# PIC16F753/HV753

## 14/16-Pin, Flash-Based 8-Bit CMOS Microcontrollers

### High-Performance RISC CPU

- Only 35 Instructions to Learn:
  - All single-cycle instructions except branches
- Operating Speed:
  - DC – 20 MHz clock input
  - DC – 200 ns instruction cycle
- 2048 x 14 On-chip Flash Program Memory
- Self-Read/Write Program Memory
- 128 x 8 General Purpose Registers (SRAM)
- Interrupt Capability
- 8-Level Deep Hardware Stack
- Direct, Indirect and Relative Addressing modes

### Microcontroller Features

- Precision Internal Oscillator:
  - Factory calibrated to ±1%, typical
  - Software selectable frequency: 8 MHz, 4 MHz, 1 MHz or 31 kHz
  - Software tunable
- Power-Saving Sleep mode
- Voltage Range (PIC16F753):
  - 2.0V to 5.5V
- Shunt Voltage Regulator (PIC16HV753):
  - 2.0V to user defined
  - 5-volt regulation
  - 1 mA to 50 mA shunt range
- Multiplexed Master Clear with Pull-up/Input Pin
- Interrupt-on-Change Pins
- Individually Programmable Weak Pull-ups
- Power-on Reset (POR)
- Power-up Timer (PWRT)
- Brown-out Reset (BOR)
- Watchdog Timer (WDT) with Internal Oscillator for Reliable Operation
- Industrial and Extended Temperature Range
- High Endurance Flash:
  - 100,000 write Flash endurance
  - Flash retention: >40 years
- Programmable Code Protection
- In-Circuit Debug (ICD) via Two Pins
- In-Circuit Serial Programming™ (ICSP™) via Two Pins

### eXtreme Low-Power (XLP) Features

- Sleep Current:
  - 50 nA @ 2.0V, typical
- Operating Current:
  - 11 uA @ 32 kHz, 2.0V, typical
  - 260 uA @ 4 MHz, 2.0V, typical
- Watchdog Timer Current:
- <1 uA @ 2.0V, typical

### Peripheral Features

- 11 I/O Pins and one Input-only Pin
- High Current Source/Sink:
  - 50 mA I/O, (two pins)
  - 25 mA I/O, (nine pins)
- Two High-Speed Analog Comparator modules:
  - 50 ns response time
  - Fixed Voltage Reference (FVR)
  - Programmable on-chip voltage reference via integrated 9-bit DAC
  - Internal/external inputs and outputs (selectable)
  - Built-in Hysteresis (software selectable)
- A/D Converter:
  - 10-bit resolution
  - Eight external channels
  - Two internal reference voltage channels
- Operational Amplifier:
  - Three terminal operations
  - Internal connections to DAC and FVR
- Digital-to-Analog Converter (DAC):
  - 9-bit resolution
  - Full Range output
  - 4 mV steps @ 2.0V (Limited Range)
- Fixed Voltage Reference (FVR), 1.2V Reference
- Capture, Compare, PWM (CCP) module:
  - 16-bit Capture, max. resolution = 12.5 ns
  - 16-bit Compare, max. resolution = 200 ns
  - 10-bit PWM, max. frequency = 20 kHz
- Timer0: 8-Bit Timer/Counter with 8-Bit Prescaler
- Enhanced Timer1:
  - 16-bit Timer/Counter with Prescaler
  - External Timer1 Gate (count enable)
  - Four Selectable Clock sources
- Timer2: 8-Bit Timer/Counter with Prescaler
  - 8-Bit Period Register and Postscaler
- Two Hardware Limit Timers (HLT):
  - 8-bit Timer with Prescaler
  - 8-bit period register and postscaler
  - Asynchronous H/W Reset sources

- Complementary Output Generator (COG):
  - Complementary Waveforms from selectable sources
  - Two I/O (50 mA) for direct MOSFET drive
  - Rising and/or Falling edge dead-band control
  - Phase control, Blanking control
  - Auto-shutdown
  - Slope Compensation Circuit for use with SMPS power supplies

**TABLE 1: PIC16F753/HV753 FAMILY TYPES**

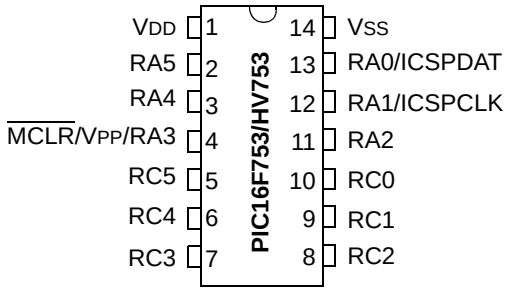
| Device | Data Sheet Index | Program Memory Flash (words) | Self-Read/Write Flash Memory | Data SRAM (bytes) | I/Os(2) | 10-bit ADC (ch) | Comparators | Timers (8/16-bit) | CCP | Complementary Output Generator (COG) | DAC | Op Amp | Shunt Regulator | Debug(1) | XLP |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PIC12F752 | (1) | 1K | Y | 64 | 6 | 4 | 2 | 3/1 | 1 | Y | 5-bit | N | N | H | Y |
| PIC12HV752 | (1) | 1K | Y | 64 | 6 | 4 | 2 | 3/1 | 1 | Y | 5-bit | N | Y | H | Y |
| PIC16F753 | (2) | 2K | Y | 128 | 12 | 8 | 2 | 3/1 | 1 | Y | 9-bit | Y | N | I/H | Y |
| PIC16HV753 | (2) | 2K | Y | 128 | 12 | 8 | 2 | 3/1 | 1 | Y | 9-bit | Y | Y | I/H | Y |

**Note 1:** I - Debugging, Integrated on Chip; H - Debugging, Requires Debug Header.
**2:** One pin is input-only.

**Data Sheet Index:** (Unshaded devices are described in this document.)
**1:** DS40001576 PIC12F752/HV752 Data Sheet, 8-Pin Flash-Based, 8-Bit CMOS Microcontrollers.
**2:** DS40001709 PIC16F753/HV753 Data Sheet, 14/16-Pin Flash-based, 8-Bit CMOS Microcontrollers.

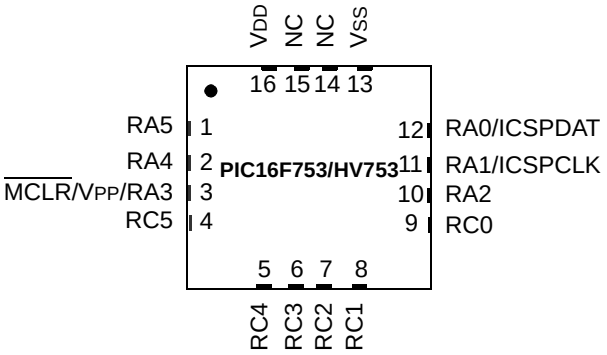
**Note:** For other small form-factor package availability and marking information, please visit **http://www.microchip.com/packaging** or contact your local sales office.

**FIGURE 1: 14-PIN PDIP, SOIC, TSSOP DIAGRAM**




**Note:** See Table 2 for location of all peripheral functions.

**FIGURE 2: 16-PIN QFN (4X4) DIAGRAM**




**Note:** See Table 2 for location of all peripheral functions.

**TABLE 2: 14/16-PIN ALLOCATION TABLE FOR PIC16F753/HV753**

| I/O | 14-Pin PDIP/SOIC/TSSOP | 16-Pin QFN | ADC | Reference | Op Amp | Comparator | Timer | CCP | Interrupt | Pull-up | Slope Compensation | Basic |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RA0 | 13 | 12 | AN0 | FVROUT DACOUT | — | C1IN0+ | — | — | IOC | Y | — | ICSPDAT |
| RA1 | 12 | 11 | AN1 | VREF+ FVRIN | — | C1IN0- C2IN0- | — | — | IOC | Y | — | ICSPCLK |
| RA2 | 11 | 10 | AN2 | COG1FLT | — | C1OUT | T0CKI | — | INT IOC | Y | — | — |
| RA3 | 4 | 3 | — | — | — | — | T1G(2) | — | IOC | Y | — | $\overline{\text{MCLR}}$/ VPP |
| RA4 | 3 | 2 | AN3 | — | — | — | T1G(1) | — | IOC | Y | — | CLKOUT |
| RA5 | 2 | 1 | — | — | — | — | T1CKI | — | IOC | Y | — | CLKIN |
| RC0 | 10 | 9 | AN4 | — | OPA1IN+ | C2IN0+ | — | — | IOC | — | — | — |
| RC1 | 9 | 8 | AN5 | — | OPA1IN- | C1IN1- C2IN1- | — | — | IOC | — | — | — |
| RC2 | 8 | 7 | AN6 | — | OPA1OUT | C1IN2- C2IN2- | — | — | IOC | — | SLPCIN | — |
| RC3 | 7 | 6 | AN7 | — | — | C1IN3- C2IN3- | — | — | IOC | — | — | — |
| RC4 | 6 | 5 | — | COG1OUT1 | — | C2OUT | — | — | IOC | — | — | — |
| RC5 | 5 | 4 | — | COG1OUT0 | — | — | — | CCP1 | IOC | — | — | — |
| VDD | 1 | 16 | — | — | — | — | — | — | — | — | — | VDD |
| VSS | 14 | 13 | — | — | — | — | — | — | — | — | — | VSS |

**Note 1:** Default location for peripheral pin function. Alternate location can be selected using the APFCON register.
**2:** Alternate location for peripheral pin function selected by the APFCON register.

## Table of Contents

## TO OUR VALUED CUSTOMERS

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

If you have any questions or comments regarding this publication, please contact the Marketing Communications Department via E-mail at **docerrors@microchip.com**. 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 Website 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 Website; **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 website at **www.microchip.com** to receive the most current information on all of our products.

## 1.0 DEVICE OVERVIEW

The PIC16F753/HV753 devices are covered by this data sheet. They are available in 14-pin PDIP, SOIC, TSSOP and 16-pin QFN packages.

Block Diagrams and pinout descriptions of the devices are shown in Figure 1-1 and Table 1-1.

**FIGURE 1-1: PIC16F753/HV753 BLOCK DIAGRAM**

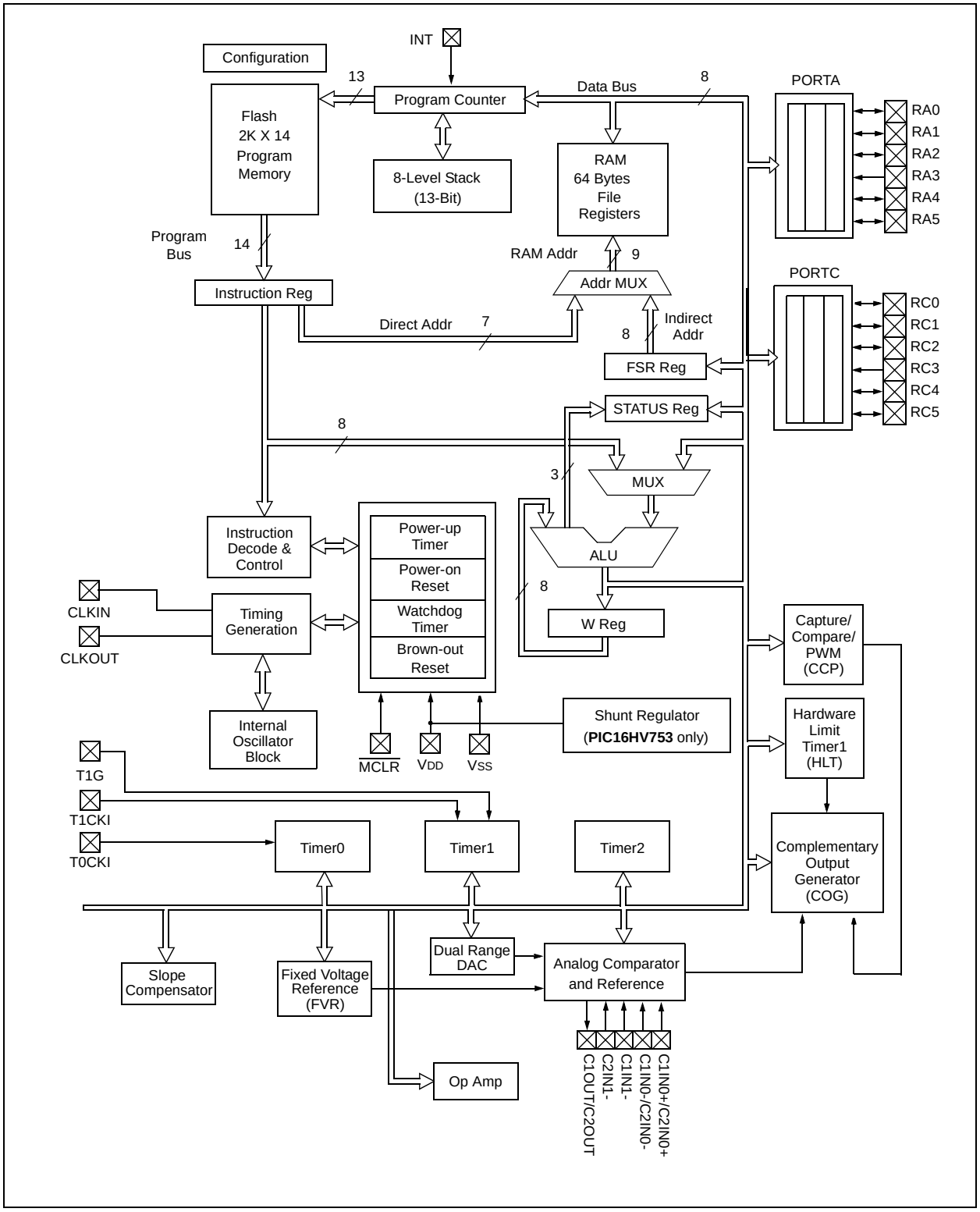


**TABLE 1-1: PIC16F753/HV753 PINOUT DESCRIPTION**

| Name | Function | Input Type | Output Type | Description |
|---|---|---|---|---|
| RA0/AN0/C1IN0+/DACOUT/ FVROUT/ICSPDAT | RA0 | TTL | HP | General purpose I/O with IOC and WPU. |
| | AN0 | AN | — | A/D Channel 0 input. |
| | C1IN0+ | AN | — | Comparator C1 positive input. |
| | DACOUT | — | AN | DAC unbuffered Voltage Reference output. |
| | FVROUT | — | AN | DAC/FVR buffered Voltage Reference output. |
| | ICSPDAT | ST | HP | Serial Programming Data I/O. |
| RA1/AN1/C1IN0-/C2IN0-/ VREF+/FVRIN/ICSPCLK | RA1 | TTL | CMOS | General purpose I/O with IOC and WPU. |
| | AN1 | AN | — | A/D Channel 1 input. |
| | C1IN0- | AN | — | Comparator C1 negative input. |
| | C2IN0- | AN | — | Comparator C2 negative input. |
| | VREF+ | AN | — | A/D Positive Voltage Reference input. |
| | FVRIN | AN | — | Voltage reference input. |
| | ICSPCLK | ST | — | Serial Programming Clock. |
| RA2/AN2/INT/C1OUT/ T0CKI/COG1FLT | RA2 | ST | HP | General purpose I/O with IOC and WPU. |
| | AN2 | AN | — | A/D Channel 2 input. |
| | INT | ST | — | External interrupt. |
| | C1OUT | — | HP | Comparator C1 output. |
| | T0CKI | ST | — | Timer0 clock input. |
| | COG1FLT | ST | — | COG auto-shutdown fault input. |
| RA3(1)/T1G(3)/VPP/$\overline{\text{MCLR}}$(4) | RA3 | TTL | — | General purpose input with WPU. |
| | T1G | ST | — | Timer1 Gate input. |
| | VPP | HV | — | Programming voltage. |
| | $\overline{\text{MCLR}}$ | ST | — | Master Clear w/internal pull-up. |
| RA4/AN3/T1G(2)/CLKOUT | RA4 | TTL | CMOS | General purpose I/O with IOC and WPU. |
| | AN3 | AN | — | A/D Channel 3 input. |
| | T1G | ST | — | Timer1 Gate input. |
| | CLKOUT | — | CMOS | FOSC/4 output. |
| RA5/T1CKI/COG1OUT0(3)/ C2IN1-/CLKIN | RA5 | TTL | CMOS | General purpose I/O with IOC and WPU. |
| | T1CKI | ST | — | Timer1 clock input. |
| | CLKIN | ST | — | External Clock input (EC mode). |
| RC0/AN4/OPA1IN+/C2IN0+ | RC0 | TTL | CMOS | General purpose I/O with IOC and WPU. |
| | AN4 | AN | — | A/D Channel 4 input. |
| | OPA1IN+ | AN | — | Op amp positive input. |
| | C2IN0+ | AN | — | Comparator C2 positive input. |
| RC1/AN5/OPA1IN-/C1IN1-/ C2IN1- | RC1 | TTL | CMOS | General purpose I/O with IOC and WPU. |
| | AN5 | AN | — | A/D Channel 5 input. |
| | OPA1IN- | AN | — | Op amp negative input. |
| | C1IN1- | AN | — | Comparator C1 negative input. |
| | C2IN1- | AN | — | Comparator C2 negative input. |

**Legend:** AN = Analog input or output CMOS = CMOS compatible input or output
TTL = TTL compatible input ST = Schmitt Trigger input with CMOS levels
HP = High Power HV = High Voltage
* Alternate pin function.

**Note 1:** Input only.
**2:** Default pin function via the APFCON register.
**3:** Alternate pin function via the APFCON register.
**4:** RA3 pull-up is enabled when pin is configured as $\overline{\text{MCLR}}$ in Configuration Word.

**TABLE 1-1: PIC16F753/HV753 PINOUT DESCRIPTION (CONTINUED)**

| Name | Function | Input Type | Output Type | Description |
|---|---|---|---|---|
| RC2/AN6/SLPCIN/ OPA1OUT/C1IN2-/C2IN2- | RC2 | TTL | CMOS | General purpose I/O with IOC and WPU. |
| | AN6 | AN | — | A/D Channel 6 input. |
| | OPA1OUT | AN | HP | Op amp output. |
| | C1IN2- | AN | — | Comparator C1 negative input. |
| | C2IN2- | AN | — | Comparator C2 negative input. |
| RC3/AN7/C1IN3-/C2IN3- | RC3 | TTL | CMOS | General purpose I/O with IOC and WPU. |
| | AN7 | AN | — | A/D Channel 7 input. |
| | C1IN3- | AN | — | Comparator C1 negative input. |
| | C2IN3- | AN | — | Comparator C2 negative input. |
| RC4/COG1OUT1/C2OUT | RC4 | TTL | CMOS | General purpose I/O with IOC and WPU. |
| | COG1OUT1 | — | CMOS | COG output Channel 1. |
| | C2OUT | — | HP | Comparator C2 output. |
| RC5/COG1OUT0/CCP1 | RC5 | TTL | CMOS | General purpose I/O with IOC and WPU. |
| | COG1OUT0 | — | CMOS | COG output Channel 0. |
| | CCP1 | — | HP | Capture/Compare/PWM 1. |
| VDD | VDD | Power | — | Positive supply. |
| VSS | VSS | Power | — | Ground reference. |

**Legend:** AN = Analog input or output CMOS = CMOS compatible input or output
TTL = TTL compatible input ST = Schmitt Trigger input with CMOS levels
HP = High Power HV = High Voltage
* Alternate pin function.

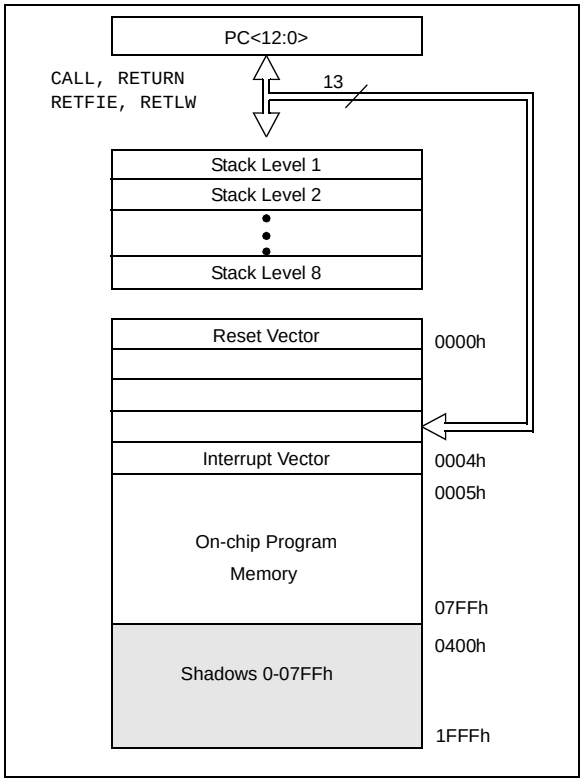
**Note 1:** Input only.
**2:** Default pin function via the APFCON register.
**3:** Alternate pin function via the APFCON register.
**4:** RA3 pull-up is enabled when pin is configured as $\overline{\text{MCLR}}$ in Configuration Word.

# 2.0 MEMORY ORGANIZATION

## 2.1 Program Memory Organization

The PIC16F753/HV753 has a 13-bit program counter capable of addressing an 8K x 14 program memory space. Only the first 2K x 14 (0000h-07FFh) is physically implemented. Accessing a location above these boundaries will cause a wrap-around within the first 2K x 14 space for PIC16F753/HV753. The Reset vector is at 0000h and the interrupt vector is at 0004h (see Figure 2-1).

**FIGURE 2-1: PROGRAM MEMORY MAP AND STACK FOR THE PIC16F753/HV753**




## 2.2 Data Memory Organization

The data memory (see Figure 2-2) is partitioned into four banks, which contain the General Purpose Registers (GPR) and the Special Function Registers (SFR). The Special Function Registers are located in the first 32 locations of each bank. Register locations 40h-6Fh in Bank 0 are General Purpose Registers, implemented as static RAM. Register locations 70h-7Fh in Bank 0 are Common RAM and shared as the last 16 addresses in all Banks. All other RAM is unimplemented and returns '0' when read. The RP<1:0> bits of the STATUS register are the bank select bits.

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

### 2.2.1 GENERAL PURPOSE REGISTER FILE

The register file is organized as 64 x 8 in the PIC16F753/HV753. Each register is accessed, either directly or indirectly, through the File Select Register (FSR) (see **Section 2.5 "Indirect Addressing, INDF and FSR Registers"**).

### 2.2.2 SPECIAL FUNCTION REGISTERS

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

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

**FIGURE 2-2: DATA MEMORY MAP OF THE PIC16F753/HV753**

| BANK 0 | | BANK 1 | | BANK 2 | | BANK 3 | |
|---|---|---|---|---|---|---|---|
| INDF | 00h | INDF | 80h | INDF | 100h | INDF | 180h |
| TMR0 | 01h | OPTION_REG | 81h | TMR0 | 101h | OPTION_REG | 181h |
| PCL | 02h | PCL | 82h | PCL | 102h | PCL | 182h |
| STATUS | 03h | STATUS | 83h | STATUS | 103h | STATUS | 183h |
| FSR | 04h | FSR | 84h | FSR | 104h | FSR | 184h |
| PORTA | 05h | TRISA | 85h | LATA | 105h | ANSELA | 185h |
| — | 06h | — | 86h | — | 106h | — | 186h |
| PORTC | 07h | TRISC | 87h | LATC | 107h | ANSELC | 187h |
| IOCAF | 08h | IOCAP | 88h | IOCAN | 108h | APFCON | 188h |
| IOCCF | 09h | IOCCP | 89h | IOCCN | 109h | OSCTUNE | 189h |
| PCLATH | 0Ah | PCLATH | 8Ah | PCLATH | 10Ah | PCLATH | 18Ah |
| INTCON | 0Bh | INTCON | 8Bh | INTCON | 10Bh | INTCON | 18Bh |
| PIR1 | 0Ch | PIE1 | 8Ch | WPUA | 10Ch | PMCON1 | 18Ch |
| PIR2 | 0Dh | PIE2 | 8Dh | WPUC | 10Dh | PMCON2 | 18Dh |
| — | 0Eh | — | 8Eh | SLRCONC | 10Eh | PMADRL | 18Eh |
| TMR1L | 0Fh | OSCCON | 8Fh | PCON | 10Fh | PMADRH | 18Fh |
| TMR1H | 10h | FVR1CON0 | 90h | TMR2 | 110h | PMDATL | 190h |
| T1CON | 11h | DAC1CON0 | 91h | PR2 | 111h | PMDATH | 191h |
| T1GCON | 12h | DAC1REFL | 92h | T2CON | 112h | COG1PHR | 192h |
| CCPR1L | 13h | DAC1REFH | 93h | HLTMR1 | 113h | COG1PHF | 193h |
| CCPR1H | 14h | — | 94h | HLTPR1 | 114h | COG1BKR | 194h |
| CCP1CON | 15h | — | 95h | HLT1CON0 | 115h | COG1BKF | 195h |
| — | 16h | OPA1CON0 | 96h | HLT1CON1 | 116h | COG1DBR | 196h |
| — | 17h | — | 97h | HLTMR2 | 117h | COG1DBF | 197h |
| — | 18h | — | 98h | HLTPR2 | 118h | COG1CON0 | 198h |
| — | 19h | — | 99h | HLT2CON0 | 119h | COG1CON1 | 199h |
| — | 1Ah | — | 9Ah | HLT2CON1 | 11Ah | COG1RIS | 19Ah |
| — | 1Bh | CM2CON0 | 9Bh | — | 11Bh | COG1RSIM | 19Bh |
| ADRESL | 1Ch | CM2CON1 | 9Ch | — | 11Ch | COG1FIS | 19Ch |
| ADRESH | 1Dh | CM1CON0 | 9Dh | — | 11Dh | COG1FSIM | 19Dh |
| ADCON0 | 1Eh | CM1CON1 | 9Eh | SLPCCON0 | 11Eh | COG1ASD0 | 19Eh |
| ADCON1 | 1Fh | CMOUT | 9Fh | SLPCCON1 | 11Fh | COG1ASD1 | 19Fh |
| General Purpose Register 80 Bytes | 20h – 6Fh | General Purpose Register 32 Bytes | A0h – BFh | Unimplemented Read as '0' | 120h – 16Fh | Unimplemented Read as '0' | 1A0h – 1EFh |
| | | Unimplemented Read as '0' | C0h – EFh | | | | |
| Common RAM 16 Bytes | 70h – 7Fh | Common RAM (Accesses 70h – 7Fh) | F0h – FFh | Common RAM (Accesses 70h – 7Fh) | 170h – 17Fh | Common RAM (Accesses 70h – 7Fh) | 1F0h – 1FFh |

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

**TABLE 2-1: PIC16F753/HV753 SPECIAL REGISTERS SUMMARY BANK 0**

| Addr | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR/BOR | Value on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Bank 0** | | | | | | | | | | | |
| 00h | INDF | INDF<7:0> | | | | | | | | xxxx xxxx | xxxx xxxx |
| 01h | TMR0 | TMR0<7:0> | | | | | | | | xxxx xxxx | uuuu uuuu |
| 02h | PCL | PCL<7:0> | | | | | | | | 0000 0000 | 0000 0000 |
| 03h | STATUS | IRP | RP1 | RP0 | $\overline{TO}$ | $\overline{PD}$ | Z | DC | C | 0001 1xxx | 000q quuu |
| 04h | FSR | FSR<7:0> | | | | | | | | xxxx xxxx | uuuu uuuu |
| 05h | PORTA | — | — | RA5 | RA4 | RA3 | RA2 | RA1 | RA0 | --xx xxxx | --uu uuuu |
| 06h | — | Unimplemented | | | | | | | | — | — |
| 07h | PORTC | — | — | RC5 | RC4 | RC3 | RC2 | RC1 | RC0 | --xx xxxx | --uu uuuu |
| 08h | IOCAF | — | — | IOCAF5 | IOCAF4 | IOCAF3 | IOCAF2 | IOCAF1 | IOCAF0 | --00 0000 | --00 0000 |
| 09h | IOCCF | — | — | IOCCF5 | IOCCF4 | IOCCF3 | IOCCF2 | IOCCF1 | IOCCF0 | --00 0000 | --00 0000 |
| 0Ah | PCLATH | — | — | — | PCLATH<4:0> | | | | | ---0 0000 | ---0 0000 |
| 0Bh | INTCON | GIE | PEIE | T0IE | INTE | IOCIE | T0IF | INTF | IOCIF | 0000 0000 | 0000 0000 |
| 0Ch | PIR1 | TMR1GIF | ADIF | — | — | HLTMR2IF | HLTMR1IF | TMR2IF | TMR1IF | 00--0000 | 00--0000 |
| 0Dh | PIR2 | — | — | C2IF | C1IF | — | COG1IF | — | CCP1IF | --00 -0-0 | --00 -0-0 |
| 0Eh | — | Unimplemented | | | | | | | | — | — |
| 0Fh | TMR1L | TMR1L<7:0> | | | | | | | | xxxx xxxx | uuuu uuuu |
| 10h | TMR1H | TMR1H<7:0> | | | | | | | | xxxx xxxx | uuuu uuuu |
| 11h | T1CON | TMR1CS<1:0> | | T1CKPS<1:0> | | T1OSCEN | $\overline{T1SYNC}$ | — | TMR1ON | 0000 00-0 | 0000 00-0 |
| 12h | T1GCON | TMR1GE | T1GPOL | T1GTM | T1GSPM | T1GGO/$\overline{DONE}$ | T1GVAL | T1GSS<1:0> | | 0000 0x00 | 0000 0x00 |
| 13h | CCPR1L | CCPR1L<7:0> | | | | | | | | xxxx xxxx | uuuu uuuu |
| 14h | CCPR1H | CCPR1H<7:0> | | | | | | | | xxxx xxxx | uuuu uuuu |
| 15h | CCP1CON | — | — | DC1B<1:0> | | CCP1M<3:0> | | | | --00 0000 | --00 0000 |
| 16h | — | Unimplemented | | | | | | | | — | — |
| 17h | — | Unimplemented | | | | | | | | — | — |
| 18h | — | Unimplemented | | | | | | | | — | — |
| 19h | — | Unimplemented | | | | | | | | — | — |
| 1Ah | — | Unimplemented | | | | | | | | — | — |
| 1Bh | — | Unimplemented | | | | | | | | — | — |
| 1Ch | ADRESL | Least Significant two bits of the left shifted result or eight bits of the right shifted result | | | | | | | | xxxx xxxx | uuuu uuuu |
| 1Dh | ADRESH | Most Significant eight bits of the left shifted A/D result or two bits of the right shifted result | | | | | | | | xxxx xxxx | uuuu uuuu |
| 1Eh | ADCON0 | ADFM | — | CHS<3:0> | | | | GO/$\overline{DONE}$ | ADON | 0-00 0000 | 0-00 0000 |
| 1Fh | ADCON1 | — | ADCS<2:0> | | | — | — | — | ADPREF1 | -000 ---0 | -000 ---0 |

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

**TABLE 2-2: PIC16F753/HV753 SPECIAL REGISTERS SUMMARY BANK 1**

| Addr | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR/BOR | Values on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Bank 1** | | | | | | | | | | | |
| 80h | INDF | INDF<7:0> | | | | | | | | xxxx xxxx | uuuu uuuu |
| 81h | OPTION_REG | $\overline{RAPU}$ | INTEDG | T0CS | T0SE | PSA | PS<2:0> | | | 1111 1111 | 1111 1111 |
| 82h | PCL | PCL<7:0> | | | | | | | | 0000 0000 | 0000 0000 |
| 83h | STATUS | IRP | RP1 | RP0 | $\overline{TO}$ | $\overline{PD}$ | Z | DC | C | 0001 1xxx | 000q quuu |
| 84h | FSR | FSR | | | | | | | | xxxx xxxx | uuuu uuuu |
| 85h | TRISA | — | — | TRISA5 | TRISA4 | TRISA3 | TRISA2 | TRISA1 | TRISA0 | --11 1111 | --11 1111 |
| 86h | — | Unimplemented | | | | | | | | — | — |
| 87h | TRISC | — | — | TRISC5 | TRISC4 | TRISC3 | TRISC2 | TRISC1 | TRISC0 | --11 1111 | --11 1111 |
| 88h | IOCAP | — | — | IOCAP5 | IOCAP4 | IOCAP3 | IOCAP2 | IOCAP1 | IOCAP0 | --00 0000 | --00 0000 |
| 89h | IOCCP | — | — | IOCCP5 | IOCCP4 | IOCCP3 | IOCCP2 | IOCCP1 | IOCCP0 | --00 0000 | --00 0000 |
| 8Ah | PCLATH | — | — | — | PCLATH<4:0> | | | | | ---0 0000 | ---0 0000 |
| 8Bh | INTCON | GIE | PEIE | T0IE | INTE | IOCIE | T0IF | INTF | IOCIF | 0000 0000 | 0000 0000 |
| 8Ch | PIE1 | TMR1GIE | ADIE | — | — | HLTMR2IE | HLTMR1IE | TMR2IE | TMR1IE | 00-- 0000 | 00-- 0000 |
| 8Dh | PIE2 | — | — | C2IE | C1IE | — | COG1IE | — | CCP1IE | --00 -0-0 | --00 -0-0 |
| 8Eh | — | Unimplemented | | | | | | | | — | — |
| 8Fh | OSCCON | — | — | IRCF<1:0> | | — | HTS | LTS | — | --01 -00- | --uu -uu- |
| 90h | FVR1CON0 | FVREN | FVRRDY | FVROE | FVRBUFSS1 | FVRBUFSS0 | — | — | FVRBUFEN | 0000 0--0 | 0000 0--0 |
| 91h | DAC1CON0 | DACEN | DACFM | DACOE | — | DACPSS1 | DACPSS0 | — | — | 000- 00-- | 000- 00-- |
| 92h | DAC1REFL | Least Significant bit of the left shifted result or eight bits of the right shifted DAC setting | | | | | | | | 0000 0000 | 0000 0000 |
| 93h | DAC1REFH | Most Significant eight bits of the left shifted DAC setting or first bit of the right shifted result | | | | | | | | 0000 0000 | 0000 0000 |
| 94h | — | Unimplemented | | | | | | | | — | — |
| 95h | — | Unimplemented | | | | | | | | — | — |
| 96h | OPA1CON | OPA1EN | — | — | OPA1UGM | OPA1NCH<1:0> | | OPA1PCH<1:0> | | 0--0 0000 | 0--0 0000 |
| 97h | — | Unimplemented | | | | | | | | — | — |
| 98h | — | Unimplemented | | | | | | | | — | — |
| 99h | — | Unimplemented | | | | | | | | — | — |
| 9Ah | — | Unimplemented | | | | | | | | — | — |
| 9Bh | CM2CON0 | C2ON | C2OUT | C2OE | C2POL | C2ZLF | C2SP | C2HYS | C2SYNC | 0000 0100 | 0000 0100 |
| 9Ch | CM2CON1 | C2INTP | C2INTN | C2PCH<2:0> | | | C2NCH<2:0> | | | 0000 0000 | 0000 0000 |
| 9Dh | CM1CON0 | C1ON | C1OUT | C1OE | C1POL | C1ZLF | C1SP | C1HYS | C1SYNC | 0000 0100 | 0000 0100 |
| 9Eh | CM1CON1 | C1INTP | C1INTN | C1PCH<2:0> | | | C1NCH<2:0> | | | 0000 0000 | 0000 0000 |
| 9Fh | CMOUT | — | — | — | — | — | — | MCOUT2 | MCOUT1 | ---- --00 | ---- --00 |

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

**TABLE 2-3: PIC16F753/HV753 SPECIAL REGISTERS SUMMARY BANK 2**

| Addr | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR/BOR | Value on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Bank 2** | | | | | | | | | | | |
| 100h | INDF | INDF<7:0> | | | | | | | | xxxx xxxx | xxxx xxxx |
| 101h | TMR0 | TMR0<7:0> | | | | | | | | xxxx xxxx | uuuu uuuu |
| 102h | PCL | PCL<7:0> | | | | | | | | 0000 0000 | 0000 0000 |
| 103h | STATUS | IRP | RP1 | RP0 | $\overline{TO}$ | $\overline{PD}$ | Z | DC | C | 0001 1xxx | 000q quuu |
| 104h | FSR | FSR<7:0> | | | | | | | | xxxx xxxx | uuuu uuuu |
| 105h | LATA | — | — | LATA5 | LATA4 | — | LATA2 | LATA1 | LATA0 | --xx -xxx | --uu -uuu |
| 106h | — | Unimplemented | | | | | | | | — | — |
| 107h | LATC | — | — | LATC5 | LATC4 | LATC3 | LATC2 | LATC1 | LATC0 | --xx xxxx | --uu uuuu |
| 108h | IOCAN | — | — | IOCAN5 | IOCAN4 | IOCAN3 | IOCAN2 | IOCAN1 | IOCAN0 | --00 0000 | --00 0000 |
| 109h | IOCCN | — | — | IOCCN5 | IOCCN4 | IOCCN3 | IOCCN2 | IOCCN1 | IOCCN0 | --00 0000 | --00 0000 |
| 10Ah | PCLATH | — | — | — | PCLATH<4:0> | | | | | ---0 0000 | ---0 0000 |
| 10Bh | INTCON | GIE | PEIE | T0IE | INTE | IOCIE | T0IF | INTF | IOCIF | 0000 0000 | 0000 0000 |
| 10Ch | WPUA | — | — | WPUA5 | WPUA4 | WPUA3 | WPUA2 | WPUA1 | WPUA0 | --11 1111 | --11 1111 |
| 10Dh | WPUC | — | — | WPUC5 | WPUC4 | WPUC3 | WPUC2 | WPUC1 | WPUC0 | --11 1111 | --11 1111 |
| 10Eh | SLRCONC | — | — | SLRC5 | SLRC4 | — | — | — | — | --00 ---- | --00 ---- |
| 10Fh | PCON | — | — | — | — | — | — | $\overline{POR}$ | $\overline{BOR}$ | ---- --qq | ---- --uu |
| 110h | TMR2 | TMR2<7:0> | | | | | | | | 0000 0000 | 0000 0000 |
| 111h | PR2 | PR2<7:0> | | | | | | | | 1111 1111 | 1111 1111 |
| 112h | T2CON | — | T2OUTPS<3:0> | | | | TMR2ON | T2CKPS<1:0> | | -000 0000 | -000 0000 |
| 113h | HLTMR1 | Holding Register for the 8-bit Hardware Limit Timer1 Count | | | | | | | | 0000 0000 | 0000 0000 |
| 114h | HLTPR1 | HLTMR1 Module Period Register | | | | | | | | 1111 1111 | 1111 1111 |
| 115h | HLT1CON0 | — | H1OUTPS<3:0> | | | | H1ON | H1CKPS<1:0> | | -000 0000 | -000 0000 |
| 116h | HLT1CON1 | H1FES | H1RES | — | H1ERS<2:0> | | | H1FEREN | H1REREN | 11-0 0000 | 11-0 0000 |
| 117h | HLTMR2 | Holding Register for the 8-bit Hardware Limit Timer2 Count | | | | | | | | 0000 0000 | 0000 0000 |
| 118h | HLTPR2 | HLTMR2 Module Period Register | | | | | | | | 1111 1111 | 1111 1111 |
| 119h | HLT2CON0 | — | H2OUTPS<3:0> | | | | H2ON | H2CKPS<1:0> | | -000 0000 | -000 0000 |
| 11Ah | HLT2CON1 | H2FES | H2RES | — | H2ERS<2:0> | | | H2FEREN | H2REREN | 11-0 0000 | 11-0 0000 |
| 11Bh | — | Unimplemented | | | | | | | | — | — |
| 11Ch | — | Unimplemented | | | | | | | | — | — |
| 11Dh | — | Unimplemented | | | | | | | | — | — |
| 11Eh | SLPCCON0 | SC1EN | — | — | SC1POL | SC1TSS<1:0> | | — | SC1INS | 0-00 00-0 | 0-00 00-0 |
| 11Fh | SLPCCON1 | — | — | — | SC1RNG | SC1ISET<3:0> | | | | ---0 0000 | ---0 0000 |

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

**TABLE 2-4: PIC16F753/HV753 SPECIAL FUNCTION REGISTERS SUMMARY BANK 3**

| Addr | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR/BOR | Values on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Bank 3** | | | | | | | | | | | |
| 180h | INDF | INDF<7:0> | | | | | | | | xxxx xxxx | uuuu uuuu |
| 181h | OPTION_REG | $\overline{RAPU}$ | INTEDG | T0CS | T0SE | PSA | PS<2:0> | | | 1111 1111 | 1111 1111 |
| 182h | PCL | PCL<7:0> | | | | | | | | 0000 0000 | 0000 0000 |
| 183h | STATUS | IRP | RP1 | RP0 | $\overline{TO}$ | $\overline{PD}$ | Z | DC | C | 0001 1xxx | 000q quuu |
| 184h | FSR | FSR<7:0> | | | | | | | | xxxx xxxx | uuuu uuuu |
| 185h | ANSELA | — | — | — | ANSA4 | — | ANSA2 | ANSA1 | ANSA0 | ---1 -111 | ---1 -111 |
| 186h | — | Unimplemented | | | | | | | | — | — |
| 187h | ANSELC | — | — | — | — | ANSC3 | ANSC2 | ANSC1 | ANSC0 | ---- 0000 | ---- 0000 |
| 188h | APFCON | — | — | — | T1GSEL | — | — | — | — | ---0 ---- | ---0 ---- |
| 189h | OSCTUNE | — | — | — | TUN<4:0> | | | | | ---0 0000 | ---0 0000 |
| 18Ah | PCLATH | — | — | — | PCLATH<4:0> | | | | | ---0 0000 | ---0 0000 |
| 18Bh | INTCON | GIE | PEIE | T0IE | INTE | IOCIE | T0IF | INTF | IOCIF | 0000 0000 | 0000 0000 |
| 18Ch | PMCON1 | — | — | — | — | — | WREN | WR | RD | ---- -000 | ---- -000 |
| 18Dh | PMCON2 | Program Memory Control Register 2 | | | | | | | | ---- ---- | ---- ---- |
| 18Eh | PMADRL | PMADRL<7:0> | | | | | | | | 0000 0000 | 0000 0000 |
| 18Fh | PMADRH | — | — | — | — | — | — | PMADRH<1:0> | | ---- --00 | ---- --00 |
| 190h | PMDATL | PMDATL<7:0> | | | | | | | | 0000 0000 | 0000 0000 |
| 191h | PMDATH | — | — | PMDATH<5:0> | | | | | | --00 0000 | --00 0000 |
| 192h | COG1PHR | — | — | — | — | G1PHR<3:0> | | | | ---- xxxx | ---- uuuu |
| 193h | COG1PHF | — | — | — | — | G1PHF<3:0> | | | | ---- xxxx | ---- uuuu |
| 194h | COG1BKR | — | — | — | — | G1BKR<3:0> | | | | ---- xxxx | ---- uuuu |
| 195h | COG1BKF | — | — | — | — | G1BKF<3:0> | | | | ---- xxxx | ---- uuuu |
| 196h | COG1DBR | — | — | — | — | G1DBR<3:0> | | | | ---- xxxx | ---- uuuu |
| 197h | COG1DBF | — | — | — | — | G1DBF<3:0> | | | | ---- xxxx | ---- uuuu |
| 198h | COG1CON0 | G1EN | G1OE1 | G1OE0 | G1POL1 | G1POL0 | G1LD | — | G1MD | 0000 00-0 | 0000 00-0 |
| 199h | COG1CON1 | G1RDBTS | G1FDBTS | — | — | — | — | G1CS<1:0> | | 00-- --00 | 00-- --00 |
| 19Ah | COG1RIS | — | G1RIHLT2 | G1RIHLT1 | G1RIT2M | G1RIFLT | G1RICCP1 | G1RIC2 | G1RIC1 | 0000 0000 | 0000 0000 |
| 19Bh | COG1RSIM | — | G1RMHLT2 | G1RMHLT1 | G1RMT2M | G1RMFLT | G1RMCCP1 | G1RMC2 | G1RMC1 | 0000 0000 | 0000 0000 |
| 19Ch | COG1FIS | — | G1FIHLT2 | G1FIHLT1 | G1FIT2M | G1FIFLT | G1FICCP1 | G1FIC2 | G1FIC1 | 0000 0000 | 0000 0000 |
| 19Dh | COG1FSIM | — | G1FMHLT2 | G1FMHLT1 | G1FMT2M | G1FMFLT | G1FMCCP1 | G1FMC2 | G1FMC1 | 0000 0000 | 0000 0000 |
| 19Eh | COG1ASD0 | C1ASDE | C1ARSEN | G1ASD1L<1:0> | | G1ASD0L<1:0> | | — | — | 0000 00-- | 0000 00-- |
| 19Fh | COG1ASD1 | — | — | — | G1ASDSHLT2 | G1ASDSHLT1 | G1ASDSC2 | G1ASDSC1 | G1ASDSFLT | 0000 0000 | 0000 0000 |

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

## 2.3 Global SFRs

### 2.3.1 STATUS REGISTER

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

- the arithmetic status of the ALU
- the Reset status
- the bank select bits for data memory (RAM)

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

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

It is recommended, therefore, that only `BCF, BSF, SWAPF` and `MOVWF` instructions are used to alter the STATUS register, because these instructions do not affect any Status bits. For other instructions not affecting any Status bits, see **Section 18.0 "Instruction Set Summary"**.

**REGISTER 2-1: STATUS: STATUS REGISTER**

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

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

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

bit 6 **RP1:** Register Bank Select bit (used for direct addressing)
00 = Bank 0 (00h-7Fh)
01 = Bank 1 (80h-FFh)
10 = Bank 2 (100h-17Fh)
11 = Bank 3 (180h-1FFh)

bit 5 **RP0:** Register Bank Select bit (used for direct addressing)
1 = Bank 1 (80h-FFh)
0 = Bank 0 (00h-7Fh)

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/$\overline{\text{Borrow}}$ bit(2) (ADDWF, ADDLW, SUBLW, SUBWF instructions), For $\overline{\text{Borrow}}$, the polarity is reversed.
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(2) (ADDWF, ADDLW, SUBLW, SUBWF instructions)
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:** The C and DC bits operate as a $\overline{\text{Borrow}}$ and $\overline{\text{Digit Borrow}}$ out bit, respectively, in subtraction. See the SUBLW and SUBWF instructions for examples.
**2:** For $\overline{\text{Borrow}}$, the polarity is reversed. A subtraction is executed by adding the two's complement of the second operand. For rotate (RRF, RLF) instructions, this bit is loaded with either the high-order or low-order bit of the source register.

### 2.3.2 OPTION REGISTER

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

- Timer0/WDT prescaler
- External RA2/INT interrupt
- Timer0
- Weak pull-ups on PORTA

**Note:** To achieve a 1:1 prescaler assignment for Timer0, assign the prescaler to the WDT by setting PSA bit to '1' of the OPTION register. See **Section 6.1.3 "Software Programmable Prescaler"**.

**REGISTER 2-2: OPTION_REG: OPTION REGISTER**

| R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 |
|---|---|---|---|---|---|---|---|
| $\overline{\text{RAPU}}$ | INTEDG | T0CS | T0SE | PSA | PS<2:0> | | |
| bit 7 | | | | | | | bit 0 |

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

bit 7 **$\overline{\text{RAPU}}$:** PORTA Pull-up Enable bit
1 = PORTA pull-ups are disabled
0 = PORTA pull-ups are enabled by individual PORT 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 **T0CS:** Timer0 Clock Source Select bit
1 = Transition on T0CKI pin
0 = Internal instruction cycle clock (FOSC/4)

bit 4 **T0SE:** 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 assigned to the WDT
0 = Prescaler is assigned to the Timer0 module

bit 2-0 **PS<2:0>:** Prescaler Rate Select bits

| BIT VALUE | TIMER0 RATE | WDT RATE |
|---|---|---|
| 000 | 1 : 2 | 1 : 1 |
| 001 | 1 : 4 | 1 : 2 |
| 010 | 1 : 8 | 1 : 4 |
| 011 | 1 : 16 | 1 : 8 |
| 100 | 1 : 32 | 1 : 16 |
| 101 | 1 : 64 | 1 : 32 |
| 110 | 1 : 128 | 1 : 64 |
| 111 | 1 : 256 | 1 : 128 |

### 2.3.3 INTCON REGISTER

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

**Note:** Interrupt flag bits are set when an interrupt condition occurs, regardless of the state of its corresponding enable bit or the Global Enable bit, GIE of the INTCON register. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

**REGISTER 2-3: INTCON: INTERRUPT CONTROL REGISTER**

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| GIE | PEIE | T0IE | INTE | IOCIE | T0IF | INTF | IOCIF |
| bit 7 | | | | | | | bit 0 |

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

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

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

bit 5 **T0IE:** Timer0 Overflow Interrupt Enable bit
1 = Enables the Timer0 interrupt
0 = Disables the Timer0 interrupt

bit 4 **INTE:** RA2/INT External Interrupt Enable bit
1 = Enables the RA2/INT external interrupt
0 = Disables the RA2/INT external interrupt

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

bit 2 **T0IF:** Timer0 Overflow Interrupt Flag bit(2)
1 = Timer0 register has overflowed (must be cleared in software)
0 = Timer0 register did not overflow

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

bit 0 **IOCIF:** Interrupt-on-Change Interrupt Flag bit
1 = An IOC pin has changed state and generated an interrupt
0 = No pin interrupts have been generated

**Note 1:** IOC register must also be enabled.
**2:** T0IF bit is set when TMR0 rolls over. TMR0 is unchanged on Reset and should be initialized before clearing T0IF bit.

### 2.3.4 PIE1 REGISTER

The PIE1 register contains the Peripheral Interrupt Enable bits, as shown in Register 2-4.

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

**REGISTER 2-4: PIE1: PERIPHERAL INTERRUPT ENABLE REGISTER 1**

| R/W-0 | R/W-0 | U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| TMR1GIE | ADIE | — | — | HLTMR2IE | HLTMR1IE | TMR2IE | TMR1IE |
| bit 7 | | | | | | | bit 0 |

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

bit 7 **TMR1GIE:** ADC Interrupt Enable bit
1 = Enables the TMR1 gate interrupt
0 = Disables the TMR1 gate interrupt

bit 6 **ADIE:** ADC Interrupt Enable bit
1 = Enables the ADC interrupt
0 = Disables the ADC interrupt

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

bit 3 **HLTMR2IE:** HLT2 Interrupt Enable bit
1 = Enables the HLT2 interrupt
0 = Disables the HLT2 interrupt

bit 2 **HLTMR1IE:** HLT1 Interrupt Enable bit
1 = Enables the HLT1 interrupt
0 = Disables the HLT1 interrupt

bit 1 **TMR2IE:** Timer2 Interrupt Enable bit
1 = Enables the Timer2 interrupt
0 = Disables the Timer2 interrupt

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

### 2.3.5 PIE2 REGISTER

The PIE2 register contains the Peripheral Interrupt Enable bits, as shown in Register 2-5.

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

**REGISTER 2-5: PIE2: PERIPHERAL INTERRUPT ENABLE REGISTER 1**

| U-0 | U-0 | R/W-0 | R/W-0 | U-0 | R/W-0 | U-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | — | C2IE | C1IE | — | COG1IE | — | CCP1IE |
| bit 7 | | | | | | | bit 0 |

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

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

bit 5 **C2IE:** Comparator 2 Interrupt Enable bit
1 = Enables the Comparator 2 interrupt
0 = Disables the Comparator 2 interrupt

bit 4 **C1IE:** Comparator 1 Interrupt Enable bit
1 = Enables the Comparator 1 interrupt
0 = Disables the Comparator 1 interrupt

bit 3 **Unimplemented:** Read as ‘0’

bit 2 **COG1IE:** COG 1 Interrupt Flag bit
1 = COG1 interrupt enabled
0 = COG1 interrupt disabled

bit 1 **Unimplemented:** Read as ‘0’

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

### 2.3.6 PIR1 REGISTER

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

**Note:** Interrupt flag bits are set when an interrupt condition occurs, regardless of the state of its corresponding enable bit or the Global Enable bit, GIE of the INTCON register. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

**REGISTER 2-6: PIR1: PERIPHERAL INTERRUPT REQUEST REGISTER 1**

| R/W-0 | R/W-0 | U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| TMR1GIF | ADIF | — | — | HLTMR2IF | HLTMR1IF | TMR2IF | TMR1IF |
| bit 7 | | | | | | | bit 0 |

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

bit 7 **TMR1GIF:** TMR1 Gate Interrupt Flag bit
1 = Timer1 gate interrupt is pending
0 = Timer1 gate interrupt is not pending

bit 6 **ADIF:** ADC Interrupt Flag bit
1 = ADC conversion complete
0 = ADC conversion has not completed or has not been started

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

bit 3 **HLTMR2IF:** HLT2 to HLTPR2 Match Interrupt Flag bit
1 = HLT2 to HLTPR2 match occurred (must be cleared in software)
0 = HLT2 to HLTPR2 match did not occur

bit 2 **HLTMR1IF:** HLT1 to HLTPR1 Match Interrupt Flag bit
1 = HLT1 to HLTPR1 match occurred (must be cleared in software)
0 = HLT1 to HLTPR1 match did not occur

bit 1 **TMR2IF:** Timer2 to PR2 Match Interrupt Flag bit
1 = Timer2 to PR2 match occurred (must be cleared in software)
0 = Timer2 to PR2 match did not occur

bit 0 **TMR1IF:** Timer1 Interrupt Flag bit
1 = Timer1 rolled over (must be cleared in software)
0 = Timer1 has not rolled over

### 2.3.7 PIR2 REGISTER

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

| | |
|---|---|
| **Note:** | Interrupt flag bits are set when an interrupt condition occurs, regardless of the state of its corresponding enable bit or the Global Enable bit, GIE of the INTCON register. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt. |

**REGISTER 2-7: PIR2: PERIPHERAL INTERRUPT REQUEST REGISTER 1**

| U-0 | U-0 | R/W-0 | R/W-0 | U-0 | R/W-0 | U-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | — | C2IF | C1IF | — | COG1IF | — | CCP1IF |
| bit 7 | | | | | | | bit 0 |

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

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

bit 5 **C2IF:** Comparator 1 Interrupt Flag bit
1 = Comparator output (C2OUT bit) has changed (must be cleared in software)
0 = Comparator output (C2OUT bit) has not changed

bit 4 **C1IF:** Comparator 1 Interrupt Flag bit
1 = Comparator output (C1OUT bit) has changed (must be cleared in software)
0 = Comparator output (C1OUT bit) has not changed

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

bit 2 **COG1IF:** COG 1 Interrupt Flag bit
1 = COG1 has generated an auto-shutdown interrupt
0 = COG1 has NOT generated an auto-shutdown interrupt

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

bit 0 **CCP1IF:** ECCP Interrupt Flag bit

Capture Mode
1 = A TMR1 register capture occurred (must be cleared in software)
0 = No TMR1 register capture occurred

Compare Mode
1 = A TMR1 register compare match occurred (must be cleared in software)
0 = No TMR1 register compare match occurred

PWM mode
Unused in this mode

### 2.3.8 PCON REGISTER

The Power Control (PCON) register (see Table 19-2) contains flag bits to differentiate between a:

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

The PCON register also controls the software enable of the $\overline{BOR}$.

The PCON register bits are shown in Register 2-8.

**REGISTER 2-8: PCON: POWER CONTROL REGISTER**

| U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | R/W-q/u | R/W-q/u |
|---|---|---|---|---|---|---|---|
| — | — | — | — | — | — | $\overline{POR}$ | $\overline{BOR}$ |
| 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 = unchanged |

bit 7-2 **Unimplemented:** Read as ‘0’

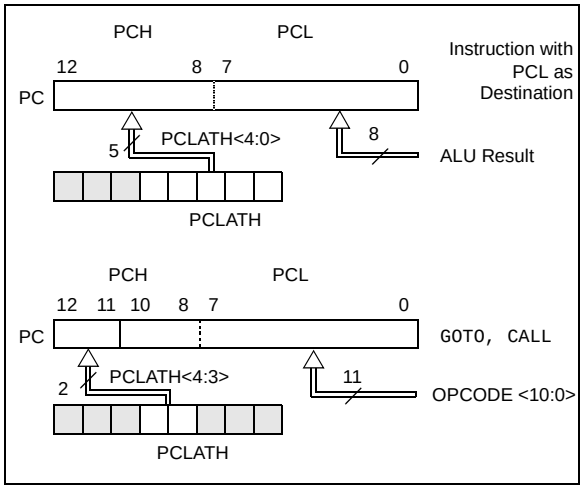
bit 1 **$\overline{POR}$:** Power-on Reset Status bit
1 = No Power-on Reset occurred
0 = A Power-on Reset occurred (must be set in software after a Power-on Reset occurs)

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

## 2.4 PCL and PCLATH

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

**FIGURE 2-3: LOADING OF PC IN DIFFERENT SITUATIONS**




### 2.4.1 MODIFYING PCL

Executing any instruction with the PCL register as the destination simultaneously causes the Program Counter PC<12:8> 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 five bits to the PCLATH register. When the lower eight bits are written to the PCL register, all 13 bits of the program counter will change to the values contained in the PCLATH register and those being written to the PCL register.

A computed GOTO is accomplished by adding an offset to the program counter (ADDWF PCL). Care should be exercised when jumping into a look-up table or program branch table (computed GOTO) by modifying the PCL register. Assuming that PCLATH is set to the table start address, if the table length is greater than 255 instructions or if the lower eight bits of the memory address rolls over from 0xFF to 0x00 in the middle of the table, then PCLATH must be incremented for each address rollover that occurs between the table beginning and the target location within the table.

For more information refer to Application Note *AN556, Implementing a Table Read* (DS00556).

### 2.4.2 STACK

The PIC16F753/HV753 Family has an 8-level x 13-bit wide hardware stack (see Figure 2-1). The stack space is not part of either program or data space and the Stack Pointer is not readable or writable. The PC is PUSHed onto the stack when a CALL instruction is executed or an interrupt causes a branch. The stack is POPed in the event of a RETURN, RETLW or a RETFIE instruction execution. PCLATH is not affected by a PUSH or POP operation.

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

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

## 2.5 Indirect Addressing, INDF and FSR Registers

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

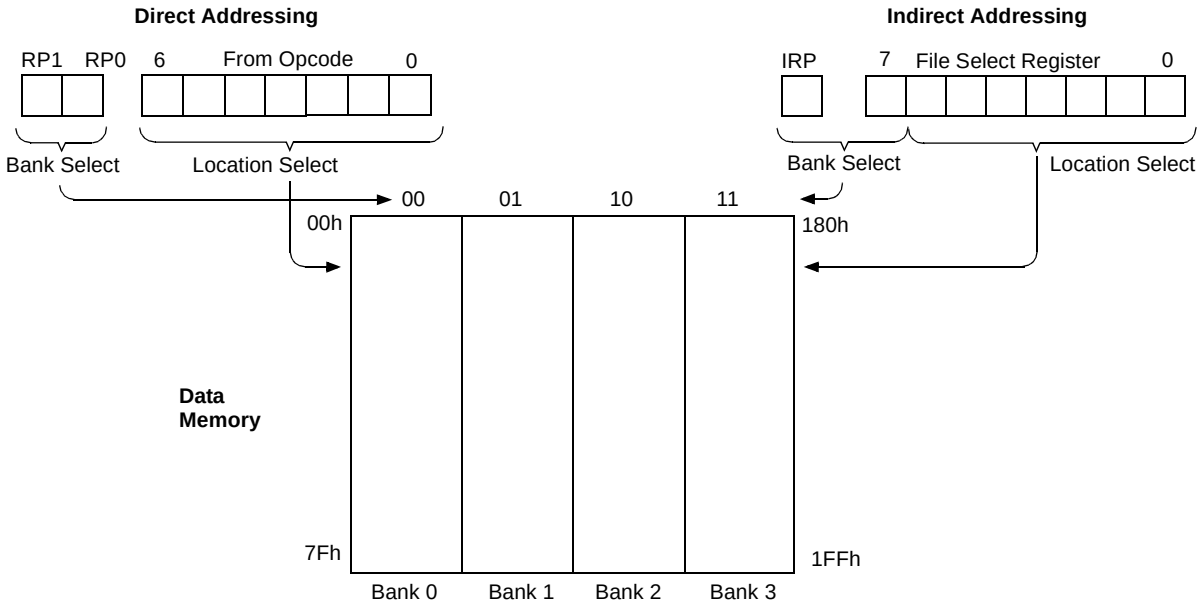
Indirect addressing is possible by using the INDF register. Any instruction using the INDF register actually accesses data pointed to by the File Select Register (FSR). Reading INDF itself indirectly will produce 00h. Writing to the INDF register indirectly results in a no operation (although Status bits may be affected). An effective 9-bit address is obtained by concatenating the 8-bit FSR and the IRP bit of the STATUS register, as shown in Figure 2-4.

A simple program to clear RAM location 40h-7Fh using indirect addressing is shown in Example 2-1.

**EXAMPLE 2-1: INDIRECT ADDRESSING**

```
        MOVLW   0x40    ;initialize pointer
        MOVWF   FSR     ;to RAM
NEXT    CLRF    INDF    ;clear INDF register
        INCF    FSR     ;inc pointer
        BTFSS   FSR,7   ;all done?
        GOTO    NEXT    ;no clear next
CONTINUE                ;yes continue
```

**FIGURE 2-4: DIRECT/INDIRECT ADDRESSING PIC16F753/HV753**




For memory map detail, see Figure 2-2.

## 3.0 FLASH PROGRAM MEMORY SELF-READ/SELF-WRITE CONTROL

The Flash program memory is readable and writable during normal operation (full $V_{DD}$ range). This memory is not directly mapped in the register file space. Instead, it is indirectly addressed through the Special Function Registers (see Registers 3-1 to 3-5). There are six SFRs used to read and write this memory:

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

When interfacing the program memory block, the PMDATL and PMDATH registers form a two-byte word which holds the 14-bit data for read/write, and the PMADRL and PMADRH registers form a two-byte word which holds the 10-bit address of the Flash location being accessed. These devices have 1K words of program Flash with an address range from 0000h to 03FFh.

The program memory allows a single-word read and a four-word write. A four-word write automatically erases the row of the location and writes the new data (erase before write).

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

When the device is code-protected, the CPU may continue to read and write the Flash program memory.

Depending on the settings of the Flash Program Memory Enable (WRT<1:0>) bits, the device may or may not be able to write certain blocks of the program memory; however, reads of the program memory are allowed.

When the Flash program memory Code Protection ($\overline{CP}$) bit in the Configuration Word register is enabled, the program memory is code-protected, and the device programmer (ICSP™) cannot access data or program memory.

### 3.1 PMADRH and PMADRL Registers

The PMADRH and PMADRL registers can address up to a maximum of 1K words of program memory.

When selecting a program address value, the Most Significant Byte (MSB) of the address is written to the PMADRH register and the Least Significant Byte (LSB) is written to the PMADRL register.

### 3.2 PMCON1 and PMCON2 Registers

PMCON1 is the control register for the data 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 in hardware at completion of the read or write operation. The inability to clear the WR bit in software prevents the accidental premature termination of a write operation.

The WREN bit, when set, will allow a write operation. On power-up, the WREN bit is clear.

PMCON2 is not a physical register. Reading PMCON2 will read all ‘0’s. The PMCON2 register is used exclusively in the Flash memory write sequence.

## 3.3 Register Definitions: Flash Program Memory Control

**REGISTER 3-1: PMDATL: PROGRAM MEMORY DATA LOW BYTE REGISTER**

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

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

bit 7-0 **PMDATL<7:0>**: Eight Least Significant Data bits to Write or Read from Program Memory

**REGISTER 3-2: PMADRL: PROGRAM MEMORY ADDRESS LOW BYTE REGISTER**

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

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

bit 7-0 **PMADRL<7:0>**: Eight Least Significant Address bits for Program Memory Read/Write Operation

**REGISTER 3-3: PMDATH: PROGRAM MEMORY DATA HIGH BYTE REGISTER**

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

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

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

bit 5-0 **PMDATH<5:0>**: Six Most Significant Data bits from Program Memory

**REGISTER 3-4: PMADRH: PROGRAM MEMORY ADDRESS HIGH BYTE REGISTER**

| U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | — | — | — | — | — | PMADRH<1:0> | |
| bit 7 | | | | | | | bit 0 |

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

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

bit 1-0 **PMADRH<1:0>**: Specifies the two Most Significant Address bits or High bits for Program Memory Reads.

**REGISTER 3-5: PMCON1: PROGRAM MEMORY CONTROL 1 REGISTER**

| U-0 | U-0 | U-0 | U-0 | U-0 | R/W-0/0 | R/S/HC-0/0 | R/S/HC-0/0 |
|---|---|---|---|---|---|---|---|
| — | — | — | — | — | 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 7-3 **Unimplemented:** Read as ‘0’

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

**REGISTER 3-6: 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 |
|---|---|---|---|---|---|---|---|
| Program Memory Control Register 2 | | | | | | | |
| 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.

## 3.4 Reading the Flash Program Memory

To read a program memory location, the user must write two bytes of the address to the PMADRL and PMADRH registers, and then set control bit RD (PMCON1<0>). Once the read control bit is set, the program memory Flash controller will use the second instruction cycle after 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 PMDATL and PMDATH registers; it can be read as two bytes in the following instructions. PMDATL and PMDATH registers will hold this value until another read or until it is written to by the user (during a write operation).

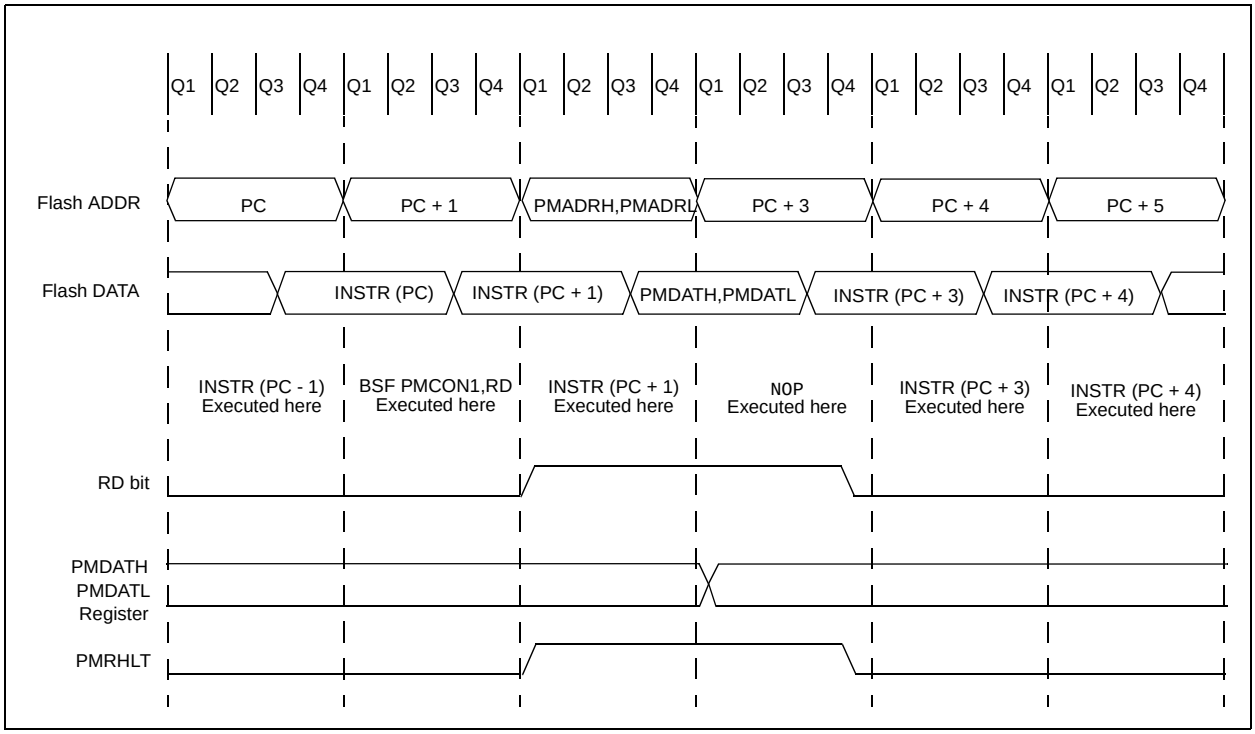
**EXAMPLE 3-1: FLASH PROGRAM READ**

```
    BANKSEL  PM_ADR          ; Change STATUS bits RP1:0 to select bank with PMADRL
    MOVLW    MS_PROG_PM_ADDR ;
    MOVWF    PMADRH          ; MS Byte of Program Address to read
    MOVLW    LS_PROG_PM_ADDR ;
    MOVWF    PMADRL          ; LS Byte of Program Address to read
    BANKSEL  PMCON1          ; Bank to containing PMCON1
    BSF      PMCON1, RD      ; PM Read

    NOP                      ; First instruction after BSF PMCON1,RD executes normally

    NOP                      ; Any instructions here are ignored as program
                             ; memory is read in second cycle after BSF PMCON1,RD
                             ;
    BANKSEL  PMDATL          ; Bank to containing PMADRL
    MOVF     PMDATL, W       ; W = LS Byte of Program PMDATL
    MOVF     PMDATH, W       ; W = MS Byte of Program PMDATL
```

**FIGURE 3-1: FLASH PROGRAM MEMORY READ CYCLE EXECUTION**



## 3.5 Writing the Flash Program Memory

A word of the Flash program memory may only be written to if the word is in an unprotected segment of memory.

Flash program memory must be written in four-word blocks. See Figure 3-2 and Figure 3-3 for more details. A block consists of four words with sequential addresses, with a lower boundary defined by an address, where PMADRL<1:0> = 00. All block writes to program memory are done as 16-word erase by four-word write operations. The write operation is edge-aligned and cannot occur across boundaries.

To write program data, it must first be loaded into the buffer registers (see Figure 3-2). This is accomplished by first writing the destination address to PMADRL and PMADRH and then writing the data to PMDATL and PMDATH. After the address and data have been set up, then the following sequence of events must be executed:

1. Write 55h, then AAh, to PMCON2 (Flash programming sequence).
2. Set the WR control bit of the PMCON1 register.

All four buffer register locations should be written to with correct data. If less than four words are being written to in the block of four words, then a read from the program memory location(s) not being written to must be performed. This takes the data from the program location(s) not being written and loads it into the PMDATL and PMDATH registers. Then the sequence of events to transfer data to the buffer registers must be executed.

To transfer data from the buffer registers to the program memory, the PMADRL and PMADRH must point to the last location in the four-word block (PMADRL<1:0> = 11). Then the following sequence of events must be executed:

1. Write 55h, then AAh, to PMCON2 (Flash programming sequence).
2. Set control bit WR of the PMCON1 register to begin the write operation.

The user must follow the same specific sequence to initiate the write for each word in the program block, writing each program word in sequence (000, 001, 010, 011). When the write is performed on the last word (PMADRL<1:0> = 11), a block of sixteen words is automatically erased and the content of the four-word buffer registers are written into the program memory.

After the "BSF PMCON1,WR" instruction, the processor requires two cycles to set up the erase/write operation. The user must place two NOP instructions after the WR bit is set. Since data is being written to buffer registers, the writing of the first three words of the block appears to occur immediately. The processor will halt internal operations for the typical 4 ms, only during the cycle in which the erase takes place (i.e., the last word of the sixteen-word block erase). This is not Sleep mode as the clocks and peripherals will continue to run. After the four-word write cycle, the processor will resume operation with the third instruction after the PMCON1 write instruction. The above sequence must be repeated for the higher 12 words.

## 3.6 Protection Against Spurious Write

There are conditions when the device should not write to the program memory. To protect against spurious writes, various mechanisms have been built in. On power-up, WREN is cleared. Also, the Power-up Timer (64 ms duration) prevents program memory writes.

The write initiate sequence and the WREN bit help prevent an accidental write during brown-out, power glitch or software malfunction.

## 3.7 Operation During Code-Protect

When the device is code-protected, the CPU is able to read and write unscrambled data to the program memory.

## 3.8 Operation During Write Protect

When the program memory is write-protected, the CPU can read and execute from the program memory. The portions of program memory that are write-protected can be modified by the CPU using the PMCON registers, but the protected program memory cannot be modified using ICSP mode.

**FIGURE 3-2: BLOCK WRITES TO 1K FLASH PROGRAM MEMORY**

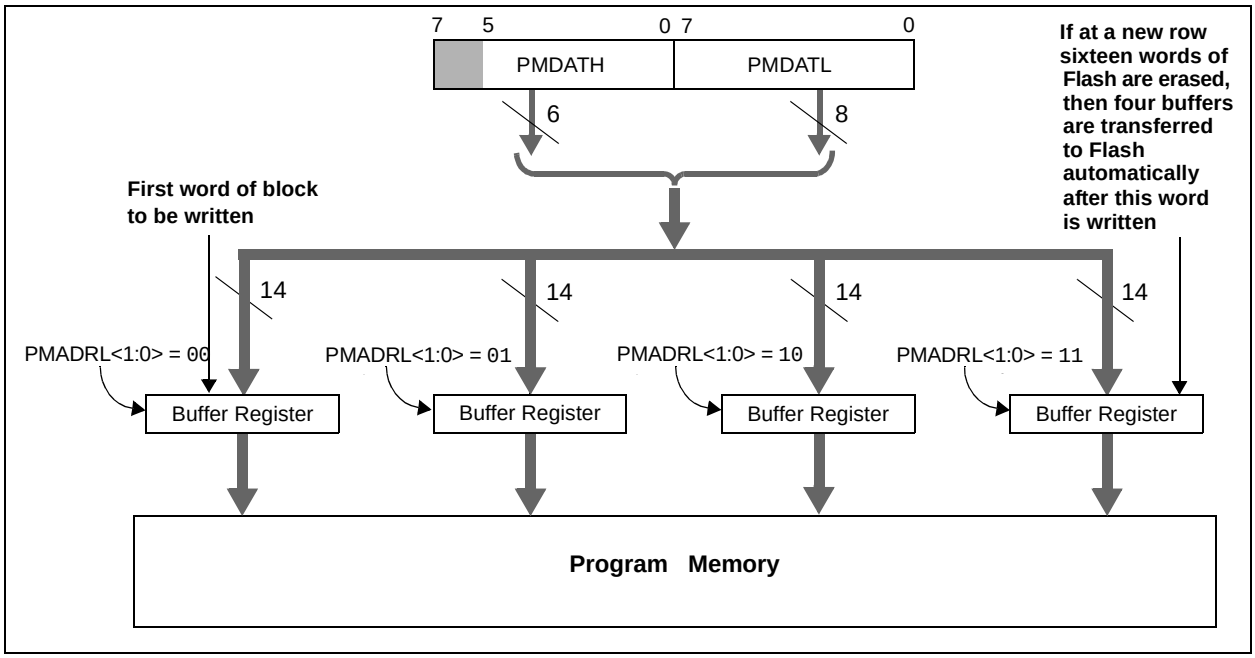
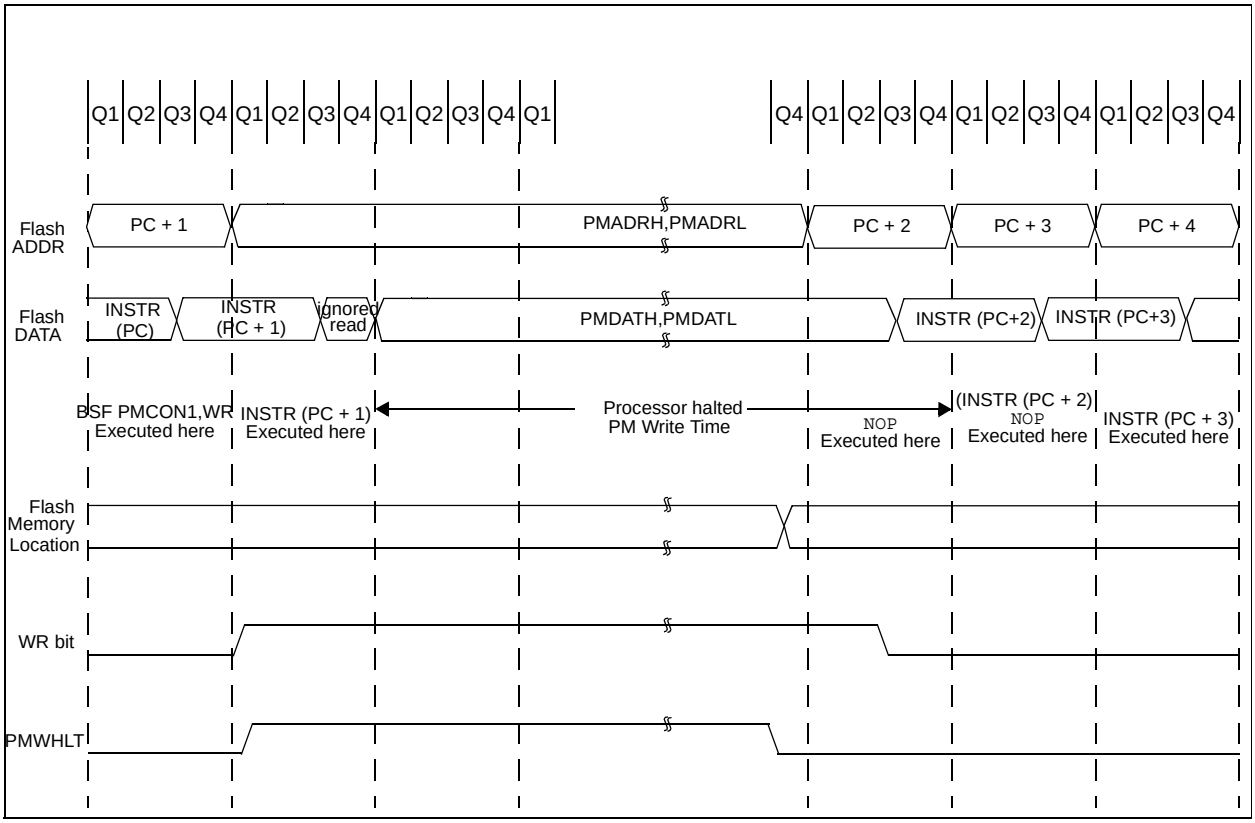



**FIGURE 3-3: FLASH PROGRAM MEMORY LONG WRITE CYCLE EXECUTION**



An example of the complete four-word write sequence is shown in Example 3-2. The initial address is loaded into the PMADRH and PMADRL register pair; the four words of data are loaded using indirect addressing.

**EXAMPLE 3-2: WRITING TO FLASH PROGRAM MEMORY**

```
    ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
    ; This write routine assumes the following:
    ;        A valid starting address (the least significant bits = '00')
    ;        is loaded in ADDRH:ADDRL
    ;        ADDRH, ADDRL and DATADDR are all located in data memory
    ;
   BANKSEL         PMADRH
   MOVF    ADDRH,W      ;Load initial address
   MOVWF   PMADRH       ;
   MOVF    ADDRL,W      ;
   MOVWF   PMADRL       ;
   MOVF    DATAADDR,W   ;Load initial data address
   MOVWF   FSR          ;
LOOP MOVF  INDF,W       ;Load first data byte into lower
   MOVWF   PMDATL       ;
   INCF    FSR,F        ;Next byte
   MOVF    INDF,W       ;Load second data byte into upper
   MOVWF   PMDATH       ;
   INCF    FSR,F        ;
   BANKSEL PMCON1
   BSF     PMCON1,WREN  ;Enable writes
   BCF     INTCON,GIE   ;Disable interrupts (if using)
   BTFSC   INTCON,GIE   ;See AN576
   GOTO    $-2
   ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
   ;       Required Sequence
   MOVLW   55h          ;Start of required write sequence:
   MOVWF   PMCON2       ;Write 55h
   MOVLW   0AAh         ;
   MOVWF   PMCON2       ;Write 0AAh
   BSF     PMCON1,WR    ;Set WR bit to begin write
   NOP                  ;Required to transfer data to the buffer
   NOP                  ;registers
   ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
   BCF     PMCON1,WREN  ;Disable writes
   BSF     INTCON,GIE   ;Enable interrupts (comment out if not using interrupts)
   BANKSEL PMADRL
   MOVF    PMADRL, W
   INCF    PMADRL,F     ;Increment address
   ANDLW   0x03         ;Indicates when sixteen words have been programmed
   SUBLW   0x03         ;Change value for different size write blocks
                        ;0x0F = 16 words
                        ;0x0B = 12 words
                        ;0x07 = 8 words
                        ;0x03 = 4 words
   BTFSS   STATUS,Z     ;Exit on a match,
   GOTO    LOOP         ;Continue if more data needs to be written
```

**TABLE 3-1: 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 |
|---|---|---|---|---|---|---|---|---|---|
| PMCON1 | — | — | — | — | — | WREN | WR | RD | 27 |
| PMCON2 | Program Memory Control Register 2 | | | | | | | | 27 |
| PMADRL | PMADRL<7:0> | | | | | | | | 26 |
| PMADRH | — | — | — | — | — | — | PMADRH<1:0> | | 26 |
| PMDATL | PMDATL<7:0> | | | | | | | | 26 |
| PMDATH | — | — | PMDATH<5:0> | | | | | | 26 |
| INTCON | GIE | PEIE | T0IE | INTE | IOCIE | T0IF | INTF | IOCIF | 17 |

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

**TABLE 3-2: SUMMARY OF CONFIGURATION WORD 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 |
|---|---|---|---|---|---|---|---|---|---|---|
| CONFIG(1) | 13:8 | — | — | $\overline{\text{DEBUG}}$ | $\overline{\text{CLKOUTEN}}$ | WRT<1:0> | | BOREN<1:0> | | 150 |
| | 7:0 | — | $\overline{\text{CP}}$ | MCLRE | $\overline{\text{PWRTE}}$ | WDTE | — | — | FOSC0 | |

**Legend:** — = unimplemented location, read as '1'. Shaded cells are not used by Flash program memory.
**Note 1:** See Configuration Word register (Register 19-1) for operation of all register bits.

## 4.0 OSCILLATOR MODULE

### 4.1 Overview

The oscillator module has a variety of clock sources and selection features that allow it to be used in a wide range of applications while maximizing performance and minimizing power consumption. Figure 4-1 illustrates a block diagram of the oscillator module.

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

1. EC (external clock)
2. INTOSC (internal oscillator)

Clock Source modes are configured by the FOSC bit in the Configuration Word register (CONFIG).

The internal oscillator module provides the following selectable system clock modes:

- 8 MHz (HFINTOSC)
- 4 MHz (HFINTOSC Postscaler)
- 1 MHz (HFINTOSC Postscaler)
- 31 kHz (LFINTOSC)

**FIGURE 4-1: PIC® MCU CLOCK SOURCE BLOCK DIAGRAM**

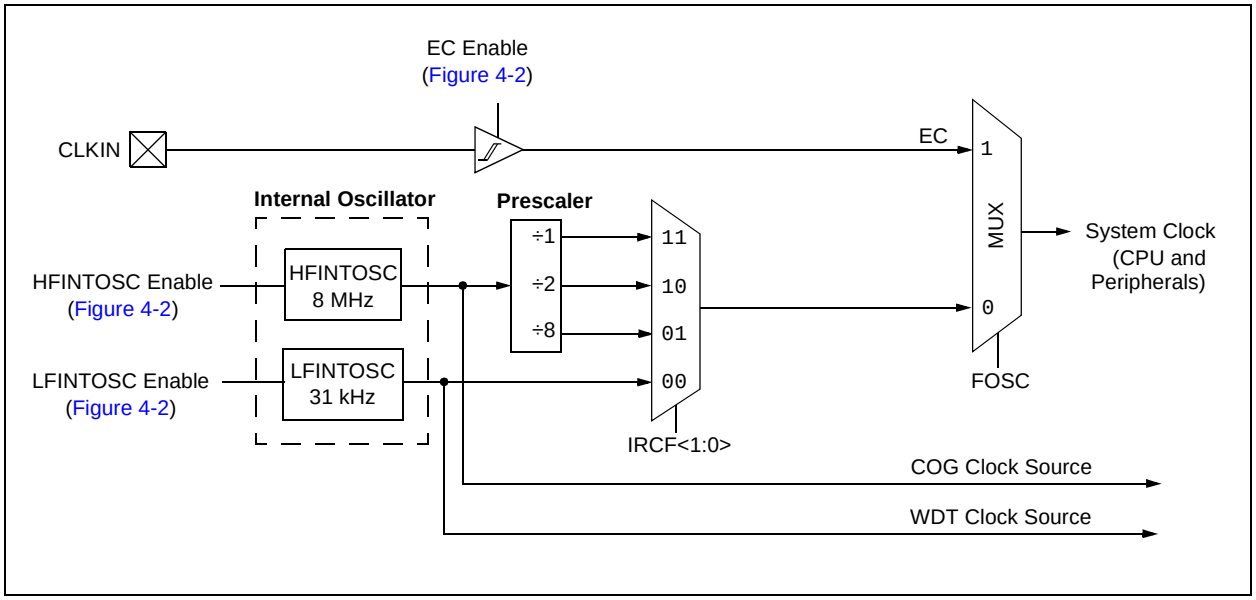
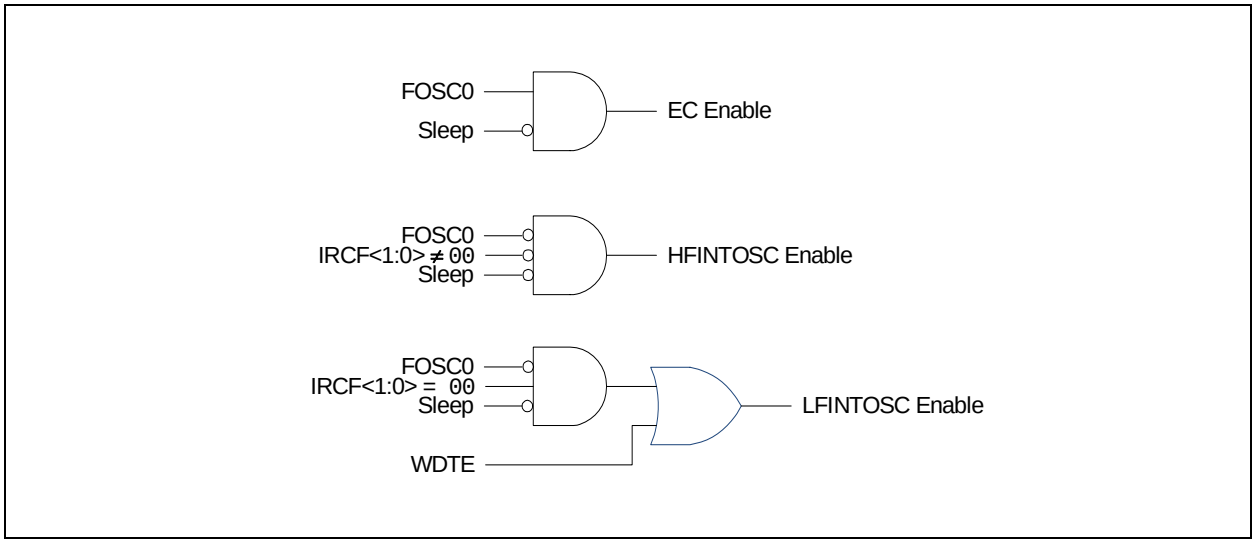



**FIGURE 4-2: OSCILLATOR ENABLE**



## 4.2 Clock Source Modes

Clock Source modes can be classified as external or internal:

- The External Clock mode relies on an external clock for the clock source. For example, a clock module or clock output from another circuit.
- Internal clock sources are contained internally within the oscillator module. The oscillator module has four selectable clock frequencies:
  - 8 MHz
  - 4 MHz
  - 1 MHz
  - 31 kHz

The system clock can be selected between external or internal clock sources via the FOSC0 bit of the Configuration Word register (CONFIG).

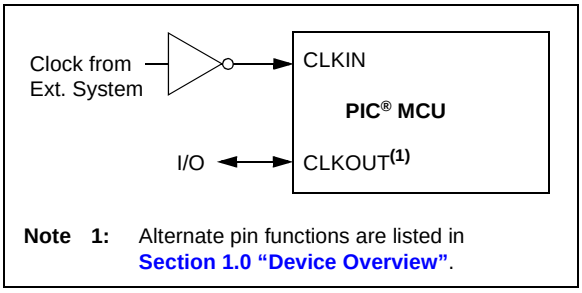
### 4.2.1 EC MODE

The External Clock (EC) mode allows an externally generated logic as the system clock source. The EC clock mode is selected when the FOSC0 bit of the Configuration Word is set.

When operating in this mode, an external clock source must be connected to the CLKIN input. The CLKOUT is available for either general purpose I/O or system clock output. Figure 4-3 shows the pin connections for EC mode.

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 4-3: EXTERNAL CLOCK (EC) MODE OPERATION**




**Note 1:** Alternate pin functions are listed in **Section 1.0 "Device Overview"**.

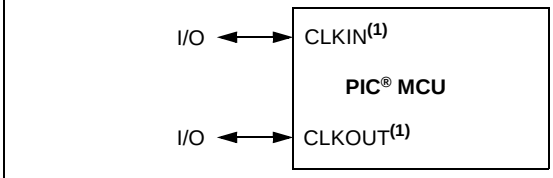
### 4.2.2 INTERNAL CLOCK MODE

Internal Clock mode configures the internal oscillators as the system clock source. The Internal Clock mode is selected when the FOSC0 bit of the Configuration Word is cleared. The source and frequency are selected with the IRCF<1:0> bits of the OSCCON register.

When one of the HFINTOSC frequencies is selected, the frequency of the internal oscillator can be trimmed by adjusting the TUN<4:0> bits of the OSCTUNE register.

Operation after a Power-on Reset (POR) or wake-up from Sleep is delayed by the oscillator start-up time. Delays are typically longer for the LFINTOSC than HFINTOSC because of the very low-power operation and relatively narrow bandwidth of the LF internal oscillator. However, when another peripheral keeps the oscillator running during Sleep, the start-up time is delayed to allow the memory bias to stabilize.

**FIGURE 4-4: INTERNAL CLOCK MODE OPERATION**




**Note 1:** Alternate pin functions are listed in **Section 1.0 "Device Overview"**.

#### 4.2.2.1 Oscillator Ready Bits

The HTS and LTS bits of the OSCCON register indicate the status of the HFINTOSC and LFINTOSC, respectively. When either bit is set, it indicates that the corresponding oscillator is running and stable.

## 4.3 System Clock Output

The CLKOUT pin is available for general purpose I/O or system clock output. The $\overline{\text{CLKOUTEN}}$ bit of the Configuration Word controls the function of the CLKOUT pin.

When the $\overline{\text{CLKOUTEN}}$ bit is cleared, the CLKOUT pin is driven by the selected internal oscillator frequency divided by 4. The corresponding I/O pin always reads '0' in this configuration.

The CLKOUT signal may be used to provide a clock for external circuitry, synchronization, calibration, test or other application requirements.

When the $\overline{\text{CLKOUTEN}}$ bit is set, the system clock out function is disabled and the CLKOUT pin is available for general purpose I/O.

## 4.4 Oscillator Delay upon Wake-Up, Power-Up, and Base Frequency Change

In applications where the OSCTUNE register is used to shift the HFINTOSC frequency, the application should not expect the frequency to stabilize immediately. In this case, the frequency may shift gradually toward the new value. The time for this frequency shift is less than eight cycles of the base frequency.

A short delay is invoked upon power-up and when waking from sleep to allow the memory bias circuitry to stabilize. Table 4-1 shows examples where the oscillator delay is invoked.

**TABLE 4-1: OSCILLATOR DELAY EXAMPLES**

| Switch From | Switch To | Frequency | Oscillator Delay |
|---|---|---|---|
| Sleep/POR | INTOSC | 31 kHz to 8 MHz | 10 μs internal delay to allow memory bias to stabilize. |
| Sleep/POR | EC | DC – 20 MHz | |

## 4.5 Register Definitions: Oscillator Control

**REGISTER 4-1: OSCCON: OSCILLATOR CONTROL REGISTER**

| U-0 | U-0 | R/W-0/u | R/W-1/u | U-0 | R-0/u | R-0/u | U-0 |
|---|---|---|---|---|---|---|---|
| — | — | IRCF<1:0> | | — | HTS | LTS | — |
| bit 7 | | | | | | | bit 0 |

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

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

bit 5-4 **IRCF<1:0>:** Internal Oscillator Frequency Select bits
11 = 8 MHz
10 = 4 MHz
01 = 1 MHz (Reset default)
00 = 31 kHz (LFINTOSC)

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

bit 2 **HTS:** HFINTOSC Status bit
1 = HFINTOSC is stable
0 = HFINTOSC is not stable

bit 1 **LTS:** LFINTOSC Status bit
1 = LFINTOSC is stable
0 = LFINTOSC is not stable

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

### 4.5.1 OSCTUNE REGISTER

The oscillator is factory-calibrated, but can be adjusted in software by writing to the OSCTUNE register (Register 4-2).

The default value of the OSCTUNE register is '0'. The value is a 5-bit two's complement number.

When the OSCTUNE register is modified, the frequency will begin shifting to the new frequency. Code execution continues during this shift. There is no indication that the shift has occurred.

**REGISTER 4-2: OSCTUNE: OSCILLATOR TUNING 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 |
|---|---|---|---|---|---|---|---|
| — | — | — | TUN<4:0> | | | | |
| bit 7 | | | | | | | bit 0 |

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

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

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

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

**TABLE 4-2: 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 | — | — | IRCF<1:0> | | — | HTS | LTS | — | 37 |
| OSCTUNE | — | — | — | TUN<4:0> | | | | | 38 |

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

**TABLE 4-3: SUMMARY OF CONFIGURATION WORD 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 |
|---|---|---|---|---|---|---|---|---|---|---|
| CONFIG(1) | 13:8 | — | — | $\overline{\text{DEBUG}}$ | $\overline{\text{CLKOUTEN}}$ | WRT<1:0> | | BOREN<1:0> | | 150 |
| | 7:0 | — | $\overline{\text{CP}}$ | MCLRE | $\overline{\text{PWRTE}}$ | WDTE | — | — | FOSC0 | |

**Legend:** — = unimplemented location, read as '1'. Shaded cells are not used by clock sources.
**Note 1:** See Configuration Word register (Register 19-1) for operation of all register bits.

## 5.0 I/O PORTS

Depending on the device selected and peripherals enabled, there are up to two ports available. In general, when a peripheral is enabled, that pin may not be used as a general purpose I/O pin.

Each port has three 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)

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

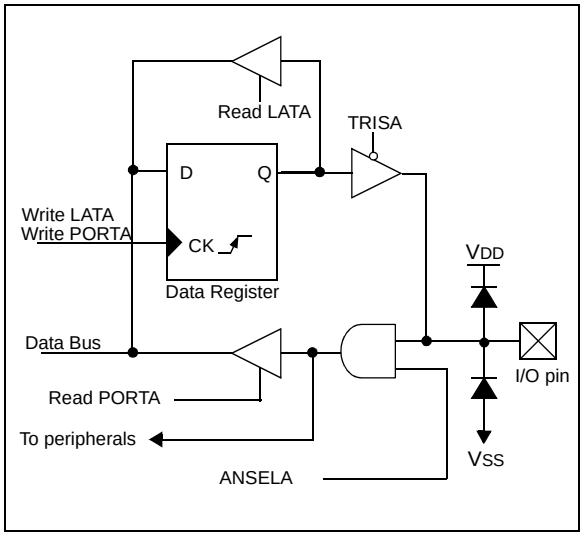
- ANSELx (analog select)
- WPUx (weak pull-up)
- SLRCONx registers (slew rate)

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

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

Ports with analog functions also have an ANSELx register which can disable the digital input and save power. A simplified model of a generic I/O port, without the interfaces to other peripherals, is shown in Figure 5-1.

**FIGURE 5-1: GENERIC I/O PORTA OPERATION**




**EXAMPLE 5-1: 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
```

## 5.1 Alternate Pin Function

The Alternate Pin Function Control (APFCON) register is used to steer specific peripheral input and output functions between different pins. The APFCON register is shown in Register 5-1. For this device family, the following functions can be moved between different pins.

- Timer1 Gate
- COG1

These bits have no effect on the values of any TRIS register. PORT and TRIS overrides will be routed to the correct pin. The unselected pin will be unaffected.

## 5.2 Register Definitions: Alternate Pin Function Control

**REGISTER 5-1: APFCON: ALTERNATE PIN FUNCTION CONTROL REGISTER**

| U-0 | U-0 | U-0 | R/W-0/0 | U-0 | U-0 | U-0 | U-0 |
|---|---|---|---|---|---|---|---|
| — | — | — | T1GSEL | — | — | — | — |
| 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 **T1GSEL:** Timer 1 Gate Input Pin Selection bit
1 = T1G function is on RA3
0 = T1G function is on RA4

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

## 5.3 PORTA and TRISA Registers

PORTA is a 6-bit wide port with five bidirectional and one input-only pin. The corresponding data direction register is TRISA (Register 5-2). 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 5-1 shows how to initialize PORTA.

Reading the PORTA register (Register 5-2) reads the status of the pins, whereas writing to it will write to the PORT latch. All write operations are read-modify-write operations. Therefore, a write to a port implies that the port pins are read, this value is modified and then written to the PORT data latch. RA3 reads '0' when MCLRE = 1.

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

| **Note:** | The ANSEL register must be initialized to configure an analog channel as a digital input. Pins configured as analog inputs will read '0' and cannot generate an interrupt. |
|---|---|

### 5.3.1 PORTA FUNCTIONS AND OUTPUT PRIORITIES

Each PORTA pin is multiplexed with other functions. The pins, their combined functions and their output priorities are shown in Table 5-1.

When multiple outputs are enabled, the actual pin control goes to the peripheral with the highest priority.

Analog input functions, such as comparator inputs, are not shown in the priority lists. These inputs are active when the peripheral is enabled and the input multiplexer for the pin is selected. The Analog mode, set with the ANSELA register, disables the digital input buffer thereby preventing excessive input current when the analog input voltage is between logic states. Digital output functions may control the pin when it is in Analog mode with the priority shown in Table 5-1.

**TABLE 5-1: PORTA OUTPUT PRIORITY**

| Pin Name | Function Priority |
|---|---|
| RA0 | ICSPDAT<br>FVROUT<br>DACOUT<br>C1IN0+<br>RA0 |
| RA1 | FVRIN<br>ICSPCLK<br>VREF+<br>C1IN0-<br>C2IN0-<br>RA1 |
| RA2 | COG1FLT<br>T0CKI<br>C1OUT<br>INT<br>RA2 |
| RA3 | $\overline{\text{MCLR}}$<br>VPP<br>T1G<br>RA3 |
| RA4 | CLKOUT<br>T1G<br>RA4 |
| RA5 | CLKIN<br>T1CKI<br>RA5 |

## 5.4 Additional Pin Functions

Every PORTA pin on the PIC16F753 has an interrupt-on-change option and a weak pull-up option. The next three sections describe these functions.

### 5.4.1 ANSELA REGISTER

The ANSELA register (Register 5-5) 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. |

### 5.4.2 WEAK PULL-UPS

Each of the PORTA pins, except RA3, has an individually configurable internal weak pull-up. Control bits WPUx enable or disable each pull-up. Refer to Register 5-6. Each weak pull-up is automatically turned off when the port pin is configured as an output. The pull-ups are disabled on a Power-on Reset by the RAPU bit of the OPTION_REG register. A weak pull-up is automatically enabled for RA3 when configured as MCLR and disabled when RA3 is an I/O. There is no software control of the MCLR pull-up.

### 5.4.3 INTERRUPT-ON-CHANGE

Each PORTA pin is individually configurable as an interrupt-on-change pin. Control bits IOCA enable or disable the interrupt function for each pin. Refer to Register 5-7. The interrupt-on-change is disabled on a Power-on Reset.

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

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

a) Any read of PORTA AND Clear flag bit IOCIF. This will end the mismatch condition;

   OR

b) Any write of PORTA AND Clear flag bit IOCIF will end the mismatch condition;

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

| | |
|---|---|
| **Note:** | If a change on the I/O pin should occur when any PORTA operation is being executed, then the IOCIF interrupt flag may not get set. |

## 5.5 Register Definitions: PORTA Control

### REGISTER 5-2: PORTA: PORTA REGISTER

| U-0 | U-0 | R/W-x/u | R/W-x/u | R-x/x | R/W-x/u | R/W-x/u | R/W-x/u |
|---|---|---|---|---|---|---|---|
| — | — | 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 ≥ $V_{IH}$
0 = Port pin is ≤ $V_{IL}$

**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 5-3: TRISA: PORTA TRI-STATE REGISTER

| U-0 | U-0 | R/W-1/1 | R/W-1/1 | R-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 |
|---|---|---|---|---|---|---|---|
| — | — | TRISA5 | TRISA4 | TRISA3(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 5-0 **TRISA<5:0>:** PORTA Tri-State Control bits(1)
1 = PORTA pin configured as an input (tri-stated)
0 = PORTA pin configured as an output

**Note 1:** TRISA3 always reads '1'.

### REGISTER 5-4: 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>**: PORTA Output Latch Value bits(1)

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

bit 2-0 **LATA<2:0>**: PORTA 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 5-5: ANSELA: PORTA ANALOG SELECT REGISTER**

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

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

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

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

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

bit 2-0 ANSA<2:0> Analog Select Between Analog or Digital Function on Pin RA<2:0> bits
1 = Analog input. Pin is assigned as analog input.(1)
0 = Digital I/O. Pin is assigned to port or special function.

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

**REGISTER 5-6: WPUA: WEAK PULL-UP PORTA REGISTER**

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

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

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

bit 5-0 **WPUA<5:0>:** Weak Pull-up Control bits(1,2,3)
1 = Pull-up enabled
0 = Pull-up disabled

**Note 1:** Global $\overline{\text{RAPU}}$ must be enabled for individual pull-ups to be enabled.
**2:** The weak pull-up device is automatically disabled if the pin is in Output mode (TRISA = 0).
**3:** The RA3 pull-up is enabled when configured as $\overline{\text{MCLR}}$ in the Configuration Word, otherwise it is disabled as an input and reads as '0'.

**REGISTER 5-7: IOCAP: INTERRUPT-ON-CHANGE 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 Positive Edge Enable bits
1 = Interrupt-on-Change enabled on the pin for a positive going edge. Associated Status bit and interrupt flag will be set upon detecting an edge.
0 = Interrupt-on-Change disabled for the associated pin.

**REGISTER 5-8: IOCAN: INTERRUPT-ON-CHANGE 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 Negative Edge Enable bits
1 = Interrupt-on-Change enabled on the pin for a negative going edge. Associated Status bit and interrupt flag will be set upon detecting an edge.
0 = Interrupt-on-Change disabled for the associated pin.

**REGISTER 5-9: IOCAF: INTERRUPT-ON-CHANGE 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 Flag bits
1 = An enabled change was detected on the associated pin.
Set when IOCAPx = 1 and a rising edge was detected on RBx, or when IOCANx = 1 and a falling edge was detected on RAx.
0 = No change was detected, or the user cleared the detected change.

**TABLE 5-2: 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 |
|---|---|---|---|---|---|---|---|---|---|
| ADCON0 | ADFM | — | CHS<3:0> | | | | GO/$\overline{\text{DONE}}$ | ADON | 109 |
| ADCON1 | — | ADCS<2:0> | | | — | — | — | ADPREF1 | 110 |
| ANSELA | — | — | — | ANSA4 | — | ANSA2 | ANSA1 | ANSA0 | 44 |
| APFCON | — | — | — | T1GSEL | — | — | — | — | 40 |
| CM1CON0 | C1ON | C1OUT | C1OE | C1POL | C1ZLF | C1SP | C1HYS | C1SYNC | 129 |
| CM2CON0 | C2ON | C2OUT | C2OE | C2POL | C2ZLF | C2SP | C2HYS | C2SYNC | 129 |
| CM1CON1 | C1INTP | C1INTN | C1PCH<2:0> | | | C1NCH<2:0> | | | 130 |
| CM2CON1 | C2INTP | C2INTN | C2PCH<2:0> | | | C2NCH<2:0> | | | 130 |
| DAC1CON0 | DACEN | DACFM | DACOE | — | DACPSS1 | DACPSS0 | — | — | 120 |
| IOCAF | — | — | IOCAF5 | IOCAF4 | IOCAF3 | IOCAF2 | IOCAF1 | IOCAF0 | 45 |
| IOCAN | — | — | IOCAN5 | IOCAN4 | IOCAN3 | IOCAN2 | IOCAN1 | IOCAN0 | 45 |
| IOCAP | — | — | IOCAP5 | IOCAP4 | IOCAP3 | IOCAP2 | IOCAP1 | IOCAP0 | 45 |
| LATA | — | — | LATA5 | LATA4 | — | LATA2 | LATA1 | LATA0 | 43 |
| OPTION_REG | $\overline{\text{RAPU}}$ | INTEDG | T0CS | T0SE | PSA | PS<2:0> | | | 16 |
| PORTA | — | — | RA5 | RA4 | RA3 | RA2 | RA1 | RA0 | 43 |
| TRISA | — | — | TRISA5 | TRISA4 | TRISA3(1) | TRISA2 | TRISA1 | TRISA0 | 43 |

**Legend:** x = unknown, u = unchanged, – = unimplemented locations read as '0'. Shaded cells are not used by PORTA.
**Note 1:** TRISA3 always reads '1'.

## 5.6 PORTC Registers

PORTC is a 6-bit wide port with five bidirectional and one input-only pin. The corresponding data direction register is TRISC (Register 5-2). Setting a TRISC bit (= 1) will make the corresponding PORTC pin an input (i.e., disable the output driver). Clearing a TRISC bit (= 0) will make the corresponding PORTC 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 5-1 shows how to initialize PORTC.

Reading the PORTC register (Register 5-2) 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. RC3 reads '0' when MCLRE = 1.

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

| | |
|---|---|
| **Note:** | The ANSEL register must be initialized to configure an analog channel as a digital input. Pins configured as analog inputs will read '0' and cannot generate an interrupt. |

### 5.6.1 PORTC FUNCTIONS AND OUTPUT PRIORITIES

Each PORTC pin is multiplexed with other functions. The pins, their combined functions and their output priorities are shown in Table 5-1.

When multiple outputs are enabled, the actual pin control goes to the peripheral with the highest priority.

Analog input functions, such as comparator inputs, are not shown in the priority lists. These inputs are active when the peripheral is enabled and the input multiplexer for the pin is selected. The Analog mode, set with the ANSELC register, disables the digital input buffer thereby preventing excessive input current when the analog input voltage is between logic states. Digital output functions may control the pin when it is in Analog mode with the priority shown in Table 5-1.

**TABLE 5-3: PORTC OUTPUT PRIORITY**

| Pin Name | Function Priority |
|---|---|
| RC0 | OPA1IN+<br>C2IN0+<br>RC0 |
| RC1 | OPA1IN-<br>C1IN1-<br>C2IN1-<br>RC1 |
| RC2 | SLPCIN<br>OPA1OUT<br>C1IN2-<br>C2IN2-<br>RC2 |
| RC3 | C1IN3-<br>C2IN3-<br>RC3 |
| RC4 | COG1OUT1<br>C2OUT<br>RC4 |
| RC5 | COG1OUT0<br>CCP1<br>RC5 |

## 5.7 Additional Pin Functions

Every PORTC pin on the PIC16F753 has an interrupt-on-change option and a weak pull-up option. The next three sections describe these functions.

### 5.7.1 ANSELC REGISTER

The ANSELC register (Register 5-5) 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 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 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. |

### 5.7.2 WEAK PULL-UPS

Each of the PORTC pins, except RC3, has an individually configurable internal weak pull-up. Control bits WPUx enable or disable each pull-up. Refer to Register 5-6. Each weak pull-up is automatically turned off when the port pin is configured as an output. The pull-ups are disabled on a Power-on Reset by the RAPU bit of the OPTION_REG register. A weak pull-up is automatically enabled for RC3 when configured as MCLR and disabled when RC3 is an I/O. There is no software control of the MCLR pull-up.

### 5.7.3 INTERRUPT-ON-CHANGE

Each PORTC pin is individually configurable as an interrupt-on-change pin. Control bit IOCC enables or disables the interrupt function for each pin. Refer to Register 5-7. The interrupt-on-change is disabled on a Power-on Reset.

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

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

a) Any read of PORTC AND Clear flag bit IOCIF. This will end the mismatch condition;

   OR

b) Any write of PORTC AND Clear flag bit IOCIF will end the mismatch condition;

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

| | |
|---|---|
| **Note:** | If a change on the I/O pin should occur when any PORTC operation is being executed, then the IOCIF interrupt flag may not get set. |

### 5.7.4 SLEW RATE CONTROL

Two of the PORTC pins, RC4 and RC5, are equipped with high current driver circuitry. The SLRCONC register provides reduced slew rate control to mitigate possible EMI radiation from these pins.

## 5.8 Register Definitions: PORTC Control

**REGISTER 5-10: PORTC: PORTC REGISTER**

| U-0 | U-0 | R/W-x/u | R/W-x/u | R-x/x | R/W-x/u | R/W-x/u | R/W-x/u |
|---|---|---|---|---|---|---|---|
| — | — | 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 7-6 **Unimplemented:** Read as ‘0’

bit 5-0 **RC<5:0>**: PORTC I/O Value bits(1)
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.

**REGISTER 5-11: TRISC: PORTC TRI-STATE REGISTER**

| U-0 | U-0 | R/W-1/1 | R/W-1/1 | R-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 |
|---|---|---|---|---|---|---|---|
| — | — | 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 7-6 **Unimplemented:** Read as ‘0’

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

**REGISTER 5-12: LATC: PORTC DATA LATCH 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 |
|---|---|---|---|---|---|---|---|
| — | — | 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 7-6 **Unimplemented:** Read as ‘0’

bit 5-0 **LATC<5:0>**: PORTC Output Latch Value bits(1)

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

**REGISTER 5-13: SLRCONC: SLEW RATE CONTROL REGISTER**

| U-0 | U-0 | R/W-1/1 | R/W-1/1 | U-0 | U-0 | U-0 | U-0 |
|---|---|---|---|---|---|---|---|
| — | — | SLRC5 | SLRC4 | — | — | — | — |
| bit 7 | | | | | | | bit 0 |

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

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

bit 5-4 **SLRC<5:4>:** Slew Rate Control Register bit
1 = Slew rate control enabled
0 = Slew rate control disabled

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

**REGISTER 5-14: ANSELC: PORTC ANALOG SELECT REGISTER**

| U-0 | U-0 | U-0 | U-0 | R/W-1 | R/W-1 | R/W-1 | R/W-1 |
|---|---|---|---|---|---|---|---|
| — | — | — | — | ANSC3 | ANSC2 | ANSC1 | ANSC0 |
| bit 7 | | | | | | | bit 0 |

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

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

bit 3-0 **ANSC<3:0>:** Analog Select Between Analog or Digital Function on Pin RC<3:0> bits
1 = Analog input. Pin is assigned as analog input.(1)
0 = Digital I/O. Pin is assigned to port or special function.

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

**REGISTER 5-15: WPUC: WEAK PULL-UP PORTC REGISTER**

| U-0 | U-0 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 |
|---|---|---|---|---|---|---|---|
| — | — | WPUC5 | WPUC4 | WPUC3 | WPUC2 | WPUC1 | WPUC0 |
| bit 7 | | | | | | | bit 0 |

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

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

bit 5-0 **WPUC<5:0>:** Weak Pull-up Control bits**(1,2,3)**
1 = Pull-up enabled
0 = Pull-up disabled

**Note 1:** Global $\overline{\text{RAPU}}$ must be enabled for individual pull-ups to be enabled.
**2:** The weak pull-up device is automatically disabled if the pin is in Output mode (TRISC = 0).
**3:** The RC3 pull-up is enabled when configured as $\overline{\text{MCLR}}$ in the Configuration Word, otherwise it is disabled as an input and reads as ‘0’.

**REGISTER 5-16: IOCCP: INTERRUPT-ON-CHANGE 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 |
|---|---|---|---|---|---|---|---|
| — | — | 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 7-6 **Unimplemented**: Read as ‘0’

bit 5-0 **IOCCP<5:0>:** Interrupt-on-Change Positive Edge Enable bits
1 = Interrupt-on-Change enabled on the pin for a positive going edge. Associated Status bit and interrupt flag will be set upon detecting an edge.
0 = Interrupt-on-Change disabled for the associated pin.

**REGISTER 5-17: IOCCN: INTERRUPT-ON-CHANGE 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 |
|---|---|---|---|---|---|---|---|
| — | — | 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 7-6 **Unimplemented**: Read as '0'

bit 5-0 **IOCCN<5:0>:** Interrupt-on-Change Negative Edge Enable bits

1 = Interrupt-on-Change enabled on the pin for a negative going edge. Associated Status bit and interrupt flag will be set upon detecting an edge.

0 = Interrupt-on-Change disabled for the associated pin.

**REGISTER 5-18: IOCCF: INTERRUPT-ON-CHANGE 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 |
|---|---|---|---|---|---|---|---|
| — | — | 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 7-6 **Unimplemented**: Read as '0'

bit 5-0 **IOCCF<5:0>:** Interrupt-on-Change Flag bits

1 = An enabled change was detected on the associated pin.
Set when IOCCPx = 1 and a rising edge was detected on RBx, or when IOCCNx = 1 and a falling edge was detected on RCx.

0 = No change was detected, or the user cleared the detected change.

**TABLE 0-1: 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 |
|---|---|---|---|---|---|---|---|---|---|
| ADCON0 | ADFM | — | CHS<3:0> | | | | GO/$\overline{\text{DONE}}$ | ADON | 109 |
| ADCON1 | — | ADCS<2:0> | | | — | — | — | ADPREF1 | 110 |
| ANSELC | — | — | — | — | ANSC3 | ANSC2 | ANSC1 | ANSC0 | 44 |
| APFCON | — | — | — | T1GSEL | — | — | — | — | 40 |
| CM1CON0 | C1ON | C1OUT | C1OE | C1POL | C1ZLF | C1SP | C1HYS | C1SYNC | 129 |
| CM2CON0 | C2ON | C2OUT | C2OE | C2POL | C2ZLF | C2SP | C2HYS | C2SYNC | 129 |
| CM1CON1 | C1INTP | C1INTN | C1PCH<2:0> | | | C1NCH<2:0> | | | 130 |
| CM2CON1 | C2NTP | C2INTN | C2PCH<2:0> | | | C2NCH<2:0> | | | 130 |
| DAC1CON0 | DACEN | DACFM | DACOE | — | DACPSS1 | DACPSS0 | — | — | 120 |
| IOCCF | — | — | IOCCF5 | IOCCF4 | IOCCF3 | IOCCF2 | IOCCF1 | IOCCF0 | 45 |
| IOCCN | — | — | IOCCN5 | IOCCN4 | IOCCN3 | IOCCN2 | IOCCN1 | IOCCN0 | 45 |
| IOCCP | — | — | IOCCP5 | IOCCP4 | IOCCP3 | IOCCP2 | IOCCP1 | IOCCP0 | 45 |
| LATC | — | — | LATC5 | LATC4 | LATC3 | LATC2 | LATC1 | LATC0 | 43 |
| OPTION_REG | $\overline{\text{RAPU}}$ | INTEDG | T0CS | T0SE | PSA | PS<2:0> | | | 16 |
| PORTC | — | — | RC5 | RC4 | RC3 | RC2 | RC1 | RC0 | 43 |
| SLRCONC | — | — | SLRC5 | SLRC4 | — | — | — | — | 50 |
| TRISC | — | — | TRISC5 | TRISC4 | TRISC3 | TRISC2 | TRISC1 | TRISC0 | 43 |

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

## 6.0 TIMER0 MODULE

The Timer0 module is an 8-bit timer/counter with the following features:

- 8-bit timer/counter register (TMR0)
- 8-bit prescaler (shared with Watchdog Timer)
- Programmable internal or external clock source
- Programmable external clock edge selection
- Interrupt on overflow

Figure 6-1 is a block diagram of the Timer0 module.

## 6.1 Timer0 Operation

When used as a timer, the Timer0 module can be used as either an 8-bit timer or an 8-bit counter.

### 6.1.1 8-BIT TIMER MODE

When used as a timer, the Timer0 module will increment every instruction cycle (without prescaler). Timer mode is selected by clearing the T0CS bit of the OPTION register to '0'.

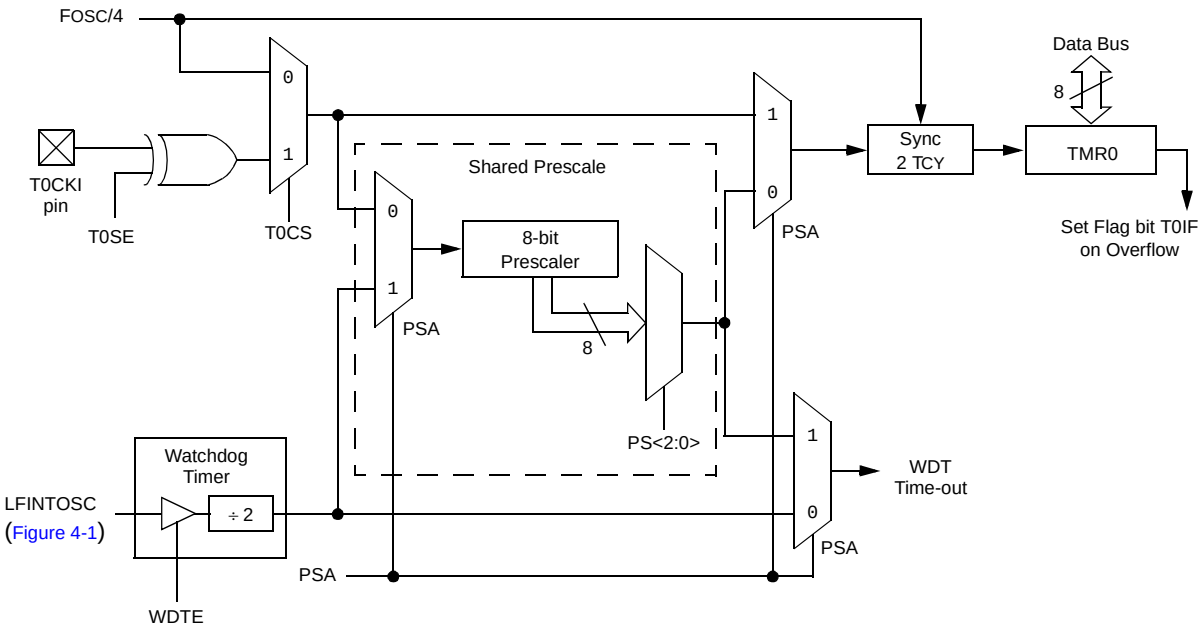
When TMR0 is written, the increment is inhibited for two instruction cycles immediately following the write.

| | |
|---|---|
| **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. |

### 6.1.2 8-BIT COUNTER MODE

When used as a counter, the Timer0 module will increment on every rising or falling edge of the T0CKI pin. The incrementing edge is determined by the T0SE bit of the OPTION_REG register. Counter mode is selected by setting the T0CS bit of the OPTION register to '1'.

**FIGURE 6-1: TIMER0 WITH SHARED PRESCALE BLOCK DIAGRAM**




**Note 1:** T0SE, T0CS, PSA, PS<2:0> are bits in Register 6-1.

**2:** WDTE bit is in Register 19-1.

### 6.1.3 SOFTWARE PROGRAMMABLE PRESCALER

A single software programmable prescaler is available for use with either Timer0 or the Watchdog Timer (WDT), but not both simultaneously. The prescaler assignment is controlled by the PSA bit of the OPTION register. To assign the prescaler to Timer0, the PSA bit must be cleared to a '0'.

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 register. In order to have a 1:1 prescaler value for the Timer0 module, the prescaler must be assigned to the WDT module.

The prescaler is not readable or writable. When assigned to the Timer0 module, all instructions writing to the TMR0 register will clear the prescaler.

When the prescaler is assigned to WDT (PSA = 1), a CLRWDT instruction will clear the prescaler along with the WDT.

#### 6.1.3.1 Switching Prescaler Between Timer0 and WDT Modules

As a result of having the prescaler assigned to either Timer0 or the WDT, it is possible to generate an unintended device Reset when switching prescaler values. When changing the prescaler assignment from Timer0 to the WDT module, the instruction sequence shown in Example 6-1 must be executed.

**EXAMPLE 6-1: CHANGING PRESCALER (TIMER0 → WDT)**

```
BANKSEL TMR0           ;
CLRWDT                 ;Clear WDT
CLRF    TMR0           ;Clear TMR0 and
                       ;prescaler
BANKSEL OPTION_REG     ;
BSF     OPTION_REG,PSA ;Select WDT
CLRWDT                 ;
                       ;
MOVLW   b'11111000'    ;Mask prescaler
ANDWF   OPTION_REG,W   ;bits
IORLW   b'00000101'    ;Set WDT prescaler
MOVWF   OPTION_REG     ;to 1:32
```

When changing the prescaler assignment from the WDT to the Timer0 module, the following instruction sequence must be executed (see Example 6-2).

**EXAMPLE 6-2: CHANGING PRESCALER (WDT → TIMER0)**

```
CLRWDT                 ;Clear WDT and
                       ;prescaler
BANKSEL OPTION_REG     ;
MOVLW   b'11110000'    ;Mask TMR0 select and
ANDWF   OPTION_REG,W   ;prescaler bits
IORLW   b'00000011'    ;Set prescale to 1:16
MOVWF   OPTION_REG     ;
```

### 6.1.4 TIMER0 INTERRUPT

Timer0 will generate an interrupt when the TMR0 register overflows from FFh to 00h. The T0IF 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 T0IF bit must be cleared in software. The Timer0 interrupt enable is the T0IE bit of the INTCON register.

| **Note:** | The Timer0 interrupt cannot wake the processor from Sleep since the timer is frozen during Sleep. |
|---|---|

### 6.1.5 USING TIMER0 WITH AN EXTERNAL CLOCK

When Timer0 is in Counter mode, the synchronization of the T0CKI input and the Timer0 register is accomplished by sampling the prescaler output on the Q2 and Q4 cycles of the internal phase clocks. Therefore, the high and low periods of the external clock source must meet the timing requirements as shown in **Section 22.0 "Electrical Specifications"**.

## 6.2 Register Definitions: Option and Timer0 Control

**REGISTER 6-1: OPTION_REG: OPTION REGISTER**

| R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 |
|---|---|---|---|---|---|---|---|
| $\overline{\text{RAPU}}$ | INTEDG | T0CS | T0SE | PSA | PS<2:0> | | |
| bit 7 | | | | | | | bit 0 |

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

bit 7 **$\overline{\text{RAPU}}$:** PORTA Pull-up Enable bit
1 = PORTA pull-ups are disabled
0 = PORTA pull-ups are enabled by individual PORT latch values in WPU register

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 **T0CS:** TMR0 Clock Source Select bit
1 = Transition on T0CKI pin
0 = Internal instruction cycle clock (FOSC/4)

bit 4 **T0SE:** TMR0 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 assigned to the WDT
0 = Prescaler is assigned to the Timer0 module

bit 2-0 **PS<2:0>:** Prescaler Rate Select bits

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

**TABLE 6-1: 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 |
|---|---|---|---|---|---|---|---|---|---|
| TMR0 | TMR0<7:0> | | | | | | | | 54* |
| INTCON | GIE | PEIE | T0IE | INTE | IOCIE | T0IF | INTF | IOCIF | 17 |
| OPTION_REG | $\overline{\text{RAPU}}$ | INTEDG | T0CS | T0SE | PSA | PS<2:0> | | | 56 |
| TRISA | — | — | TRISA5 | TRISA4 | TRISA3(1) | TRISA2 | TRISA1 | TRISA0 | 43 |

**Legend:** – = Unimplemented locations, read as '0', u = unchanged, x = unknown. Shaded cells are not used by the Timer0 module.
* Page provides register information.
**Note 1:** TRISA3 always reads '1'.

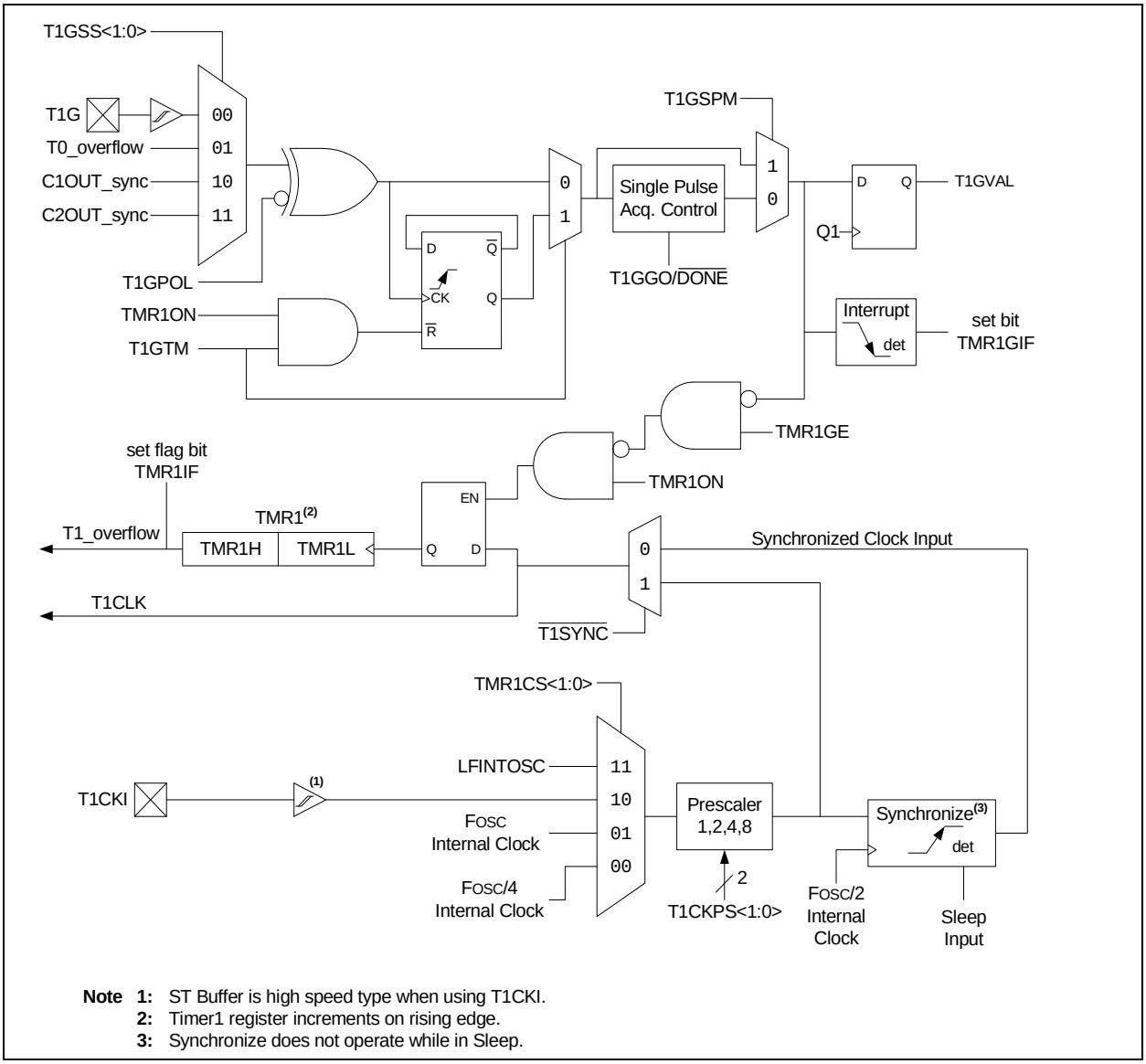
## 7.0 TIMER1 MODULE WITH GATE CONTROL

The Timer1 module is a 16-bit timer/counter with the following features:

- 16-bit timer/counter register pair (TMR1H:TMR1L)
- Selectable internal or external clock sources
- 2-bit prescaler
- Synchronous or asynchronous operation
- 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
- Special Event Trigger (with CCP)
- Selectable Gate Source Polarity
- Gate Toggle mode
- Gate Single-pulse mode
- Gate Value Status
- Gate Event Interrupt

Figure 7-1 is a block diagram of the Timer1 module.

**FIGURE 7-1: 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.

## 7.1 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 7-1 displays the Timer1 enable selections.

**TABLE 7-1: TIMER1 ENABLE SELECTIONS**

| TMR1ON | TMR1GE | Timer1 Operation |
|---|---|---|
| 0 | 0 | Off |
| 0 | 1 | Off |
| 1 | 0 | Always On |
| 1 | 1 | Count Enabled |

## 7.2 Clock Source Selection

The TMR1CS<1:0> bits of the T1CON register are used to select the clock source for Timer1. Table 7-2 displays the clock source selections.

**TABLE 7-2: CLOCK SOURCE SELECTIONS**

| TMR1CS<1:0> | Clock Source |
|---|---|
| 11 | Temperature Sense Oscillator |
| 10 | External Clocking on T1CKI Pin |
| 01 | System Clock (FOSC) |
| 00 | Instruction Clock (FOSC/4) |

### 7.2.1 INTERNAL CLOCK SOURCE

When the internal clock source is selected, the TMR1H:TMR1L register pair will increment on multiples of FOSC or FOSC/4 as determined by the Timer1 prescaler.

### 7.2.2 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.

**Note:** In Counter mode, a falling edge must be registered by the counter prior to the first incrementing rising edge (see Figure 7-2) after any one or more of the following conditions:

- Timer1 enabled after POR Reset
- 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.

### 7.2.3 WDT OSCILLATOR

When the Watchdog is selected, Timer 1 will use the LFINTOSC that is used to operate the Watchdog Timer. This is the same oscillator as the LFINTOSC used as the system clock. Selecting this option will enable the oscillator even when the LFINTOSC or the Watchdog are not in use. This oscillator will continue to operate when in Sleep mode.

## 7.3 Timer1 Prescaler

Timer1 has four prescaler options allowing one, two, four or eight 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.

## 7.4 Timer1 Operation in Asynchronous Counter Mode

If 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 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 7.4.1 "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. |

### 7.4.1 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.

## 7.5 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 count enable.

Timer1 gate can also be driven by multiple selectable sources.

### 7.5.1 TIMER1 GATE COUNT ENABLE

The Timer1 gate is enabled by setting the TMR1GE bit of the T1GCON register. The polarity of the Timer1 gate is configured using the T1GPOL bit of the T1GCON register.

When Timer1 Gate (T1G) input is active, Timer1 will increment on the rising edge of the Timer1 clock source. When Timer1 gate input is inactive, no incrementing will occur and Timer1 will hold the current count. See Figure 7-3 for timing details.

**TABLE 7-3: TIMER1 GATE ENABLE SELECTIONS**

| T1CLK | T1GPOL | T1G | Timer1 Operation |
|---|---|---|---|
| ↑ | 0 | 0 | Counts |
| ↑ | 0 | 1 | Holds Count |
| ↑ | 1 | 0 | Holds Count |
| ↑ | 1 | 1 | Counts |

### 7.5.2 TIMER1 GATE SOURCE SELECTION

The Timer1 gate source can be selected from one of four different sources. 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 7-4: TIMER1 GATE SOURCES**

| T1GSS | Timer1 Gate Source |
|---|---|
| 11 | SYNCC2OUT |
| 10 | SYNCC1OUT |
| 01 | Overflow of Timer0<br>(TMR0 increments from FFh to 00h) |
| 00 | Timer1 Gate Pin |

#### 7.5.2.1 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.

#### 7.5.2.2 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.

#### 7.5.2.3 C1OUT/C2OUT Gate Operation

The outputs from the Comparator C1 and C2 modules can be used as gate sources for the Timer1 module.

### 7.5.3 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 7-4 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. |
|---|---|

### 7.5.4 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/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/DONE bit will automatically be cleared. No other gate events will be allowed to increment Timer1 until the T1GGO/DONE bit is once again set in software.

Clearing the T1GSPM bit of the T1GCON register will also clear the T1GGO/DONE bit. See Figure 7-5 for timing details.

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 7-6 for timing details.

### 7.5.5 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).

### 7.5.6 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).

## 7.6 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. |
|---|---|

## 7.7 Timer1 Operation During Sleep

Timer1 can only operate during Sleep when set up in Asynchronous Counter mode or with the internal watchdog clock source. In this mode, the 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
- TMR1GE bit of the T1GCON 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 (0004h).

## 7.8 CCP Capture/Compare Time Base

The CCP module uses 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 a Special Event Trigger.

For more information, see **Section 10.0 "Capture/Compare/PWM Modules"**.

## 7.9 CCP Special Event Trigger

When the CCP is configured to trigger a special event, the trigger will clear the TMR1H:TMR1L register pair. This special event does not cause a Timer1 interrupt. The CCP module may still be configured to generate a CCP interrupt.

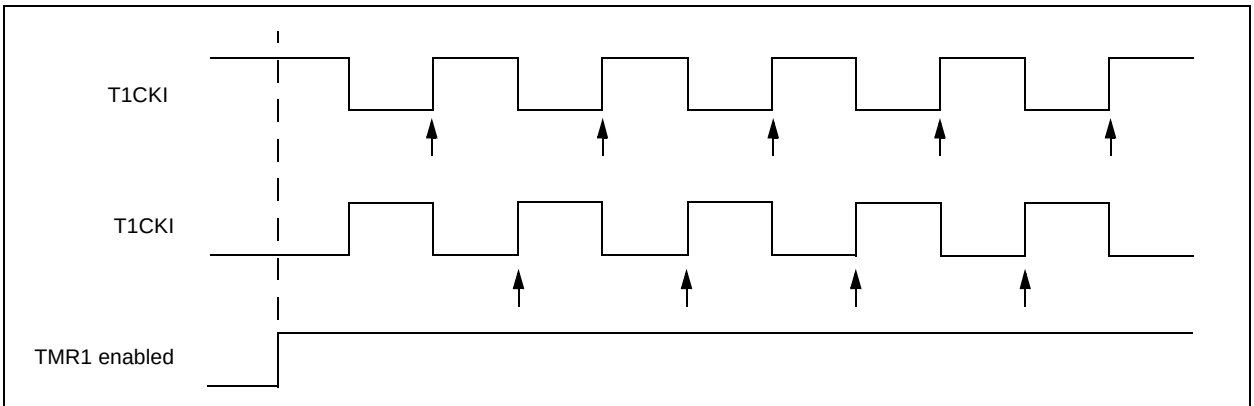
In this mode of operation, the CCPR1H:CCPR1L register pair becomes the period register for Timer1.

Timer1 should be synchronized to the FOSC/4 to utilize the Special Event Trigger. Asynchronous operation of Timer1 can cause a Special Event Trigger to be missed.

In the event that a write to TMR1H or TMR1L coincides with a Special Event Trigger from the CCP, the write will take precedence.

For more information, see **Section 12.2.5 "Special Event Trigger"**.

**FIGURE 7-2: 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 7-3: TIMER1 GATE COUNT ENABLE MODE**

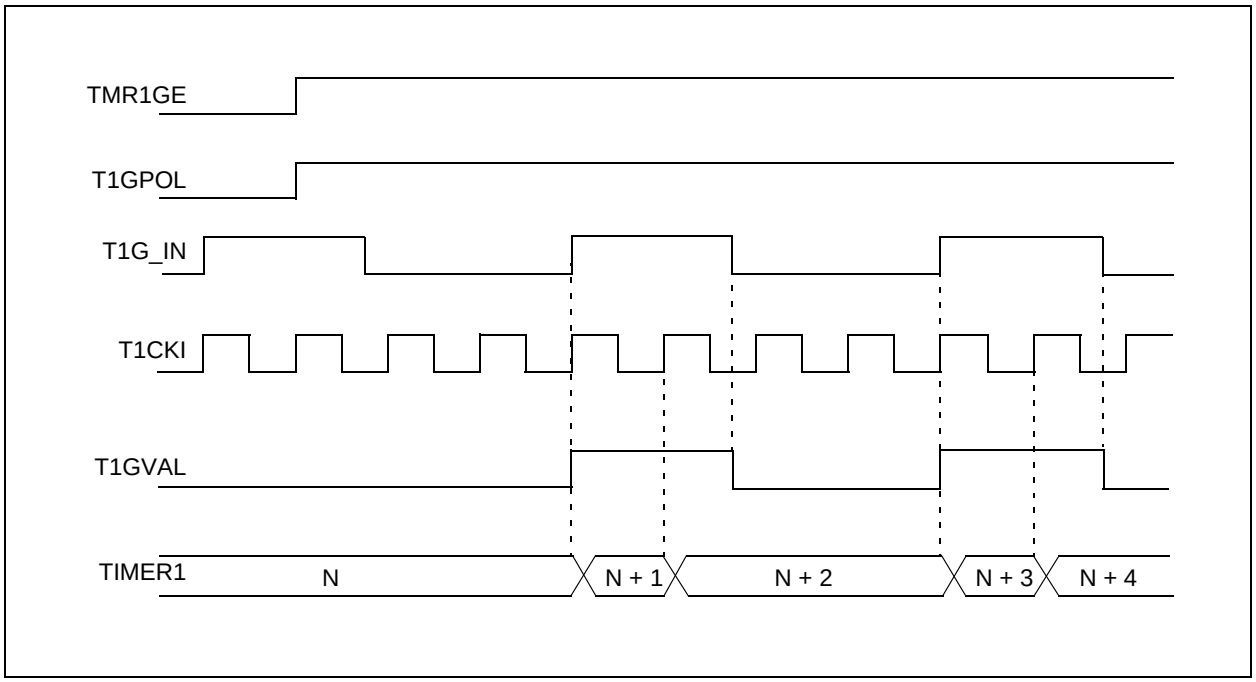



**FIGURE 7-4: TIMER1 GATE TOGGLE MODE**

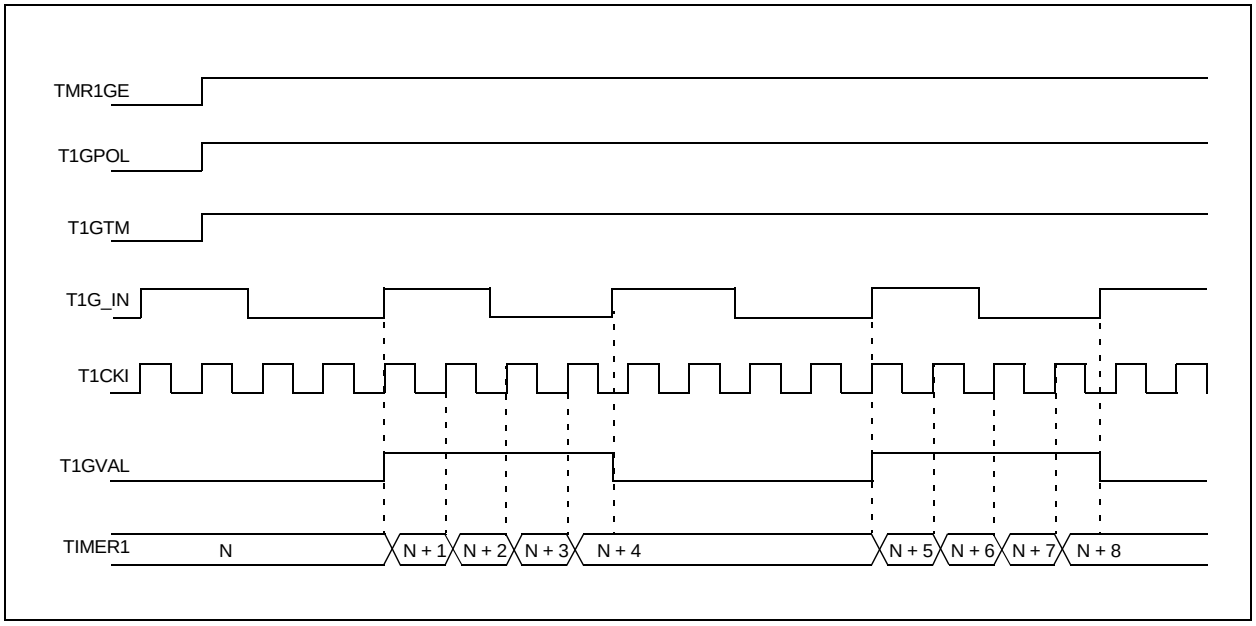


**FIGURE 7-5: TIMER1 GATE SINGLE-PULSE MODE**

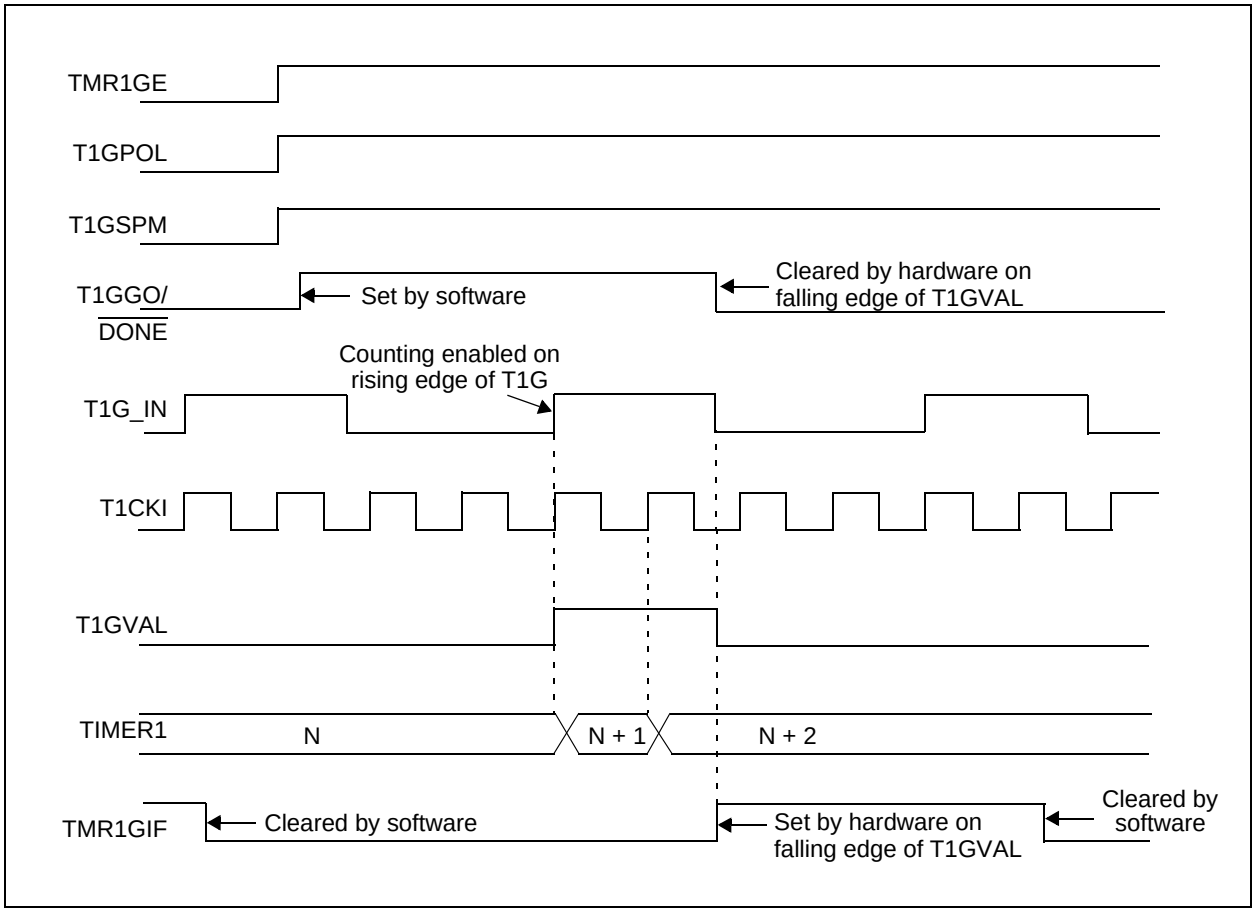
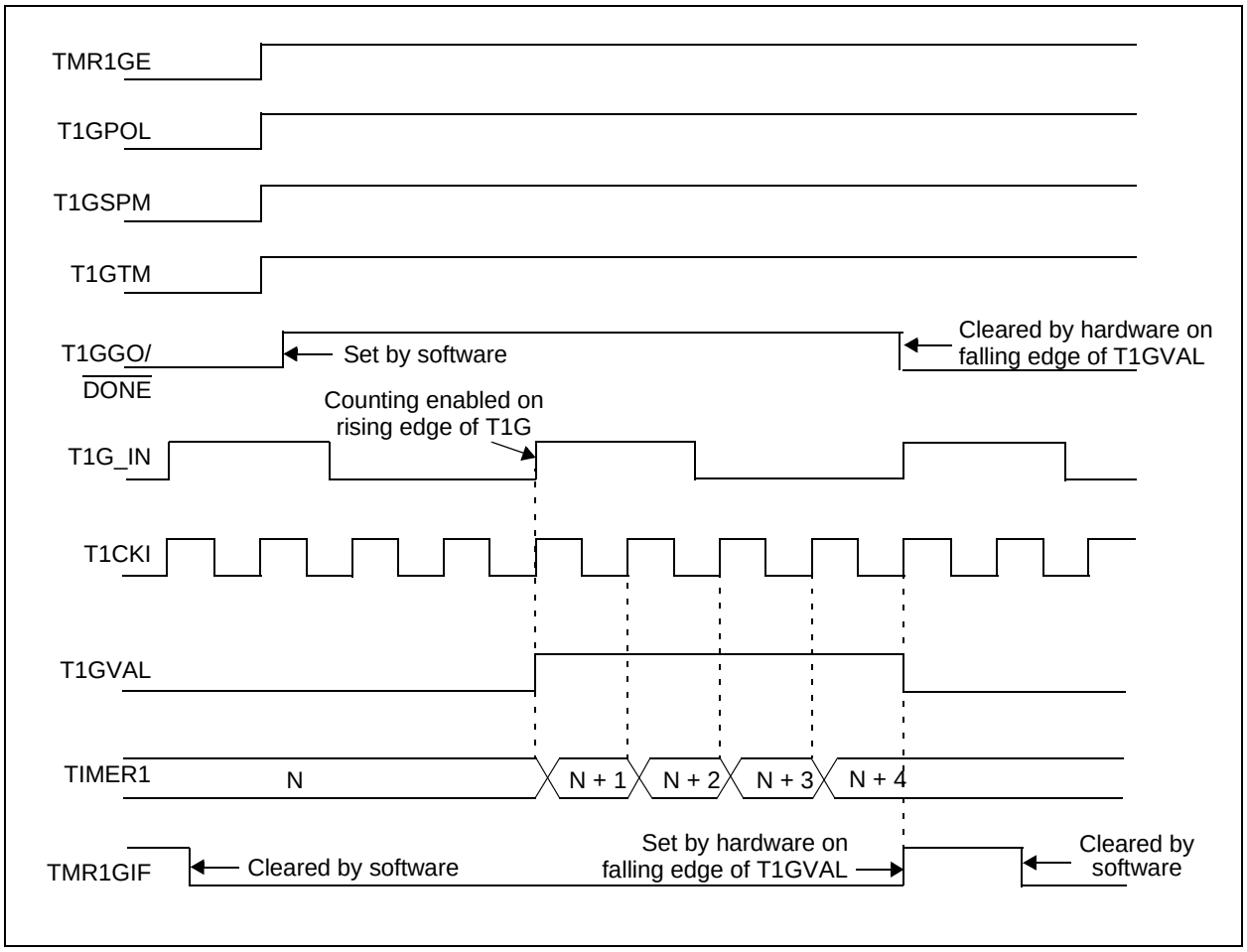


**FIGURE 7-6: TIMER1 GATE SINGLE-PULSE AND TOGGLE COMBINED MODE**



## 7.10 Register Definitions: Timer1 Control

**REGISTER 7-1: T1CON: TIMER1 CONTROL REGISTER**

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | U-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| TMR1CS<1:0> | | T1CKPS<1:0> | | T1OSCEN | $\overline{\text{T1SYNC}}$ | — | TMR1ON |
| bit 7 | | | | | | | bit 0 |

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

bit 7-6 **TMR1CS<1:0>:** Timer1 Clock Source Select bits
- 11 = Watchdog timer oscillator
- 10 = External clock from T1CKI pin (on the 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 **$\overline{\text{T1OSCEN}}$**: This bit is ignored.

bit 2 **T1SYNC:** Timer1 External Clock Input Synchronization Control bit

TMR1CS<1:0> = 1X
- 1 = Do not synchronize external clock input
- 0 = Synchronize external clock input with system clock (FOSC)

TMR1CS<1:0> = 0X
This bit is ignored. Timer1 uses the internal clock when TMR1CS<1:0> = 1X.

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

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

Clears Timer1 gate flip-flop

**REGISTER 7-2: T1GCON: TIMER1 GATE CONTROL REGISTER**

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R-x | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| 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' | |
| -n = Value at POR | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |

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
This bit is automatically cleared when T1GSPM is cleared.

bit 2 **T1GVAL:** Timer1 Gate Current State 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 = SYNCC2OUT
10 = SYNCC1OUT
01 = Timer0 overflow output
00 = Timer1 gate pin

**TABLE 7-5: 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 | 44 |
| APFCON | — | — | — | T1GSEL | — | — | — | — | 40 |
| CCP1CON | — | — | DC1B<1:0> | | CCP1M<3:0> | | | | 80 |
| INTCON | GIE | PEIE | T0IE | INTE | IOCIE | T0IF | INTF | IOCIF | 17 |
| PIE1 | TMR1GIE | ADIE | — | — | HLTMR2IE | HLTMR1IE | TMR2IE | TMR1IE | 18 |
| PIR1 | TMR1GIF | ADIF | — | — | HLTMR2IF | HLTMR1IF | TMR2IF | TMR1IF | 20 |
| PORTA | — | — | RA5 | RA4 | RA3 | RA2 | RA1 | RA0 | 43 |
| TMR1H | TMR1H<7:0> | | | | | | | | 57* |
| TMR1L | TMR1L<7:0> | | | | | | | | 57* |
| TRISA | — | — | TRISA5 | TRISA4 | TRISA3 | TRISA2 | TRISA1 | TRISA0 | 43 |
| T1CON | TMR1CS<1:0> | | T1CKPS<1:0> | | T1OSCEN | $\overline{\text{T1SYNC}}$ | — | TMR1ON | 65 |
| T1GCON | TMR1GE | T1GPOL | T1GTM | T1GSPM | T1GGO/$\overline{\text{DONE}}$ | T1GVAL | T1GSS<1:0> | | 66 |

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

## 8.0 TIMER2 MODULE

The Timer2 module is an 8-bit timer with the following features:

- 8-bit timer register (TMR2)
- 8-bit period register (PR2)
- Interrupt on TMR2 match with PR2
- Software programmable prescaler (1:1, 1:4, 1:16, 1:64)
- Software programmable postscaler (1:1 to 1:16)

See Figure 8-1 for a block diagram of Timer2.

### 8.1 Timer2 Operation

The clock input to the Timer2 module is the system instruction clock (FOSC/4). The clock is fed into the Timer2 prescaler, which has prescale options of 1:1, 1:4 or 1:16. The output of the prescaler is then used to increment the TMR2 register.

The values of TMR2 and PR2 are constantly compared to determine when they match. TMR2 will increment from 00h until it matches the value in PR2. When a match occurs, two things happen:

- TMR2 is reset to 00h on the next increment cycle.
- The Timer2 postscaler is incremented

The match output of the Timer2/PR2 comparator is then fed into the Timer2 postscaler. The postscaler has postscale options of 1:1 to 1:16 inclusive. The output of the Timer2 postscaler is used to set the TMR2IF interrupt flag bit in the PIR1 register.

The TMR2 and PR2 registers are both fully readable and writable. On any Reset, the TMR2 register is set to 00h and the PR2 register is set to FFh.

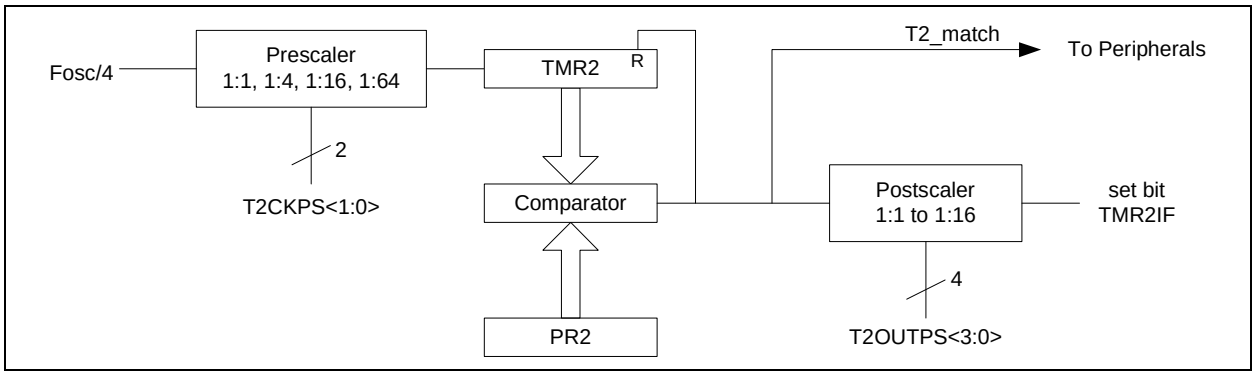
Timer2 is turned on by setting the TMR2ON bit in the T2CON register to a '1'. Timer2 is turned off by clearing the TMR2ON bit to a '0'.

The Timer2 prescaler is controlled by the T2CKPS bits in the T2CON register. The Timer2 postscaler is controlled by the T2OUTPS bits in the T2CON register. The prescaler and postscaler counters are cleared when:

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

| **Note:** | TMR2 is not cleared when T2CON is written. |
|---|---|

**FIGURE 8-1: TIMER2 BLOCK DIAGRAM**



## 8.2 Register Definitions: Timer2 Control

**REGISTER 8-1: T2CON: TIMER2 CONTROL REGISTER**

| U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-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’ | |
| -n = Value at POR | ‘1’ = Bit is set | ‘0’ = Bit is cleared | x = Bit is unknown |

bit 7 **Unimplemented:** Read as ‘0’

bit 6-3 **T2OUTPS<3:0>:** Timer2 Output Postscaler Select bits
0000 = 1:1 Postscaler
0001 = 1:2 Postscaler
0010 = 1:3 Postscaler
0011 = 1:4 Postscaler
0100 = 1:5 Postscaler
0101 = 1:6 Postscaler
0110 = 1:7 Postscaler
0111 = 1:8 Postscaler
1000 = 1:9 Postscaler
1001 = 1:10 Postscaler
1010 = 1:11 Postscaler
1011 = 1:12 Postscaler
1100 = 1:13 Postscaler
1101 = 1:14 Postscaler
1110 = 1:15 Postscaler
1111 = 1:16 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
00 = Prescaler is 1
01 = Prescaler is 4
10 = Prescaler is 16
11 = Prescaler is 64

**TABLE 8-1: 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 |
|---|---|---|---|---|---|---|---|---|---|
| INTCON | GIE | PEIE | T0IE | INTE | IOCIE | T0IF | INTF | IOCIF | 17 |
| PIE1 | TMR1GIE | ADIE | — | — | HLTMR2IE | HLTMR1IE | TMR2IE | TMR1IE | 18 |
| PIR1 | TMR1GIF | ADIF | — | — | HLTMR2IF | HLTMR1IF | TMR2IF | TMR1IF | 20 |
| PR2 | PR2<7:0> | | | | | | | | 68* |
| TMR2 | TMR2<7:0> | | | | | | | | 68* |
| T2CON | — | T2OUTPS<3:0> | | | | TMR2ON | T2CKPS<1:0> | | 69 |

**Legend:** x = unknown, u = unchanged, - = unimplemented read as ‘0’. Shaded cells are not used for Timer2 module.
* Page provides register information.

## 9.0 HARDWARE LIMIT TIMER (HLT) MODULE

The Hardware Limit Timer (HLT) module is a version of the Timer2-type modules. In addition to all the Timer2-type features, the HLT can be reset on rising and falling events from selected peripheral outputs.

The HLT primary purpose is to act as a timed hardware limit to be used in conjunction with asynchronous analog feedback applications. The external Reset source synchronizes the HLTMRx to an analog application.

In normal operation, the external Reset source from the analog application should occur before the HLTMRx matches the HLTPRx. This resets HLTMRx for the next period and prevents the HLTimerx Output from going active.

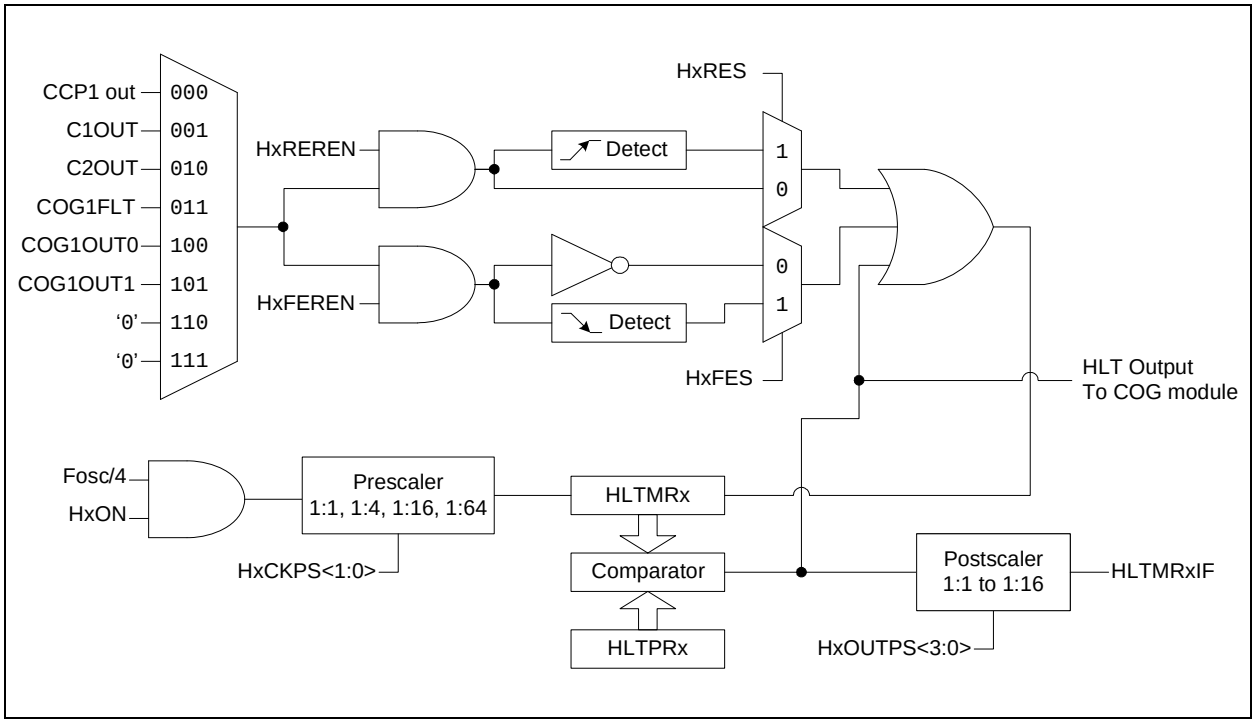
When the external Reset source fails to generate a signal within the expected time, (allowing the HLTMRx to match the HLTPRx), then the HLTimerx Output becomes active.

The HLT module incorporates the following features:

- 8-bit Read-Write Timer Register (HLTMRx)
- 8-bit Read-Write Period register (HLTPRx)
- Software programmable prescaler:
  - 1:1
  - 1:4
  - 1:16
  - 1:64
- Software programmable postscaler
  - 1:1 to 1:16, inclusive
- Interrupt on HLTMRx match with HLTPRx
- Eight selectable timer Reset inputs (two reserved)
- Reset on rising and falling event

Refer to Figure 9-1 for a block diagram of the HLT.

**FIGURE 9-1: HLTMRx BLOCK DIAGRAM**



## 9.1 HLT Operation

The clock input to the HLT module is the system instruction clock (FOSC/4). HLTMRx increments on each rising clock edge.

A 4-bit counter/prescaler on the clock input provides the following prescale options:

- Direct input
- Divide-by-4
- Divide-by-16
- Divide-by-64

The prescale options are selected by the prescaler control bits, HxCKPS<1:0> of the HLTxCON0 register.

The value of HLTMRx is compared to that of the Period register, HLTPRx, on each clock cycle. When the two values match, then the comparator generates a match signal as the HLTimerx output. This signal also resets the value of HLTMRx to 00h on the next clock rising edge and drives the output counter/postscaler (see **Section 9.2 "HLT Interrupt"**).

The HLTMRx and HLTPRx registers are both directly readable and writable. The HLTMRx register is cleared on any device Reset, whereas the HLTPRx register initializes to FFh. Both the prescaler and postscaler counters are cleared on any of the following events:

- A write to the HLTMRx register
- A write to the HLTxCON0 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:** | HLTMRx is not cleared when HLTxCON0 is written. |
|---|---|

## 9.2 HLT Interrupt

The HLT can also generate an optional device interrupt. The HLTMRx output signal (HLTMRx-to-HLTPRx match) provides the input for the 4-bit counter/postscaler. The overflow output of the postscaler sets the HLTMRxIF bit of the PIR1 register. The interrupt is enabled by setting the HLTMRx Match Interrupt Enable bit, HLTMRxIE 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, HxOUTPS<3:0>, of the HLTxCON0 register.

## 9.3 Peripheral Resets

Resets driven from the selected peripheral output prevents the HLTMRx from matching the HLTPRx register and generating an output. In this manner, the HLT can be used as a hardware time limit to other peripherals.

In this device, the primary purpose of the HLT is to limit the COG PWM duty cycle. Normally, the COG operation uses analog feedback to determine the PWM duty cycle. The same feedback signal is used as an HLT Reset input. The HLTPRx register is set to occur at the maximum allowed duty cycle. If the analog feedback to the COG exceeds the maximum time, then an HLTMRx-to-HLTPRx match will occur and generate the output needed to limit the COG drive output.

The HLTMRx can be reset by one of several selectable peripheral sources. Reset inputs include:

- CCP1 output
- Comparator 1 output
- Comparator 2 output
- COGxFLT pin
- COG1OUT0
- COG1OUT1

The external Reset input is selected with the HxERS<2:0> bits of the HLTxCON1 register. High and low Reset enables are selected with the HxREREN and HxFEREN bits, respectively. Setting the HxRES and HxFES bits makes the respective rising and falling Reset events edge sensitive. Reset inputs that are not edge sensitive are level sensitive.

HLTMRx Resets are synchronous with the HLT clock. In other words, HLTMRx is cleared on the rising edge of the HLT clock after the enabled Reset event occurs.

If an enabled external Reset occurs at the same time a write occurs to the TMR4A register, the write to the timer takes precedence and pending Resets are cleared.

## 9.4 HLTimerx Output

The unscaled output of HLTMRx is available only to the COG module, where it is used as a selectable limit to the maximum COG period.

## 9.5 HLT Operation During Sleep

The HLT cannot be operated while the processor is in Sleep mode. The contents of the HLTMRx register will remain unchanged while the processor is in Sleep mode.

## 9.6 Register Definitions: HLT Control Registers

**REGISTER 9-1: HLTxCON0: HLTx 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 |
|---|---|---|---|---|---|---|---|
| — | HxOUTPS<3:0> | | | | HxON | HxCKPS<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 **HxOUTPS<3:0>:** Hardware Limit Timerx Output Postscaler Select bits

0000 = 1:1 Postscaler
0001 = 1:2 Postscaler
0010 = 1:3 Postscaler
0011 = 1:4 Postscaler
0100 = 1:5 Postscaler
0101 = 1:6 Postscaler
0110 = 1:7 Postscaler
0111 = 1:8 Postscaler
1000 = 1:9 Postscaler
1001 = 1:10 Postscaler
1010 = 1:11 Postscaler
1011 = 1:12 Postscaler
1100 = 1:13 Postscaler
1101 = 1:14 Postscaler
1110 = 1:15 Postscaler
1111 = 1:16 Postscaler

bit 2 **HxON:** Hardware Limit Timerx On bit

1 = Timer is ON
0 = Timer is OFF

bit 1-0 **HxCKPS<1:0>:** Hardware Limit Timer x Clock Prescale Select bits

00 = Prescaler is 1
01 = Prescaler is 4
10 = Prescaler is 16
11 = Prescaler is 64

**REGISTER 9-2: HLTxCON1: HLTx CONTROL REGISTER 1**

| R/W-0/0 | R/W-0/0 | U-0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| HxFES | HxRES | — | HxERS<2:0> | | | HxFEREN | HxREREN |
| 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 **HxFES:** Hardware Limit Timerx Falling Edge Sensitivity bit
1 = Edge sensitive
0 = Level sensitive

bit 6 **HxRES:** Hardware Limit Timerx Rising Edge Sensitivity bit
1 = Edge sensitive
0 = Level sensitive

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

bit 4-2 **HxERS<2:0>:** Hardware Limit Timerx External Reset Source Select bits
000 = CCP1 Out
001 = C1OUT
010 = C2OUT
011 = COG1FLT
100 = COG1OUT0
101 = COG1OUT1
110 = Reserved - '0' input
111 = Reserved - '0' input

bit 1 **HxFEREN:** Hardware Limit Timerx Falling Event Reset Enable bit
1 = HLTMRx will reset on the first clock after a falling event of selected Reset source
0 = Falling events of selected source have no effect

bit 0 **HxREREN:** Hardware Limit Timerx Rising Event Reset Enable bit
1 = HLTMRx will reset on the first clock after a rising event of selected Reset source
0 = Rising events of selected source have no effect

**TABLE 9-1: SUMMARY OF REGISTERS ASSOCIATED WITH HLT**

| 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> | | | | 80 |
| CM1CON0 | C1ON | C1OUT | C1OE | C1POL | C1ZLF | C1SP | C1HYS | C1SYNC | 129 |
| CM1CON1 | C1INTP | C1INTN | C1PCH<2:0> | | | C1NCH<2:0> | | | 130 |
| CM2CON0 | C2ON | C2OUT | C2OE | C2POL | C2ZLF | C2SP | C2HYS | C2SYNC | 129 |
| CM2CON1 | C2INTP | C2INTN | C2PCH<2:0> | | | C2NCH<2:0> | | | 130 |
| INTCON | GIE | PEIE | T0IE | INTE | IOCIE | T0IF | INTF | IOCIF | 17 |
| PIE1 | TMR1GIE | ADIE | — | — | HLTMR2IE | HLTMR1IE | TMR2IE | TMR1IE | 18 |
| PIR1 | TMR1GIF | ADIF | — | — | HLTMR2IF | HLTMR1IF | TMR2IF | TMR1IF | 20 |
| HLTMRx | Holding Register for the 8-bit Hardware Limit TimerX Count | | | | | | | | 70* |
| HLTPRx | HLTMRx Module Period Register | | | | | | | | 70* |
| HLTxCON0 | — | HxOUTPS<3:0> | | | | HxON | HxCKPS<1:0> | | 72 |
| HLTxCON1 | HxFES | HxRES | — | HxERS<2:0> | | | HxFEREN | HxREREN | 73 |

**Legend:** — = unimplemented location, read as '0'. Shaded cells do not affect the HLT module operation.
* Page provides register information.

## 10.0 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.

## 10.1 Capture Mode

Capture mode makes use of the 16-bit Timer1 resource. When an event occurs on the CCP1 pin, the 16-bit CCPR1H:CCPR1L 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 CCP1M<3:0> bits of the CCP1CON 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 CCP1IF of the PIR2 register is set. The interrupt flag must be cleared in software. If another capture occurs before the value in the CCPR1H, CCPR1L register pair is read, the old captured value is overwritten by the new captured value.

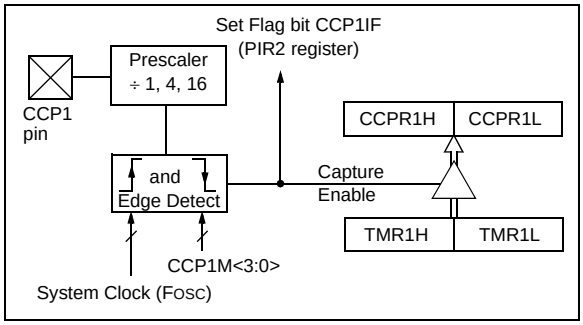
Figure 10-1 shows a simplified diagram of the Capture operation.

### 10.1.1 CCP1 PIN CONFIGURATION

In Capture mode, the CCP1 pin should be configured as an input by setting the associated TRIS control bit.

| **Note:** | If the CCP1 pin is configured as an output, a write to the port can cause a capture condition. |
|---|---|

**FIGURE 10-1: CAPTURE MODE OPERATION BLOCK DIAGRAM**




### 10.1.2 TIMER1 MODE RESOURCE

Timer1 must be running in Timer mode or Synchronized Counter mode for the CCP1 module to use the capture feature. In Asynchronous Counter mode, the capture operation may not work.

See **Section 7.0 "Timer1 Module with Gate Control"** for more information on configuring Timer1.

### 10.1.3 SOFTWARE INTERRUPT MODE

When the Capture mode is changed, a false capture interrupt may be generated. The user should keep the CCP1IE interrupt enable bit of the PIE2 register clear to avoid false interrupts. Additionally, the user should clear the CCP1IF interrupt flag bit of the PIR2 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 CCP1 pin, Timer1 must be clocked from the instruction clock (FOSC/4) or from an external clock source. |
|---|---|

### 10.1.4 CCP1 PRESCALER

There are four prescaler settings specified by the CCP1M<3:0> bits of the CCP1CON register. Whenever the CCP1 module is turned off, or the CCP1 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 CCP1CON register before changing the prescaler. Example 10-1 demonstrates the code to perform this function.

**EXAMPLE 10-1: CHANGING BETWEEN CAPTURE PRESCALERS**

```
BANKSEL CCP1CON     ;Set Bank bits to point
                    ;to CCP1CON
CLRF    CCP1CON     ;Turn CCP1 module off
MOVLW   NEW_CAPT_PS;Load the W reg with
                    ;the new prescaler
                    ;move value and CCP1 ON
MOVWF   CCP1CON     ;Load CCP1CON with this
                    ;value
```

### 10.1.5 CAPTURE DURING SLEEP

Capture mode depends upon the Timer1 module for proper operation. If the Timer1 clock input source is a clock that is not disabled during Sleep, Timer1 will continue to operate and Capture mode will operate during Sleep to wake the device. The T1CKI is an example of a clock source that will operate during Sleep.

When the input source to Timer1 is disabled during Sleep, such as the HFINTOSC, Timer1 will not increment during Sleep. When the device wakes from Sleep, Timer1 will continue from its previous state.

**TABLE 10-1: SUMMARY OF REGISTERS ASSOCIATED WITH CAPTURE**

| 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> | | | | 80 |
| CCPR1L | CCPR1L<7:0> | | | | | | | | 74 |
| CCPR1H | CCPR1H<7:0> | | | | | | | | 74 |
| INTCON | GIE | PEIE | T0IE | INTE | IOCIE | T0IF | INTF | IOCIF | 17 |
| PIE1 | TMR1GIE | ADIE | — | — | HLTMR2IE | HLTMR1IE | TMR2IE | TMR1IE | 18 |
| PIE2 | — | — | C2IE | C1IE | — | COG1IE | — | CCP1IE | 19 |
| PIR1 | TMR1GIF | ADIF | — | — | HLTMR2IF | HLTMR1IF | TMR2IF | TMR1IF | 20 |
| PIR2 | — | — | C2IF | C1IF | — | COG1IF | — | CCP1IF | 21 |
| T1CON | TMR1CS<1:0> | | T1CKPS<1:0> | | T1OSCEN | $\overline{\text{T1SYNC}}$ | — | TMR1ON | 65 |
| T1GCON | TMR1GE | T1GPOL | T1GTM | T1GSPM | T1GGO/$\overline{\text{DONE}}$ | T1GVAL | T1GSS<1:0> | | 66 |
| TMR1H | TMR1H<7:0> | | | | | | | | 57* |
| TMR1L | TMR1L<7:0> | | | | | | | | 57* |
| TRISA | — | — | TRISA5 | TRISA4 | TRISA3(1) | TRISA2 | TRISA1 | TRISA0 | 43 |

**Legend:** — = Unimplemented location, read as '0'. Shaded cells are not used by Capture mode.
* Page provides register information.

**Note 1:** TRISA3 always reads '1'.

## 10.2 Compare Mode

Compare mode makes use of the 16-bit Timer1 resource. The 16-bit value of the CCPR1H:CCPR1L 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:

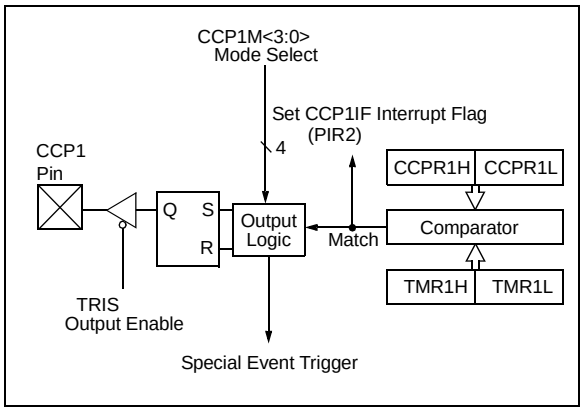
- Toggle the CCP1 output
- Set the CCP1 output
- Clear the CCP1 output
- Generate a Special Event Trigger
- Generate a Software Interrupt

The action on the pin is based on the value of the CCP1M<3:0> control bits of the CCP1CON register. At the same time, the interrupt flag CCP1IF bit is set.

All Compare modes can generate an interrupt.

Figure 10-2 shows a simplified diagram of the Compare operation.

**FIGURE 10-2: COMPARE MODE OPERATION BLOCK DIAGRAM**




### 10.2.1 CCP1 PIN CONFIGURATION

The user must configure the CCP1 pin as an output by clearing the associated TRIS bit.

| | |
|---|---|
| **Note:** | Clearing the CCP1CON register will force the CCP1 compare output latch to the default low level. This is not the PORT I/O data latch. |

### 10.2.2 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 7.0 "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 CCP1 pin, TImer1 must be clocked from the instruction clock (FOSC/4) or from an external clock source. |

### 10.2.3 SOFTWARE INTERRUPT MODE

When Generate Software Interrupt mode is chosen (CCP1M<3:0> = 1010), the CCP1 module does not assert control of the CCP1 pin (see the CCP1CON register).

### 10.2.4 SPECIAL EVENT TRIGGER

When Special Event Trigger mode is chosen (CCP1M<3:0> = 1011), the CCP1 module does the following:

- Resets Timer1
- Starts an ADC conversion if ADC is enabled

The CCP1 module does not assert control of the CCP1 pin in this mode.

The Special Event Trigger output of the CCP1 occurs immediately upon a match between the TMR1H, TMR1L register pair and the CCPR1H, CCPR1L register pair. The TMR1H, TMR1L register pair is not reset until the next rising edge of the Timer1 clock. The Special Event Trigger output starts an A/D conversion (if the A/D module is enabled). This allows the CCPR1H, CCPR1L register pair to effectively provide a 16-bit programmable period register for Timer1.

**TABLE 10-2: SPECIAL EVENT TRIGGER**

| Device | CCP1 |
|---|---|
| PIC16F753<br>PIC16HV753 | CCP1 |

Refer to **Section 12.0 "Analog-to-Digital Converter (ADC) Module"** for more information.

| | |
|---|---|
| **Note 1:** | The Special Event 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 CCPR1H and CCPR1L register pair, between the clock edge that generates the Special Event Trigger and the clock edge that generates the Timer1 Reset, will preclude the Reset from occurring. |

### 10.2.5 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.

**TABLE 10-3: SUMMARY OF REGISTERS ASSOCIATED WITH COMPARE**

| 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> | | | | 80 |
| CCPR1L | CCPR1L<7:0> | | | | | | | | 74 |
| CCPR1H | CCPR1H<7:0> | | | | | | | | 74 |
| INTCON | GIE | PEIE | T0IE | INTE | IOCIE | T0IF | INTF | IOCIF | 17 |
| PIE1 | TMR1-GIE | ADIE | — | — | HLTMR2IE | HLTMR1IE | TMR2IE | TMR1IE | 18 |
| PIE2 | — | — | C2IE | C1IE | — | COG1IE | — | CCP1IE | 19 |
| PIR1 | TMR1-GIF | ADIF | — | — | HLTMR2IF | HLTMR1IF | TMR2IF | TMR1IF | 20 |
| PIR2 | — | — | C2IF | C1IF | — | COG1IF | — | CCP1IF | 21 |
| T1CON | TMR1CS<1:0> | | T1CKPS<1:0> | | T1OSCEN | $\overline{\text{T1SYNC}}$ | — | TMR1ON | 65 |
| T1GCON | TMR1GE | T1GPOL | T1GTM | T1GSPM | T1GGO/$\overline{\text{DONE}}$ | T1GVAL | T1GSS<1:0> | | 66 |
| TMR1H | TMR1H<7:0> | | | | | | | | 57* |
| TMR1L | TMR1L<7:0> | | | | | | | | 57* |
| TRISA | — | — | TRISA5 | TRISA4 | TRISA3(1) | TRISA2 | TRISA1 | TRISA0 | 43 |

**Legend:** — = Unimplemented location, read as '0'. Shaded cells are not used by Compare mode.

* Page provides register information.

**Note 1:** TRISA3 always reads '1'.

## 10.3 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 10-3 shows a typical waveform of the PWM signal.

### 10.3.1 STANDARD PWM OPERATION

The standard PWM mode generates a Pulse-Width Modulation (PWM) signal on the CCP1 pin with up to 10 bits of resolution. The period, duty cycle, and resolution are controlled by the following registers:

- PR2 registers
- T2CON registers
- CCPR1L registers
- CCP1CON registers

Figure 10-4 shows a simplified block diagram of PWM operation.

| | |
|---|---|
| **Note 1:** | The corresponding TRIS bit must be cleared to enable the PWM output on the CCP1 pin. |
| **2:** | Clearing the CCP1CON register will relinquish control of the CCP1 pin. |

**FIGURE 10-3: CCP1 PWM OUTPUT SIGNAL**

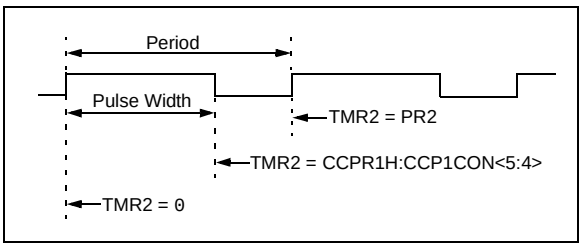
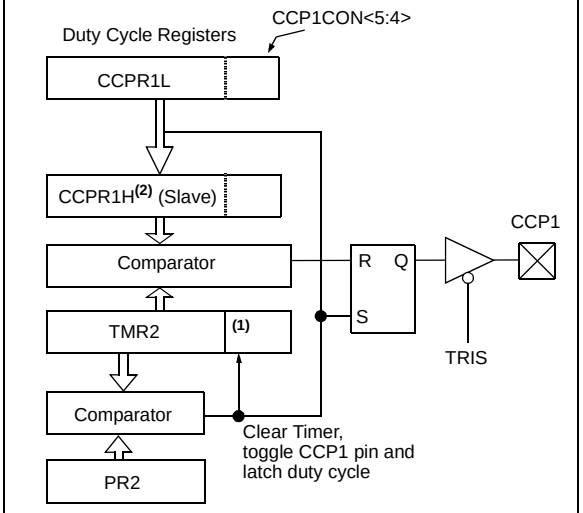



**FIGURE 10-4: SIMPLIFIED PWM BLOCK DIAGRAM**




**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.

### 10.3.2 SETUP FOR PWM OPERATION

The following steps should be taken when configuring the CCP1 module for standard PWM operation:

1. Disable the CCP1 pin output driver by setting the associated TRIS bit.
2. Load the PR2 register with the PWM period value.
3. Configure the CCP1 module for the PWM mode by loading the CCP1CON register with the appropriate values.
4. Load the CCPR1L register and the DC1B<1:0> bits of the CCP1CON register, with the PWM duty cycle value.
5. Configure and start Timer2:
   - Clear the TMR2IF interrupt flag bit of the PIR1 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 CCP1 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. |
|---|---|

### 10.3.3 PWM PERIOD

The PWM period is specified by the PR2 register of Timer2. The PWM period can be calculated using the formula of Equation 10-1.

**EQUATION 10-1: 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 CCP1 pin is set. (Exception: If the PWM duty cycle = 0%, the pin will not be set.)
- The PWM duty cycle is latched from CCPR1L into CCPR1H.

| **Note:** | The Timer postscaler (see **Section 8.1 "Timer2 Operation"**) is not used in the determination of the PWM frequency. |
|---|---|

### 10.3.4 PWM DUTY CYCLE

The PWM duty cycle is specified by writing a 10-bit value to multiple registers: CCPR1L register and DC1B<1:0> bits of the CCP1CON register. The CCPR1L contains the eight MSbs and the DC1B<1:0> bits of the CCP1CON register contain the two LSbs. CCPR1L and DC1B<1:0> bits of the CCP1CON register can be written to at any time. The duty cycle value is not latched into CCPR1H until after the period completes (i.e., a match between PR2 and TMR2 registers occurs). While using the PWM, the CCPR1H register is read-only.

Equation 10-2 is used to calculate the PWM pulse width.

Equation 10-3 is used to calculate the PWM duty cycle ratio.

**EQUATION 10-2: PULSE WIDTH**

$$Pulse\ Width = (CCPR1L{:}CCP1CON\langle 5{:}4\rangle) \bullet T_{OSC} \bullet (TMR2\ Prescale\ Value)$$

**EQUATION 10-3: DUTY CYCLE RATIO**

$$Duty\ Cycle\ Ratio = \frac{(CCPRxL{:}CCPxCON\langle 5{:}4\rangle)}{4(PRx + 1)}$$

The CCPR1H 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 CCPR1H and 2-bit latch, then the CCP1 pin is cleared (see Figure 10-4).

## 10.4 Register Definitions: CCP Control

### REGISTER 10-1: CCP1CON: CCP1 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 |
|---|---|---|---|---|---|---|---|
| — | — | DC1B<1:0> | | CCP1M<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 **DC1B<1:0>:** PWM Duty Cycle Least Significant bits

Capture mode:
Unused

Compare mode:
Unused

PWM mode:
These bits are the two LSbs of the PWM duty cycle. The eight MSbs are found in CCPR1L.

bit 3-0 **CCP1M<3:0>:** CCP1 Mode Select bits

0000 = Capture/Compare/PWM off (resets CCP1 module)
0001 = Reserved
0010 = Compare mode: toggle output on match
0011 = Reserved

0100 = Capture mode: every falling edge
0101 = Capture mode: every rising edge
0110 = Capture mode: every 4th rising edge
0111 = Capture mode: every 16th rising edge

1000 = Compare mode: initialize CCP1 pin low; set output on compare match (set CCP1IF)
1001 = Compare mode: initialize CCP1 pin high; clear output on compare match (set CCP1IF)
1010 = Compare mode: generate software interrupt only; CCP1 pin reverts to I/O state
1011 = Compare mode: Special Event Trigger (CCP1 resets Timer, sets CCP1IF bit, and starts A/D conversion if A/D module is enabled)
11xx = PWM mode

## 11.0 COMPLEMENTARY OUTPUT GENERATOR (COG) MODULE

The primary purpose of the Complementary Output Generator (COG) is to convert a single output PWM signal into a two output complementary PWM signal. The COG can also convert two separate input events into a single or complementary PWM output.

The COG PWM frequency and duty cycle are determined by a rising event input and a falling event input. The rising event and falling event may be the same source. Sources may be synchronous or asynchronous to the COG_clock.

The rate at which the rising event occurs determines the PWM frequency. The time from the rising event input to the falling event input determines the duty cycle.

A selectable clock input is used to generate the phase delay, blanking and dead-band times.

A simplified block diagram of the COG is shown in Figure 11-1.

The COG module has the following features:

- Two modes of operation:
  - Synchronous PWM
  - Push-pull
- Selectable clock source
- Independently selectable rising event sources
- Independently selectable falling event sources
- Independently selectable edge or level event sensitivity
- Independent output enables
- Independent output polarity selection
- Phase delay with independent rising and falling delay times
- Dead-band control with:
  - Independent rising and falling event dead-band times
  - Synchronous and asynchronous timing
- Blanking control with independent rising and falling event blanking times
- Auto-shutdown control with:
  - Independently selectable shutdown sources
  - Auto-restart enable
  - Auto-shutdown pin override control (high, low, off, and High-Z)

## 11.1 Fundamental Operation

### 11.1.1 SYNCHRONOUS PWM MODE

In synchronous PWM mode, the COG generates a two output complementary PWM waveform from rising and falling event sources. In the simplest configuration, the rising and falling event sources have the same signal, which is a PWM signal with the desired period and duty cycle. The COG converts this single PWM input into a dual complementary PWM output. The frequency and duty cycle of the dual PWM output match those of the single input PWM signal. The off-to-on transition of each output can be delayed from the on-to-off transition of the other output, thereby creating a time immediately after the PWM transition where neither output is driven. This is referred to as dead time and is covered in **Section 11.5 “Dead-Band Control”**.

A typical operating waveform, with dead band, generated from a single CCP1 input is shown in Figure 11-4.

### 11.1.2 PUSH-PULL MODE

In Push-Pull mode, the COG generates a single PWM output that alternates every PWM period, between the two COG output pins. The output drive activates with the rising input event and terminates with the falling event input. Each rising event starts a new period and causes the output to switch to the COG pin not used in the previous period.

A typical push-pull waveform generated from a single CCP1 input is shown in Figure 11-6.

Push-Pull mode is selected by setting the GxMD bit of the COGxCON0 register.

### 11.1.3 ALL MODES

In addition to generating a complementary output from a single PWM input, the COG can also generate PWM waveforms from a periodic rising event and a separate falling event. In this case, the falling event is usually derived from analog feedback within the external PWM driver circuit. In this configuration, high-power switching transients may trigger a false falling event that needs to be blanked out. The COG can be configured to blank falling (and rising) event inputs for a period of time immediately following the rising (and falling) event drive output. This is referred to as input blanking and is described in **Section 11.6 “Blanking Control”**.

It may be necessary to guard against the possibility of circuit faults. In this case, the active drive must be terminated before the Fault condition causes damage. This is referred to as auto-shutdown and is described in **Section 11.8 “Auto-shutdown Control”**.

A feedback falling event arriving too late or not at all can be terminated with auto-shutdown or by enabling one of the Hardware Limit Timer (HLT) event inputs. See **Section 9.0 “Hardware Limit Timer (HLT) Module”** for more information about the HLT.

The COG can be configured to operate in phase delayed conjunction with another PWM. The active drive cycle is delayed from the rising event by a phase delay timer. Phase delay is covered in more detail in **Section 11.7 “Phase Delay”**. A typical operating waveform, with phase delay and dead band, generated from a single CCP1 input, is shown in Figure 11-5.

**FIGURE 11-1: SIMPLIFIED COG BLOCK DIAGRAM**

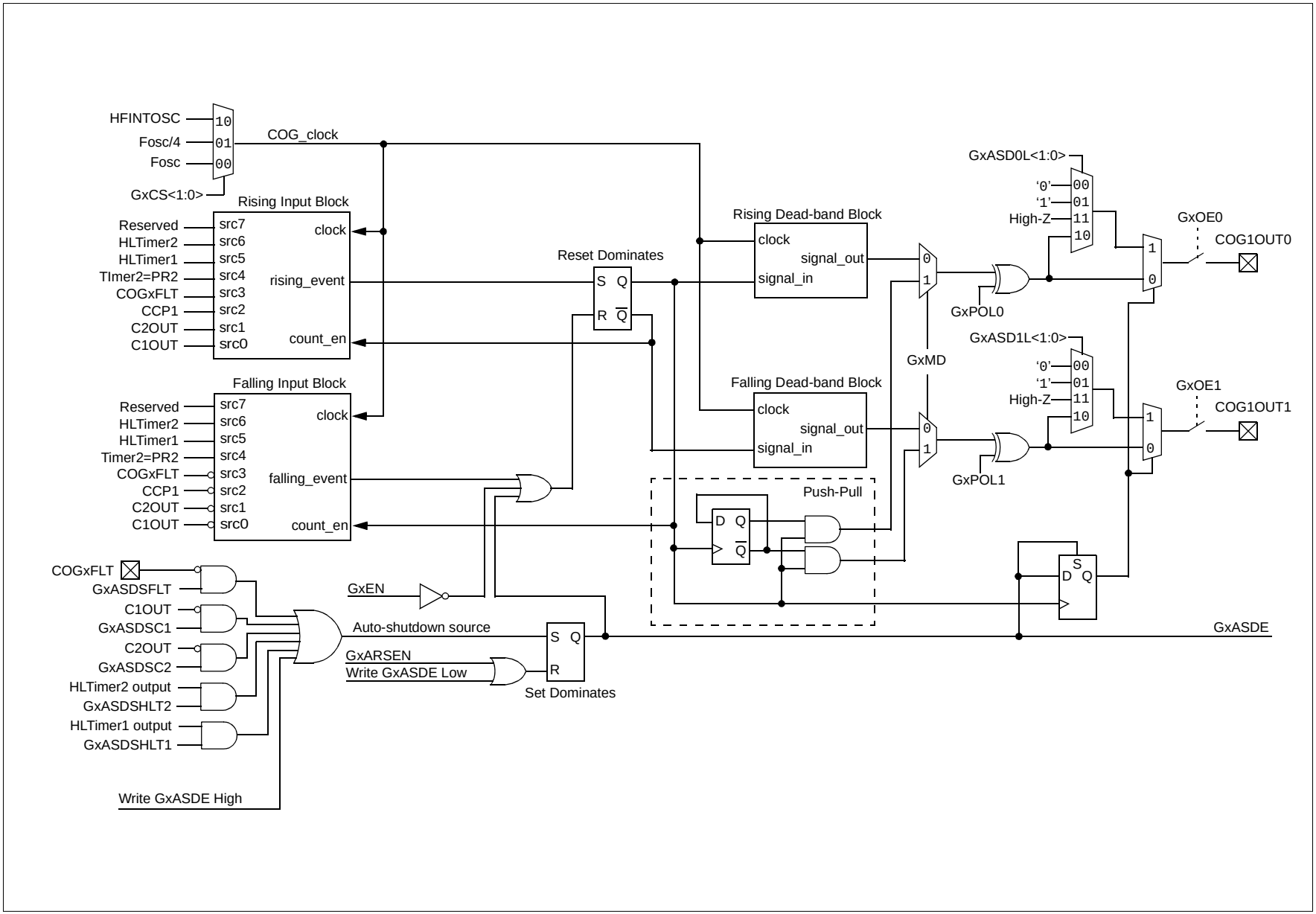


**FIGURE 11-2: COG (RISING/FALLING) INPUT BLOCK**

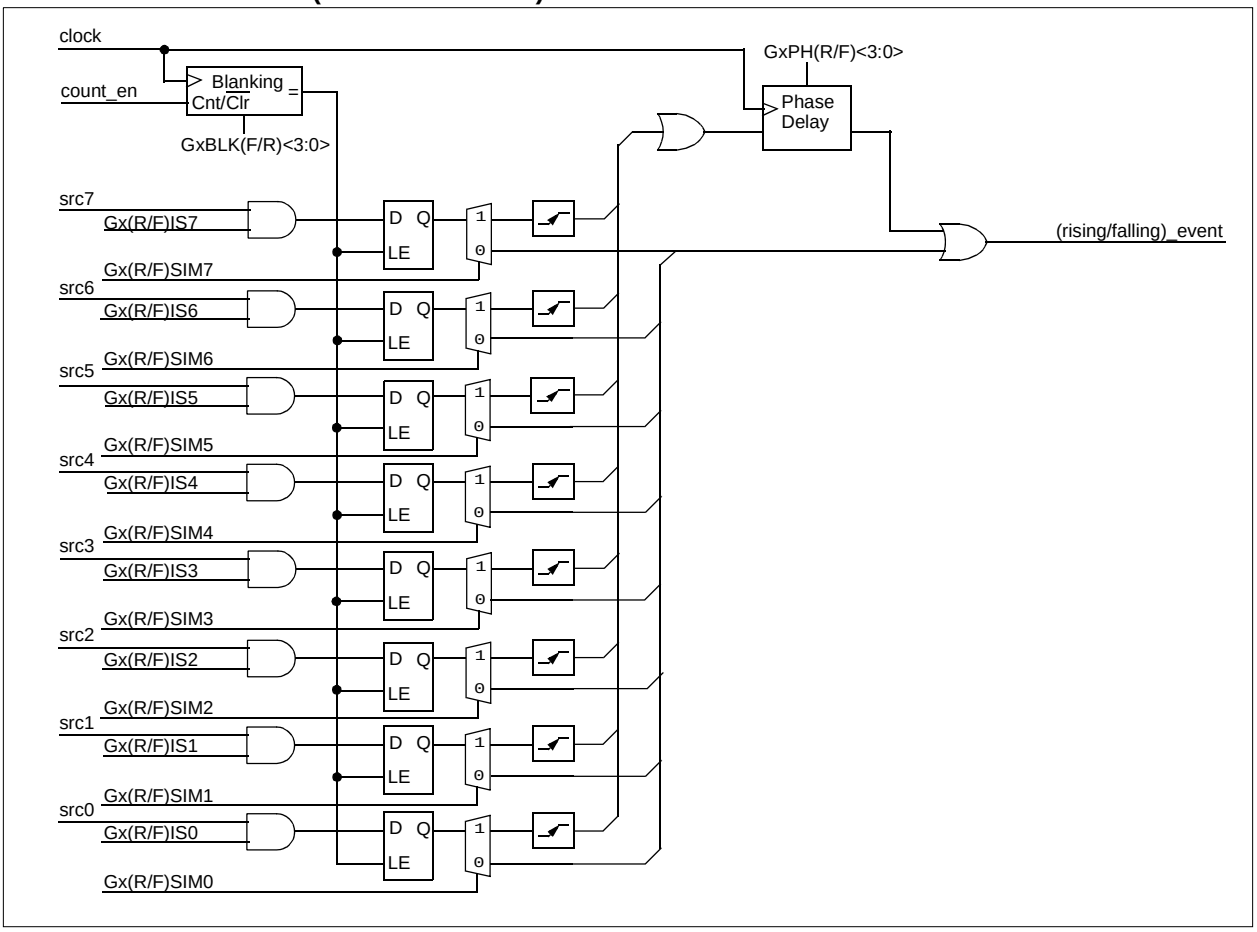



**FIGURE 11-3: COG (RISING/FALLING) DEAD-BAND BLOCK**

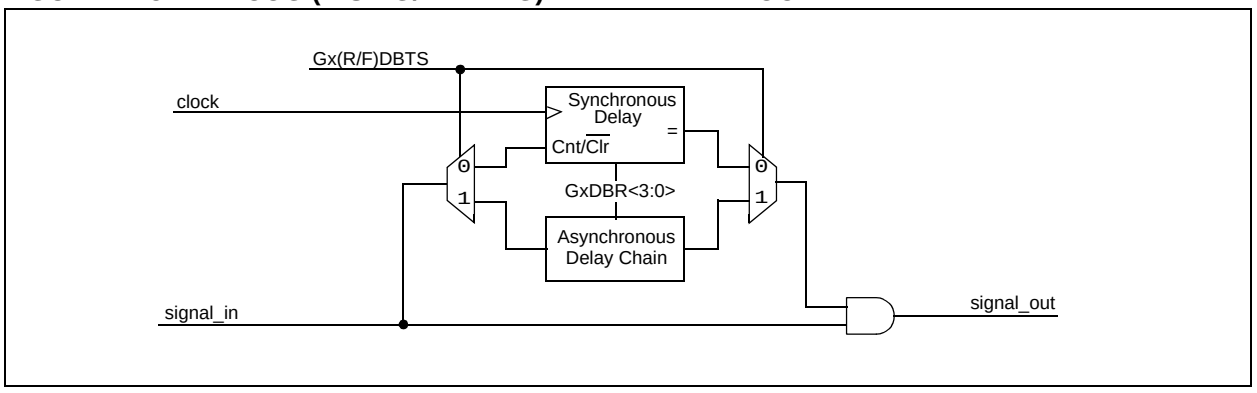


**FIGURE 11-4: TYPICAL COG OPERATION WITH CCP1**

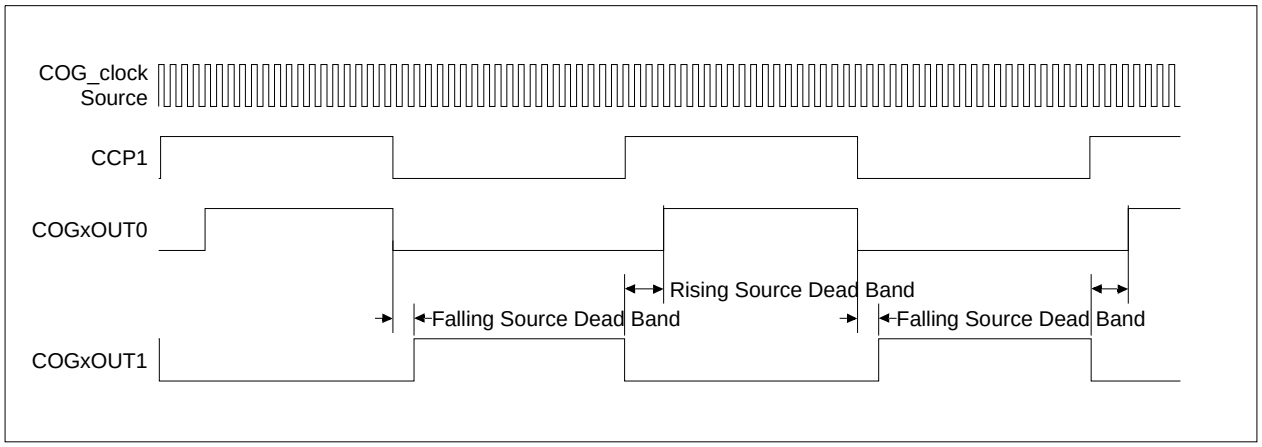



**FIGURE 11-5: COG OPERATION WITH CCP1 AND PHASE DELAY**

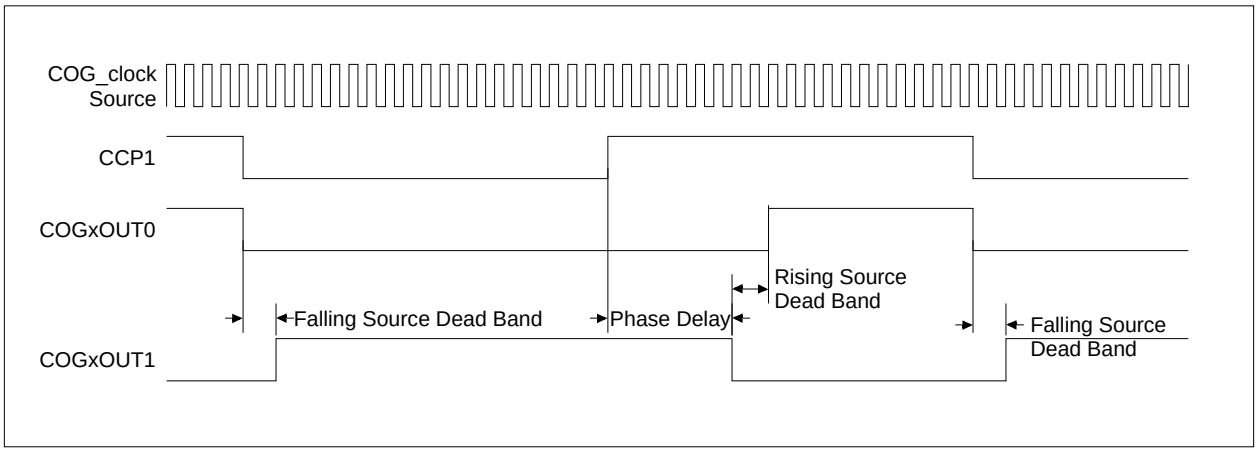
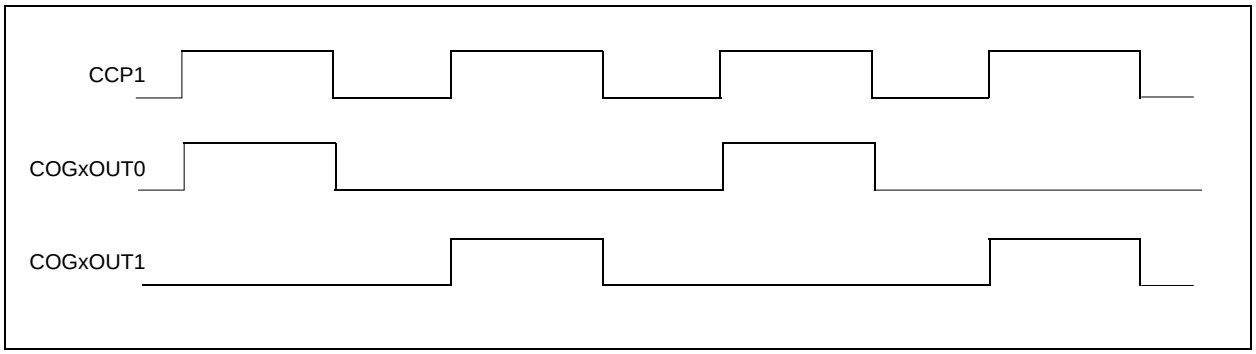



**FIGURE 11-6: COG OPERATION IN PUSH-PULL MODE WITH CCP1**



## 11.2 Clock Sources

The COG_clock is used as the reference clock to the various timers in the peripheral. Timers that use the COG_clock include:

- Rising and falling dead-band time
- Rising and falling blanking time
- Rising and falling event phase delay

Clock sources available for selection include:

- 8 MHz HFINTOSC (active during Sleep)
- Instruction clock (FOSC/4)
- System clock (FOSC)

The clock source is selected with the GxCS<1:0> bits of the COGxCON1 register (Register 11-2).

## 11.3 Selectable Event Sources

The COG uses any combination of independently selectable event sources to generate the complementary waveform. Sources fall into two categories:

- Rising event sources
- Falling event sources

The rising event sources are selected by setting bits in the COGxRIS register (Register 11-3). The falling event sources are selected by setting bits in the COGxFIS register (Register 11-5). All selected sources are 'OR'd together to generate the corresponding event signal. Refer to Figure 11-2.

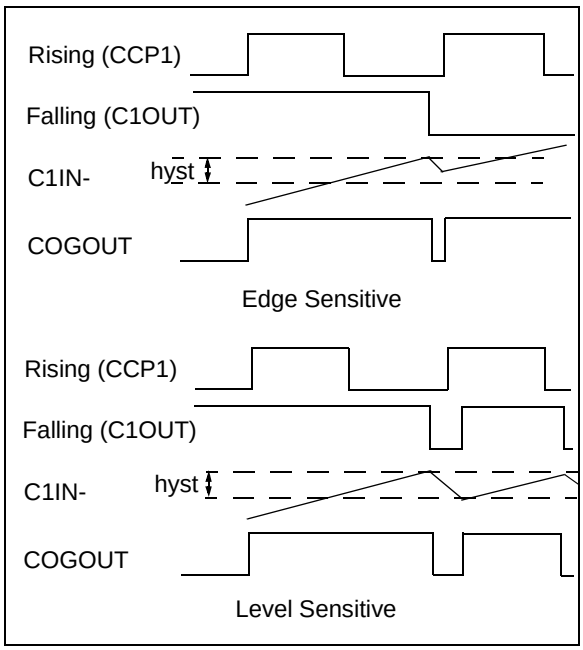
### 11.3.1 EDGE VS. LEVEL SENSING

Event input detection may be selected as level or edge-sensitive. The Detection mode is individually selectable for every source. Rising source detection modes are selected with the COGxRSIM register (Register 11-4). Falling source detection modes are selected with the COGxFSIM register (Register 11-6). A set bit enables edge detection for the corresponding event source. A cleared bit enables level detection.

In general, events that are driven from a periodic source should be edge-detected and events that are derived from voltage thresholds at the target circuit should be level-sensitive. Consider the following two examples:

1. The first example is an application in which the period is determined by a 50% duty cycle clock and the COG output duty cycle is determined by a voltage level fed back through a comparator. If the clock input is level sensitive, duty cycles less than 50% will exhibit erratic operation.
2. The second example is similar to the first, except that the duty cycle is close to 100%. The feedback comparator high-to-low transition trips the COG drive off, but almost immediately the period source turns the drive back on. If the off cycle is short enough, the comparator input may not reach the low side of the hysteresis band precluding an output change. The comparator output stays low and without a high-to-low transition to trigger the edge sense, the drive of the COG output will be stuck in a constant drive-on condition. See Figure 11-7.

**FIGURE 11-7: EDGE VS. LEVEL SENSE**




### 11.3.2 RISING EVENT

The rising event starts the PWM output active duty cycle period. The rising event is the low-to-high transition of the rising_event output. When the rising event phase delay and dead-band time values are zero, the COGxOUT0 output starts immediately. Otherwise, the COGxOUT0 output is delayed. The rising event source causes all the following actions:

- Start rising event phase delay counter (if enabled).
- Clear COGxOUT1 after phase delay.
- Start falling event input blanking (if enabled).
- Start dead-band delay (if enabled).
- Set COGxOUT0 output after dead-band delay expires.

### 11.3.3 FALLING EVENT

The falling event terminates the PWM output active duty cycle period. The falling event is the high-to-low transition of the falling_event output. When the falling event phase delay and dead-band time values are zero, the COGxOUT1 output starts immediately. Otherwise, the COGxOUT1 output is delayed. The falling event source causes all the following actions:

- Start falling event phase delay counter (if enabled).
- Clear COGxOUT0.
- Start rising event input blanking (if enabled).
- Start falling event dead-band delay (if enabled).
- Set COGxOUT1 output after dead-band delay expires.

## 11.4 Output Control

Upon disabling, or immediately after enabling the COG module, the complementary drive is configured with COGxOUT0 drive inactive and COGxOUT1 drive active.

### 11.4.1 OUTPUT ENABLES

Each COG output pin has an individual output enable control. Output enables are selected with the GxOE0 and GxOE1 bits of the COGxCON0 register (Register 11-1). When an output enable control is cleared, the module asserts no control over the pin. When an output enable is set, the override value or PWM waveform is applied to the pin per the port priority selection.

The device pin output enable control bits are independent of the GxEN bit of the COGxCON0 register, which enables the COG. When GxEN is cleared, and shutdown is not active, the Reset state PWM levels are present on the COG output pins. The PWM levels are affected by the polarity controls. If shutdown is active when GxEN is cleared, the shutdown override levels will be present on the COG output pins. Note that setting the GxASE bit while the GxEN bit is cleared will activate shutdown which can only be cleared by either a rising event while the GxEN bit is set, or a device Reset.

### 11.4.2 POLARITY CONTROL

The polarity of each COG output can be selected independently. When the output polarity bit is set, the corresponding output is active-low. Clearing the output polarity bit configures the corresponding output as active-high. However, polarity does not affect the shutdown override levels.

Output polarity is selected with the GxPOL0 and GxPOL1 bits of the COGxCON0 register (Register 11-1).

## 11.5 Dead-Band Control

The dead-band control provides for non-overlapping PWM output signals to prevent shoot-through current in the external power switches.

The COG contains two dead-band timers. One dead-band timer is used for rising event dead-band control. The other is used for falling event dead-band control. Timer modes are selectable as either:

- Asynchronous delay chain
- Synchronous counter

The dead-band Timer mode is selected for the COGxOUT0 and COGxOUT1 dead-band times with the respective GxRDBTS and GxFDBTS bits of the COGxCON1 register (Register 11-2).

### 11.5.1 ASYNCHRONOUS DELAY CHAIN DEAD-BAND DELAY

Asynchronous dead-band delay is determined by the time it takes the input to propagate through a series of delay elements. Each delay element is a nominal five nanoseconds.

Set the COGxDBR register (Register 11-9) value to the desired number of delay elements in the COGxOUT0 dead band. Set the COGxDBF register (Register 11-10) value to the desired number of delay elements in the COGxOUT1 dead band. When the value is zero, dead-band delay is disabled.

### 11.5.2 SYNCHRONOUS COUNTER DEAD-BAND DELAY

Synchronous counter dead band is timed by counting COG_clock periods from zero up to the value in the dead-band count register. Use Equation 11-1 to calculate dead-band times.

Set the COGxDBR count register value to obtain the desired dead-band time of the COGxOUT0 output. Set the COGxDBF count register value to obtain the desired dead-band time of the COGxOUT1 output. When the value is zero, dead-band delay is disabled.

### 11.5.3 SYNCHRONOUS COUNTER DEAD-BAND TIME UNCERTAINTY

When the rising and falling events that trigger the dead-band counters come from asynchronous inputs, it creates uncertainty in the synchronous counter dead-band time. The maximum uncertainty is equal to one COG_clock period. Refer to Equation 11-1 for more detail.

When event input sources are asynchronous with no phase delay, use the asynchronous delay chain dead-band mode to avoid the dead-band time uncertainty.

#### 11.5.4 RISING EVENT DEAD BAND

Rising event dead band adds a delay between the COGxOUT1 signal deactivation and the COGxOUT0 signal activation. The rising event dead-band time starts when the rising_event output goes true.

See Section 11.5.1, Asynchronous Delay Chain Dead-band Delay and Section 11.5.2, Synchronous Counter Dead-band Delay for more information on setting the rising edge dead-band time.

#### 11.5.5 FALLING EVENT DEAD BAND

Falling event dead band adds a delay between the COGxOUT1 signal deactivation and the COGxOUT0 signal activation. The falling event dead-band time starts when the falling_event output goes true.

See Section 11.5.1, Asynchronous Delay Chain Dead-band Delay and Section 11.5.2, Synchronous Counter Dead-band Delay for more information on setting the rising edge dead-band time.

#### 11.5.6 DEAD-BAND OVERLAP

There are two cases of dead-band overlap:

- Rising-to-falling
- Falling-to-rising

##### 11.5.6.1 Rising-to-Falling Overlap

In this case, the falling event occurs while the rising event dead-band counter is still counting. When this happens, the COGxOUT0 drive is suppressed and the dead band extends by the falling event dead-band time. At the termination of the extended dead-band time, the COGxOUT1 drive goes true.

##### 11.5.6.2 Falling-to-Rising Overlap

In this case, the rising event occurs while the falling event dead-band counter is still counting. When this happens, the COGxOUT1 drive is suppressed and the dead band extends by the rising event dead-band time. At the termination of the extended dead-band time, the COGxOUT0 drive goes true.

### 11.6 Blanking Control

Input blanking is a function, whereby the event inputs can be masked or blanked for a short period of time. This is to prevent electrical transients caused by the turn-on/off of power components from generating a false input event.

The COG contains two blanking counters: one triggered by the rising event and the other triggered by the falling event. The counters are cross-coupled with the events they are blanking. The falling event blanking counter is used to blank rising input events and the rising event blanking counter is used to blank falling input events. Once started, blanking extends for the time specified by the corresponding blanking counter.

Blanking is timed by counting COG_clock periods from zero up to the value in the blanking count register. Use Equation 11-1 to calculate blanking times.

#### 11.6.1 FALLING EVENT BLANKING OF RISING EVENT INPUTS

The falling event blanking counter inhibits rising event inputs from triggering a rising event. The falling event blanking time starts when the rising event output drive goes false.

The falling event blanking time is set by the value contained in the COGxBKF register (Register 11-12). Blanking times are calculated using the formula shown in Equation 11-1.

When the COGxBKF value is zero, the falling event blanking is disabled and the blanking counter output is true, thereby allowing the event signal to pass straight through to the event trigger circuit.

#### 11.6.2 RISING EVENT BLANKING OF FALLING EVENT INPUTS

The rising event blanking counter inhibits falling event inputs from triggering a falling event. The rising event blanking time starts when the falling event output drive goes false.

The rising event blanking time is set by the value contained in the COGxBKR register (Register 11-11).

When the COGxBKR value is zero, the rising event blanking is disabled and the blanking counter output is true, thereby allowing the event signal to pass straight through to the event trigger circuit.

#### 11.6.3 BLANKING TIME UNCERTAINTY

When the rising and falling sources that trigger the blanking counters are asynchronous to the COG_clock, it creates uncertainty in the blanking time. The maximum uncertainty is equal to one COG_clock period. Refer to Equation 11-1 and Example 11-2 for more detail.

### 11.7 Phase Delay

It is possible to delay the assertion of either or both the rising event and falling event. This is accomplished by placing a non-zero value in COGxPHR or COGxPHF phase delay count register, respectively (Register 11-13 and Register 11-14). Refer to Figure 11-5 for COG operation with CCP1 and phase delay. The delay from the input rising event signal switching to the actual assertion of the events is calculated the same as the dead-band and blanking delays. Please see Equation 11-1.

When the phase delay count value is zero, phase delay is disabled and the phase delay counter output is true, thereby allowing the event signal to pass straight through to complementary output driver flop.

### 11.7.1 CUMULATIVE UNCERTAINTY

It is not possible to create more than one COG_clock of uncertainty by successive stages. Consider that the phase delay stage comes after the blanking stage, the dead-band stage comes after either the blanking or phase delay stages, and the blanking stage comes after the dead-band stage. When the preceding stage is enabled, the output of that stage is necessarily synchronous with the COG_clock, which removes any possibility of uncertainty in the succeeding stage.

**EQUATION 11-1: PHASE, DEAD-BAND AND BLANKING TIME CALCULATION**

$$T_{\text{min}} = \frac{\text{Count}}{F_{COG\_clock}}$$

$$T_{\text{max}} = \frac{\text{Count} + 1}{F_{COG\_\text{clock}}}$$

$$T_{\text{uncertainty}} = T_{\text{max}} - T_{\text{min}}$$

*Also:*

$$T_{\text{uncertainty}} = \frac{1}{F_{COG\_\text{clock}}}$$

*Where:*

| T | Count |
|---|---|
| Rising Phase Delay | COGxPHR |
| Falling Phase Delay | COGxPHF |
| Rising Dead Band | COGxDBR |
| Falling Dead Band | COGxDBF |
| Rising Event Blanking | COGxBKR |
| Falling Event Blanking | COGxBKF |

**EQUATION 11-2: TIMER UNCERTAINTY**

*Given:*

$$Count = Ah = 10d$$

$$F_{COG\_\text{Clock}} = 8\text{MHz}$$

*Therefore:*

$$T_{\text{uncertainty}} = \frac{1}{F_{COG\_\text{clock}}}$$

$$= \frac{1}{8MHz} = 125ns$$

---

*Proof:*

$$T_{\text{min}} = \frac{Count}{F_{COG\_\text{clock}}}$$

$$= 125ns \bullet 10d = 1.25\mu s$$

$$T_{\text{max}} = \frac{Count + 1}{F_{COG\_\text{clock}}}$$

$$= 125ns \bullet (10d + 1)$$

$$= 1.375\mu s$$

*Therefore:*

$$T_{\text{uncertainty}} = T_{\text{max}} - T_{\text{min}}$$

$$= 1.375\mu s - 1.25\mu s$$

$$= 125ns$$

## 11.8 Auto-shutdown Control

Auto-shutdown is a method to immediately override the COG output levels with specific overrides that allow for safe shutdown of the circuit.

The shutdown state can be either cleared automatically or held until cleared by software. In either case, the shutdown overrides remain in effect until the first rising event after the shutdown is cleared.

### 11.8.1 SHUTDOWN

The shutdown state can be entered by either of the following two mechanisms:

- Software generated
- External Input

#### 11.8.1.1 Software Generated Shutdown

Setting the GxASDE bit of the COGxASD0 register (Register 11-7) will force the COG into the shutdown state.

When auto-restart is disabled, the shutdown state will persist until the first rising event after the GxASDE bit is cleared by software.

When auto-restart is enabled, the GxASDE bit will clear automatically and resume operation on the first rising event after the shutdown input clears. See Figure 11-8 and **Section 11.8.3.2 "Auto-Restart"**.

#### 11.8.1.2 External Shutdown Source

External shutdown inputs provide the fastest way to safely suspend COG operation in the event of a Fault condition. When any of the selected shutdown inputs goes true, the output drive latches are reset and the COG outputs immediately go to the selected override levels without software delay.

Any combination of the input sources can be selected to cause a shutdown condition. Shutdown input sources include:

- HLTimer1 output
- HLTimer2 output
- C2OUT (low true)
- C1OUT (low true)
- COG1FLT pin (low true)

Shutdown inputs are selected independently with bits of the COGxASD1 register (Register 11-8).

| | |
|---|---|
| **Note:** | Shutdown inputs are level-sensitive, not edge-sensitive. The shutdown state cannot be cleared as long as the shutdown input level persists, except by disabling auto-shutdown. |

### 11.8.2 PIN OVERRIDE LEVELS

The levels driven to the output pins, while the shutdown is active, are controlled by the GxASD0L<1:0> and GxASD1L<1:0> bits of the COGxASD0 register (Register 11-7). GxASD0L<1:0> controls the GxOUT0 override level and GxASD1L<1:0> controls the GxOUT1 override level. There are four override options for each output:

- Forced low
- Forced high
- Tri-state
- PWM inactive state (same state as that caused by a falling event)

| | |
|---|---|
| **Note:** | The polarity control does not apply to the forced low and high override levels. |

### 11.8.3 AUTO-SHUTDOWN RESTART

After an auto-shutdown event has occurred, there are two ways to have the module resume operation:

- Software controlled
- Auto-restart

The restart method is selected with the GxARSEN bit of the COGxASD0 register. Waveforms of a software controlled automatic restart are shown in Figure 11-8.

#### 11.8.3.1 Software Controlled Restart

When the GxARSEN bit of the COGxASD0 register is cleared, software must clear the GxASDE bit to restart COG operation after an auto-shutdown event.

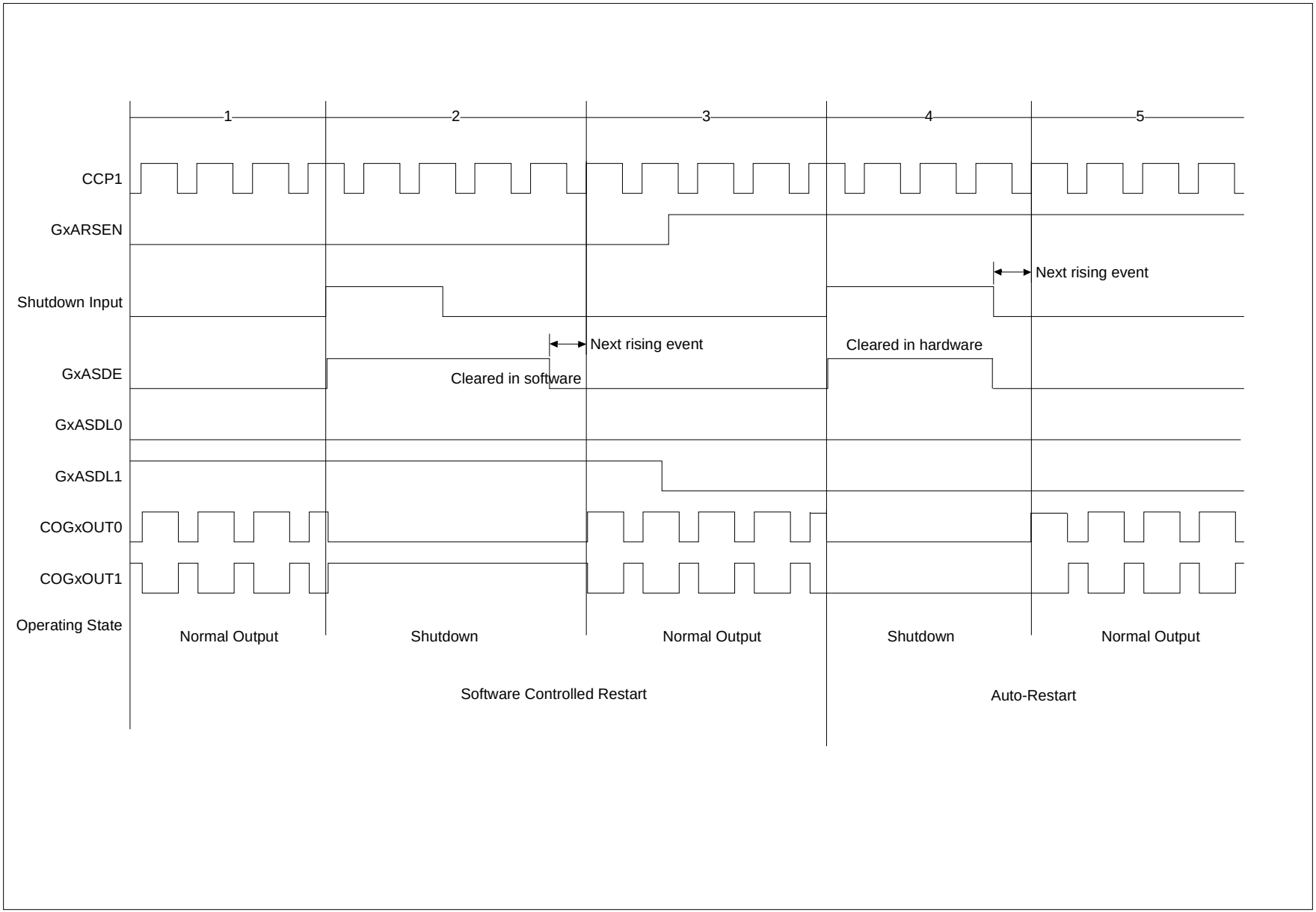
The COG will resume operation on the first rising event after the GxASDE bit is cleared. Clearing the shutdown state requires all selected shutdown inputs to be false, otherwise, the GxASDE bit will remain set.

#### 11.8.3.2 Auto-Restart

When the GxARSEN bit of the COGxASD0 register is set, the COG will restart from the auto-shutdown state automatically.

The GxASDE bit will clear automatically and the COG will resume operation on the first rising event after all selected shutdown inputs go false.

**FIGURE 11-8: AUTO-SHUTDOWN WAVEFORM – CCP1 AS RISING AND FALLING EVENT INPUT SOURCE**



## 11.9 Buffer Updates

Changes to the phase, dead-band, and blanking count registers need to occur simultaneously during COG operation to avoid unintended operation that may occur as a result of delays between each register write. This is accomplished with the GxLD bit of the COGxCON0 register and double buffering of the phase, blanking, and dead-band count registers.

Before the COG module is enabled, writing the count registers loads the count buffers without need of the GxLD bit. However, when the COG is enabled, the count buffers updates are suspended after writing the count registers until after the GxLD bit is set. When the GxLD bit is set, the phase, dead-band, and blanking register values are transferred to the corresponding buffers synchronous with COG operation. The GxLD bit is cleared by hardware when the transfer is complete.

## 11.10 Alternate Pin Selection

The COGxOUT0, COGxOUT1 and COGxFLT functions can be directed to alternate pins with control bits of the APFCON register. Refer to Register 5-1.

| | |
|---|---|
| **Note:** | The default COG outputs have high drive strength capability, whereas the alternate outputs do not. |

## 11.11 Operation During Sleep

The COG continues to operate in Sleep provided that the COG_clock, rising event, and falling event sources remain active.

The HFINTSOC remains active during Sleep when the COG is enabled and the HFINTOSC is selected as the COG_clock source.

## 11.12 Configuring the COG

The following steps illustrate how to properly configure the COG to ensure a synchronous start with the rising event input:

1. Configure the desired COGxFLT input, COGxOUT0 and COGxOUT1 pins with the corresponding bits in the APFCON register.
2. Clear all ANSELA register bits associated with pins that are used for COG functions.
3. Ensure that the TRIS control bits corresponding to COGxOUT0 and COGxOUT1 are set so that both are configured as inputs. These will be set as outputs later.
4. Clear the GxEN bit, if not already cleared.
5. Set desired dead-band times with the COGxDBR and COGxDBF registers.
6. Set desired blanking times with the COGxBKR and COGxBKF registers.
7. Set desired phase delay with the COGxPHR and COGxPHF registers.
8. Select the desired shutdown sources with the COGxASD1 register.
9. Set up the following controls in COGxASD0 auto-shutdown register:
   - Select both output overrides to the desired levels (this is necessary, even if not using auto-shutdown because start-up will be from a shutdown state).
   - Set the GxASDE bit and clear the GxARSEN bit.
10. Select the desired rising and falling event sources with the COGxRIS and COGxFIS registers.
11. Select the desired rising and falling event modes with the COGxRSIM and COGxFSIM registers.
12. Configure the following controls in the COGxCON1 register:
    - Select the desired clock source
    - Select the desired dead-band timing sources
13. Configure the following controls in the COGxCON0 register:
    - Select the desired output polarities.
    - Set the output enables of the outputs to be used.
14. Set the GxEN bit.
15. Clear TRIS control bits corresponding to COGxOUT0 and COGxOUT1 to be used, thereby configuring those pins as outputs.
16. If auto-restart is to be used, set the GxARSEN bit and the GxASDE will be cleared automatically. Otherwise, clear the GxASDE bit to start the COG.

## 11.13 Register Definitions: COG Control

**REGISTER 11-1: COGxCON0: COG CONTROL REGISTER 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 | U-0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| GxEN | GxOE1 | GxOE0 | GxPOL1 | GxPOL0 | GxLD | — | GxMD |
| 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 **GxEN:** COGx Enable bit
1 = Module is enabled
0 = Module is disabled

bit 6 **GxOE1:** COGxOUT1 Output Enable bit
1 = COGxOUT1 is available on associated I/O pin
0 = COGxOUT1 is not available on associated I/O pin

bit 5 **GxOE0:** COGxOUT0 Output Enable bit
1 = COGxOUT0 is available on associated I/O pin
0 = COGxOUT0 is not available on associated I/O pin

bit 4 **GxPOL1:** COGxOUT1 Output Polarity bit
1 = Output is inverted polarity
0 = Output is normal polarity

bit 3 **GxPOL0:** COGxOUT0 Output Polarity bit
1 = Output is inverted polarity
0 = Output is normal polarity

bit 2 **GxLD:** COGx Load Buffers bit
1 = Phase, blanking, and dead-band buffers to be loaded with register values on next input events
0 = Register to buffer transfer is complete

bit 1 **Unimplemented:** Read as ‘0’

bit 0 **GxMD:** COGx Mode bit
1 = COG outputs operate in Push-Pull mode
0 = COG outputs operate in Synchronous mode

**REGISTER 11-2: COGxCON1: COG CONTROL REGISTER 1**

| R/W-0/0 | R/W-0/0 | U-0 | U-0 | U-0 | U-0 | R/W-0/0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| GxRDBTS | GxFDBTS | — | — | — | — | GxCS<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 **GxRDBTS:** COGx Rising Event Dead-band Timing Source Select bit
1 = Delay chain and COGxDBR are used for dead-band timing generation
0 = COGx_clk and COGxDBR are used for dead-band timing generation

bit 6 **GxFDBTS:** COGx Falling Event Dead-band Timing Source Select bit
1 = Delay chain and COGxDF are used for dead-band timing generation
0 = COGx_clk and COGxDBF are used for dead-band timing generation

bit 5-2 **Unimplemented:** Read as ‘0’

bit 1-0 **GxCS<1:0>:** COGx Clock Source Select bits
11 = Reserved
10 = HFINTOSC (stays active during Sleep)
01 = FOSC/4
00 = FOSC

**REGISTER 11-3: COGxRIS: COG RISING EVENT INPUT SELECTION 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 |
|---|---|---|---|---|---|---|---|
| — | GxRIHLT2 | GxRIHLT1 | GxRIT2M | GxRIFLT | GxRICCP1 | GxRIC2 | GxRIC1 |
| 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 **Unimplemented:** Read as ‘0’

bit 6 **GxRIHLT2:** COGx Rising Event Input Source 6 Enable bit
1 = HLTimer2 output is enabled as a rising event input
0 = HLTimer2 has no effect on the rising event

bit 5 **GxRIHLT1:** COGx Rising Event Input Source 5 Enable bit
1 = HLTimer1 output is enabled as a rising event input
0 = HLTimer1 has no effect on the rising event

bit 4 **GxRIT2M:** COGx Rising Event Input Source 4 Enable bit
1 = Timer2 match with PR2 is enabled as a rising event input
0 = Timer2 match with PR2 has no effect on the rising event

bit 3 **GxRIFLT:** COGx Rising Event Input Source 3 Enable bit
1 = COGxFLT pin is enabled as a rising event input
0 = COGxFLT pin has no effect on the rising event

bit 2 **GxRICCP1:** COGx Rising Event Input Source 2 Enable bit
1 = CCP1 output is enabled as a rising event input
0 = CCP1 has no effect on the rising event

bit 1 **GxRIC2:** COGx Rising Event Input Source 1 Enable bit
1 = Comparator 2 output is enabled as a rising event input
0 = Comparator 2 output has no effect on the rising event

bit 0 **GxRIC1:** COGx Rising Event Input Source 0 Enable bit
1 = Comparator 1 output is enabled as a rising event input
0 = Comparator 1 output has no effect on the rising event

**REGISTER 11-4: COGxRSIM: COG RISING EVENT SOURCE INPUT MODE 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 |
|---|---|---|---|---|---|---|---|
| — | GxRMHLT2 | GxRMHLT1 | GxRMT2M | GxRMFLT | GxRMCCP1 | GxRMC2 | GxRMC1 |
| 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 **Unimplemented:** Read as '0'

bit 6 **GxRMHLT2:** COGx Rising Event Input Source 6 Mode bit(1)
GxRIHLT2 = 1:
1 = HLTimer2 low-to-high transition will cause a rising event after rising event phase delay
0 = HLTimer2 high level will cause an immediate rising event
GxRIHLT2 = 0:
HLTimer2 has no effect on rising event

bit 5 **GxRMHLT1:** COGx Rising Event Input Source 5 Mode bit(1)
GxRIHLT1 = 1:
1 = HLTimer1 low-to-high transition will cause a rising event after rising event phase delay
0 = HLTimer1 high level will cause an immediate rising event
GxRIHLT1 = 0:
HLTimer1 has no effect on rising event

bit 4 **GxRMT2M:** COGx Rising Event Input Source 4 Mode bit(1)
GxRIT2M = 1:
1 = Timer2 match with PR2 low-to-high transition will cause a rising event after rising event phase delay
0 = Timer2 match with PR2 high level will cause an immediate rising event
GxRIT2M = 0:
Timer2 match with PR2 has no effect on rising event

bit 3 **GxRMFLT:** COGx Rising Event Input Source 3 Mode bit
GxRIFLT = 1:
1 = COGxFLT pin low-to-high transition will cause a rising event after rising event phase delay
0 = COGxFLT pin high level will cause an immediate rising event
GxRIFLT = 0:
COGxFLT pin has no effect on rising event

bit 2 **GxRMCCP1:** COGx Rising Event Input Source 2 Mode bit
GxRICCP1 = 1:
1 = CCP1 low-to-high transition will cause a rising event after rising event phase delay
0 = CCP1 high level will cause an immediate rising event
GxRICCP1 = 0:
CCP1 has no effect on rising event

bit 1 **GxRMC2:** COGx Rising Event Input Source 1 Mode bit
GxRIC2 = 1:
1 = Comparator 2 low-to-high transition will cause a rising event after rising event phase delay
0 = Comparator 2 high level will cause an immediate rising event
GxRIC2 = 0:
Comparator 2 has no effect on rising event

bit 0 **GxRMC1:** COGx Rising Event Input Source 0 Mode bit
GxRIC1 = 1:
1 = Comparator 1 low-to-high transition will cause a rising event after rising event phase delay
0 = Comparator 1 high level will cause an immediate rising event
GxRIC1 = 0:
Comparator 1 has no effect on rising event

**Note 1:** These sources are pulses and therefore the only benefit of Edge mode over Level mode is that they can be delayed by rising event phase delay.

**REGISTER 11-5: COGxFIS: COG FALLING EVENT INPUT SELECTION 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 |
|---|---|---|---|---|---|---|---|
| — | GxFIHLT2 | GxFIHLT1 | GxFIT2M | GxFIFLT | GxFICCP1 | GxFIC2 | GxFIC1 |
| 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 **Unimplemented:** Read as ‘0’

bit 6 **GxFIHLT2:** COGx Falling Event Input Source 6 Enable bit
1 = HLTimer2 output is enabled as a falling event input
0 = HLTimer2 has no effect on the falling event

bit 5 **GxFIHLT1:** COGx Falling Event Input Source 5 Enable bit
1 = HLTimer1 output is enabled as a falling event input
0 = HLTimer1 has no effect on the falling event

bit 4 **GxFIT2M:** COGx Falling Event Input Source 4 Enable bit
1 = Timer2 match with PR2 is enabled as a falling event input
0 = Timer2 match with PR2 has no effect on the falling event

bit 3 **GxFIFLT:** COGx Falling Event Input Source 3 Enable bit
1 = COGxFLT pin is enabled as a falling event input
0 = COGxFLT pin has no effect on the falling event

bit 2 **GxFICCP1:** COGx Falling Event Input Source 2 Enable bit
1 = CCP1 output is enabled as a falling event input
0 = CCP1 has no effect on the falling event

bit 1 **GxFIC2:** COGx Falling Event Input Source 1 Enable bit
1 = Comparator 2 output is enabled as a falling event input
0 = Comparator 2 output has no effect on the falling event

bit 0 **GxFIC1:** COGx Falling Event Input Source 0 Enable bit
1 = Comparator 1 output is enabled as a falling event input
0 = Comparator 1 output has no effect on the falling event

**REGISTER 11-6: COGxFSIM: COG FALLING EVENT SOURCE INPUT MODE 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 |
|---|---|---|---|---|---|---|---|
| — | GxFMHLT2 | GxFMHLT1 | GxFMT2M | GxFMFLT | GxFMCCP1 | GxFMC2 | GxFMC1 |
| 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 **Unimplemented:** Read as '0'

bit 6 **GxFMHLT2:** COGx Falling Event Input Source 6 Mode bit(1)
GxFIHLT2 = 1:
1 = HLTimer2 low-to-high transition will cause a falling event after falling event phase delay
0 = HLTimer2 high level will cause an immediate falling event
GxFIHLT2 = 0:
HLTimer2 has no effect on falling event

bit 5 **GxFMHLT1:** COGx Falling Event Input Source 5 Mode bit(1)
GxFIHLT1 = 1:
1 = HLTimer1 low-to-high transition will cause a falling event after falling event phase delay
0 = HLTimer1 high level will cause an immediate falling event
GxFIHLT1 = 0:
HLTimer1 has no effect on falling event

bit 4 **GxFMT2M:** COGx Falling Event Input Source 4 Mode bit(1)
GxFIT2M = 1:
1 = Timer2 match with PR2 low-to-high transition will cause a falling event after falling event phase delay
0 = Timer2 match with PR2 high level will cause an immediate falling event
GxFIT2M = 0:
Timer2 match with PR2 has no effect on falling event

bit 3 **GxFMFLT:** COGx Falling Event Input Source 3 Mode bit
GxFIFLT = 1:
1 = COGxFLT pin low-to-high transition will cause a falling event after falling event phase delay
0 = COGxFLT pin high level will cause an immediate falling event
GxFIFLT = 0:
COGxFLT pin has no effect on falling event

bit 2 **GxFMCCP1:** COGx Falling Event Input Source 2 Mode bit
GxFICCP1 = 1:
1 = CCP1 low-to-high transition will cause a falling event after falling event phase delay
0 = CCP1 high level will cause an immediate falling event
GxFICCP1 = 0:
CCP1 has no effect on falling event

bit 1 **GxFMC2:** COGx Falling Event Input Source 1 Mode bit
GxFIC2 = 1:
1 = Comparator 2 low-to-high transition will cause a falling event after falling event phase delay
0 = Comparator 2 high level will cause an immediate falling event
GxFIC2 = 0:
Comparator 2 has no effect on falling event

bit 0 **GxFMC1:** COGx Falling Event Input Source 0 Mode bit
GxFIC1 = 1:
1 = Comparator 1 low-to-high transition will cause a falling event after falling event phase delay
0 = Comparator 1 high level will cause an immediate falling event
GxFIC1 = 0:
Comparator 1 has no effect on falling event

**Note 1:** These sources are pulses and therefore the only benefit of Edge mode over Level mode is that they can be delayed by falling event phase delay.

**REGISTER 11-7: COGxASD0: COG AUTO-SHUTDOWN CONTROL REGISTER 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 | U-0 | U-0 |
|---|---|---|---|---|---|---|---|
| GxASDE | GxARSEN | GxASD1L<1:0> | | GxASD0L<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 **GxASDE:** Auto-Shutdown Event Status bit
1 = COG is in the shutdown state
0 = COG is either not in the shutdown state or will exit the shutdown state on the next rising event

bit 6 **GxARSEN:** Auto-Restart Enable bit
1 = Auto-restart is enabled
0 = Auto-restart is disabled

bit 5-4 **GxASD1L<1:0>:** COGxOUT1 Auto-Shutdown Override Level Select bits
11 = COGxOUT1 is tri-stated when shutdown is active
10 = The inactive state of the pin, including polarity, is placed on COGxOUT1 when shutdown is active
01 = A logic ‘1’ is placed on COGxOUT1 when shutdown is active
00 = A logic ‘0’ is placed on COGxOUT1 when shutdown is active

bit 3-2 **GxASD0L<1:0>:** COGxOUT0 Auto-Shutdown Override Level Select bits
11 = COGxOUT0 is tri-stated when shutdown is active
10 = The inactive state of the pin, including polarity, is placed on COGxOUT0 when shutdown is active
01 = A logic ‘1’ is placed on COGxOUT0when shutdown is active
00 = A logic ‘0’ is placed on COGxOUT0when shutdown is active

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

**REGISTER 11-8: COGxASD1: COG AUTO-SHUTDOWN CONTROL REGISTER 1**

| U-0 | U-0 | U-0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| — | — | — | GxASDSHLT2 | GxASDSHLT1 | GxASDSC2 | GxASDSC1 | GxASDSFLT |
| 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 **GxASDSHLT2:** COGx Auto-Shutdown Source Enable bit 4
1 = COGx is shutdown when HLTMR2 equals HLTPR2
0 = HLTimer 2 has no effect on shutdown

bit 3 **GxASDSHLT1:** COGx Auto-Shutdown Source Enable bit 3
1 = COGx is shutdown when HLTMR1 equals HLTPR1
0 = HLTimer 1 has no effect on shutdown

bit 2 **GxASDSC2:** COGx Auto-Shutdown Source Enable bit 2
1 = COGx is shutdown when Comparator 2 output is low
0 = Comparator 2 output has no effect on shutdown

bit 1 **GxASDSC1:** COGx Auto-Shutdown Source Enable bit 1
1 = COGx is shutdown when Comparator 1 output is low
0 = Comparator 1 output has no effect on shutdown

bit 0 **GxASDSFLT:** COGx Auto-Shutdown Source Enable bit 0
1 = COGx is shutdown when COGxFLT pin is low
0 = COGxFLT pin has no effect on shutdown

**REGISTER 11-9: COGxDBR: COG RISING EVENT DEAD-BAND COUNT REGISTER**

| U-0 | U-0 | U-0 | U-0 | R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u |
|---|---|---|---|---|---|---|---|
| — | — | — | — | GxDBR<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 | q = Value depends on condition |

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

bit 3-0 **GxDBR<3:0>:** Rising Event Dead-band Count Value bits

GxRDBTS = 1:
= Number of delay chain element periods to delay primary output after rising event

GxRDBTS = 0:
= Number of COGx clock periods to delay primary output after rising event

**REGISTER 11-10: COGxDBF: COG FALLING EVENT DEAD-BAND COUNT REGISTER**

| U-0 | U-0 | U-0 | U-0 | R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u |
|---|---|---|---|---|---|---|---|
| — | — | — | — | GxDBF<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 | q = Value depends on condition |

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

bit 3-0 **GxDBF<3:0>:** Falling Event Dead-Band Count Value bits

GxFDBTS = 1:
= Number of delay chain element periods to delay complementary output after falling event input

GxFDBTS = 0:
= Number of COGx clock periods to delay complementary output after falling event input

**REGISTER 11-11: COGxBKR: COG RISING EVENT BLANKING COUNT REGISTER**

| U-0 | U-0 | U-0 | U-0 | R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u |
|---|---|---|---|---|---|---|---|
| — | — | — | — | GxBKR<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 | q = Value depends on condition |

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

bit 3-0 **GxBKR<3:0>:** Rising Event Blanking Count Value bits

= Number of COGx clock periods to inhibit falling event inputs

**REGISTER 11-12: COGxBKF: COG FALLING EVENT BLANKING COUNT REGISTER**

| U-0 | U-0 | U-0 | U-0 | R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u |
|---|---|---|---|---|---|---|---|
| — | — | — | — | GxBKF<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 | q = Value depends on condition |

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

bit 3-0 **GxBKF<3:0>:** Falling Event Blanking Count Value bits

= Number of COGx clock periods to inhibit rising event inputs

**REGISTER 11-13: COGxPHR: COG RISING EDGE PHASE DELAY COUNT REGISTER**

| U-0 | U-0 | U-0 | U-0 | R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u |
|---|---|---|---|---|---|---|---|
| — | — | — | — | GxPHR<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 | q = Value depends on condition |

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

bit 3-0 **GxPHR<3:0>:** Rising Edge Phase Delay Count Value bits
= Number of COGx clock periods to delay rising edge event

**REGISTER 11-14: COGxPHF: COG FALLING EDGE PHASE DELAY COUNT REGISTER**

| U-0 | U-0 | U-0 | U-0 | R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u |
|---|---|---|---|---|---|---|---|
| — | — | — | — | GxPHF<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 | q = Value depends on condition |

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

bit 3-0 **GxPHF<3:0>:** Falling Edge Phase Delay Count Value bits
= Number of COGx clock periods to delay falling edge event

**TABLE 11-1: SUMMARY OF REGISTERS ASSOCIATED WITH COG**

| 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 | 44 |
| APFCON | — | — | — | T1GSEL | — | — | — | — | 40 |
| COG1PHR | — | — | — | — | G1PHR<3:0> | | | | 102 |
| COG1PHF | — | — | — | — | G1PHF<3:0> | | | | 102 |
| COG1BKR | — | — | — | — | G1BKR<3:0> | | | | 101 |
| COG1BKF | — | — | — | — | G1BKF<3:0> | | | | 101 |
| COG1DBR | — | — | — | — | G1DBR<3:0> | | | | 100 |
| COG1DBF | — | — | — | — | G1DBF<3:0> | | | | 100 |
| COG1RIS | — | G1RIHLT2 | G1RIHLT1 | G1RIT2M | G1RIFLT | G1RICCP1 | G1RIC2 | G1RIC1 | 94 |
| COG1RSIM | — | G1RMHLT2 | G1RMHLT1 | G1RMT2M | G1RMFLT | G1RMCCP1 | G1RMC2 | G1RMC1 | 95 |
| COG1FIS | — | G1FIHLT2 | G1FIHLT1 | G1FIT2M | G1FIFLT | G1FICCP1 | G1FIC2 | G1FIC1 | 96 |
| COG1FSIM | — | G1FMHLT2 | G1FMHLT1 | G1FMT2M | G1FMFLT | G1FMCCP1 | G1FMC2 | G1FMC1 | 97 |
| COG1CON0 | G1EN | G1OE1 | G1OE0 | G1POL1 | G1POL0 | G1LD | — | G1MD | 92 |
| COG1CON1 | G1RDBTS | G1FDBTS | — | — | — | — | G1CS<1:0> | | 93 |
| COG1ASD0 | G1ASDE | G1ARSEN | G1ASD1L<1:0> | | G1ASD0L<1:0> | | — | — | 98 |
| COG1ASD1 | — | — | — | G1ASDSHLT2 | G1ASDSHLT1 | G1ASDSC2 | G1ASDSC1 | G1ASDSFLT | 99 |
| INTCON | GIE | PEIE | T0IE | INTE | IOCIE | T0IF | INTF | IOCIF | 17 |
| LATA | — | — | LATA5 | LATA4 | — | LATA2 | LATA1 | LATA0 | 43 |
| PIE2 | — | — | C2IE | C1IE | — | COG1IE | — | CCP1IE | 19 |
| PIR2 | — | — | C2IF | C1IF | — | COG1IF | — | CCP1IF | 21 |
| TRISA | — | — | TRISA5 | TRISA4 | TRISA3 | TRISA2 | TRISA1 | TRISA0 | 43 |

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

## 12.0 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 (ADRESL and ADRESH).

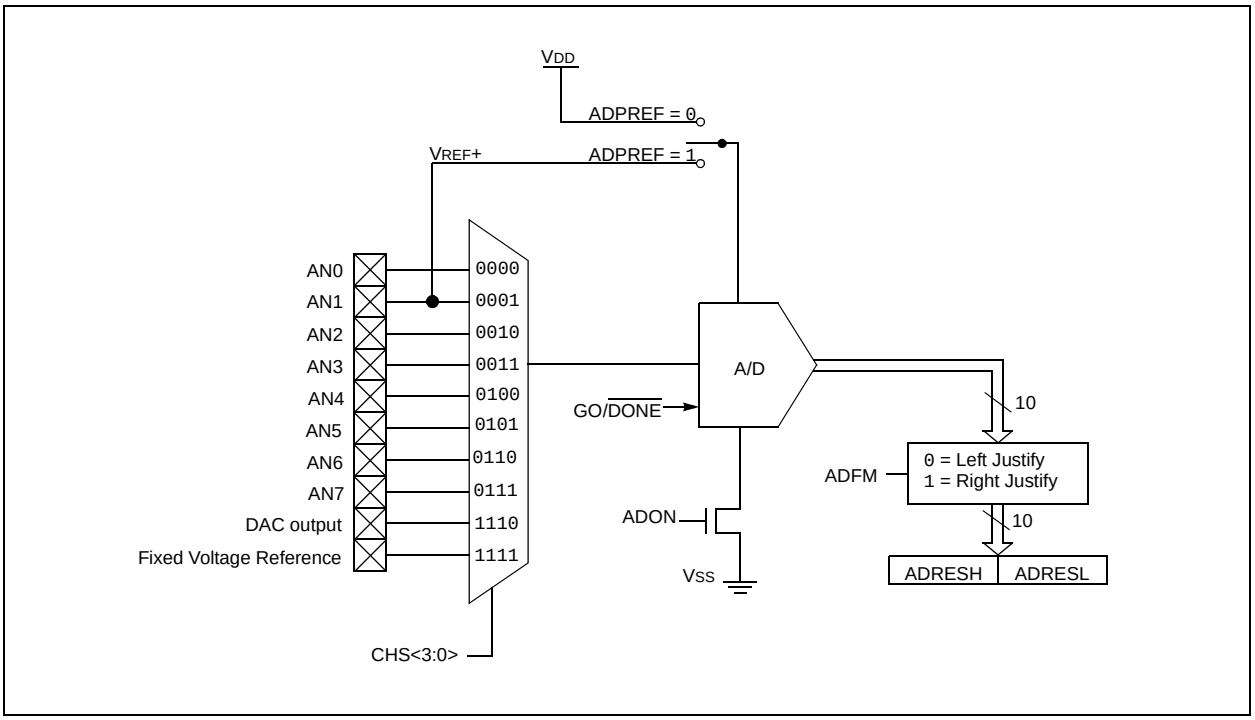
The ADC voltage reference is software selectable to either VDD or a voltage applied to the external reference pins.

The ADC can generate an interrupt upon completion of a conversion. This interrupt can be used to wake up the device from Sleep.

Figure 12-1 shows the block diagram of the ADC.

**Note:** The ADRESL and ADRESH registers are read-only.

**FIGURE 12-1: ADC BLOCK DIAGRAM**



## 12.1 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
- Results formatting

### 12.1.1 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. See the corresponding port section 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. |

### 12.1.2 CHANNEL SELECTION

The CHS bits of the ADCON0 register 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 12.2 "ADC Operation"** for more information.

### 12.1.3 ADC VOLTAGE REFERENCE

The ADPREF1 bit of the ADCON1 register provides control of the positive voltage reference. The positive voltage reference can be either VDD or an external voltage source. The negative voltage reference is always connected to the ground reference.

### 12.1.4 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:

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

The time to complete one bit conversion is defined as TAD. One full 10-bit conversion requires 11 TAD periods as shown in Figure 12-2.

For correct conversion, the appropriate TAD specification must be met. See A/D conversion requirements in **Section 22.0 "Electrical Specifications"** for more information. Table 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. |

**TABLE 12-1: ADC CLOCK PERIOD (TAD) VS. DEVICE OPERATING FREQUENCIES (VDD ≥ 3.0V)**

| ADC Clock Period (TAD) | | Device Frequency (FOSC) | | | |
|---|---|---|---|---|---|
| **ADC Clock Source** | **ADCS<2:0>** | **20 MHz** | **8 MHz** | **4 MHz** | **1 MHz** |
| FOSC/2 | 000 | 100 ns(2) | 250 ns(2) | 500 ns(2) | 2.0 μs |
| FOSC/4 | 100 | 200 ns(2) | 500 ns(2) | 1.0 μs(2) | 4.0 μs |
| FOSC/8 | 001 | 400 ns(2) | 1.0 μs(2) | 2.0 μs | 8.0 μs(3) |
| FOSC/16 | 101 | 800 ns(2) | 2.0 μs | 4.0 μs | 16.0 μs(3) |
| FOSC/32 | 010 | 1.6 μs | 4.0 μs | 8.0 μs(3) | 32.0 μs(3) |
| FOSC/64 | 110 | 3.2 μs | 8.0 μs(3) | 16.0 μs(3) | 64.0 μs(3) |
| FRC | x11 | 2-6 μs(1,4) | 2-6 μs(1,4) | 2-6 μs(1,4) | 2-6 μs(1,4) |

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

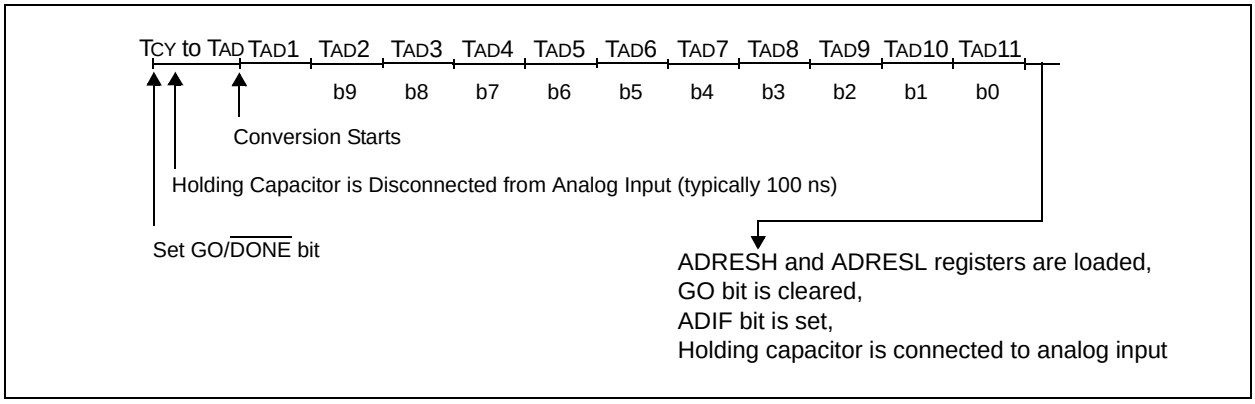
**Note 1:** The FRC source has a typical TAD time of 4 μs for VDD > 3.0V.

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

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

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

**FIGURE 12-2: ANALOG-TO-DIGITAL CONVERSION TAD CYCLES**




### 12.1.5 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:** | The ADIF bit is set at the completion of every conversion, regardless of whether or not the ADC interrupt is enabled. |
|---|---|

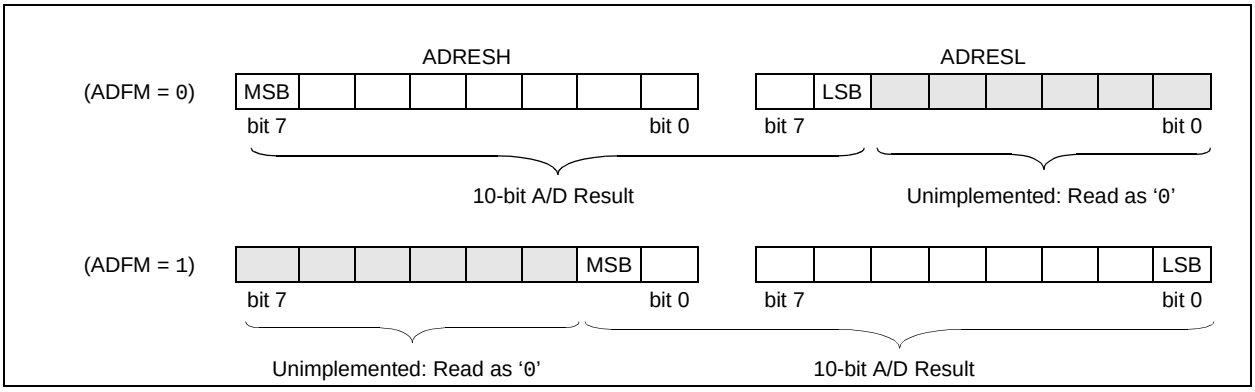
This interrupt can be generated while the device is operating or while in Sleep. If the device is in Sleep, the interrupt will wake-up the device. Upon waking from Sleep, the next instruction following the SLEEP instruction is always executed. If the user is attempting to wake-up from Sleep and resume in-line code execution, the global interrupt must be disabled. If the global interrupt is enabled, execution will switch to the Interrupt Service Routine.

### 12.1.6 RESULT FORMATTING

The 10-bit A/D conversion result can be supplied in two formats, left justified or right justified. The ADFM bit of the ADCON0 register controls the output format.

Figure 12-4 shows the two output formats.

**FIGURE 12-3: 10-BIT A/D CONVERSION RESULT FORMAT**




## 12.2 ADC Operation

### 12.2.1 STARTING A CONVERSION

To enable the ADC module, the ADON bit of the ADCON0 register must be set to a '1'. Setting the GO/$\overline{\text{DONE}}$ bit of the ADCON0 register to a '1' will start the Analog-to-Digital conversion.

| | |
|---|---|
| **Note:** | The GO/$\overline{\text{DONE}}$ bit should not be set in the same instruction that turns on the ADC. Refer to **Section 12.2.6 "A/D Conversion Procedure"**. |

### 12.2.2 COMPLETION OF A CONVERSION

When the conversion is complete, the ADC module will:

- Clear the GO/$\overline{\text{DONE}}$ bit
- Set the ADIF flag bit
- Update the ADRESH:ADRESL registers with new conversion result

### 12.2.3 TERMINATING A CONVERSION

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

| | |
|---|---|
| **Note:** | A device Reset forces all registers to their Reset state. Thus, the ADC module is turned off and any pending conversion is terminated. |

### 12.2.4 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 clock 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.

### 12.2.5 SPECIAL EVENT TRIGGER

The CCP Special Event Trigger allows periodic ADC measurements without software intervention. When this trigger occurs, the GO/$\overline{\text{DONE}}$ bit is set by hardware and the Timer1 counter resets to zero.

Using the Special Event Trigger does not assure proper ADC timing. It is the user's responsibility to ensure that the ADC timing requirements are met.

See **Section 10.0 "Capture/Compare/PWM Modules"** for more information.

### 12.2.6 A/D 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 (See TRIS register)
   - Configure pin as analog
2. Configure the ADC module:
   - Select ADC conversion clock
   - Configure voltage reference
   - Select ADC input channel
   - Select result format
   - 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:** See **Section 12.4 "A/D Acquisition Requirements"**.

**EXAMPLE 12-1: A/D CONVERSION**

```
;This code block configures the ADC
;for polling, Vdd reference, Frc clock
;and RA0 input.
;
;Conversion start & polling for completion
; are included.
;
  BANKSEL  TRISA       ;
  BSF      TRISA,0     ;Set RA0 to input
  BANKSEL  ADCON1      ;
  MOVLW    B'01110000' ;ADC Frc clock,
  IORWF    ADCON1      ; and RA0 as analog
  BANKSEL  ADCON0      ;
  MOVLW    B'10000001' ;Right justify,
  MOVWF    ADCON0      ;Vdd Vref, AN0, On
  CALL     SampleTime  ;Acquisiton delay
  BSF      ADCON0,GO   ;Start conversion
TEST AGAIN
  BTFSC    ADCON0,GO   ;Is conversion done?
  GOTO     TEST AGAIN  ;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
```

## 12.3 Register Definitions: ADC Control

**REGISTER 12-1: ADCON0: A/D CONTROL REGISTER 0**

| R/W-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| ADFM | — | CHS<3:0> | | | | GO/$\overline{\text{DONE}}$ | ADON |
| bit 7 | | | | | | | bit 0 |

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

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

bit 6 **Unimplemented:** Read as ‘0’

bit 5-2 **CHS<3:0>:** Analog Channel Select bits
0000 = AN0
0001 = AN1
0010 = AN2
0011 = AN3
0100 = AN4
0101 = AN5
0110 = AN6
0111 = AN7
1110 = DAC output
1111 = Fixed Voltage Reference

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

bit 0 **ADON:** ADC Enable bit
1 = ADC is enabled
0 = ADC is disabled and consumes no operating current

**REGISTER 12-2: ADCON1: A/D CONTROL REGISTER 1**

| U-0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | U-0 | U-0 | U-0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| — | ADCS<2:0> | | | — | — | — | ADPREF1 |
| 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 6-4 **ADCS<2:0>:** A/D Conversion Clock Select bits
000 = FOSC/2
001 = FOSC/8
010 = FOSC/32
011 = FRC (clock supplied from an internal oscillator with a divisor of 16)
100 = FOSC/4
101 = FOSC/16
110 = FOSC/64

bit 3-1 **Unimplemented:** Read as ‘0’

bit 0 **ADPREF1:** ADC Positive Voltage Reference Selection bit
0 = VDD
1 = VREF+

**REGISTER 12-3: ADRESH: ADC RESULT REGISTER HIGH (ADRESH) ADFM = 0 (READ-ONLY)**

| R-x | R-x | R-x | R-x | R-x | R-x | R-x | R-x |
|---|---|---|---|---|---|---|---|
| ADRESH<9:2> | | | | | | | |
| bit 7 | | | | | | | bit 0 |

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

bit 7-0 **ADRESH<9:2>**: ADC Result Register bits
Upper eight bits of 10-bit conversion result

**REGISTER 12-4: ADRESL: ADC RESULT REGISTER LOW (ADRESL) ADFM = 0 (READ-ONLY)**

| R-x | R-x | U-0 | U-0 | U-0 | U-0 | U-0 | U-0 |
|---|---|---|---|---|---|---|---|
| ADRESL<7:0> | | | | | | | |
| bit 7 | | | | | | | bit 0 |

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

bit 7-0 **ADRESL<7:0>**: ADC Result Register bits
Lower two bits of 10-bit conversion result

**REGISTER 12-5: ADRESH: ADC RESULT REGISTER HIGH (ADRESH) ADFM = 1 (READ-ONLY)**

| U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | R-x | R-x |
|---|---|---|---|---|---|---|---|
| — | — | — | — | — | — | ADRESH<9:8> | |
| bit 7 | | | | | | | bit 0 |

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

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

bit 1-0 **ADRESH<9:8>**: ADC Result Register bits
Upper two bits of 10-bit conversion result

**REGISTER 12-6: ADRESL: ADC RESULT REGISTER LOW (ADRESL) ADFM = 1 (READ-ONLY)**

| R-x | R-x | R-x | R-x | R-x | R-x | R-x | R-x |
|---|---|---|---|---|---|---|---|
| ADRESL<7:0> | | | | | | | |
| bit 7 | | | | | | | bit 0 |

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

bit 7-0 **ADRESL<7:0>**: ADC Result Register bits
Lower eight bits of 10-bit conversion result

## 12.4 A/D 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 12-4. 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), see Figure 12-4. **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 A/D acquisition must be done before the conversion can be started. To calculate the minimum acquisition time, Equation 12-1 may be used. This equation assumes that 1/2 LSb error is used (1024 steps for the ADC). The 1/2 LSb error is the maximum error allowed for the ADC to meet its specified resolution.

**EQUATION 12-1: ACQUISITION TIME EXAMPLE**

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

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

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

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

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

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

*Solving for* $T_C$*:*

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

*Therefore:*

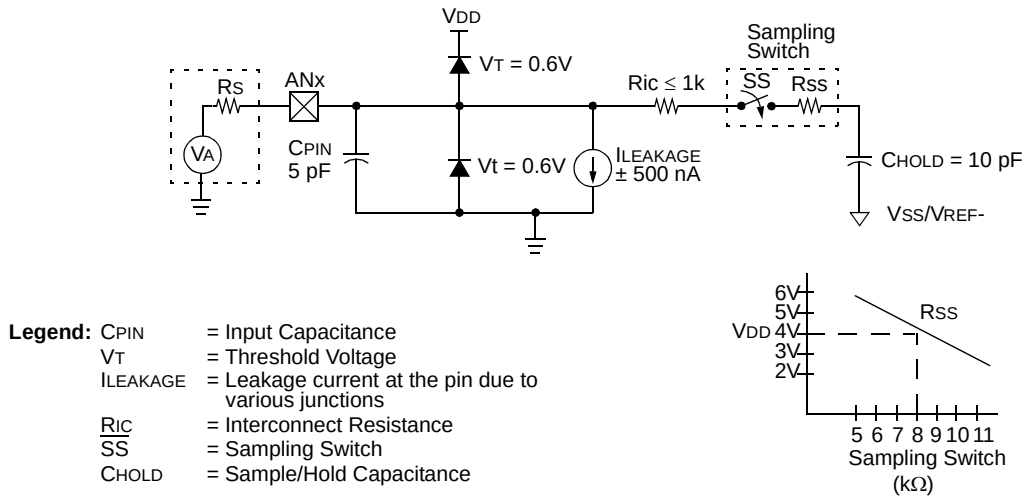
$$\begin{aligned} T_{ACQ} &= 2\mu s + 1.37\mu s + [(50°C - 25°C)(0.05\mu s/°C)] \\ &= 4.67\mu s \end{aligned}$$

**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.

**FIGURE 12-4: ANALOG INPUT MODEL**




**Legend:** CPIN = Input Capacitance
VT = Threshold Voltage
ILEAKAGE = Leakage current at the pin due to various junctions
RIC = Interconnect Resistance
SS = Sampling Switch
CHOLD = Sample/Hold Capacitance

**FIGURE 12-5: ADC TRANSFER FUNCTION**

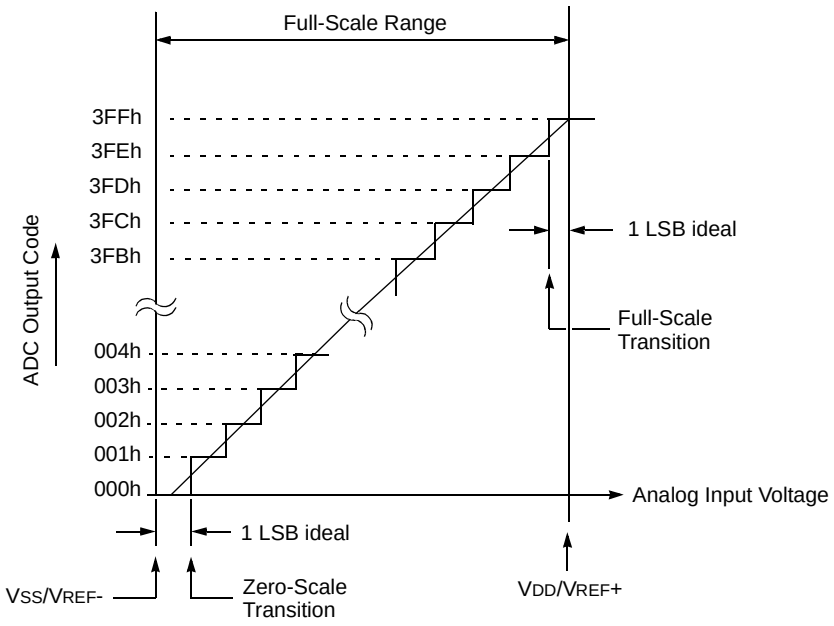


**TABLE 12-2: SUMMARY OF ASSOCIATED ADC REGISTERS**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|
| ADCON0 | ADFM | — | CHS<3:0> | | | | GO/$\overline{\text{DONE}}$ | ADON | 109 |
| ADCON1 | — | ADCS<2:0> | | | — | — | — | ADPREF1 | 110 |
| ANSELA | — | — | — | ANSA4 | — | ANSA2 | ANSA1 | ANSA0 | 44 |
| ADRESH(2) | Most Significant eight bits of the left shifted A/D result or two bits of the right shifted result | | | | | | | | 111* |
| ADRESL(2) | Least Significant two bits of the left shifted result or eight bits of the right shifted result | | | | | | | | 109* |
| PORTA | — | — | RA5 | RA4 | RA3 | RA2 | RA1 | RA0 | 43 |
| INTCON | GIE | PEIE | T0IE | INTE | IOCIE | T0IF | INTF | IOCIF | 17 |
| PIE1 | TMR1GIE | ADIE | — | — | HLTMR2IE | HLTMR1IE | TMR2IE | TMR1IE | 18 |
| PIR1 | TMR1GIF | ADIF | — | — | HLTMR2IF | HLTMR1IF | TMR2IF | TMR1IF | 20 |
| TRISA | — | — | TRISA5 | TRISA4 | TRISA3(1) | TRISA2 | TRISA1 | TRISA0 | 43 |

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

* Page provides register information.

**Note 1:** TRISA3 always reads '1'.

**2:** Read-only register.

## 13.0 FIXED VOLTAGE REFERENCE (FVR)

The Fixed Voltage Reference (FVR) is a stable voltage reference, independent of VDD, with 1.2V output level. The output of the FVR can be configured to supply a reference voltage to the following:

- ADC input channel
- Comparator 1 positive input (C1VP)
- Comparator 2 positive input (C2VP)
- FVR_out pin
- Shunt regulator

On the PIC16F753, the FVR is enabled by setting the FVREN bit of the FVRCON register. The FVR is always enabled on the PIC16HV753 device.

### 13.1 Fixed Voltage Reference Output

The FVR output can be applied to the FVROUT pin by setting the FVRBUFSS and FVRBUFEN bits of the FVRCON register. The FVRBUFSS bit selects the op amp, FVR or DAC output reference to the FVROUT pin buffer. The FVRBUFEN bit enables the output buffer to the FVROUT pin.

Enabling the FVROUT pin automatically overrides any digital input or output functions of the pin. Reading the FVROUT pin when it has been configured for a reference voltage output will always return a '0'.

### 13.2 FVR Stabilization Period

When the Fixed Voltage Reference module is enabled, it requires time for the reference circuit to stabilize. Once the circuit stabilizes and is ready for use, the FVRRDY bit of the FVRCON register will be set. See **Section 22.0 "Electrical Specifications"** for the minimum delay requirement.

### 13.3 Operation During Sleep

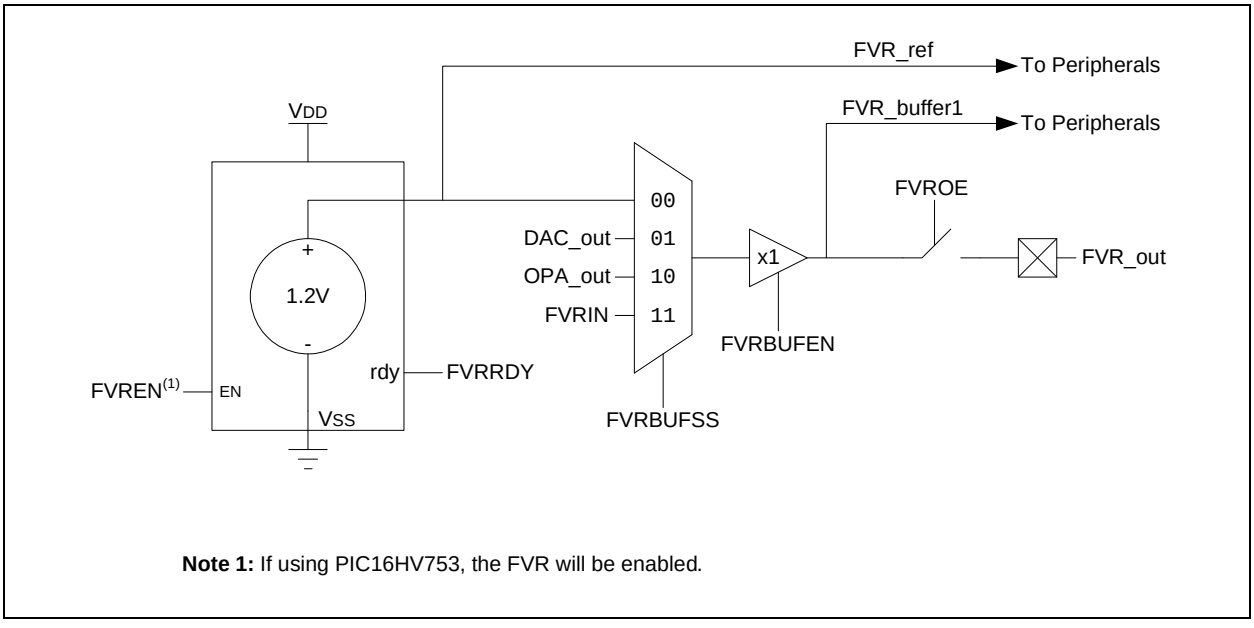
When the device wakes up from Sleep through an interrupt or a Watchdog Timer time-out, the contents of the FVRCON register are not affected. To minimize current consumption in Sleep mode, the FVR voltage reference should be disabled.

### 13.4 Effects of a Reset

A device Reset clears the FVRCON register. As a result:

- The FVR module is disabled
- The FVR voltage output is disabled on the FVROUT pin

**FIGURE 13-1: VOLTAGE REFERENCE BLOCK DIAGRAM**




**Note 1:** If using PIC16HV753, the FVR will be enabled.

## 13.5 Register Definitions: FVR Control

**REGISTER 13-1: FVR1CON0: FIXED VOLTAGE REFERENCE CONTROL REGISTER**

| R/W-0/0 | R-q/q | R/W-0/0 | R/W-0/0 | R/W-0/0 | U-0 | U-0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| FVREN | FVRRDY | FVROE | FVRBUFSS1 | FVRBUFSS0 | — | — | FVRBUFEN |
| 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
0 = Fixed Voltage Reference is disabled
1 = Fixed Voltage Reference is enabled

bit 6 **FVRRDY:** Fixed Voltage Reference Ready Flag bit
0 = Fixed Voltage Reference output is not ready or not enabled bit
1 = Fixed Voltage Reference output is ready for use

bit 5 **FVROE:** Voltage Reference Output Pin Buffer Enable bit
0 = Output pass gate is disabled
1 = Output pass gate is enabled

bit 4-3 **FVRBUFSS<1:0>:** Voltage Reference Pin Buffer Source Select bits
00 = Selects the output of the band gap as the input
01 = DAC output
10 = Op amp buffered output
11 = Selects FVRIN (RA1)

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

bit 0 **FVRBUFEN:** Voltage Reference Output Pin Buffer Enable bit
0 = Output buffer is disabled
1 = Output buffer is enabled

**TABLE 13-1: 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 |
|---|---|---|---|---|---|---|---|---|---|
| FVR1CON0 | FVREN | FVRRDY | FVROE | FVRBUFSS1 | FVRBUFSS0 | — | — | FVRBUFEN | 116 |

**Legend:** Shaded cells are not used with the Fixed Voltage Reference.

## 14.0 DIGITAL-TO-ANALOG CONVERTER (DAC) MODULE

The Digital-to-Analog Converter supplies a variable voltage reference, ratiometric with the input source, with 512 selectable output levels.

The input of the DAC can be connected to:

- External VREF pins
- VDD supply voltage
- FVR (Fixed Voltage Reference)

The output of the DAC can be configured to supply a reference voltage to the following:

- Comparator positive input
- ADC input channel
- DACxOUT pin
- Op amp

The Digital-to-Analog Converter (DAC) is enabled by setting the DACEN bit of the DACxCON0 register.

### 14.1 Output Voltage Selection

The DAC has 512 voltage level ranges. The 512 levels are set with the DACR<8:1> bits of the DACxREFH register and DACR0 of the DACxREFL.

The DAC output voltage is determined by Equation 14-1:

**EQUATION 14-1: DAC OUTPUT VOLTAGE**

*IF DACEN = 1*

$$V_{OUT} = \left( (V_{SOURCE+} - V_{SOURCE-}) \times \frac{DACR[8]}{2^9} \right) + V_{SOURCE-}$$

$V_{SOURCE+} = V_{DD}, V_{REF}$, *OPA1OUTor FVR BUFFER 2*

$V_{SOURCE-} = V_{SS}$

### 14.2 Ratiometric Output Level

The DAC output value is derived using a resistor ladder with each end of the ladder tied to a positive and negative voltage reference input source. If the voltage of either input source fluctuates, a similar fluctuation will result in the DAC output value.

The value of the individual resistors within the ladder can be found in **Section 22.0 "Electrical Specifications"**.

### 14.3 DAC Voltage Reference Output

The DAC voltage can be output to the DACxOUT pins by setting the DACOE1 bit of the DACxCON0 register. Selecting the DAC reference voltage for output on the DACxOUT pin automatically overrides the digital output buffer and digital input threshold detector functions of that pin. Reading the DACxOUT pin when it has been configured for DAC reference voltage output will always return a '0'.

Due to the limited current drive capability, a buffer must be used on the DAC voltage reference output for external connections to either DACxOUT pin. Figure 14-2 shows a buffering technique example.

### 14.4 DAC Justification

The DAC can be configured to be left or right justified based on application needs. In order for justification to work properly, all 16 bits of the DAC buffer register (DACxREFH:DACxREFL register pair) must be loaded in the correct sequence to get the effective 9-bit result. In most applications, DACxREFL is written prior to DACxREFH, regardless of justification. The DAC buffer is loaded at the end of the write cycle that writes DACxREFH register.

**FIGURE 14-1: DIGITAL-TO-ANALOG CONVERTER BLOCK DIAGRAM**

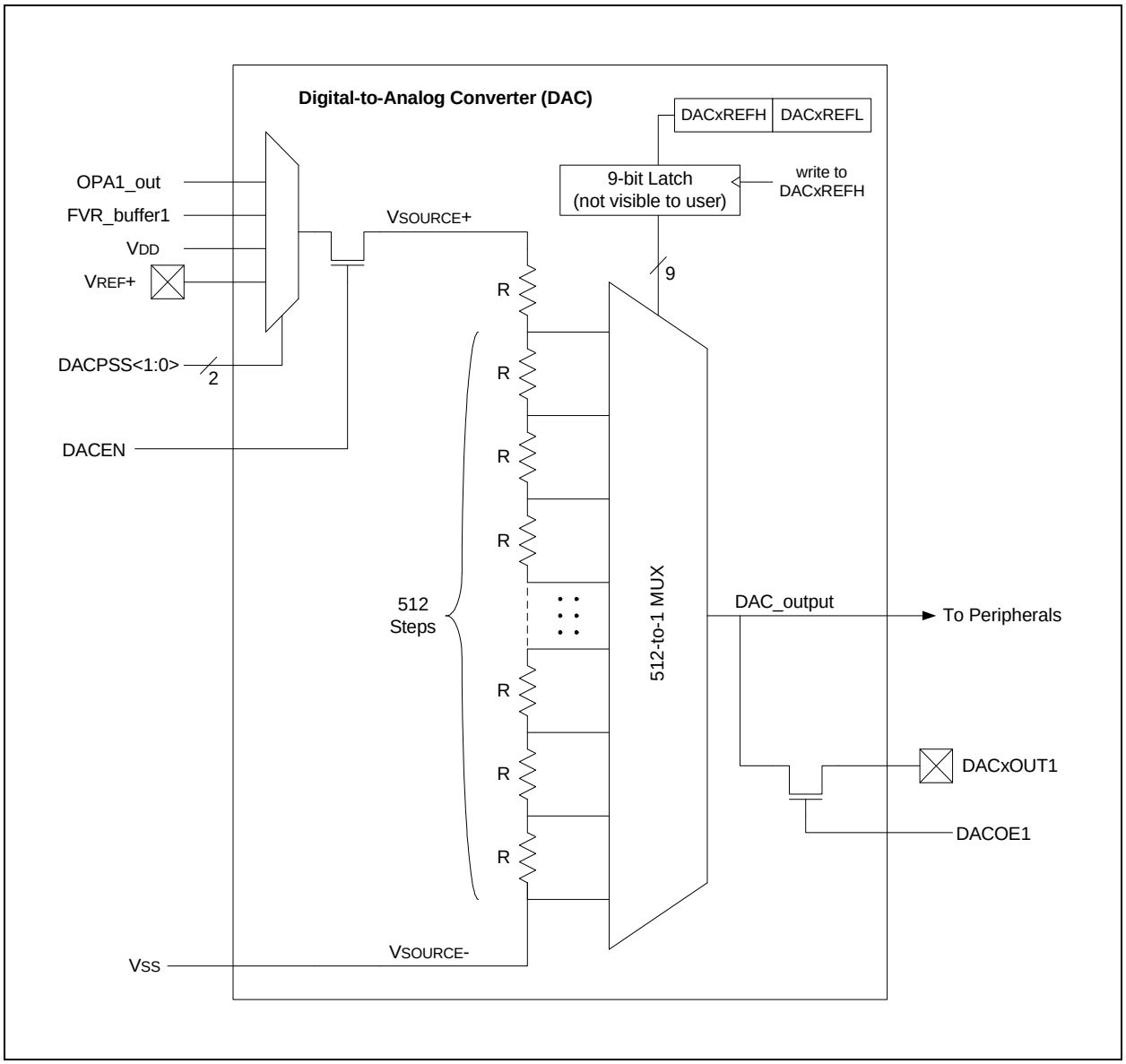
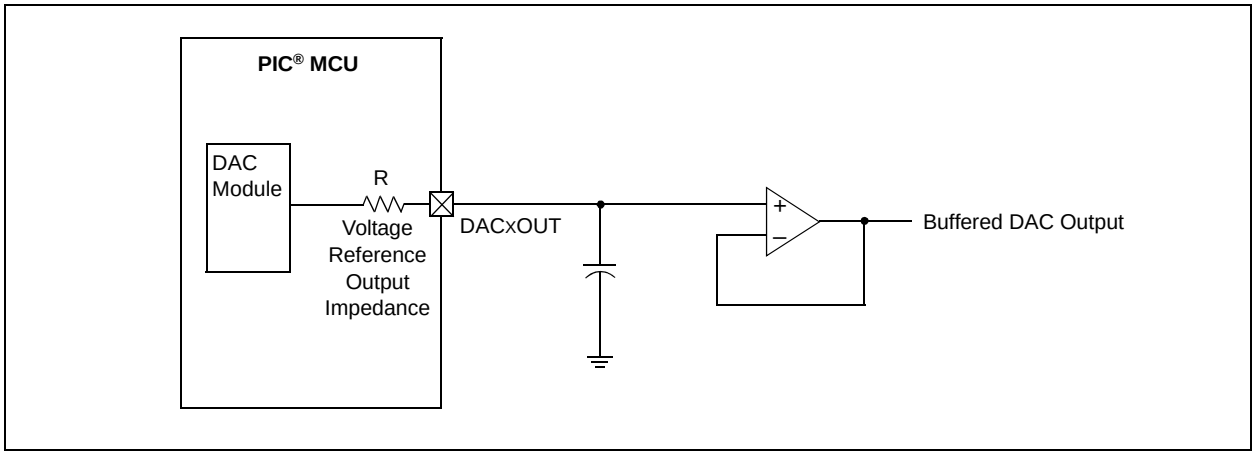



**FIGURE 14-2: VOLTAGE REFERENCE OUTPUT BUFFER EXAMPLE**



## 14.5 Operation During Sleep

When the device wakes up from Sleep through an interrupt or a Watchdog Timer time-out, the contents of the DACxCON0 register are not affected. To minimize current consumption in Sleep mode, the voltage reference should be disabled.

## 14.6 Effects of a Reset

A device Reset affects the following:

- DAC is disabled
- DAC output voltage is removed from the DACxOUT pin
- The DACR<8:0> range select bits are cleared

## 14.7 Register Definitions: DAC Control

**REGISTER 14-1: DACxCON0: VOLTAGE REFERENCE CONTROL REGISTER 0**

| R/W-0/0 | R/W-0/0 | R/W-0/0 | U-0 | R/W-0/0 | R/W-0/0 | U-0 | U-0 |
|---|---|---|---|---|---|---|---|
| DACEN | DACFM | DACOE | — | DACPSS<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 **DACEN:** DAC Enable bit
1 = DACx is enabled
0 = DACx is disabled

bit 6 **DACFM:** DAC Output Format bit
1 = DACx output result is right justified
0 = DACx output result is left justified

bit 5 **DACOE:** DAC Voltage Output Enable bit
1 = DACx voltage level is also an output on the DACxOUT pin
0 = DACx voltage level is disconnected from the DACxOUT pin

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

bit 3-2 **DACPSS<1:0>:** DAC Positive Source Select bits
11 = FVR output
10 = $V_{REF}$+ pin
01 = OPA1OUT pin
00 = $V_{DD}$

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

**REGISTER 14-2: DACxREFH: DAC REFERENCE HIGH REGISTER (DACxFM = 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 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| DACR<8:1> | | | | | | | |
| 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 7-0 **DACR<8:1>**: DAC Reference Selection bits
DACxOUT = (DACR<8:0> x (Vdac_ref)/512)

**REGISTER 14-3: DACxREFL: DAC REFERENCE LOW REGISTER (DACxFM = 0)**

| R/W-0/0 | U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | U-0 |
|---|---|---|---|---|---|---|---|
| DACR0 | — | — | — | — | — | — | — |
| 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 **DACR0**: DAC Reference Selection bits
DACxOUT = (DACR<8:0> x (Vdac_ref)/512)

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

**REGISTER 14-4: DACxREFH: DAC REFERENCE HIGH REGISTER (DACxFM = 1)**

| U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| — | — | — | — | — | — | — | DACR8 |
| 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 **DACR8**: DAC Reference Selection bits
DACxOUT = (DACR<8:0> x (Vdac_ref) / 512)

**REGISTER 14-5: DACxREFL: DAC REFERENCE LOW REGISTER (DACxFM = 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 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| DACR<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 **DACR<7:0>**: DAC Reference Selection bits
DACxOUT = (DACR<8:0> x (Vdac_ref) / 512)

**TABLE 14-1: SUMMARY OF REGISTERS ASSOCIATED WITH THE DAC MODULE**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|
| DACxCON0 | DACEN | DACFM | DACOE | — | DACPSS<1:0> | | — | — | 120 |
| DACxREFH | DACR<8:1> | | | | | | | | 121 |
| DACxREFH | — | — | — | — | — | — | — | DACR8 | 122 |
| DACxREFL | DACR<7:0> | | | | | | | | 121 |
| DACxREFL | DACR0 | — | — | — | — | — | — | — | 122 |

**Legend:** — = Unimplemented location, read as ‘0’. Shaded cells are not used with the DAC module.

## 15.0 COMPARATOR MODULE

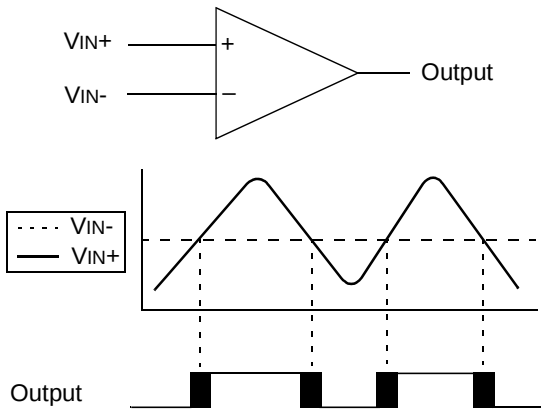
Comparators are used to interface analog circuits to a digital circuit by comparing two analog voltages and providing a digital indication of their relative magnitudes. Comparators are very useful mixed-signal building blocks because they provide analog functionality independent of program execution. The analog comparator module includes the following features:

- Independent comparator control
- Programmable input selection
- Comparator output is available internally/externally
- Programmable output polarity
- Interrupt-on-change
- Wake-up from Sleep
- Programmable Speed/Power optimization
- PWM shutdown
- Programmable and Fixed Voltage Reference

### 15.1 Comparator Overview

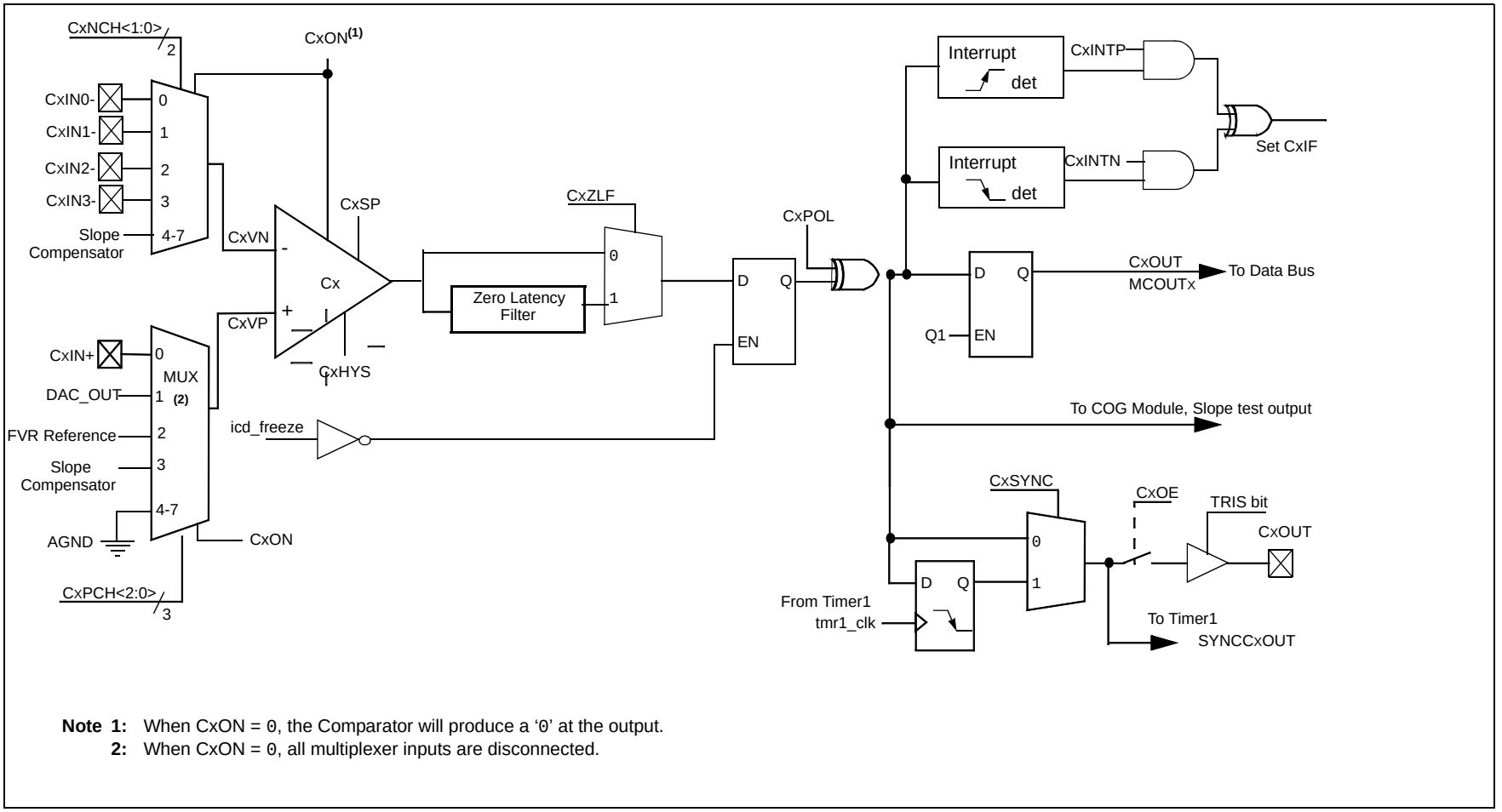
A single comparator is shown in Figure 15-1 along with the relationship between the analog input levels and the digital output. When the analog voltage at VIN+ is less than the analog voltage at VIN-, the output of the comparator is a digital low level. When the analog voltage at VIN+ is greater than the analog voltage at VIN-, the output of the comparator is a digital high level.

**FIGURE 15-1: SINGLE COMPARATOR**




**Note:** The black areas of the output of the comparator represents the uncertainty due to input offsets and response time.

**FIGURE 15-2: COMPARATOR MODULE SIMPLIFIED BLOCK DIAGRAM**




**Note 1:** When CxON = 0, the Comparator will produce a '0' at the output.
**2:** When CxON = 0, all multiplexer inputs are disconnected.

## 15.2 Comparator Control

Each comparator has two control registers: CMxCON0 and CMxCON1.

The CMxCON0 registers (see Register 15-1) contain Control and Status bits for the following:

- Enable
- Output selection
- Output pin enable
- Output polarity
- Speed/Power selection
- Hysteresis enable
- Output synchronization

The CMxCON1 registers (see Register 15-2) contain Control bits for the following:

- Interrupt edge polarity (rising and/or falling)
- Positive input channel selection
- Negative input channel selection

### 15.2.1 COMPARATOR ENABLE

Setting the CxON bit of the CMxCON0 register enables the comparator for operation. Clearing the CxON bit disables the comparator resulting in minimum current consumption.

### 15.2.2 COMPARATOR OUTPUT SELECTION

The output of the comparator can be monitored by reading either the CxOUT bit of the CMxCON0 register or the MCOUTx bit of the CMOUT register. In order to make the output available for an external connection, the following conditions must be true:

- CxOE bit of the CMxCON0 register must be set
- Corresponding TRIS bit must be cleared
- CxON bit of the CMxCON0 register must be set

| | |
|---|---|
| **Note 1:** | The CxOE bit of the CMxCON0 register overrides the PORT data latch. Setting the CxON bit of the CMxCON0 register has no impact on the port override. |
| **2:** | The internal output of the comparator is latched with each instruction cycle. Unless otherwise specified, external outputs are not latched. |

### 15.2.3 COMPARATOR OUTPUT POLARITY

Inverting the output of the comparator is functionally equivalent to swapping the comparator inputs. The polarity of the comparator output can be inverted by setting the CxPOL bit of the CMxCON0 register. Clearing the CxPOL bit results in a non-inverted output.

Table 15-1 shows the output state versus input conditions, including polarity control.

**TABLE 15-1: COMPARATOR OUTPUT STATE VS. INPUT CONDITIONS**

| Input Condition | CxPOL | CxOUT |
|---|---|---|
| CxVN > CxVP | 0 | 0 |
| CxVN < CxVP | 0 | 1 |
| CxVN > CxVP | 1 | 1 |
| CxVN < CxVP | 1 | 0 |

### 15.2.4 COMPARATOR SPEED/POWER SELECTION

The trade-off between speed or power can be optimized during program execution with the CxSP control bit. The default state for this bit is '1' which selects the normal speed mode. Device power consumption can be optimized at the cost of slower comparator propagation delay by clearing the CxSP bit to '0'.

## 15.3 Comparator Hysteresis

A selectable amount of separation voltage can be added to the input pins of each comparator to provide a hysteresis function to the overall operation. Hysteresis is enabled by setting the CxHYS bit of the CMxCON0 register.

See **Section 22.0 "Electrical Specifications"** for more information.

## 15.4 Timer1 Gate Operation

The output resulting from a comparator operation can be used as a source for gate control of Timer1. See **Section 7.5 "Timer1 Gate"** for more information. This feature is useful for timing the duration or interval of an analog event.

It is recommended that the comparator output be synchronized to Timer1. This ensures that Timer1 does not increment while a change in the comparator is occurring.

#### 15.4.1 COMPARATOR OUTPUT SYNCHRONIZATION

The output from either comparator, C1 or C2, can be synchronized with Timer1 by setting the CxSYNC bit of the CMxCON0 register.

Once enabled, the comparator output is latched on the falling edge of the Timer1 source clock. If a prescaler is used with Timer1, the comparator output is latched after the prescaling function. To prevent a race condition, the comparator output is latched on the falling edge of the Timer1 clock source and Timer1 increments on the rising edge of its clock source. See the Comparator Block Diagram (Figure 15-2) and the Timer1 Block Diagram (Figure 7-1) for more information.

## 15.5 Comparator Interrupt

An interrupt can be generated upon a change in the output value of the comparator for each comparator, a rising edge detector and a falling edge detector are present.

When either edge detector is triggered and its associated enable bit is set (CxINTP and/or CxINTN bits of the CMxCON1 register), the Corresponding Interrupt Flag bit (CxIF bit of the PIR2 register) will be set.

To enable the interrupt, you must set the following bits:

- CxON, CxPOL and CxSP bits of the CMxCON0 register
- CxIE bit of the PIE2 register
- CxINTP bit of the CMxCON1 register (for a rising edge detection)
- CxINTN bit of the CMxCON1 register (for a falling edge detection)
- PEIE and GIE bits of the INTCON register

The associated interrupt flag bit, CxIF bit of the PIR2 register, must be cleared in software. If another edge is detected while this flag is being cleared, the flag will still be set at the end of the sequence.

| | |
|---|---|
| **Note:** | Although a comparator is disabled, an interrupt can be generated by changing the output polarity with the CxPOL bit of the CMxCON0 register, or by switching the comparator on or off with the CxON bit of the CMxCON0 register. |

## 15.6 Comparator Positive Input Selection

Configuring the CxPCH<1:0> bits of the CMxCON1 register directs an internal voltage reference or an analog pin to the non-inverting input of the comparator:

- CxIN0+ analog pin
- DAC Reference Voltage (DAC_REF)
- FVR Reference Voltage (FVR_REF)
- VSS (Ground)

See **Section 13.0 “Fixed Voltage Reference (FVR)”** for more information on the Fixed Voltage Reference module.

See **Section 14.0 “Digital-to-Analog Converter (DAC) Module”** for more information on the DAC input signal.

Any time the comparator is disabled (CxON = 0), all comparator inputs are disabled.

## 15.7 Comparator Negative Input Selection

The CxNCH0 bit of the CMxCON0 register selects the analog input pin to the comparator inverting input.

| | |
|---|---|
| **Note:** | To use CxIN0+ and CxIN1x- pins as analog input, the appropriate bits must be set in the ANSEL register and the corresponding TRIS bits must also be set to disable the output drivers. |

## 15.8 Comparator Response Time

The comparator output is indeterminate for a period of time after the change of an input source or the selection of a new reference voltage. This period is referred to as the response time. The response time of the comparator differs from the settling time of the voltage reference. Therefore, both of these times must be considered when determining the total response time to a comparator input change. See the Comparator and Voltage Reference Specifications in **Section 22.0 “Electrical Specifications”** for more details.

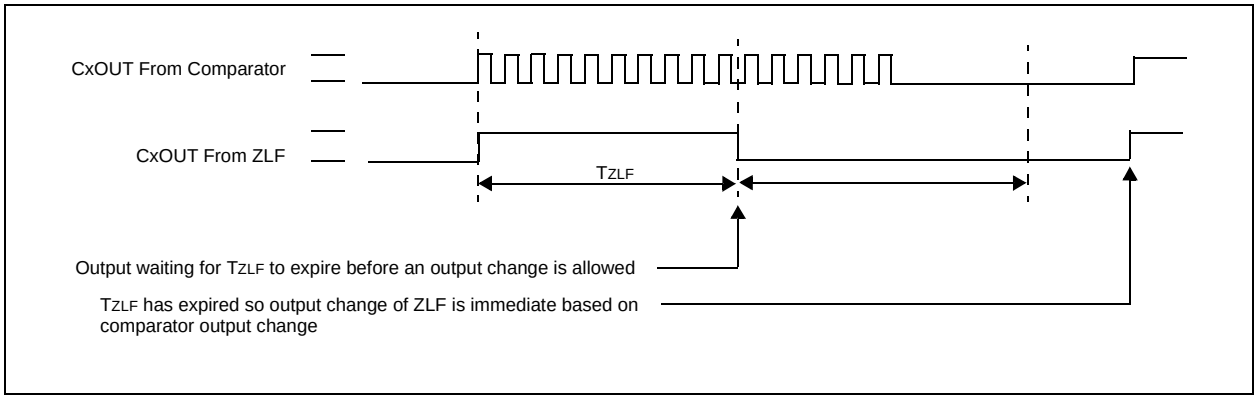
## 15.9 Interaction with the COG Module

The comparator outputs can be brought to the COG module in order to facilitate auto-shutdown. If auto-restart is also enabled, the comparators can be configured as a closed loop analog feedback to the COG, thereby creating an analog controlled PWM.

## 15.10 Zero Latency Filter

In high-speed operation, and under proper circuit conditions, it is possible for the comparator output to oscillate. This oscillation can have adverse effects on the hardware and software relying on this signal. Therefore, a digital filter has been added to the comparator output to suppress the comparator output oscillation. Once the comparator output changes, the output is prevented from reversing the change for a nominal time of 20 ns. This allows the comparator output to stabilize without affecting other dependent devices. Refer to Figure 15-3.

**FIGURE 15-3: COMPARATOR ZERO LATENCY FILTER OPERATION**



## 15.11 Analog Input Connection Considerations

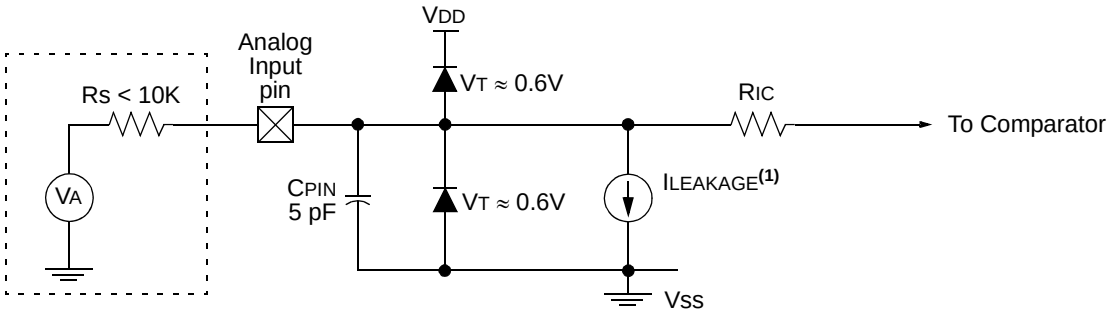
A simplified circuit for an analog input is shown in Figure 15-4. Since the analog input pins share their connection with a digital input, they have reverse biased ESD protection diodes to VDD and VSS. The analog input, therefore, must be between VSS and VDD. If the input voltage deviates from this range by more than 0.6V in either direction, one of the diodes is forward biased and a latch-up may occur.

A maximum source impedance of 10 kΩ is recommended for the analog sources. Also, any external component connected to an analog input pin, such as a capacitor or a Zener diode, should have very little leakage current to minimize inaccuracies introduced.

**Note 1:** When reading a PORT register, all pins configured as analog inputs will read as a '0'. Pins configured as digital inputs will convert as an analog input, according to the input specification.

**2:** Analog levels on any pin defined as a digital input, may cause the input buffer to consume more current than is specified.

**FIGURE 15-4: ANALOG INPUT MODEL**




**Legend:** CPIN = Input Capacitance
ILEAKAGE = Leakage Current at the pin due to various junctions
RIC = Interconnect Resistance
RS = Source Impedance
VA = Analog Voltage
VT = Threshold Voltage

**Note 1:** See **Section 22.0 "Electrical Specifications"**.

## 15.12 Register Definitions: Comparator Control

**REGISTER 15-1: CMxCON0: COMPARATOR Cx CONTROL REGISTER 0**

| R/W-0/0 | R-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-1/1 | R/W-0/0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| CxON | CxOUT | CxOE | CxPOL | CxZLF | CxSP | CxHYS | CxSYNC |
| 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 **CxON:** Comparator Enable bit
1 = Comparator is enabled
0 = Comparator is disabled and consumes no active power

bit 6 **CxOUT:** Comparator Output bit
If CxPOL = 1 (inverted polarity):
1 = CxVP < CxVN
0 = CxVP > CxVN
If CxPOL = 0 (non-inverted polarity):
1 = CxVP > CxVN
0 = CxVP < CxVN

bit 5 **CxOE:** Comparator Output Enable bit
1 = CxOUT is present on the CxOUT pin. Requires that the associated TRIS bit be cleared to actually drive the pin. Not affected by CxON.
0 = CxOUT is internal only

bit 4 **CxPOL:** Comparator Output Polarity Select bit
1 = Comparator output is inverted
0 = Comparator output is not inverted

bit 3 **CxZLF:** Zero Latency Filter Enable bit
1 = Zero latency filter is enabled
0 = Zero latency filter is disabled

bit 2 **CxSP:** Comparator Speed/Power Select bit
1 = Comparator operates in normal power, higher speed mode
0 = Comparator operates in low-power, low-speed mode

bit 1 **CxHYS:** Comparator Hysteresis Enable bit
1 = Comparator hysteresis enabled
0 = Comparator hysteresis disabled

bit 0 **CxSYNC:** Comparator Output Synchronous Mode bit
1 = Comparator output to Timer1 and I/O pin is synchronous to changes on Timer1 clock source. Output updated on the falling edge of Timer1 clock source.
0 = Comparator output to Timer1 and I/O pin is asynchronous.

**REGISTER 15-2: CMxCON1: COMPARATOR Cx CONTROL REGISTER 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 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| CxINTP | CxINTN | CxPCH<1:0> | | | CxNCH<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 **CxINTP:** Comparator Interrupt on Positive Going Edge Enable bit
1 = The CxIF interrupt flag will be set upon a positive going edge of the CxOUT bit
0 = No interrupt flag will be set on a positive going edge of the CxOUT bit

bit 6 **CxINTN:** Comparator Interrupt on Negative Going Edge Enable bit
1 = The CxIF interrupt flag will be set upon a negative going edge of the CxOUT bit
0 = No interrupt flag will be set on a negative going edge of the CxOUT bit

bit 5-3 **CxPCH<1:0>:** Comparator Positive Input Channel Select bits
000 = CxVP connects to CxIN+ pin
001 = CxVP connects to dac_out
010 = CxVP connects to FVR
011 = CxVP connects to Slope Compensator Output
1xx = CxVP connects to AGND

bit 2-0 **CxNCH<2:0>:** Comparator Negative Input Channel Select bits
000 = CxVN connects to CxIN0- pin
001 = CxVN connects to CxIN1- pin
010 = CxVN connects to CxIN2- pin
011 = CxVN connects to CxIN3- pin
1xx = CxVN connects to Slope Compensator Output

**REGISTER 15-3: CMOUT: COMPARATOR OUTPUT REGISTER**

| U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | R-0/0 | R-0/0 |
|---|---|---|---|---|---|---|---|
| — | — | — | — | — | — | MCOUT2 | MCOUT1 |
| 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 **MCOUT2:** Mirror Copy of C2OUT bit

bit 0 **MCOUT1:** Mirror Copy of C1OUT bit

**TABLE 15-2: SUMMARY OF REGISTERS ASSOCIATED WITH COMPARATOR MODULE**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|
| CM1CON0 | C1ON | C1OUT | C1OE | C1POL | C1ZLF | C1SP | C1HYS | C1SYNC | 129 |
| CM1CON1 | C1INTP | C1INTN | C1PCH<2:0> | | | C1NCH<2:0> | | | 130 |
| CM2CON0 | C2ON | C2OUT | C2OE | C2POL | C2ZLF | C2SP | C2HYS | C2SYNC | 129 |
| CM2CON1 | C2INTP | C2INTN | C2PCH<2:0> | | | C2NCH<2:0> | | | 130 |
| CMOUT | — | — | — | — | — | — | MCOUT2 | MCOUT1 | 130 |
| DAC1CON0 | DACEN | DACFM | DACOE | — | DACPSS1 | DACPSS0 | — | — | 120 |
| DAC1REFL | Least Significant bit of the left shifted result or eight bits of the right shifted DAC setting | | | | | | | | 122 |
| FVR1CON0 | FVREN | FVRRDY | FVROE | FVRBUFSS1 | FVRBUFSS0 | — | — | FVRBUFEN | 116 |
| INTCON | GIE | PEIE | T0IE | INTE | IOCIE | T0IF | INTF | IOCIF | 17 |
| PIE2 | — | — | C2IE | C1IE | — | COG1IE | — | CCP1IE | 19 |
| PIR2 | — | — | C2IF | C1IF | — | COG1IF | — | CCP1IF | 21 |
| TRISA | — | — | TRISA5 | TRISA4 | TRISA3(1) | TRISA2 | TRISA1 | TRISA0 | 43 |
| ANSELA | — | — | — | ANSA4 | — | ANSA2 | ANSA1 | ANSA0 | 44 |

**Legend:** — = unimplemented location, read as '0'. Shaded cells are unused by the comparator module.
**Note 1:** TRISA3 always reads '1'.

## 16.0 OPERATIONAL AMPLIFIER (OPA) MODULE

The Operational Amplifier (OPA) is a standard three-terminal device requiring external feedback to operate. The OPA module has the following features:

- External Connections to I/O Ports
- Selectable Unity Gain Bandwidth Product Option
- Low-Leakage Inputs
- Factory-Calibrated Input Offset Voltage

### 16.1 OPAxCON0 Register

The OPAxCON0 register, shown in Register 16-1, controls the OPA module.

The OPA module is enabled by setting the OPAxEN bit of the OPAxCON register. When enabled, the OPA forces the output driver of the OPAxOUT pin into tri-state to prevent contention between the driver and the OPA output.

The OPAxUGM bit of the OPAxCON register enables the Unity Gain Bandwidth mode (voltage follower) of the amplifier. In Unity Gain mode, the OPAxNCH<1:0> inputs are disabled. The default mode is normal three-terminal operation.

| | |
|---|---|
| **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 **Section 22.0 "Electrical Specifications"** for the op amp output drive capability. |

### 16.2 Effects of a Reset

A device Reset forces all registers to their Reset state. This disables the OPA module.

### 16.3 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

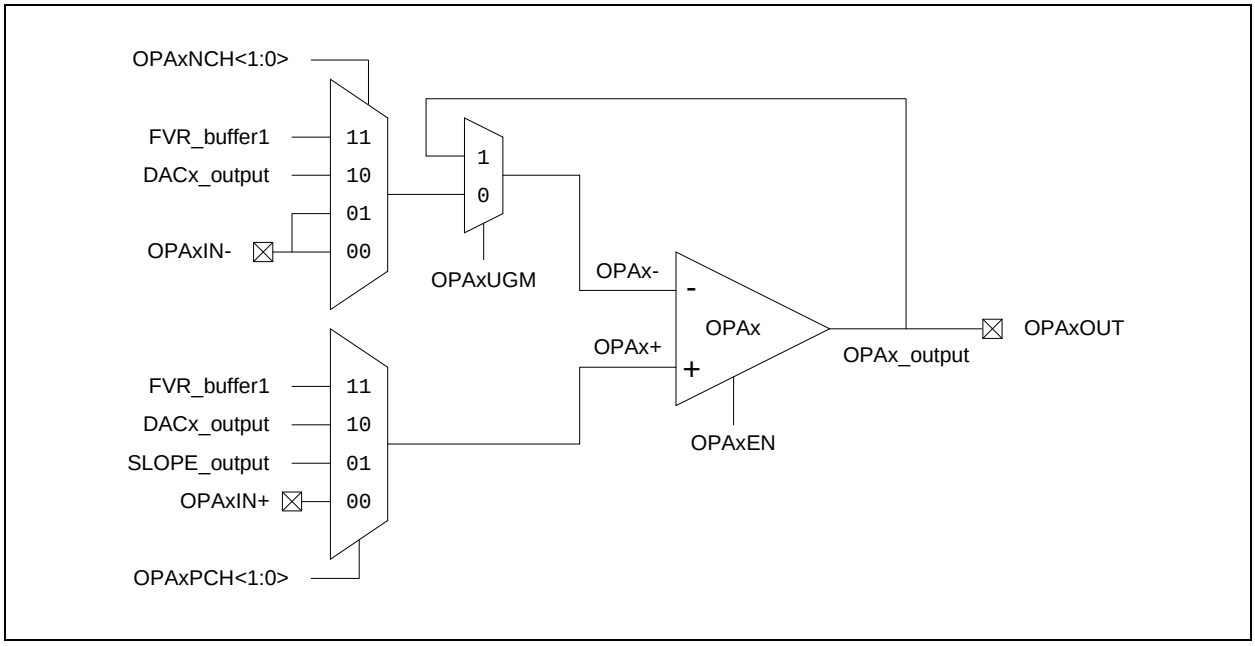
Common mode voltage range is the specified voltage range for the OPAx+ and OPAx- inputs, for which the OPA module will perform within its specifications. The OPA module is designed to operate with input voltages between $V_{SS}$ and $V_{DD}$. Behavior for Common mode voltages greater than $V_{DD}$ or below $V_{SS}$ is not guaranteed.

Leakage current is a measure of the small source or sink currents on the OPAx+ and OPAx- inputs. To minimize the effect of leakage currents, the effective impedances connected to the OPAx+ and OPAx- inputs should be kept as small as possible and equal. Input offset voltage is a measure of the voltage difference between the OPAx+ and OPAx- 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 (OPAx+) - (OPAx-). 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. The lower GBWP is optimized for systems requiring low-frequency response and low-power consumption.

**FIGURE 16-1: OPA MODULE BLOCK DIAGRAM**



## 16.4 Register Definitions: OPA Control

**REGISTER 16-1: OPAxCON0: OP AMP CONTROL REGISTER**

| R/W-0/0 | U-0 | U-0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| OPAxEN | — | — | OPAxUGM | OPAxNCH<1:0> | | OPAxPCH<1:0> | |
| bit 7 | | | | | | | bit 0 |

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

bit 7 **OPAxEN:** OPAx Enable bit
1 = OPAx is enabled
0 = OPAx is disabled

bit 6-5 **Unimplemented:** Read as '0'

bit 4 **OPAxUGM:** OPAx Unity Gain Mode Enable bit
1 = OPAx is in Unity gain mode
0 = OPAx is not in Unity gain mode - operates as a three-terminal op amp

bit 3-2 **OPAxNCH<1:0>:** OPAx Negative Input Source Selection bit
11 = OPAx- connects to FVR_buffer1
10 = OPAx- connects to DAC1_output
0x = OPAx- connects to OPAxIN- pin

bit 1-0 **OPAxPCH<1:0>:** OPAx Positive Input Source Selection bit
11 = OPAx+ connects to FVR_buffer1
10 = OPAx+ connects to DAC1_output
01 = OPAx+ connects to SLOPE_output
00 = OPAx+ connects to OPAxIN+ pin

**TABLE 16-1: REGISTERS ASSOCIATED WITH THE OPA MODULE**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|
| OPA1CON0 | OPA1EN | — | — | OPA1UGM | OPA1NCH<1:0> | | OPA1PCH<1:0> | | 134 |
| TRISC | — | — | TRISC5 | TRISC4 | TRISC3 | TRISC2 | TRISC1 | TRISC0 | 49 |
| ANSELC | — | — | — | — | ANSC3 | ANSC2 | ANSC1 | ANSC0 | 50 |

**Legend:** x = unknown, u = unchanged, - = unimplemented, read as '0'. Shaded cells are not used for the OPA module.

## 17.0 SLOPE COMPENSATION (SC) MODULE

The Slope Compensation (SC) module is designed to provide the necessary slope compensation for fixed frequency, continuous current, and current mode switched power supplies. Slope compensation is a necessary feature of these power supplies because it prevents frequency instabilities at duty cycles greater than 50%.

### 17.1 Theory of Operation

The SC module works by quickly discharging an internal capacitor at the beginning of each PWM period. An internal current sink charges this capacitor at a programmable rate. As the capacitor charges, the capacitor voltage is subtracted from the reference voltage, producing a linear voltage decay at the required rate. The current reference voltage can be supplied by either an I/O pin or by the buffered output of the FVR peripheral. The FVR module provides either a fixed voltage or a programmable DAC output. The Reset source can be derived from either the COG output or the synchronized output of either comparator. Additionally, the Reset source can be inverted before triggering the Reset. The slope voltage can be sent to either comparator or the op amp.

The core of the SC module is:

- an on-chip capacitor in series with the voltage source,
- a shorting switch across the capacitor, and
- a calibrated current sink.

A one-shot pulse generator ensures that the switch is closed long enough to completely discharge the capacitor. This typically takes 50 ns.

**FIGURE 17-1: SIMPLIFIED SC MODULE BLOCK DIAGRAM**

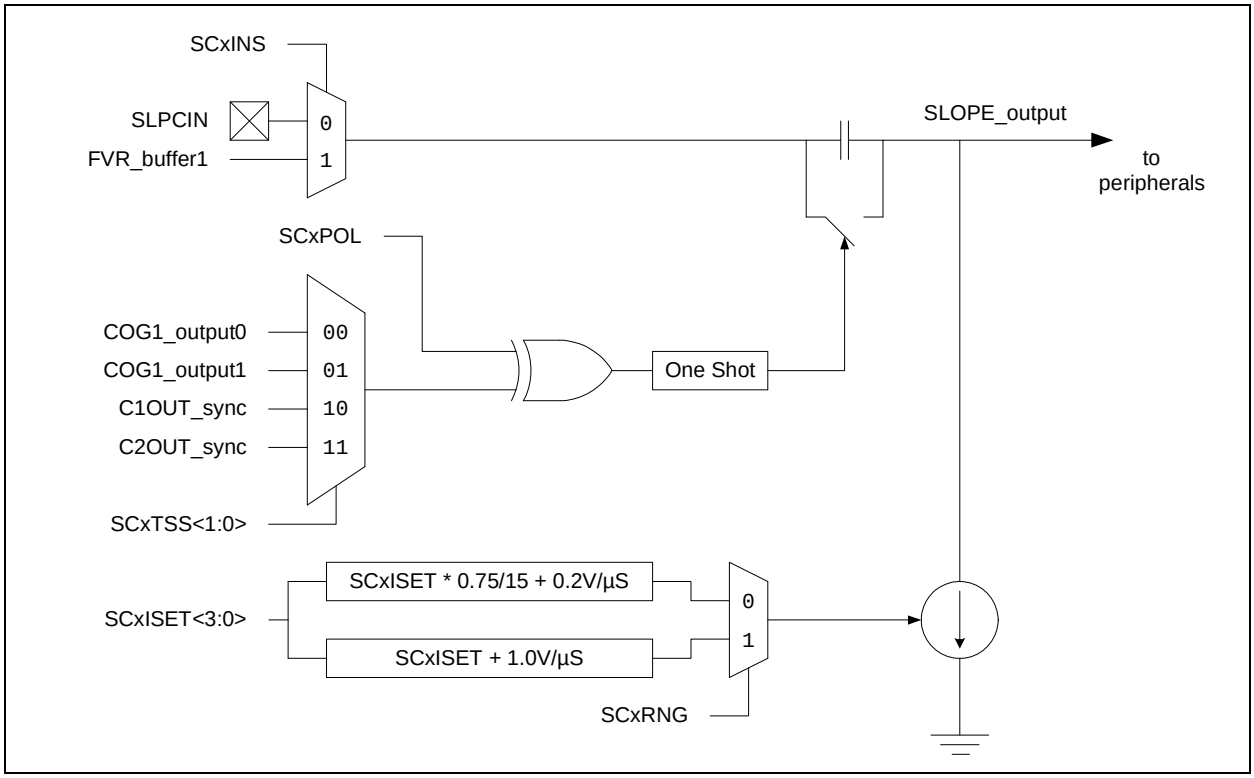
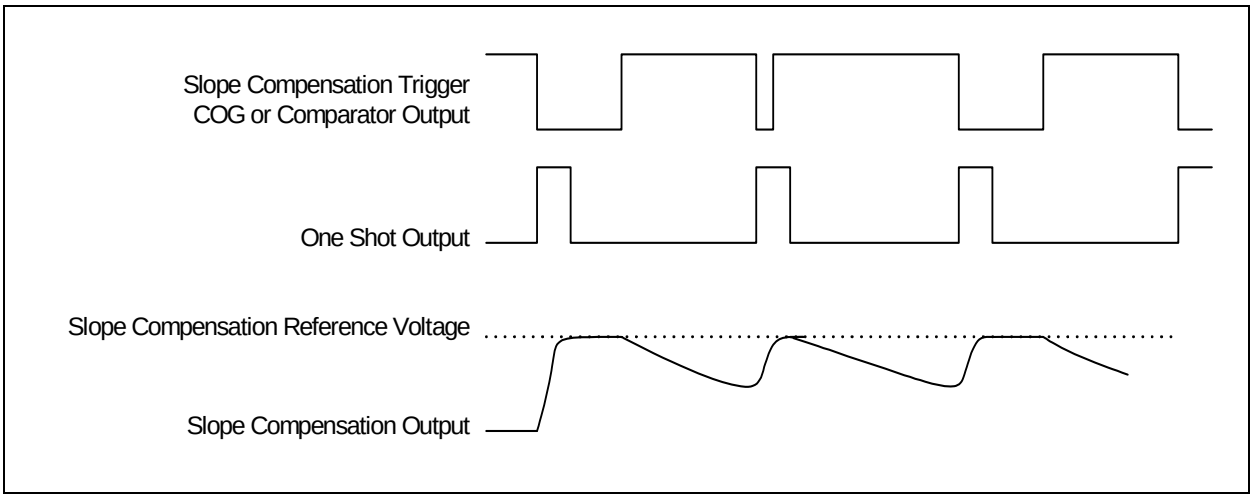


**FIGURE 17-2: SLOPE COMPENSATION TIMING DIAGRAM**




## 17.2 Using the SC Module

The slope compensator input reference voltage should be set to the target circuit peak current sense voltage. The slope compensator output voltage starts at the input reference voltage and should fall at a rate less than half the target circuit current sense voltage rate of rise. Therefore, the compensator slope expressed as volts per µs can be computed as shown in Equation 17-2.

**EQUATION 17-1: SC MODULE**

$$\frac{V}{\mu s} \geq \frac{\frac{V_{REF}}{2}}{PWM\ Period\ (\mu s)}$$

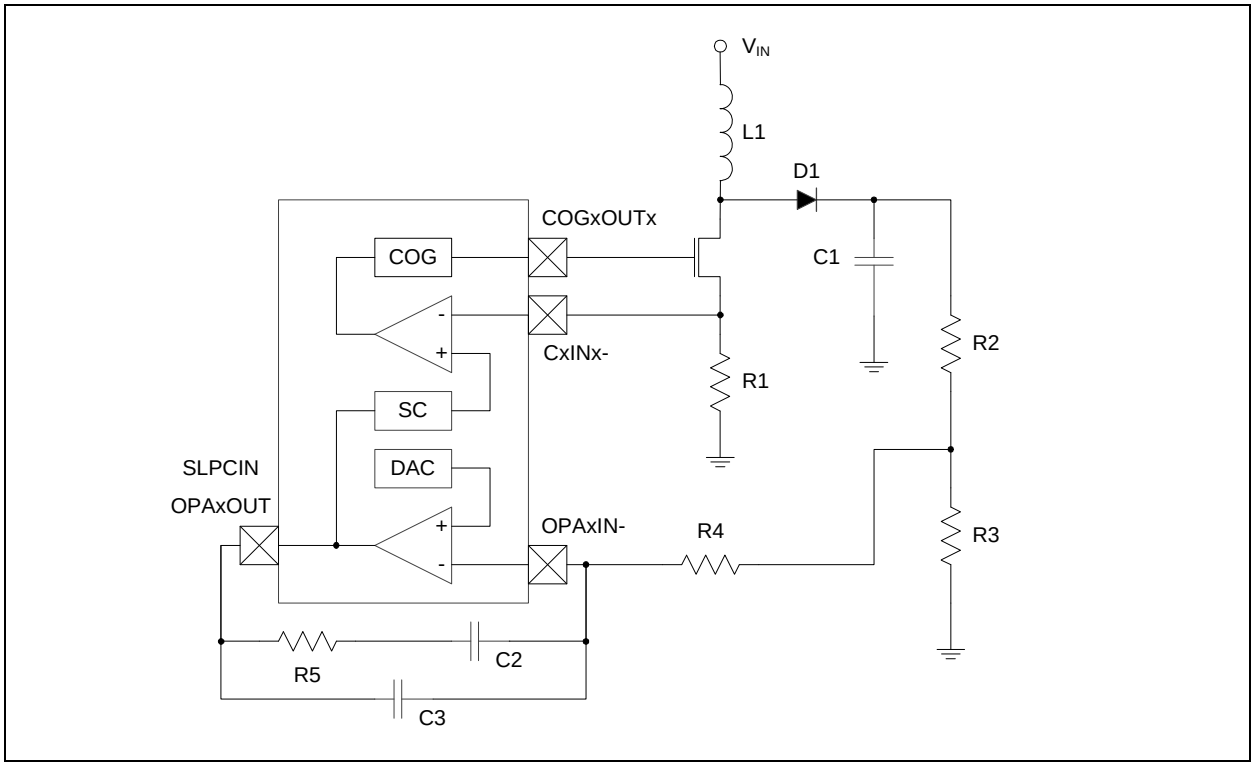
For example, when the circuit is using a 1Ω current sense resistor and the peak current is 1A, then the peak current expressed as a voltage ($V_{REF}$) is 1V. If your power supply is running at 1 MHz, then the period is 1 µs. Therefore, the desired slope is:

**EQUATION 17-2: SLOPE COMPENSATION VOLTAGE**

$$\frac{\frac{V_{REF}}{2}}{PWM\ Period\ (\mu s)} = \frac{\frac{1}{2}}{1\mu s} = 0.5V/\mu s$$

**Note**: The setting for 0.5V/µs is SCxISET<3:0> = 6 and SCxRNG = 0.

**FIGURE 17-3: EXAMPLE SLOPE COMPENSATION CIRCUIT**




## 17.3 Inputs

The SC module connects to the following inputs:

- COG1
- COG2
- Comparator C1
- Comparator C2

## 17.4 Outputs

The SC module connects to the following outputs:

- Comparator C1
- Comparator C2
- Op amp

## 17.5 Operation During Sleep

The SC module is unaffected by Sleep.

## 17.6 Effects of a Reset

The SC module resets to a disabled condition.

## 17.7 Register Definitions: Slope Compensation Control

**REGISTER 17-1: SLPCCON0: SLOPE COMPENSATION CONTROL 0 REGISTER**

| R/W-0/0 | U-0 | U-0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | U-0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| SCxEN | — | — | SCxPOL | SCxTSS<1:0> | | — | SCxINS |
| 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 **SCxEN:** Slope Compensation Enable bit
1 = Slope compensation is enabled
0 = Slope compensation is disabled

bit 6-5 **Unimplemented:** Read as ‘0’

bit 4 **SCxPOL:** Slope Compensation Input Polarity bit
1 = Signal is inverted polarity (active-low)
0 = Signal is normal polarity (active-high)

bit 3-2 **SCxTSS<1:0>:** Slope Compensation Timing Select bits
11 = C2OUT_sync
10 = C1OUT_sync
01 = COG1_output1
00 = COG1_output0

bit 1 **Unimplemented:** Read as ‘0’

bit 0 **SCxINS:** Slope Compensation Input Select bit
1 = FVR_buffer1 is selected
0 = SLPC1IN pin is selected

**REGISTER 17-2: SLPCCON1: SLOPE COMPENSATION CONTROL 1 REGISTER**

| U-0 | U-0 | U-0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| — | — | — | SCxRNG | SCxISET<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 | q = value depends on configuration bits |

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

bit 4 **SCxRNG:** Slope Compensator Range bit
1 = Range setting is SCxISET +1.0V/μs
0 = Range setting is SCxISET * 0.75/15 +0.2V/μs

bit 3-0 **SCxISET<3:0>:** Slope Compensator Current Sink Set bits
xxxxx = SC module Slope Selection

**TABLE 17-1: SLOPE COMPENSATOR CURRENT SETTINGS**

| SC1ISET Value | Current Setting (uA) | Slope Value (V/us) | SC1ISET Value | Current Setting (uA) | Slope Value (V/us) |
|---|---|---|---|---|---|
| 0h | 2 | 0.2 | 10h | 10 | 1.0 |
| 1h | 2.5 | 0.25 | 11h | 11 | 1.1 |
| 2h | 3 | 0.3 | 12h | 12 | 1.2 |
| 3h | 3.5 | 0.35 | 13h | 13 | 1.3 |
| 4h | 4 | 0.4 | 14h | 14 | 1.4 |
| 5h | 4.5 | 0.45 | 15h | 15 | 1.5 |
| 6h | 5 | 0.5 | 16h | 16 | 1.6 |
| 7h | 5.5 | 0.55 | 17h | 17 | 1.7 |
| 8h | 6 | 0.6 | 18h | 18 | 1.8 |
| 9h | 6.5 | 0.65 | 19h | 19 | 1.9 |
| Ah | 7 | 0.7 | 1Ah | 20 | 2.0 |
| Bh | 7.5 | 0.75 | 1Bh | 21 | 2.1 |
| Ch | 8 | 0.8 | 1Ch | 22 | 2.2 |
| Dh | 8.5 | 0.85 | 1Dh | 23 | 2.3 |
| Eh | 9 | 0.9 | 1Eh | 24 | 2.4 |
| Fh | 9.5 | 0.95 | 1Fh | 25 | 2.5 |

**TABLE 17-2: SUMMARY OF REGISTERS ASSOCIATED WITH THE SC MODULE**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|
| SLPCCON0 | SC1EN | — | — | SC1POL | SC1TSS<1:0> | | — | SC1INS | 138 |
| SLPCCON1 | — | — | — | SC1RNG | SC1ISET<3:0> | | | | 138 |
| PORTC | — | — | RC5 | RC4 | RC3 | RC2 | RC1 | RC0 | 49 |
| TRISC | — | — | TRISC5 | TRISC4 | TRISC3 | TRISC2 | TRISC1 | TRISC0 | 49 |
| ANSELC | — | — | — | — | ANSC3 | ANSC2 | ANSC1 | ANSC0 | 50 |
| WPUC | — | — | WPUC5 | WPUC4 | WPUC3 | WPUC2 | WPUC1 | WPUC0 | 51 |

**Legend:** — = unimplemented, read as '0'. Shaded cells are unused by the slope compensation module.

## 18.0 INSTRUCTION SET SUMMARY

The PIC16F753/HV753 instruction set is highly orthogonal and is comprised of three basic categories:

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

Each PIC16 instruction is a 14-bit word divided into an **opcode,** which specifies the instruction type and one or more **operands,** which further specify the operation of the instruction. The formats for each of the categories is presented in Figure 18-1, while the various opcode fields are summarized in Table 18-1.

Table 18-2 lists the instructions recognized by the MPASM™ assembler.

For **byte-oriented** instructions, 'f' represents a file register designator and 'd' represents a destination designator. The file register designator specifies which file register is to be used by the instruction.

The destination designator specifies where the result of the operation is to be placed. If 'd' is zero, the result is placed in the W register. If 'd' is one, the result is placed in the file register specified in the instruction.

For **bit-oriented** instructions, 'b' represents a bit field designator, which selects the bit affected by the operation, while 'f' represents the address of the file in which the bit is located.

For **literal and control** operations, 'k' represents an 8-bit or 11-bit constant, or literal value.

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

All instruction examples use the format '0xhh' to represent a hexadecimal number, where 'h' signifies a hexadecimal digit.

### 18.1 Read-Modify-Write Operations

Any instruction that specifies a file register as part of the instruction performs a Read-Modify-Write (RMW) operation. The register is read, the data is modified, and the result is stored according to either the instruction or the destination designator 'd'. A read operation is performed on a register even if the instruction writes to that register.

For example, a CLRF PORTA instruction will read PORTA, clear all the data bits, then write the result back to PORTA. This example would have the unintended consequence of clearing the condition that set the IOCIF flag.

**TABLE 18-1: OPCODE FIELD DESCRIPTIONS**

| Field | Description |
|---|---|
| f | Register file address (0x00 to 0x7F) |
| W | Working register (accumulator) |
| b | Bit address within an 8-bit file register |
| k | Literal field, constant data or label |
| x | Don't care location (= 0 or 1).<br>The assembler will generate code with x = 0.<br>It is the recommended form of use for compatibility with all Microchip software tools. |
| d | Destination select; d = 0: store result in W,<br>d = 1: store result in file register f.<br>Default is d = 1. |
| PC | Program Counter |
| $\overline{TO}$ | Time-out bit |
| C | Carry bit |
| DC | Digit carry bit |
| Z | Zero bit |
| $\overline{PD}$ | Power-down bit |

**FIGURE 18-1: GENERAL FORMAT FOR INSTRUCTIONS**

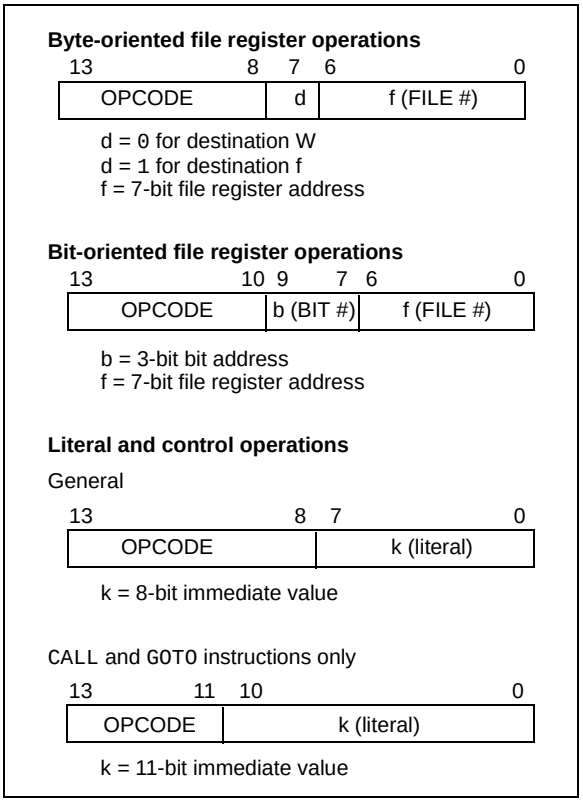


**TABLE 18-2: PIC16F753/HV753 INSTRUCTION SET**

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

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

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

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

## 18.2 Instruction Descriptions

| ADDLW | Add literal and W |
|---|---|
| Syntax: | [ *label* ] ADDLW k |
| Operands: | 0 ≤ k ≤ 255 |
| Operation: | (W) + k → (W) |
| Status Affected: | C, DC, Z |
| Description: | The contents of the W register are added to the 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'. |

| 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'. |

| BCF | Bit Clear f |
|---|---|
| Syntax: | [ *label* ] BCF f,b |
| Operands: | 0 ≤ f ≤ 127<br>0 ≤ b ≤ 7 |
| Operation: | 0 → (f<b>) |
| Status Affected: | None |
| Description: | Bit 'b' in register 'f' is cleared. |

| BSF | Bit Set f |
|---|---|
| Syntax: | [ *label* ] BSF f,b |
| Operands: | 0 ≤ f ≤ 127<br>0 ≤ b ≤ 7 |
| Operation: | 1 → (f<b>) |
| Status Affected: | None |
| Description: | Bit 'b' in register 'f' is set. |

| BTFSC | Bit Test f, Skip if Clear |
|---|---|
| Syntax: | [ *label* ] BTFSC f,b |
| Operands: | 0 ≤ f ≤ 127<br>0 ≤ b ≤ 7 |
| Operation: | skip if (f<b>) = 0 |
| Status Affected: | None |
| Description: | If bit 'b' in register 'f' is '1', the next instruction is executed.<br>If bit 'b' in register 'f' is '0', the next instruction is discarded, and a NOP is executed instead, making this a 2-cycle instruction. |

**BTFSS** **Bit Test f, Skip if Set**

| | |
|---|---|
| Syntax: | [ *label* ] BTFSS f,b |
| Operands: | $0 \le f \le 127$<br>$0 \le b < 7$ |
| 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. |

**CALL** **Call Subroutine**

| | |
|---|---|
| Syntax: | [ *label* ] CALL k |
| Operands: | $0 \le k \le 2047$ |
| Operation: | (PC)+ 1→ TOS,<br>k → PC<10:0>,<br>(PCLATH<4:3>) → PC<12:11> |
| Status Affected: | None |
| Description: | Call Subroutine. First, return address (PC + 1) is pushed onto the stack. The 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. |

**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.<br>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: | $(\bar{f}) \rightarrow$ (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<4:3> → PC<12:11> |
| 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'. |

| MOVF | Move f |
|---|---|
| Syntax: | [ *label* ] MOVF f,d |
| Operands: | $0 \le f \le 127$<br>$d \in [0,1]$ |
| 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<br>After Instruction<br>W = value in FSR register<br>Z = 1 |

| MOVWF | Move W to f |
|---|---|
| Syntax: | [ *label* ] MOVWF f |
| Operands: | $0 \le f \le 127$ |
| Operation: | (W) → (f) |
| Status Affected: | None |
| Description: | Move data from W register to register 'f'. |
| Words: | 1 |
| Cycles: | 1 |
| Example: | MOVW OPTION<br>F<br>Before Instruction<br>OPTION = 0xFF<br>W = 0x4F<br>After Instruction<br>OPTION = 0x4F<br>W = 0x4F |

| MOVLW | Move literal to W |
|---|---|
| Syntax: | [ *label* ] MOVLW k |
| Operands: | $0 \le k \le 255$ |
| Operation: | k → (W) |
| Status Affected: | None |
| Description: | The 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<br>After Instruction<br>W = 0x5A |

| NOP | No Operation |
|---|---|
| Syntax: | [ *label* ] NOP |
| Operands: | None |
| Operation: | No operation |
| Status Affected: | None |
| Description: | No operation. |
| Words: | 1 |
| Cycles: | 1 |
| Example: | NOP |

| RETFIE | Return from Interrupt |
|---|---|
| Syntax: | [ *label* ] RETFIE |
| Operands: | None |
| Operation: | TOS → PC,<br>1 → GIE |
| Status Affected: | None |
| Description: | Return from Interrupt. Stack is POPed and Top-of-Stack (TOS) is loaded in the PC. Interrupts are enabled by setting Global Interrupt Enable bit, GIE (INTCON<7>). This is a 2-cycle instruction. |
| Words: | 1 |
| Cycles: | 2 |
| Example: | `RETFIE`<br>After Interrupt<br>PC = TOS<br>GIE = 1 |

| RETLW | Return with literal in W |
|---|---|
| Syntax: | [ *label* ] RETLW k |
| Operands: | 0 ≤ k ≤ 255 |
| Operation: | k → (W);<br>TOS → PC |
| Status Affected: | None |
| Description: | The W register is loaded with the 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`<br>`;table offset`<br>`;value`<br>`GOTO DONE`<br>`TABLE`<br>•<br>•<br>`ADDWF PC ;W = offset`<br>`RETLW k1 ;Begin table`<br>`RETLW k2 ;`<br>•<br>•<br>•<br>`RETLW kn ;End of table`<br>`DONE`<br>Before Instruction<br>W = 0x07<br>After Instruction<br>W = value of k8 |

| RETURN | Return from Subroutine |
|---|---|
| Syntax: | [ *label* ] RETURN |
| Operands: | None |
| Operation: | TOS → PC |
| Status Affected: | None |
| Description: | Return from subroutine. The stack is POPed and the top of the stack (TOS) is loaded into the program counter. This is a 2-cycle instruction. |

**RLF** **Rotate Left f through Carry**

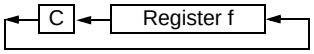
Syntax: [ *label* ] RLF f,d

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

Operation: See description below

Status Affected: C

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




Words: 1

Cycles: 1

Example:

```
RLF     REG1,0
```

Before Instruction

```
REG1    =   1110 0110
C       =   0
```

After Instruction

```
REG1    =   1110 0110
W       =   1100 1100
C       =   1
```

**RRF** **Rotate Right f through Carry**

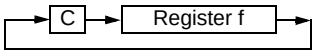
Syntax: [ *label* ] RRF f,d

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

Operation: See description below

Status Affected: C

Description: The contents of register 'f' are rotated 1 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.

| Result | Condition |
|---|---|
| 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> |

**SWAPF** **Swap Nibbles in f**

Syntax: [ *label* ] SWAPF f,d

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

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

Status Affected: None

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

**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: $0 \le f \le 127$
$d \in [0,1]$

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'.

## 19.0 SPECIAL FEATURES OF THE CPU

The PIC16F753/HV753 has a host of features intended to maximize system reliability, minimize cost through elimination of external components, provide power-saving features and offer code protection.

These features are:

- Reset
  - Power-on Reset (POR)
  - Power-up Timer (PWRT)
  - Brown-out Reset (BOR)
- Interrupts
- Watchdog Timer (WDT)
- Oscillator selection
- Sleep
- Code protection
- ID Locations
- In-Circuit Serial Programming™

The Power-up Timer (PWRT), which provides a fixed delay of 64 ms (nominal) on power-up only, is designed to keep the part in Reset while the power supply stabilizes. There is also circuitry to reset the device if a brown-out occurs, which can use the Power-up Timer to provide at least a 64 ms Reset. With these functions-on-chip, most applications need no external Reset circuitry.

The Sleep mode is designed to offer a very low-current Power-Down mode. The user can wake up from Sleep through:

- External Reset
- Watchdog Timer Wake-up
- An interrupt

Oscillator selection options are available to allow the part to fit the application. The INTOSC options save system cost, while the External Clock (EC) option provides a means for specific frequency and accurate clock sources. Configuration bits are used to select various options (see Register 19-1).

### 19.1 Configuration Bits

The Configuration bits can be programmed (read as '0'), or left unprogrammed (read as '1') to select various device configurations as shown in Register 19-1. These bits are mapped in program memory location 2007h.

| | |
|---|---|
| **Note:** | Address 2007h is beyond the user program memory space. It belongs to the special configuration memory space (2000h-3FFFh), which can be accessed only during programming. See the *PIC16F753/HV753 Flash Memory Programming Specification* (DS41686) for more information. |

**REGISTER 19-1: CONFIGURATION WORD**

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

| U-1 | R/P-1 | R/P-1 | R/P-1 | R/P-1 | U-1 | U-1 | R/P-1 |
|---|---|---|---|---|---|---|---|
| — | $\overline{\text{CP}}$ | MCLRE | $\overline{\text{PWRTE}}$ | WDTE | — | — | FOSC0 |
| 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 **$\overline{\text{DEBUG}}$:** Debug Mode Enable bit(2)
1 = Background debugger is disabled
0 = Background debugger is enabled

bit 12 **$\overline{\text{CLKOUTEN}}$:** Clock Out Enable bit
1 = Clock out function disabled. CLKOUT pin acts as I/O pin
0 = General purpose I/O disabled. CLKOUT pin acts as CLKOUT

bit 11-10 **WRT<1:0>:** Flash Program Memory Self-Write Enable bit
11 = Write protection off
10 = 000h to FFh write-protected, 100h to 3FFh may be modified by PMCON1 control
01 = 000h to 1FFh write-protected, 200h to 3FFh may be modified by PMCON1 control
00 = 000h to 3FFh write-protected, entire program is write-protected

bit 8-9 **BOREN<1:0>:** Brown-out Reset Enable bits
11 = BOR enabled
10 = BOR enabled during operation and disabled in Sleep
0x = BOR disabled

bit 7 **Unimplemented:** Read as ‘1’

bit 6 **$\overline{\text{CP}}$:** Code Protection bit
1 = Program memory code protection is disabled
0 = Program memory code protection is enabled

bit 5 **MCLRE:** $\overline{\text{MCLR}}$/$V_{PP}$ Pin Function Select bit
1 = $\overline{\text{MCLR}}$ pin is $\overline{\text{MCLR}}$ function and weak internal pull-up is enabled
0 = $\overline{\text{MCLR}}$ pin is input function, $\overline{\text{MCLR}}$ function is internally disabled

bit 4 **$\overline{\text{PWRTE}}$:** Power-up Timer Enable bit(1)
1 = PWRT disabled
0 = PWRT enabled

bit 3 **WDTE:** Watchdog Timer Enable bit
1 = WDT enabled
0 = WDT disabled

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

bit 0 **FOSC:** Oscillator Selection bits
1 = EC oscillator selected: CLKIN on RA5/CLKIN
0 = Internal oscillator: I/O function on RA5/CLKIN

**Note 1:** Enabling Brown-out Reset does not automatically enable Power-up Timer.
**2:** The Configuration bit is managed automatically by the device development tools. The user should not attempt to manually write this bit location. However, the user should ensure that this location has been programmed to a ‘1’ and the device checksum is correct for proper operation of production software.

## 19.2 Calibration Bits

The 8 MHz internal oscillator is factory-calibrated. These calibration values are stored in fuses located in the Calibration Word (2008h). The Calibration Word is not erased when using the specified bulk erase sequence in the *PIC16F753/HV753 Flash Memory Programming Specification* (DS41686) and thus, does not require reprogramming.

## 19.3 Reset

The PIC16F753/HV753 device differentiates between various kinds of Reset:

a) Power-on Reset (POR)
b) WDT Reset during normal operation
c) WDT Reset during Sleep
d) $\overline{\text{MCLR}}$ Reset during normal operation
e) $\overline{\text{MCLR}}$ Reset during Sleep
f) Brown-out Reset (BOR)

Some registers are not affected in any Reset condition; their status is unknown on POR and unchanged in any other Reset. Most other registers are reset to a "Reset state" on:

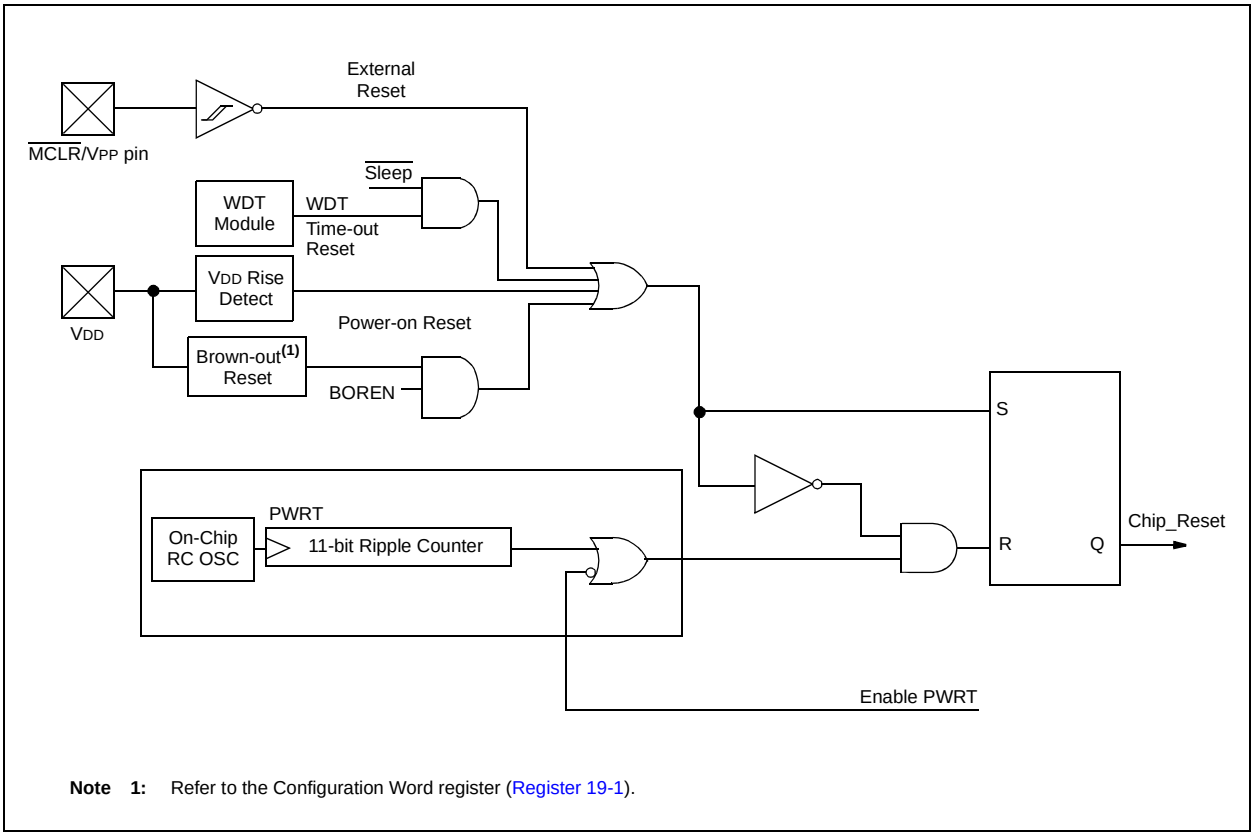
- Power-on Reset
- $\overline{\text{MCLR}}$ Reset
- $\overline{\text{MCLR}}$ Reset during Sleep
- WDT Reset
- Brown-out Reset (BOR)

WDT wake-up does not cause register Resets in the same manner as a WDT Reset since wake-up is viewed as the resumption of normal operation. $\overline{\text{TO}}$ and $\overline{\text{PD}}$ bits are set or cleared differently in different Reset situations, as indicated in Table 19-2. Software can use these bits to determine the nature of the Reset. See Table 19-4 for a full description of Reset states of all registers.

A simplified block diagram of the On-Chip Reset Circuit is shown in Figure 19-1.

The $\overline{\text{MCLR}}$ Reset path has a noise filter to detect and ignore small pulses. See **Section 22.0 "Electrical Specifications"** for pulse-width specifications.

**FIGURE 19-1: SIMPLIFIED BLOCK DIAGRAM OF ON-CHIP RESET CIRCUIT**




**Note 1:** Refer to the Configuration Word register (Register 19-1).

**TABLE 19-1: TIME-OUT IN VARIOUS SITUATIONS**

| Oscillator Configuration | Power-up | | Brown-out Reset | | Wake-up from Sleep |
|---|---|---|---|---|---|
| | $\overline{PWRTE}$ = 0 | $\overline{PWRTE}$ = 1 | $\overline{PWRTE}$ = 0 | $\overline{PWRTE}$ = 1 | |
| EC, INTOSC | TPWRT | — | TPWRT | — | — |

**TABLE 19-2: STATUS/PCON BITS AND THEIR SIGNIFICANCE**

| $\overline{POR}$ | $\overline{BOR}$ | $\overline{TO}$ | $\overline{PD}$ | Condition |
|---|---|---|---|---|
| 0 | x | 1 | 1 | Power-on Reset |
| u | 0 | 1 | 1 | Brown-out Reset |
| u | u | 0 | u | WDT Reset |
| u | u | 0 | 0 | WDT Wake-up |
| u | u | u | u | $\overline{MCLR}$ Reset during normal operation |
| u | u | 1 | 0 | $\overline{MCLR}$ Reset during Sleep |

**Legend:** u = unchanged, x = unknown

### 19.3.1 POWER-ON RESET (POR)

The on-chip POR circuit holds the chip in Reset until VDD has reached a high enough level for proper operation. To take advantage of the POR, simply connect the $\overline{MCLR}$ pin through a resistor to VDD. This will eliminate external RC components usually needed to create Power-on Reset. A maximum rise time for VDD is required. See **Section 22.0 "Electrical Specifications"** for details. If the BOR is enabled, the maximum rise time specification does not apply. The BOR circuitry will keep the device in Reset until VDD reaches VBOR (see **Section 19.3.4 "Brown-out Reset (BOR)"**).

| **Note:** | The POR circuit does not produce an internal Reset when VDD declines. To re-enable the POR, VDD must reach Vss for a minimum of 100 µs. |
|---|---|

When the device starts normal operation (exits the Reset condition), device operating parameters (i.e., voltage, frequency, temperature, etc.) must be met to ensure proper operation. If these conditions are not met, the device must be held in Reset until the operating conditions are met.

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

### 19.3.2 $\overline{MCLR}$

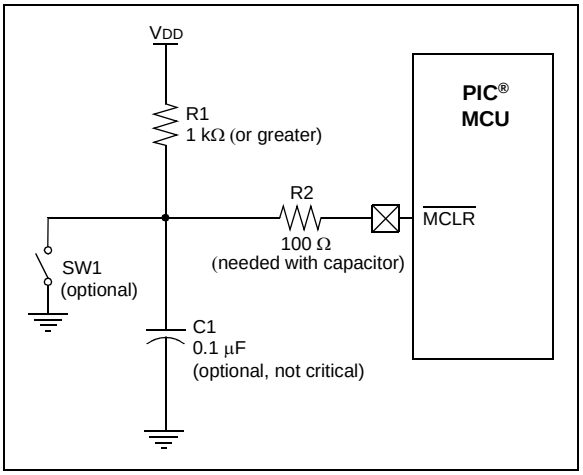
PIC16F753/HV753 has a noise filter in the $\overline{MCLR}$ Reset path. The filter will detect and ignore small pulses.

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

Voltages applied to the $\overline{MCLR}$ pin that exceed its specification can result in both $\overline{MCLR}$ Resets and excessive current beyond the device specification during the ESD event. For this reason, Microchip recommends that the $\overline{MCLR}$ pin no longer be tied directly to VDD. The use of an RC network, as shown in Figure 19-2, is suggested.

An internal $\overline{MCLR}$ option is enabled by clearing the MCLRE bit in the Configuration Word register. When MCLRE = 0, the Reset signal to the chip is generated internally. When the MCLRE = 1, the $\overline{MCLR}$ pin becomes an external Reset input. In this mode, the $\overline{MCLR}$ pin has a weak pull-up to VDD.

**FIGURE 19-2: RECOMMENDED $\overline{MCLR}$ CIRCUIT**



### 19.3.3 POWER-UP TIMER (PWRT)

The Power-up Timer provides a fixed 64 ms (nominal) time-out on power-up only, from POR or Brown-out Reset. The Power-up Timer operates from an internal RC oscillator. For more information, see **Section 4.2.2 "Internal Clock Mode"**. The chip is kept in Reset as long as PWRT is active. The PWRT delay allows the VDD to rise to an acceptable level. A Configuration bit, $\overline{\text{PWRTE}}$, can disable (if set) or enable (if cleared or programmed) the Power-up Timer. The Power-up Timer should be enabled when Brown-out Reset is enabled, although it is not required.

The Power-up Timer delay will vary from chip-to-chip due to:

- VDD variation
- Temperature variation
- Process variation

See DC parameters for details (**Section 22.0 "Electrical Specifications"**).

| | |
|---|---|
| **Note:** | Voltage spikes below VSS at the $\overline{\text{MCLR}}$ pin, inducing currents greater than 80 mA, may cause latch-up. Thus, a series resistor of 50-100Ω should be used when applying a "low" level to the $\overline{\text{MCLR}}$ pin, rather than pulling this pin directly to VSS. |

### 19.3.4 BROWN-OUT RESET (BOR)

The BOREN<1:0> bits in the Configuration Word register select one of three BOR modes. One mode has been added to allow control of the BOR enable for lower current during Sleep. By selecting BOREN<1:0> = 10, the BOR is automatically disabled in Sleep to conserve power and enabled on wake-up. See Register 19-1 for the Configuration Word definition.

A brown-out occurs when VDD falls below VBOR for greater than parameter TBOR (see **Section 22.0 "Electrical Specifications"**). The brown-out condition will reset the device. This will occur regardless of VDD slew rate. A Brown-out Reset may not occur if VDD falls below VBOR for less than parameter TBOR.

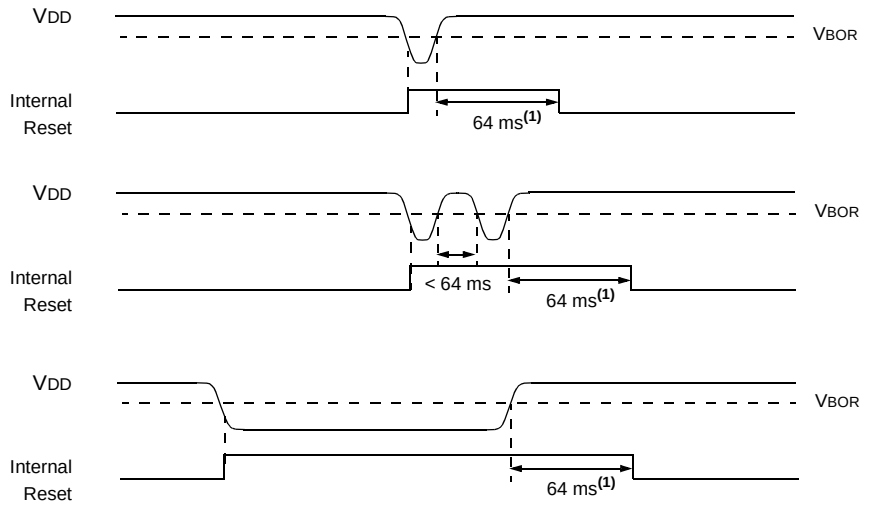
On any Reset (Power-on, Brown-out Reset, Watchdog timer, etc.), the chip will remain in Reset until VDD rises above VBOR (see Figure 19-3). If enabled, the Power-up Timer will be invoked by the Reset and keep the chip in Reset an additional 64 ms.

| | |
|---|---|
| **Note:** | The Power-up Timer is enabled by the $\overline{\text{PWRTE}}$ bit in the Configuration Word register. |

If VDD drops below VBOR while the Power-up Timer is running, the chip will go back into a Brown-out Reset and the Power-up Timer will be re-initialized. Once VDD rises above VBOR, the Power-up Timer will execute a 64 ms Reset.

Table 19-2 summarizes the registers associated with BOR.

**FIGURE 19-3: BROWN-OUT SITUATIONS**




**Note 1:** 64 ms delay only if $\overline{\text{PWRTE}}$ bit is programmed to '0'.

**TABLE 19-3: SUMMARY OF REGISTERS ASSOCIATED WITH BROWN-OUT RESET**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|
| PCON | — | — | — | — | — | — | $\overline{POR}$ | $\overline{BOR}$ | 22 |
| STATUS | IRP | RP1 | RP0 | $\overline{TO}$ | $\overline{PD}$ | Z | DC | C | 15 |

**Legend:** u = unchanged, x = unknown, – = unimplemented bit, reads as '0', q = value depends on condition. Shaded cells are not used by BOR.

### 19.3.5 TIME-OUT SEQUENCE

On power-up, the time-out sequence is as follows:

- PWRT time-out is invoked after POR has expired.
- OST is activated after the PWRT time-out has expired.

The total time-out will vary based on oscillator configuration and $\overline{PWRTE}$ bit status. For example, in EC mode with $\overline{PWRTE}$ bit erased (PWRT disabled), there will be no time-out at all. Figure 19-4, Figure 19-5 and Figure 19-6 depict time-out sequences.

Since the time-outs occur from the POR pulse, if $\overline{MCLR}$ is kept low long enough, the time-outs will expire. Then, bringing $\overline{MCLR}$ high will begin execution immediately (see Figure 19-5). This is useful for testing purposes or to synchronize more than one PIC16F753/HV753 device operating in parallel.

Table shows the Reset conditions for some special registers, while Table 19-4 shows the Reset conditions for all the registers.

### 19.3.6 POWER CONTROL (PCON) REGISTER

The Power Control register PCON (address 8Eh) has two Status bits to indicate what type of Reset occurred last.

Bit 0 is $\overline{BOR}$ (Brown-out). $\overline{BOR}$ is unknown on Power-on Reset. It must then be set by the user and checked on subsequent Resets to see if $\overline{BOR}$ = 0, indicating that a Brown-out has occurred. The $\overline{BOR}$ Status bit is a "don't care" and is not necessarily predictable if the brown-out circuit is disabled (BOREN<1:0> = 00 in the Configuration Word register).

Bit 1 is $\overline{POR}$ (Power-on Reset). It is a '0' on Power-on Reset and unaffected otherwise. The user must write a '1' to this bit following a Power-on Reset. On a subsequent Reset, if $\overline{POR}$ is '0', it will indicate that a Power-on Reset has occurred (i.e., $V_{DD}$ may have gone too low).

For more information, see **Section 19.3.4 "Brown-out Reset (BOR)"**.

**FIGURE 19-4: TIME-OUT SEQUENCE ON POWER-UP (DELAYED $\overline{MCLR}$): CASE 1**

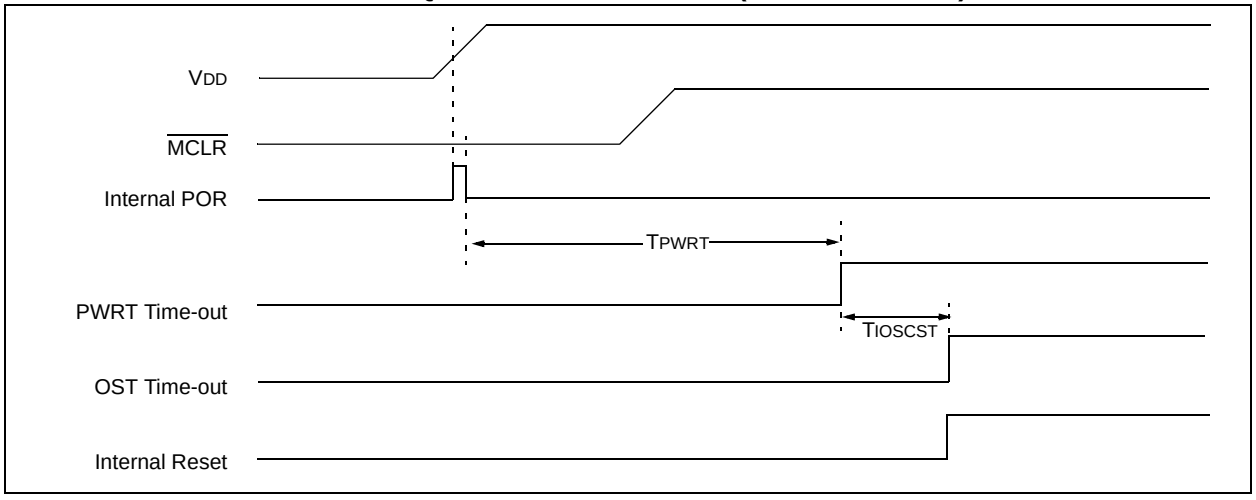


**FIGURE 19-5: TIME-OUT SEQUENCE ON POWER-UP (DELAYED $\overline{MCLR}$): CASE 2**

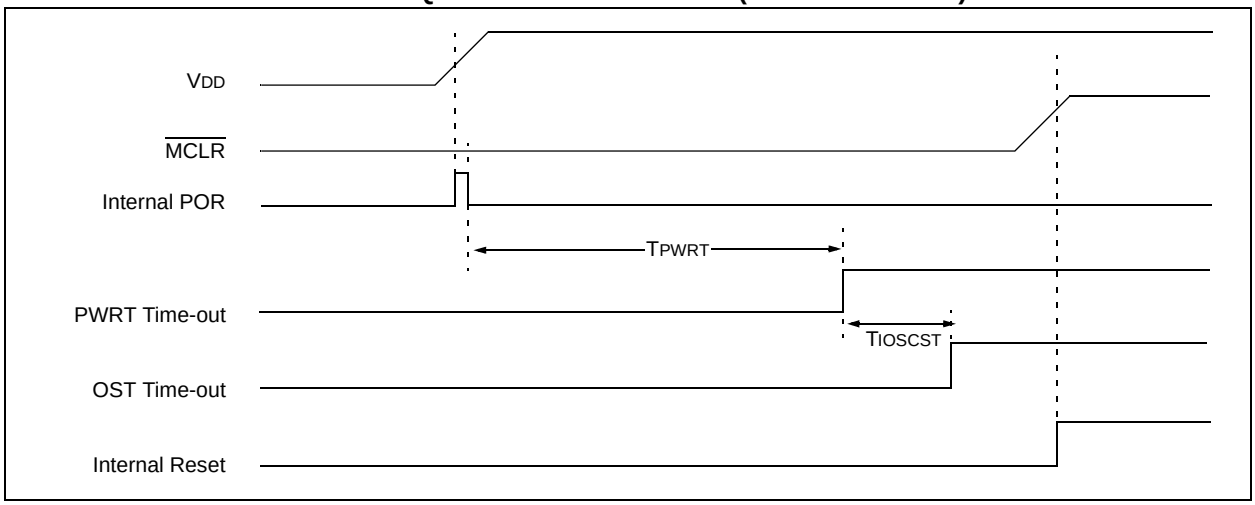



**FIGURE 19-6: TIME-OUT SEQUENCE ON POWER-UP ($\overline{MCLR}$ WITH VDD)**

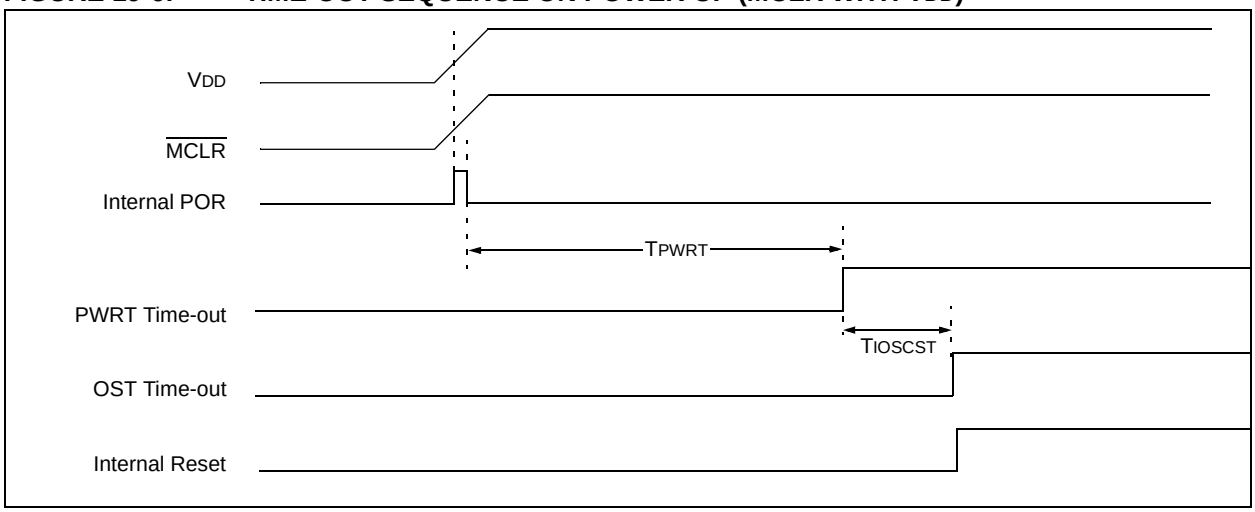


**TABLE 19-4: INITIALIZATION CONDITION FOR REGISTERS**

| Register | Address | Power-on Reset | $\overline{\text{MCLR}}$ Reset<br>WDT Reset<br>Brown-out Reset(1) | Wake-up from Sleep through Interrupt<br>Wake-up from Sleep through WDT Time-out |
|---|---|---|---|---|
| W | — | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| INDF | 00h/80h/<br>100h/180h | xxxx xxxx | xxxx xxxx | uuuu uuuu |
| TMR0 | 01h | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| PCL | 02h/82h/<br>102h/182h | 0000 0000 | 0000 0000 | PC + 1(3) |
| STATUS | 03h/83h/<br>103h/183h | 0001 1xxx | 000q quuu(4) | uuuq quuu(4) |
| FSR | 04h/84h/<br>104h/184h | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| PORTA | 05h | --xx xxxx | --uu uuuu | --uu uuuu |
| IOCAF | 08h | --00 0000 | --00 0000 | --uu uuuu |
| PCLATH | 0Ah/8Ah/<br>10Ah/18Ah | ---0 0000 | ---0 0000 | ---u uuuu |
| INTCON | 0Bh/8Bh/<br>10Bh/18Bh | 0000 0000 | 0000 0000 | uuuu uuuu(2) |
| PIR1 | 0Ch | 00-- -0-0 | 00-- -0-0 | uu-- -u-u(2) |
| PIR2 | 0Dh | --00 -0-0 | --00 -0-0 | --uu -u-u(2) |
| TMR1L | 0Fh | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| TMR1H | 10h | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| T1CON | 11h | 0000 00-0 | uuuu uu-u | uuuu uu-u |
| T1GCON | 12h | 0000 0x00 | 0000 0x00 | uuuu uuuu |
| CCPR1L(1) | 13h | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| CCPR1H(1) | 14h | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| CCP1CON(1) | 15h | --00 0000 | --00 0000 | --uu uuuu |
| ADRESL(1) | 1Ch | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| ADRESH(1) | 1Dh | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| ADCON0(1) | 1Eh | 0000 0000 | 0000 0000 | uuuu uuuu |
| ADCON1(1) | 1Fh | -000 ---- | -000 ---- | -uuu ---- |
| OPTION_REG | 81h/181h | 1111 1111 | 1111 1111 | uuuu uuuu |
| TRISA | 85h | --11 1111 | --11 1111 | --uu uuuu |
| IOCAP | 88h | --00 0000 | --00 0000 | --uu uuuu |
| PIE1 | 8Ch | 00-- -000 | 00-- -000 | uu-- -uuu |
| PIE2 | 8Dh | --00 -0-0 | --00 -0-0 | --uu -u-u |
| OSCCON | 8Fh | --01 -00- | --uu -uu- | --uu -uu- |
| FVRCON | 90h | 0000 ---- | 0000 ---- | uuuu ---- |
| DACCON0 | 91h | 000- -0-- | 000- -0-- | uuu- -u-- |
| DACCON1 | 92h | ---0 0000 | ---0 0000 | ---u uuuu |
| CM2CON0 | 9Bh | 0000 0100 | 0000 0100 | uuuu uuuu |
| CM2CON1 | 9Ch | 0000 ---0 | 0000 ---0 | uuuu ---u |

**Legend:** u = unchanged, x = unknown, – = unimplemented bit, reads as '0', q = value depends on condition.

**Note 1:** If $V_{DD}$ goes too low, Power-on Reset will be activated and registers will be affected differently.

**2:** One or more bits in INTCON and/or PIRx will be affected (to cause wake-up).

**3:** When the wake-up is due to an interrupt and the GIE bit is set, the PC is loaded with the interrupt vector (0004h).

**4:** See Table 19-5 for Reset value for specific condition.

**5:** If Reset was due to brown-out, then bit 0 = 0. All other Resets will cause bit 0 = u.

**TABLE 19-4: INITIALIZATION CONDITION FOR REGISTERS (CONTINUED)**

| Register | Address | Power-on Reset | $\overline{\text{MCLR}}$ Reset<br>WDT Reset<br>Brown-out Reset(1) | Wake-up from Sleep through Interrupt<br>Wake-up from Sleep through WDT Time-out |
|---|---|---|---|---|
| CM1CON0 | 9Dh | 0000 0100 | 0000 0100 | uuuu uuuu |
| CM1CON1 | 9Eh | 0000 ---0 | 0000 ---0 | uuuu ---u |
| CMOUT | 9Fh | ---- --00 | ---- --00 | ---- --uu |
| LATA | 105h | --xx -xxx | --uu -uuu | --uu -uuu |
| IOCAN | 108h | --00 0000 | --00 0000 | --uu uuuu |
| WPUA | 10Ch | --00 0000 | --00 0000 | --uu uuuu |
| SLRCON0 | 10Dh | ---- -0-0 | ---- -0-0 | ---- -u-u |
| PCON | 10Fh | ---- --qq | ---- --uu(1, 5) | ---- --uu |
| TMR2 | 110h | 0000 0000 | 0000 0000 | uuuu uuuu |
| PR2 | 111h | 1111 1111 | 1111 1111 | uuuu uuuu |
| T2CON | 112h | -000 0000 | -000 0000 | -uuu uuuu |
| HLTMR1 | 113h | 0000 0000 | 0000 0000 | uuuu uuuu |
| HLTPR1 | 114h | 1111 1111 | 1111 1111 | uuuu uuuu |
| HLT1CON0 | 115h | -000 0000 | -000 0000 | -uuu uuuu |
| HLT1CON1 | 116h | ---0 0000 | ---0 0000 | ---u uuuu |
| ANSELA | 185h | --11 -111 | --11 -111 | --uu -uuu |
| APFCON | 188h | ---0 -000 | ---0 -000 | ---u -uuu |
| OSCTUNE | 189h | ---0 0000 | ---u uuuu | ---u uuuu |
| PMCON1 | 18Ch | ---- -000 | ---- -000 | ---- -uuu |
| PMCON2 | 18Dh | ---- ---- | ---- ---- | ---- ---- |
| PMADRL | 18Eh | 0000 0000 | 0000 0000 | uuuu uuuu |
| PMADRH | 18Fh | ---- --00 | ---- --00 | ---- --uu |
| PMDATL | 190h | 0000 0000 | 0000 0000 | uuuu uuuu |
| PMDATH | 191h | --00 0000 | --00 0000 | --uu uuuu |
| COG1PH | 192h | ---- xxxx | ---- uuuu | ---- uuuu |
| COG1BLK | 193h | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| COG1DB | 194h | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| COG1CON0 | 195h | 0000 0000 | 0000 0000 | uuuu uuuu |
| COG1CON1 | 196h | --00 0000 | --00 0000 | --uu uuuu |
| COG1ASD | 197h | 0000 0000 | 0000 0000 | uuuu uuuu |

**Legend:** u = unchanged, x = unknown, – = unimplemented bit, reads as '0', q = value depends on condition.

**Note 1:** If $V_{DD}$ goes too low, Power-on Reset will be activated and registers will be affected differently.

**2:** One or more bits in INTCON and/or PIRx will be affected (to cause wake-up).

**3:** When the wake-up is due to an interrupt and the GIE bit is set, the PC is loaded with the interrupt vector (0004h).

**4:** See Table 19-5 for Reset value for specific condition.

**5:** If Reset was due to brown-out, then bit 0 = 0. All other Resets will cause bit 0 = u.

**TABLE 19-5: INITIALIZATION CONDITION FOR SPECIAL REGISTERS**

| Condition | Program Counter | Status Register | PCON Register |
|---|---|---|---|
| Power-on Reset | 000h | 0001 1xxx | ---- --0x |
| $\overline{MCLR}$ Reset during normal operation | 000h | 000u uuuu | ---- --uu |
| $\overline{MCLR}$ Reset during Sleep | 000h | 0001 0uuu | ---- --uu |
| WDT Reset | 000h | 0000 uuuu | ---- --uu |
| WDT Wake-up | PC + 1 | uuu0 0uuu | ---- --uu |
| Brown-out Reset | 000h | 0001 1uuu | ---- --u0 |
| Interrupt Wake-up from Sleep | PC + 1(1) | uuu1 0uuu | ---- --uu |

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

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

## 19.4 Interrupts

The PIC16F753/HV753 has multiple sources of interrupt:

- External Interrupt (INT pin)
- Interrupt-On-Change (IOC) Interrupts
- Timer0 Overflow Interrupt
- Timer1 Overflow Interrupt
- Timer2 Match Interrupt
- Hardware Limit Timer (HLT) Interrupt
- Comparator Interrupt (C1/C2)
- ADC Interrupt
- Complementary Output Generator (COG)
- CCP1 Interrupt
- Flash Memory Self-Write

The Interrupt Control register (INTCON) and Peripheral Interrupt Request Registers (PIRx) record individual interrupt requests in flag bits. The INTCON register also has individual and global interrupt enable bits.

The Global Interrupt Enable bit, GIE of the INTCON register, enables (if set) all unmasked interrupts, or disables (if cleared) all interrupts. Individual interrupts can be disabled through their corresponding enable bits in the INTCON register and PIEx registers. GIE is cleared on Reset.

When an interrupt is serviced, the following actions occur automatically:

- The GIE is cleared to disable any further interrupt.
- The return address is pushed onto the stack.
- The PC is loaded with 0004h.

The Return from Interrupt instruction, `RETFIE`, exits the interrupt routine, as well as sets the GIE bit, which re-enables unmasked interrupts.

The following interrupt flags are contained in the INTCON register:

- INT Pin Interrupt
- Interrupt-On-Change (IOC) Interrupts
- Timer0 Overflow Interrupt

The peripheral interrupt flags are contained in the PIR1 and PIR2 registers. The corresponding interrupt enable bit is contained in the PIE1 and PIE2 registers.

The following interrupt flags are contained in the PIR1 register:

- A/D Interrupt
- Comparator Interrupt
- Timer1 Overflow Interrupt
- Timer2 Match Interrupt
- Enhanced CCP Interrupt

For external interrupt events, such as the INT pin or PORTA change interrupt, the interrupt latency will be three or four instruction cycles. The exact latency depends upon when the interrupt event occurs (see Figure 19-8). The latency is the same for one or two-cycle instructions. Once in the Interrupt Service Routine, the source(s) of the interrupt can be determined by polling the interrupt flag bits. The interrupt flag bit(s) must be cleared in software before re-enabling interrupts to avoid multiple interrupt requests.

| | |
|---|---|
| **Note 1:** | Individual interrupt flag bits are set, regardless of the status of their corresponding mask bit or the GIE bit. |
| **2:** | When an instruction that clears the GIE bit is executed, any interrupts that were pending for execution in the next cycle are ignored. The interrupts, which were ignored, are still pending to be serviced when the GIE bit is set again. |

For additional information on Timer1, Timer2, comparators, ADC, Enhanced CCP modules, refer to the respective peripheral section.

### 19.4.1 RA2/INT INTERRUPT

The external interrupt on the RA2/INT pin is edge-triggered; either on the rising edge if the INTEDG bit of the OPTION register is set, or the falling edge, if the INTEDG bit is clear. When a valid edge appears on the RA2/INT pin, the INTF bit of the INTCON register is set. This interrupt can be disabled by clearing the INTE control bit of the INTCON register. The INTF bit must be cleared by software in the Interrupt Service Routine before re-enabling this interrupt. The RA2/INT interrupt can wake-up the processor from Sleep, if the INTE bit was set prior to going into Sleep. See **Section 19.7 "Power-Down Mode (Sleep)"** for details on Sleep and Figure 19-10 for timing of wake-up from Sleep through RA2/INT interrupt.

| | |
|---|---|
| **Note:** | The ANSEL register must be initialized to configure an analog channel as a digital input. Pins configured as analog inputs will read '0' and cannot generate an interrupt. |

### 19.4.2 TIMER0 INTERRUPT

An overflow (FFh → 00h) in the TMR0 register will set the T0IF bit of the INTCON register. The interrupt can be enabled/disabled by setting/clearing T0IE bit of the INTCON register. See **Section 6.0 "Timer0 Module"** for operation of the Timer0 module.

### 19.4.3 PORTA INTERRUPT-ON-CHANGE

An input change on PORTA sets the IOCIF bit of the INTCON register. The interrupt can be enabled/disabled by setting/clearing the IOCIE bit of the INTCON register. Plus, individual pins can be configured through the IOC register.

| | |
|---|---|
| **Note:** | If a change on the I/O pin should occur when any PORTA operation is being executed, then the IOCIF interrupt flag may not get set. |

**FIGURE 19-7: INTERRUPT LOGIC**

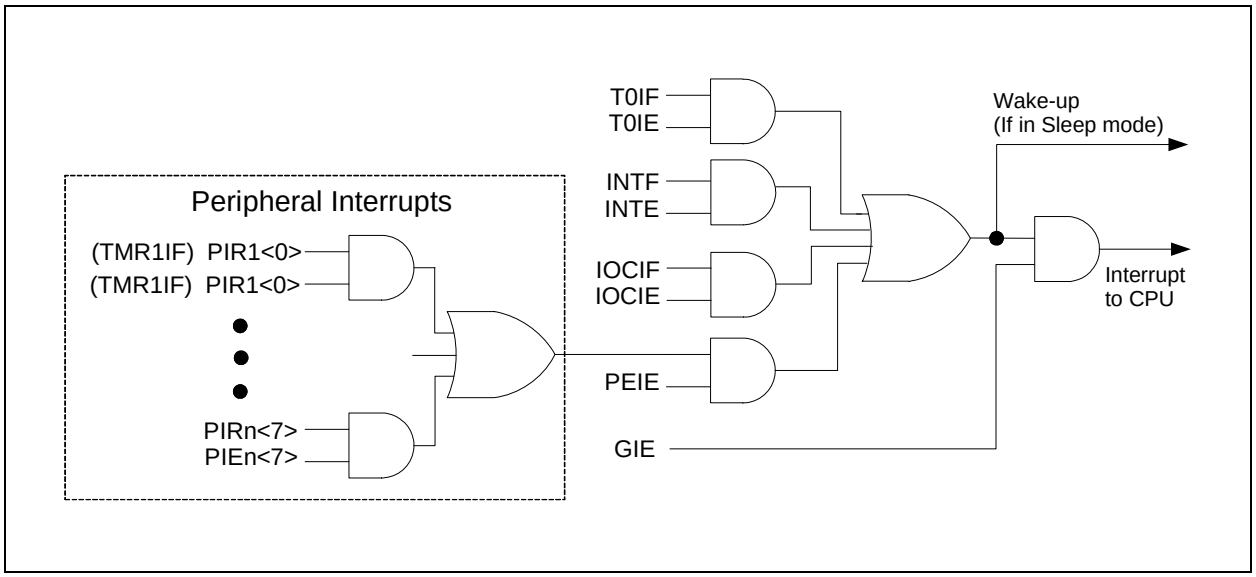
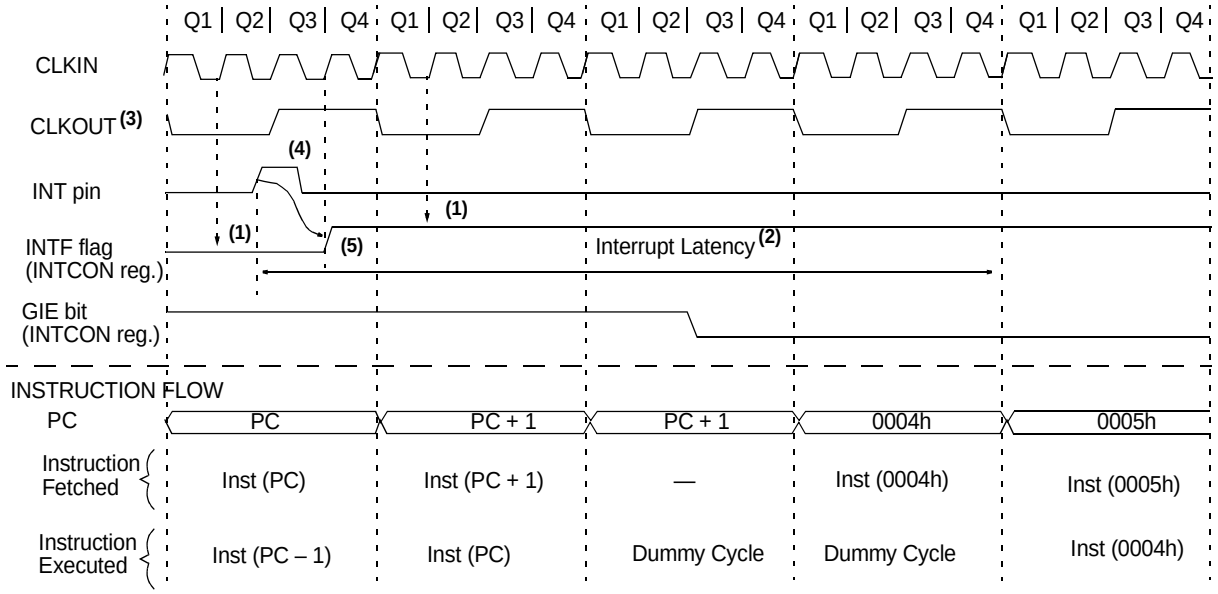


**FIGURE 19-8: INT PIN INTERRUPT TIMING**




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

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

**3:** CLKOUT is available only in INTOSC and RC Oscillator modes.

**4:** For minimum width of INT pulse, refer to AC specifications in **Section 22.0 "Electrical Specifications"**.

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

**TABLE 19-6: 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 | T0IE | INTE | IOCIE | T0IF | INTF | IOCIF | 17 |
| IOCAF | — | — | IOCAF5 | IOCAF4 | IOCAF3 | IOCAF2 | IOCAF1 | IOCAF0 | 45 |
| IOCAN | — | — | IOCAN5 | IOCAN4 | IOCAN3 | IOCAN2 | IOCAN1 | IOCAN0 | 45 |
| IOCAP | — | — | IOCAP5 | IOCAP4 | IOCAP3 | IOCAP2 | IOCAP1 | IOCAP0 | 45 |
| LATA | — | — | LATA5 | LATA4 | — | LATA2 | LATA1 | LATA0 | 43 |
| PIE1 | TMR1GIE | ADIE | — | — | HLTMR2IE | HLTMR1IE | TMR2IE | TMR1IE | 18 |
| PIR1 | TMR1GIF | ADIF | — | — | HLTMR2IF | HLTMR1IF | TMR2IF | TMR1IF | 20 |

**Legend:** x = unknown, u = unchanged, – = unimplemented read as '0', q = value depends upon condition. Shaded cells are not used by the interrupt module.

## 19.5 Context Saving During Interrupts

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

Temporary holding registers W_TEMP and STATUS_TEMP should be placed in the last 16 bytes of GPR (see Figure 2-2). These 16 locations are common to all banks and do not require banking. This makes context save and restore operations simpler. The code shown in Example 19-1 can be used to:

- Store the W register
- Store the STATUS register
- Execute the ISR code
- Restore the Status (and Bank Select Bit register)
- Restore the W register

| | |
|---|---|
| **Note:** | The PIC16F753/HV753 does not require saving the PCLATH. However, if computed GOTOs are used in both the ISR and the main code, the PCLATH must be saved and restored in the ISR. |

**EXAMPLE 19-1: SAVING STATUS AND W REGISTERS IN RAM**

```
MOVWF   W_TEMP              ;Copy W to TEMP register
SWAPF   STATUS,W            ;Swap status to be saved into W
                            ;Swaps are used because they do not affect the status bits
MOVWF   STATUS_TEMP         ;Save status to bank zero STATUS_TEMP register
:
:(ISR)                      ;Insert user code here
:
SWAPF   STATUS_TEMP,W       ;Swap STATUS_TEMP register into W
                            ;(sets bank to original state)
MOVWF   STATUS              ;Move W into STATUS register
SWAPF   W_TEMP,F            ;Swap W_TEMP
SWAPF   W_TEMP,W            ;Swap W_TEMP into W
```

## 19.6 Watchdog Timer (WDT)

The Watchdog Timer is a free running timer, using LFINTOSC oscillator as its clock source. The WDT is enabled by setting the WDTE bit of the Configuration Word (default setting). When WDTE is set, the LFINTOSC will always be enabled to provide a clock source to the WDT module.

During normal operation, a WDT time-out generates a device Reset. If the device is in Sleep mode, a WDT time-out causes the device to wake-up and continue with normal operation.

The WDT can be permanently disabled by programming the Configuration bit, WDTE, as clear (**Section 19.1 "Configuration Bits"**).

### 19.6.1 WDT PERIOD

The WDT has a nominal time-out period of 18 ms (with no prescaler). The time-out periods vary with temperature, $V_{DD}$ and process variations from part to part (see DC specs). If longer time-out periods are desired, a prescaler with a division ratio of up to 1:128 can be assigned to the WDT under software control by writing to the OPTION register. Thus, time-out periods up to 2.3 seconds can be realized.

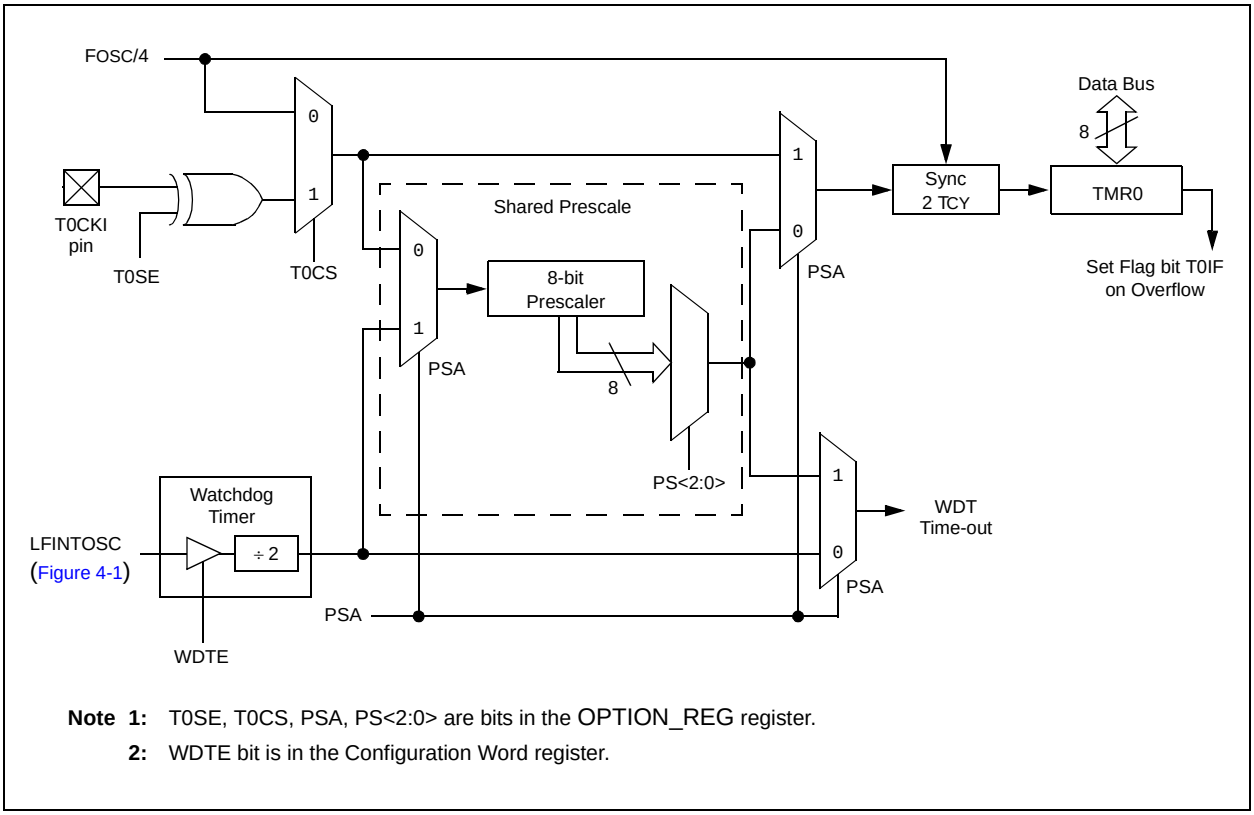
The CLRWDT and SLEEP instructions clear the WDT and the prescaler, if assigned to the WDT, and prevent it from timing out and generating a device Reset.

The $\overline{TO}$ bit in the STATUS register will be cleared upon a Watchdog Timer time-out.

### 19.6.2 WDT PROGRAMMING CONSIDERATIONS

It should also be taken in account that under worst-case conditions (i.e., VDD = Min., Temperature = Max., Max. WDT prescaler) it may take several seconds before a WDT time-out occurs.

**FIGURE 19-9: WATCHDOG TIMER WITH SHARED PRESCALE BLOCK DIAGRAM**




**Note 1:** T0SE, T0CS, PSA, PS<2:0> are bits in the OPTION_REG register.

**2:** WDTE bit is in the Configuration Word register.

**TABLE 19-7: WDT STATUS**

| Conditions | WDT |
|---|---|
| WDTE = 0 | Cleared |
| CLRWDT Command | |
| Exit Sleep | |

**TABLE 19-8: 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 |
|---|---|---|---|---|---|---|---|---|---|
| OPTION_REG | $\overline{\text{RAPU}}$ | INTEDG | T0CS | T0SE | PSA | PS<2:0> | | | 56 |

**Legend:** Shaded cells are not used by the Watchdog Timer.

**TABLE 19-9: SUMMARY OF CONFIGURATION WORD 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 |
|---|---|---|---|---|---|---|---|---|---|---|
| CONFIG(1) | 13:8 | — | — | $\overline{\text{DEBUG}}$ | $\overline{\text{CLKOUTEN}}$ | WRT<1:0> | | BOREN<1:0> | | 150 |
| | 7:0 | — | $\overline{\text{CP}}$ | MCLRE | $\overline{\text{PWRTE}}$ | WDTE | — | — | FOSC0 | |

**Legend:** — = unimplemented location, read as '1'. Shaded cells are not used by Watchdog Timer.

**Note 1:** See Register 19-1 for operation of all Configuration Word register bits.

## 19.7 Power-Down Mode (Sleep)

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

If the Watchdog Timer is enabled:

- WDT will be cleared but keeps running
- $\overline{PD}$ bit in the STATUS register is cleared
- $\overline{TO}$ bit is set
- Oscillator driver is turned off
- I/O ports maintain the status they had before SLEEP was executed (driving high, low or high-impedance).

For lowest current consumption in this mode, all I/O pins should be either at VDD or VSS, with no external circuitry drawing current from the I/O pin and the comparators, DAC and FVR should be disabled. I/O pins that are high-impedance inputs should be pulled high or low externally to avoid switching currents caused by floating inputs. The T0CKI input should also be at VDD or VSS for lowest current consumption. The contribution from on-chip pull-ups on PORTA should be considered.

The $\overline{MCLR}$ pin must be at a logic high level.

| | |
|---|---|
| **Note:** | It should be noted that a Reset generated by a WDT time-out does not drive $\overline{MCLR}$ pin low. |

### 19.7.1 WAKE-UP FROM SLEEP

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

1. External Reset input on $\overline{MCLR}$ pin.
2. Watchdog Timer wake-up.
3. Interrupt from INT pin.
4. Interrupt-On-Change input change.
5. Peripheral interrupt.

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

The following peripheral interrupts can wake the device from Sleep:

1. Timer1 interrupt. Timer1 must be operating as an asynchronous counter.
2. CCP Capture mode interrupt.
3. A/D conversion (when A/D clock source is RC).
4. Comparator output changes state.
5. Interrupt-on-change.
6. External Interrupt from INT pin.

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

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

| | |
|---|---|
| **Note:** | If the global interrupts are disabled (GIE is cleared) and any interrupt source has both its interrupt enable bit and the corresponding interrupt flag bits set, the device will immediately wake-up from Sleep. |

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

### 19.7.2 WAKE-UP USING INTERRUPTS

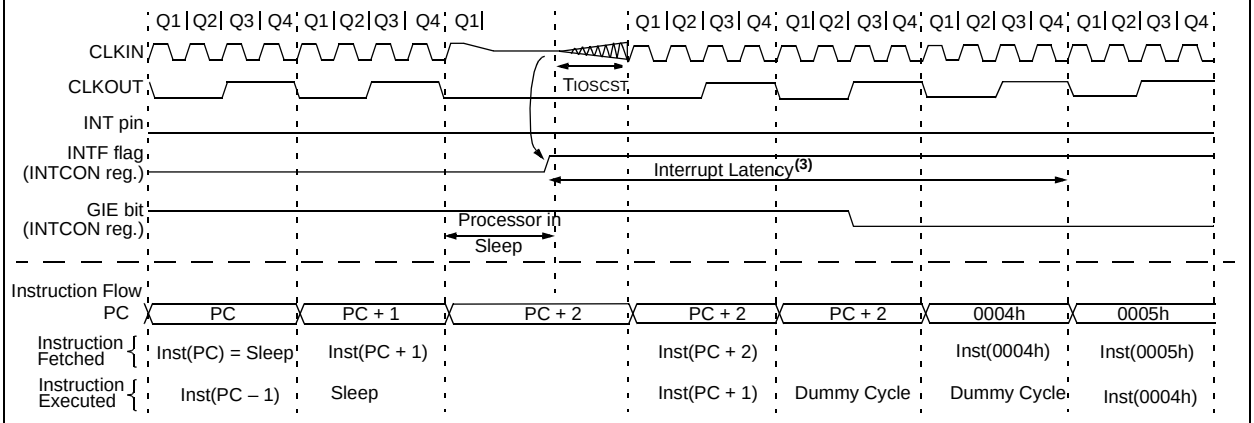
When global interrupts are disabled (GIE cleared) and any interrupt source has both its interrupt enable bit and interrupt flag bit set, one of the following will occur:

- If the interrupt occurs **before** the execution of a SLEEP instruction, the SLEEP instruction will complete as a NOP. Therefore, the WDT and WDT prescaler and postscaler (if enabled) will not be cleared, the $\overline{TO}$ bit will not be set and the $\overline{PD}$ bit will not be cleared.
- If the interrupt occurs **during or after** the execution of a SLEEP instruction, the device will Immediately wake-up from Sleep. The SLEEP instruction is executed. Therefore, the WDT and WDT prescaler and postscaler (if enabled) will be cleared, the $\overline{TO}$ bit will be set and the $\overline{PD}$ bit will be cleared.

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

To ensure that the WDT is cleared, a CLRWDT instruction should be executed before a SLEEP instruction. See Figure 19-10 for more details.

**FIGURE 19-10: WAKE-UP FROM SLEEP THROUGH INTERRUPT**




**Note 1:** HFINTOSC Oscillator mode assumed.

**2:** GIE = '1' assumed. In this case after wake-up, the processor jumps to 0004h. If GIE = '0', execution will continue in-line.

## 19.8 Code Protection

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

| **Note:** | The entire Flash program memory will be erased when the code protection is turned off. See the *PIC16F753/HV753 Flash Memory Programming Specification* (DS41686) for more information. |
|---|---|

## 19.9 ID Locations

Four memory locations (2000h-2003h) are designated as ID locations where the user can store checksum or other code identification numbers. These locations are not accessible during normal execution but are readable and writable during Program/Verify mode. Only the Least Significant seven bits of the ID locations are reported when using MPLAB® IDE.

## 19.10 In-Circuit Serial Programming™

The PIC16F753/HV753 microcontrollers can be serially programmed while in the end application circuit. This is simply done with five connections for:

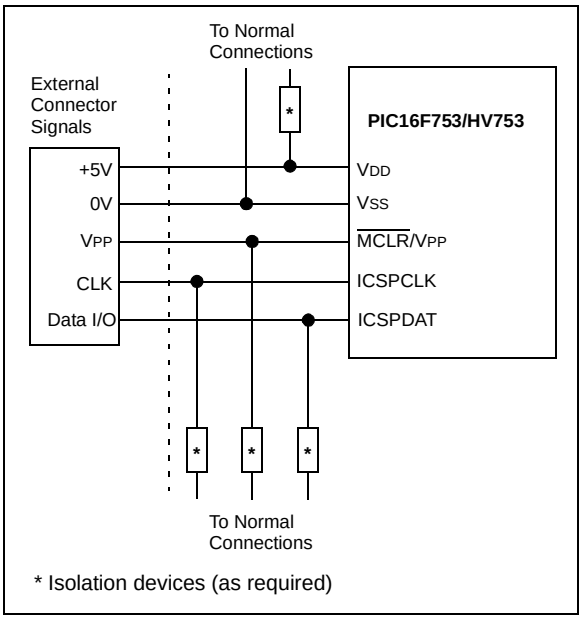
- clock
- data
- power
- ground
- programming voltage

This allows customers to manufacture boards with unprogrammed devices and then program the microcontroller just before shipping the product. This also allows the most recent firmware or a custom firmware to be programmed.

The device is placed into a Program/Verify mode by holding the ICSPDAT and ICSPCLK pins low, while raising the $\overline{MCLR}$ ($V_{PP}$) pin from $V_{IL}$ to $V_{IHH}$. See the *PIC16F753/HV753 Flash Memory Programming Specification* (DS41686) for more information. ICSPDAT becomes the programming data and ICSPCLK becomes the programming clock. Both ICSPDAT and ICSPCLK are Schmitt Trigger inputs in Program/Verify mode.

A typical In-Circuit Serial Programming connection is shown in Figure 19-11.

**FIGURE 19-11: TYPICAL IN-CIRCUIT SERIAL PROGRAMMING CONNECTION**




| **Note:** | To erase the device, $V_{DD}$ must be above the Bulk Erase $V_{DD}$ minimum given in the *PIC16F753/HV753 Flash Memory Programming Specification* (DS41686). |
|---|---|

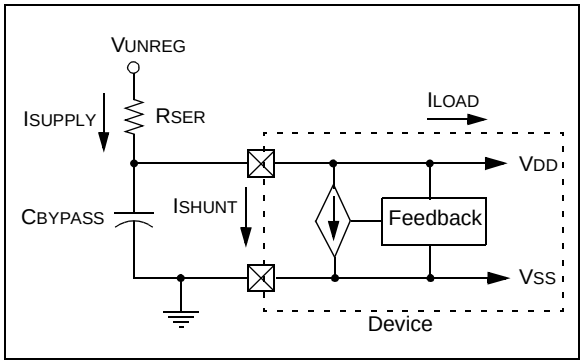
## 20.0 SHUNT REGULATOR (PIC16HV753 ONLY)

The PIC16HV753 devices include a permanent internal 5 volt (nominal) shunt regulator in parallel with the VDD pin. This eliminates the need for an external voltage regulator in systems sourced by an unregulated supply. All external devices connected directly to the VDD pin will share the regulated supply voltage and contribute to the total VDD supply current (ILOAD).

### 20.1 Regulator Operation

A shunt regulator generates a specific supply voltage by creating a voltage drop across a pass resistor RSER. The voltage at the VDD pin of the microcontroller is monitored and compared to an internal voltage reference. The current through the resistor is then adjusted, based on the result of the comparison, to produce a voltage drop equal to the difference between the supply voltage VUNREG and the VDD of the microcontroller. See Figure 20-1 for voltage regulator schematic.

**FIGURE 20-1: SHUNT REGULATOR**




An external current limiting resistor, RSER, located between the unregulated supply, VUNREG, and the VDD pin, drops the difference in voltage between VUNREG and VDD. RSER must be between RMAX and RMIN as defined by Equation 20-1.

**EQUATION 20-1: RSER LIMITING RESISTOR**

$$R_{MAX} = \frac{(VU_{MIN} - 5V)}{1.05 \bullet (1\ mA + I_{LOAD})}$$

$$R_{MIN} = \frac{(VU_{MAX} - 5V)}{0.95 \bullet (50\ mA)}$$

Where:

RMAX = maximum value of RSER (ohms)

RMIN = minimum value of RSER (ohms)

VUMIN = minimum value of VUNREG

VUMAX = maximum value of VUNREG

VDD = regulated voltage (5V nominal)

ILOAD = maximum expected load current in mA including I/O pin currents and external circuits connected to VDD.

1.05 = compensation for +5% tolerance of RSER

0.95 = compensation for -5% tolerance of RSER

### 20.2 Regulator Considerations

The supply voltage VUNREG and load current are not constant. Therefore, the current range of the regulator is limited. Selecting a value for RSER must take these three factors into consideration.

Since the regulator uses the band gap voltage as the regulated voltage reference, this voltage reference is permanently enabled in the PIC16HV753 devices.

The shunt regulator will still consume current when below operating voltage range for the shunt regulator.

### 20.3 Design Considerations

For more information on using the shunt regulator and managing current load, see Application Note *AN1035*, *Designing with HV Microcontrollers* (DS01035).

## 21.0 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

### 21.1 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

## 21.2 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

## 21.3 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

## 21.4 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

## 21.5 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

## 21.6 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.

## 21.7 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 upgradable 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.

## 21.8 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.

## 21.9 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™).

## 21.10 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.

## 21.11 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.

## 21.12 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®

## 22.0 ELECTRICAL SPECIFICATIONS

### Absolute Maximum Ratings(†)

Ambient temperature under bias ..... -40° to +125°C

Storage temperature ..... -65°C to +150°C

Voltage on pins with respect to $V_{SS}$

- on $V_{DD}$ pin
  - PIC16HV753 ..... -0.3V to +6.5V
  - PIC16F753 ..... -0.3V to +6.5V
- on $\overline{MCLR}$ ..... -0.3V to +13.5V
- on all other pins ..... -0.3V to ($V_{DD}$ + 0.3V)

Maximum current

- on $V_{SS}$ pin(1)
  - -40°C ≤ $T_A$ ≤ +85°C ..... 95 mA
  - -40°C ≤ $T_A$ ≤ +125°C ..... 95 mA
- on $V_{DD}$ pin(1)
  - -40°C ≤ $T_A$ ≤ +85°C ..... 95 mA
  - -40°C ≤ $T_A$ ≤ +125°C ..... 95 mA
- on RA1, RA4, RA5 ..... 25 mA
- on RC4, RC5 ..... 50 mA

Clamp current, $I_K$ ($V_{PIN}$ < 0 or $V_{PIN}$ > $V_{DD}$) ..... ± 20 mA

**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 characteristics. See Table 22-6 to calculate device specific limitations.

† 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.

## 22.1 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)**

PIC16F753

$V_{DDMIN}$ ($F_{OSC} \leq 8$ MHz) ..... +2.0V
$V_{DDMIN}$ (8 MHz < $F_{OSC} \leq 10$ MHz) ..... +3.0V
$V_{DDMAX}$ (10 MHz < $F_{OSC} \leq 20$ MHz) ..... +5.5V

PIC16HV753

$V_{DDMIN}$ ($F_{OSC} \leq 8$ MHz) ..... +2.0V
$V_{DDMIN}$ (8 MHz < $F_{OSC} \leq 10$ MHz) ..... +3.0V
$V_{DDMAX}$ (10 MHz < $F_{OSC} \leq 20$ MHz) ..... +5.0V

**$T_A$ — Operating Ambient Temperature Range**

Industrial Temperature

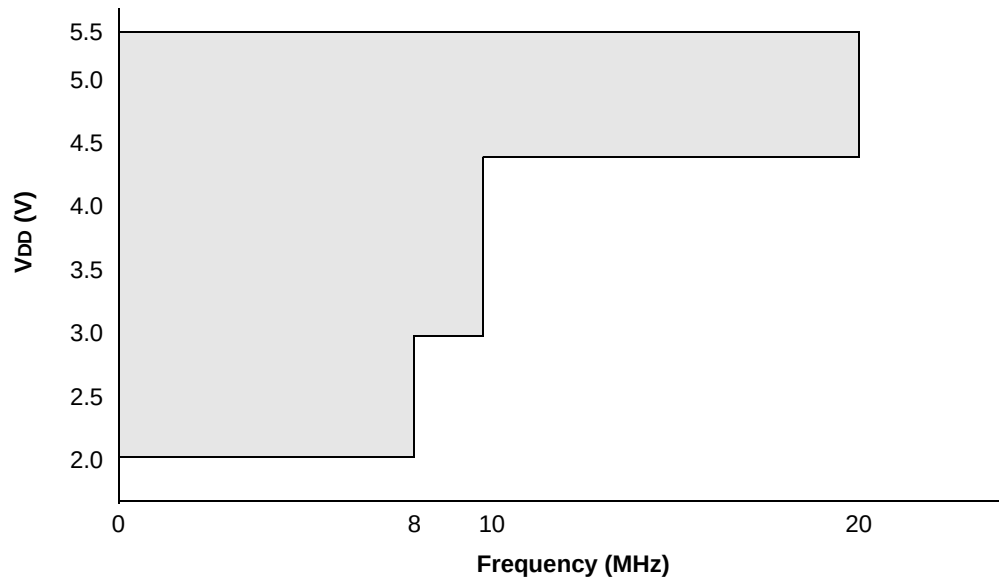
$T_{A\_MIN}$ ..... -40°C
$T_{A\_MAX}$ ..... +85°C

Extended Temperature

$T_{A\_MIN}$ ..... -40°C
$T_{A\_MAX}$ ..... +125°C

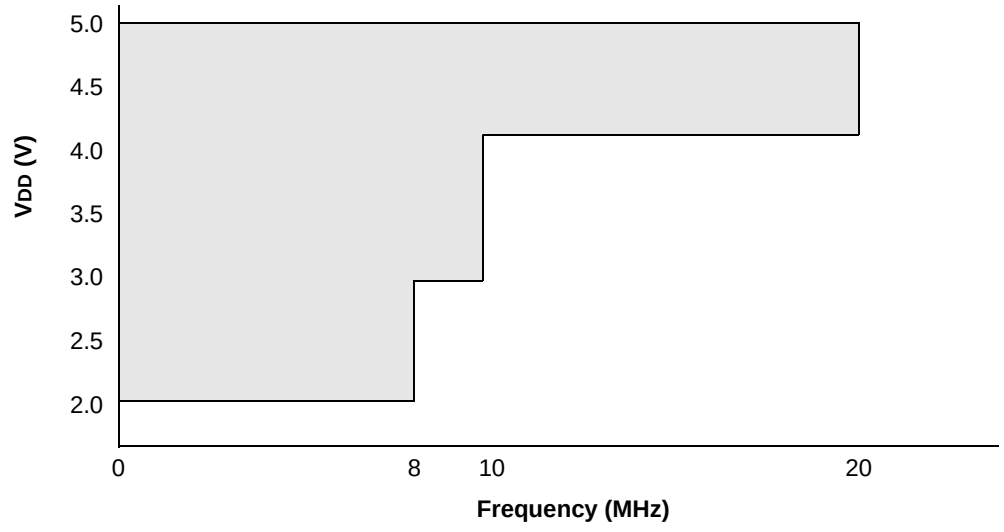
**Note 1:** See Parameter D001, DS Characteristics: Supply Voltage.

**FIGURE 22-1: PIC16F753 VOLTAGE-FREQUENCY GRAPH, -40°C ≤ TA ≤ +125°C**




**Note 1:** The shaded region indicates the permissible combinations of voltage and frequency.

**FIGURE 22-2: PIC16HV753 VOLTAGE-FREQUENCY GRAPH, -40°C ≤ TA ≤ +125°C**




**Note 1:** The shaded region indicates the permissible combinations of voltage and frequency.

## 22.2 DC Characteristics

**TABLE 22-1: SUPPLY VOLTAGE**

| PIC16F753 | | | Standard Operating Conditions (unless otherwise stated) | | | | |
|---|---|---|---|---|---|---|---|
| PIC16HV753 | | | | | | | |
| Param No. | Sym. | Characteristic | Min. | Typ† | Max. | Units | Conditions |
| D001 | VDD | **Supply Voltage** | | | | | |
| | | | VDDMIN | | VDDMAX | | |
| | | | 2.0 | — | 5.5 | V | FOSC ≤ 8 MHz |
| | | | 3.0 | — | 5.5 | V | FOSC ≤ 10 MHz |
| | | | 4.5 | — | 5.5 | V | FOSC≤ 20 MHz |
| D001 | | | 2.0 | — | 5.0 | V | FOSC ≤ 8 MHz(2) |
| | | | 3.0 | — | 5.0 | V | FOSC ≤ 10 MHz(2) |
| | | | 4.5 | — | 5.0 | V | FOSC ≤ 20 MHz(2) |
| D002* | VDR | **RAM Data Retention Voltage(1)** | | | | | |
| | | | 1.5 | — | — | V | Device in Sleep mode |
| D002 | | | 1.5 | — | — | V | Device in Sleep mode |
| D003* | VPOR | **VDD Start Voltage** to ensure internal Power-on Reset signal | | | | | |
| | | | — | 1.6 | — | V | |
| D003 | | | — | 1.6 | — | V | |
| D004* | SVDD | **VDD Rise Rate** to ensure VDD Rise Rate internal Power-on Reset signal | | | | | |
| | | | 0.05 | — | — | V/ms | See Table for details. |

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** This is the limit to which VDD can be lowered in Sleep mode without losing RAM data.

**2:** On the PIC16HV753, VDD is regulated by a Shunt Regulator and is dependent on series resistor (connected between the unregulated supply voltage and the VDD pin) to limit the current to 50 mA. See **Section 20.0 "Shunt Regulator (PIC16HV753 Only)"** for design requirements.

**TABLE 22-2: SUPPLY CURRENT ($I_{DD}$)(1,2)**

| PIC16F753 | | | Standard Operating Conditions (unless otherwise stated) | | | | | |
|---|---|---|---|---|---|---|---|---|
| PIC16HV753 | | | | | | | | |
| Param No. | Device Characteristics | Min. | Typ† | Max. 85°C | Max. 125°C | Units | Conditions | |
| | | | | | | | $V_{DD}$ | Note |
| | Supply Current ($I_{DD}$)(1, 2) | | | | | | | |
| D010 | | — | 10 | 31 | 31 | µA | 2.0 | $F_{OSC}$ = 31 kHz LFINTOSC mode |
| | | — | 15 | 36 | 36 | µA | 3.0 | |
| | | — | 28 | 62 | 62 | µA | 5.0 | |
| D010 | | — | 75 | 158 | 158 | µA | 2.0 | $F_{OSC}$ = 31 kHz LFINTOSC mode |
| | | — | 151 | 192 | 192 | µA | 3.0 | |
| | | — | 201 | 385 | 385 | µA | 4.5 | |
| D011 | | — | 97 | 140 | 140 | µA | 2.0 | $F_{OSC}$ = 1 MHz EC Oscillator mode |
| | | — | 155 | 235 | 235 | µA | 3.0 | |
| | | — | 334 | 475 | 475 | µA | 5.0 | |
| D011 | | — | 135 | 225 | 225 | µA | 2.0 | $F_{OSC}$ = 1 MHz EC Oscillator mode |
| | | — | 260 | 370 | 370 | µA | 3.0 | |
| | | — | 395 | 595 | 595 | µA | 4.5 | |
| D012 | | — | 172 | 260 | 260 | µA | 2.0 | $F_{OSC}$ = 1 MHz HFINTOSC mode |
| | | — | 220 | 360 | 360 | µA | 3.0 | |
| | | — | 398 | 516 | 516 | µA | 5.0 | |
| D012 | | — | 210 | 338 | 338 | µA | 2.0 | $F_{OSC}$ = 1 MHz HFINTOSC mode |
| | | — | 334 | 432 | 432 | µA | 3.0 | |
| | | — | 461 | 680 | 680 | µA | 4.5 | |
| D013 | | — | 243 | 333 | 333 | µA | 2.0 | $F_{OSC}$ = 4 MHz EC Oscillator mode |
| | | — | 365 | 485 | 485 | µA | 3.0 | |
| | | — | 762 | 956 | 956 | µA | 5.0 | |
| D013 | | — | 261 | 385 | 385 | µA | 2.0 | $F_{OSC}$ = 4 MHz EC Oscillator mode |
| | | — | 490 | 620 | 620 | µA | 3.0 | |
| | | — | 710 | 1045 | 1045 | µA | 4.5 | |

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** The test conditions for all $I_{DD}$ measurements in active operation mode are: OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to $V_{SS}$; $\overline{MCLR}$ = $V_{DD}$; 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.

**TABLE 22-2: SUPPLY CURRENT ($I_{DD}$)(1,2) (CONTINUED)**

| PIC16F753 | | | Standard Operating Conditions (unless otherwise stated) | | | | | |
|---|---|---|---|---|---|---|---|---|
| PIC16HV753 | | | | | | | | |
| Param No. | Device Characteristics | Min. | Typ† | Max. 85°C | Max. 125°C | Units | Conditions | |
| | | | | | | | $V_{DD}$ | Note |
| | Supply Current ($I_{DD}$)(1, 2) | | | | | | | |
| D014 | | — | 318 | 382 | 382 | µA | 2.0 | $F_{OSC}$ = 4 MHz HFINTOSC mode |
| | | — | 450 | 502 | 502 | µA | 3.0 | |
| | | — | 825 | 100 | 100 | µA | 5.0 | |
| D014 | | — | 330 | 485 | 485 | µA | 2.0 | $F_{OSC}$ = 4 MHz HFINTOSC mode |
| | | — | 526 | 658 | 658 | µA | 3.0 | |
| | | — | 775 | 980 | 980 | µA | 4.5 | |
| D015 | | — | 505 | 595 | 595 | µA | 2.0 | $F_{OSC}$ = 8 MHz HFINTOSC mode |
| | | — | 740 | 1200 | 1200 | µA | 3.0 | |
| | | — | 1.5 | 1.8 | 1.8 | mA | 5.0 | |
| D015 | | — | 500 | 690 | 690 | µA | 2.0 | $F_{OSC}$ = 8 MHz HFINTOSC mode |
| | | — | 800 | 1100 | 1100 | µA | 3.0 | |
| | | — | 1.23 | 1.7 | 1.7 | mA | 4.5 | |
| D016 | | — | 2.6 | 3.08 | 3.08 | mA | 4.5 | $F_{OSC}$ = 20 MHz EC Oscillator mode |
| | | — | 2.97 | 3.53 | 3.53 | mA | 5.0 | |
| D016 | | — | 2.6 | 3.3 | 3.3 | mA | 4.5 | $F_{OSC}$ = 20 MHz EC Oscillator mode |

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** The test conditions for all $I_{DD}$ measurements in active operation mode are: OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to $V_{SS}$; $\overline{MCLR}$ = $V_{DD}$; 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.

**TABLE 22-3: POWER-DOWN CURRENTS ($I_{PD}$) (1,2)**

| PIC16F753 | | Standard Operating Conditions (unless otherwise stated) Sleep mode | | | | | | |
|---|---|---|---|---|---|---|---|---|
| PIC16HV753 | | | | | | | | |
| Param No. | Device Characteristics | Min. | Typ† | Max. 85°C | Max. 125°C | Units | Conditions | |
| | | | | | | | $V_{DD}$ | Note |
| | **Power-down Base Current ($I_{PD}$)(2)** | | | | | | | |
| D020 | | — | 0.05 | 0.50 | 3.50 | µA | 2.0 | WDT, BOR, Comparator, $V_{REF}$ and T1OSC disabled |
| | | — | 0.15 | 1.00 | 4.00 | µA | 3.0 | |
| | | — | 0.35 | 1.50 | 5.00 | µA | 5.0 | |
| D020 | | — | 70 | 130 | 140 | µA | 2.0 | |
| | | — | 140 | 175 | 185 | µA | 3.0 | |
| | | — | 175 | 230 | 250 | µA | 4.5 | |
| | **Power-down Base Current ($I_{PD}$)(2, 3)** | | | | | | | |
| D021 | | — | 0.96 | 1.30 | 3.72 | µA | 2.0 | WDT Current(1) |
| | | — | 1.05 | 2.10 | 6.50 | µA | 3.0 | |
| | | — | 1.87 | 2.92 | 6.86 | µA | 5.0 | |
| D021 | | — | 66 | 127 | 141 | µA | 2.0 | |
| | | — | 137 | 172 | 176 | µA | 3.0 | |
| | | — | 176 | 228 | 233 | µA | 4.5 | |
| D022 | | — | 4 | 7 | 10 | µA | 3.0 | BOR Current(1) |
| | | — | 5 | 8 | 11 | µA | 5.0 | |
| D022 | | — | 140 | 175 | 180 | µA | 3.0 | |
| | | — | 178 | 230 | 236 | µA | 4.5 | |
| D023 | | — | 160 | 345 | 375 | µA | 2.0 | CxSP = 1, Comparator Current(1), single comparator enabled |
| | | — | 180 | 370 | 405 | µA | 3.0 | |
| | | — | 220 | 410 | 445 | µA | 5.0 | |
| D023 | | — | 225 | 380 | 380 | µA | 2.0 | |
| | | — | 250 | 420 | 420 | µA | 3.0 | |
| | | — | 381 | 500 | 500 | µA | 4.5 | |
| D024 | | — | 50 | 105 | 115 | µA | 2.0 | CxSP = 0, Comparator Current(1), single comparator enabled |
| | | — | 55 | 110 | 120 | µA | 3.0 | |
| | | — | 70 | 120 | 132 | µA | 5.0 | |
| D024 | | — | 115 | 200 | 200 | µA | 2.0 | |
| | | — | 150 | 220 | 220 | µA | 3.0 | |
| | | — | 240 | 277 | 277 | µA | 4.5 | |

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** The peripheral Δ current can be determined by subtracting the base $I_{PD}$ 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:** Shunt regulator is always ON and always draws operating current.

**TABLE 22-3: POWER-DOWN CURRENTS ($I_{PD}$) (CONTINUED)(1,2)**

| PIC16F753 | | Standard Operating Conditions (unless otherwise stated)<br>Sleep mode | | | | | | |
|---|---|---|---|---|---|---|---|---|
| PIC16HV753 | | | | | | | | |
| Param No. | Device Characteristics | Min. | Typ† | Max. 85°C | Max. 125°C | Units | Conditions | |
| | | | | | | | $V_{DD}$ | Note |
| | Power-down Base Current ($I_{PD}$)(2, 3) | | | | | | | |
| D025 | | — | 0.10 | 0.41 | 3.51 | µA | 3.0 | A/D Current(1), no conversion in progress |
| | | — | 0.12 | 0.55 | 4.41 | µA | 5.0 | |
| D025 | | — | 145 | 171 | 175 | µA | 3.0 | |
| | | — | 185 | 226 | 231 | µA | 4.5 | |
| D026 | | — | 20 | 37 | 37 | µA | 2.0 | DAC Current(1) |
| | | — | 30 | 46 | 46 | µA | 3.0 | |
| | | — | 50 | 76 | 76 | µA | 5.0 | |
| D026 | | — | 85 | 155 | 155 | µA | 2.0 | |
| | | — | 165 | 213 | 213 | µA | 3.0 | |
| | | — | 215 | 284 | 284 | µA | 4.5 | |
| D027 | | — | 115 | 185 | 203 | µA | 2.0 | FVR Current(1), FVRBUFEN = 1, FVROUT buffer enabled |
| | | — | 120 | 193 | 219 | µA | 3.0 | |
| | | — | 125 | 196 | 224 | µA | 5.0 | |
| D027 | | — | 65 | 126 | 145 | µA | 2.0 | |
| | | — | 136 | 171 | 182 | µA | 3.0 | |
| | | — | 175 | 226 | 231 | µA | 4.5 | |
| D028 | | — | 1 | 2 | 4 | µA | 2.0 | T1OSC Current, TMR1CS <1:0> = 11 |
| | | — | 2 | 3 | 5 | µA | 3.0 | |
| | | — | 9 | 20 | 21 | µA | 5.0 | |
| D028 | | — | 65 | 126 | 140 | µA | 2.0 | |
| | | — | 136 | 172 | 180 | µA | 3.0 | |
| | | — | 175 | 228 | 235 | µA | 4.5 | |
| D029 | | — | 140 | 258 | 265 | µA | 2.0 | Op-Amp Current(1) |
| | | — | 155 | 326 | 340 | µA | 3.0 | |
| | | — | 165 | 421 | 422 | µA | 5.0 | |
| D029 | | — | 140 | 260 | 265 | µA | 2.0 | |
| | | — | 155 | 325 | 340 | µA | 3.0 | |
| | | — | 165 | 400 | 410 | µA | 4.5 | |

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** The peripheral Δ current can be determined by subtracting the base $I_{PD}$ 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:** Shunt regulator is always ON and always draws operating current.

**TABLE 22-4: I/O PORTS**

| DC CHARACTERISTICS | | | Standard Operating Conditions (unless otherwise stated) | | | | |
|---|---|---|---|---|---|---|---|
| Param No. | Sym. | Characteristic | Min. | Typ† | Max. | Units | Conditions |
| | VIL | **Input Low Voltage** | | | | | |
| | | I/O PORT: | | | | | |
| D030 | | with TTL buffer | — | — | 0.8 | V | 4.5V ≤ VDD ≤ 5.5V |
| D030A | | | — | — | 0.15 VDD | V | 2.0V ≤ VDD ≤ 4.5V |
| D031 | | with Schmitt Trigger buffer | — | — | 0.2 VDD | V | 2.0V ≤ VDD ≤ 5.5V |
| | VIH | **Input High Voltage** | | | | | |
| | | I/O PORT: | | | | | |
| D040 | | with TTL buffer | 2.0 | — | — | V | 4.5V ≤ VDD ≤ 5.5V |
| D040A | | | 0.25 VDD + 0.8 | — | — | V | 2.0V ≤ VDD ≤ 4.5V |
| D041 | | with Schmitt Trigger buffer | 0.8 VDD | — | — | V | 2.0V ≤ VDD ≤ 5.5V |
| D042 | | $\overline{MCLR}$ | 0.8 VDD | — | — | V | |
| | IIL | **Input Leakage Current(1)** | | | | | |
| D060 | | I/O ports | — | ± 0.1 | ± 1 | µA | VSS ≤ VPIN ≤ VDD, Pin at high-impedance, 85°C |
| D061 | | RA3/$\overline{MCLR}$(2) | — | ± 0.7 | ± 5 | µA | VSS ≤ VPIN ≤ VDD, Pin at high-impedance, 85°C |
| D063 | | | — | ± 0.1 | ± 5 | µA | EC Configuration |
| | IPUR | **Weak Pull-up Current(3)** | | | | | |
| D070* | | | 50 | 250 | 400 | µA | VDD = 5.0V, VPIN = VSS |
| | VOL | **Output Low Voltage** | | | | | |
| D080 | | I/O Ports (excluding RC4, RC5) | — | — | 0.6 | V | IOL = 7 mA, VDD = 4.5V -40°C ≤ TA ≤ +125°C IOL = 8.5 mA, VDD = 4.5V -40°C ≤ TA ≤ +85°C |
| | | I/O Ports RC4 and RC5 | — | — | 0.6 | V | IOL = 14 mA, VDD = 4.5V -40°C ≤ TA ≤ +125°C IOL = 17 mA, VDD = 4.5V -40°C ≤ TA ≤ +85°C |
| | VOH | **Output High Voltage** | | | | | |
| D090 | | I/O Ports (excluding RC4, RC5) | VDD-0.7 | — | — | V | IOH = -2.5 mA, VDD = 4.5V -40°C ≤ TA ≤ +125°C IOH = -3 mA, VDD = 4.5V -40°C ≤ TA ≤ +85°C |
| | | I/O Ports RC4 and RC5 | VDD-0.7 | — | — | V | IOH = -5 mA, VDD = 4.5V -40°C ≤ TA ≤ +125°C IOH = -6 mA, VDD = 4.5V -40°C ≤ TA ≤ +85°C |

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** 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.

**3:** This specification applies to all weak pull-up pins, including the weak pull-up found on RA3/$\overline{MCLR}$. When RA3/$\overline{MCLR}$ is configured as $\overline{MCLR}$ Reset pin, the weak pull-up is always enabled.

**TABLE 22-4: I/O PORTS (CONTINUED)**

| DC CHARACTERISTICS | | | Standard Operating Conditions (unless otherwise stated) | | | | |
|---|---|---|---|---|---|---|---|
| Param No. | Sym. | Characteristic | Min. | Typ† | Max. | Units | Conditions |
| | | **Capacitive Loading Specs on Output Pins** | | | | | |
| D101* | COSC2 | OSC2 pin | — | — | 15 | pF | In XT, HS, LP modes when external clock is used to drive OSC1 |
| D101A* | CIO | All I/O pins | — | — | 50 | pF | |

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** 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.

**3:** This specification applies to all weak pull-up pins, including the weak pull-up found on RA3/$\overline{MCLR}$. When RA3/$\overline{MCLR}$ is configured as $\overline{MCLR}$ Reset pin, the weak pull-up is always enabled.

**TABLE 22-5: 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 $\overline{MCLR}$/$V_{PP}$ pin | 10.0 | — | 13.0 | V | **(Note 2)** |
| D112 | $V_{BE}$ | $V_{DD}$ for Bulk Erase | 4.5 | — | $V_{DDMAX}$ | V | |
| D113 | $V_{PEW}$ | $V_{DD}$ for Write or Row Erase | 4.5 | — | $V_{DDMAX}$ | V | |
| D114 | $I_{PPPGM}$ | Current on $\overline{MCLR}$/$V_{PP}$ during Erase/Write | — | 300 | 1000 | µA | |
| | | **Program Flash Memory** | | | | | |
| D121 | $E_P$ | Cell Endurance | 10K | 100K | — | E/W | -40°C ≤ $T_A$ ≤ +85°C **(Note 1)** |
| D121A | $E_P$ | Cell Endurance | 1K | 10K | — | E/W | -40°C ≤ $T_A$ ≤ +125°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 | N/A | — | — | E/W | 0°C to +60°C, Lower byte last 128 addresses |

† Data in "Typ" column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** Self-write and Block Erase.

**2:** Required only if single-supply programming is disabled.

**TABLE 0-2: OPERATIONAL AMPLIFIER (OPA) MODULE**

| Standard Operating Conditions (unless otherwise stated) | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Symbol** | **Parameters** | **Min.** | **Typ†** | **Max.** | **Units** | **Conditions** |
| OPA01* | VOS | Input Offset Voltage | — | ±8 | ±15 | mV | |
| OPA02* | IB | Input Bias Current | — | ±2 | — | nA | |
| OPA03* | IOS | Input Offset Bias Current | — | ±1 | — | pA | |
| OPA04* | VCM | Common Mode Input Range | VSS | — | VDD - 1.4 | V | |
| OPA05* | CMR | Common Mode Rejection Ratio | 60 | 70 | ±5 | dB | |
| OPA06* | AOL | DC Open Loop Gain | — | — | — | dB | |
| OPA07* | VOUT | Output Voltage Swing | VSS - 50 | — | VSS + 50 | mV | |
| OPA08* | ISC | Output Short Circuit Current | — | 10 | 15 | mA | |
| OPA10* | PSR | Power Supply Rejection | — | 60 | — | dB | |

* 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:** See **Section 23.0 "DC and AC Characteristics Graphs and Charts"** for operating characterization.

**2:** Response time is measured with one comparator input at (VDD - 1.5)/2 - 100 mV to (VDD - 1.5)/2 + 20mV.

**3:** Input offset voltage is measured with one comparator input at (VDD - 1.5V)/2.

**TABLE 22-6: THERMAL CHARACTERISTICS**

| Standard Operating Conditions (unless otherwise stated) | | | | | |
|---|---|---|---|---|---|
| **Param No.** | **Sym.** | **Characteristic** | **Typ.** | **Units** | **Conditions** |
| TH01 | $\theta_{JA}$ | Thermal Resistance Junction to Ambient | 84.6 | °C/W | 8-pin PDIP package |
| | | | 149.5 | °C/W | 8-pin SOIC package |
| | | | 60 | °C/W | 8-pin DFN 3x3mm package |
| TH02 | $\theta_{JC}$ | Thermal Resistance Junction to Case | 41.2 | °C/W | 8-pin PDIP package |
| | | | 39.9 | °C/W | 8-pin SOIC package |
| | | | 9 | °C/W | 8-pin DFN 3x3mm 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 | $P_{DER} = PD_{MAX} (T_J - T_A)/\theta_{JA}$ (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

## 22.3 Timing Parameter Symbology

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

1. TppS2ppS
2. TppS

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

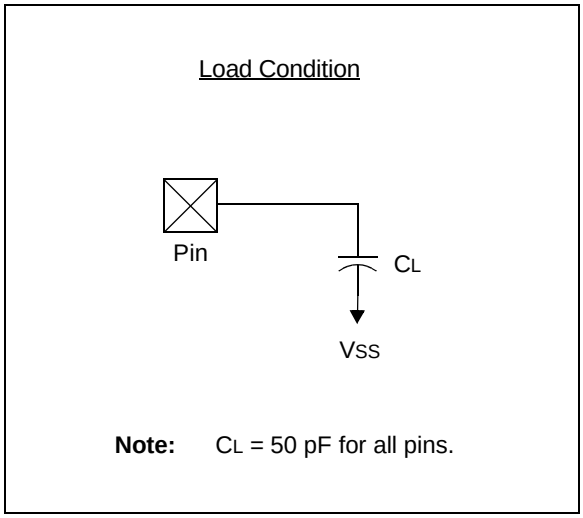
Lowercase letters (pp) and their meanings:

| | | | |
|---|---|---|---|
| **pp** | | | |
| cc | 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 22-3: LOAD CONDITIONS**




**Note:** $C_L$ = 50 pF for all pins.

## 22.4 AC Characteristics: PIC16F753/HV753 (Industrial, Extended)

**FIGURE 22-4: CLOCK TIMING**

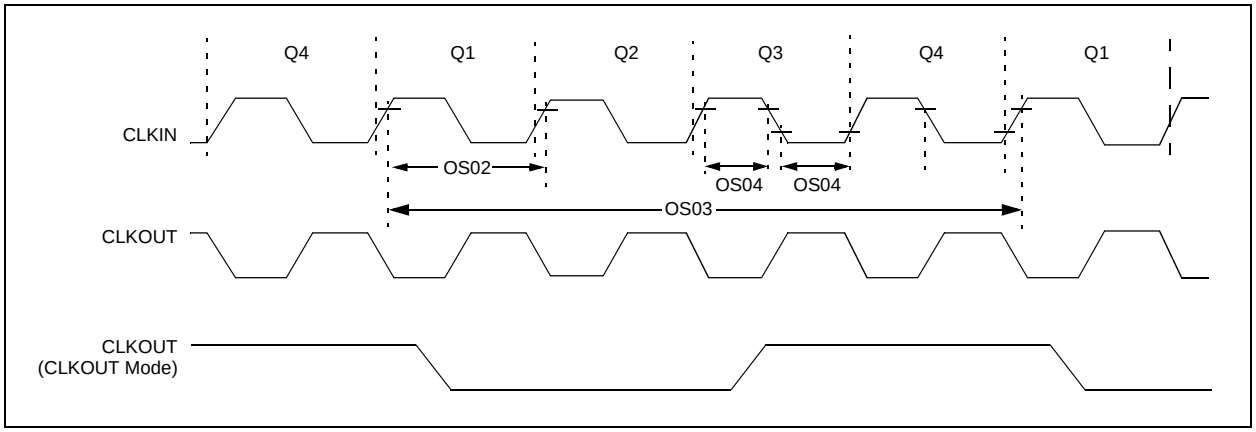



**TABLE 22-7: CLOCK OSCILLATOR TIMING REQUIREMENTS**

| **Standard Operating Conditions (unless otherwise stated)** | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Sym.** | **Characteristic** | **Min.** | **Typ†** | **Max.** | **Units** | **Conditions** |
| OS01 | FOSC | External CLKIN Frequency(1) | DC | — | 20 | MHz | EC Oscillator mode |
| OS02 | TOSC | External CLKIN Period(1) | 50 | — | ∞ | ns | EC Oscillator mode |
| OS03 | TCY | Instruction Cycle Time(1) | 200 | TCY | DC | ns | TCY = 4/FOSC |

* These parameters are characterized but not tested.

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

**Note 1:** Instruction cycle period (TCY) equals four times the input oscillator time base period. All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at "min" values with an external clock applied to CLKIN pin. When an external clock input is used, the "max" cycle time limit is "DC" (no clock) for all devices.

**TABLE 22-8: OSCILLATOR PARAMETERS**

| Standard Operating Conditions (unless otherwise stated) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Param No.** | **Sym.** | **Characteristic** | **Freq. Tolerance** | **Min.** | **Typ†** | **Max.** | **Units** | **Conditions** |
| OS06 | TWARM | Internal Oscillator Switch when running | — | — | — | 2 | TOSC | |
| OS07 | INTOSC | Internal Calibrated INTOSC Frequency(1) (4 MHz) | ±1% | 3.96 | 4.0 | 4.04 | MHz | VDD = 3.5V, TA = 25°C |
| | | | ±2% | 3.92 | 4.0 | 4.08 | MHz | 2.5V ≤ VDD ≤ 5.5V, 0°C ≤ TA ≤ +85°C |
| | | | ±5% | 3.80 | 4.0 | 4.20 | MHz | 2.0V ≤ VDD ≤ 5.5V, -40°C ≤ TA ≤ +85°C (Ind.), -40°C ≤ TA ≤ +125°C (Ext.) |
| OS08 | HFOSC | Internal Calibrated HFINTOSC Frequency(1) | ±1% | 7.92 | 8 | 8.08 | MHz | VDD = 3.5V, TA = 25°C |
| | | | ±2% | 7.84 | 8 | 8.16 | MHz | 2.5V ≤ VDD ≤ 5.5V, 0°C ≤ TA ≤ +85°C |
| | | | ±5% | 7.60 | 8 | 8.40 | MHz | 2.0V ≤ VDD ≤ 5.5V, -40°C ≤ TA ≤ +85°C (Ind.), -40°C ≤ TA ≤ +125°C (Ext.) |
| OS09 | LFOSC | Internal LFINTOSC Frequency | — | — | 31 | — | kHz | |
| OS10* | TIOSC ST | HFINTOSC Wake-up from Sleep Start-up Time | — | — | 12 | 24 | µs | VDD = 2.0V -40°C ≤ TA ≤ +85°C |
| | | | | — | 7 | 14 | µs | VDD = 3.0V -40°C ≤ TA ≤ +85°C |
| | | | | — | 6 | 11 | µs | VDD = 5.0V -40°C ≤ TA ≤ +85°C |

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** To ensure these oscillator frequency tolerances, VDD and VSS must be capacitively decoupled as close to the device as possible. 0.1 µF and 0.01 µF values in parallel are recommended.

**FIGURE 22-5: CLKOUT AND I/O TIMING**

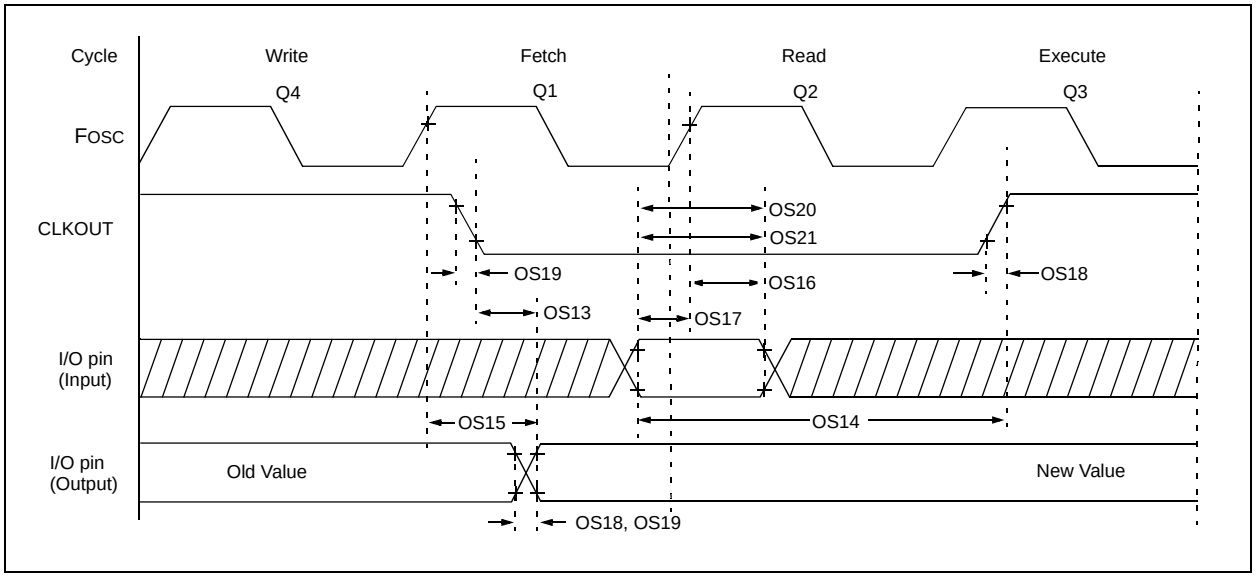



**TABLE 22-9: CLKOUT AND I/O TIMING PARAMETERS**

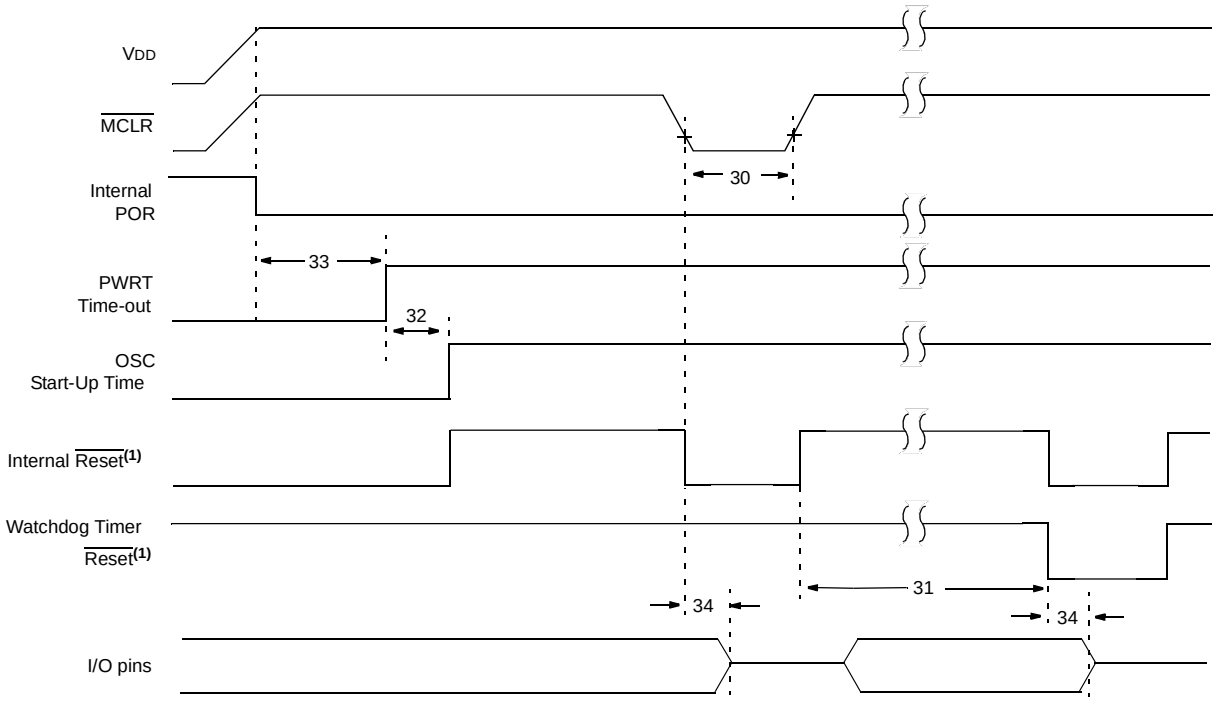
| **Standard Operating Conditions (unless otherwise stated)** | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Sym.** | **Characteristic** | **Min.** | **Typ †** | **Max.** | **Units** | **Conditions** |
| OS13 | TCKL2IOV | CLKOUT↓ to Port out valid(1) | — | — | 20 | ns | |
| OS14 | TIOV2CKH | Port input valid before CLKOUT↑(1) | TOSC + 200 ns | — | — | ns | |
| OS15 | TOSH2IOV | FOSC↑ (Q1 cycle) to Port out valid | — | 50 | 70* | ns | VDD = 5.0V |
| OS16 | TOSH2IOI | FOSC↑ (Q2 cycle) to Port input invalid (I/O in setup time) | 50 | — | — | ns | 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<br>ns | VDD = 2.0V<br>VDD = 5.0V |
| OS19* | TIOF | Port output fall time | —<br>— | 28<br>15 | 55<br>30 | ns<br>ns | VDD = 2.0V<br>VDD = 5.0V |
| OS20* | TINP | INT pin input high or low time | 25 | — | — | ns | |
| OS21* | TIOC | Interrupt-on-change new input level time | TCY | — | — | ns | |

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5.0V, 25°C unless otherwise stated.

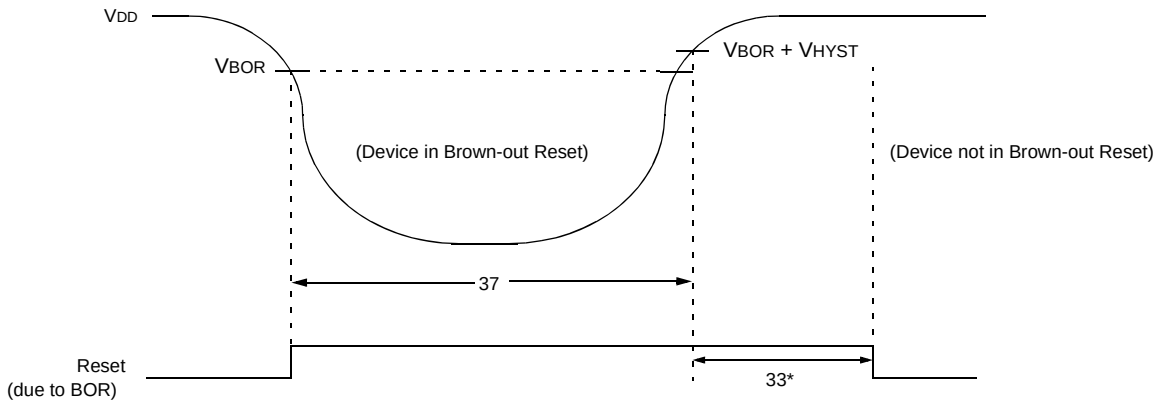
**Note 1:** Measurements are taken in RC mode where CLKOUT output is 4 x TOSC.

**FIGURE 22-6: RESET, WATCHDOG TIMER AND POWER-UP TIMER TIMING**




**Note:** Asserted low.

**FIGURE 22-7: BROWN-OUT RESET TIMING AND CHARACTERISTICS**




* 64 ms delay only if $\overline{\text{PWRTE}}$ bit in the Configuration Word register is programmed to '0'.

**TABLE 22-10: 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 | $\overline{\text{MCLR}}$ Pulse Width (low) | 2<br>5 | —<br>— | —<br>— | µs<br>µs | VDD = 5V, -40°C to +85°C<br>VDD = 5V, -40°C to +125°C |
| 31 | TWDTLP | Low-Power Watchdog Timer Time-out Period | 10<br>10 | 20<br>20 | 30<br>35 | ms<br>ms | VDD = 5V, -40°C to +85°C<br>VDD = 5V, -40°C to +125°C |
| 32* | TPWRT | Power-up Timer Period, PWRTE = 0 (No Prescaler) | 40 | 65 | 140 | ms | |
| 33* | TIOZ | I/O high impedance from $\overline{\text{MCLR}}$ Low or Watchdog Timer Reset | — | — | 2.0 | µs | |
| 34 | VBOR | Brown-out Reset Voltage (1) | 2 | 2.15 | 2.3 | V | |
| 35* | VHYST | Brown-out Reset Hysteresis | — | 100 | — | mV | -40°C ≤ TA ≤ +85°C |
| 36* | TBOR | Brown-out Reset DC Minimum Detection Period | — | 100 | — | µs | VDD ≤ VBOR |

* These parameters are characterized but not tested.

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

**Note 1:** 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 22-8: TIMER0 AND TIMER1 EXTERNAL CLOCK TIMINGS**

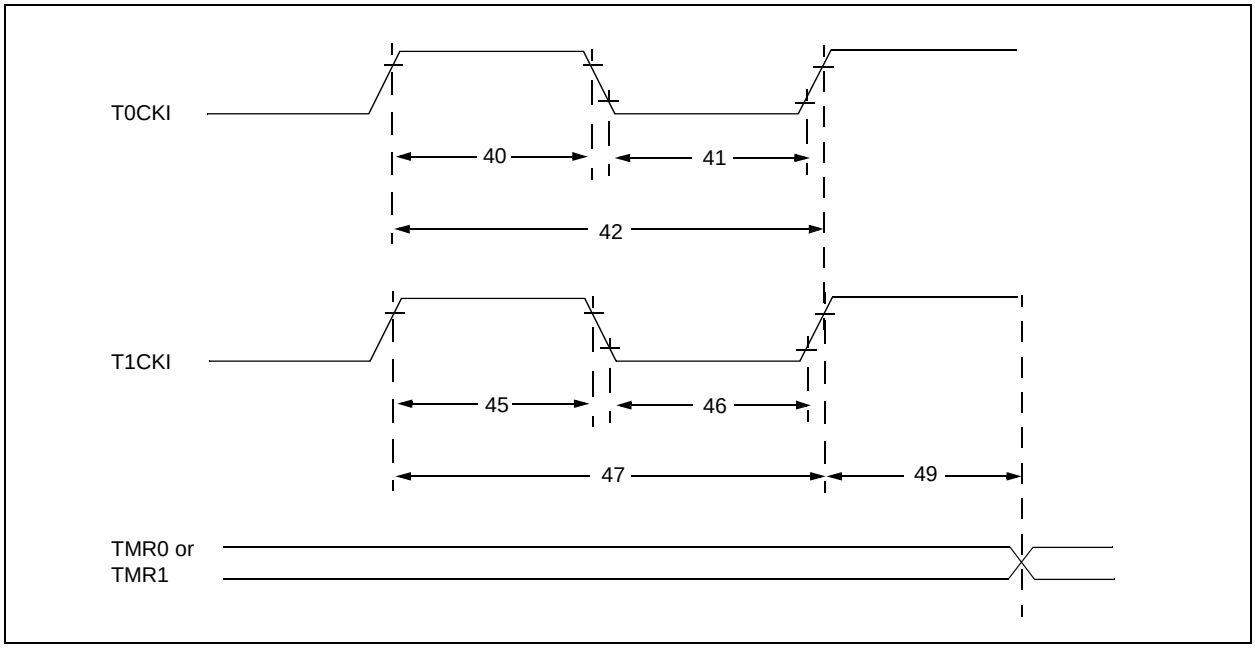


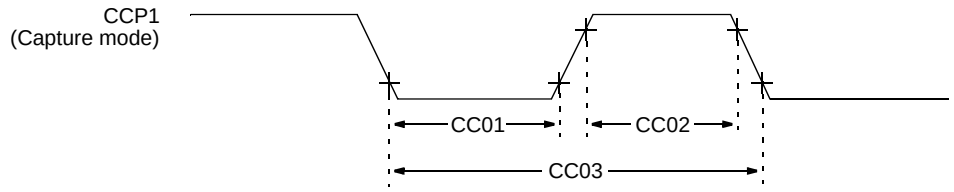

**TABLE 22-11: TIMER0 AND TIMER1 EXTERNAL CLOCK REQUIREMENTS**

| **Standard Operating Conditions (unless otherwise stated)** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **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* | TCKEZT-MR1 | 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 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**FIGURE 22-9: PIC16F753/HV753 CAPTURE/COMPARE/PWM TIMINGS (CCP)**




**Note:** Refer to Figure 22-3 for load conditions.

**TABLE 22-12: CAPTURE/COMPARE/PWM REQUIREMENTS (CCP)**

| Standard Operating Conditions (unless otherwise stated) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Param No.** | **Sym.** | **Characteristic** | | **Min.** | **Typ†** | **Max.** | **Units** | **Conditions** |
| CC01* | TccL | CCP1 Input Low Time | No Prescaler | 0.5TCY + 20 | — | — | ns | |
| | | | With Prescaler | 20 | — | — | ns | |
| CC02* | TccH | CCP1 Input High Time | No Prescaler | 0.5TCY + 20 | — | — | ns | |
| | | | With Prescaler | 20 | — | — | ns | |
| CC03* | TccP | CCP1 Input Period | | $\frac{3T_{CY} + 40}{N}$ | — | — | ns | N = prescale value (1, 4 or 16) |

* These parameters are characterized but not tested.

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

**TABLE 22-13: COMPARATOR SPECIFICATIONS(1)**

| Standard Operating Conditions (unless otherwise stated)<br>VDD = 5.0V, -40°C ≤ TA ≤ +125°C | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Sym.** | **Characteristics** | **Min.** | **Typ†** | **Max.** | **Units** | **Comments** |
| CM01 | VIOFF | Input Offset Voltage(3) | — | ± 10<br>± 10 | ± 20<br>± 20 | mV<br>mV | CxSP = 1<br>CxSP = 0 |
| CM02 | VICM | Input Common Mode Voltage(2) | 0 | — | VDD – 1.5 | V | |
| CM03 | CMRR | Common Mode Rejection Ratio | — | 55 | — | dB | |
| CM04A* | TRT(2) | Response Time | — | 55 | 70 | ns | CxSP = 1 |
| | | | — | 65 | 100 | ns | CxSP = 0 |
| CM05* | TMC20V | Comparator Mode Change to Output Valid | — | — | 10 | µs | |
| CM06 | CHYSTER | Comparator Hysteresis | — | 20 | 50 | mV | |

* These parameters are characterized but not tested.

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

**Note 1:** See **Section 23.0 "DC and AC Characteristics Graphs and Charts"** and **Section 22.0 "Electrical Specifications"** for operating characterization.

**2:** Response time is measured with one comparator input at (VDD - 1.5V)/2 - 100 mV to (VDD - 1.5V)/2 + 20 mV. The other input is at (VDD -1.5V)/2.

**3:** Input offset voltage is measured with one comparator input at (VDD - 1.5V)/2.

**TABLE 22-14: DIGITAL-TO-ANALOG CONVERTER (DAC) SPECIFICATIONS(1)**

| Standard Operating Conditions (unless otherwise stated)<br>VDD = 3.0V, TA = 25°C | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Sym.** | **Characteristics** | **Min.** | **Typ†** | **Max.** | **Units** | **Comments** |
| DAC01* | CLSB | Step Size | — | VDD/512 | — | V | |
| DAC02 | CACC | Absolute Accuracy | — | ± 1/2 | ± 2 | LSb | |
| DAC03* | CR | Unit Resistor Value (R) | — | 5K | — | Ω | |
| DAC04* | CST | Settling Time(2) | — | — | 10 | µs | |

* These parameters are characterized but not tested.

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

**Note 1:** See **Section 23.0 "DC and AC Characteristics Graphs and Charts"** for operating characterization.

**2:** Settling time measured while DACR<4:0> transitions from '0000' to '1111'.

**TABLE 22-15: FIXED VOLTAGE REFERENCE SPECIFICATIONS**

| Standard Operating Conditions (unless otherwise stated)<br>VDD = 3.0V, TA = 25°C | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Symbol** | **Characteristics** | **Min.** | **Typ.** | **Max.** | **Units** | **Comments** |
| VR01* | VFVR | FVR Voltage Output | 1.128 | 1.2 | 1.272 | V | |
| VR02* | TSTABLE | FVR Turn On Time | — | 200 | — | µs | |

* These parameters are characterized but not tested.

**TABLE 22-16: SHUNT REGULATOR SPECIFICATIONS (PIC16HV753 only)**

| Standard Operating Conditions (unless otherwise stated)<br>VDD = 3.0V, TA = 25°C | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Symbol** | **Characteristics** | **Min.** | **Typ.** | **Max.** | **Units** | **Comments** |
| SR01 | VSHUNT | Shunt Voltage | 4.75 | 5 | 5.5 | V | |
| | | | 4.70 | 5 | 5.5 | V | TA = -40°C |
| SR02 | ISHUNT | Shunt Current | 1 | — | 50 | mA | |
| SR03* | TSETTLE | Settling Time | — | — | 150 | ns | To 1% of final value |
| SR04 | CLOAD | Load Capacitance | 0.01 | — | 10 | µF | Bypass capacitor on VDD pin |
| SR05 | ΔISNT | Regulator operating current | — | 180 | — | µA | Includes band gap reference current |

* These parameters are characterized but not tested.

**TABLE 22-17: ANALOG-TO-DIGITAL CONVERTER (ADC) CHARACTERISTICS(1,2,3)**

| Standard Operating Conditions (unless otherwise stated)<br>VDD = 3.0V, TA = 25°C | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Sym.** | **Characteristic** | **Min.** | **Typ†** | **Max.** | **Units** | **Conditions** |
| AD01 | NR | Resolution | — | — | 10 | bit | |
| AD02 | EIL | Integral Error | — | — | ±1 | LSb | VREF = 3.0V |
| AD03 | EDL | Differential Error | — | — | ±1 | LSb | No missing codes<br>VREF = 3.0V |
| AD04 | EOFF | Offset Error | — | ±1.5 | ±3.0 | LSb | VREF = 3.0V |
| AD05 | EGN | Gain Error | — | — | ±1.0 | LSb | VREF = 3.0V |
| AD06 | VREF | Reference Voltage | 2.2<br>2.5 | —<br>— | —<br>VDD | V | Absolute minimum to ensure 1 LSb accuracy |
| AD07 | VAIN | Full-Scale Range | VSS | — | VREF | V | |
| AD08 | ZAIN | Recommended Impedance of Analog Voltage Source | — | — | 10 | kΩ | Can go higher if external 0.01 µF capacitor is present on input pin. |
| AD09* | IREF | VREF Input Current | 10 | — | 1000 | µA | During VAIN acquisition.<br>Based on differential of VHOLD to VAIN. |
| | | | — | — | 50 | µA | During A/D conversion cycle. |

* These parameters are characterized but not tested.

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

**Note 1:** Total Absolute Error includes integral, differential, offset and gain errors.

**2:** The ADC conversion result never decreases with an increase in the input voltage and has no missing codes.

**3:** See **Section 23.0 "DC and AC Characteristics Graphs and Charts"** for operating characterization.

**TABLE 22-18: ADC CONVERSION REQUIREMENTS**

| **Standard Operating Conditions (unless otherwise stated)**<br>$V_{DD}$ = 3.0V, $T_A$ = 25°C | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Sym.** | **Characteristic** | **Min.** | **Typ†** | **Max.** | **Units** | **Conditions** |
| AD130 * | TAD | ADC Internal FRC Oscillator Period | 3.0 | 6.0 | 9.0 | µs | At VDD = 2.5V |
| | | | 1.6 | 4.0 | 6.0 | µs | At VDD = 5.0V |
| | | ADC Clock Period | 1.6 | — | 9.0 | µs | FOSC-based, VREF ≥ 3.0V |
| | | | 3.0 | — | 9.0 | µs | TOSC-based, VREF full range(2) |
| AD131 | TCNV | Conversion Time (not including Acquisition Time)(1) | — | 11 | — | TAD | Set GO/$\overline{DONE}$ bit to conversion complete |
| AD132 * | TACQ | Acquisition Time | — | 11.5 | — | µs | |
| AD133 * | TAMP | Amplifier Settling Time | — | — | 5 | µs | |
| AD134 | TGO | Q4 to A/D Clock Start | — | TOSC/2 | — | — | |
| | THCD | Holding Capacitor Disconnect Time | —<br>— | 1/2 TAD<br>1/2 TAD + 1 TCY | —<br>— | — | FOSC-based<br>ADCS<2:0> = x11 (ADC FRC mode) |

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** The ADRES register may be read on the following TCY cycle. See **Section 12.4 "A/D Acquisition Requirements"** for minimum conditions.

**2:** Full range for PIC16HV753 powered by the shunt regulator is the 5V regulated voltage.

**FIGURE 22-10: PIC16F753/HV753 A/D CONVERSION TIMING (ADC CLOCK FOSC-BASED)**

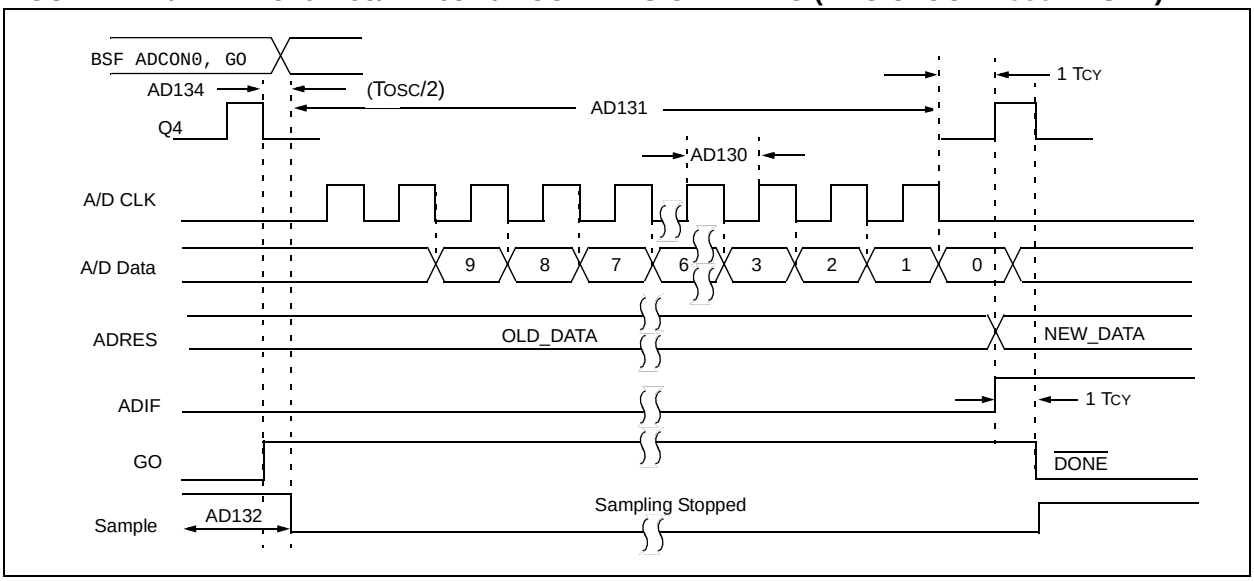


**FIGURE 22-11: PIC16F753/HV753 A/D CONVERSION TIMING (ADC CLOCK FROM FRC)**

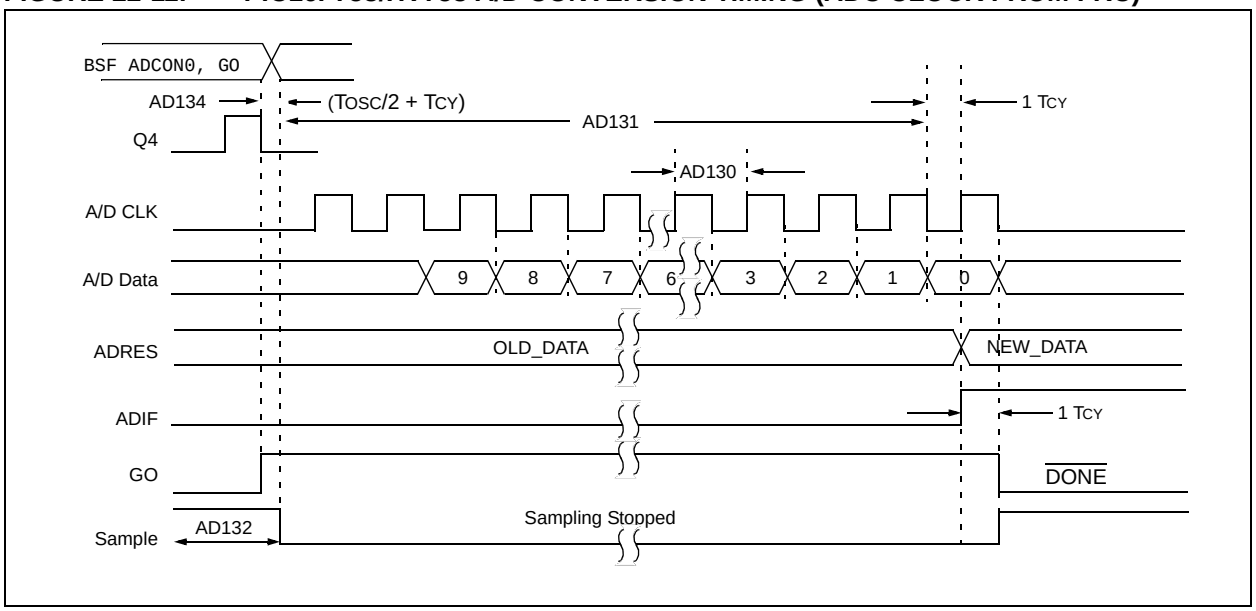



**TABLE 22-19: OPERATIONAL AMPLIFIER (OPA)**

| DC CHARACTERISTICS | | | Standard Operating Conditions (unless otherwise stated): VDD = 3.0 Temperature 25°C, High-Power Mode | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Symbol** | **Parameters** | **Min.** | **Typ†** | **Max.** | **Units** | **Conditions** |
| OPA12 | GBWP | Gain Bandwidth Product | — | 3 | — | MHz | |
| OPA13* | TON | Turn on Time | — | — | 10 | µs | |
| OPA14* | PM | Phase Margin | — | 60 | — | degrees | |
| OPA15* | SR | Slew Rate | 2 | — | — | V/µs | |

* 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 OSC1 pin. When an external clock input is used, the "max" cycle time limit is "DC" (no clock) for all devices.

## 23.0 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.

| | |
|---|---|
| **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.**

**FIGURE 23-1: IDD TYPICAL, EC OSCILLATOR, MEDIUM-POWER MODE, PIC16HV753 ONLY**

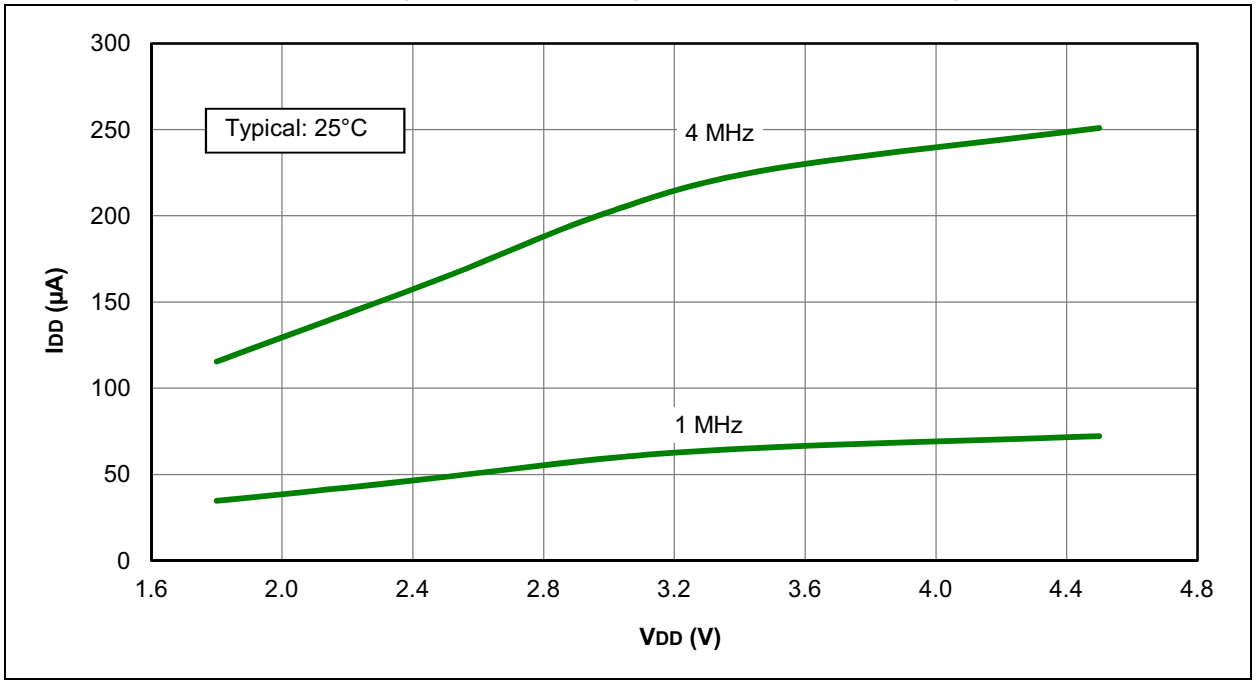



**FIGURE 23-2: IDD MAXIMUM, EC OSCILLATOR, MEDIUM-POWER MODE, PIC16HV753 ONLY**

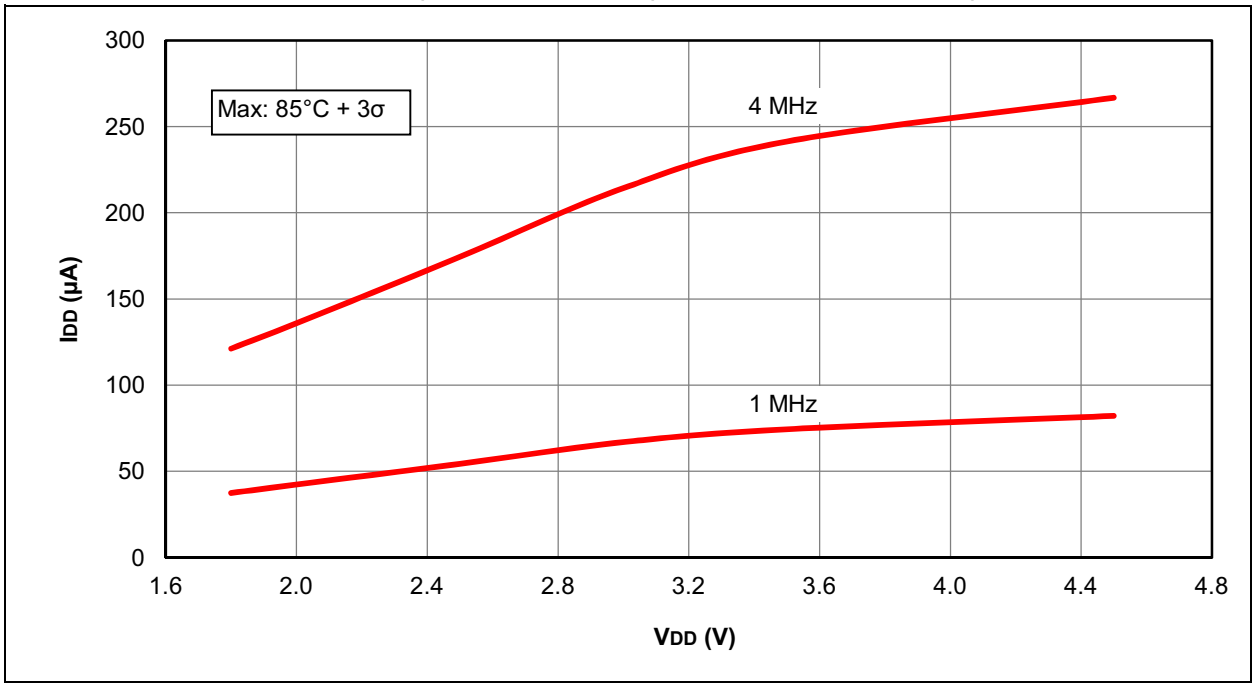


**FIGURE 23-3: IDD TYPICAL, EC OSCILLATOR, MEDIUM-POWER MODE, PIC16F753 ONLY**

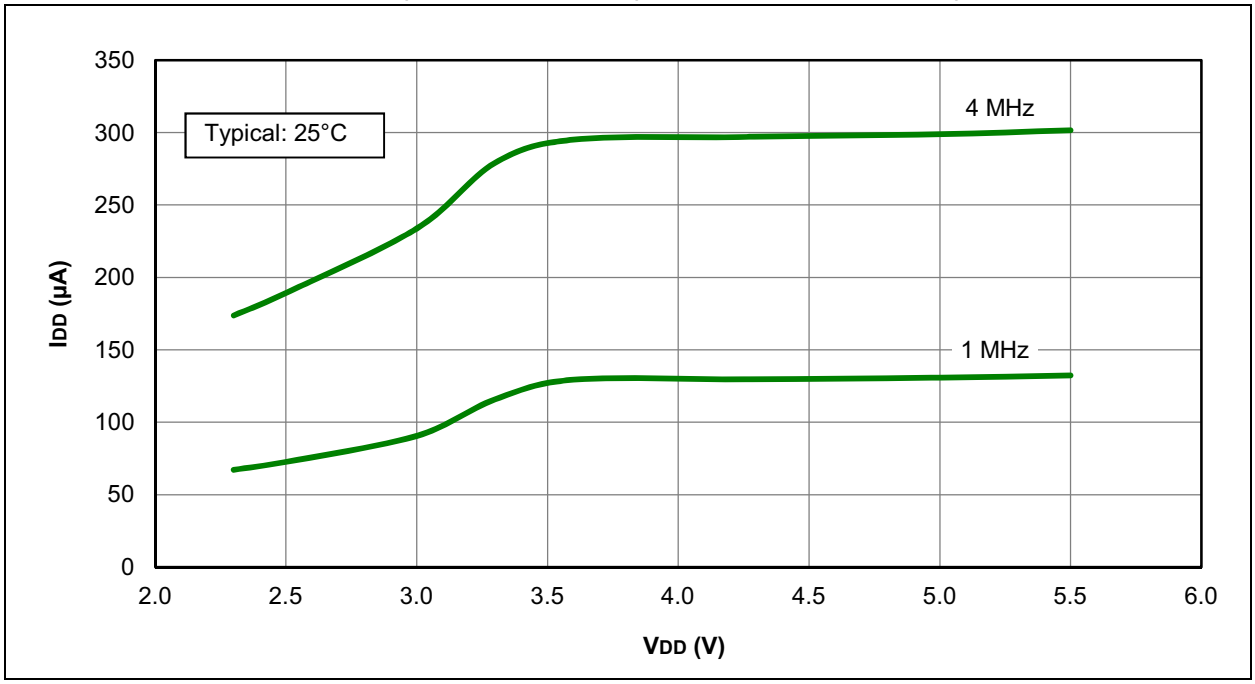



**FIGURE 23-4: IDD MAXIMUM, EC OSCILLATOR, MEDIUM-POWER MODE, PIC16F753 ONLY**

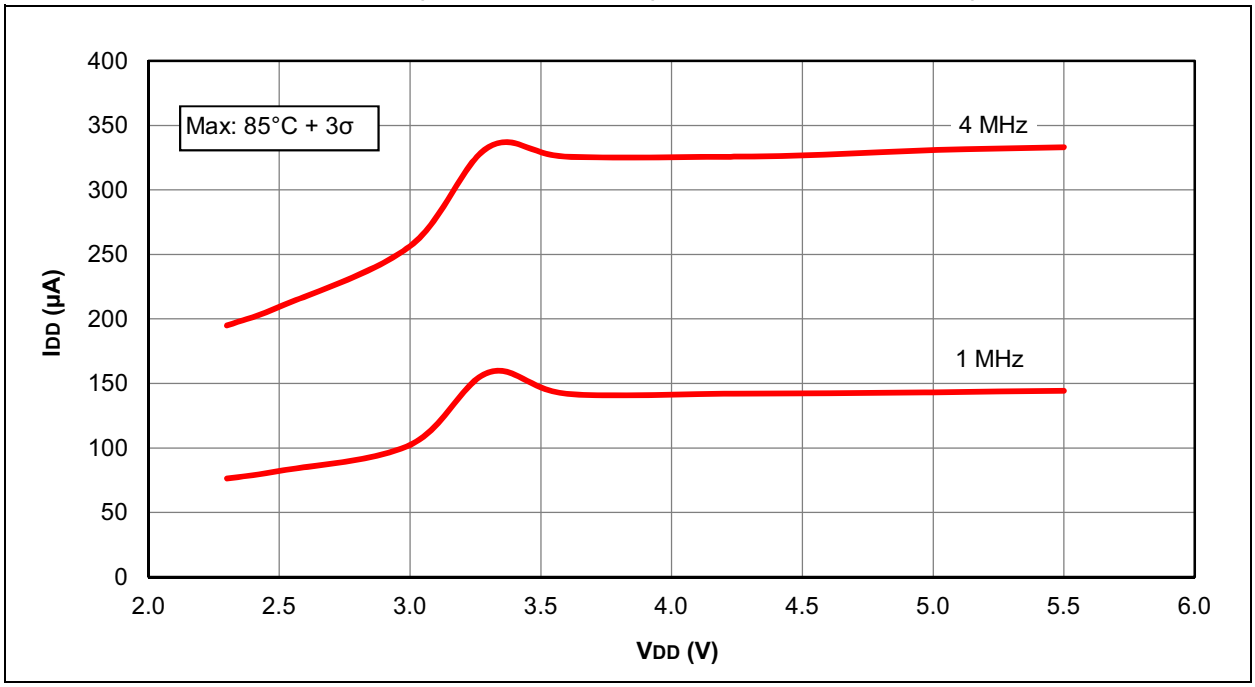


**FIGURE 23-5: IDD TYPICAL, EC OSCILLATOR, HIGH-POWER MODE, PIC16HV753 ONLY**

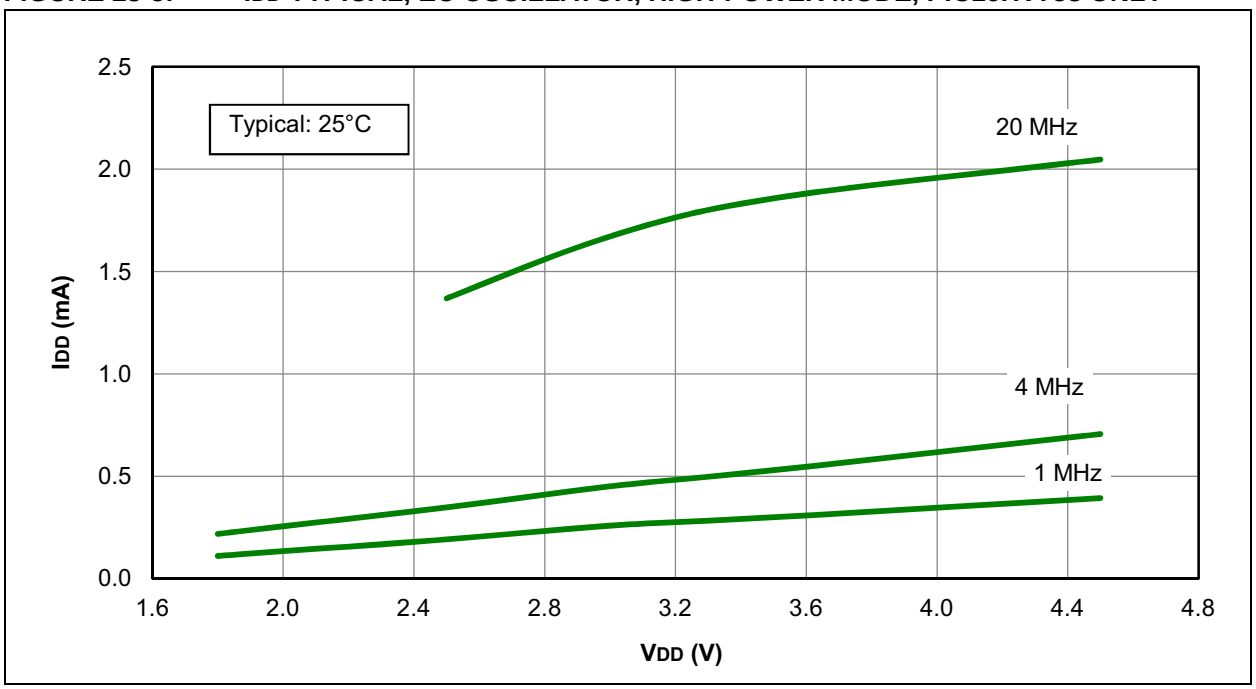



**FIGURE 23-6: IDD MAXIMUM, EC OSCILLATOR, HIGH-POWER MODE, PIC16HV753 ONLY**

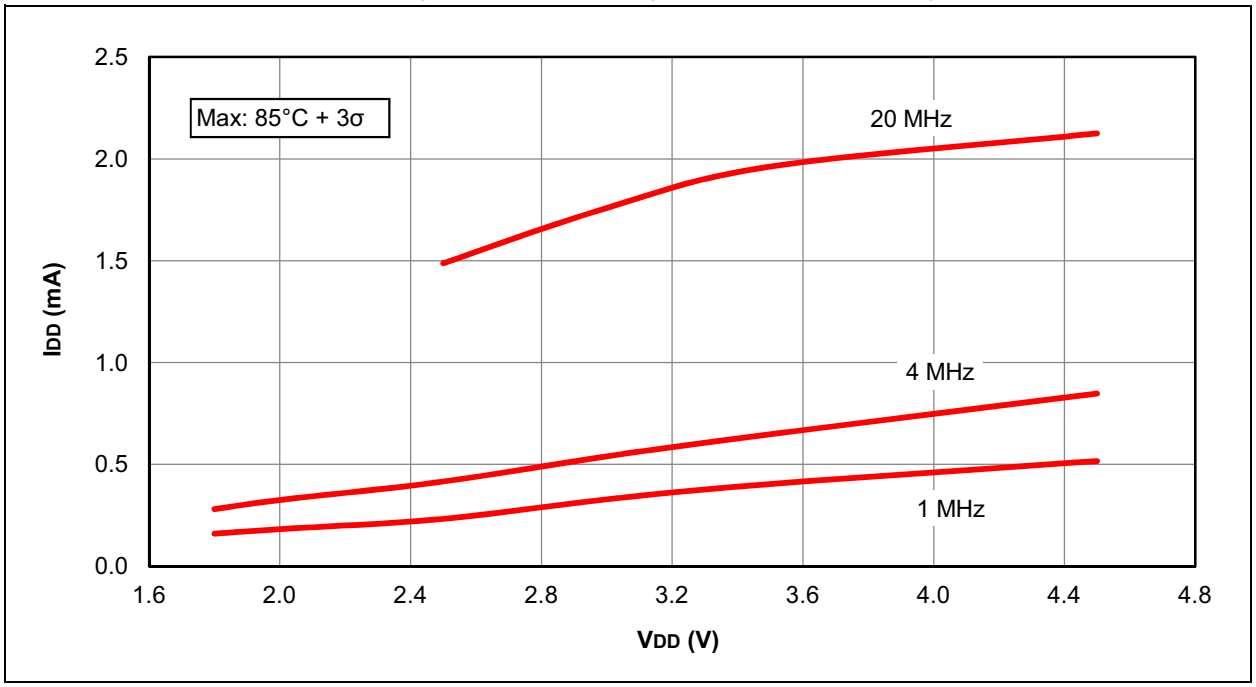


**FIGURE 23-7: IDD TYPICAL, EC OSCILLATOR, HIGH-POWER MODE, PIC16F753 ONLY**

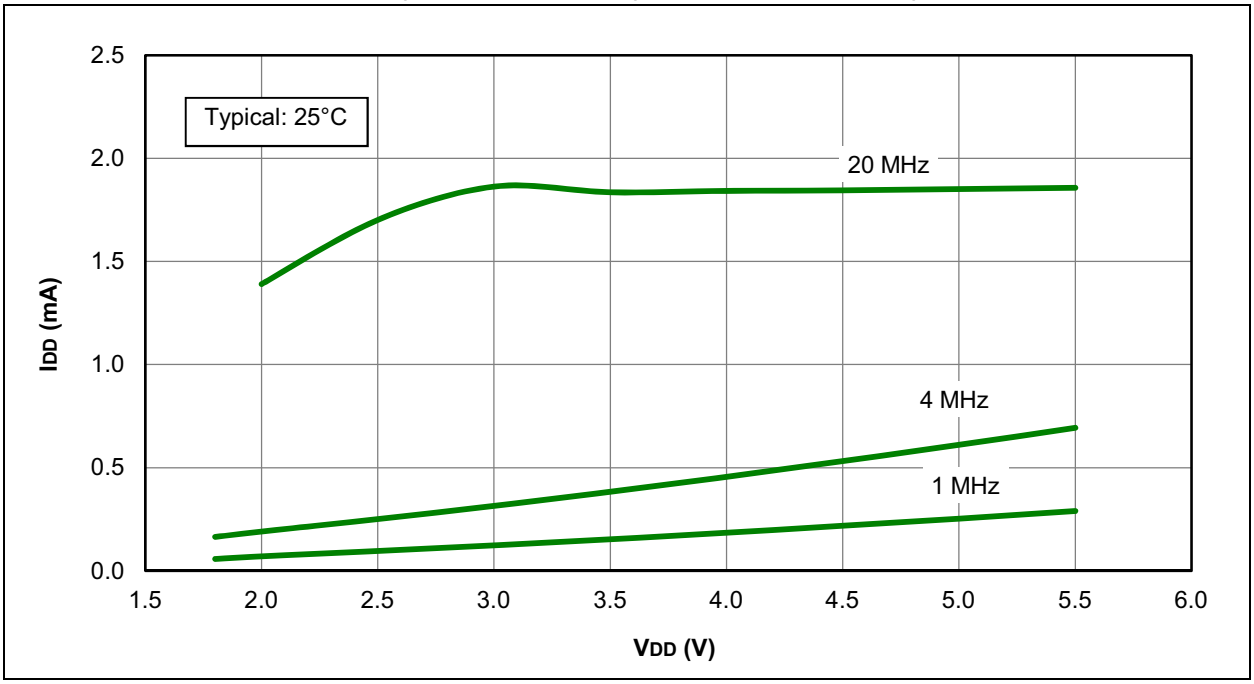



**FIGURE 23-8: IDD MAXIMUM, EC OSCILLATOR, HIGH-POWER MODE, PIC16F753 ONLY**

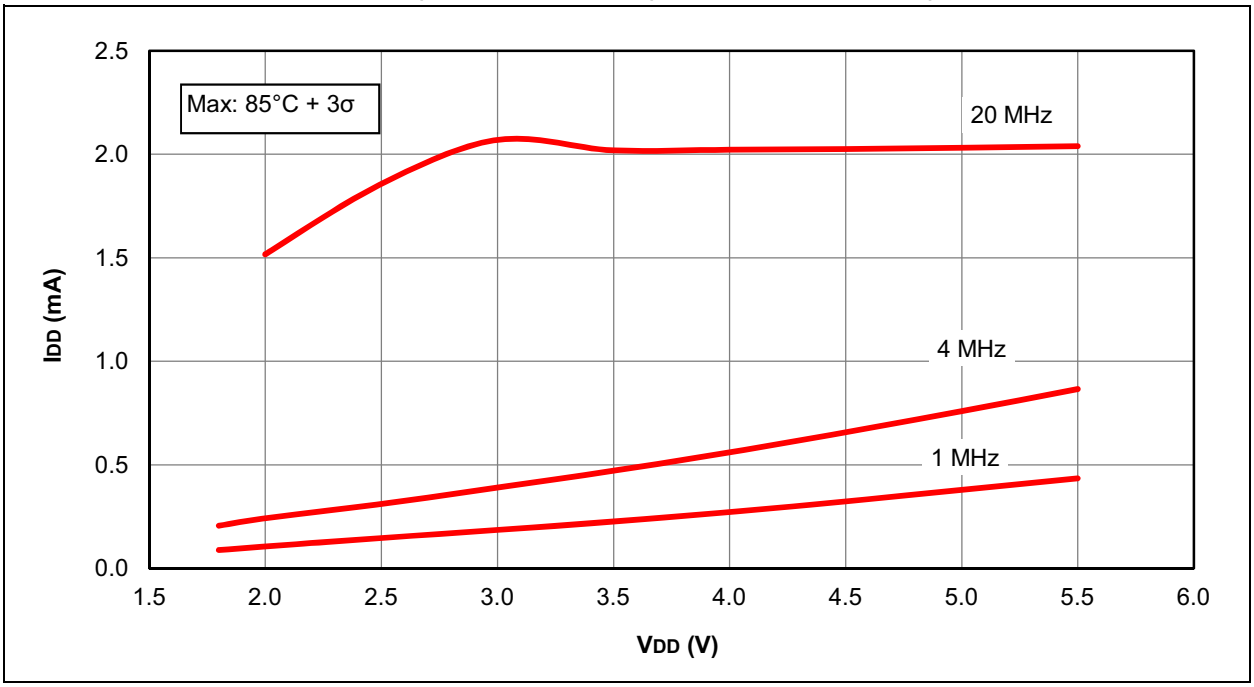


**FIGURE 23-9: IDD, LFINTOSC, FOSC = 31 kHz, PIC16HV753 ONLY**

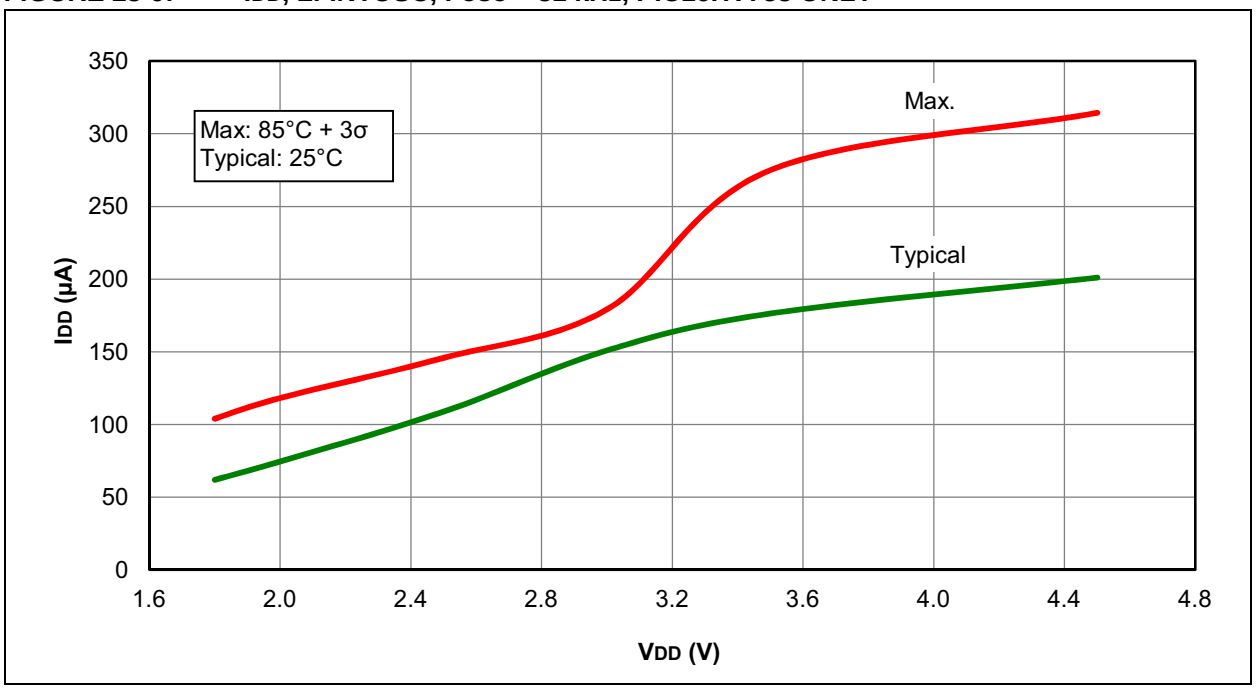



**FIGURE 23-10: IDD, LFINTOSC, FOSC = 31 kHz, PIC16F753 ONLY**

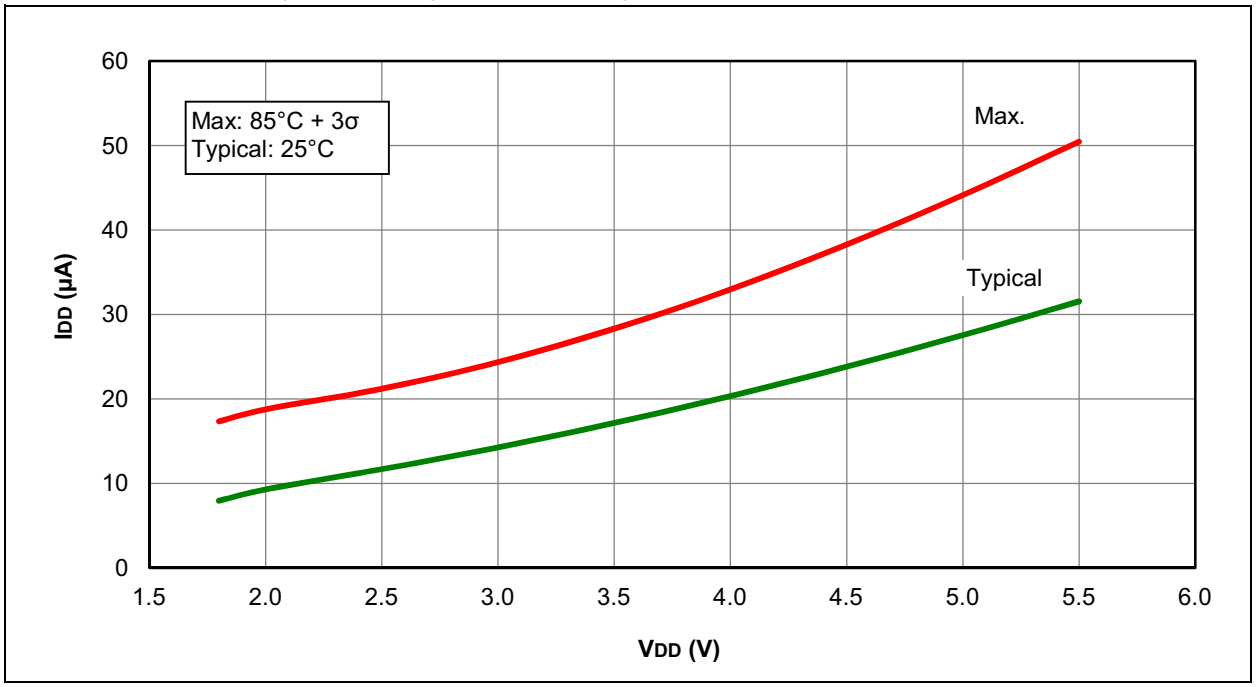


**FIGURE 23-11: IDD TYPICAL, HFINTOSC, PIC16HV753 ONLY**

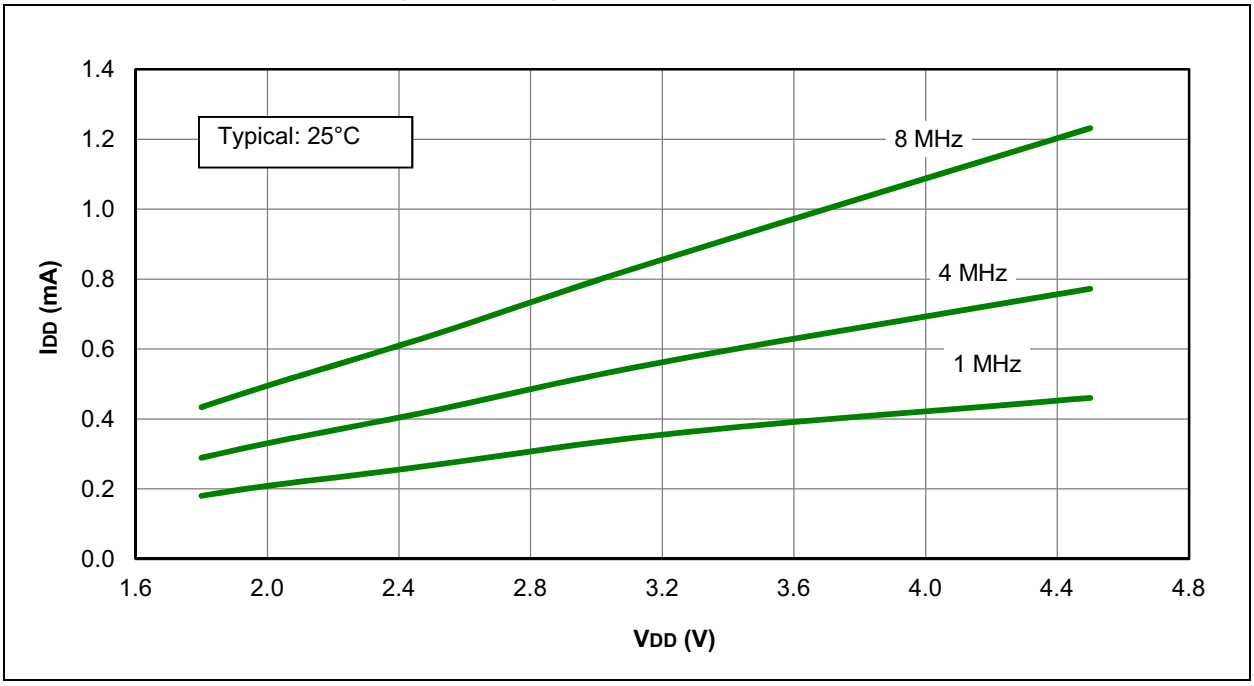



**FIGURE 23-12: IDD MAXIMUM, HFINTOSC, PIC16HV753 ONLY**

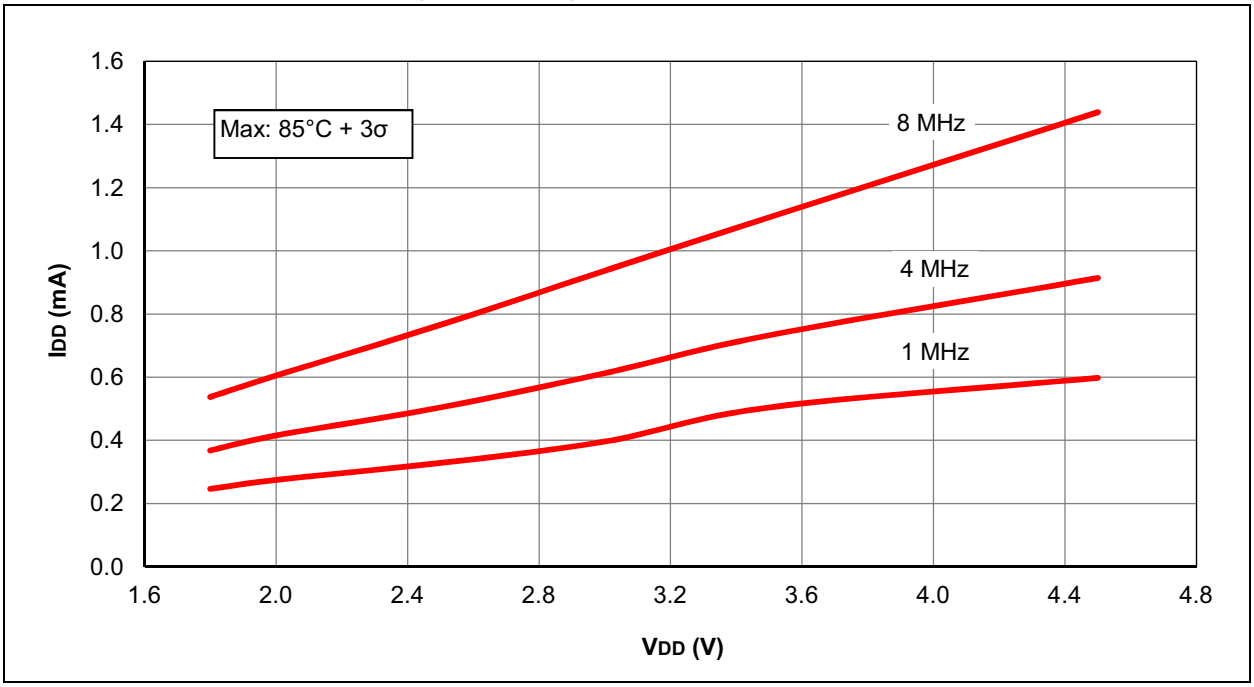


**FIGURE 23-13: IDD MAXIMUM, HFINTOSC, PIC16F753 ONLY**

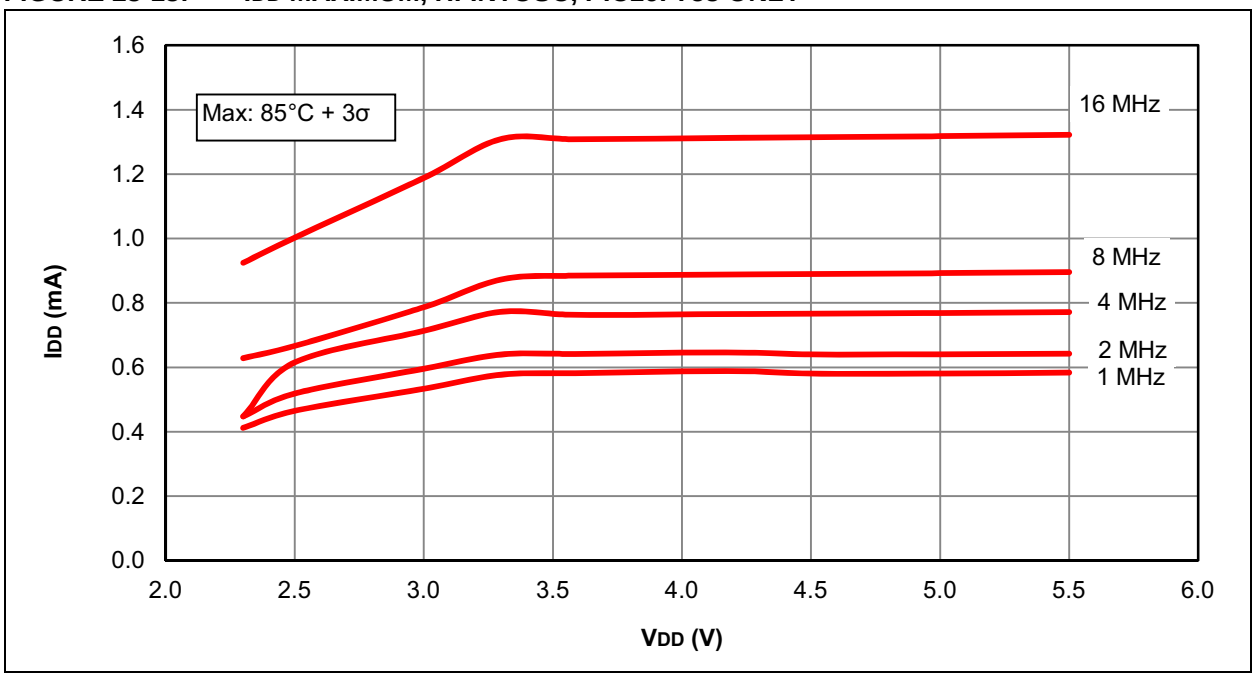


**FIGURE 23-14: IPD BASE, LOW-POWER SLEEP MODE, PIC16HV753 ONLY**

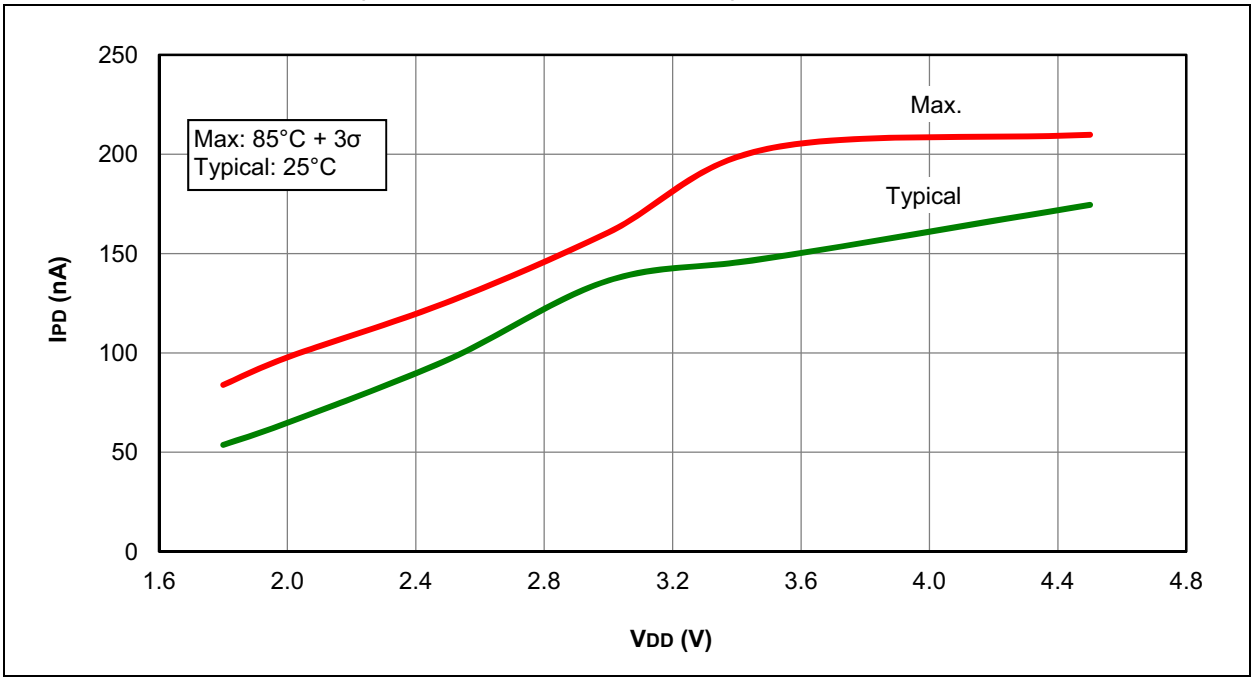



**FIGURE 23-15: IPD BASE, LOW-POWER SLEEP MODE, PIC16F753 ONLY**

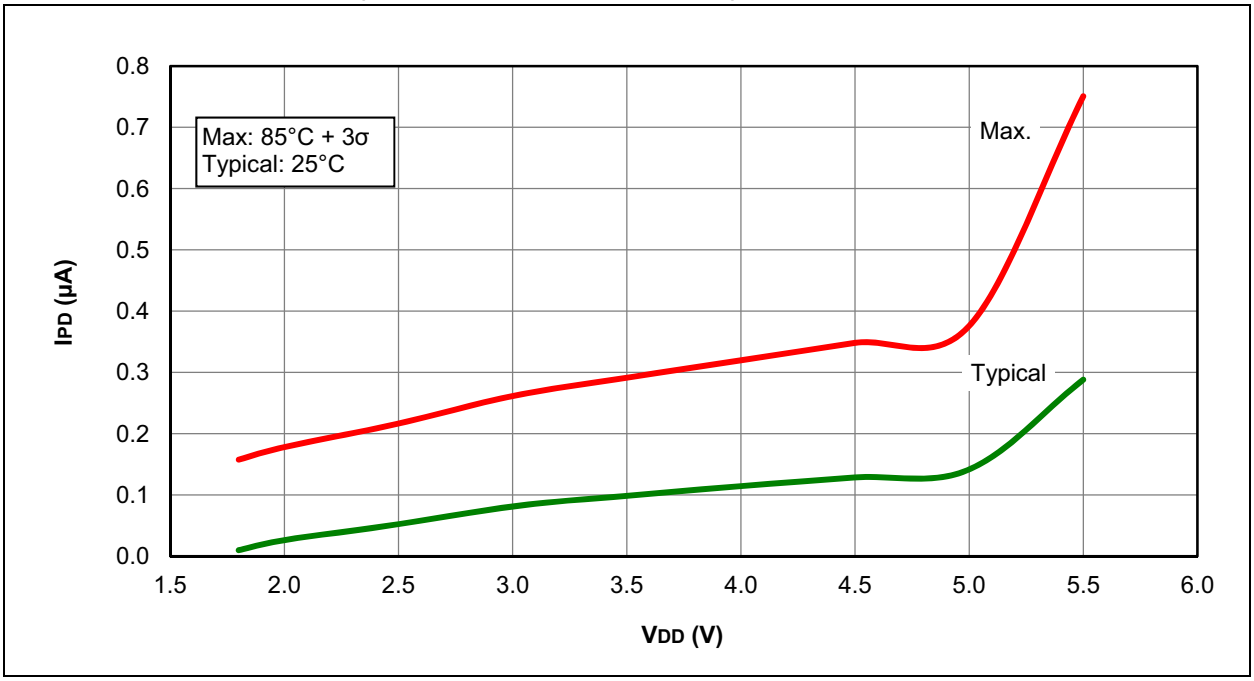


**FIGURE 23-16: IPD, WATCHDOG TIMER (WDT), PIC16HV753 ONLY**

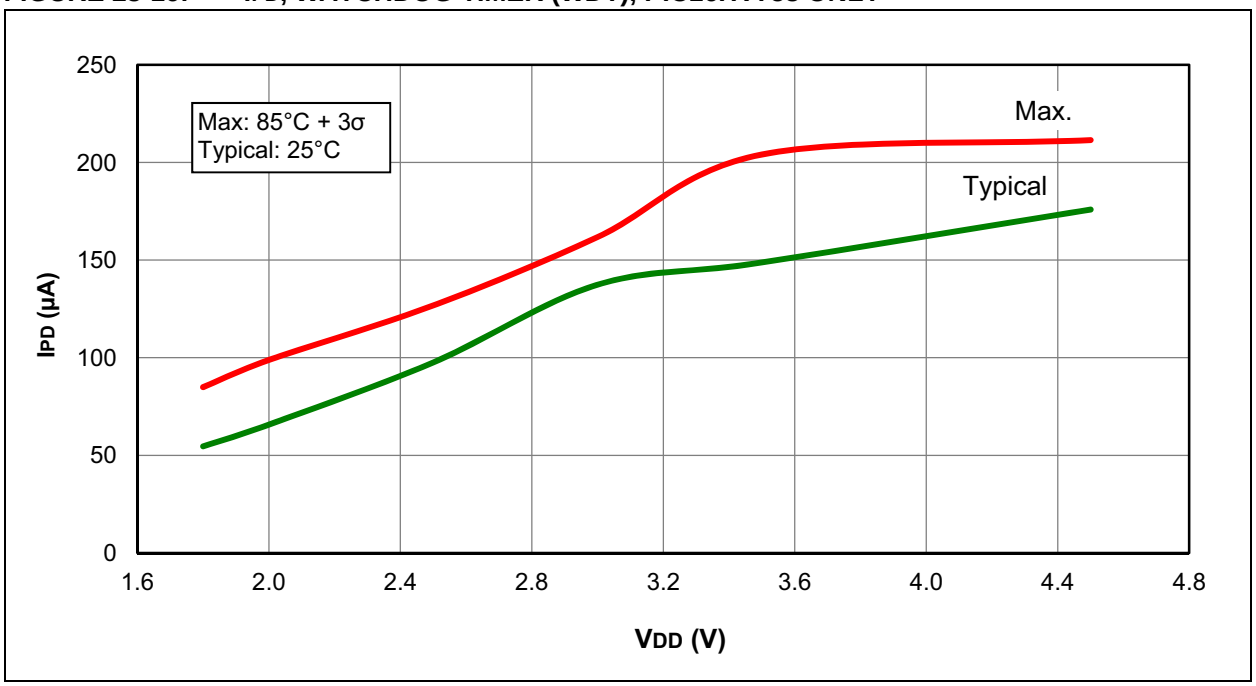



**FIGURE 23-17: IPD, WATCHDOG TIMER (WDT), PIC16F753 ONLY**

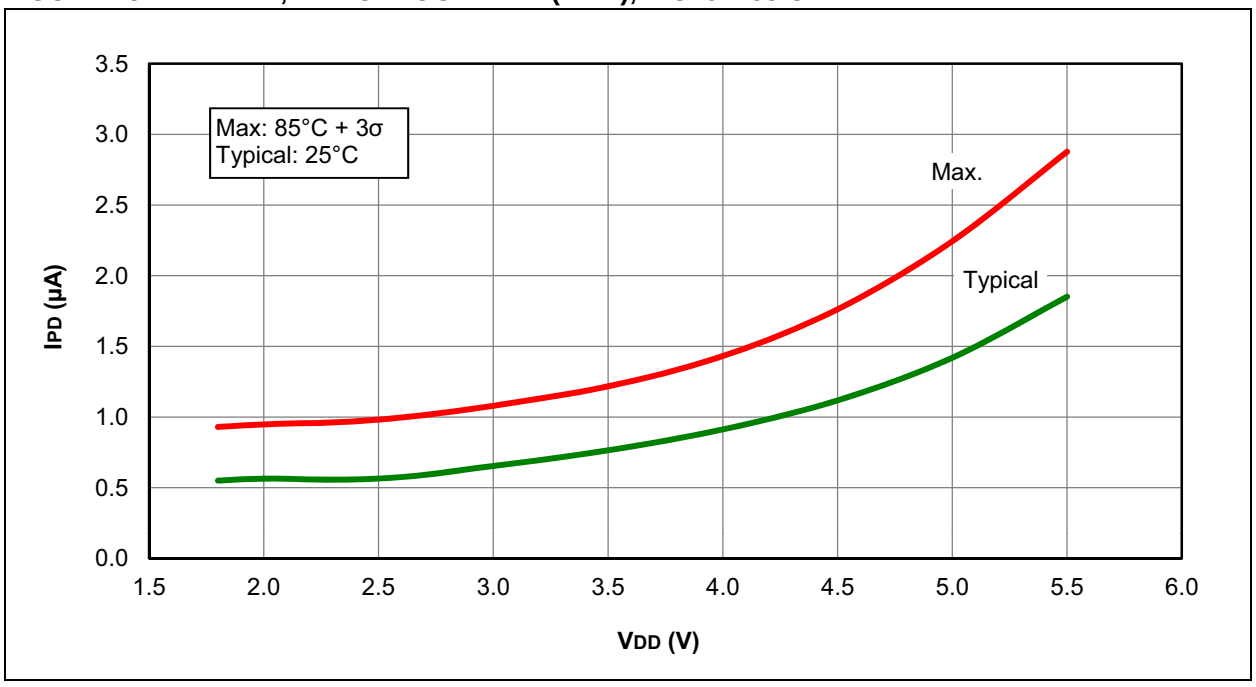


**FIGURE 23-18: IPD, FIXED VOLTAGE REFERENCE (FVR), PIC16HV753 ONLY**

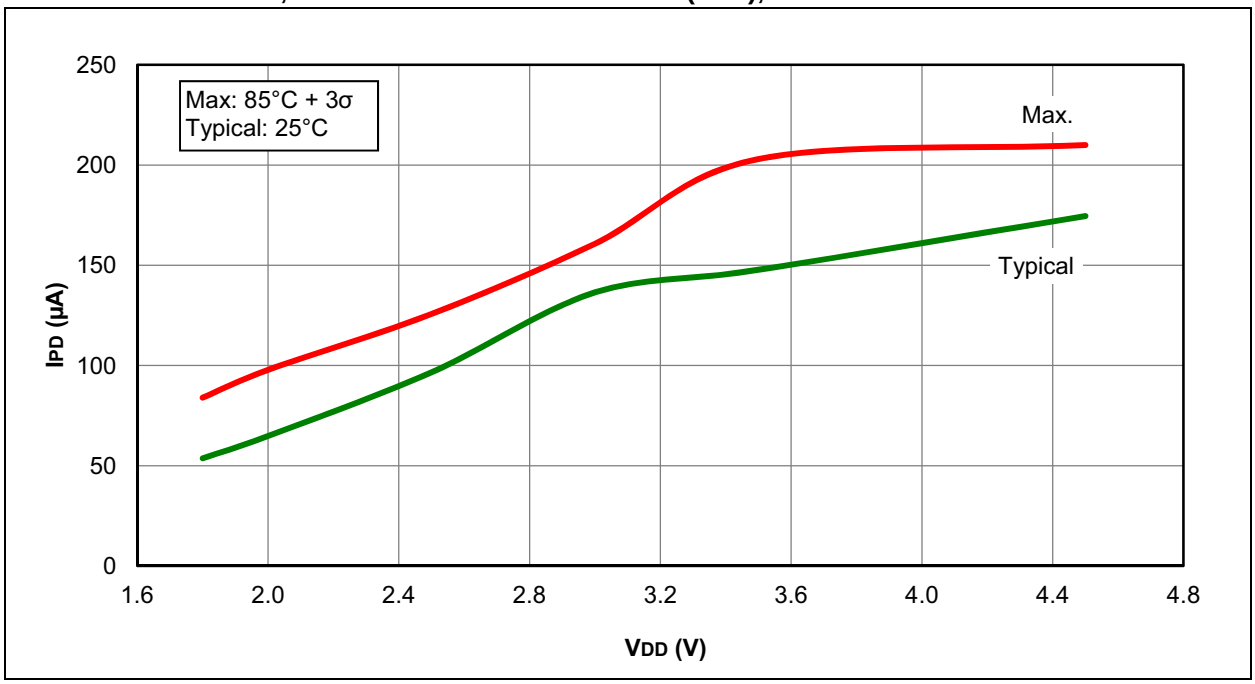



**FIGURE 23-19: IPD, FIXED VOLTAGE REFERENCE (FVR), PIC16F753 ONLY**

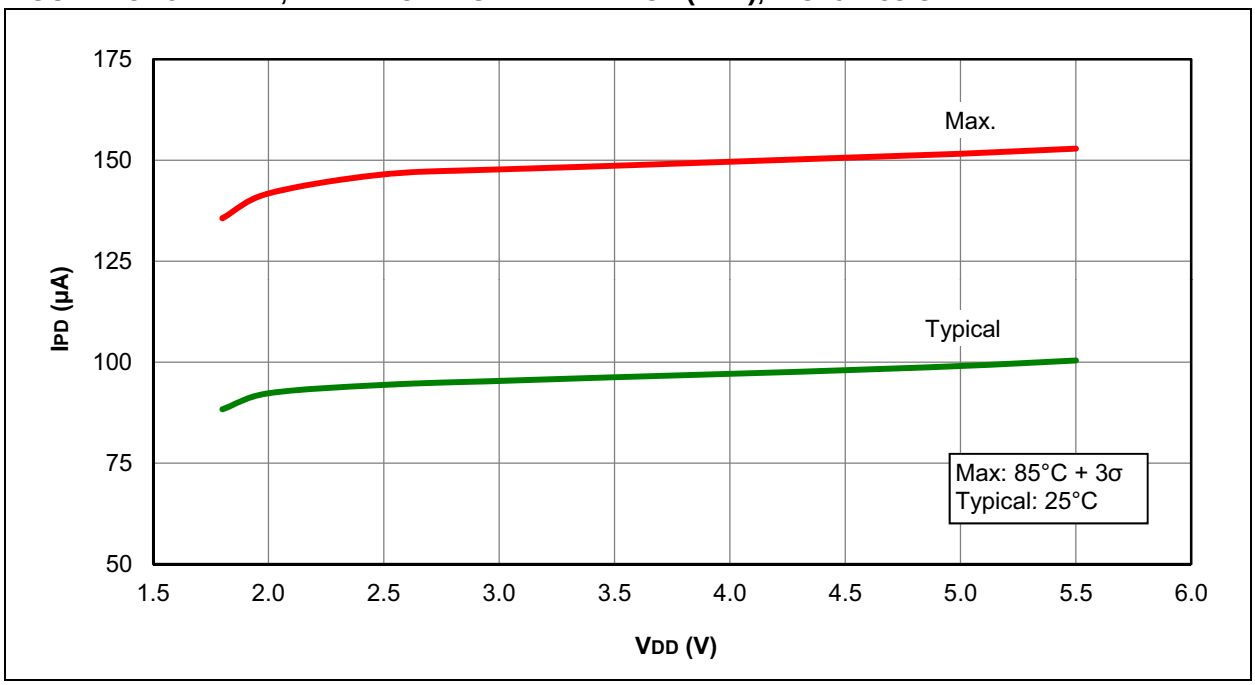


**FIGURE 23-20: IPD, BROWN-OUT RESET (BOR), PIC16HV753 ONLY**

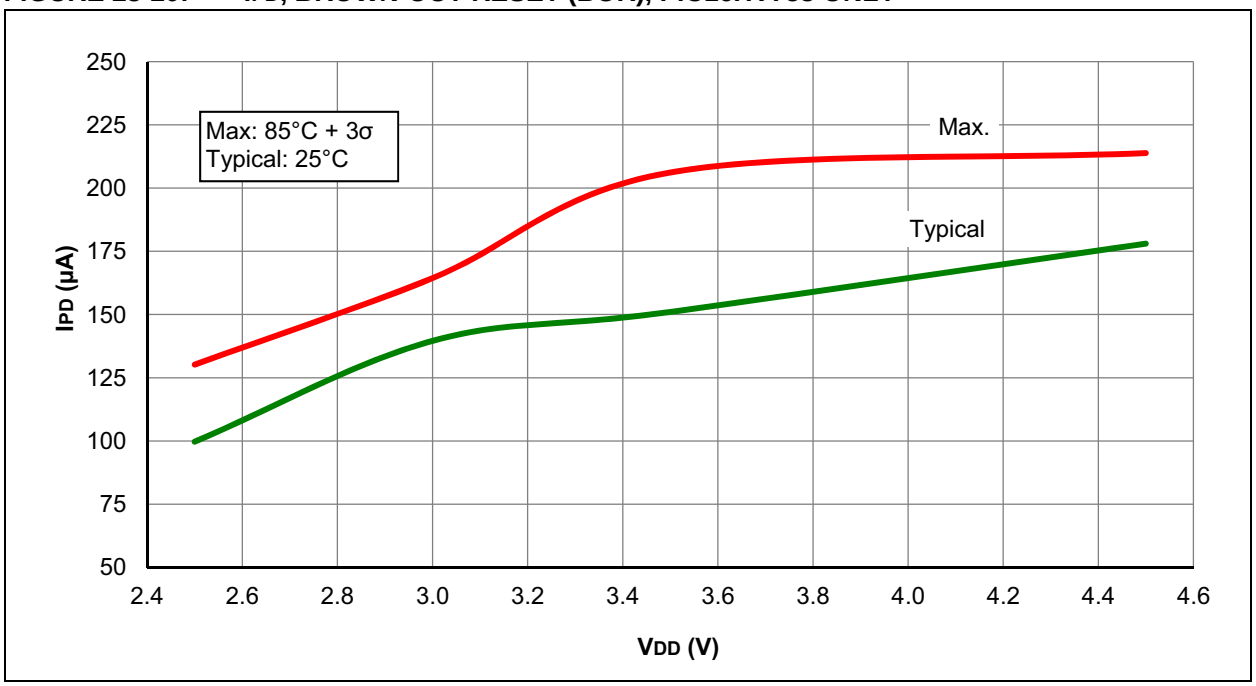



**FIGURE 23-21: IPD, BROWN-OUT RESET (BOR), PIC16F753 ONLY**

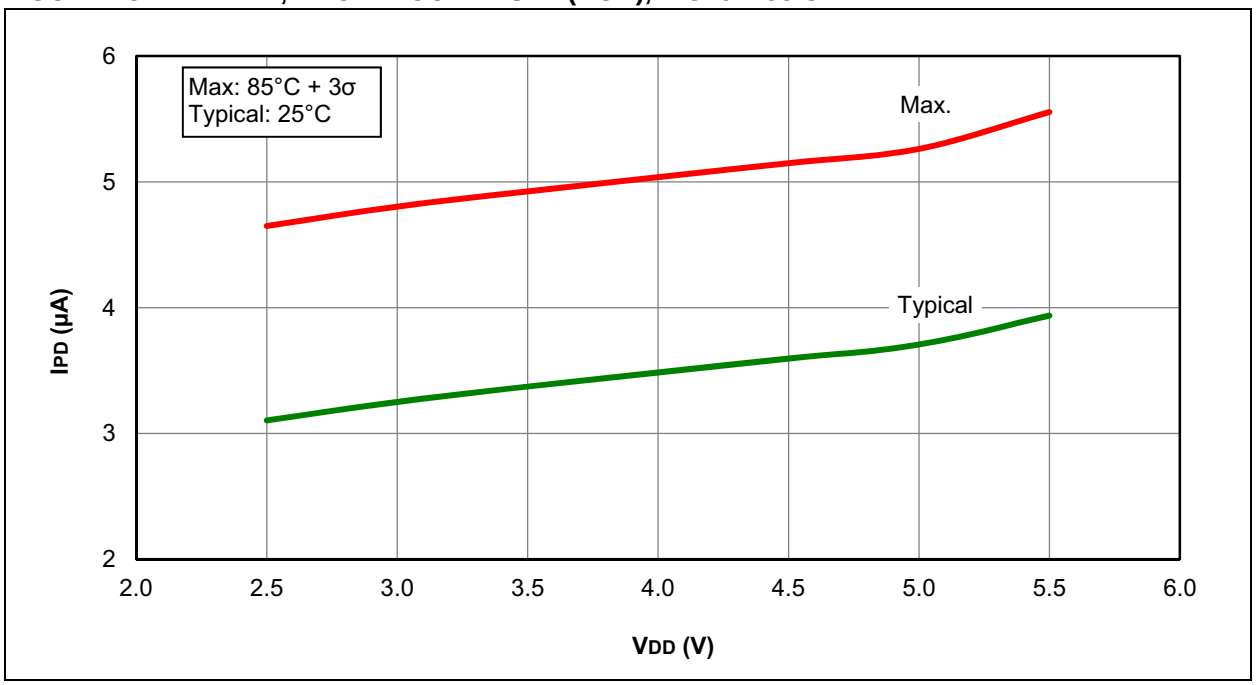


**FIGURE 23-22: IPD, DAC, PIC16HV753 ONLY**

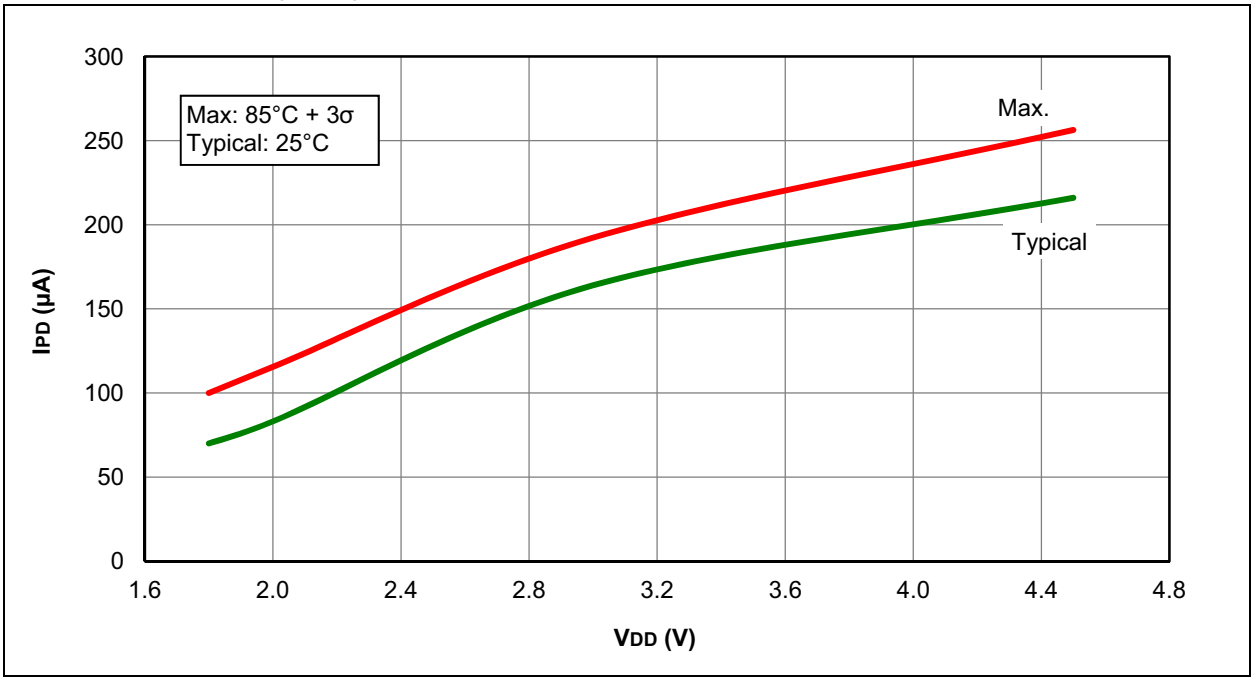



**FIGURE 23-23: IPD, DAC, PIC16F753 ONLY**

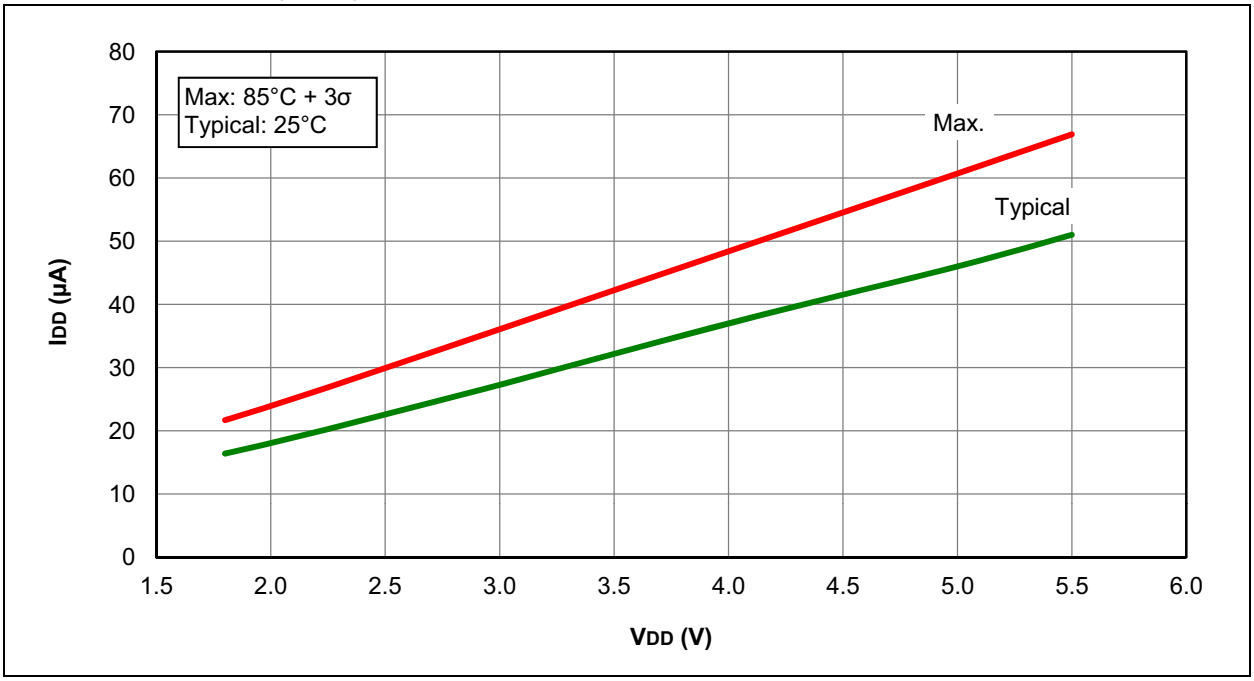


**FIGURE 23-24: IPD, TIMER1 OSCILLATOR, FOSC = 32 kHz, PIC16HV753 ONLY**

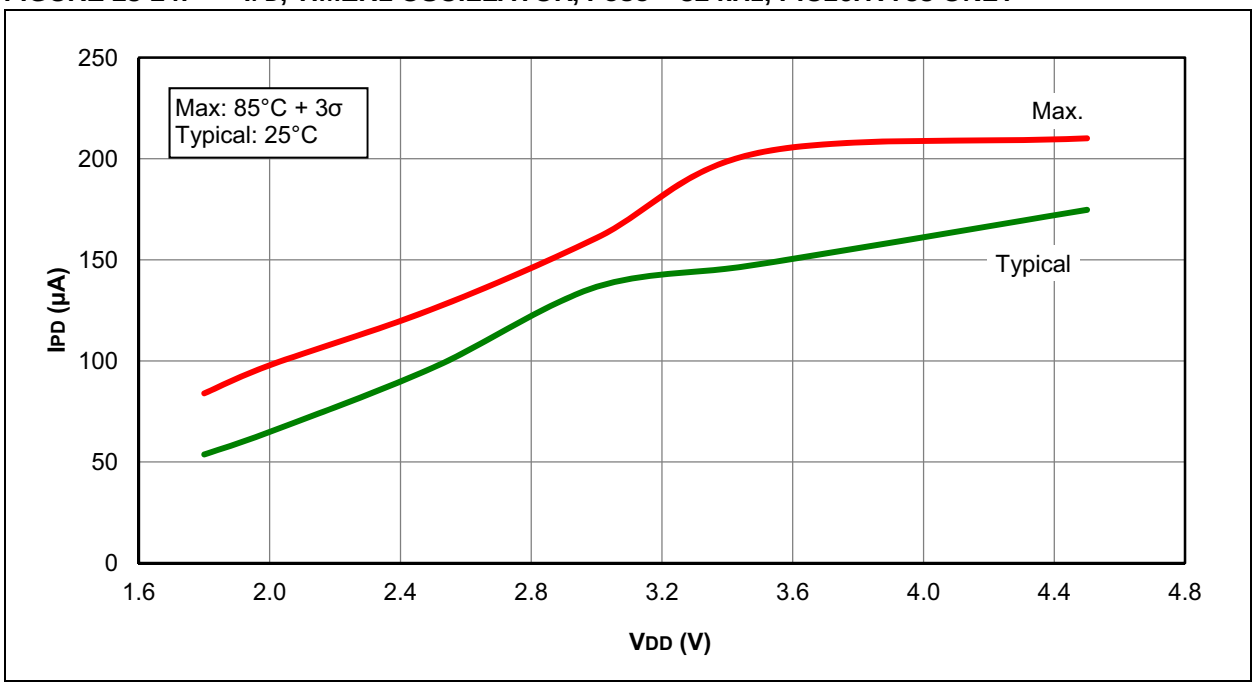



**FIGURE 23-25: IPD, TIMER1 OSCILLATOR, FOSC = 32 kHz, PIC16F753 ONLY**

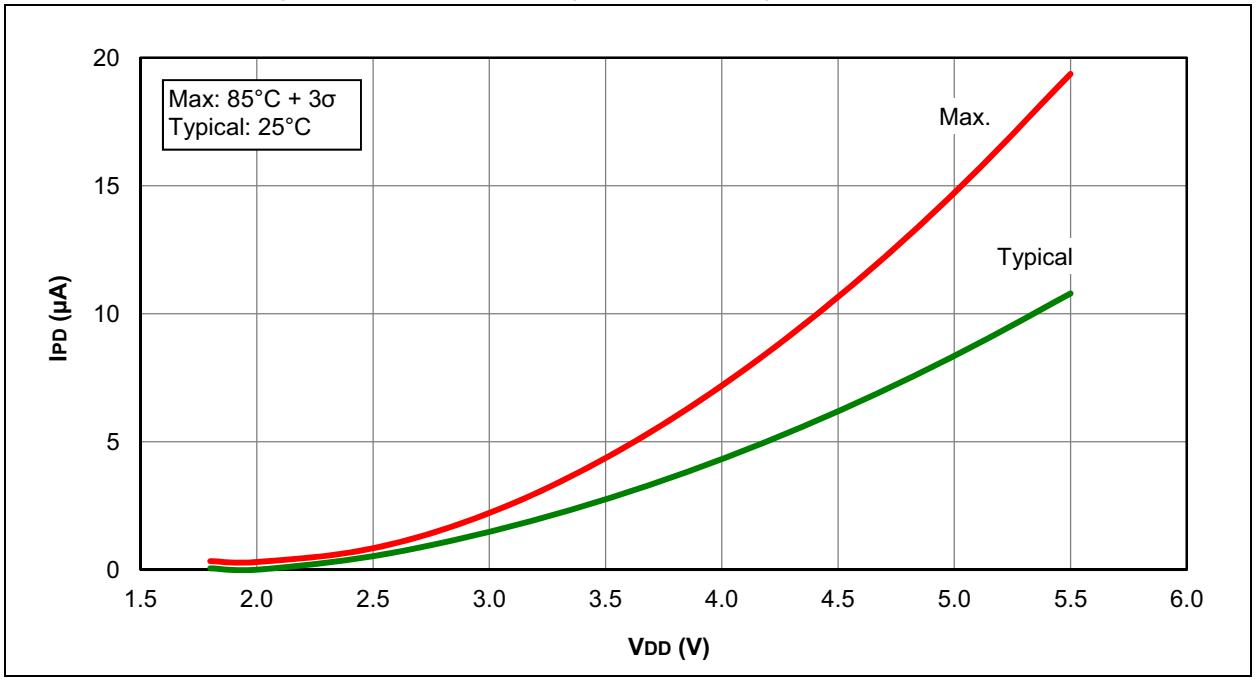


**FIGURE 23-26: IPD, OP AMP, PIC16HV753 ONLY**

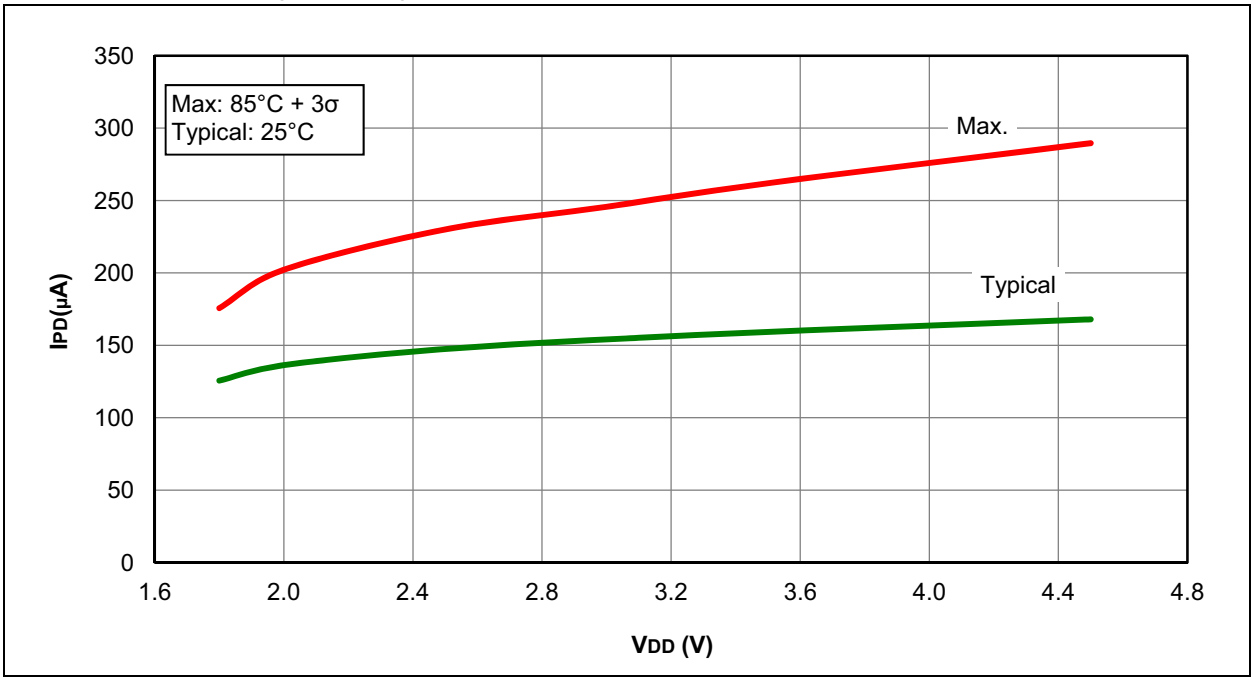



**FIGURE 23-27: IPD, OP AMP, PIC16F753 ONLY**

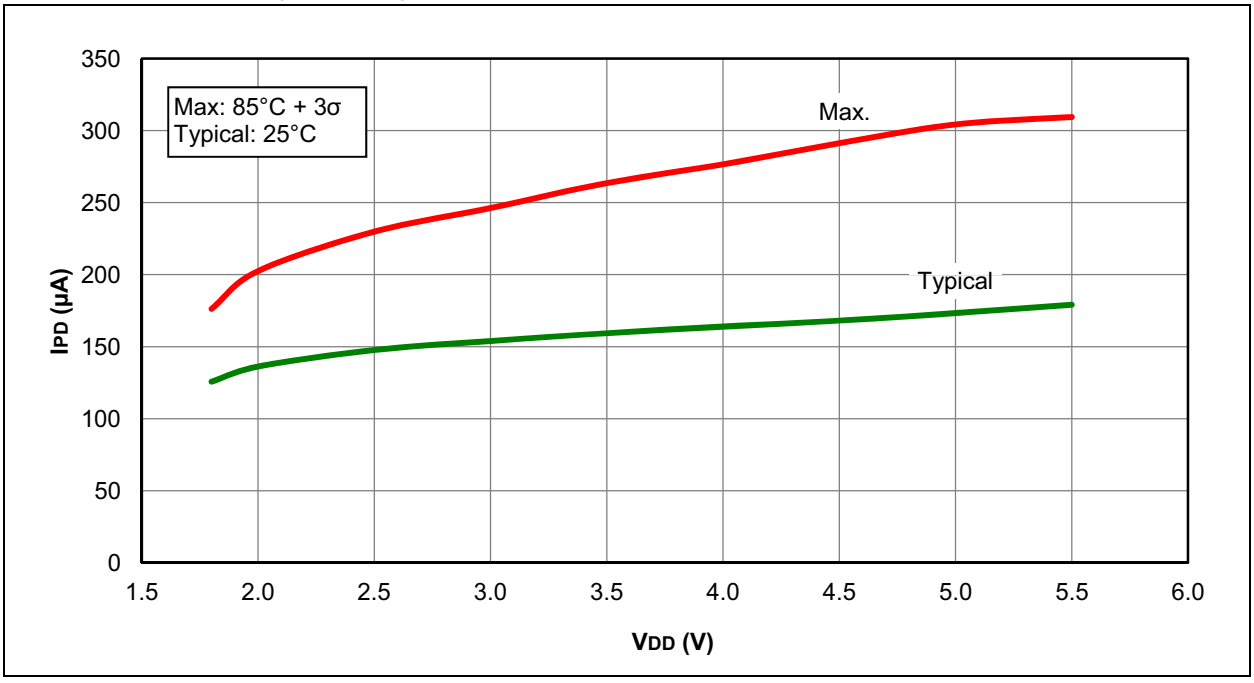


**FIGURE 23-28: IPD, ADC NO CONVERSION IN PROGRESS, PIC16HV753 ONLY**

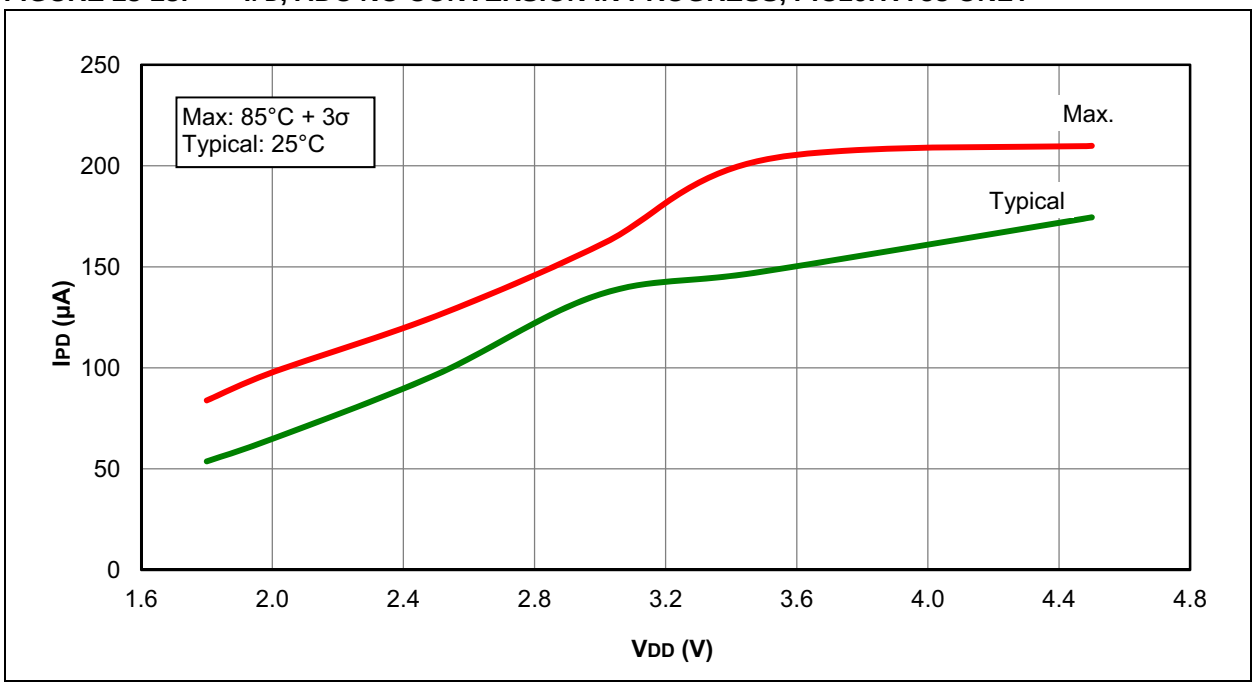



**FIGURE 23-29: IPD, ADC NO CONVERSION IN PROGRESS, PIC16F753 ONLY**

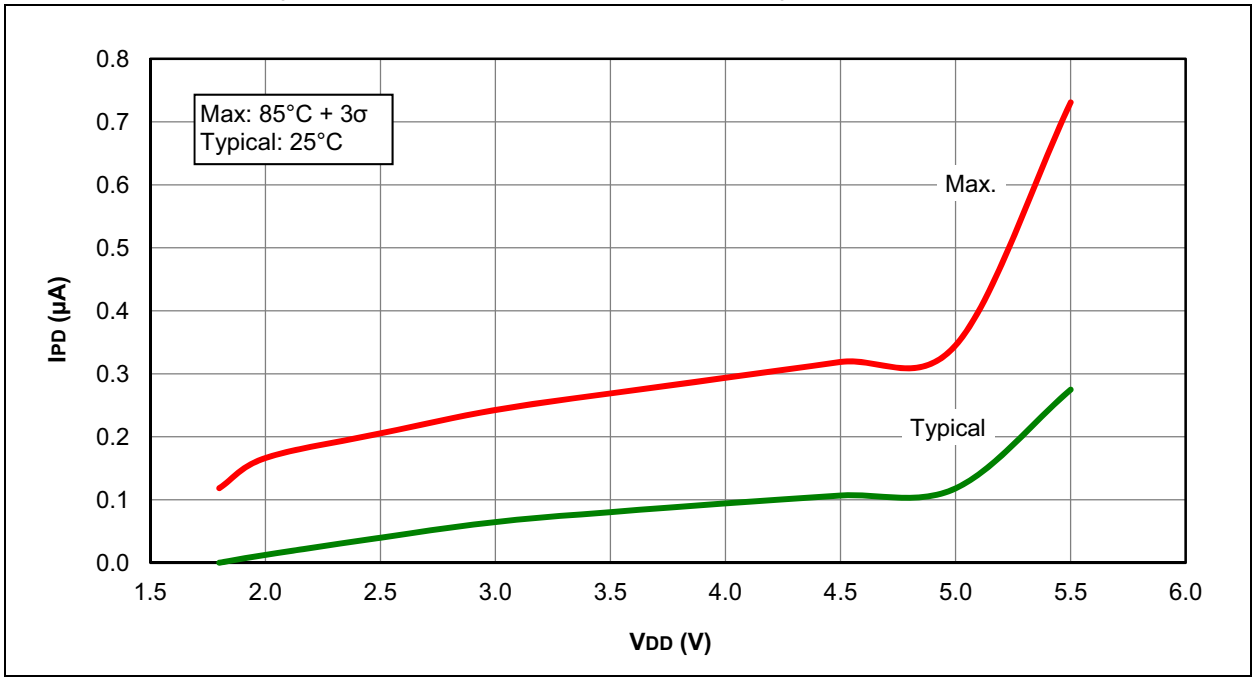


**FIGURE 23-30: IPD, COMPARATOR, LOW-POWER MODE, PIC16HV753 ONLY**

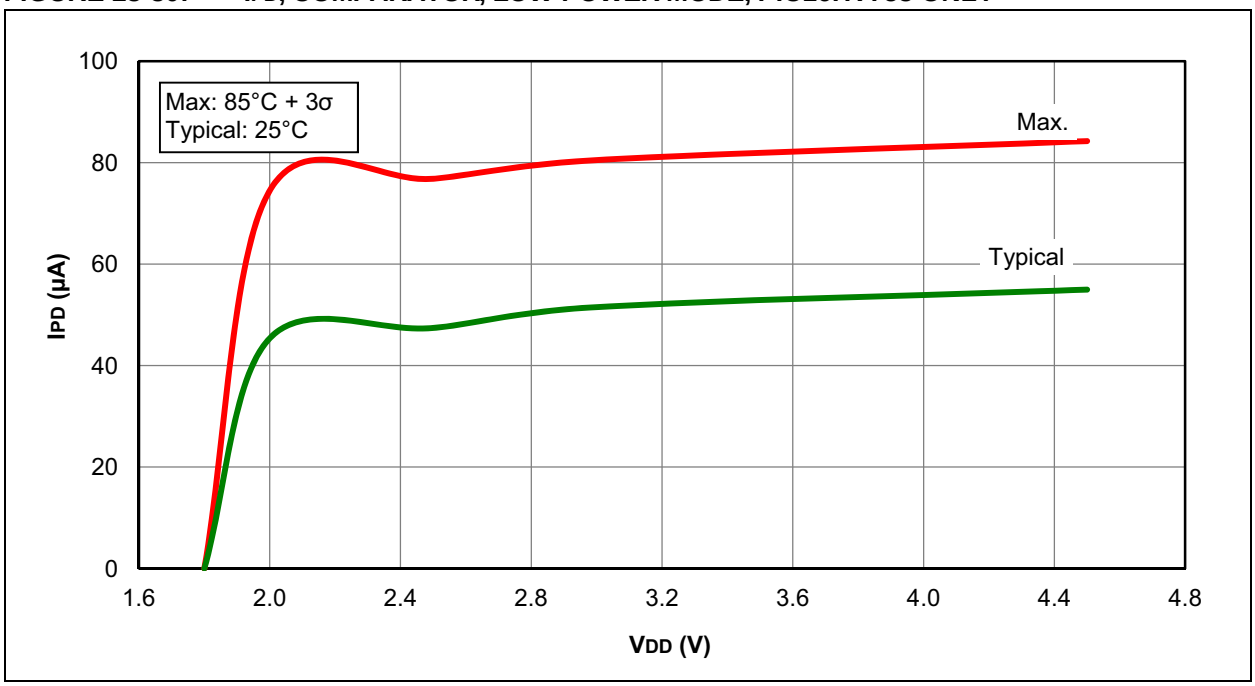



**FIGURE 23-31: IPD, COMPARATOR, LOW-POWER MODE, PIC16F753 ONLY**

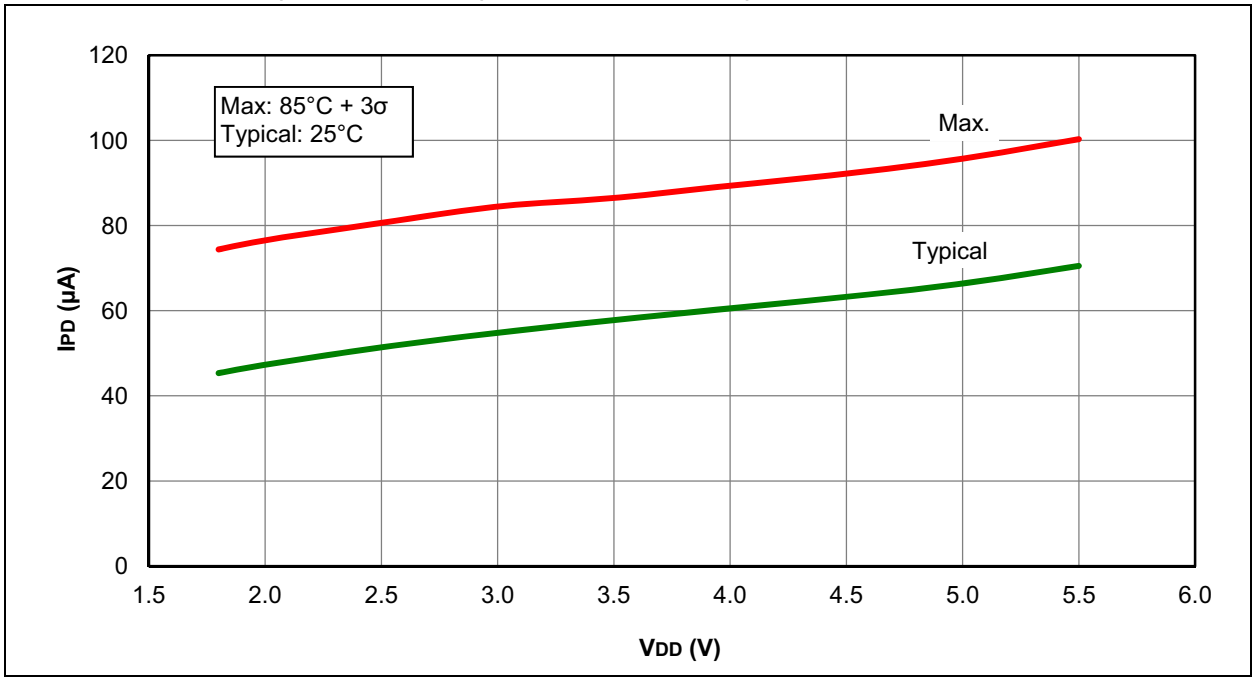


**FIGURE 23-32: IPD, COMPARATOR, HIGH-POWER MODE, PIC16HV753 ONLY**

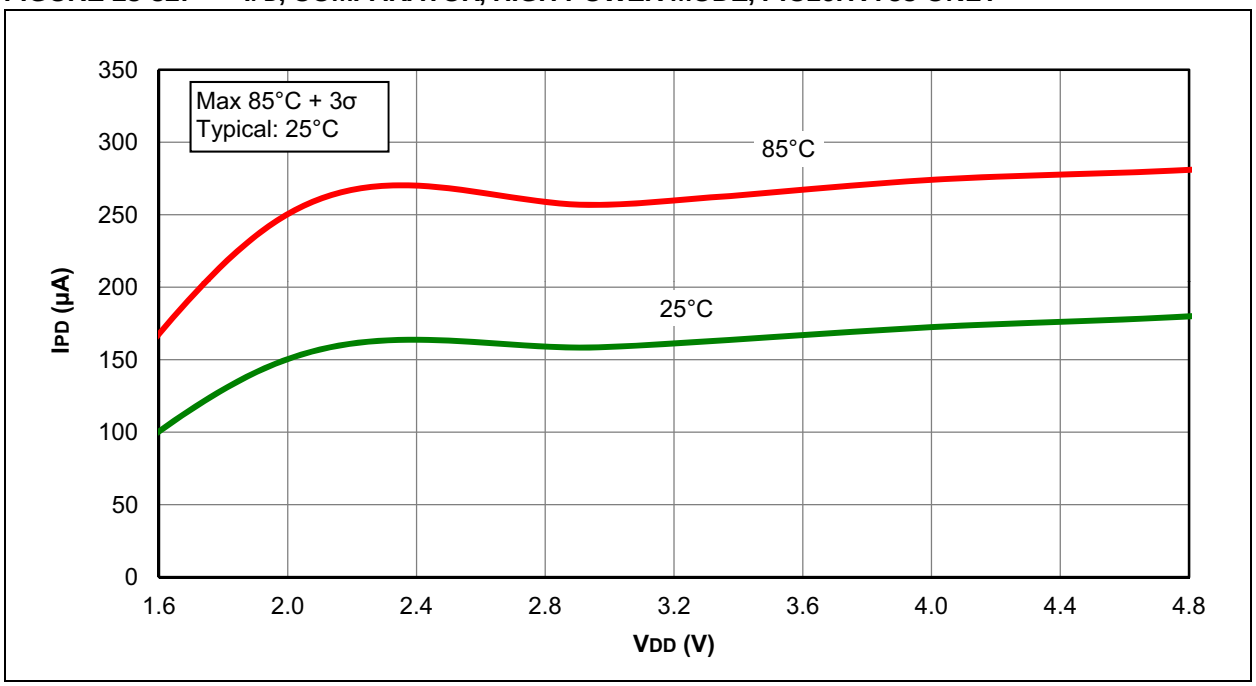
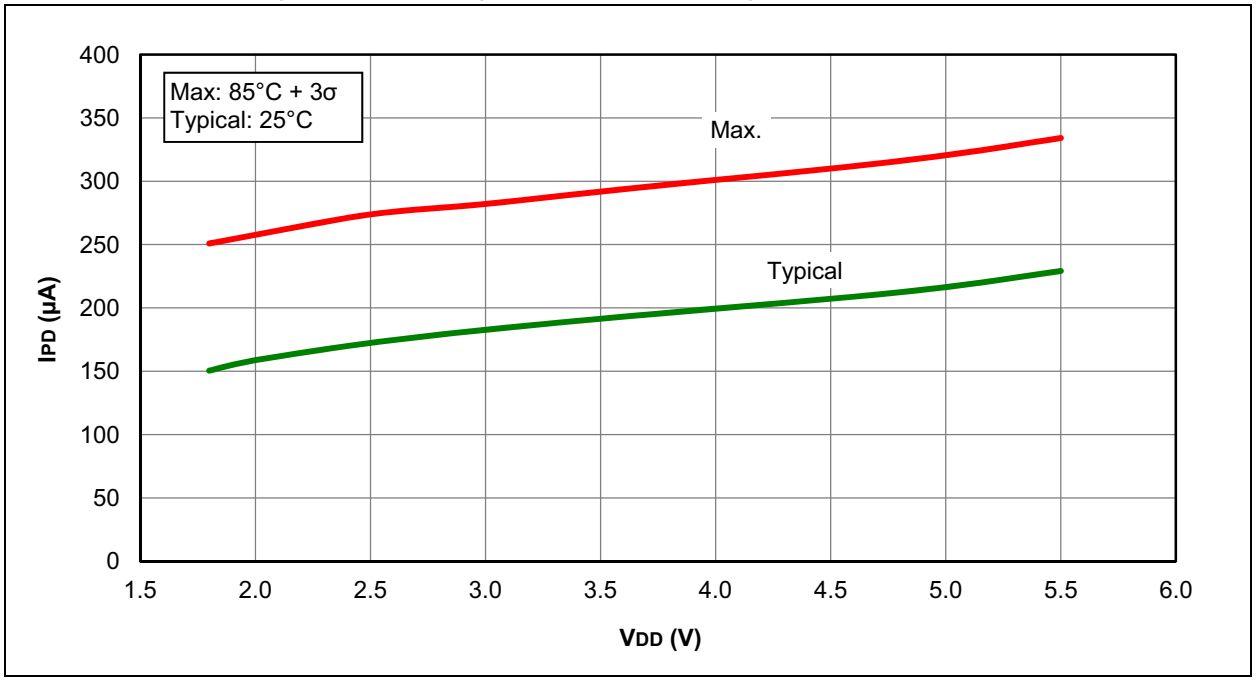


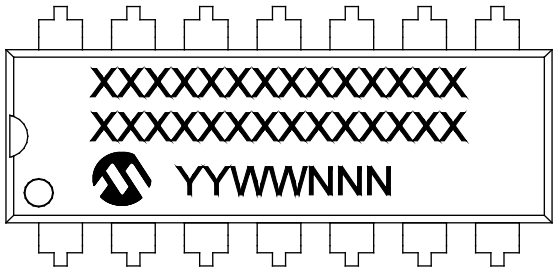

**FIGURE 23-33: IPD, COMPARATOR, HIGH-POWER MODE, PIC16F753 ONLY**



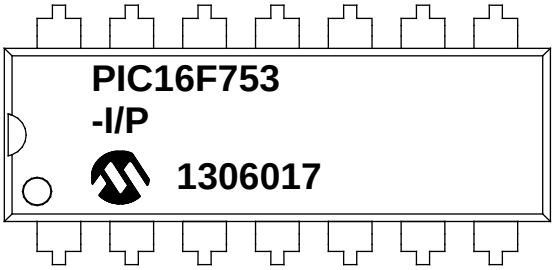
## 24.0 PACKAGING INFORMATION

### 24.1 Package Marking Information

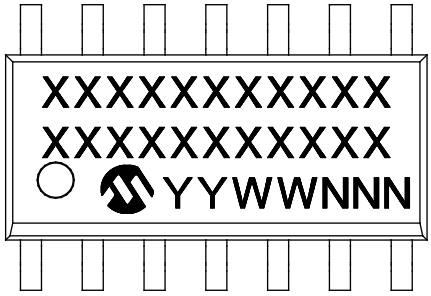
14-Lead PDIP (300 mil)



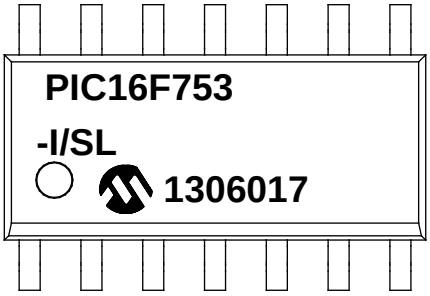

Example




14-Lead SOIC (3.90 mm)




Example

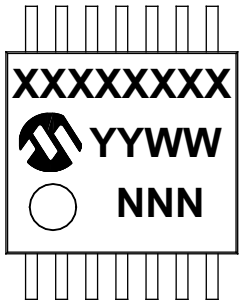



| | | |
|---|---|---|
| **Legend:** | XX...X | Customer-specific information |
| | Y | Year code (last digit of calendar year) |
| | YY | Year code (last 2 digits of calendar year) |
| | WW | Week code (week of January 1 is week ‘01’) |
| | NNN | Alphanumeric traceability code |
| | (e3) | Pb-free JEDEC® 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. |

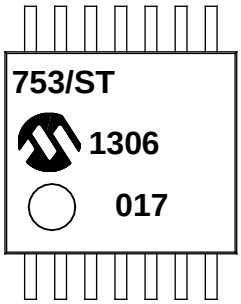
**Note**: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line, thus limiting the number of available characters for customer-specific information.

### 24.2 Package Marking Information

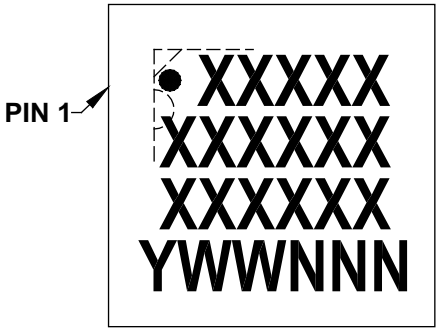
14-Lead TSSOP (4.4 mm)



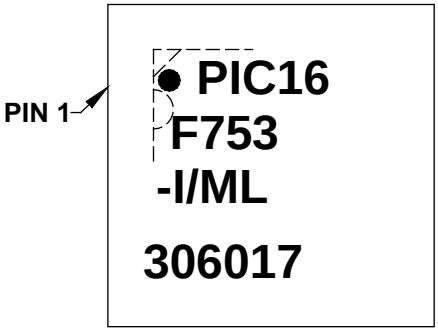

Example




16-Lead QFN (4x4x0.9 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. |

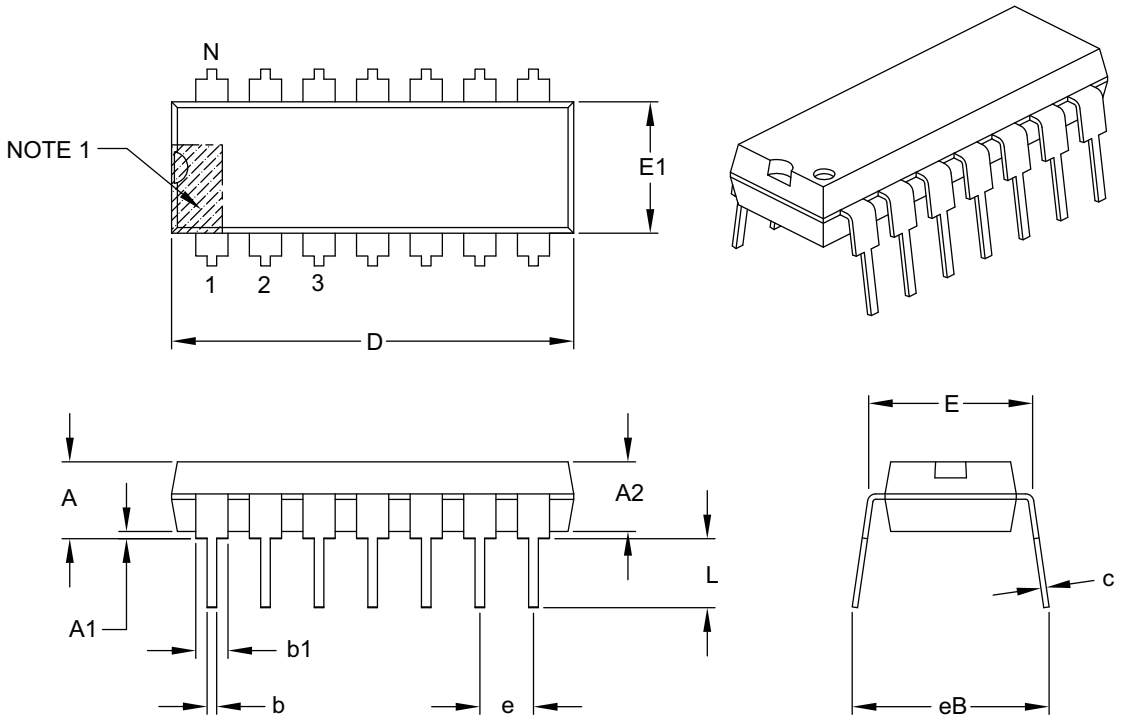
**Note**: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line, thus limiting the number of available characters for customer-specific information.

## 24.3 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:**

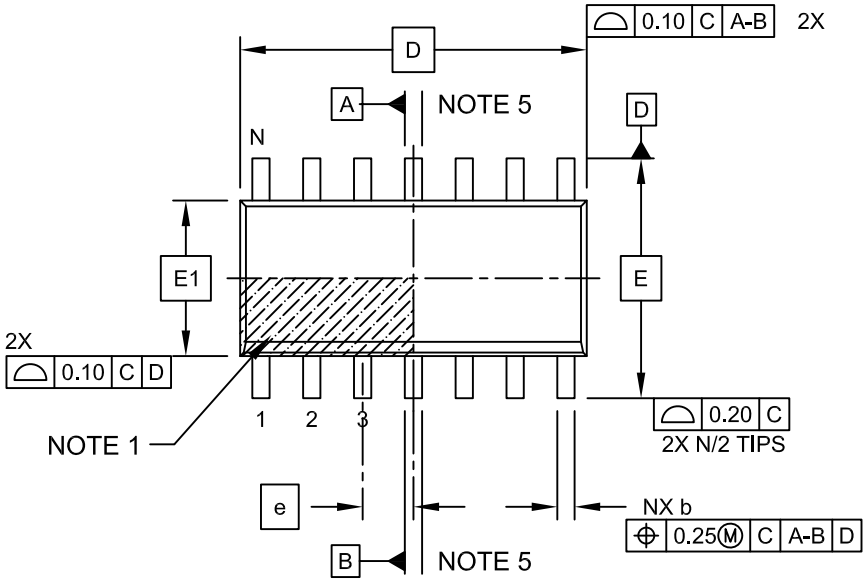
1. Pin 1 visual index feature may vary, but must be located with the hatched area.
2. § Significant Characteristic.
3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.
4. Dimensioning and tolerancing per ASME Y14.5M.

   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

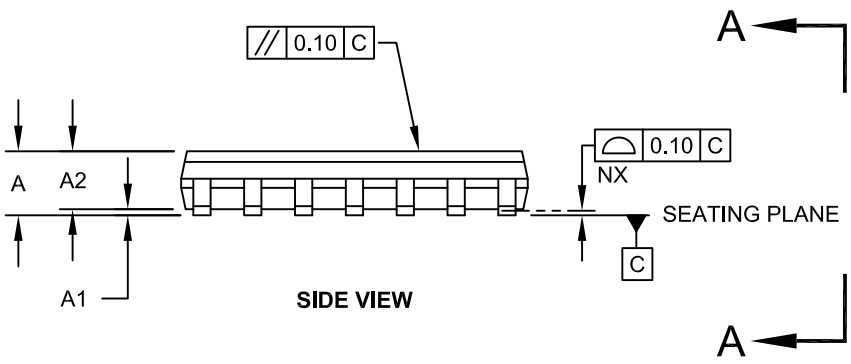
Microchip Technology Drawing C04-005B

### 14-Lead Plastic Small Outline (SL) - Narrow, 3.90 mm Body [SOIC]

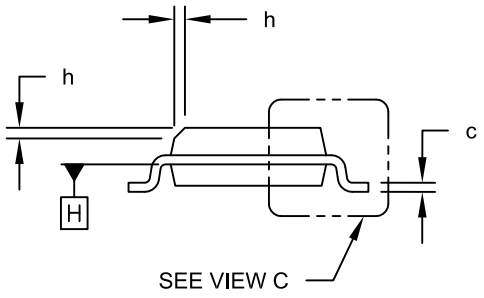
**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging




**TOP VIEW**




**SIDE VIEW**

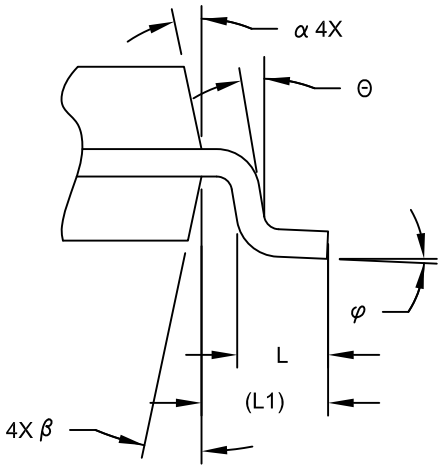


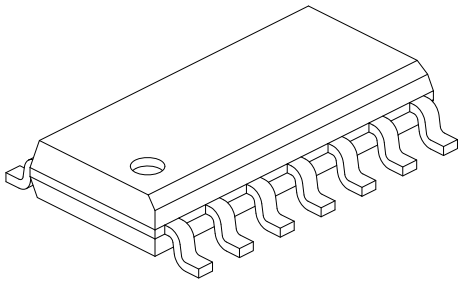

**VIEW A-A**

Microchip Technology Drawing No. C04-065C Sheet 1 of 2

### 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

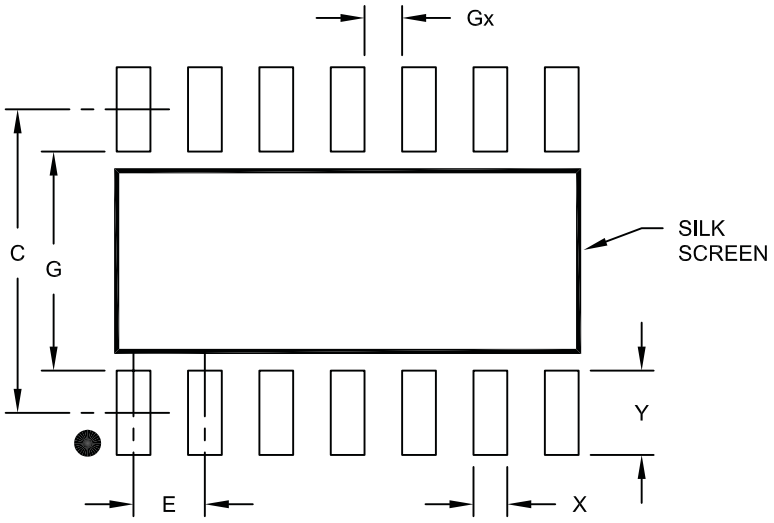
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

   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:

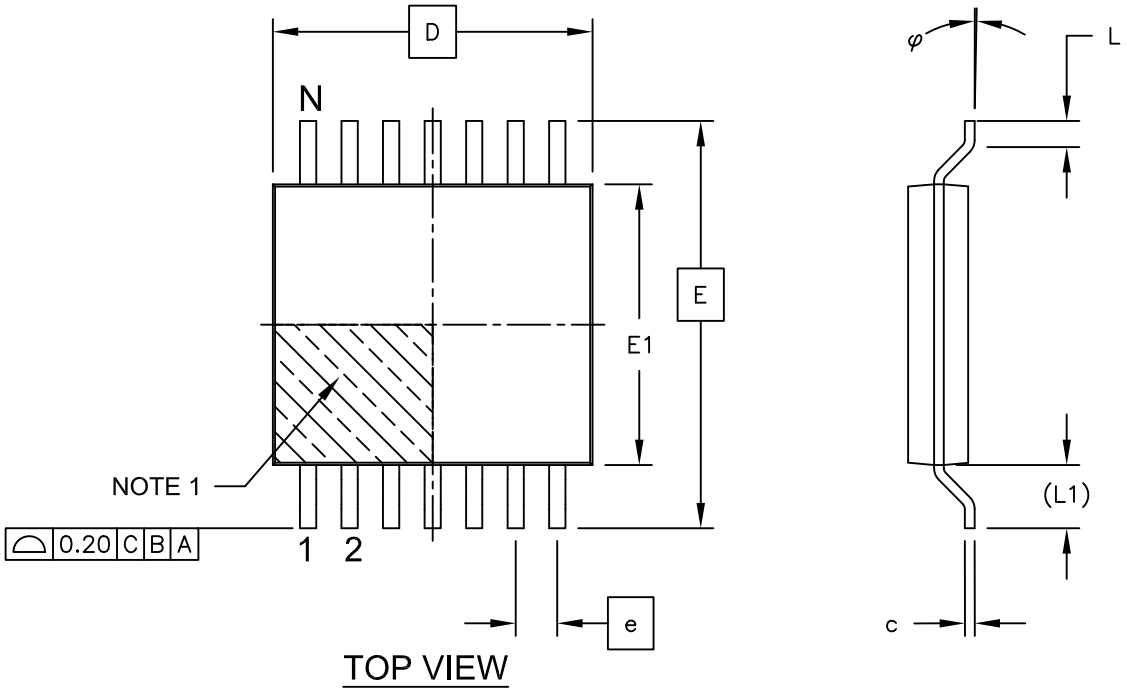
1. Dimensioning and tolerancing per ASME Y14.5M

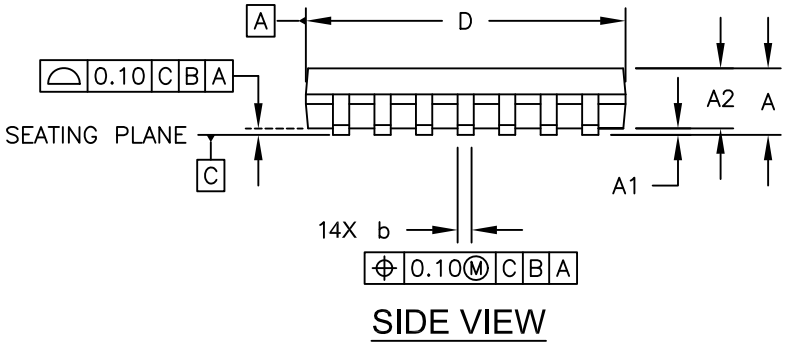
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2065A

**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

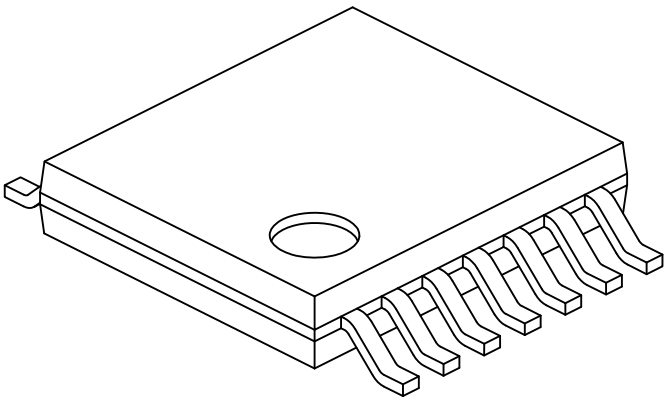



Microchip Technology Drawing C04-087C Sheet 1 of 2

### 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

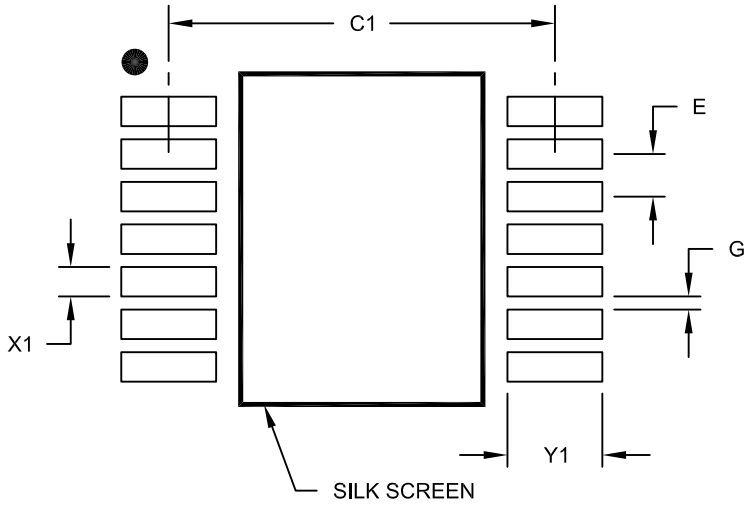
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

   REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing No. C04-087C Sheet 2 of 2

### 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:

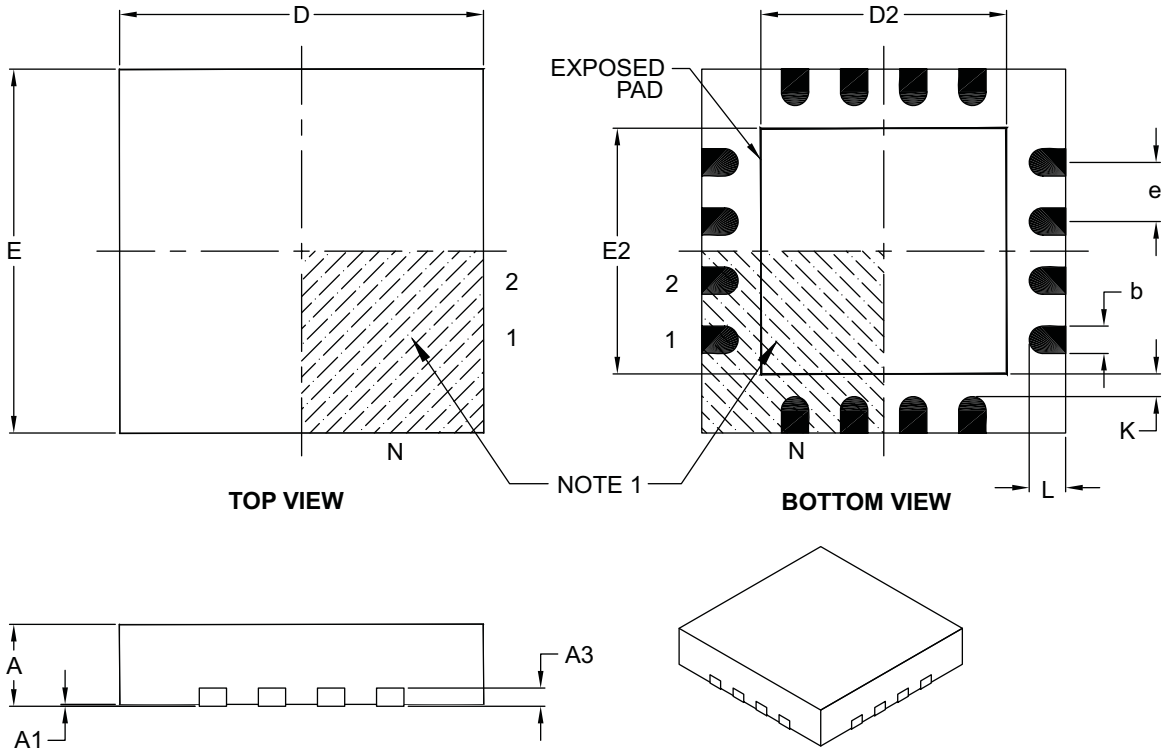
1. Dimensioning and tolerancing per ASME Y14.5M

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:**

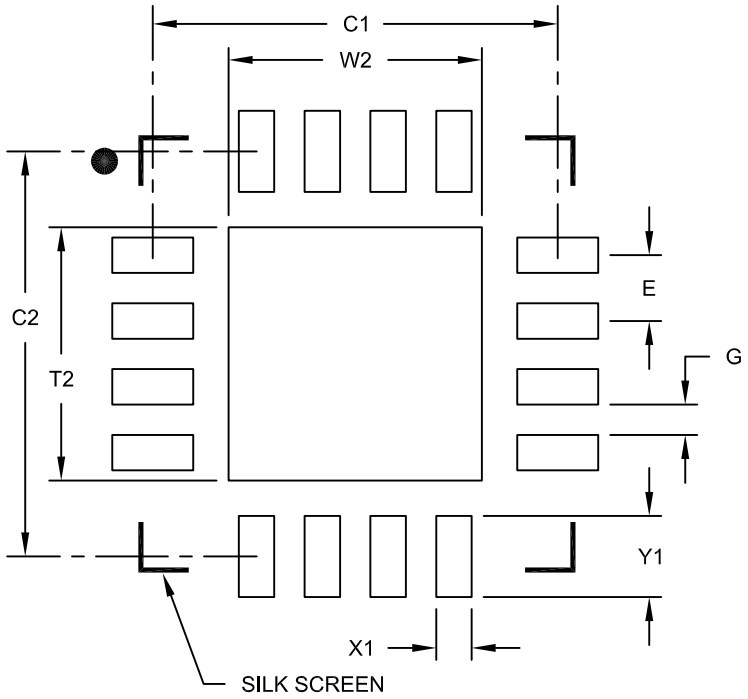
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Package is saw singulated.
3. Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-127B

### 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

## APPENDIX A: DATA SHEET REVISION HISTORY

### Revision A (05/2013)

Original release of this document.

### Revision B (11/2013)

Electrical Specification chapter updated, Characterization Data chapter updated. Miscellaneous corrections to the following chapters: Device Overview, Memory Organization, I/O Ports, COG Module, Fixed Voltage Reference (FVR), Slope Compensation (SC) Module.

### Revision C (03/2015)

Updated Figures 2-2, 13-1, 14-1, 17-1, and 17-2; Registers 5-11, 5-12, 11-11, and 11-12; Sections 5.5.4 and 22.0; Table 1-1, 22-3, 22-4, 22-15, and 22-17.

### Revision D (06/2016)

Updated the ‘eXtreme Low-Power (XLP) Features’ section; Other minor corrections.

## THE MICROCHIP WEBSITE

Microchip provides online support via our website at www.microchip.com. This website is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the website 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 website 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 website 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:** | PIC16F753<br>PIC16HV753 | |
| **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:** | P<br>SL<br>ST<br>ML | = 14-lead Plastic Dual In-line (PDIP)<br>= 14-lead Plastic Small Outline (3.90 mm) (SOIC)<br>= 14-lead Plastic Thin Shrink Small Outline (4.4 mm) (TSSOP)<br>= 16-lead Plastic Quad Flat, No Lead Package (4x4x0.9 mm) (QFN) |
| **Pattern:** | QTP, SQTP, Code or Special Requirements (blank otherwise) | |

**Examples:**

a) PIC16F753-I/ML301
Tape and Reel,
Industrial temperature,
QFN 4x43 package,
QTP pattern #301

b) PIC16F753-E/P
Extended temperature
PDIP package

c) PIC16F753-E/SL
Extended temperature,
SOIC package

d) PIC16HV753-E/ST
Extended temperature,
TSSOP 4.4 mm 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.

**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.



*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.*

**QUALITY MANAGEMENT SYSTEM**
**CERTIFIED BY DNV**
**═ ISO/TS 16949 ═**

**Trademarks**



ISBN: 978-1-5224-0737-9

# 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: 905-695-1980
Fax: 905-695-2078

## 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 - Guangzhou**
Tel: 86-20-8755-8029

**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

06/23/16