | — | 0.5 | 4 | 10 | µA | 3.0 | LPBOR Current |
| D027 | | — | 0.8 | 6 | 14 | µA | 3.0 | LPBOR Current |
| | | — | 1 | 8 | 17 | µA | 5.0 | |
| D029 | | — | 0.05 | 2 | 9 | µA | 1.8 | ADC Current **(Note 3)**, no conversion in progress |
| | | — | 0.08 | 3 | 10 | µA | 3.0 | |
| D029 | | — | 0.3 | 4 | 12 | µA | 2.3 | ADC Current **(Note 3)**, no conversion in progress |
| | | — | 0.4 | 5 | 13 | µA | 3.0 | |
| | | — | 0.5 | 7 | 16 | µA | 5.0 | |
| D030 | | — | 250 | — | — | µA | 1.8 | ADC Current **(Note 3)**, conversion in progress |
| | | — | 250 | — | — | µA | 3.0 | |
| D030 | | — | 280 | — | — | µA | 2.3 | ADC Current **(Note 3)**, conversion in progress |
| | | — | 280 | — | — | µA | 3.0 | |
| | | — | 280 | — | — | µA | 5.0 | |

* These parameters are characterized but not tested.

† Data in "Typ" column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** The peripheral current is the sum of the base $I_{PD}$ and the additional current consumed when this peripheral is enabled. The peripheral Δ current can be determined by subtracting the base $I_{DD}$ or $I_{PD}$ current from this limit. Max values should be used when calculating total current consumption.

**2:** 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 $V_{SS}$.

**3:** ADC clock source is FRC.

**TABLE 26-3: POWER-DOWN CURRENTS ($I_{PD}$)(1,2) (CONTINUED)**

| PIC16LF1703/7 | | | Operating Conditions: (unless otherwise stated)<br>Low-Power Sleep Mode | | | | | |
|---|---|---|---|---|---|---|---|---|
| PIC16F1703/7 | | | Low-Power Sleep Mode, VREGPM = 1 | | | | | |
| Param No. | Device Characteristics | Min. | Typ† | Max. +85°C | Max. +125°C | Units | Conditions | |
| | | | | | | | $V_{DD}$ | Note |
| D031 | | — | 250 | 650 | — | µA | 3.0 | Op Amp (High-power) |
| D031 | | — | 250 | 650 | — | µA | 3.0 | Op Amp (High-power) |
| | | — | 350 | 850 | — | µA | 5.0 | |

* These parameters are characterized but not tested.

† Data in "Typ" column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** The peripheral current is the sum of the base $I_{PD}$ and the additional current consumed when this peripheral is enabled. The peripheral Δ current can be determined by subtracting the base $I_{DD}$ or $I_{PD}$ current from this limit. Max values should be used when calculating total current consumption.

**2:** 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 $V_{SS}$.

**3:** ADC clock source is FRC.


© 2013-2015 Microchip Technology Inc.

**TABLE 26-4: I/O PORTS**

| Standard Operating Conditions (unless otherwise stated) | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Sym.** | **Characteristic** | **Min.** | **Typ†** | **Max.** | **Units** | **Conditions** |
| | VIL | **Input Low Voltage** | | | | | |
| | | I/O PORT: | | | | | |
| D034 | | with TTL buffer | — | — | 0.8 | V | 4.5V ≤ VDD ≤ 5.5V |
| D034A | | | — | — | 0.15 VDD | V | 1.8V ≤ VDD ≤ 4.5V |
| D035 | | with Schmitt Trigger buffer | — | — | 0.2 VDD | V | 2.0V ≤ VDD ≤ 5.5V |
| | | with $I^2C$ levels | — | — | 0.3 VDD | V | |
| | | with SMBus levels | — | — | 0.8 | V | 2.7V ≤ VDD ≤ 5.5V |
| D036 | | $\overline{MCLR}$ | — | — | 0.2 VDD | V | |
| | VIH | **Input High Voltage** | | | | | |
| | | I/O ports: | | | | | |
| D040 | | with TTL buffer | 2.0 | — | — | V | 4.5V ≤ VDD ≤ 5.5V |
| D040A | | | 0.25 VDD + 0.8 | — | — | V | 1.8V ≤ VDD ≤ 4.5V |
| D041 | | with Schmitt Trigger buffer | 0.8 VDD | — | — | V | 2.0V ≤ VDD ≤ 5.5V |
| | | with $I^2C$ levels | 0.7 VDD | — | — | V | |
| | | with SMBus levels | 2.1 | — | — | V | 2.7V ≤ VDD ≤ 5.5V |
| D042 | | $\overline{MCLR}$ | 0.8 VDD | — | — | V | |
| | IIL | **Input Leakage Current(1)** | | | | | |
| D060 | | I/O Ports | — | ± 5 | ± 125 | nA | VSS ≤ VPIN ≤ VDD, Pin at high-impedance, 85°C |
| | | | — | ± 5 | ± 1000 | nA | VSS ≤ VPIN ≤ VDD, Pin at high-impedance, 125°C |
| D061 | | $\overline{MCLR}$(2) | — | ± 50 | ± 200 | nA | VSS ≤ VPIN ≤ VDD, Pin at high-impedance, 85°C |
| | IPUR | **Weak Pull-up Current** | | | | | |
| D070* | | | 25 | 100 | 200 | µA | VDD = 3.0V, VPIN = VSS |
| | VOL | **Output Low Voltage** | | | | | |
| D080 | | I/O ports | — | — | 0.6 | V | IOL = -8mA, VDD = 5V<br>IOL = -6mA, VDD = 3.3V<br>IOL = -1.8mA, VDD = 1.8V |
| | VOH | **Output High Voltage** | | | | | |
| D090 | | I/O ports | VDD - 0.7 | — | — | V | IOH = 3.5mA, VDD = 5V<br>IOH = 3mA, VDD = 3.3V<br>IOH = 1mA, VDD = 1.8V |
| | CIO | **Capacitive Loading Specs on Output Pins** | | | | | |
| D101A* | | All I/O pins | — | — | 50 | pF | |

* These parameters are characterized but not tested.

† Data in "Typ" column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

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

**2:** 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.

**TABLE 26-5: MEMORY PROGRAMMING SPECIFICATIONS**

| Standard Operating Conditions (unless otherwise stated) | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Sym.** | **Characteristic** | **Min.** | **Typ†** | **Max.** | **Units** | **Conditions** |
| | | **Program Memory Programming Specifications** | | | | | |
| D110 | $V_{IHH}$ | Voltage on $\overline{MCLR}/V_{PP}$ | 8.0 | — | 9.0 | V | **(Note 3, Note 4)** |
| D111 | $I_{DDP}$ | Supply Current during Programming | — | — | 10 | mA | |
| D112 | $V_{BE}$ | $V_{DD}$ for Bulk Erase | 2.7 | — | $V_{DDMAX}$ | V | |
| D113 | $V_{PEW}$ | $V_{DD}$ for Write or Row Erase | $V_{DDMIN}$ | — | $V_{DDMAX}$ | V | |
| D114 | $I_{PPGM}$ | Current on $\overline{MCLR}/V_{PP}$ during Erase/Write | — | 1.0 | — | mA | |
| D115 | $I_{DDPGM}$ | Current on $V_{DD}$ during Erase/Write | — | 5.0 | — | mA | |
| | | **Program Flash Memory** | | | | | |
| D121 | $E_P$ | Cell Endurance | 10K | — | — | E/W | -40°C ≤ $T_A$ ≤ +85°C **(Note 1)** |
| D122 | $V_{PRW}$ | $V_{DD}$ for Read/Write | $V_{DDMIN}$ | — | $V_{DDMAX}$ | V | |
| D123 | $T_{IW}$ | Self-timed Write Cycle Time | — | 2 | 2.5 | ms | |
| D124 | $T_{RETD}$ | Characteristic Retention | — | 40 | — | Year | Provided no other specifications are violated |
| D125 | $E_{HEFC}$ | High-Endurance Flash Cell | 100K | — | — | E/W | -0°C ≤ $T_A$ ≤ +60°C, Lower byte last 128 addresses |

† Data in "Typ" column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** Self-write and Block Erase.

**2:** Refer to Section **Section 3.2 "High-Endurance Flash"** for a more detailed discussion on data EEPROM endurance.

**3:** Required only if single-supply programming is disabled.

**4:** The MPLAB ICD 2 does not support variable VPP output. Circuitry to limit the ICD 2 VPP voltage must be placed between the ICD 2 and target system when programming or debugging with the ICD 2.


© 2013-2015 Microchip Technology Inc.

**TABLE 26-6: THERMAL CHARACTERISTICS**

| Standard Operating Conditions (unless otherwise stated) | | | | | |
|---|---|---|---|---|---|
| **Param No.** | **Sym.** | **Characteristic** | **Typ.** | **Units** | **Conditions** |
| TH01 | $\theta_{JA}$ | Thermal Resistance Junction to Ambient | 70.0 | °C/W | 14-pin PDIP package |
| | | | 95.3 | °C/W | 14-pin SOIC package |
| | | | 100.0 | °C/W | 14-pin TSSOP package |
| | | | 51.5 | °C/W | 16-pin QFN 4x4mm package |
| | | | 62.2 | °C/W | 20-pin PDIP package |
| | | | 87.3 | °C/W | 20-pin SSOP |
| | | | 77.7 | °C/W | 20-pin SOIC package |
| | | | 43.0 | °C/W | 20-pin QFN 4x4mm package |
| TH02 | $\theta_{JC}$ | Thermal Resistance Junction to Case | 32.75 | °C/W | 14-pin PDIP package |
| | | | 31.0 | °C/W | 14-pin SOIC package |
| | | | 24.4 | °C/W | 14-pin TSSOP package |
| | | | 5.4 | °C/W | 16-pin QFN 4x4mm package |
| | | | 27.5 | °C/W | 20-pin PDIP package |
| | | | 31.1 | °C/W | 20-pin SSOP |
| | | | 23.1 | °C/W | 20-pin SOIC package |
| | | | 5.3 | °C/W | 20-pin QFN 4x4mm package |
| TH03 | $T_{JMAX}$ | Maximum Junction Temperature | 150 | °C | |
| TH04 | PD | Power Dissipation | — | W | $PD = P_{INTERNAL} + P_{I/O}$ |
| TH05 | $P_{INTERNAL}$ | Internal Power Dissipation | — | W | $P_{INTERNAL} = I_{DD} \times V_{DD}$ (1) |
| TH06 | $P_{I/O}$ | I/O Power Dissipation | — | W | $P_{I/O} = \Sigma (I_{OL} * V_{OL}) + \Sigma (I_{OH} * (V_{DD} - V_{OH}))$ |
| TH07 | $P_{DER}$ | Derated Power | — | W | $P_{DER} = P_{DMAX} (T_J - T_A)/\theta_{JA}$ (2) |

**Note 1:** $I_{DD}$ is current to run the chip alone without driving any load on the output pins.
**2:** $T_A$ = Ambient Temperature, $T_J$ = Junction Temperature

## 26.4 AC Characteristics

Timing Parameter Symbology has been created with one of the following formats:

1. TppS2ppS
2. TppS

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

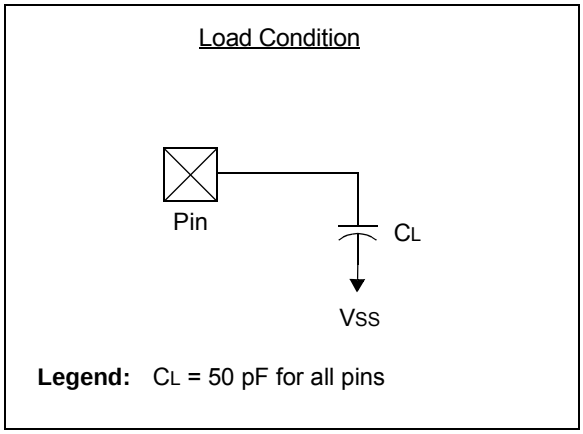
Lowercase letters (pp) and their meanings:

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

Uppercase letters and their meanings:

| | | | |
|---|---|---|---|
| **S** | | | |
| F | Fall | P | Period |
| H | High | R | Rise |
| I | Invalid (High-impedance) | V | Valid |
| L | Low | Z | High-impedance |

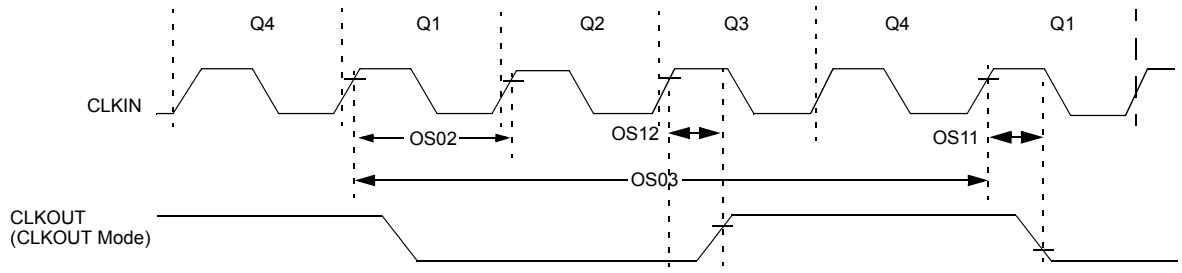
**FIGURE 26-4: LOAD CONDITIONS**




**Legend:** CL = 50 pF for all pins


© 2013-2015 Microchip Technology Inc.

**FIGURE 26-5: CLOCK TIMING**




**Note 1:** See Table 26-10.

**TABLE 26-7: CLOCK OSCILLATOR TIMING REQUIREMENTS**

| Standard Operating Conditions (unless otherwise stated) | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Sym.** | **Characteristic** | **Min.** | **Typ†** | **Max.** | **Units** | **Conditions** |
| OS01 | FOSC | External CLKIN Frequency(1) | DC | — | 0.5 | MHz | External Clock (ECL) |
| | | | DC | — | 4 | MHz | External Clock (ECM) |
| | | | DC | — | 20 | MHz | External Clock (ECH) |
| OS02 | TOSC | External CLKIN Period(1) | 50 | — | ∞ | ns | External Clock (EC) |
| OS03 | TCY | Instruction Cycle Time(1) | 125 | TCY | DC | ns | TCY = 4/FOSC |

* These parameters are characterized but not tested.

† Data in "Typ" column is at 3.0V, 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 CLKIN pin. When an external clock input is used, the "max" cycle time limit is "DC" (no clock) for all devices.

**TABLE 26-8: OSCILLATOR PARAMETERS**

| Standard Operating Conditions (unless otherwise stated) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Param No.** | **Sym.** | **Characteristic** | **Freq. Tolerance** | **Min.** | **Typ†** | **Max.** | **Units** | **Conditions** |
| OS08 | HFOSC | Internal Calibrated HFINTOSC Frequency(1) | ±2% | — | 16.0 | — | MHz | |
| OS08A | MFOSC | Internal Calibrated MFINTOSC Frequency(1) | ±2% | — | 500 | — | kHz | |
| OS09 | LFOSC | Internal LFINTOSC Frequency | — | — | 31 | — | kHz | -40°C ≤ TA ≤ +125°C |
| OS10* | TIOSC ST | HFINTOSC Wake-up from Sleep Start-up Time | — | — | 3.2 | 8 | µs | |
| | | MFINTOSC Wake-up from Sleep Start-up Time | — | — | 24 | 35 | µs | |
| OS10A* | TLFOSC ST | LFINTOSC Wake-up from Sleep Start-up Time | — | — | 0.5 | — | ms | -40°C ≤ TA ≤ +125°C |

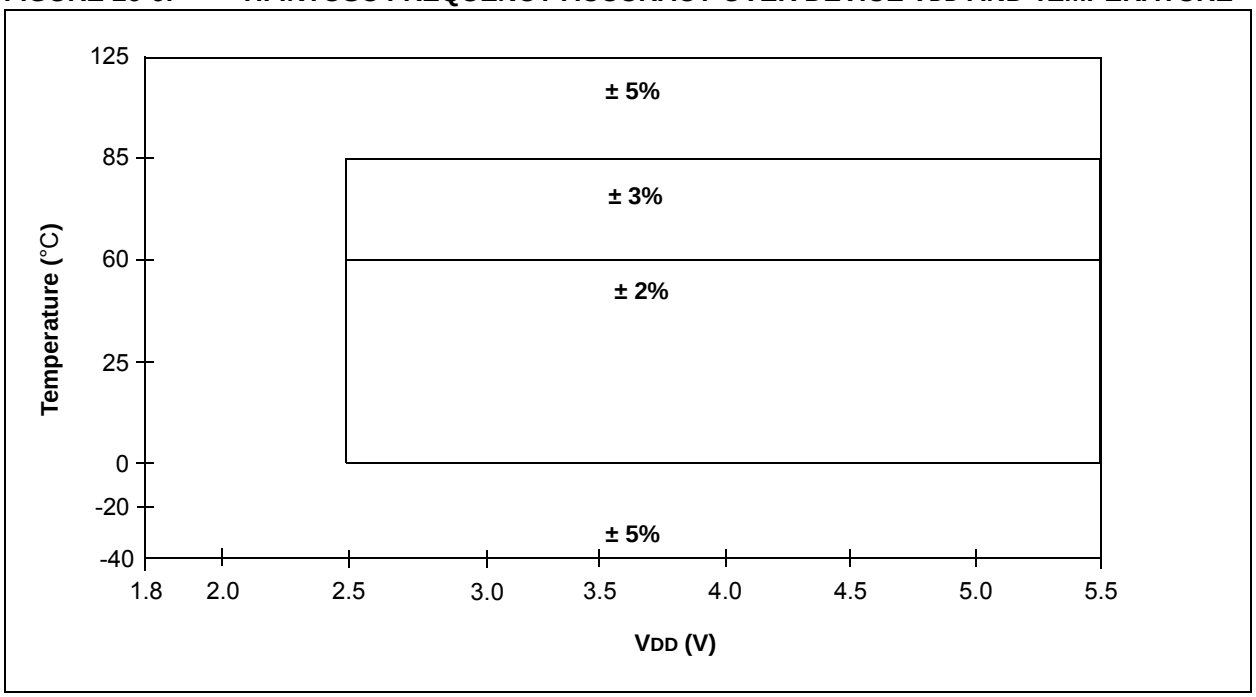
* These parameters are characterized but not tested.

† Data in "Typ" column is at 3.0V, 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.

**2:** See Figure 27-43: LFINTOSC Frequency, PIC16LF1703/7 Only., and Figure 27-44: LFINTOSC Frequency, PIC16F1703/7 Only..

**FIGURE 26-6: HFINTOSC FREQUENCY ACCURACY OVER DEVICE VDD AND TEMPERATURE**





© 2013-2015 Microchip Technology Inc.

**TABLE 26-9: PLL CLOCK TIMING SPECIFICATIONS**

| Standard Operating Conditions (unless otherwise stated) | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Sym.** | **Characteristic** | **Min.** | **Typ†** | **Max.** | **Units** | **Conditions** |
| F10 | FOSC | Oscillator Frequency Range | 4 | — | 8 | MHz | |
| F11 | FSYS | On-Chip VCO System Frequency | 16 | — | 32 | MHz | |
| F12 | TRC | PLL Start-up Time (Lock Time) | — | — | 2 | ms | |
| F13* | ΔCLK | CLKOUT Stability (Jitter) | -0.25% | — | +0.25% | % | |

* 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 26-7: CLKOUT AND I/O TIMING**

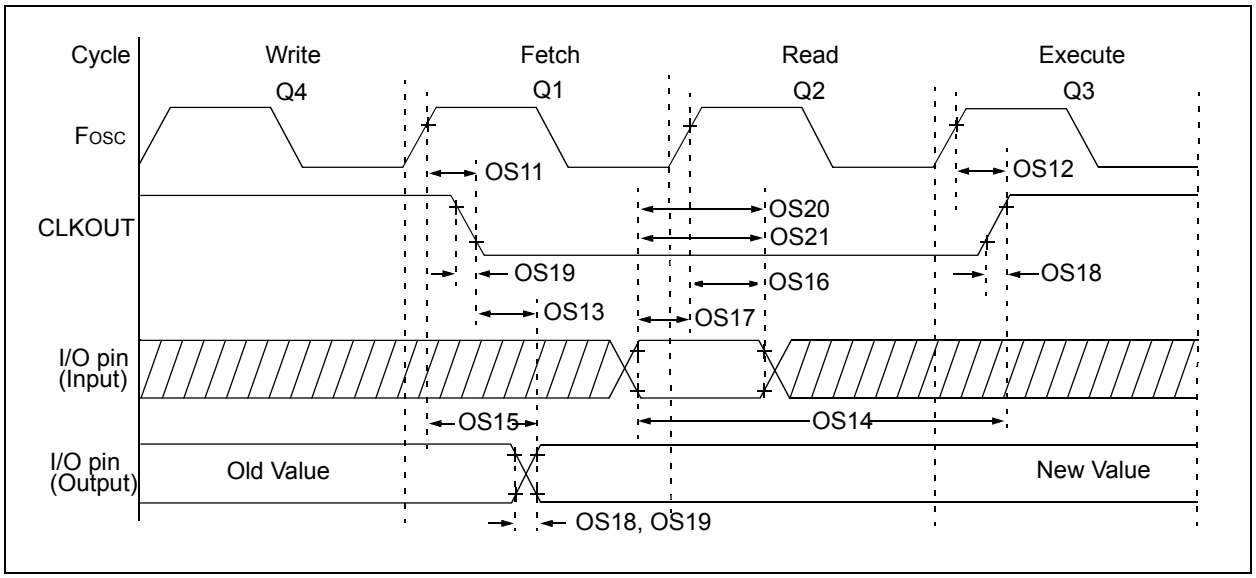



**TABLE 26-10: CLKOUT AND I/O TIMING PARAMETERS**

| Standard Operating Conditions (unless otherwise stated) | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Sym.** | **Characteristic** | **Min.** | **Typ†** | **Max.** | **Units** | **Conditions** |
| OS11* | TosH2ckL | FOSC↑ to CLKOUT↓ (1) | — | — | 70 | ns | 3.3V ≤ VDD ≤ 5.0V |
| OS12* | TosH2ckH | FOSC↑ to CLKOUT↑ (1) | — | — | 72 | ns | 3.3V ≤ VDD ≤ 5.0V |
| OS13* | TckL2ioV | CLKOUT↓ to Port out valid(1) | — | — | 20 | ns | |
| OS14* | TioV2ckH | Port input valid before CLKOUT↑(1) | TOSC + 200 ns | — | — | ns | |
| OS15* | TosH2ioV | Fosc↑ (Q1 cycle) to Port out valid | — | 50 | 70* | ns | 3.3V ≤ VDD ≤ 5.0V |
| OS16* | TosH2ioI | Fosc↑ (Q2 cycle) to Port input invalid (I/O in hold time) | 50 | — | — | ns | 3.3V ≤ VDD ≤ 5.0V |
| OS17* | TioV2osH | Port input valid to Fosc↑ (Q2 cycle) (I/O in setup time) | 20 | — | — | ns | |
| OS18* | TioR | Port output rise time | —<br>— | 40<br>15 | 72<br>32 | ns | VDD = 1.8V<br>3.3V ≤ VDD ≤ 5.0V |
| OS19* | TioF | Port output fall time | —<br>— | 28<br>15 | 55<br>30 | ns | VDD = 1.8V<br>3.3V ≤ VDD ≤ 5.0V |
| OS20* | Tinp | INT pin input high or low time | 25 | — | — | ns | |
| OS21* | Tioc | Interrupt-on-change new input level time | 25 | — | — | ns | |

* These parameters are characterized but not tested.

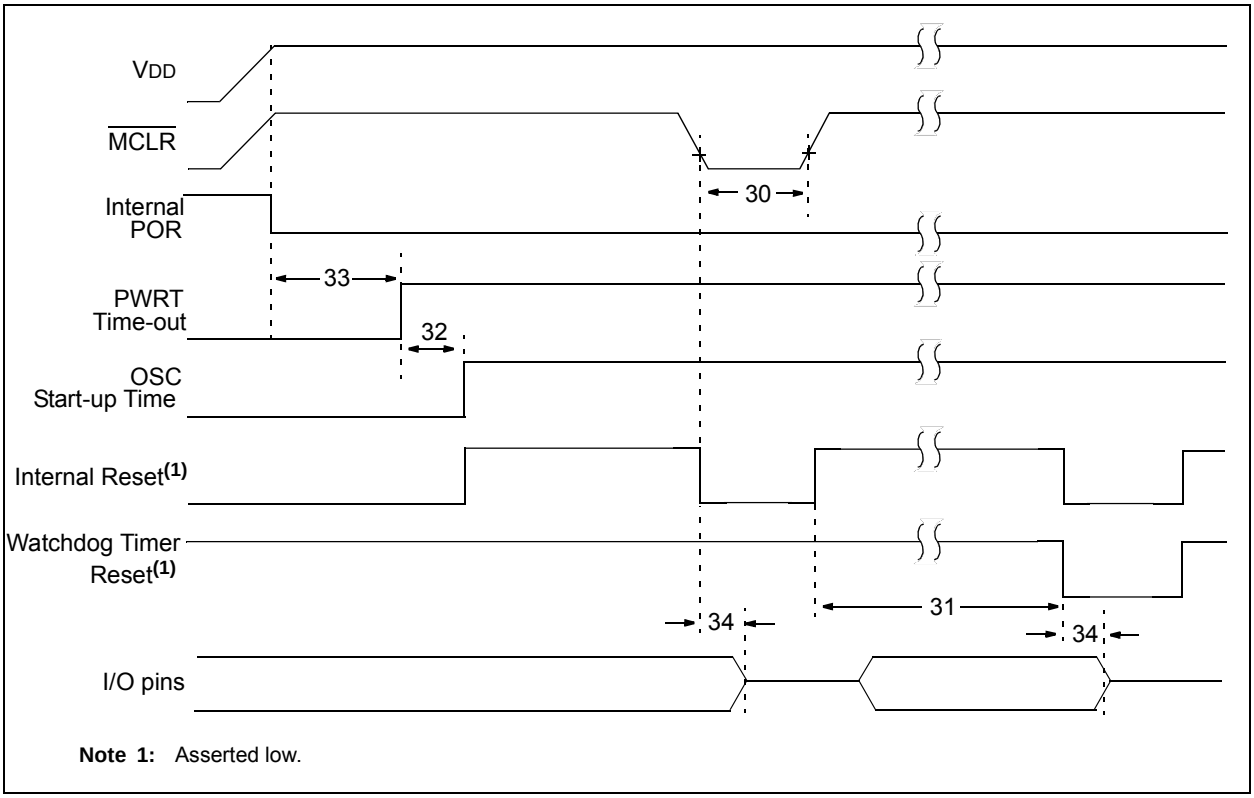
† Data in "Typ" column is at 3.0V, 25°C unless otherwise stated.

**Note 1:** Measurements are taken in EXTRC mode where CLKOUT output is 4 x TOSC.

**2:** Slew rate limited.


© 2013-2015 Microchip Technology Inc.

**FIGURE 26-8: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING**




**Note 1:** Asserted low.

**TABLE 26-11: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER, POWER-UP TIMER AND BROWN-OUT RESET PARAMETERS**

| Standard Operating Conditions (unless otherwise stated) | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Sym.** | **Characteristic** | **Min.** | **Typ†** | **Max.** | **Units** | **Conditions** |
| 30 | TMCL | $\overline{MCLR}$ Pulse Width (low) | 2 | — | — | µs | |
| 31 | TWDTLP | Low-Power Watchdog Timer Time-out Period | 10 | 16 | 27 | ms | VDD = 3.3V-5V<br>1:16 Prescaler used |
| 33* | TPWRT | Power-up Timer Period, $\overline{PWRTE}$ = 0 | 40 | 65 | 140 | ms | |
| 34* | TIOZ | I/O high-impedance from $\overline{MCLR}$ Low or Watchdog Timer Reset | — | — | 2.0 | µs | |
| 35 | VBOR | Brown-out Reset Voltage(1) | 2.55<br><br>2.30<br>1.80 | 2.70<br><br>2.45<br>1.90 | 2.85<br><br>2.60<br>2.10 | V<br><br>V<br>V | BORV = 0<br><br>BORV = 1 (PIC16F1703/7)<br>BORV = 1 (PIC16LF1703/7) |
| 35A | VLPBOR | Low-Power Brown-out | 1.8 | 2.1 | 2.5 | V | LPBOR = 1 |
| 36* | VHYST | Brown-out Reset Hysteresis | 0 | 25 | 75 | mV | -40°C ≤ TA ≤ +85°C |
| 37* | TBORDC | Brown-out Reset DC Response Time | 1 | 3 | 35 | µs | VDD ≤ VBOR |

* These parameters are characterized but not tested.

† Data in "Typ" column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** To ensure these voltage 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 26-9: TIMER0 AND TIMER1 EXTERNAL CLOCK TIMINGS**

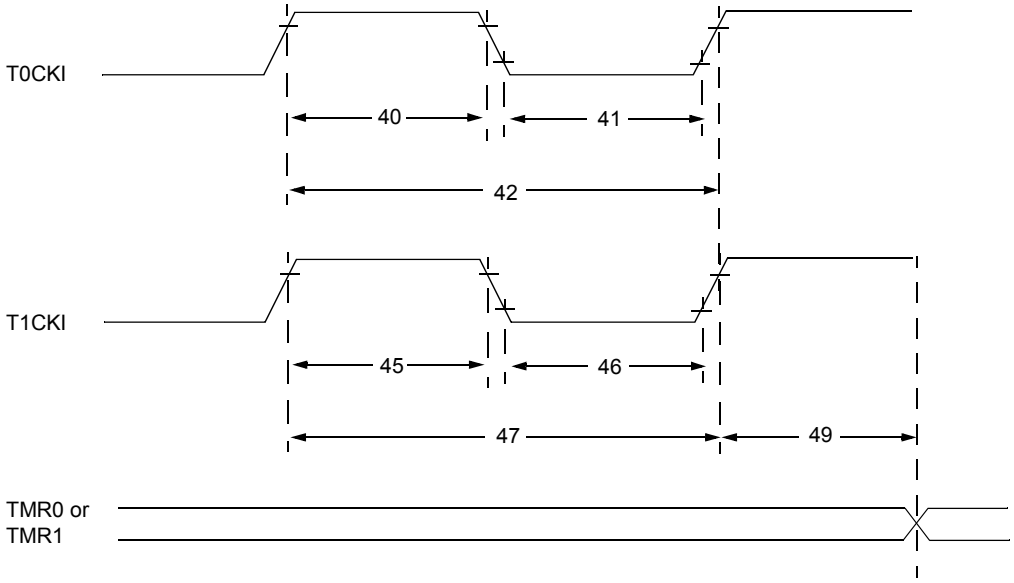
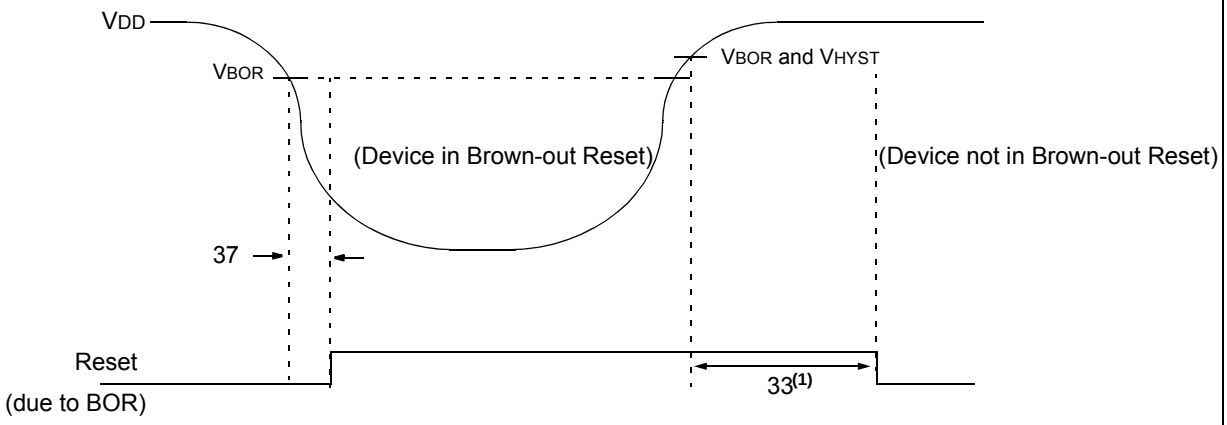



**FIGURE 26-10: BROWN-OUT RESET TIMING AND CHARACTERISTICS**




**Note 1:** 64 ms delay only if $\overline{\text{PWRTE}}$ bit in the Configuration Word register is programmed to '0'; 2 ms delay if $\overline{\text{PWRTE}}$ = 0.


© 2013-2015 Microchip Technology Inc.

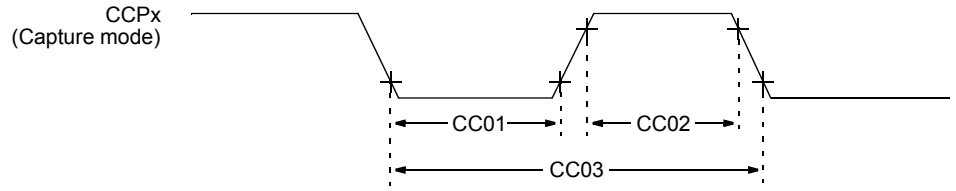
**TABLE 26-12: TIMER0 AND TIMER1 EXTERNAL CLOCK REQUIREMENTS**

| **Standard Operating Conditions (unless otherwise stated)**<br>Operating Temperature -40°C ≤ TA ≤ +125°C | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **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 |
| 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 |
| | | | Asynchronous | 60 | — | — | ns | |
| 49* | TCKEZTMR1 | Delay from External Clock Edge to Timer Increment | | 2 TOSC | — | 7 TOSC | — | Timers in Sync mode |

* These parameters are characterized but not tested.

† Data in "Typ" column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**FIGURE 26-11: CAPTURE/COMPARE/PWM TIMINGS (CCP)**




**Note:** Refer to Figure 26-4 for load conditions.

**TABLE 26-13: CAPTURE/COMPARE/PWM REQUIREMENTS (CCP)**

| **Standard Operating Conditions (unless otherwise stated)**<br>Operating Temperature -40°C ≤ TA ≤ +125°C | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Param No.** | **Sym.** | **Characteristic** | | **Min.** | **Typ†** | **Max.** | **Units** | **Conditions** |
| CC01* | TccL | CCPx Input Low Time | No Prescaler | 0.5TCY + 20 | — | — | ns | |
| | | | With Prescaler | 20 | — | — | ns | |
| CC02* | TccH | CCPx Input High Time | No Prescaler | 0.5TCY + 20 | — | — | ns | |
| | | | With Prescaler | 20 | — | — | ns | |
| CC03* | TccP | CCPx Input Period | | (3TCY + 40)*N | — | — | ns | N = prescale value |

* These parameters are characterized but not tested.

† Data in "Typ" column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**TABLE 26-14: ANALOG-TO-DIGITAL CONVERTER (ADC) CHARACTERISTICS(1,2,3)**

| **Operating Conditions (unless otherwise stated)**<br>VDD = 3.0V, TA = 25°C, Single-ended, 2 μs TAD, VREF+ = 3V, VREF- = VSS | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Sym.** | **Characteristic** | **Min.** | **Typ†** | **Max.** | **Units** | **Conditions** |
| AD01 | NR | Resolution | — | — | 10 | bit | |
| AD02 | EIL | Integral Error | — | — | ±1.7 | LSb | VREF = 3.0V |
| AD03 | EDL | Differential Error | — | — | ±1 | LSb | No missing codes, VREF = 3.0V |
| AD04 | EOFF | Offset Error | — | — | ±2.5 | LSb | VREF = 3.0V |
| AD05 | EGN | Gain Error | — | — | ±2.0 | LSb | VREF = 3.0V |
| AD06 | VREF | Reference Voltage | 1.8 | — | VDD | V | VREF = (VREF+ minus VREF-) |
| AD07 | VAIN | Full-Scale Range | VSS | — | VREF | V | |
| AD08 | ZAIN | Recommended Impedance of Analog Voltage Source | — | — | 10 | kΩ | Can go higher if external 0.01μF capacitor is present on input pin. |

* These parameters are characterized but not tested.

† Data in "Typ" column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** The ADC conversion result never decreases with an increase in the input voltage and has no missing codes.

**2:** ADC VREF is from external VREF+ pin, VDD pin or FVR, whichever is selected as reference input.

**3:** See **Section 27.0 "DC and AC Characteristics Graphs and Charts"** for operating characterization.


© 2013-2015 Microchip Technology Inc.

**TABLE 26-15: ADC CONVERSION REQUIREMENTS**

| Standard Operating Conditions (unless otherwise stated) | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Sym.** | **Characteristic** | **Min.** | **Typ†** | **Max.** | **Units** | **Conditions** |
| AD130* | TAD | ADC Clock Period (TADC) | 1.0 | — | 9.0 | µs | FOSC-based |
| | | ADC Internal FRC Oscillator Period (TFRC) | 1.0 | 2 | 6.0 | µs | ADCS<1:0> = 11 (ADC FRC mode) |
| AD131 | TCNV | Conversion Time (not including Acquisition Time)(1) | — | 11 | — | TAD | Set GO/$\overline{\text{DONE}}$ bit to conversion complete |
| AD132* | TACQ | Acquisition Time | — | 5.0 | — | µs | |
| AD133* | THCD | Holding Capacitor Disconnect Time | — | 1/2 TAD | — | | ADCS<2:0> ≠ x11 (FOSC based) |
| | | | — | 1/2 TAD + 1TCY | — | | ADCS<2:0> = x11 (FRC based) |

* These parameters are characterized but not tested.

† Data in "Typ" column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** The ADRES register may be read on the following TCY cycle.

**FIGURE 26-12: ADC CONVERSION TIMING (ADC CLOCK FOSC-BASED)**

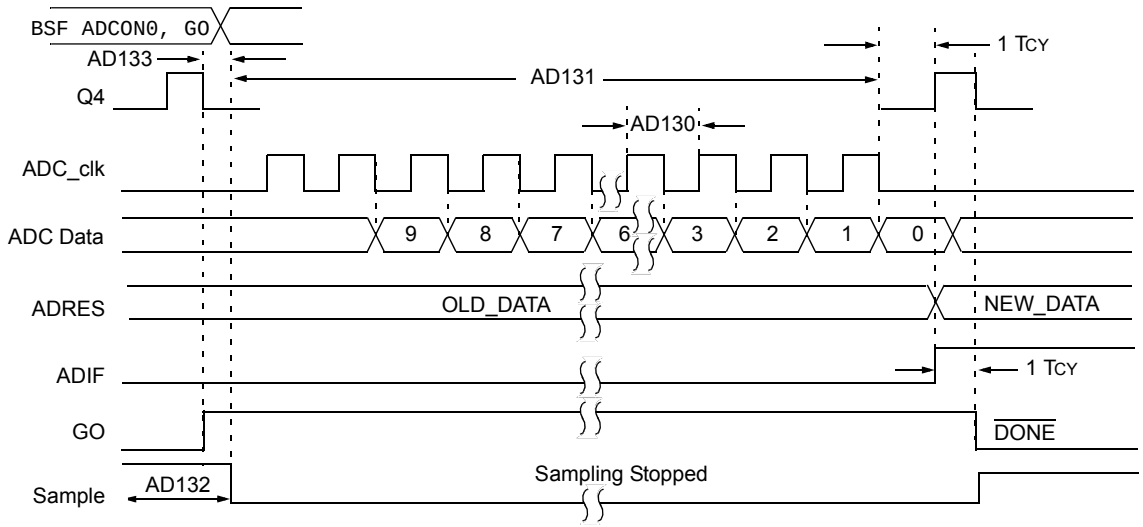
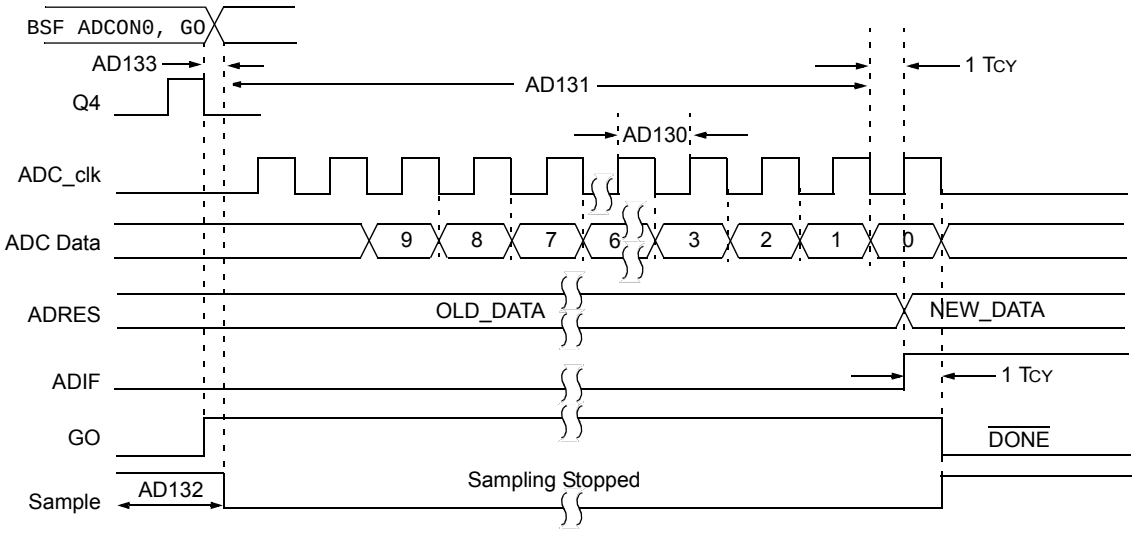



**FIGURE 26-13: ADC CONVERSION TIMING (ADC CLOCK FROM FRC)**




**Note 1:** If the ADC clock source is selected as FRC, a time of TCY is added before the ADC clock starts. This allows the SLEEP instruction to be executed.


© 2013-2015 Microchip Technology Inc.

**TABLE 26-16: OPERATIONAL AMPLIFIER (OPA)**

| **Operating Conditions (unless otherwise stated)**<br>$V_{DD}$ = 3.0V, $T_A$ = 25°C, OPAxSP = 1 (High GBWP mode) | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Symbol** | **Parameters** | **Min.** | **Typ.** | **Max.** | **Units** | **Conditions** |
| OPA01* | GBWP | Gain Bandwidth Product | — | 3.5 | — | MHz | |
| OPA02* | TON | Turn-on Time | — | 10 | — | μs | |
| OPA03* | PM | Phase Margin | — | 40 | — | degrees | |
| OPA04* | SR | Slew Rate | — | 3 | — | V/μs | |
| OPA05 | OFF | Offset | — | ±3 | ±9 | mV | |
| OPA06 | CMRR | Common-Mode Rejection Ratio | 52 | 70 | — | dB | |
| OPA07* | AOL | Open Loop Gain | — | 90 | — | dB | |
| OPA08 | VICM | Input Common-Mode Voltage | 0 | — | VDD | V | VDD > 2.5V |
| OPA09* | PSRR | Power Supply Rejection Ratio | — | 80 | — | dB | |

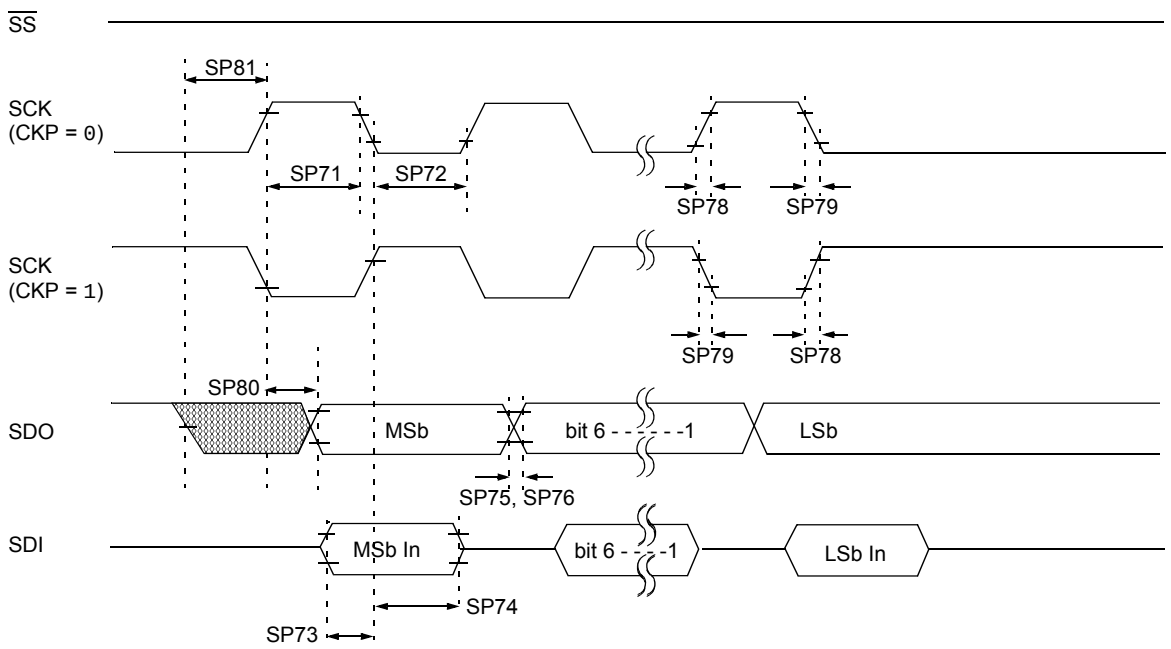
* These parameters are characterized but not tested.

**TABLE 26-17: ZERO CROSS PIN SPECIFICATIONS**

| **Operating Conditions (unless otherwise stated)**<br>$V_{DD}$ = 3.0V, $T_A$ = 25°C | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Param. No.** | **Sym.** | **Characteristics** | **Min.** | **Typ.** | **Max.** | **Units** | **Comments** |
| ZC01 | ZCPINV | Voltage on Zero Cross Pin | — | 0.75 | — | V | |
| ZC02 | ZCSRC | Source current | — | — | -600 | μA | |
| ZC03 | ZCSNK | Sink current | 600 | — | — | μA | |
| ZC04 | ZCISW | Response Time Rising Edge | — | 1 | — | μs | |
| | | Response Time Falling Edge | — | 1 | — | μs | |
| ZC05 | ZCOUT | Response Time Rising Edge | — | 1 | — | μs | |
| | | Response Time Falling Edge | — | 1 | — | μs | |

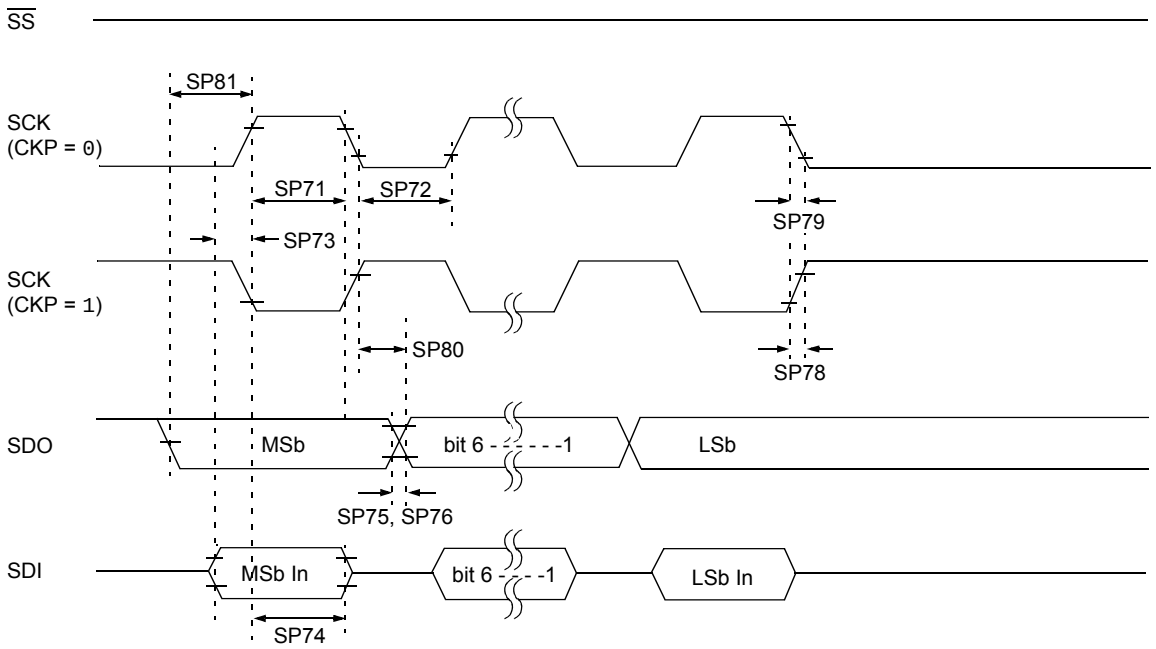
* These parameters are characterized but not tested.

**FIGURE 26-14: SPI MASTER MODE TIMING (CKE = 0, SMP = 0)**




**Note:** Refer to Figure 26-4 for load conditions.

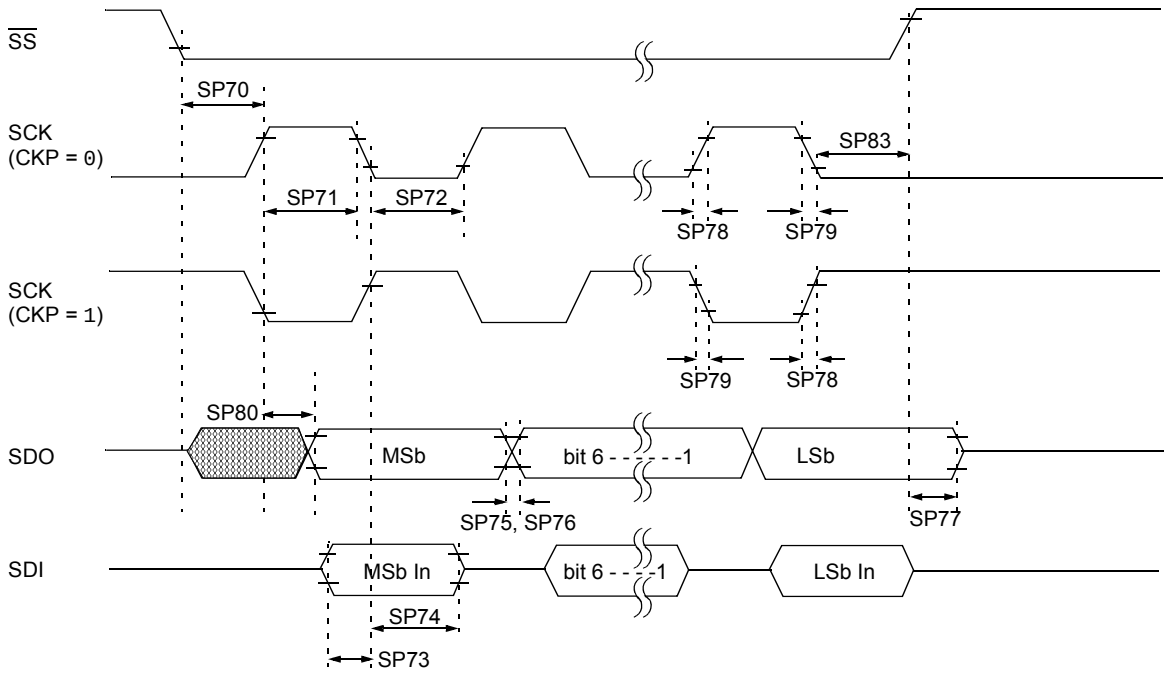
**FIGURE 26-15: SPI MASTER MODE TIMING (CKE = 1, SMP = 1)**




**Note:** Refer to Figure 26-4 for load conditions.

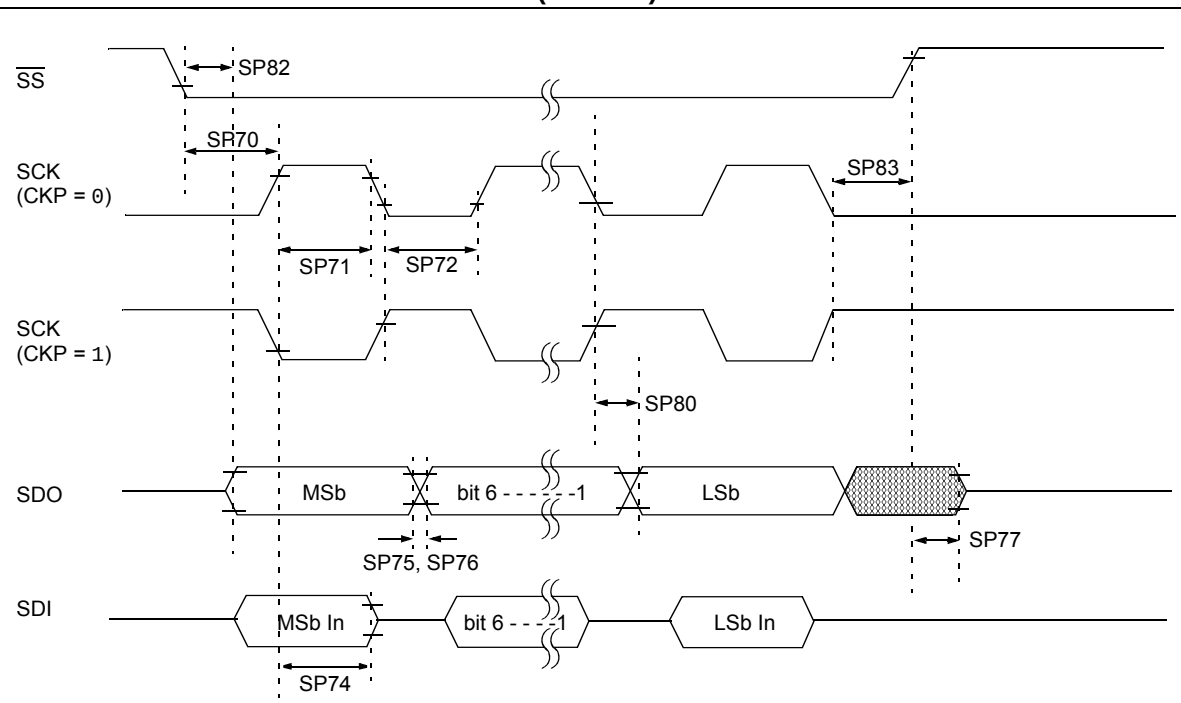

© 2013-2015 Microchip Technology Inc.

**FIGURE 26-16: SPI SLAVE MODE TIMING (CKE = 0)**




**Note:** Refer to Figure 26-4 for load conditions.

**FIGURE 26-17: SPI SLAVE MODE TIMING (CKE = 1)**




**Note:** Refer to Figure 26-4 for load conditions.

**TABLE 26-18: SPI MODE REQUIREMENTS**

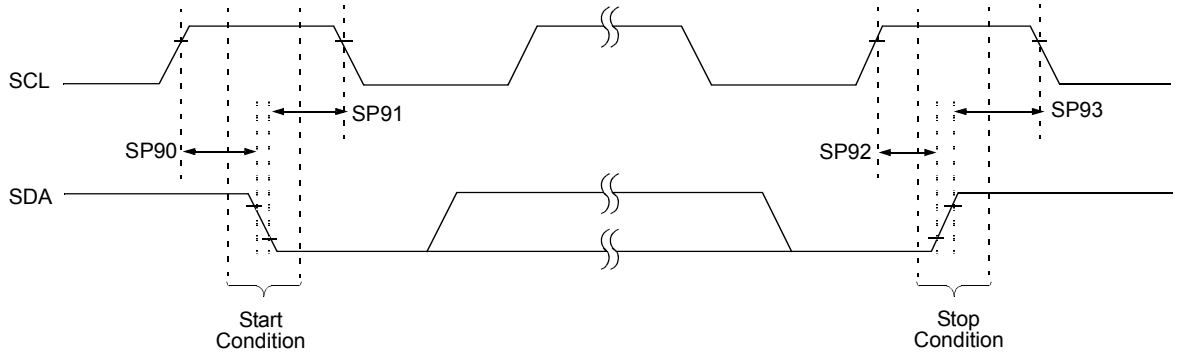
| Standard Operating Conditions (unless otherwise stated) | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Symbol** | **Characteristic** | **Min.** | **Typ†** | **Max.** | **Units** | **Conditions** |
| SP70* | TssL2scH, TssL2scL | $\overline{SS}$↓ to SCK↓ or SCK↑ input | 2.25 Tcy | — | — | ns | |
| SP71* | TscH | SCK input high time (Slave mode) | Tcy + 20 | — | — | ns | |
| SP72* | TscL | SCK input low time (Slave mode) | Tcy + 20 | — | — | ns | |
| SP73* | TdiV2scH, TdiV2scL | Setup time of SDI data input to SCK edge | 100 | — | — | ns | |
| SP74* | TscH2diL, TscL2diL | Hold time of SDI data input to SCK edge | 100 | — | — | ns | |
| SP75* | TdoR | SDO data output rise time | — | 10 | 25 | ns | 3.0V ≤ VDD ≤ 5.5V |
| | | | — | 25 | 50 | ns | 1.8V ≤ VDD ≤ 5.5V |
| SP76* | TdoF | SDO data output fall time | — | 10 | 25 | ns | |
| SP77* | TssH2doZ | $\overline{SS}$↑ to SDO output high-impedance | 10 | — | 50 | ns | |
| SP78* | TscR | SCK output rise time (Master mode) | — | 10 | 25 | ns | 3.0V ≤ VDD ≤ 5.5V |
| | | | — | 25 | 50 | ns | 1.8V ≤ VDD ≤ 5.5V |
| SP79* | TscF | SCK output fall time (Master mode) | — | 10 | 25 | ns | |
| SP80* | TscH2doV, TscL2doV | SDO data output valid after SCK edge | — | — | 50 | ns | 3.0V ≤ VDD ≤ 5.5V |
| | | | — | — | 145 | ns | 1.8V ≤ VDD ≤ 5.5V |
| SP81* | TdoV2scH, TdoV2scL | SDO data output setup to SCK edge | 1 Tcy | — | — | ns | |
| SP82* | TssL2doV | SDO data output valid after $\overline{SS}$↓ edge | — | — | 50 | ns | |
| SP83* | TscH2ssH, TscL2ssH | $\overline{SS}$ ↑ after SCK edge | 1.5 Tcy + 40 | — | — | ns | |

* These parameters are characterized but not tested.

† Data in "Typ" column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.


© 2013-2015 Microchip Technology Inc.

**FIGURE 26-18: I²C BUS START/STOP BITS TIMING**



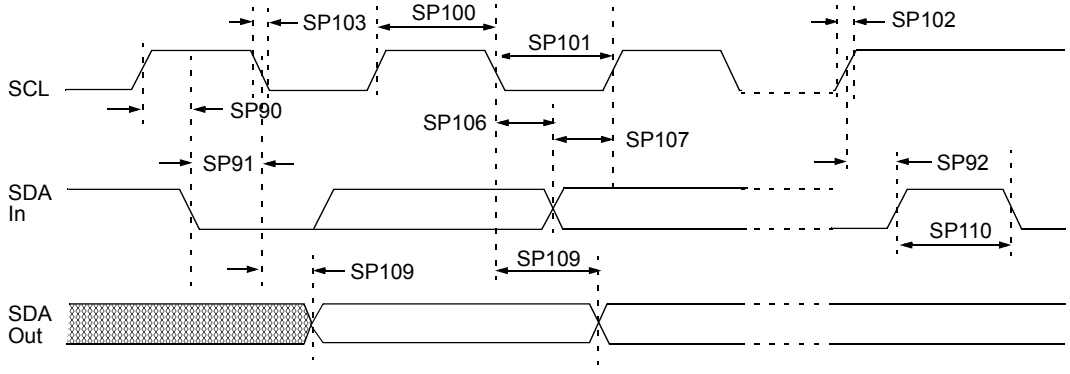

**Note**: Refer to Figure 26-4 for load conditions.

**TABLE 26-19: I²C BUS START/STOP BITS REQUIREMENTS**

| Standard Operating Conditions (unless otherwise stated) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Param No.** | **Symbol** | **Characteristic** | | **Min.** | **Typ** | **Max.** | **Units** | **Conditions** |
| SP90* | TSU:STA | Start condition Setup time | 100 kHz mode | 4700 | — | — | ns | Only relevant for Repeated Start condition |
| | | | 400 kHz mode | 600 | — | — | | |
| SP91* | THD:STA | Start condition Hold time | 100 kHz mode | 4000 | — | — | ns | After this period, the first clock pulse is generated |
| | | | 400 kHz mode | 600 | — | — | | |
| SP92* | TSU:STO | Stop condition Setup time | 100 kHz mode | 4700 | — | — | ns | |
| | | | 400 kHz mode | 600 | — | — | | |
| SP93* | THD:STO | Stop condition Hold time | 100 kHz mode | 4000 | — | — | ns | |
| | | | 400 kHz mode | 600 | — | — | | |

* These parameters are characterized but not tested.

**FIGURE 26-19: I²C BUS DATA TIMING**




**Note:** Refer to Figure 26-4 for load conditions.

**TABLE 26-20: I²C BUS DATA REQUIREMENTS**

| Standard Operating Conditions (unless otherwise stated) | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Param. No.** | **Symbol** | **Characteristic** | | **Min.** | **Max.** | **Units** | **Conditions** |
| SP100* | 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 | — | | |
| SP101* | 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 | — | | |
| SP102* | 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 |
| SP103* | TF | SDA and SCL fall time | 100 kHz mode | — | 250 | ns | |
| | | | 400 kHz mode | 20 + 0.1CB | 250 | ns | CB is specified to be from 10-400 pF |
| SP106* | THD:DAT | Data input hold time | 100 kHz mode | 0 | — | ns | |
| | | | 400 kHz mode | 0 | 0.9 | µs | |
| SP107* | TSU:DAT | Data input setup time | 100 kHz mode | 250 | — | ns | **(Note 2)** |
| | | | 400 kHz mode | 100 | — | ns | |
| SP109* | TAA | Output valid from clock | 100 kHz mode | — | 3500 | ns | **(Note 1)** |
| | | | 400 kHz mode | — | — | ns | |
| SP110* | 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 | |
| SP111* | 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.


© 2013-2015 Microchip Technology Inc.

## 27.0 DC AND AC CHARACTERISTICS GRAPHS AND CHARTS

The graphs and tables provided in this section are for **design guidance** and are **not tested**.

In some graphs or tables, the data presented are **outside specified operating range** (i.e., outside specified $V_{DD}$ range). This is for **information only** and devices are ensured to operate properly only within the specified range.

Unless otherwise noted, all graphs apply to both the L and LF devices.

| | |
|---|---|
| **Note:** | The graphs and tables provided following this note are a statistical summary based on a limited number of samples and are provided for informational purposes only. The performance characteristics listed herein are not tested or guaranteed. In some graphs or tables, the data presented may be outside the specified operating range (e.g., outside specified power supply range) and therefore, outside the warranted range. |

**"Typical" represents the mean of the distribution at 25°C. "MAXIMUM", "Max.", "MINIMUM" or "Min." represents (mean + 3σ) or (mean - 3σ) respectively, where σ is a standard deviation, over each temperature range.**

**Note:** Unless otherwise noted, VIN = 5V, FOSC = 500 kHz, CIN = 0.1 µF, TA = 25°C.

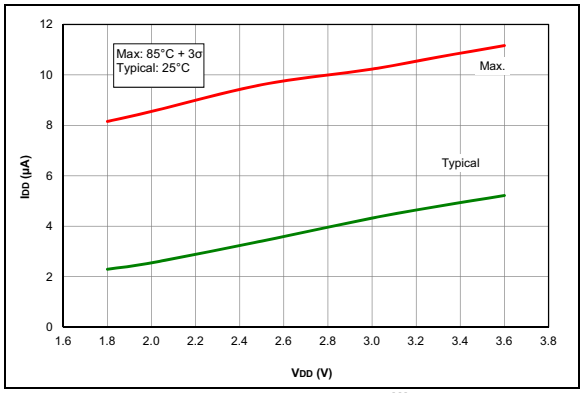



***FIGURE 27-1:*** *IDD, EC Oscillator, Low-Power Mode, FOSC = 32 kHz, PIC16LF1703/7 Only.*

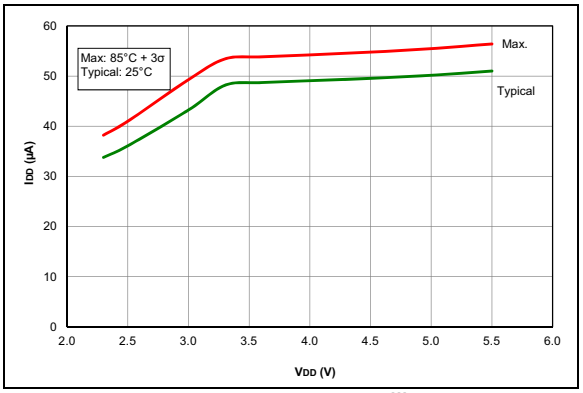



***FIGURE 27-4:*** *IDD, EC Oscillator, Low-Power Mode, FOSC = 500 kHz, PIC16F1703/7 Only.*

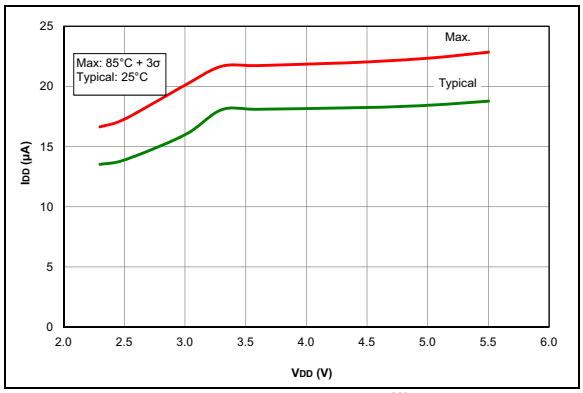



***FIGURE 27-2:*** *IDD, EC Oscillator, Low-Power Mode, FOSC = 32 kHz, PIC16F1703/7 Only.*

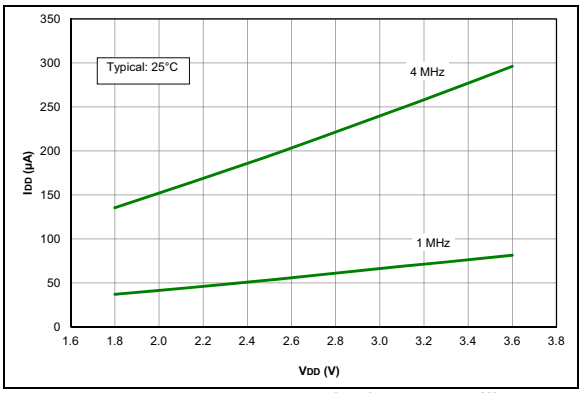



***FIGURE 27-5:*** *IDD Typical, EC Oscillator, Medium-Power Mode, PIC16LF1703/7 Only.*

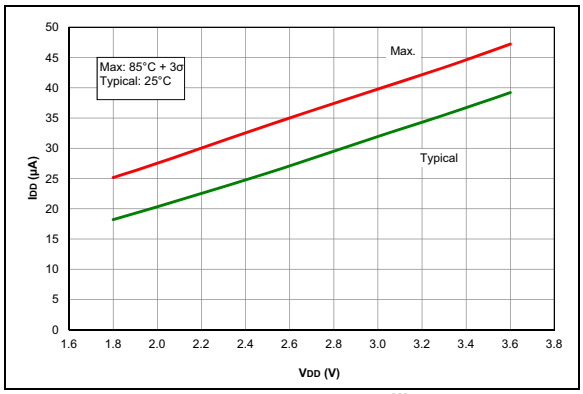



***FIGURE 27-3:*** *IDD, EC Oscillator, Low-Power Mode, FOSC = 500 kHz, PIC16LF1703/7 Only.*

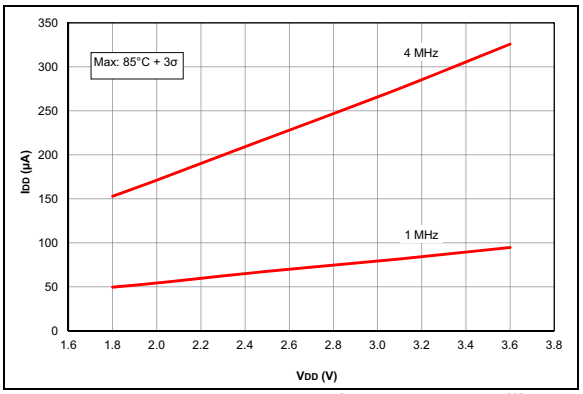



***FIGURE 27-6:*** *IDD Maximum, EC Oscillator, Medium-Power Mode, PIC16LF1703/7 Only.*


© 2013-2015 Microchip Technology Inc.

**Note:** Unless otherwise noted, VIN = 5V, FOSC = 500 kHz, CIN = 0.1 µF, TA = 25°C.

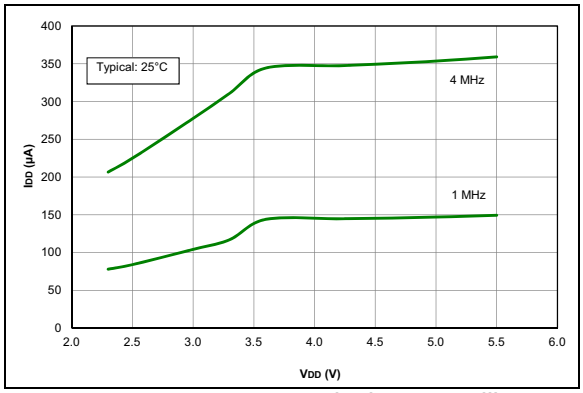



***FIGURE 27-7:*** *IDD Typical, EC Oscillator, Medium-Power Mode, PIC16F1703/7 Only.*

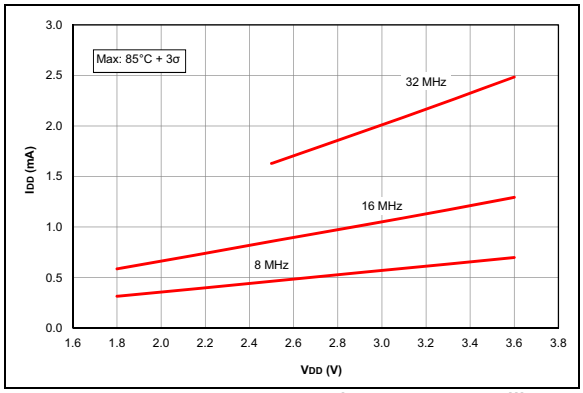



***FIGURE 27-10:*** *IDD Maximum, EC Oscillator, High-Power Mode, PIC16LF1703/7 Only.*

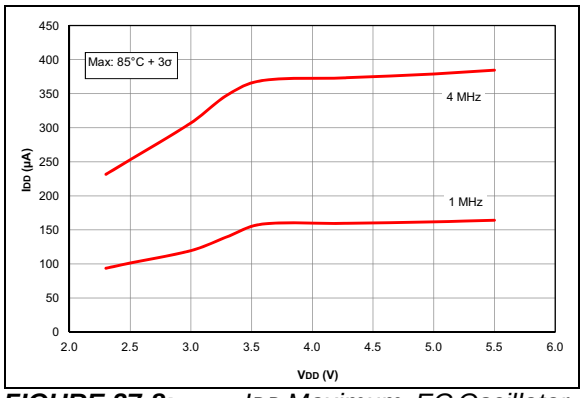



***FIGURE 27-8:*** *IDD Maximum, EC Oscillator, Medium-Power Mode, PIC16F1703/7 Only.*

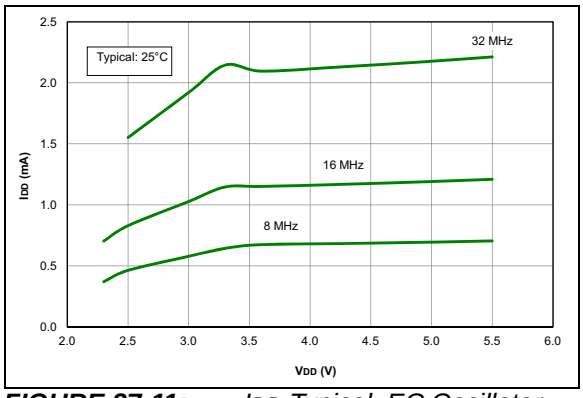



***FIGURE 27-11:*** *IDD Typical, EC Oscillator, High-Power Mode, PIC16F1703/7 Only.*

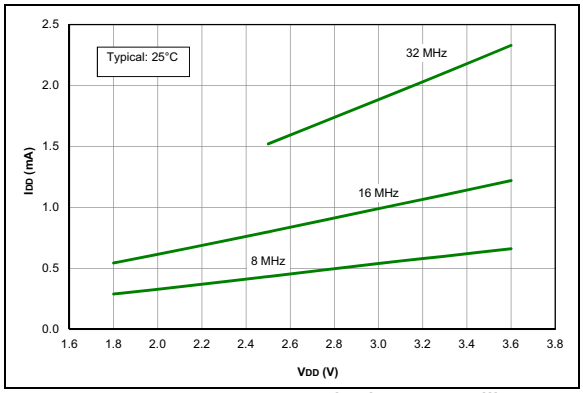



***FIGURE 27-9:*** *IDD Typical, EC Oscillator, High-Power Mode, PIC16LF1703/7 Only.*

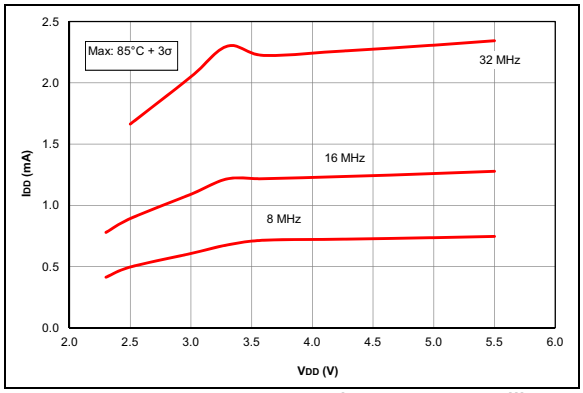



***FIGURE 27-12:*** *IDD Maximum, EC Oscillator, High-Power Mode, PIC16F1703/7 Only.*

**Note:** Unless otherwise noted, VIN = 5V, FOSC = 500 kHz, CIN = 0.1 µF, TA = 25°C.

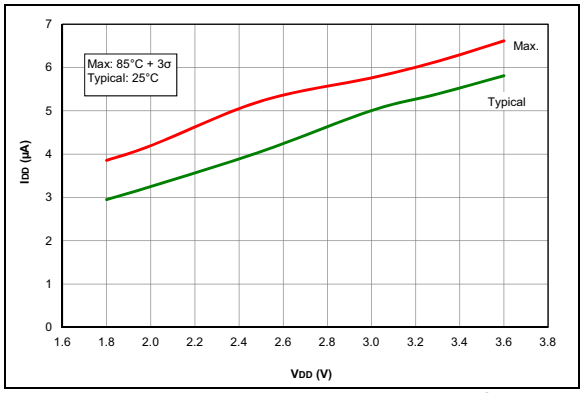



***FIGURE 27-13:*** *IDD, LFINTOSC Mode, Fosc = 31 kHz, PIC16LF1703/7 Only.*

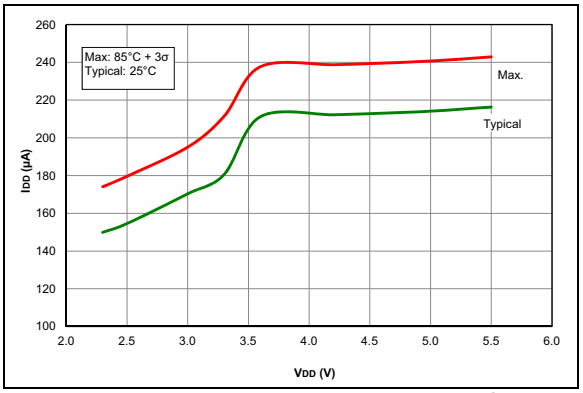



***FIGURE 27-16:*** *IDD, MFINTOSC Mode, Fosc = 500 kHz, PIC16F1703/7 Only.*

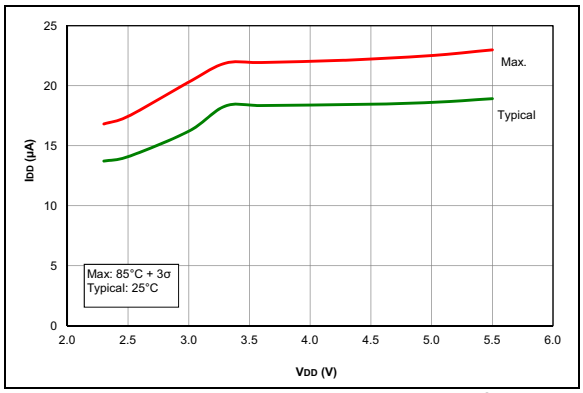



***FIGURE 27-14:*** *IDD, LFINTOSC Mode, Fosc = 31 kHz, PIC16F1703/7 Only.*

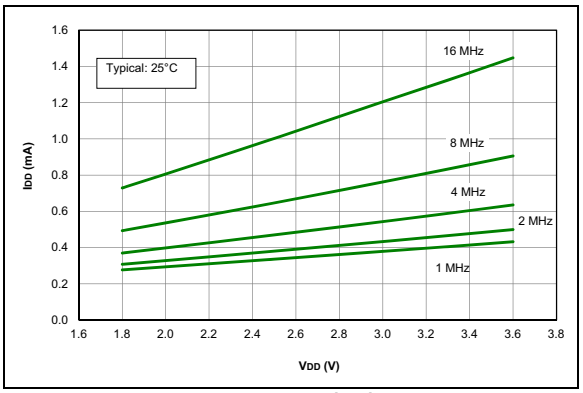



***FIGURE 27-17:*** *IDD Typical, HFINTOSC Mode, PIC16LF1703/7 Only.*

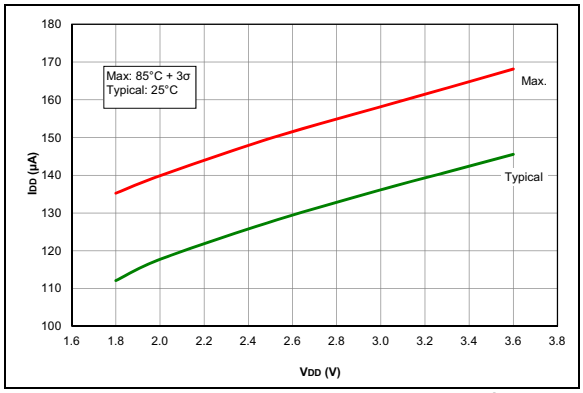



***FIGURE 27-15:*** *IDD, MFINTOSC Mode, Fosc = 500 kHz, PIC16LF1703/7 Only.*

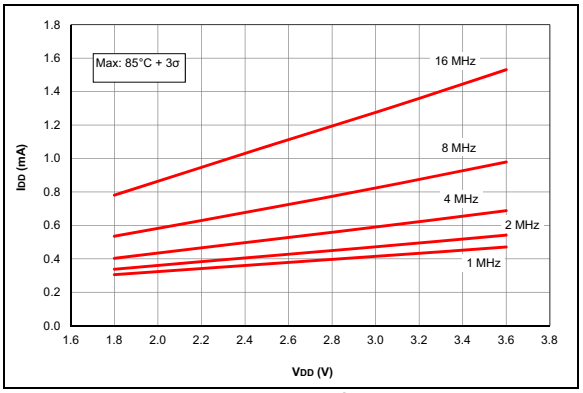



***FIGURE 27-18:*** *IDD Maximum, HFINTOSC Mode, PIC16LF1703/7 Only.*


© 2013-2015 Microchip Technology Inc.

**Note:** Unless otherwise noted, VIN = 5V, FOSC = 500 kHz, CIN = 0.1 µF, TA = 25°C.

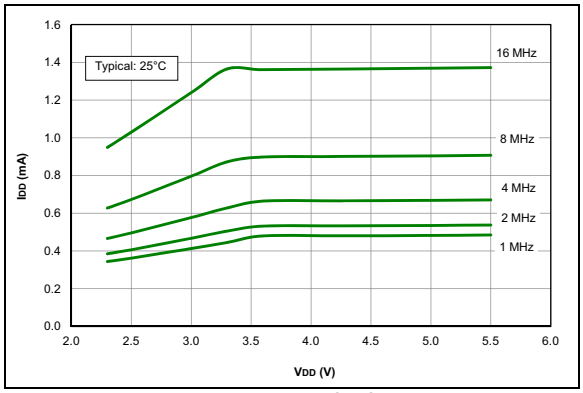



***FIGURE 27-19:*** *IDD Typical, HFINTOSC Mode, PIC16F1703/7 Only.*

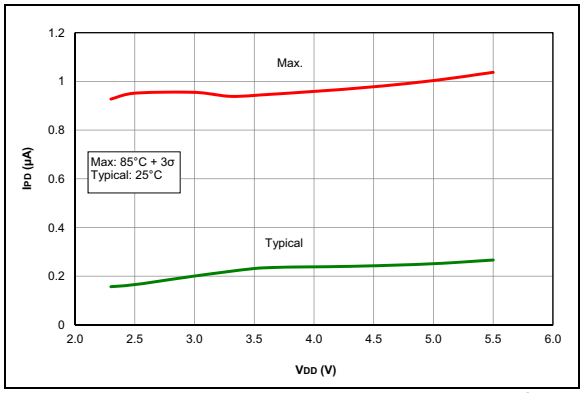



***FIGURE 27-22:*** *IPD Base, Low-Power Sleep Mode (VREGPM = 1), PIC16F1703/7 Only.*

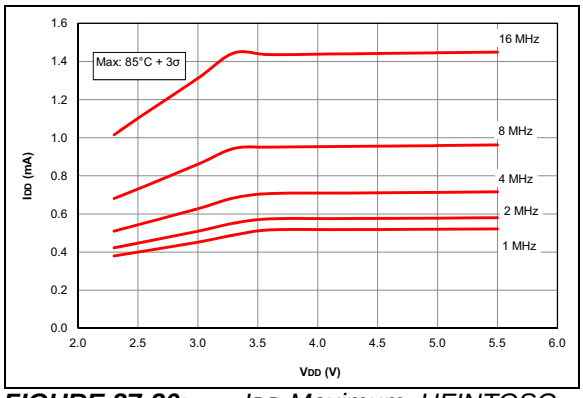



***FIGURE 27-20:*** *IDD Maximum, HFINTOSC Mode, PIC16F1703/7 Only.*

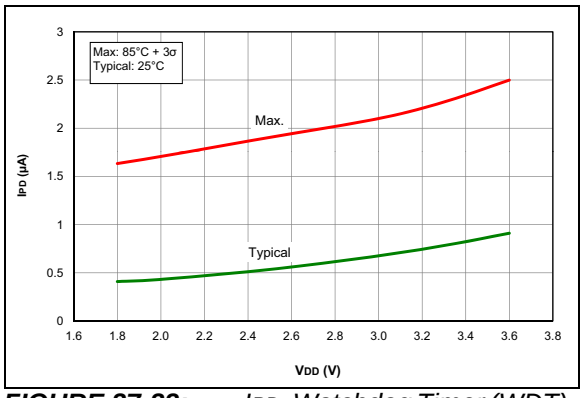



***FIGURE 27-23:*** *IPD, Watchdog Timer (WDT), PIC16LF1703/7 only.*

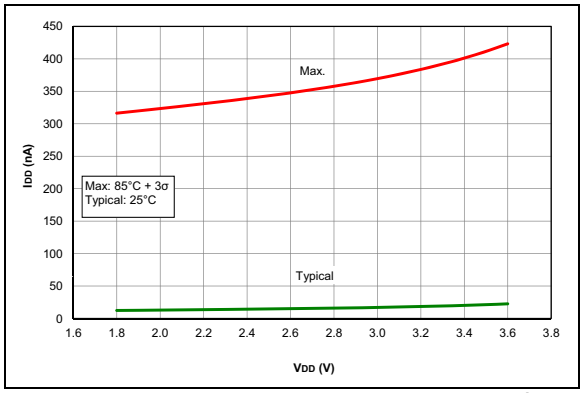



***FIGURE 27-21:*** *IPD Base, Low-Power Sleep Mode, PIC16LF1703/7 Only.*

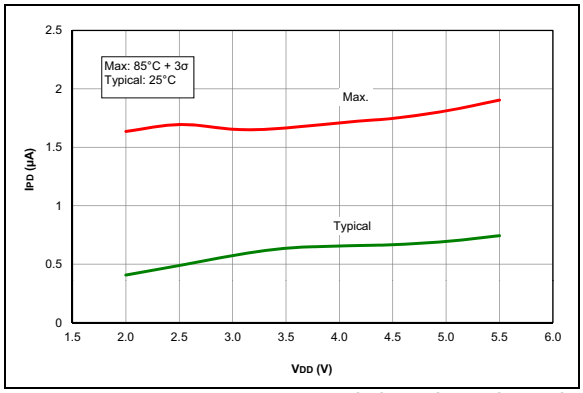



***FIGURE 27-24:*** *IPD, Watchdog Timer (WDT), PIC16F1703/7 only.*

**Note:** Unless otherwise noted, VIN = 5V, FOSC = 500 kHz, CIN = 0.1 µF, TA = 25°C.

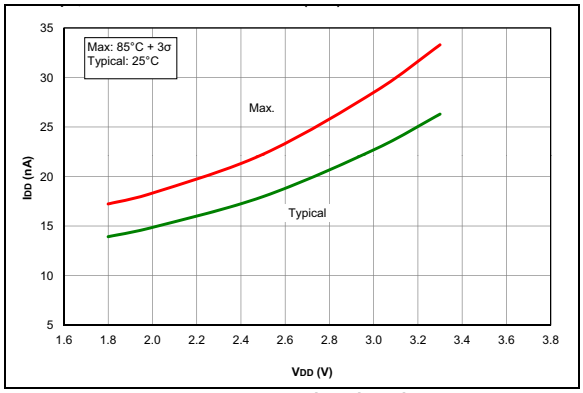



***FIGURE 27-25:*** *IPD, Fixed Voltage Reference (FVR), PIC16LF1703/7 only.*

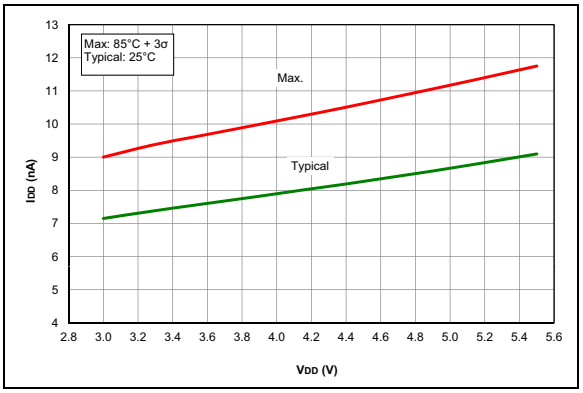



***FIGURE 27-28:*** *IPD, Brown-Out Reset (BOR), BORV = 1, PIC16F1703/7 only.*

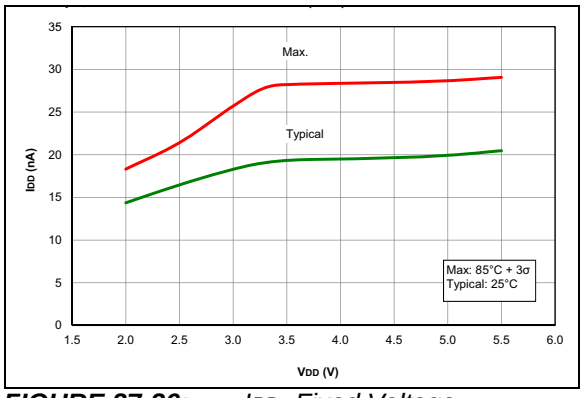



***FIGURE 27-26:*** *IPD, Fixed Voltage Reference (FVR), PIC16F1703/7 only.*

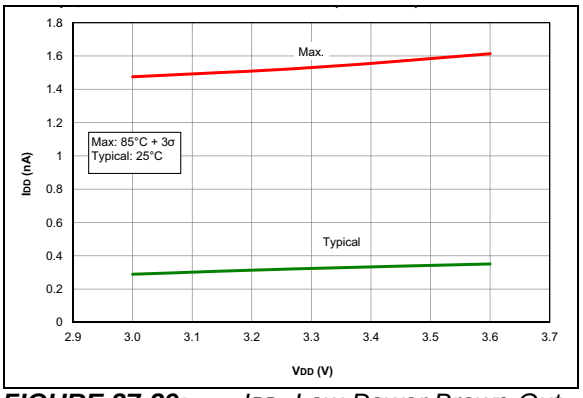



***FIGURE 27-29:*** *IPD, Low-Power Brown-Out Reset (BOR), LPBOR = 0, PIC16LF1703/7 only.*

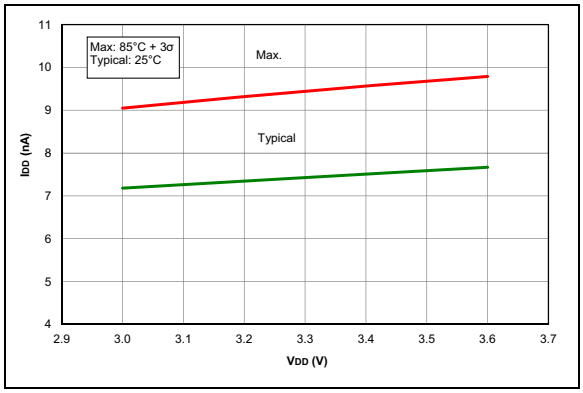



***FIGURE 27-27:*** *IPD, Brown-Out Reset (BOR), BORV = 1, PIC16LF1703/7 only.*

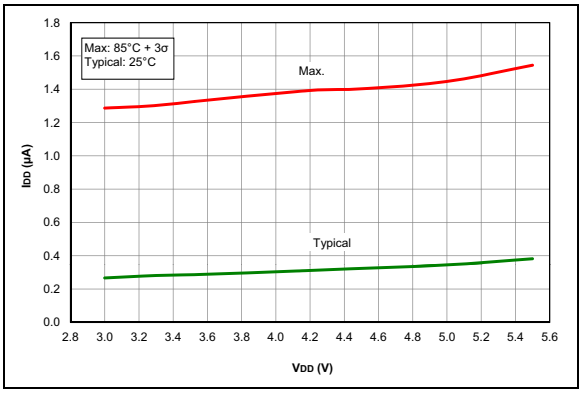



***FIGURE 27-30:*** *IPD, Low-Power Brown-Out Reset (BOR), LPBOR = 0, PIC16F1703/7 only.*


© 2013-2015 Microchip Technology Inc.

**Note:** Unless otherwise noted, VIN = 5V, FOSC = 500 kHz, CIN = 0.1 µF, TA = 25°C.

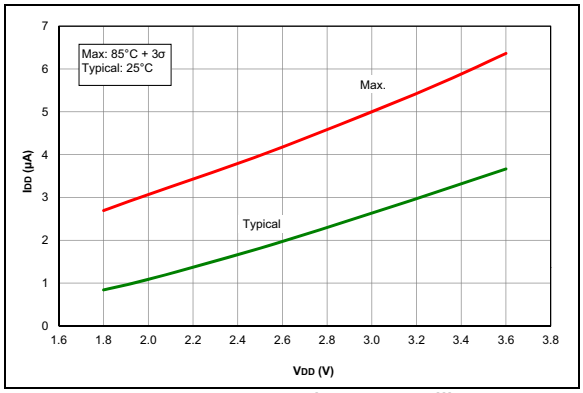



***FIGURE 27-31:*** *IPD, Timer1 Oscillator, Fosc = 32 kHz, PIC16LF1703/7 Only.*

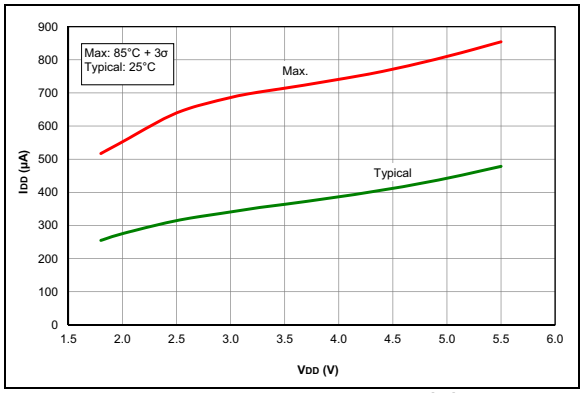



***FIGURE 27-34:*** *IPD, Op Amp, High GBWP Mode (OPAxSP = 1), PIC16F1703/7 Only.*

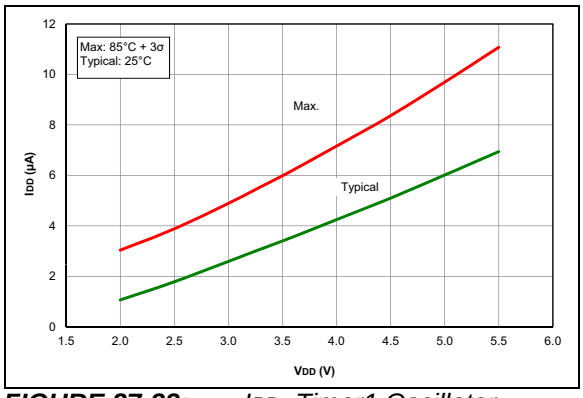



***FIGURE 27-32:*** *IPD, Timer1 Oscillator, Fosc = 32 kHz, PIC16F1703/7 Only.*

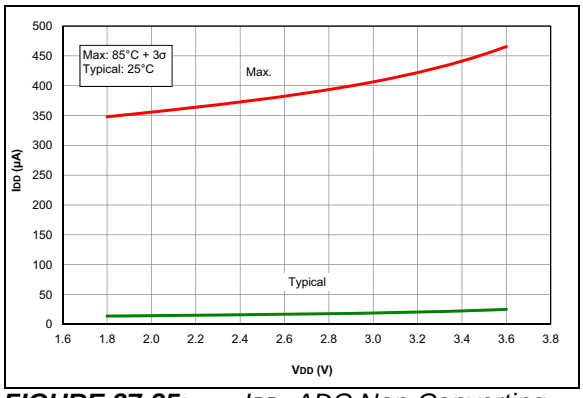



***FIGURE 27-35:*** *IPD, ADC Non-Converting, PIC16LF1703/7 only*

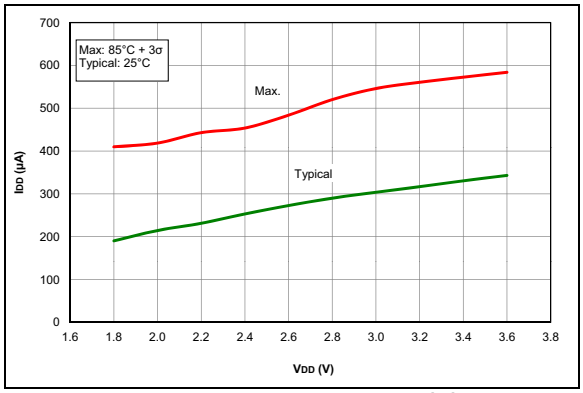



***FIGURE 27-33:*** *IPD, Op Amp, High GBWP Mode (OPAxSP = 1), PIC16LF1703/7 Only.*

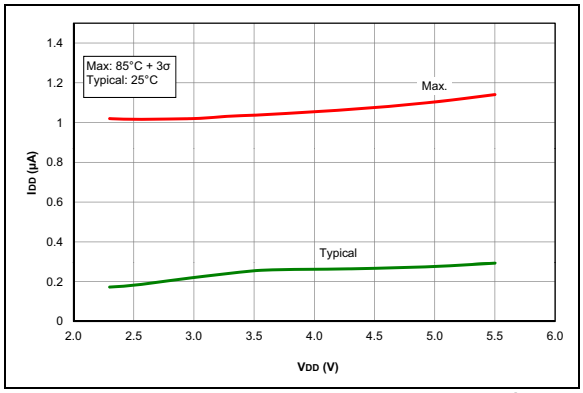



***FIGURE 27-36:*** *IPD, ADC Non-Converting, PIC16F1703/7 only*

**Note:** Unless otherwise noted, $V_{IN}$ = 5V, $F_{OSC}$ = 500 kHz, $C_{IN}$ = 0.1 µF, $T_A$ = 25°C.

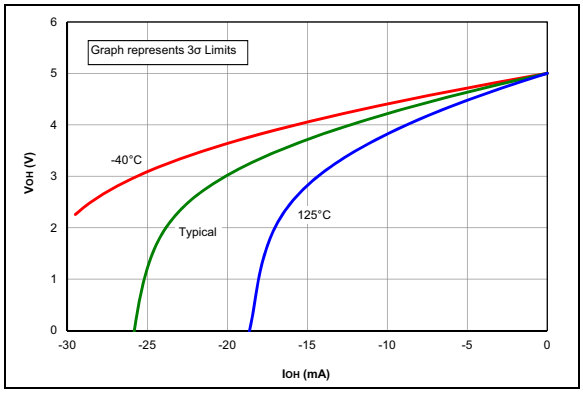



***FIGURE 27-37:*** *$V_{OH}$ vs. $I_{OH}$ Over Temperature, $V_{DD}$ = 5.0V, PIC16F1703/7 Only.*

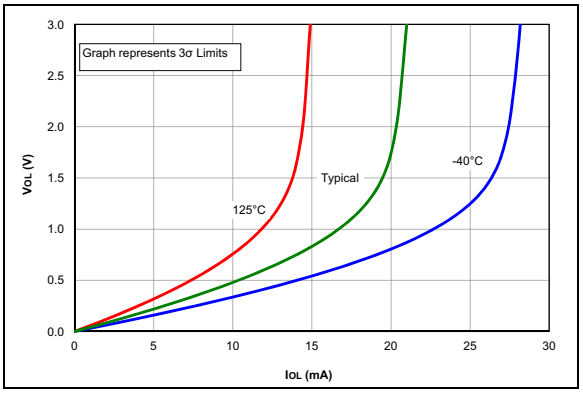



***FIGURE 27-40:*** *$V_{OL}$ vs. $I_{OL}$ Over Temperature, $V_{DD}$ = 3.0V.*

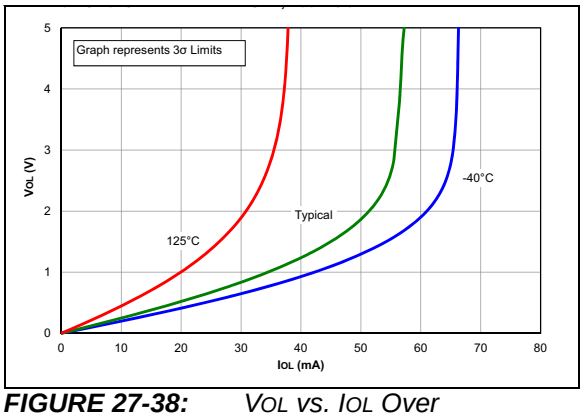



***FIGURE 27-38:*** *$V_{OL}$ vs. $I_{OL}$ Over Temperature, $V_{DD}$ = 5.0V, PIC16F1703/7 Only.*

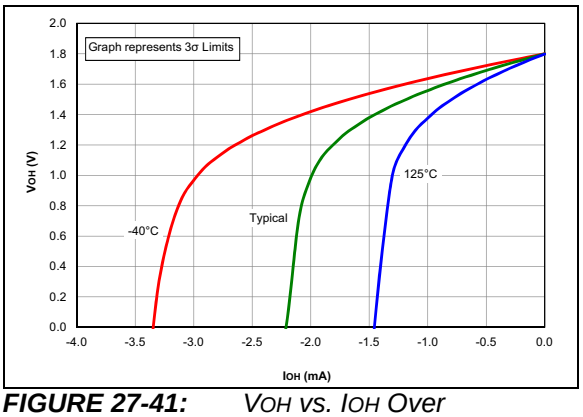



***FIGURE 27-41:*** *$V_{OH}$ vs. $I_{OH}$ Over Temperature, $V_{DD}$ = 1.8V, PIC16LF1703/7 Only.*

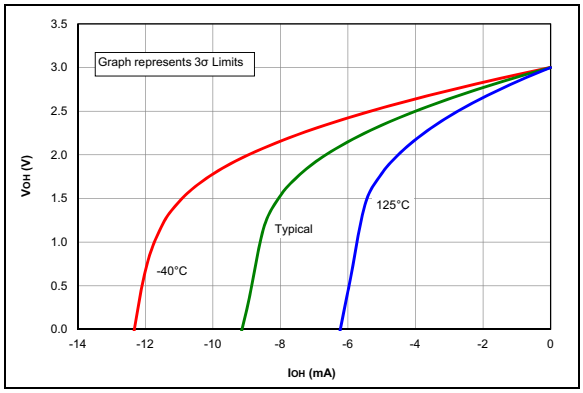



***FIGURE 27-39:*** *$V_{OH}$ vs. $I_{OH}$ Over Temperature, $V_{DD}$ = 3.0V.*

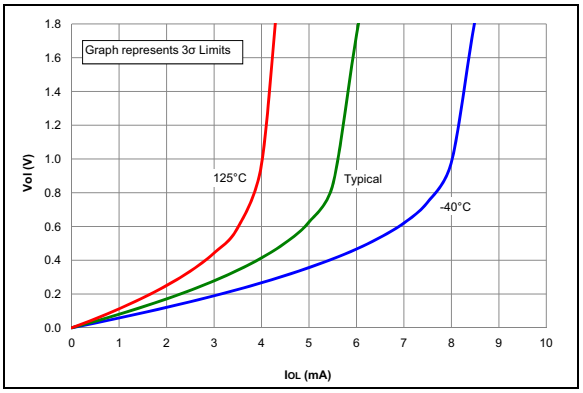



***FIGURE 27-42:*** *$V_{OL}$ vs. $I_{OL}$ Over Temperature, $V_{DD}$ = 1.8V, PIC16F1703/7 Only.*


© 2013-2015 Microchip Technology Inc.

**Note:** Unless otherwise noted, VIN = 5V, FOSC = 500 kHz, CIN = 0.1 µF, TA = 25°C.

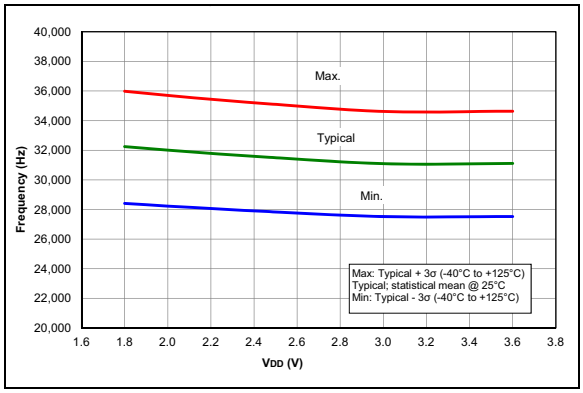



***FIGURE 27-43:*** *LFINTOSC Frequency, PIC16LF1703/7 Only.*

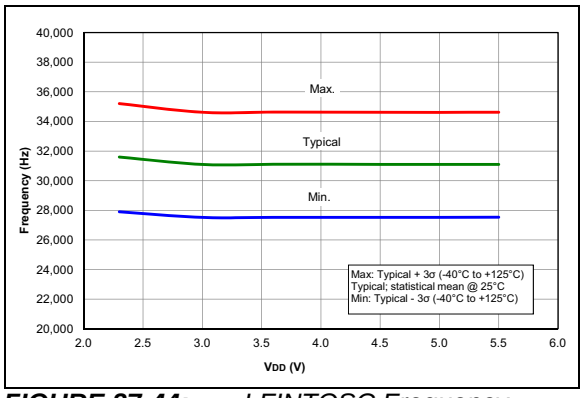



***FIGURE 27-44:*** *LFINTOSC Frequency, PIC16F1703/7 Only.*

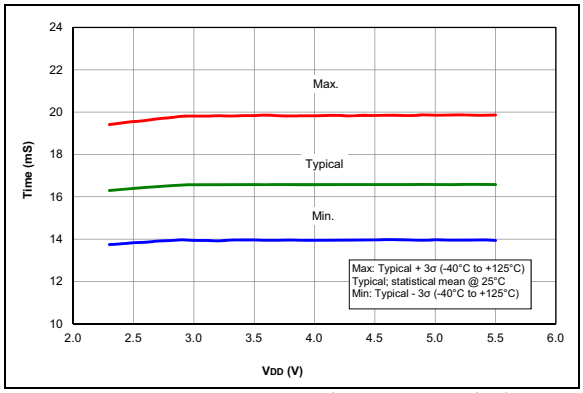



***FIGURE 27-45:*** *WDT Time-Out Period, PIC16F1703/7 Only.*

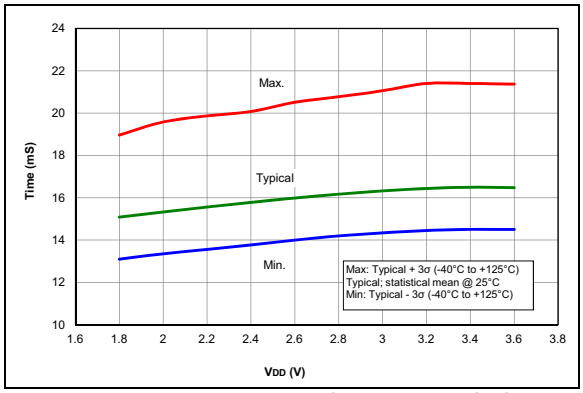



***FIGURE 27-46:*** *WDT Time-Out Period, PIC16LF1703/7 Only.*

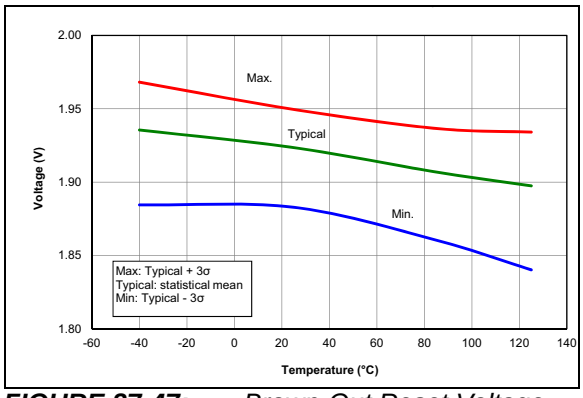



***FIGURE 27-47:*** *Brown-Out Reset Voltage, Low Trip Point (BORV = 1), PIC16LF1703/7 only.*

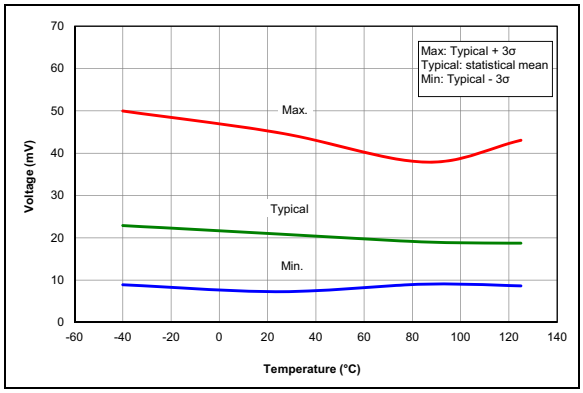



***FIGURE 27-48:*** *Brown-Out Reset Hysteresis, Low Trip Point (BORV = 1), PIC16LF1703/7 only.*

**Note:** Unless otherwise noted, $V_{IN}$ = 5V, $F_{OSC}$ = 500 kHz, $C_{IN}$ = 0.1 µF, $T_A$ = 25°C.

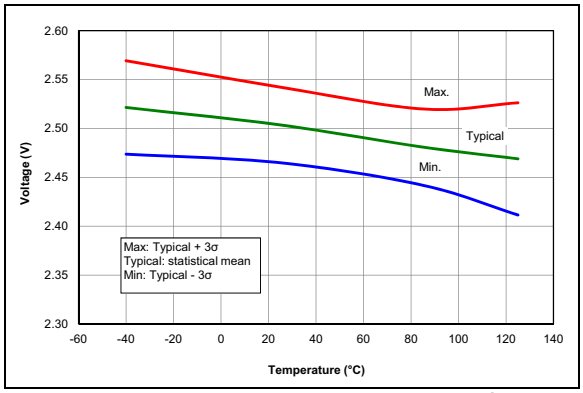



***FIGURE 27-49:*** *Brown-Out Reset Voltage, Low Trip Point (BORV = 1), PIC16F1703/7 only.*

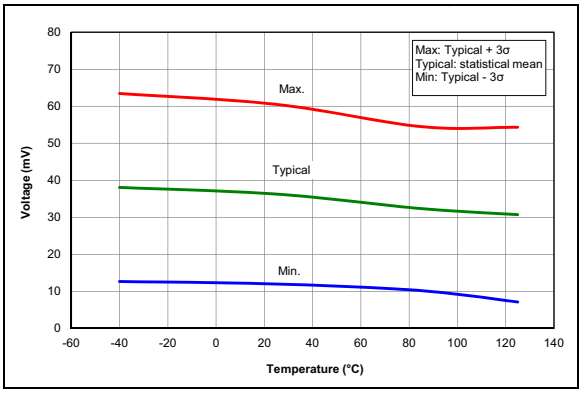



***FIGURE 27-52:*** *Brown-Out Reset Hysteresis, High Trip Point (BORV = 0).*

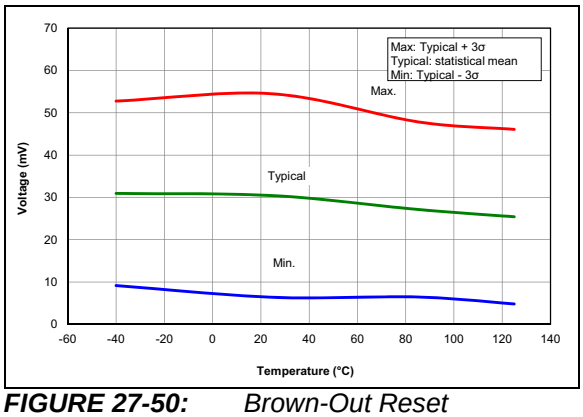



***FIGURE 27-50:*** *Brown-Out Reset Hysteresis, Low Trip Point (BORV = 1), PIC16F1703/7 only.*

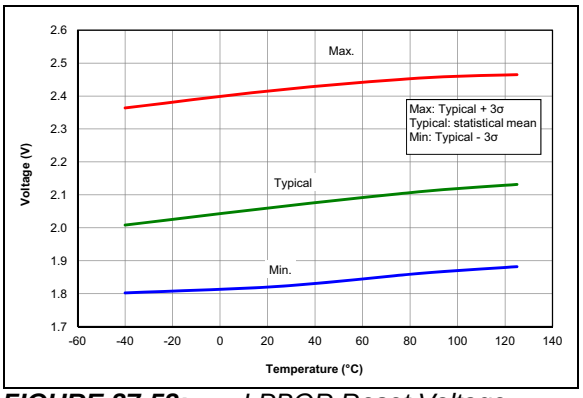



***FIGURE 27-53:*** *LPBOR Reset Voltage.*

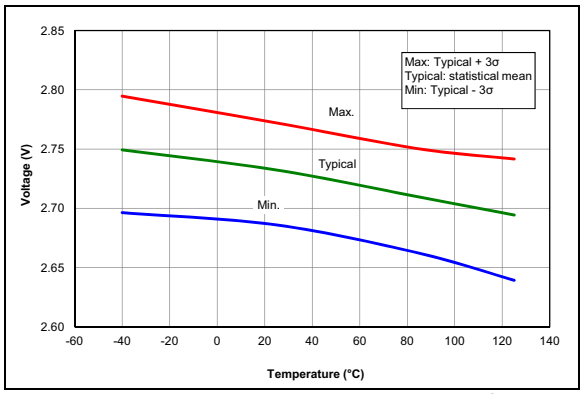



***FIGURE 27-51:*** *Brown-Out Reset Voltage, High Trip Point (BORV = 0).*

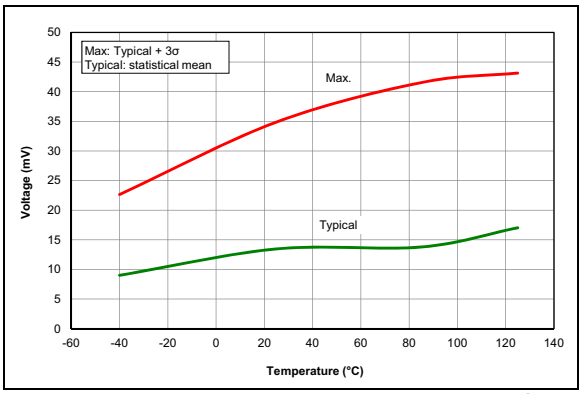



***FIGURE 27-54:*** *LPBOR Reset Hysteresis.*


© 2013-2015 Microchip Technology Inc.

**Note:** Unless otherwise noted, VIN = 5V, FOSC = 500 kHz, CIN = 0.1 µF, TA = 25°C.

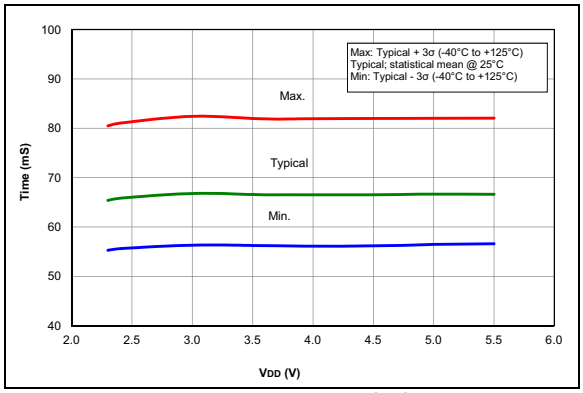



***FIGURE 27-55:*** *PWRT Period, PIC16F1703/7 Only.*

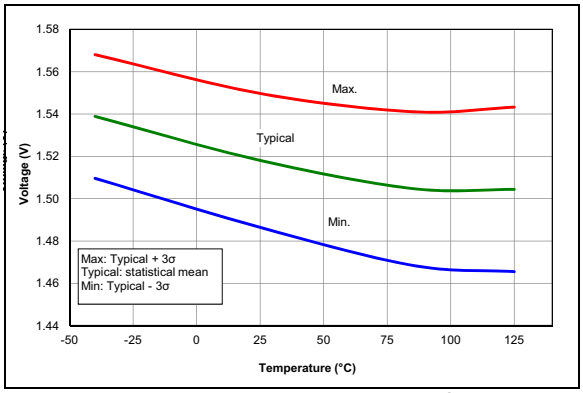



***FIGURE 27-58:*** *POR REARM Voltage, Normal-Power Mode (VREGPM1 = 0), PIC16F1703/7 Only.*

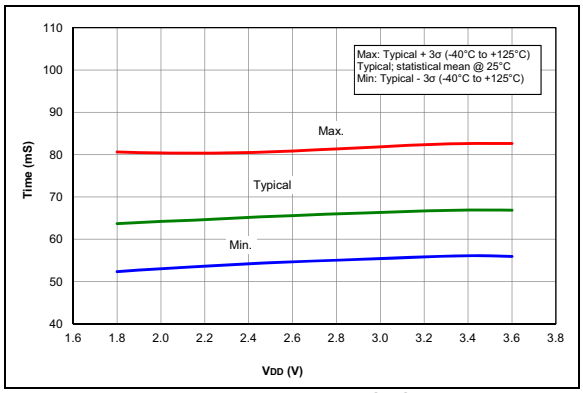



***FIGURE 27-56:*** *PWRT Period, PIC16LF1703/7 Only.*

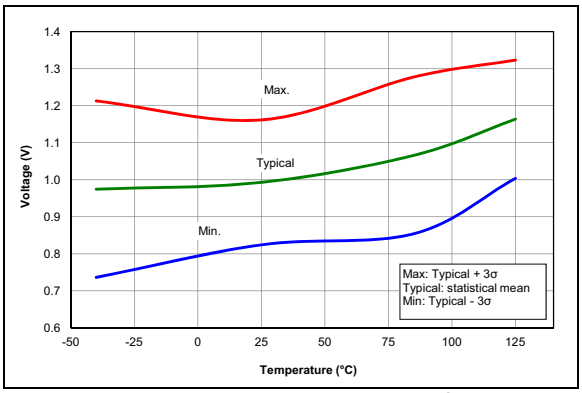



***FIGURE 27-59:*** *POR REARM Voltage, Normal-Power Mode, PIC16LF1703/7 Only.*

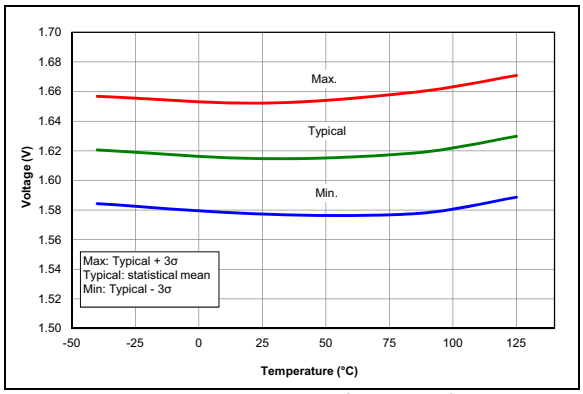



***FIGURE 27-57:*** *POR Release Voltage.*

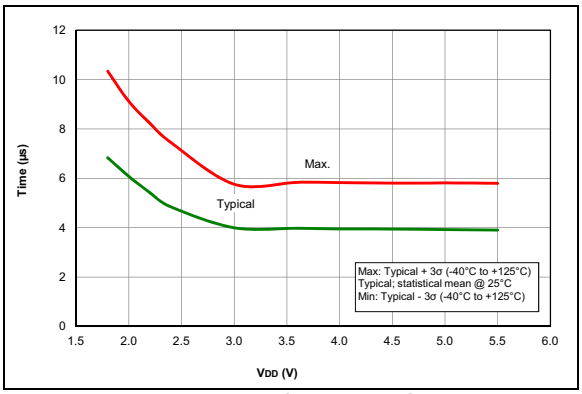



***FIGURE 27-60:*** *Wake From Sleep, VREGPM = 0.*

**Note:** Unless otherwise noted, VIN = 5V, FOSC = 500 kHz, CIN = 0.1 µF, TA = 25°C.

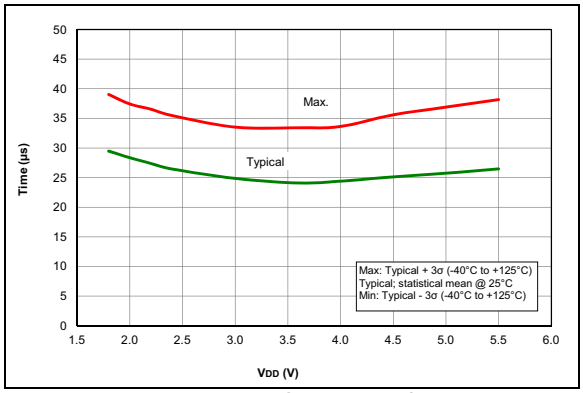



***FIGURE 27-61:*** *Wake From Sleep, VREGPM = 1.*

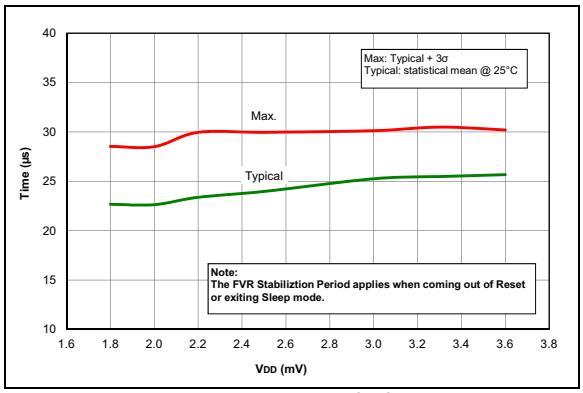



***FIGURE 27-62:*** *FVR Period, PIC16LF1703/7 Only.*

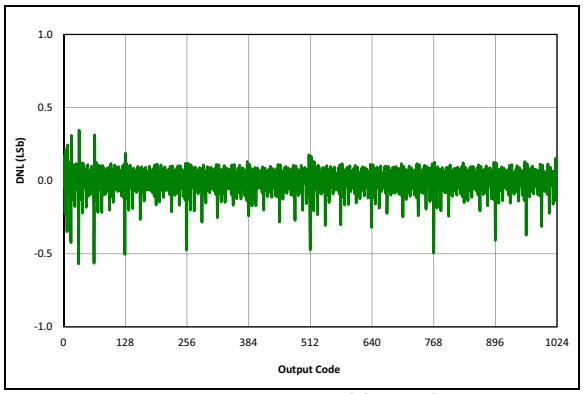



***FIGURE 27-63:*** *ADC 10-bit Mode, Single-ended DNL, VDD = 3.0V, TAD = 1 µS, 25°C.*

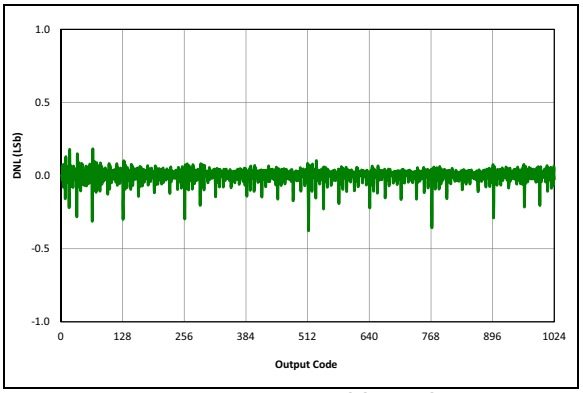



***FIGURE 27-64:*** *ADC 10-bit Mode, Single-ended DNL, VDD = 3.0V, TAD = 4 µS, 25°C.*

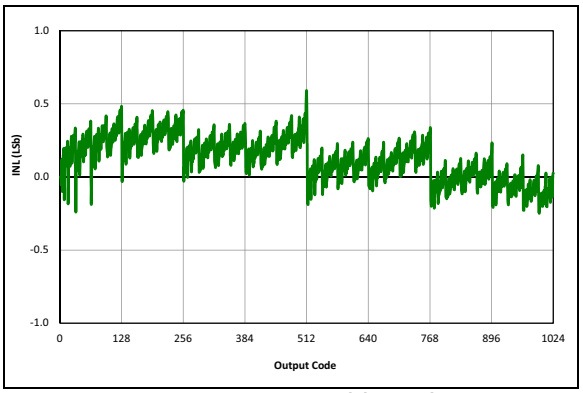



***FIGURE 27-65:*** *ADC 10-bit Mode, Single-ended INL, VDD = 3.0V, TAD = 1 µS, 25°C.*

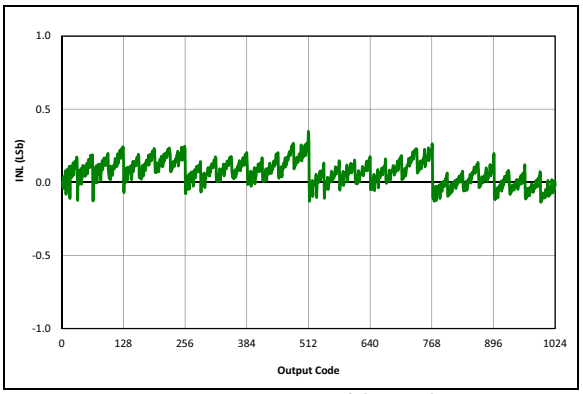



***FIGURE 27-66:*** *ADC 10-bit Mode, Single-ended INL, VDD = 3.0V, TAD = 4 µS, 25°C.*


© 2013-2015 Microchip Technology Inc.

**Note:** Unless otherwise noted, VIN = 5V, FOSC = 500 kHz, CIN = 0.1 µF, TA = 25°C.

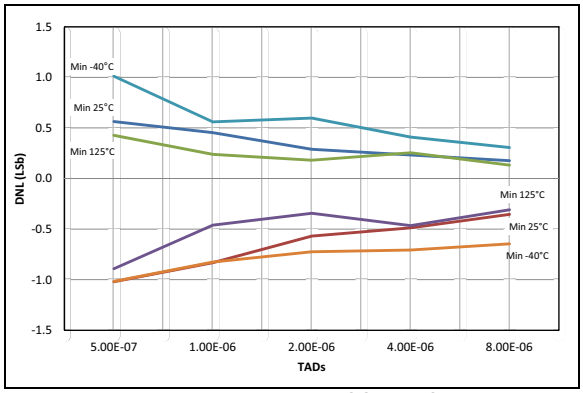



***FIGURE 27-67:*** *ADC 10-bit Mode, Single-ended DNL, VDD = 3.0V, VREF = 3.0V.*

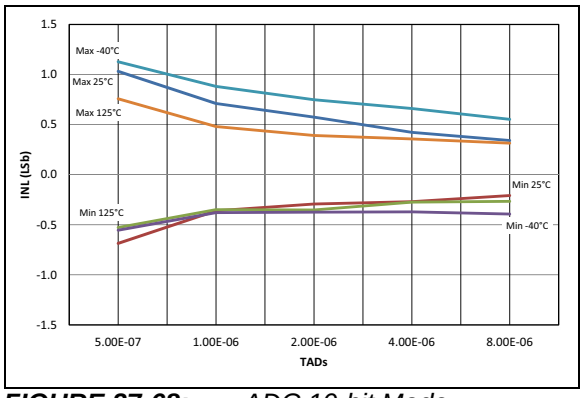



***FIGURE 27-68:*** *ADC 10-bit Mode, Single-ended INL, VDD = 3.0V, VREF = 3.0V.*

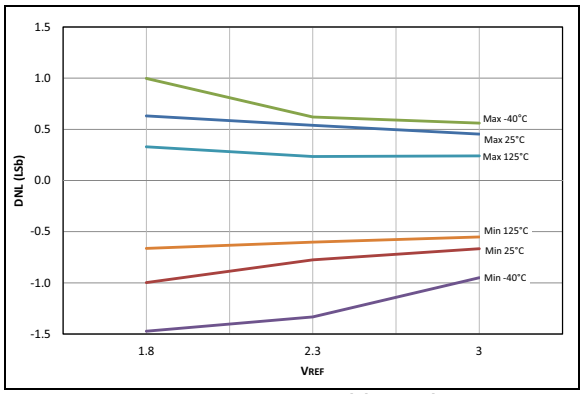



***FIGURE 27-69:*** *ADC 10-bit Mode, Single-ended DNL, VDD = 3.0V, TAD = 1 µS.*

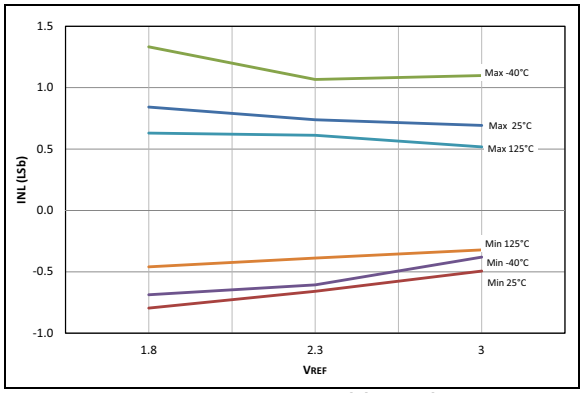



***FIGURE 27-70:*** *ADC 10-bit Mode, Single-ended INL, VDD = 3.0V, TAD = 1 µS.*

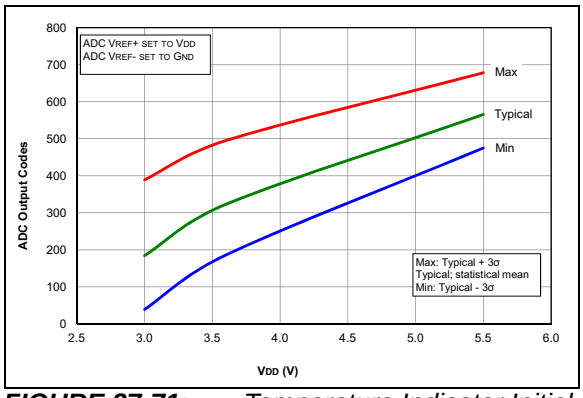



***FIGURE 27-71:*** *Temperature Indicator Initial Offset, High Range, Temp. = 20°C, PIC16F1703/7 only.*

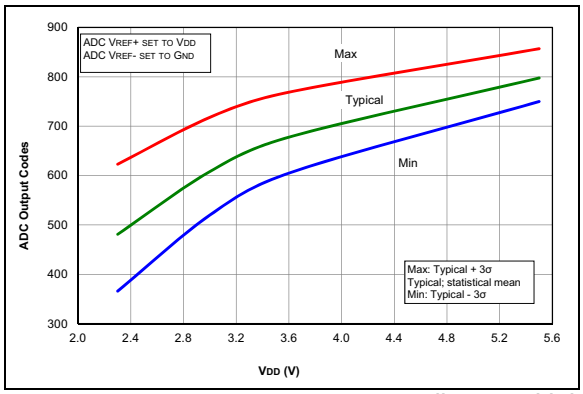



***FIGURE 27-72:*** *Temperature Indicator Initial Offset, Low Range, Temp. = 20°C, PIC16F1703/7 only.*

**Note:** Unless otherwise noted, VIN = 5V, FOSC = 500 kHz, CIN = 0.1 µF, TA = 25°C.

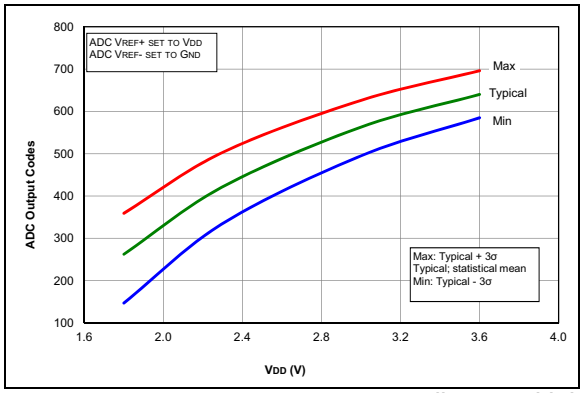



***FIGURE 27-73:*** *Temperature Indicator Initial Offset, Low Range, Temp. = 20°C, PIC16LF1703/7 only.*

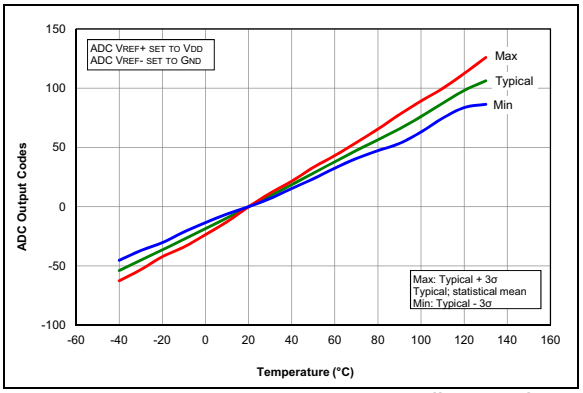



***FIGURE 27-76:*** *Temperature Indicator Slope Normalized to 20°C, Low Range, VDD = 3.0, PIC16F1703/7 only.*

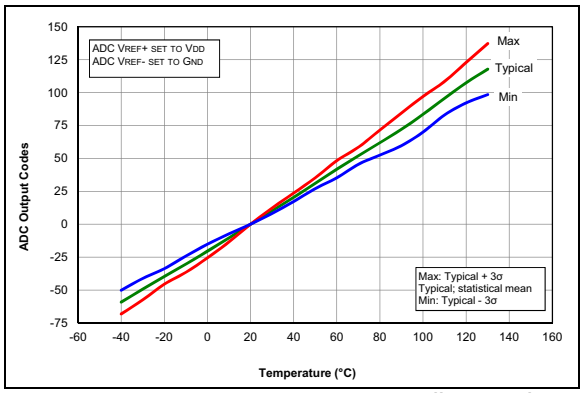



***FIGURE 27-74:*** *Temperature Indicator Slope Normalized to 20°C, High Range, VDD = 5.0, PIC16F1703/7 only.*

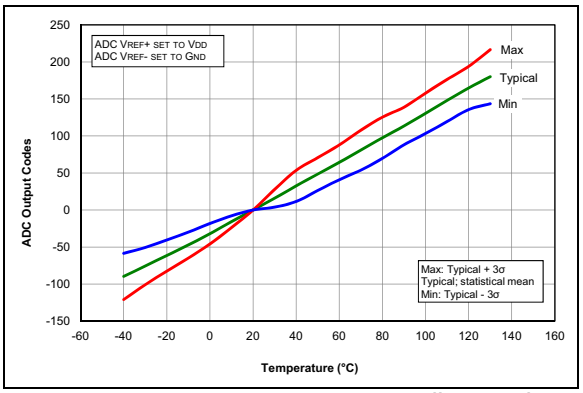



***FIGURE 27-77:*** *Temperature Indicator Slope Normalized to 20°C, Low Range, VDD = 1.8, PIC16LF1703/7 only.*

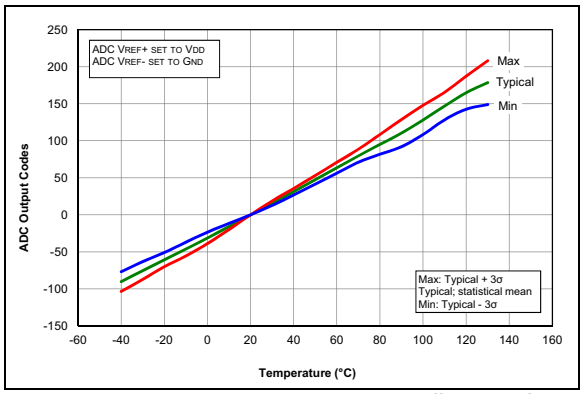



***FIGURE 27-75:*** *Temperature Indicator Slope Normalized to 20°C, High Range, VDD = 3.0, PIC16F1703/7 only.*

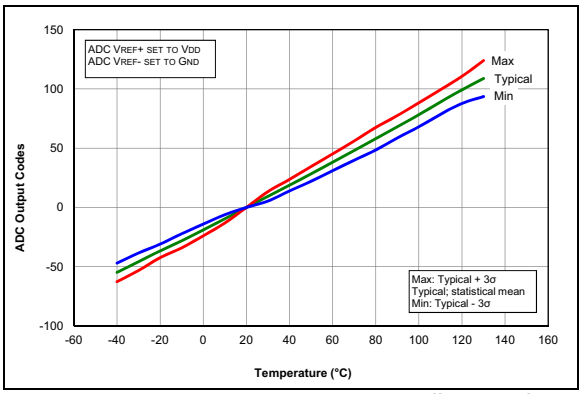



***FIGURE 27-78:*** *Temperature Indicator Slope Normalized to 20°C, Low Range, VDD = 3.0, PIC16LF1703/7 only.*


© 2013-2015 Microchip Technology Inc.

**Note:** Unless otherwise noted, VIN = 5V, FOSC = 500 kHz, CIN = 0.1 µF, TA = 25°C.

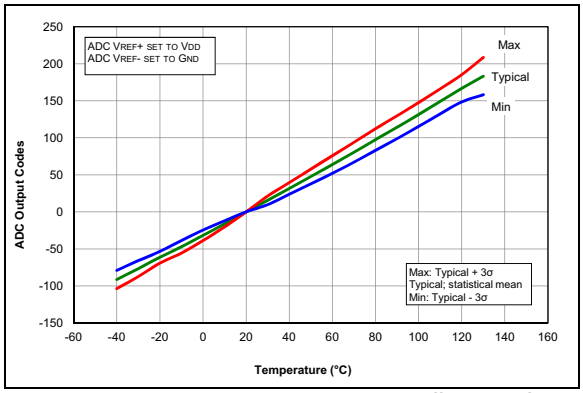



***FIGURE 27-79:*** *Temperature Indicator Slope Normalized to 20°C, High Range, VDD = 3.0, PIC16LF1703/7 only.*

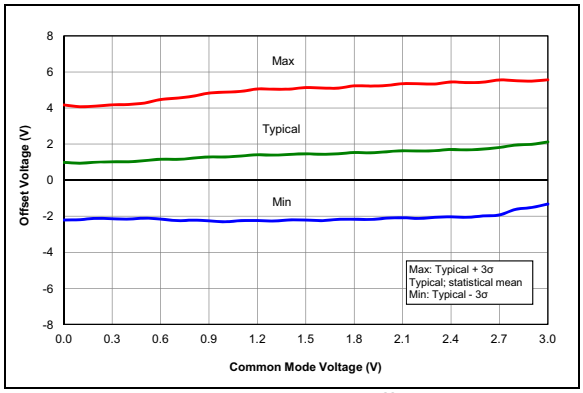



***FIGURE 27-82:*** *Op Amp, Offset Over Common-Mode Voltage, VDD = 3.0V, Temp. = 25°C*

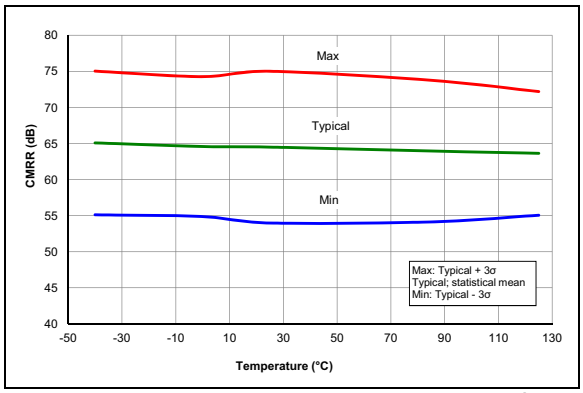



***FIGURE 27-80:*** *Op Amp, Common-Mode Rejection Ratio (CMRR), VDD = 3.0V*

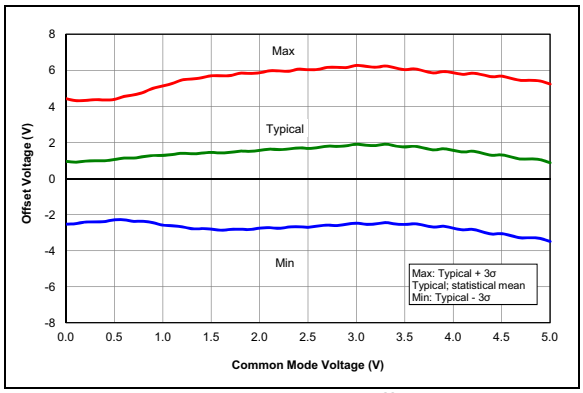



***FIGURE 27-83:*** *Op Amp, Offset Over Common-Mode Voltage, VDD = 5.0V, Temp. = 25°C, PIC16F1703/7 only.*

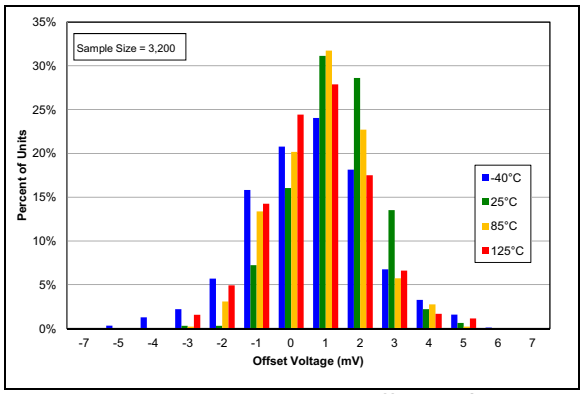



***FIGURE 27-81:*** *Op Amp, Offset Voltage Histogram, VDD = 3.0V, VCM = VDD/2*

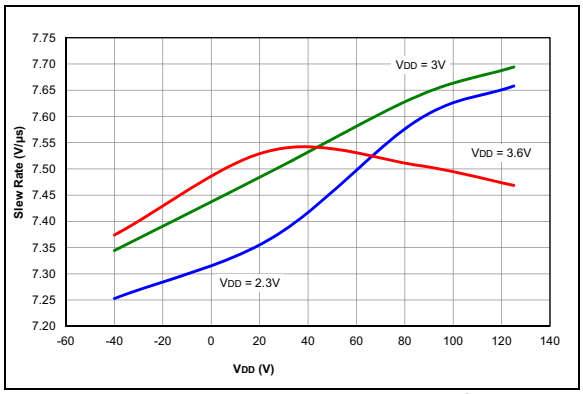



***FIGURE 27-84:*** *Op Amp, Output Slew Rate, Rising Edge, PIC16LF1703/7 only.*

**Note:** Unless otherwise noted, VIN = 5V, FOSC = 500 kHz, CIN = 0.1 µF, TA = 25°C.

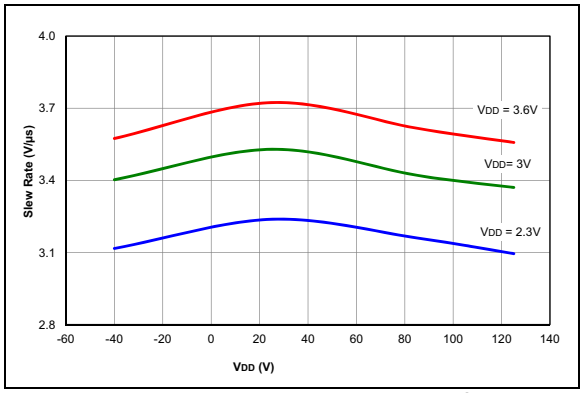



***FIGURE 27-85:*** *Op Amp, Output Slew Rate, Falling Edge, PIC16LF1703/7 only.*

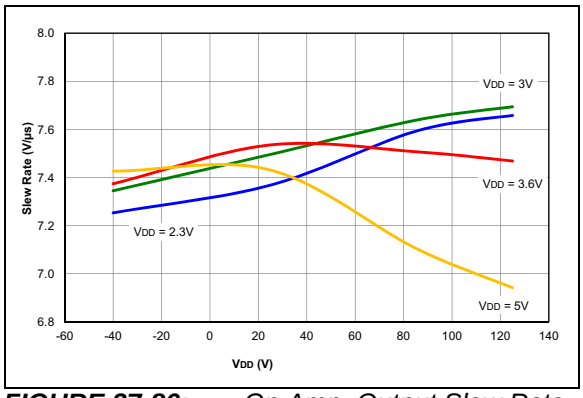



***FIGURE 27-86:*** *Op Amp, Output Slew Rate, Rising Edge, PIC16F1703/7 only.*

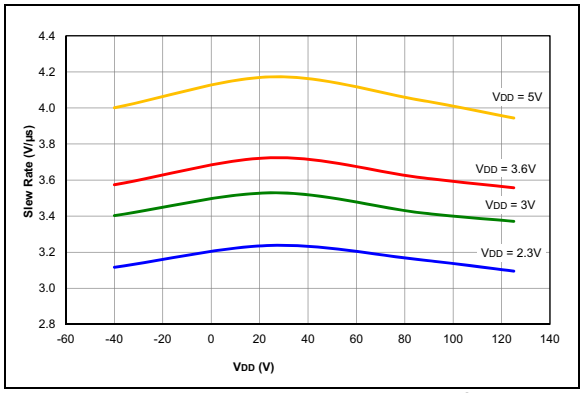



***FIGURE 27-87:*** *Op Amp, Output Slew Rate, Falling Edge, PIC16F1703/7 only.*

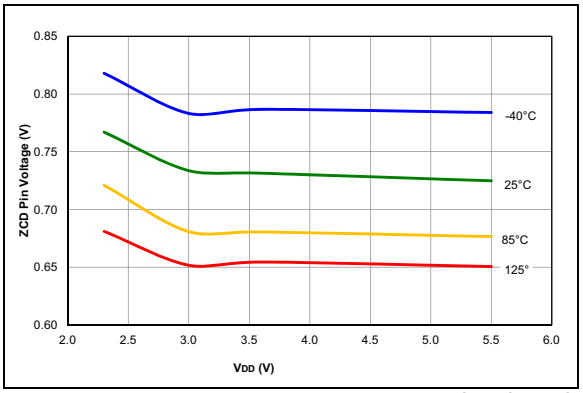



***FIGURE 27-88:*** *Zero-Cross Detection (ZCD) Pin Voltage, Typical Measured Values.*

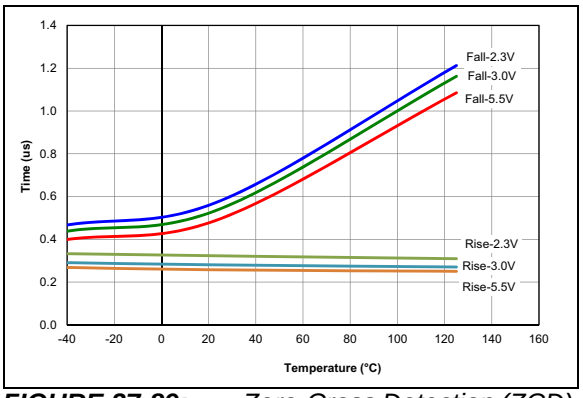



***FIGURE 27-89:*** *Zero-Cross Detection (ZCD) Time Over Voltage, Typical Measured Values.*

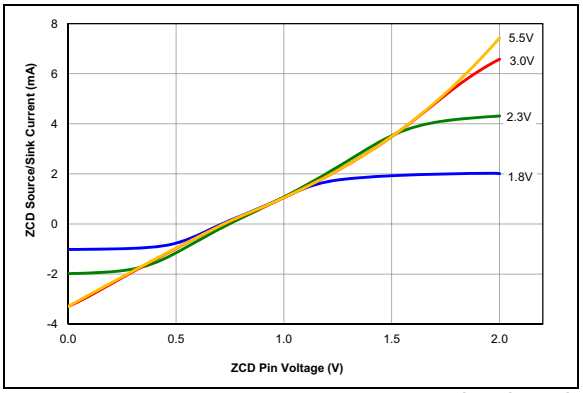



***FIGURE 27-90:*** *Zero-Cross Detection (ZCD) Pin Current Over ZCD Pin Voltage, Typical Measured Values from -40°C to 125°C.*


© 2013-2015 Microchip Technology Inc.

**Note:** Unless otherwise noted, VIN = 5V, FOSC = 500 kHz, CIN = 0.1 µF, TA = 25°C.

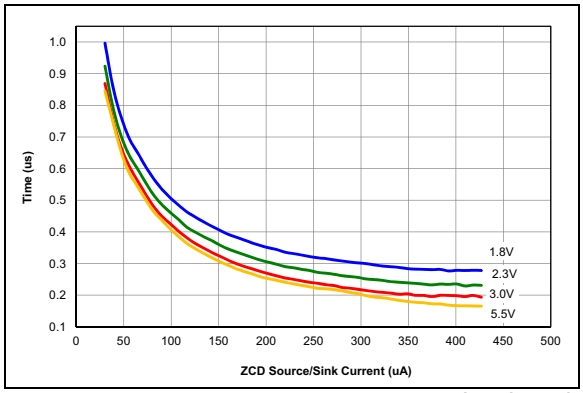



***FIGURE 27-91:*** *Zero-Cross Detection (ZCD) Pin Response Time Over Current, Typical Measured Values from -40°C to 125°C.*

## 28.0 DEVELOPMENT SUPPORT

The PIC® microcontrollers (MCU) and dsPIC® digital signal controllers (DSC) are supported with a full range of software and hardware development tools:

- Integrated Development Environment
  - MPLAB® X IDE Software
- Compilers/Assemblers/Linkers
  - MPLAB XC Compiler
  - MPASM™ Assembler
  - MPLINK™ Object Linker/ MPLIB™ Object Librarian
  - MPLAB Assembler/Linker/Librarian for Various Device Families
- Simulators
  - MPLAB X SIM Software Simulator
- Emulators
  - MPLAB REAL ICE™ In-Circuit Emulator
- In-Circuit Debuggers/Programmers
  - MPLAB ICD 3
  - PICkit™ 3
- Device Programmers
  - MPLAB PM3 Device Programmer
- Low-Cost Demonstration/Development Boards, Evaluation Kits and Starter Kits
- Third-party development tools

### 28.1 MPLAB X Integrated Development Environment Software

The MPLAB X IDE is a single, unified graphical user interface for Microchip and third-party software, and hardware development tool that runs on Windows®, Linux and Mac OS® X. Based on the NetBeans IDE, MPLAB X IDE is an entirely new IDE with a host of free software components and plug-ins for high-performance application development and debugging. Moving between tools and upgrading from software simulators to hardware debugging and programming tools is simple with the seamless user interface.

With complete project management, visual call graphs, a configurable watch window and a feature-rich editor that includes code completion and context menus, MPLAB X IDE is flexible and friendly enough for new users. With the ability to support multiple tools on multiple projects with simultaneous debugging, MPLAB X IDE is also suitable for the needs of experienced users.

Feature-Rich Editor:

- Color syntax highlighting
- Smart code completion makes suggestions and provides hints as you type
- Automatic code formatting based on user-defined rules
- Live parsing

User-Friendly, Customizable Interface:

- Fully customizable interface: toolbars, toolbar buttons, windows, window placement, etc.
- Call graph window

Project-Based Workspaces:

- Multiple projects
- Multiple tools
- Multiple configurations
- Simultaneous debugging sessions

File History and Bug Tracking:

- Local file history feature
- Built-in support for Bugzilla issue tracker


© 2013-2015 Microchip Technology Inc.

## 28.2 MPLAB XC Compilers

The MPLAB XC Compilers are complete ANSI C compilers for all of Microchip's 8, 16, and 32-bit MCU and DSC devices. These compilers provide powerful integration capabilities, superior code optimization and ease of use. MPLAB XC Compilers run on Windows, Linux or MAC OS X.

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

The free MPLAB XC Compiler editions support all devices and commands, with no time or memory restrictions, and offer sufficient code optimization for most applications.

MPLAB XC Compilers include an assembler, linker and utilities. 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. MPLAB XC Compiler uses the assembler to produce its object file. Notable features of the assembler include:

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

## 28.3 MPASM Assembler

The MPASM Assembler is a full-featured, universal macro assembler for PIC10/12/16/18 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 X IDE projects
- User-defined macros to streamline assembly code
- Conditional assembly for multipurpose source files
- Directives that allow complete control over the assembly process

## 28.4 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK Object Linker combines relocatable objects created by the MPASM Assembler. 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

## 28.5 MPLAB Assembler, Linker and Librarian for Various Device Families

MPLAB Assembler produces relocatable machine code from symbolic assembly language for PIC24, PIC32 and dsPIC DSC devices. MPLAB XC Compiler uses the assembler to produce its object file. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. Notable features of the assembler include:

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

## 28.6 MPLAB X SIM Software Simulator

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

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

## 28.7 MPLAB REAL ICE In-Circuit Emulator System

The MPLAB REAL ICE In-Circuit Emulator System is Microchip's next generation high-speed emulator for Microchip Flash DSC and MCU devices. It debugs and programs all 8, 16 and 32-bit MCU, and DSC devices with the easy-to-use, powerful graphical user interface of the MPLAB X IDE.

The emulator is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with either a connector compatible with in-circuit debugger systems (RJ-11) or with the new high-speed, noise tolerant, Low-Voltage Differential Signal (LVDS) interconnection (CAT5).

The emulator is field upgradeable through future firmware downloads in MPLAB X IDE. MPLAB REAL ICE offers significant advantages over competitive emulators including full-speed emulation, run-time variable watches, trace analysis, complex breakpoints, logic probes, a ruggedized probe interface and long (up to three meters) interconnection cables.

## 28.8 MPLAB ICD 3 In-Circuit Debugger System

The MPLAB ICD 3 In-Circuit Debugger System is Microchip's most cost-effective, high-speed hardware debugger/programmer for Microchip Flash DSC and MCU devices. It debugs and programs PIC Flash microcontrollers and dsPIC DSCs with the powerful, yet easy-to-use graphical user interface of the MPLAB IDE.

The MPLAB ICD 3 In-Circuit Debugger probe is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with a connector compatible with the MPLAB ICD 2 or MPLAB REAL ICE systems (RJ-11). MPLAB ICD 3 supports all MPLAB ICD 2 headers.

## 28.9 PICkit 3 In-Circuit Debugger/ Programmer

The MPLAB PICkit 3 allows debugging and programming of PIC and dsPIC Flash microcontrollers at a most affordable price point using the powerful graphical user interface of the MPLAB IDE. The MPLAB PICkit 3 is connected to the design engineer's PC using a full-speed USB interface and can be connected to the target via a Microchip debug (RJ-11) connector (compatible with MPLAB ICD 3 and MPLAB REAL ICE). The connector uses two device I/O pins and the Reset line to implement in-circuit debugging and In-Circuit Serial Programming™ (ICSP™).

## 28.10 MPLAB PM3 Device Programmer

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


© 2013-2015 Microchip Technology Inc.

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

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

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

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

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

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

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

## 28.12 Third-Party Development Tools

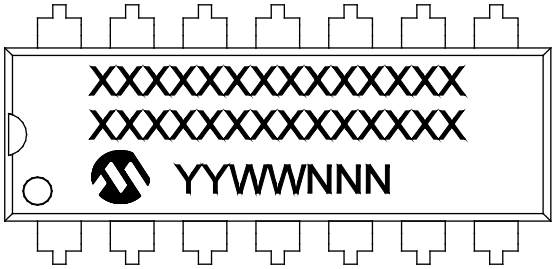
Microchip also offers a great collection of tools from third-party vendors. These tools are carefully selected to offer good value and unique functionality.

- Device Programmers and Gang Programmers from companies, such as SoftLog and CCS
- Software Tools from companies, such as Gimpel and Trace Systems
- Protocol Analyzers from companies, such as Saleae and Total Phase
- Demonstration Boards from companies, such as MikroElektronika, Digilent® and Olimex
- Embedded Ethernet Solutions from companies, such as EZ Web Lynx, WIZnet and IPLogika®

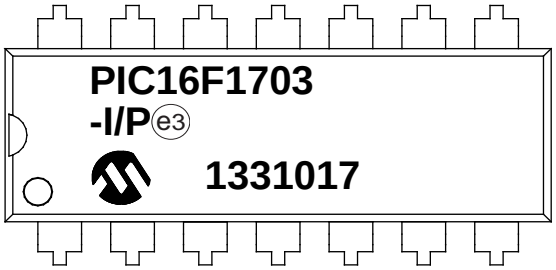
## 29.0 PACKAGING INFORMATION

### 29.1 Package Marking Information

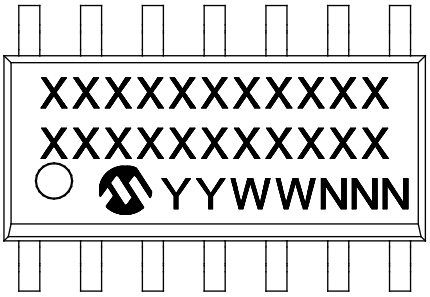
14-Lead PDIP (300 mil)



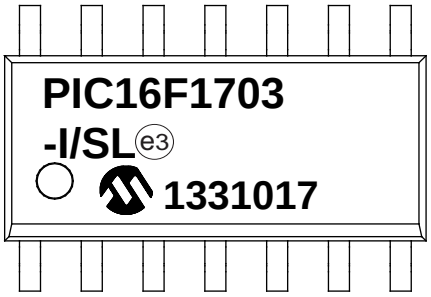

Example



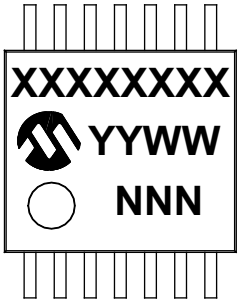

14-Lead SOIC (3.90 mm)



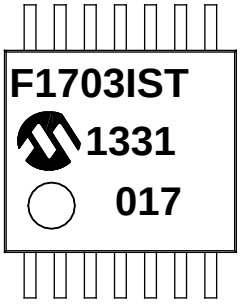

Example




14-Lead TSSOP (4.4 mm)




Example



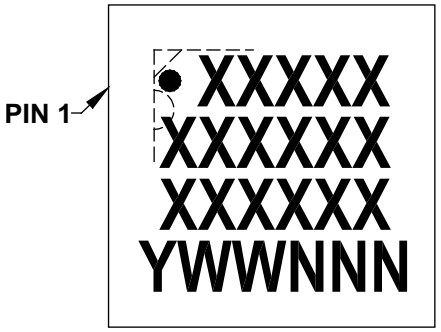

| | | |
|---|---|---|
| **Legend:** | XX...X | Customer-specific information |
| | Y | Year code (last digit of calendar year) |
| | YY | Year code (last 2 digits of calendar year) |
| | WW | Week code (week of January 1 is week '01') |
| | NNN | Alphanumeric traceability code |
| | (e3) | Pb-free JEDEC® 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.

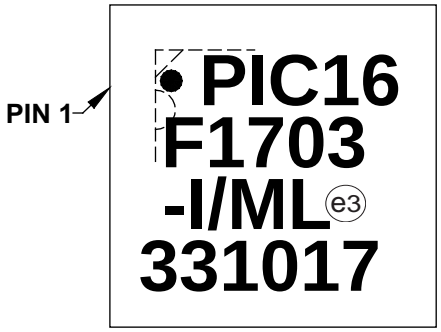

© 2013-2015 Microchip Technology Inc.

**Package Marking Information (Continued)**

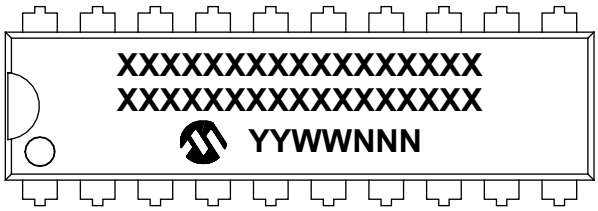
16-Lead QFN (4x4x0.9 mm)



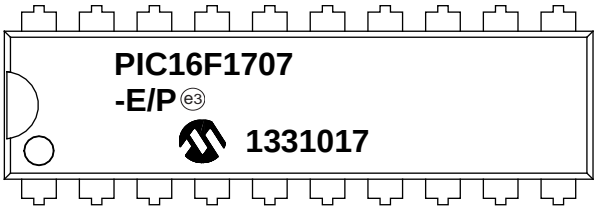

Example



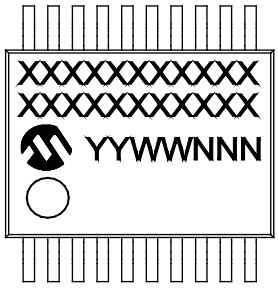

20-Lead PDIP (300 mil)



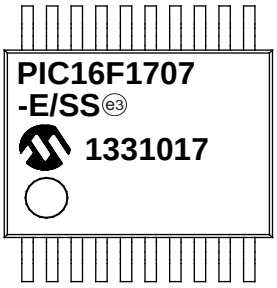

Example



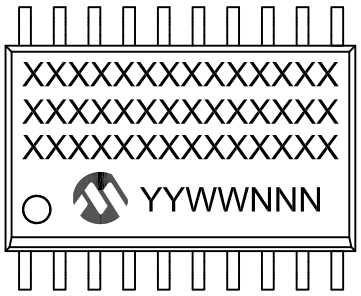

20-Lead SSOP (5.30 mm)




Example




20-Lead SOIC (7.50 mm)

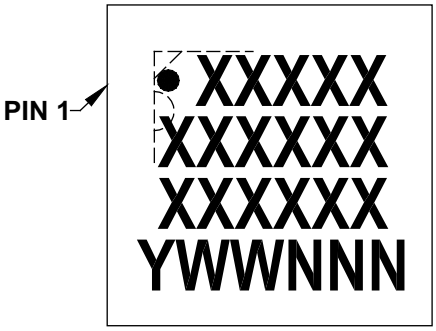



Example

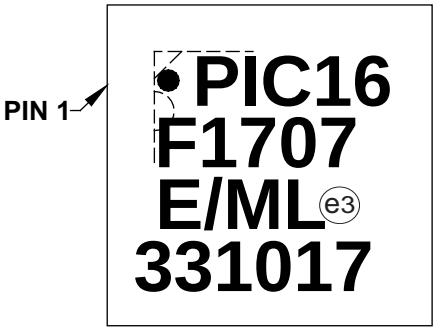


**Package Marking Information (Continued)**

20-Lead QFN (4x4x0.9 mm)




Example





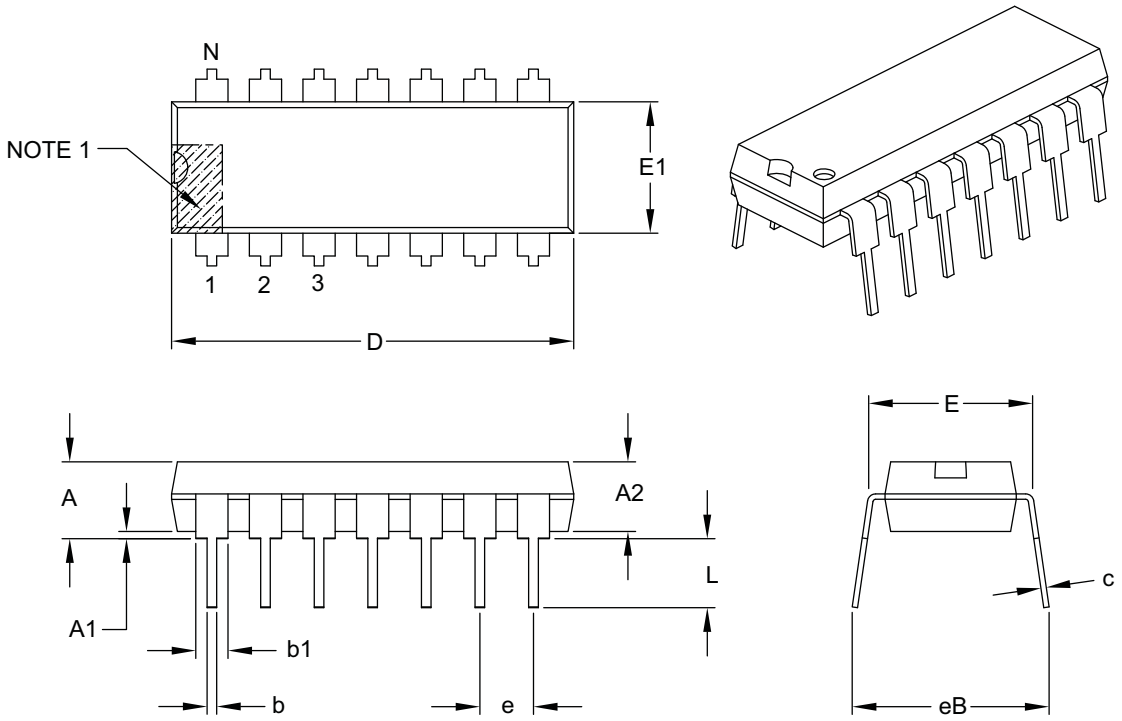
© 2013-2015 Microchip Technology Inc.

## 29.2 Package Details

The following sections give the technical details of the packages.

### 14-Lead Plastic Dual In-Line (P) – 300 mil Body [PDIP]

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging




| Units | | INCHES | | |
|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX |
| Number of Pins | N | | 14 | |
| Pitch | e | | .100 BSC | |
| Top to Seating Plane | A | – | – | .210 |
| Molded Package Thickness | A2 | .115 | .130 | .195 |
| Base to Seating Plane | A1 | .015 | – | – |
| Shoulder to Shoulder Width | E | .290 | .310 | .325 |
| Molded Package Width | E1 | .240 | .250 | .280 |
| Overall Length | D | .735 | .750 | .775 |
| Tip to Seating Plane | L | .115 | .130 | .150 |
| Lead Thickness | c | .008 | .010 | .015 |
| Upper Lead Width | b1 | .045 | .060 | .070 |
| Lower Lead Width | b | .014 | .018 | .022 |
| Overall Row Spacing § | eB | – | – | .430 |

**Notes:**

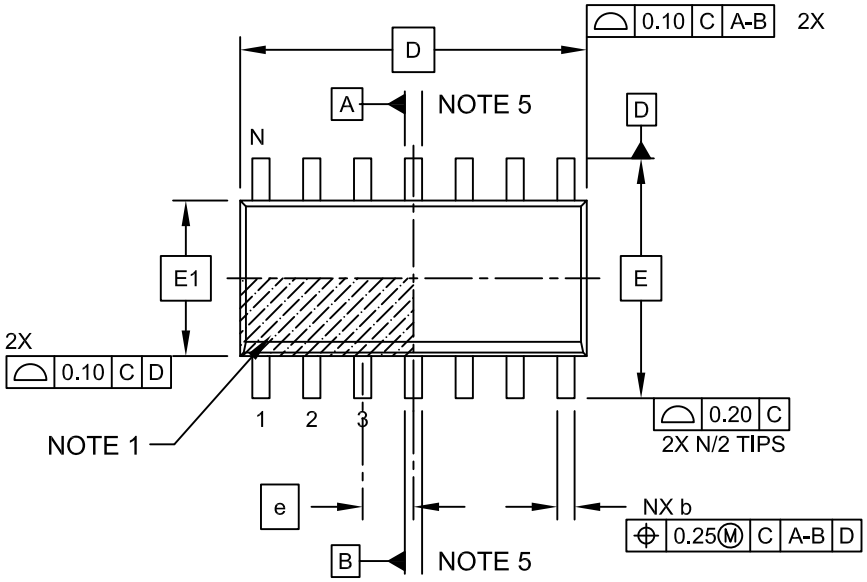
1. Pin 1 visual index feature may vary, but must be located with the hatched area.
2. § Significant Characteristic.
3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.
4. Dimensioning and tolerancing per ASME Y14.5M.

   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

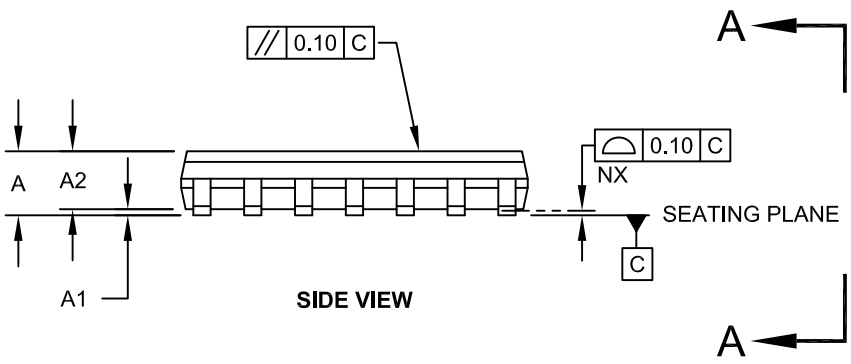
Microchip Technology Drawing C04-005B

### 14-Lead Plastic Small Outline (SL) - Narrow, 3.90 mm Body [SOIC]

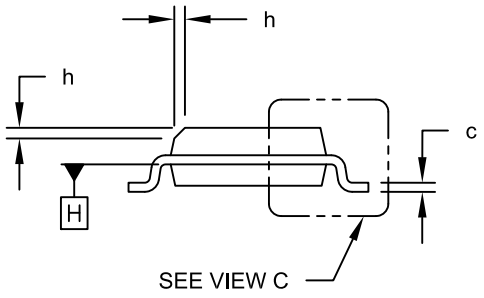
**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging




**TOP VIEW**




**SIDE VIEW**



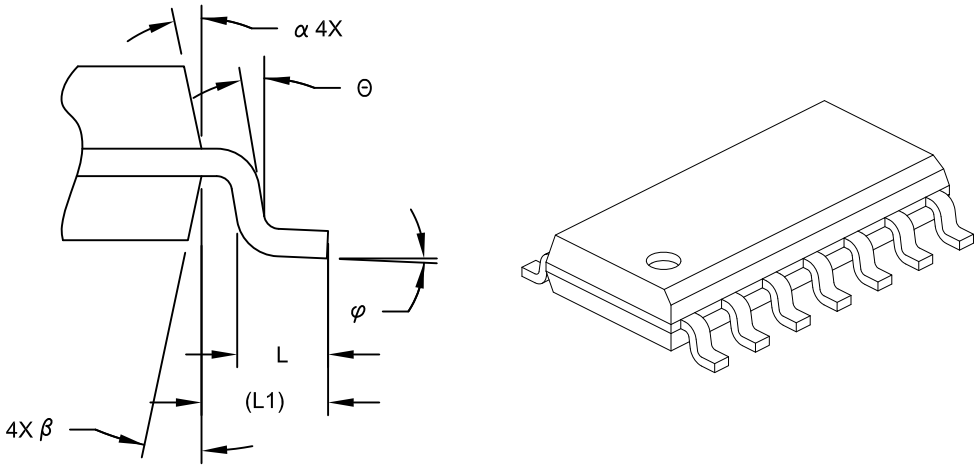

**VIEW A-A**

Microchip Technology Drawing No. C04-065C Sheet 1 of 2


© 2013-2015 Microchip Technology Inc.

### 14-Lead Plastic Small Outline (SL) - Narrow, 3.90 mm Body [SOIC]

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging




**VIEW C**

| Units | | MILLIMETERS | | |
|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX |
| Number of Pins | N | | 14 | |
| Pitch | e | | 1.27 BSC | |
| Overall Height | A | - | - | 1.75 |
| Molded Package Thickness | A2 | 1.25 | - | - |
| Standoff § | A1 | 0.10 | - | 0.25 |
| Overall Width | E | | 6.00 BSC | |
| Molded Package Width | E1 | | 3.90 BSC | |
| Overall Length | D | | 8.65 BSC | |
| Chamfer (Optional) | h | 0.25 | - | 0.50 |
| Foot Length | L | 0.40 | - | 1.27 |
| Footprint | L1 | | 1.04 REF | |
| Lead Angle | Θ | 0° | - | - |
| Foot Angle | φ | 0° | - | 8° |
| Lead Thickness | c | 0.10 | - | 0.25 |
| Lead Width | b | 0.31 | - | 0.51 |
| Mold Draft Angle Top | α | 5° | - | 15° |
| Mold Draft Angle Bottom | β | 5° | - | 15° |

**Notes:**

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. § Significant Characteristic
3. Dimension D does not include mold flash, protrusions or gate burrs, which shall not exceed 0.15 mm per end. Dimension E1 does not include interlead flash or protrusion, which shall not exceed 0.25 mm per side.
4. Dimensioning and tolerancing per ASME Y14.5M

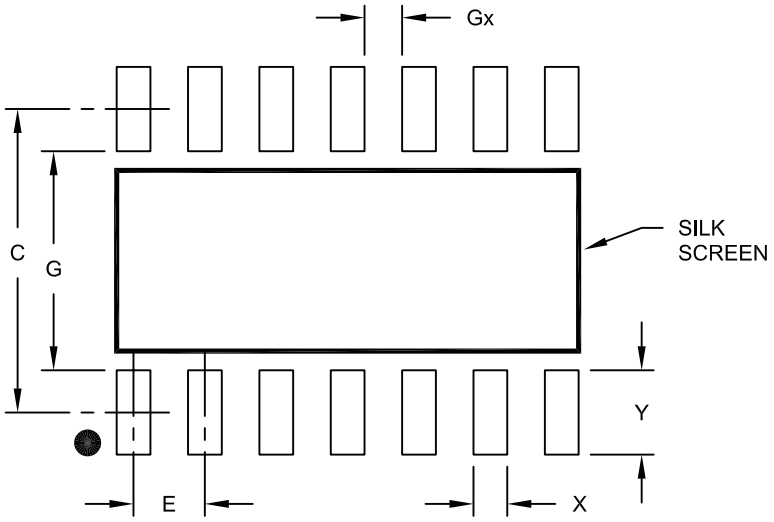
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

   REF: Reference Dimension, usually without tolerance, for information purposes only.
5. Datums A & B to be determined at Datum H.

Microchip Technology Drawing No. C04-065C Sheet 2 of 2

14-Lead Plastic Small Outline (SL) - Narrow, 3.90 mm Body [SOIC]

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging




RECOMMENDED LAND PATTERN

| Units | | MILLIMETERS | | |
|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX |
| Contact Pitch | E | | 1.27 BSC | |
| Contact Pad Spacing | C | | 5.40 | |
| Contact Pad Width | X | | | 0.60 |
| Contact Pad Length | Y | | | 1.50 |
| Distance Between Pads | Gx | 0.67 | | |
| Distance Between Pads | G | 3.90 | | |

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

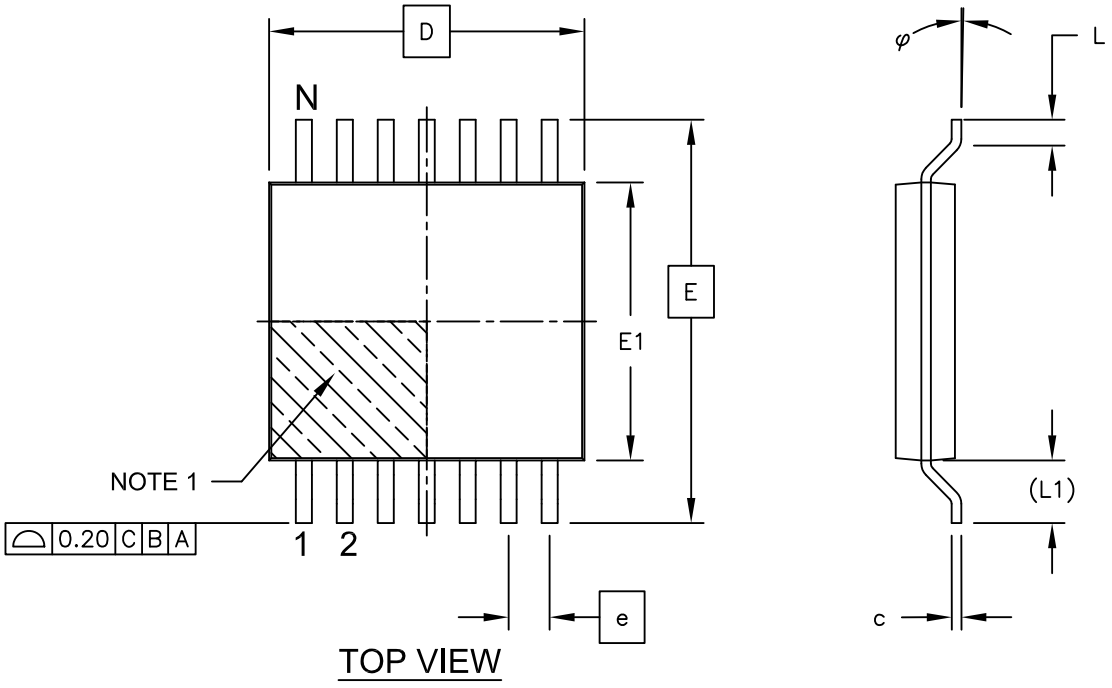
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

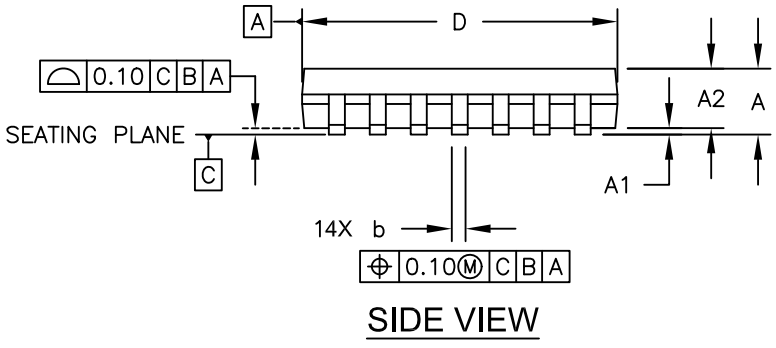
Microchip Technology Drawing No. C04-2065A


© 2013-2015 Microchip Technology Inc.

### 14-Lead Plastic Thin Shrink Small Outline (ST) - 4.4 mm Body [TSSOP]

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging

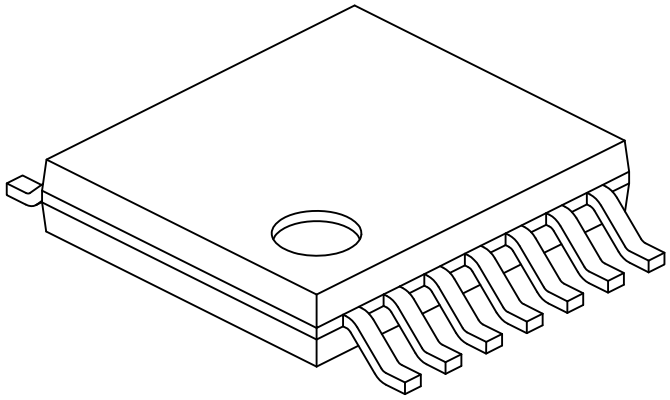



Microchip Technology Drawing C04-087C Sheet 1 of 2

### 14-Lead Plastic Thin Shrink Small Outline (ST) - 4.4 mm Body [TSSOP]

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



| Units | | MILLIMETERS | | |
|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX |
| Number of Pins | N | | 14 | |
| Pitch | e | | 0.65 BSC | |
| Overall Height | A | - | - | 1.20 |
| Molded Package Thickness | A2 | 0.80 | 1.00 | 1.05 |
| Standoff | A1 | 0.05 | - | 0.15 |
| Overall Width | E | | 6.40 BSC | |
| Molded Package Width | E1 | 4.30 | 4.40 | 4.50 |
| Molded Package Length | D | 4.90 | 5.00 | 5.10 |
| Foot Length | L | 0.45 | 0.60 | 0.75 |
| Footprint | (L1) | | 1.00 REF | |
| Foot Angle | φ | 0° | - | 8° |
| Lead Thickness | c | 0.09 | - | 0.20 |
| Lead Width | b | 0.19 | - | 0.30 |

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.15mm per side.
3. Dimensioning and tolerancing per ASME Y14.5M

   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

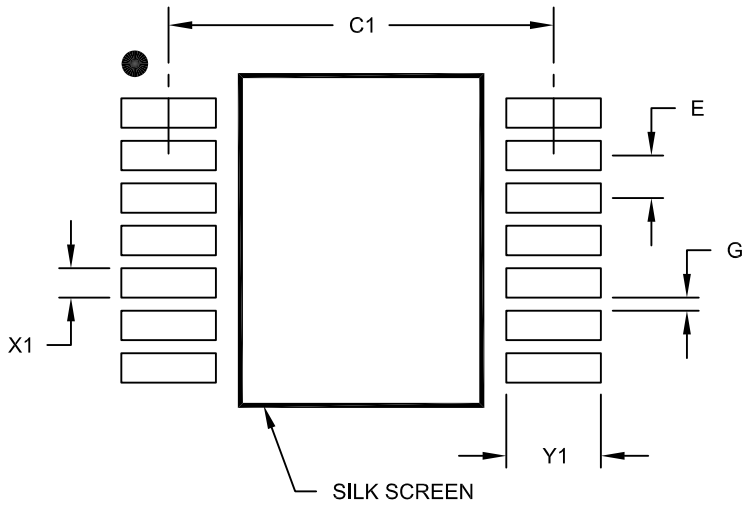
   REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing No. C04-087C Sheet 2 of 2


© 2013-2015 Microchip Technology Inc.

### 14-Lead Plastic Thin Shrink Small Outline (ST) - 4.4 mm Body [TSSOP]

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging




RECOMMENDED LAND PATTERN

| Units | | MILLIMETERS | | |
|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX |
| Contact Pitch | E | 0.65 BSC | | |
| Contact Pad Spacing | C1 | | 5.90 | |
| Contact Pad Width (X14) | X1 | | | 0.45 |
| Contact Pad Length (X14) | Y1 | | | 1.45 |
| Distance Between Pads | G | 0.20 | | |

Notes:

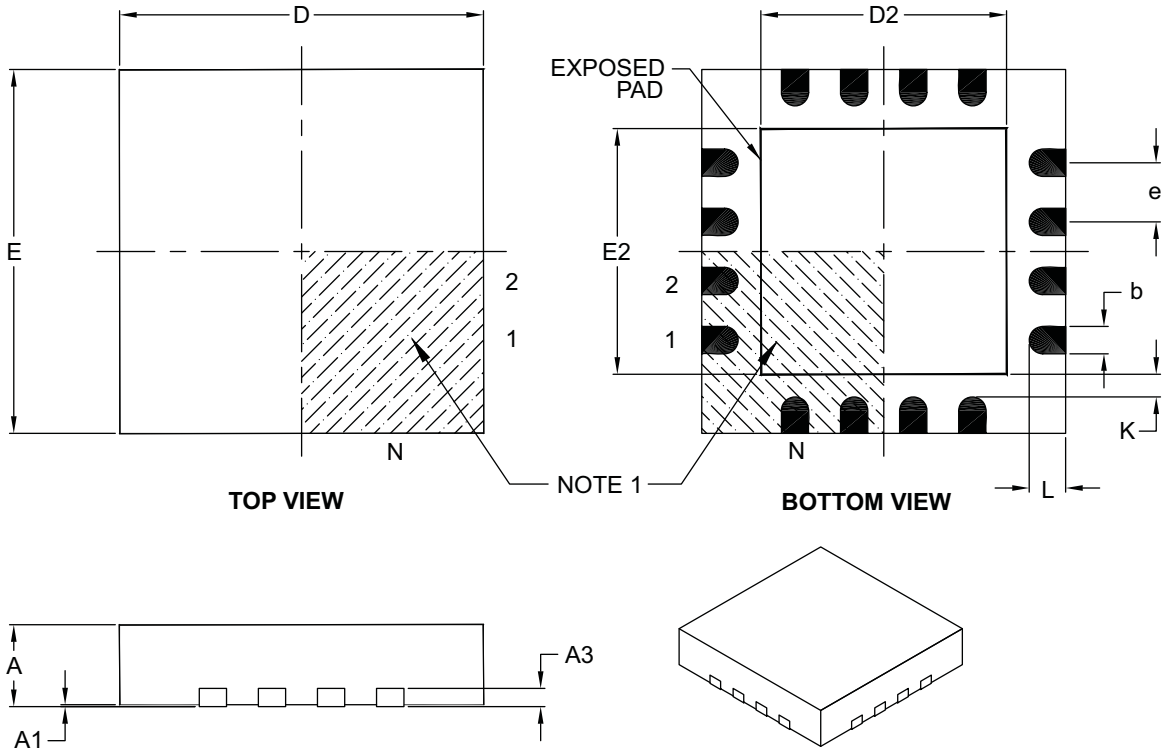
1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2087A

### 16-Lead Plastic Quad Flat, No Lead Package (ML) – 4x4x0.9 mm Body [QFN]

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging









| Units | | MILLIMETERS | | |
|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX |
| Number of Pins | N | | 16 | |
| Pitch | e | | 0.65 BSC | |
| Overall Height | A | 0.80 | 0.90 | 1.00 |
| Standoff | A1 | 0.00 | 0.02 | 0.05 |
| Contact Thickness | A3 | | 0.20 REF | |
| Overall Width | E | | 4.00 BSC | |
| Exposed Pad Width | E2 | 2.50 | 2.65 | 2.80 |
| Overall Length | D | | 4.00 BSC | |
| Exposed Pad Length | D2 | 2.50 | 2.65 | 2.80 |
| Contact Width | b | 0.25 | 0.30 | 0.35 |
| Contact Length | L | 0.30 | 0.40 | 0.50 |
| Contact-to-Exposed Pad | K | 0.20 | – | – |

**Notes:**

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Package is saw singulated.
3. Dimensioning and tolerancing per ASME Y14.5M.

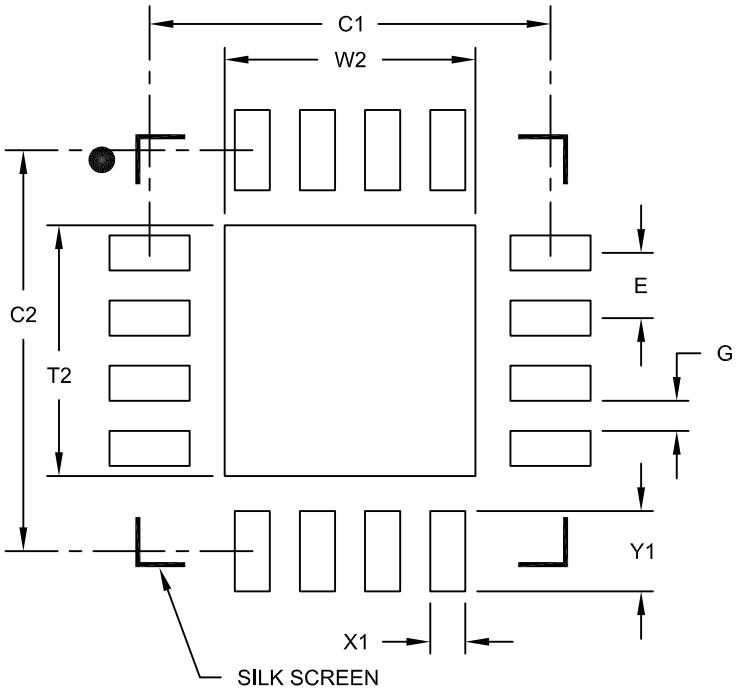
BSC: Basic Dimension. Theoretically exact value shown without tolerances.
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-127B


© 2013-2015 Microchip Technology Inc.

### 16-Lead Plastic Quad Flat, No Lead Package (ML) - 4x4x0.9mm Body [QFN]

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging




RECOMMENDED LAND PATTERN

| Units | | MILLIMETERS | | |
|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX |
| Contact Pitch | E | | 0.65 BSC | |
| Optional Center Pad Width | W2 | | | 2.50 |
| Optional Center Pad Length | T2 | | | 2.50 |
| Contact Pad Spacing | C1 | | 4.00 | |
| Contact Pad Spacing | C2 | | 4.00 | |
| Contact Pad Width (X16) | X1 | | | 0.35 |
| Contact Pad Length (X16) | Y1 | | | 0.80 |
| Distance Between Pads | G | 0.30 | | |

Notes:

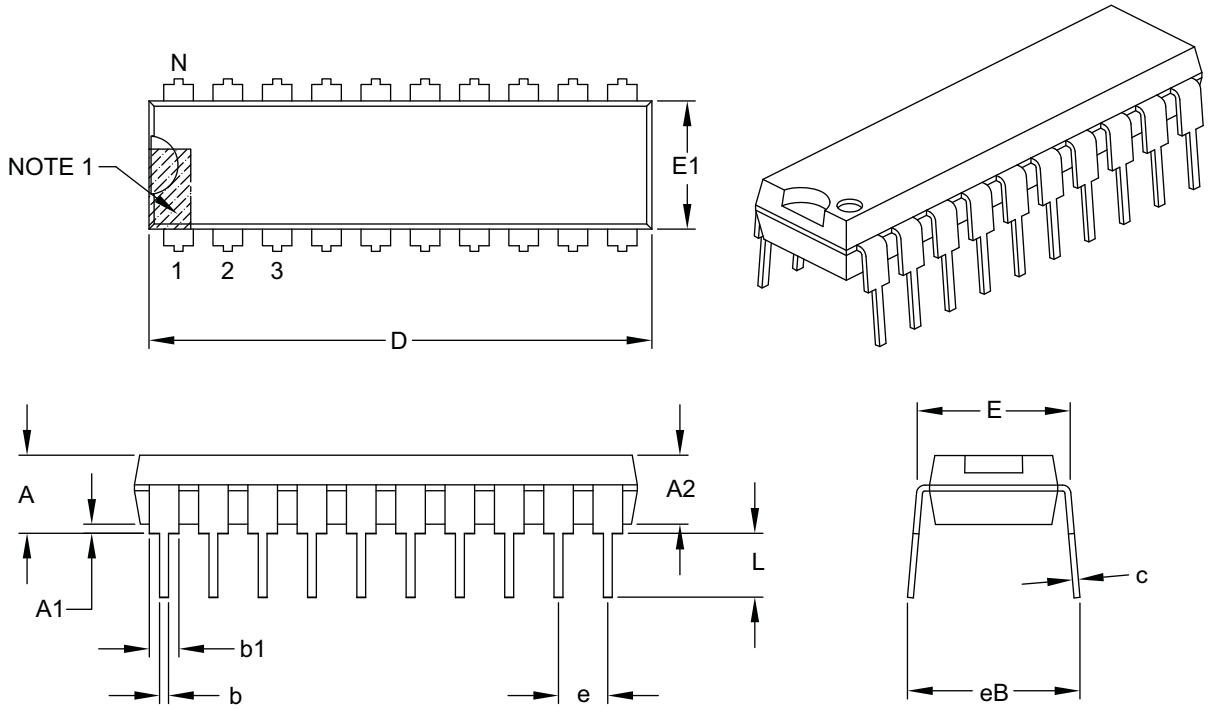
1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2127A

### 20-Lead Plastic Dual In-Line (P) – 300 mil Body [PDIP]

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging




| Units | | INCHES | | |
|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX |
| Number of Pins | N | | 20 | |
| Pitch | e | | .100 BSC | |
| Top to Seating Plane | A | – | – | .210 |
| Molded Package Thickness | A2 | .115 | .130 | .195 |
| Base to Seating Plane | A1 | .015 | – | – |
| Shoulder to Shoulder Width | E | .300 | .310 | .325 |
| Molded Package Width | E1 | .240 | .250 | .280 |
| Overall Length | D | .980 | 1.030 | 1.060 |
| Tip to Seating Plane | L | .115 | .130 | .150 |
| Lead Thickness | c | .008 | .010 | .015 |
| Upper Lead Width | b1 | .045 | .060 | .070 |
| Lower Lead Width | b | .014 | .018 | .022 |
| Overall Row Spacing § | eB | – | – | .430 |

**Notes:**

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. § Significant Characteristic.
3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.
4. Dimensioning and tolerancing per ASME Y14.5M.

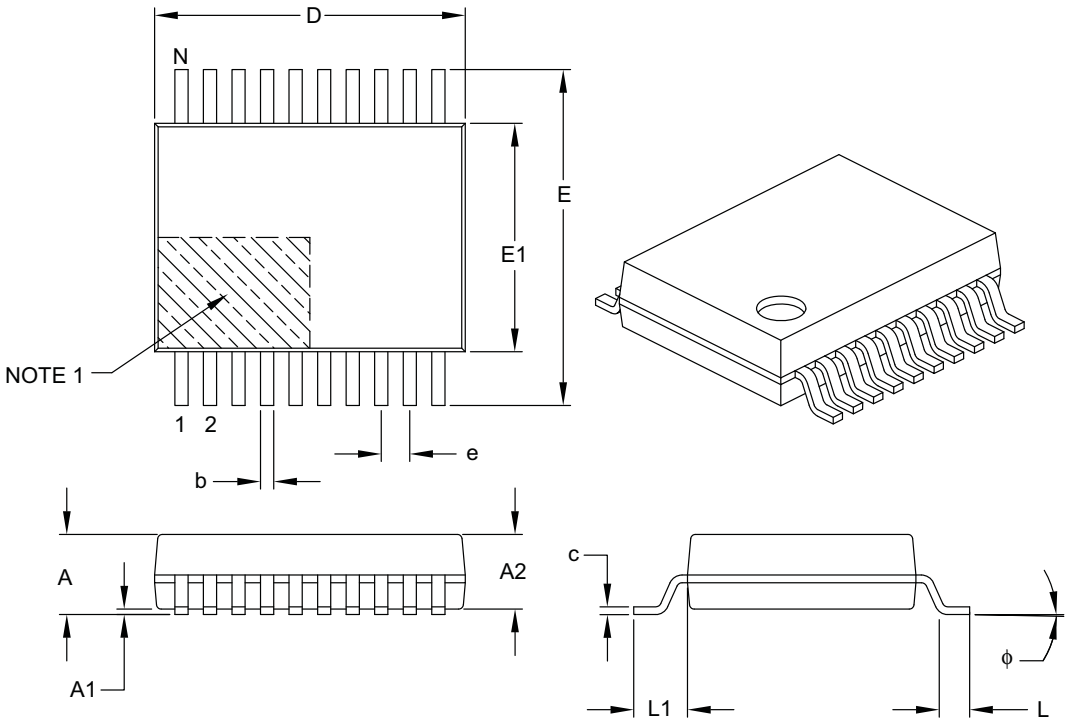
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing C04-019B


© 2013-2015 Microchip Technology Inc.

### 20-Lead Plastic Shrink Small Outline (SS) – 5.30 mm Body [SSOP]

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging




| Units | | MILLIMETERS | | |
|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX |
| Number of Pins | N | | 20 | |
| Pitch | e | | 0.65 BSC | |
| Overall Height | A | – | – | 2.00 |
| Molded Package Thickness | A2 | 1.65 | 1.75 | 1.85 |
| Standoff | A1 | 0.05 | – | – |
| Overall Width | E | 7.40 | 7.80 | 8.20 |
| Molded Package Width | E1 | 5.00 | 5.30 | 5.60 |
| Overall Length | D | 6.90 | 7.20 | 7.50 |
| Foot Length | L | 0.55 | 0.75 | 0.95 |
| Footprint | L1 | | 1.25 REF | |
| Lead Thickness | c | 0.09 | – | 0.25 |
| Foot Angle | ϕ | 0° | 4° | 8° |
| Lead Width | b | 0.22 | – | 0.38 |

**Notes:**

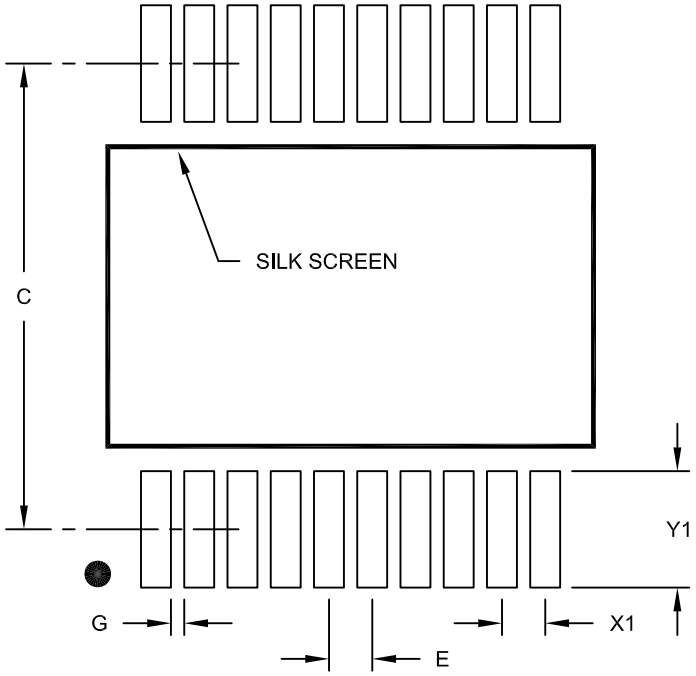
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.20 mm per side.
3. Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-072B

20-Lead Plastic Shrink Small Outline (SS) - 5.30 mm Body [SSOP]

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging




RECOMMENDED LAND PATTERN

| Units | | MILLIMETERS | | |
|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX |
| Contact Pitch | E | | 0.65 BSC | |
| Contact Pad Spacing | C | | 7.20 | |
| Contact Pad Width (X20) | X1 | | | 0.45 |
| Contact Pad Length (X20) | Y1 | | | 1.75 |
| Distance Between Pads | G | 0.20 | | |

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

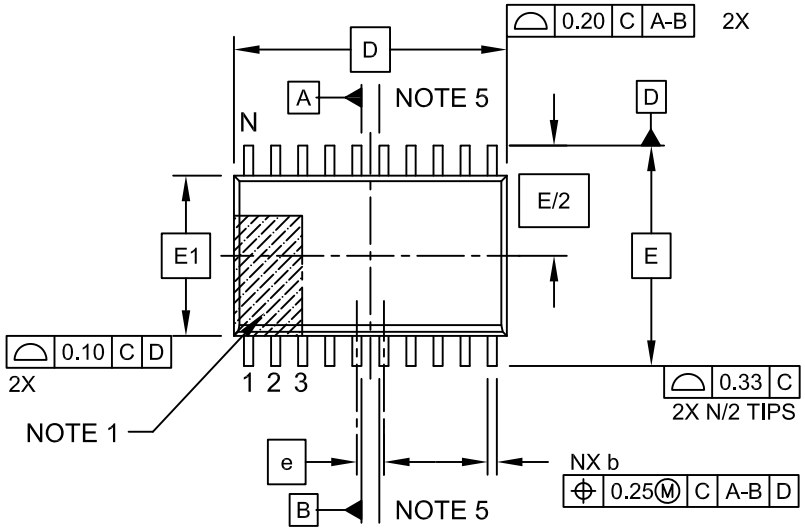
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2072A

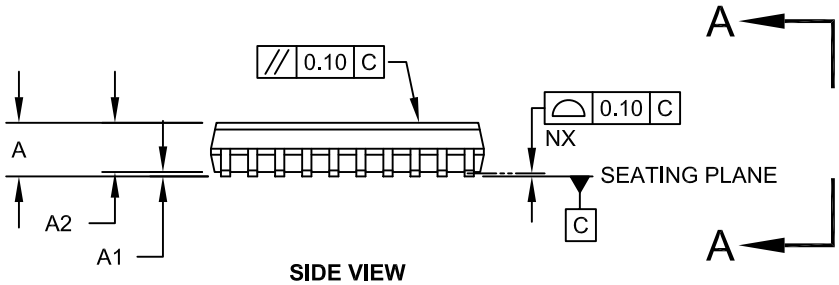

© 2013-2015 Microchip Technology Inc.

### 20-Lead Plastic Small Outline (SO) - Wide, 7.50 mm Body [SOIC]

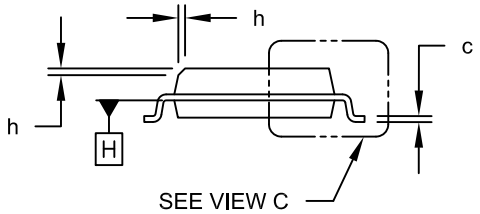
**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging




**TOP VIEW**




**SIDE VIEW**

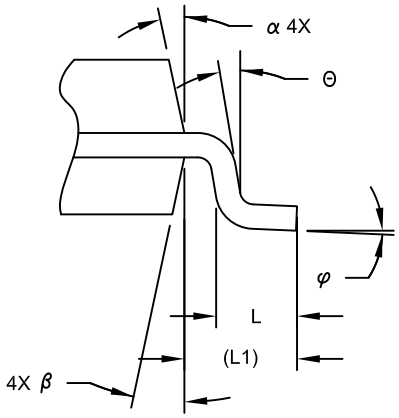



**VIEW A-A**

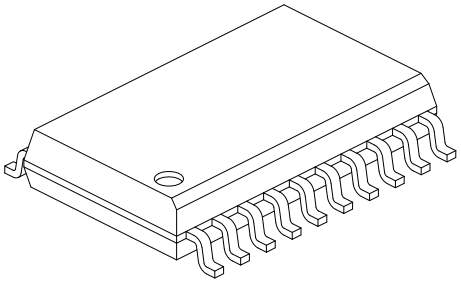
Microchip Technology Drawing C04-094C Sheet 1 of 2

### 20-Lead Plastic Small Outline (SO) - Wide, 7.50 mm Body [SOIC]

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging




VIEW C



| Units | | MILLIMETERS | | |
|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX |
| Number of Pins | N | | 20 | |
| Pitch | e | | 1.27 BSC | |
| Overall Height | A | - | - | 2.65 |
| Molded Package Thickness | A2 | 2.05 | - | - |
| Standoff § | A1 | 0.10 | - | 0.30 |
| Overall Width | E | | 10.30 BSC | |
| Molded Package Width | E1 | | 7.50 BSC | |
| Overall Length | D | | 12.80 BSC | |
| Chamfer (Optional) | h | 0.25 | - | 0.75 |
| Foot Length | L | 0.40 | - | 1.27 |
| Footprint | L1 | | 1.40 REF | |
| Lead Angle | Θ | 0° | - | - |
| Foot Angle | φ | 0° | - | 8° |
| Lead Thickness | c | 0.20 | - | 0.33 |
| Lead Width | b | 0.31 | - | 0.51 |
| Mold Draft Angle Top | α | 5° | - | 15° |
| Mold Draft Angle Bottom | β | 5° | - | 15° |

**Notes:**

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. § Significant Characteristic
3. Dimension D does not include mold flash, protrusions or gate burrs, which shall not exceed 0.15 mm per end. Dimension E1 does not include interlead flash or protrusion, which shall not exceed 0.25 mm per side.
4. Dimensioning and tolerancing per ASME Y14.5M

   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

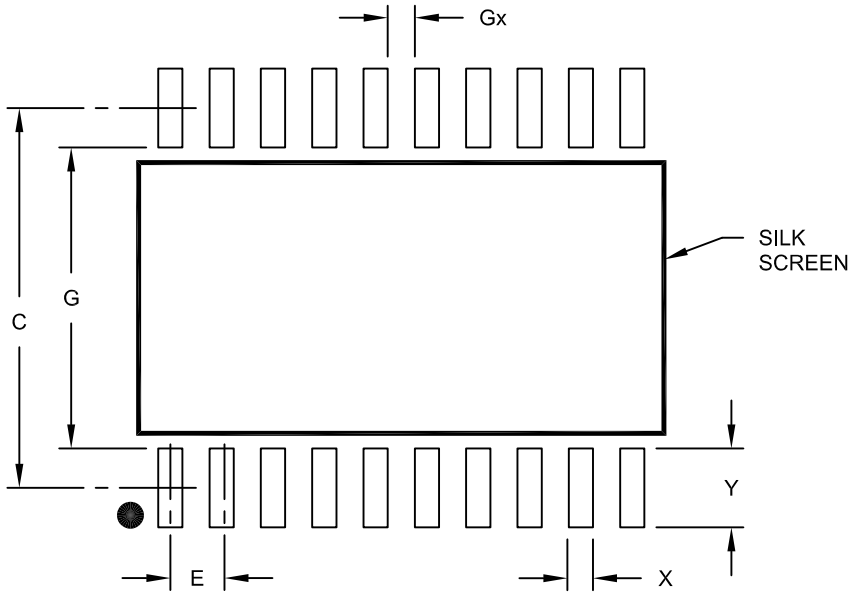
   REF: Reference Dimension, usually without tolerance, for information purposes only.
5. Datums A & B to be determined at Datum H.

Microchip Technology Drawing No. C04-094C Sheet 2 of 2


© 2013-2015 Microchip Technology Inc.

20-Lead Plastic Small Outline (SO) - Wide, 7.50 mm Body [SOIC]

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging




RECOMMENDED LAND PATTERN

| Units | | MILLIMETERS | | |
|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX |
| Contact Pitch | E | | 1.27 BSC | |
| Contact Pad Spacing | C | | 9.40 | |
| Contact Pad Width (X20) | X | | | 0.60 |
| Contact Pad Length (X20) | Y | | | 1.95 |
| Distance Between Pads | Gx | 0.67 | | |
| Distance Between Pads | G | 7.45 | | |

Notes:

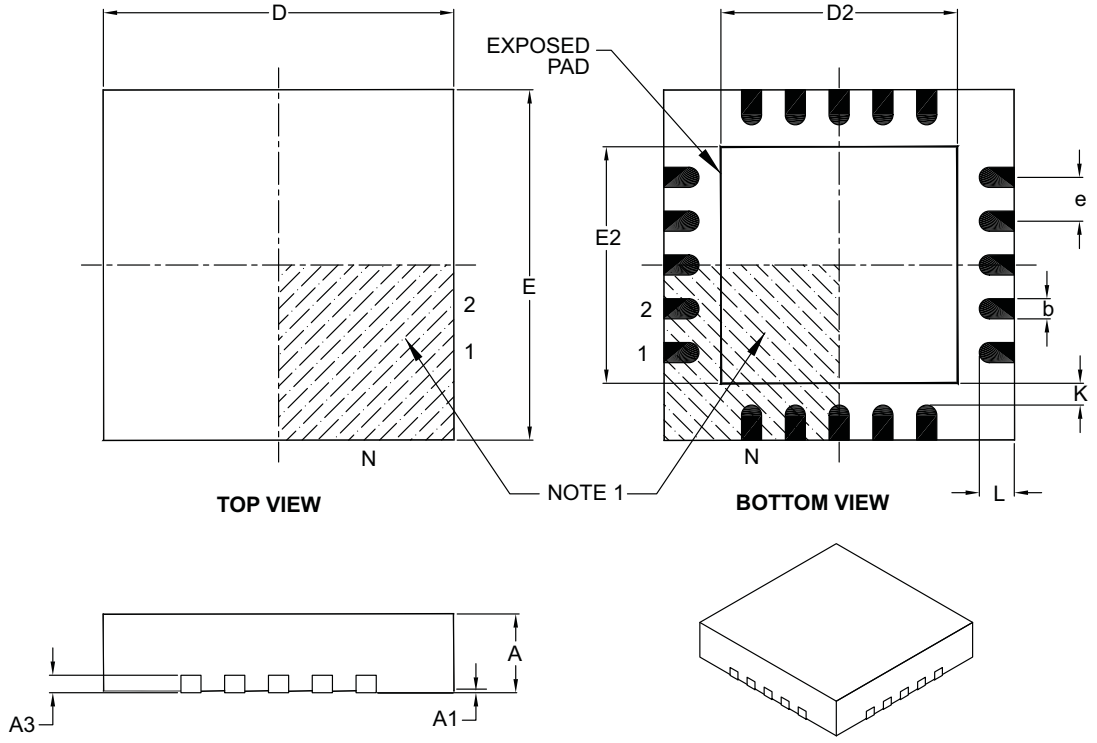
1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2094A

### 20-Lead Plastic Quad Flat, No Lead Package (ML) – 4x4x0.9 mm Body [QFN]

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging







| Units | | MILLIMETERS | | |
|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX |
| Number of Pins | N | | 20 | |
| Pitch | e | | 0.50 BSC | |
| Overall Height | A | 0.80 | 0.90 | 1.00 |
| Standoff | A1 | 0.00 | 0.02 | 0.05 |
| Contact Thickness | A3 | | 0.20 REF | |
| Overall Width | E | | 4.00 BSC | |
| Exposed Pad Width | E2 | 2.60 | 2.70 | 2.80 |
| Overall Length | D | | 4.00 BSC | |
| Exposed Pad Length | D2 | 2.60 | 2.70 | 2.80 |
| Contact Width | b | 0.18 | 0.25 | 0.30 |
| Contact Length | L | 0.30 | 0.40 | 0.50 |
| Contact-to-Exposed Pad | K | 0.20 | – | – |

**Notes:**

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Package is saw singulated.
3. Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

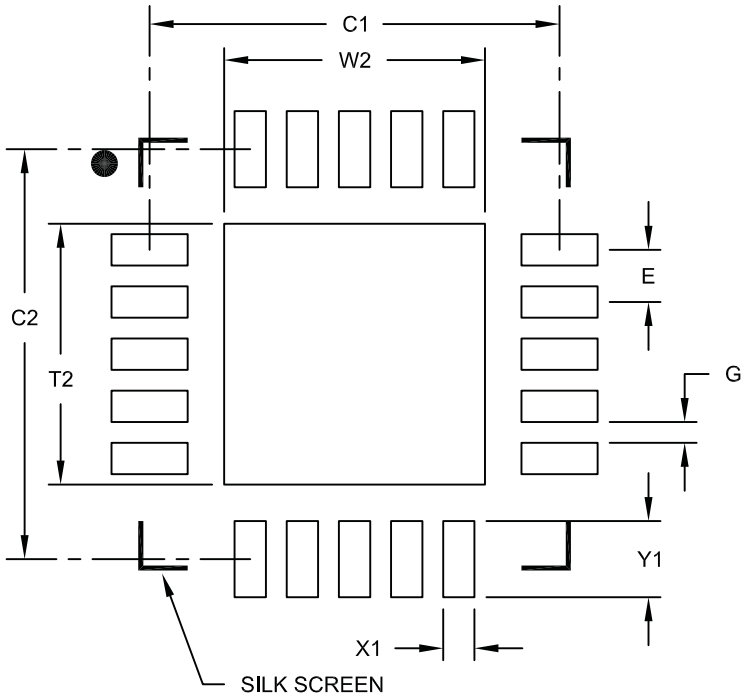
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-126B


© 2013-2015 Microchip Technology Inc.

20-Lead Plastic Quad Flat, No Lead Package (ML) - 4x4 mm Body [QFN]
With 0.40 mm Contact Length

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging




RECOMMENDED LAND PATTERN

| Units | | MILLIMETERS | | |
|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX |
| Contact Pitch | E | | 0.50 BSC | |
| Optional Center Pad Width | W2 | | | 2.50 |
| Optional Center Pad Length | T2 | | | 2.50 |
| Contact Pad Spacing | C1 | | 3.93 | |
| Contact Pad Spacing | C2 | | 3.93 | |
| Contact Pad Width | X1 | | | 0.30 |
| Contact Pad Length | Y1 | | | 0.73 |
| Distance Between Pads | G | 0.20 | | |

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2126A

## APPENDIX A: DATA SHEET REVISION HISTORY

### Revision A (10/2013)

Initial release of this document.

### Revision B (02/2015)

Changed from Preliminary to Final data sheet.

Added Section 3.2: High-Endurance Flash.

Removed Figures 6-3, 6-4, 6-6, 6-10.

Removed Sections 6.2.1.2, 6.2.1.3, 6.2.1.6, 6.3.2, 6.4, 6.4.2, 6.5.

Replaced Section 18.0.

Updated Figures 2-1, 6-1, 6-2, 6-10, 7-2, 7-3, 14-1, 16-1, 17-1.

Updated PIC16(L)F170x Family Types Table.

Updated Registers 4-1, 6-1, 6-2, 14-1, 16-1, 16-2, 17-1.

Updated Section 23.0 from SSP to SSPx.

Updated Sections 6.1, 6.2, 6.2.1, 6.2.1.1, 6.2.2, 6.2.2.1, 6.2.2.2, 6.2.2.4, 6.2.2.6, 6.3.1, 8.2.2, 11.3.6, 16.1.3, 17.0, 17.1, 17.1.1.

Updated Tables 1, 2, 1-2, 1-3, 3-1, 6-1, 6-2, 26-18.

### Revision C (09/2015)

Added High Endurance Flash Data Memory (HEF) bullet on front page. Updated PIC16(L)F170x Family Types Table. Updated graphs 27-63 through 27-70.


© 2013-2015 Microchip Technology Inc.

## THE MICROCHIP WEB SITE

Microchip provides online support via our web 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. Under "Support", 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

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://www.microchip.com/support**

## 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](1) | - | X | /XX | XXX |
|---|---|---|---|---|---|
| Device | Tape and Reel Option | | Temperature Range | Package | Pattern |

| | | |
|---|---|---|
| **Device:** | PIC16F1703, PIC16LF1703, PIC16F1707, 7PIC16LF1707 | |
| **Tape and Reel Option:** | Blank<br>T | = Standard packaging (tube or tray)<br>= Tape and Reel(1) |
| **Temperature Range:** | I<br>E | = -40°C to +85°C (Industrial)<br>= -40°C to +125°C (Extended) |
| **Package:(2)** | ML<br>P<br>SL<br>SO<br>SS<br>ST | = QFN<br>= PDIP<br>= SOIC<br>= SOIC<br>= SSOP<br>= TSSOP |
| **Pattern:** | QTP, SQTP, Code or Special Requirements (blank otherwise) | |

**Examples:**

a) PIC16LF1703- I/P
Industrial temperature
PDIP package

b) PIC16F1707- E/SS
Extended temperature,
SSOP package

**Note 1:** Tape and Reel identifier only appears in the catalog part number description. This identifier is used for ordering purposes and is not printed on the device package. Check with your Microchip Sales Office for package availability with the Tape and Reel option.

**2:** Small form-factor packaging options may be available. Please check www.microchip.com/packaging for small-form factor package availability, or contact your local Sales Office.


© 2013-2015 Microchip Technology Inc.

**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.

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE**.** Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

**Trademarks**

The Microchip name and logo, the Microchip logo, dsPIC, FlashFlex, flexPWR, JukeBlox, KEELOQ, KEELOQ logo, Kleer, LANCheck, MediaLB, MOST, MOST logo, MPLAB, OptoLyzer, PIC, PICSTART, PIC$^{32}$ logo, RightTouch, SpyNIC, SST, SST Logo, SuperFlash and UNI/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

The Embedded Control Solutions Company and mTouch are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, BodyCom, chipKIT, chipKIT logo, CodeGuard, dsPICDEM, dsPICDEM.net, ECAN, In-Circuit Serial Programming, ICSP, Inter-Chip Connectivity, KleerNet, KleerNet logo, MiWi, motorBench, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, RightTouch logo, REAL ICE, SQI, Serial Quad I/O, Total Endurance, TSHARC, USBCheck, VariSense, ViewSpan, WiperLock, Wireless DNA, and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

Silicon Storage Technology is a registered trademark of Microchip Technology Inc. in other countries.

GestIC is a registered trademark of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

© 2013-2015, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

ISBN: 978-1-63277-846-8

**QUALITY MANAGEMENT SYSTEM**
**CERTIFIED BY DNV**
**═ ISO/TS 16949 ═**

*Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, 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.*

# 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://www.microchip.com/support
Web Address:
www.microchip.com

**Atlanta**
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

**Austin, TX**
Tel: 512-257-3370

**Boston**
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

**Chicago**
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

**Cleveland**
Independence, OH
Tel: 216-447-0464
Fax: 216-447-0643

**Dallas**
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

**Detroit**
Novi, MI
Tel: 248-848-4000

**Houston, TX**
Tel: 281-894-5983

**Indianapolis**
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453

**Los Angeles**
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

**New York, NY**
Tel: 631-435-6000

**San Jose, CA**
Tel: 408-735-9110

**Canada - Toronto**
Tel: 905-673-0699
Fax: 905-673-6509

## ASIA/PACIFIC

**Asia Pacific Office**
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon

**Hong Kong**
Tel: 852-2943-5100
Fax: 852-2401-3431

**Australia - Sydney**
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

**China - Beijing**
Tel: 86-10-8569-7000
Fax: 86-10-8528-2104

**China - Chengdu**
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

**China - Chongqing**
Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

**China - Dongguan**
Tel: 86-769-8702-9880

**China - Hangzhou**
Tel: 86-571-8792-8115
Fax: 86-571-8792-8116

**China - Hong Kong SAR**
Tel: 852-2943-5100
Fax: 852-2401-3431

**China - Nanjing**
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

**China - Qingdao**
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

**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: 86-755-8864-2200
Fax: 86-755-8203-1760

**China - Wuhan**
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

**China - Xian**
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

## ASIA/PACIFIC

**China - Xiamen**
Tel: 86-592-2388138
Fax: 86-592-2388130

**China - Zhuhai**
Tel: 86-756-3210040
Fax: 86-756-3210049

**India - Bangalore**
Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

**India - New Delhi**
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

**India - Pune**
Tel: 91-20-3019-1500

**Japan - Osaka**
Tel: 81-6-6152-7160
Fax: 81-6-6152-9310

**Japan - Tokyo**
Tel: 81-3-6880- 3770
Fax: 81-3-6880-3771

**Korea - Daegu**
Tel: 82-53-744-4301
Fax: 82-53-744-4302

**Korea - Seoul**
Tel: 82-2-554-7200
Fax: 82-2-558-5932 or
82-2-558-5934

**Malaysia - Kuala Lumpur**
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

**Malaysia - Penang**
Tel: 60-4-227-8870
Fax: 60-4-227-4068

**Philippines - Manila**
Tel: 63-2-634-9065
Fax: 63-2-634-9069

**Singapore**
Tel: 65-6334-8870
Fax: 65-6334-8850

**Taiwan - Hsin Chu**
Tel: 886-3-5778-366
Fax: 886-3-5770-955

**Taiwan - Kaohsiung**
Tel: 886-7-213-7828

**Taiwan - Taipei**
Tel: 886-2-2508-8600
Fax: 886-2-2508-0102

**Thailand - Bangkok**
Tel: 66-2-694-1351
Fax: 66-2-694-1350

## EUROPE

**Austria - Wels**
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

**Denmark - Copenhagen**
Tel: 45-4450-2828
Fax: 45-4485-2829

**France - Paris**
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

**Germany - Dusseldorf**
Tel: 49-2129-3766400

**Germany - Karlsruhe**
Tel: 49-721-625370

**Germany - Munich**
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

**Italy - Milan**
Tel: 39-0331-742611
Fax: 39-0331-466781

**Italy - Venice**
Tel: 39-049-7625286

**Netherlands - Drunen**
Tel: 31-416-690399
Fax: 31-416-690340

**Poland - Warsaw**
Tel: 48-22-3325737

**Spain - Madrid**
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

**Sweden - Stockholm**
Tel: 46-8-5090-4654

**UK - Wokingham**
Tel: 44-118-921-5800
Fax: 44-118-921-5820

07/14/15


© 2013-2015 Microchip Technology Inc.

# Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

Microchip:

PIC16F1703-I/SL PIC16F1707-E/SO PIC16F1707-E/SS PIC16F1707-I/P PIC16F1707-I/SS PIC16F1707-E/ML PIC16F1707-E/P PIC16F1703-I/P PIC16F1703-E/ML PIC16F1707-I/ML PIC16F1703T-I/ST PIC16F1707T-I/SO PIC16F1703-E/P PIC16F1703T-I/SL PIC16F1707-I/SO PIC16F1703-E/ST PIC16F1703-E/SL PIC16F1703-I/ST PIC16F1703-I/ML PIC16F1707T-I/SS PIC16F1703T-I/ML PIC16LF1703T-I/ST PIC16LF1707-E/SS PIC16LF1707-I/ML PIC16LF1703-I/SL PIC16LF1707-E/SO PIC16LF1703-I/P PIC16LF1707-E/P PIC16LF1703-E/ST PIC16LF1707-E/ML PIC16LF1707-I/P PIC16LF1703-I/ST PIC16LF1703-I/ML PIC16LF1707-I/SS PIC16LF1703-E/ML PIC16LF1707T-I/SS PIC16LF1703-E/SL PIC16LF1707T-I/SO PIC16LF1707T-I/ML PIC16LF1703-E/P PIC16LF1703T-I/SL PIC16F1707T-I/ML PIC16LF1707-I/SO PIC16LF1703T-I/ML

Общество с ограниченной ответственностью «МосЧип» ИНН 7719860671 / КПП 771901001
Адрес: 105318, г.Москва, ул.Щербаковская д.3, офис 1107

**Данный компонент на территории Российской Федерации**

**Вы можете приобрести в компании MosChip.**

Для оперативного оформления запроса Вам необходимо перейти по данной ссылке:

http://moschip.ru/get-element

Вы можете разместить у нас заказ для любого Вашего проекта, будь то серийное производство или разработка единичного прибора.

В нашем ассортименте представлены ведущие мировые производители активных и пассивных электронных компонентов.

Нашей специализацией является поставка электронной компонентной базы двойного назначения, продукции таких производителей как XILINX, Intel (ex.ALTERA), Vicor, Microchip, Texas Instruments, Analog Devices, Mini-Circuits, Amphenol, Glenair.

Сотрудничество с глобальными дистрибьюторами электронных компонентов, предоставляет возможность заказывать и получать с международных складов практически любой перечень компонентов в оптимальные для Вас сроки.

На всех этапах разработки и производства наши партнеры могут получить квалифицированную поддержку опытных инженеров.

Система менеджмента качества компании отвечает требованиям в соответствии с ГОСТ Р ИСО 9001, ГОСТ РВ 0015-002 и ЭС РД 009

**Офис по работе с юридическими лицами:**

105318, г.Москва, ул.Щербаковская д.3, офис 1107, 1118, ДЦ «Щербаковский»

Телефон: +7 495 668-12-70 (многоканальный)

Факс: +7 495 668-12-70 (доб.304)

E-mail: info@moschip.ru

Skype отдела продаж:
moschip.ru
moschip.ru_4
moschip.ru_6
moschip.ru_9