# PIC16F882/883/884/886/887

## 28/40/44-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 oscillator/clock input
  - DC – 200 ns instruction cycle
- Interrupt Capability
- 8-Level Deep Hardware Stack
- Direct, Indirect and Relative Addressing modes

### Special Microcontroller Features:

- Precision Internal Oscillator:
  - Factory calibrated to ±1%
  - Software selectable frequency range of 8 MHz to 31 kHz
  - Software tunable
  - Two-Speed Start-up mode
  - Crystal fail detect for critical applications
  - Clock mode switching during operation for power savings
- Power-Saving Sleep mode
- Wide Operating Voltage Range (2.0V-5.5V)
- Industrial and Extended Temperature Range
- Power-on Reset (POR)
- Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Brown-out Reset (BOR) with Software Control Option
- Enhanced Low-Current Watchdog Timer (WDT) with On-Chip Oscillator (software selectable nominal 268 seconds with full prescaler) with software enable
- Multiplexed Master Clear with Pull-up/Input Pin
- Programmable Code Protection
- High Endurance Flash/EEPROM Cell:
  - 100,000 write Flash endurance
  - 1,000,000 write EEPROM endurance
  - Flash/Data EEPROM retention: > 40 years
- Program Memory Read/Write during run time
- In-Circuit Debugger (on board)

### Low-Power Features:

- Standby Current:
  - 50 nA @ 2.0V, typical
- Operating Current:
  - 11 μA @ 32 kHz, 2.0V, typical
  - 220 μA @ 4 MHz, 2.0V, typical
- Watchdog Timer Current:
  - 1 μA @ 2.0V, typical

### Peripheral Features:

- 24/35 I/O Pins with Individual Direction Control:
  - High current source/sink for direct LED drive
  - Interrupt-on-Change pin
  - Individually programmable weak pull-ups
  - Ultra Low-Power Wake-up (ULPWU)
- Analog Comparator Module with:
  - Two analog comparators
  - Programmable on-chip voltage reference ($CV_{REF}$) module (% of $V_{DD}$)
  - Fixed voltage reference (0.6V)
  - Comparator inputs and outputs externally accessible
  - SR Latch mode
  - External Timer1 Gate (count enable)
- A/D Converter:
  - 10-bit resolution and 11/14 channels
- Timer0: 8-bit Timer/Counter with 8-bit Programmable Prescaler
- Enhanced Timer1:
  - 16-bit timer/counter with prescaler
  - External Gate Input mode
  - Dedicated low-power 32 kHz oscillator
- Timer2: 8-bit Timer/Counter with 8-bit Period Register, Prescaler and Postscaler
- Enhanced Capture, Compare, PWM+ Module:
  - 16-bit Capture, max. resolution 12.5 ns
  - Compare, max. resolution 200 ns
  - 10-bit PWM with 1, 2 or 4 output channels, programmable “dead time”, max. frequency 20 kHz
  - PWM output steering control
- Capture, Compare, PWM Module:
  - 16-bit Capture, max. resolution 12.5 ns
  - 16-bit Compare, max. resolution 200 ns
  - 10-bit PWM, max. frequency 20 kHz
- Enhanced USART Module:
  - Supports RS-485, RS-232, and LIN 2.0
  - Auto-Baud Detect
  - Auto-Wake-Up on Start bit
- In-Circuit Serial Programming™ (ICSP™) via Two Pins
- Master Synchronous Serial Port (MSSP) Module supporting 3-wire SPI (all 4 modes) and $I^2C$™ Master and Slave Modes with $I^2C$ Address Mask

© 2006-2012 Microchip Technology Inc.

### PIC16F882/883/884/886/887 Family Types

| Device | Program Memory | Data Memory | | I/O | 10-bit A/D (ch) | ECCP/ CCP | EUSART | MSSP | Comparators | Timers 8/16-bit |
|---|---|---|---|---|---|---|---|---|---|---|
| | Flash (words) | SRAM (bytes) | EEPROM (bytes) | | | | | | | |
| PIC16F882 | 2048 | 128 | 128 | 24 | 11 | 1/1 | 1 | 1 | 2 | 2/1 |
| PIC16F883 | 4096 | 256 | 256 | 24 | 11 | 1/1 | 1 | 1 | 2 | 2/1 |
| PIC16F884 | 4096 | 256 | 256 | 35 | 14 | 1/1 | 1 | 1 | 2 | 2/1 |
| PIC16F886 | 8192 | 368 | 256 | 24 | 11 | 1/1 | 1 | 1 | 2 | 2/1 |
| PIC16F887 | 8192 | 368 | 256 | 35 | 14 | 1/1 | 1 | 1 | 2 | 2/1 |


© 2006-2012 Microchip Technology Inc.

## Pin Diagrams – PIC16F882/883/886, 28-Pin PDIP, SOIC, SSOP

**28-Pin PDIP, SOIC, SSOP**

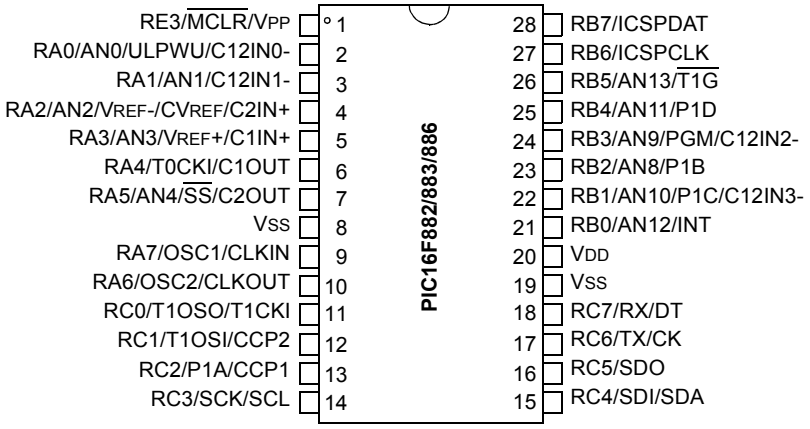


**TABLE 1: 28-PIN PDIP, SOIC, SSOP ALLOCATION TABLE (PIC16F882/883/886)**

| I/O | 28-Pin PDIP/SOIC/SSOP | Analog | Comparators | Timers | ECCP | EUSART | MSSP | Interrupt | Pull-up | Basic |
|---|---|---|---|---|---|---|---|---|---|---|
| RA0 | 2 | AN0/ULPWU | C12IN0- | — | — | — | — | — | — | — |
| RA1 | 3 | AN1 | C12IN1- | — | — | — | — | — | — | — |
| RA2 | 4 | AN2 | C2IN+ | — | — | — | — | — | — | VREF-/CVREF |
| RA3 | 5 | AN3 | C1IN+ | — | — | — | — | — | — | VREF+ |
| RA4 | 6 | — | C1OUT | T0CKI | — | — | — | — | — | — |
| RA5 | 7 | AN4 | C2OUT | — | — | — | $\overline{SS}$ | — | — | — |
| RA6 | 10 | — | — | — | — | — | — | — | — | OSC2/CLKOUT |
| RA7 | 9 | — | — | — | — | — | — | — | — | OSC1/CLKIN |
| RB0 | 21 | AN12 | — | — | — | — | — | IOC/INT | Y | — |
| RB1 | 22 | AN10 | C12IN3- | — | P1C | — | — | IOC | Y | — |
| RB2 | 23 | AN8 | — | — | P1B | — | — | IOC | Y | — |
| RB3 | 24 | AN9 | C12IN2- | — | — | — | — | IOC | Y | PGM |
| RB4 | 25 | AN11 | — | — | P1D | — | — | IOC | Y | — |
| RB5 | 26 | AN13 | — | $\overline{T1G}$ | — | — | — | IOC | Y | — |
| RB6 | 27 | — | — | — | — | — | — | IOC | Y | ICSPCLK |
| RB7 | 28 | — | — | — | — | — | — | IOC | Y | ICSPDAT |
| RC0 | 11 | — | — | T1OSO/T1CKI | — | — | — | — | — | — |
| RC1 | 12 | — | — | T1OSI | CCP2 | — | — | — | — | — |
| RC2 | 13 | — | — | — | CCP1/P1A | — | — | — | — | — |
| RC3 | 14 | — | — | — | — | — | SCK/SCL | — | — | — |
| RC4 | 15 | — | — | — | — | — | SDI/SDA | — | — | — |
| RC5 | 16 | — | — | — | — | — | SDO | — | — | — |
| RC6 | 17 | — | — | — | — | TX/CK | — | — | — | — |
| RC7 | 18 | — | — | — | — | RX/DT | — | — | — | — |
| RE3 | 1 | — | — | — | — | — | — | — | Y(1) | $\overline{MCLR}$/VPP |
| — | 20 | — | — | — | — | — | — | — | — | VDD |
| — | 8 | — | — | — | — | — | — | — | — | VSS |
| — | 19 | — | — | — | — | — | — | — | — | VSS |

**Note 1:** Pull-up activated only with external $\overline{MCLR}$ configuration.


© 2006-2012 Microchip Technology Inc.

## Pin Diagrams – PIC16F882/883/886, 28-Pin QFN

**28-Pin QFN**

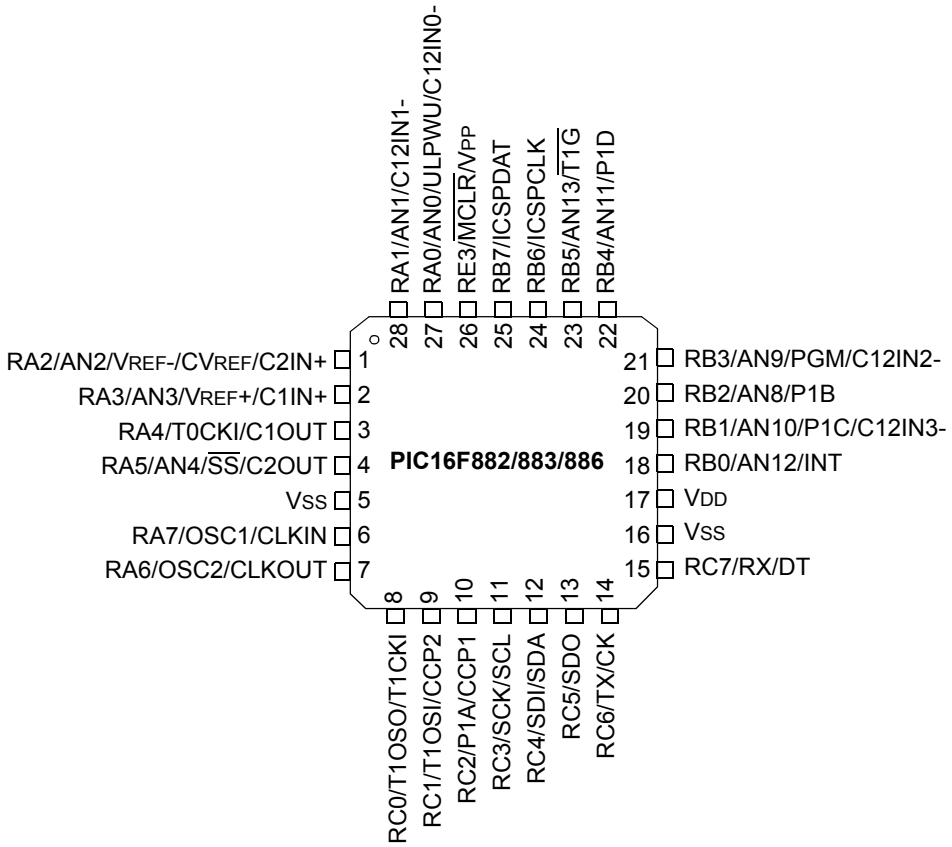


**TABLE 2: 28-PIN QFN ALLOCATION TABLE (PIC16F882/883/886)**

| I/O | 28-Pin QFN | Analog | Comparators | Timers | ECCP | EUSART | MSSP | Interrupt | Pull-up | Basic |
|---|---|---|---|---|---|---|---|---|---|---|
| RA0 | 27 | AN0/ULPWU | C12IN0- | — | — | — | — | — | — | — |
| RA1 | 28 | AN1 | C12IN1- | — | — | — | — | — | — | — |
| RA2 | 1 | AN2 | C2IN+ | — | — | — | — | — | — | VREF-/CVREF |
| RA3 | 2 | AN3 | C1IN+ | — | — | — | — | — | — | VREF+ |
| RA4 | 3 | — | C1OUT | T0CKI | — | — | — | — | — | — |
| RA5 | 4 | AN4 | C2OUT | — | — | — | $\overline{SS}$ | — | — | — |
| RA6 | 7 | — | — | — | — | — | — | — | — | OSC2/CLKOUT |
| RA7 | 6 | — | — | — | — | — | — | — | — | OSC1/CLKIN |
| RB0 | 18 | AN12 | — | — | — | — | — | IOC/INT | Y | — |
| RB1 | 19 | AN10 | C12IN3- | — | P1C | — | — | IOC | Y | — |
| RB2 | 20 | AN8 | — | — | P1B | — | — | IOC | Y | — |
| RB3 | 21 | AN9 | C12IN2- | — | — | — | — | IOC | Y | PGM |
| RB4 | 22 | AN11 | — | — | P1D | — | — | IOC | Y | — |
| RB5 | 23 | AN13 | — | $\overline{T1G}$ | — | — | — | IOC | Y | — |
| RB6 | 24 | — | — | — | — | — | — | IOC | Y | ICSPCLK |
| RB7 | 25 | — | — | — | — | — | — | IOC | Y | ICSPDAT |
| RC0 | 8 | — | — | T1OSO/T1CKI | — | — | — | — | — | — |
| RC1 | 9 | — | — | T1OSI | CCP2 | — | — | — | — | — |
| RC2 | 10 | — | — | — | CCP1/P1A | — | — | — | — | — |
| RC3 | 11 | — | — | — | — | — | SCK/SCL | — | — | — |
| RC4 | 12 | — | — | — | — | — | SDI/SDA | — | — | — |
| RC5 | 13 | — | — | — | — | — | SDO | — | — | — |
| RC6 | 14 | — | — | — | — | TX/CK | — | — | — | — |
| RC7 | 15 | — | — | — | — | RX/DT | — | — | — | — |
| RE3 | 26 | — | — | — | — | — | — | — | Y(1) | $\overline{MCLR}$/VPP |
| — | 17 | — | — | — | — | — | — | — | — | VDD |
| — | 5 | — | — | — | — | — | — | — | — | VSS |
| — | 16 | — | — | — | — | — | — | — | — | VSS |

**Note 1:** Pull-up activated only with external $\overline{MCLR}$ configuration.


© 2006-2012 Microchip Technology Inc.

## Pin Diagrams – PIC16F884/887, 40-Pin PDIP

**40-Pin PDIP**

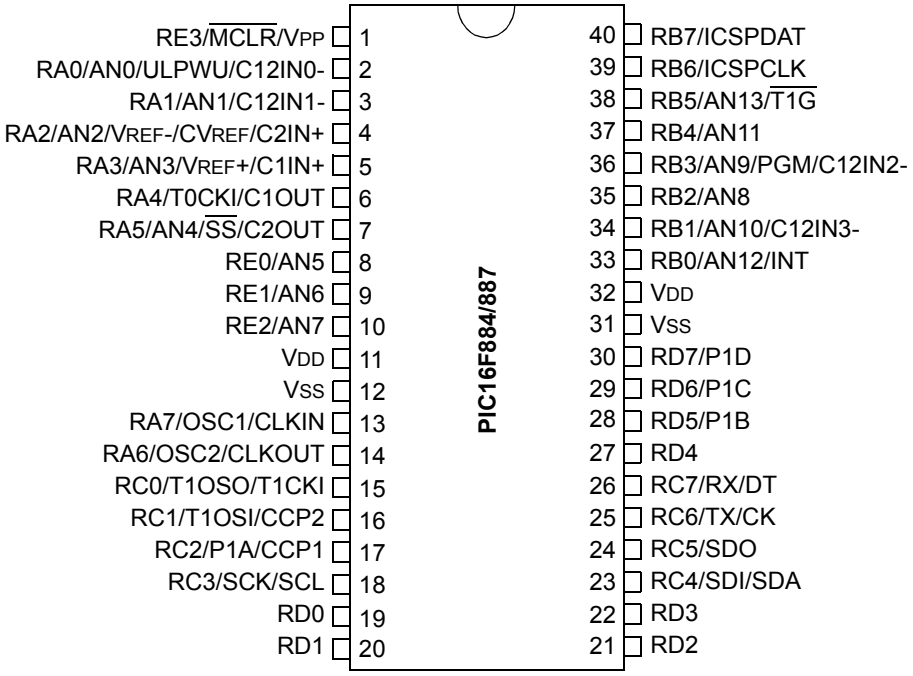


**TABLE 3: 40-PIN PDIP ALLOCATION TABLE (PIC16F884/887)**

| I/O | 40-Pin PDIP | Analog | Comparators | Timers | ECCP | EUSART | MSSP | Interrupt | Pull-up | Basic |
|---|---|---|---|---|---|---|---|---|---|---|
| RA0 | 2 | AN0/ULPWU | C12IN0- | — | — | — | — | — | — | — |
| RA1 | 3 | AN1 | C12IN1- | — | — | — | — | — | — | — |
| RA2 | 4 | AN2 | C2IN+ | — | — | — | — | — | — | VREF-/CVREF |
| RA3 | 5 | AN3 | C1IN+ | — | — | — | — | — | — | VREF+ |
| RA4 | 6 | — | C1OUT | T0CKI | — | — | — | — | — | — |
| RA5 | 7 | AN4 | C2OUT | — | — | — | $\overline{SS}$ | — | — | — |
| RA6 | 14 | — | — | — | — | — | — | — | — | OSC2/CLKOUT |
| RA7 | 13 | — | — | — | — | — | — | — | — | OSC1/CLKIN |
| RB0 | 33 | AN12 | — | — | — | — | — | IOC/INT | Y | — |
| RB1 | 34 | AN10 | C12IN3- | — | — | — | — | IOC | Y | — |
| RB2 | 35 | AN8 | — | — | — | — | — | IOC | Y | — |
| RB3 | 36 | AN9 | C12IN2- | — | — | — | — | IOC | Y | PGM |
| RB4 | 37 | AN11 | — | — | — | — | — | IOC | Y | — |
| RB5 | 38 | AN13 | — | $\overline{T1G}$ | — | — | — | IOC | Y | — |
| RB6 | 39 | — | — | — | — | — | — | IOC | Y | ICSPCLK |
| RB7 | 40 | — | — | — | — | — | — | IOC | Y | ICSPDAT |
| RC0 | 15 | — | — | T1OSO/T1CKI | — | — | — | — | — | — |
| RC1 | 16 | — | — | T1OSI | CCP2 | — | — | — | — | — |
| RC2 | 17 | — | — | — | CCP1/P1A | — | — | — | — | — |
| RC3 | 18 | — | — | — | — | — | SCK/SCL | — | — | — |
| RC4 | 23 | — | — | — | — | — | SDI/SDA | — | — | — |
| RC5 | 24 | — | — | — | — | — | SDO | — | — | — |
| RC6 | 25 | — | — | — | — | TX/CK | — | — | — | — |
| RC7 | 26 | — | — | — | — | RX/DT | — | — | — | — |
| RD0 | 19 | — | — | — | — | — | — | — | — | — |
| RD1 | 20 | — | — | — | — | — | — | — | — | — |
| RD2 | 21 | — | — | — | — | — | — | — | — | — |
| RD3 | 22 | — | — | — | — | — | — | — | — | — |
| RD4 | 27 | — | — | — | — | — | — | — | — | — |
| RD5 | 28 | — | — | — | P1B | — | — | — | — | — |
| RD6 | 29 | — | — | — | P1C | — | — | — | — | — |
| RD7 | 30 | — | — | — | P1D | — | — | — | — | — |
| RE0 | 8 | AN5 | — | — | — | — | — | — | — | — |
| RE1 | 9 | AN6 | — | — | — | — | — | — | — | — |
| RE2 | 10 | AN7 | — | — | — | — | — | — | — | — |
| RE3 | 1 | — | — | — | — | — | — | — | Y(1) | $\overline{MCLR}$/VPP |
| — | 11 | — | — | — | — | — | — | — | — | VDD |
| — | 32 | — | — | — | — | — | — | — | — | VDD |
| — | 12 | — | — | — | — | — | — | — | — | VSS |
| — | 31 | — | — | — | — | — | — | — | — | VSS |

**Note 1:** Pull-up activated only with external $\overline{MCLR}$ configuration.


© 2006-2012 Microchip Technology Inc.

## Pin Diagrams – PIC16F884/887, 44-Pin QFN

**44-Pin QFN**

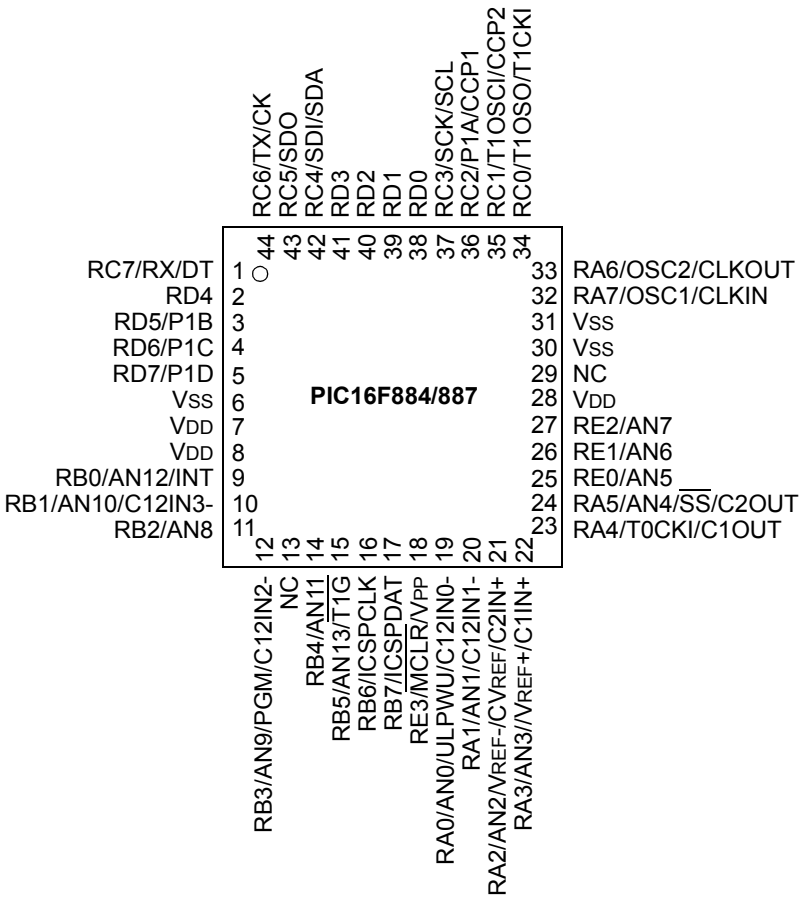


**TABLE 4: 44-PIN QFN ALLOCATION TABLE (PIC16F884/887)**

| I/O | 44-Pin QFN | Analog | Comparators | Timers | ECCP | EUSART | MSSP | Interrupt | Pull-up | Basic |
|---|---|---|---|---|---|---|---|---|---|---|
| RA0 | 19 | AN0/ULPWU | C12IN0- | — | — | — | — | — | — | — |
| RA1 | 20 | AN1 | C12IN1- | — | — | — | — | — | — | — |
| RA2 | 21 | AN2 | C2IN+ | — | — | — | — | — | — | VREF-/CVREF |
| RA3 | 22 | AN3 | C1IN+ | — | — | — | — | — | — | VREF+ |
| RA4 | 23 | — | C1OUT | T0CKI | — | — | — | — | — | — |
| RA5 | 24 | AN4 | C2OUT | — | — | — | $\overline{SS}$ | — | — | — |
| RA6 | 33 | — | — | — | — | — | — | — | — | OSC2/CLKOUT |
| RA7 | 32 | — | — | — | — | — | — | — | — | OSC1/CLKIN |
| RB0 | 9 | AN12 | — | — | — | — | — | IOC/INT | Y | — |
| RB1 | 10 | AN10 | C12IN3- | — | — | — | — | IOC | Y | — |
| RB2 | 11 | AN8 | — | — | — | — | — | IOC | Y | — |
| RB3 | 12 | AN9 | C12IN2- | — | — | — | — | IOC | Y | PGM |
| RB4 | 14 | AN11 | — | — | — | — | — | IOC | Y | — |
| RB5 | 15 | AN13 | — | $\overline{T1G}$ | — | — | — | IOC | Y | — |
| RB6 | 16 | — | — | — | — | — | — | IOC | Y | ICSPCLK |
| RB7 | 17 | — | — | — | — | — | — | IOC | Y | ICSPDAT |
| RC0 | 34 | — | — | T1OSO/T1CKI | — | — | — | — | — | — |
| RC1 | 35 | — | — | T1OSI | CCP2 | — | — | — | — | — |
| RC2 | 36 | — | — | — | CCP1/P1A | — | — | — | — | — |
| RC3 | 37 | — | — | — | — | — | SCK/SCL | — | — | — |
| RC4 | 42 | — | — | — | — | — | SDI/SDA | — | — | — |
| RC5 | 43 | — | — | — | — | — | SDO | — | — | — |
| RC6 | 44 | — | — | — | — | TX/CK | — | — | — | — |
| RC7 | 1 | — | — | — | — | RX/DT | — | — | — | — |
| RD0 | 38 | — | — | — | — | — | — | — | — | — |
| RD1 | 39 | — | — | — | — | — | — | — | — | — |
| RD2 | 40 | — | — | — | — | — | — | — | — | — |
| RD3 | 41 | — | — | — | — | — | — | — | — | — |
| RD4 | 2 | — | — | — | — | — | — | — | — | — |
| RD5 | 3 | — | — | — | P1B | — | — | — | — | — |
| RD6 | 4 | — | — | — | P1C | — | — | — | — | — |
| RD7 | 5 | — | — | — | P1D | — | — | — | — | — |
| RE0 | 25 | AN5 | — | — | — | — | — | — | — | — |
| RE1 | 26 | AN6 | — | — | — | — | — | — | — | — |
| RE2 | 27 | AN7 | — | — | — | — | — | — | — | — |
| RE3 | 18 | — | — | — | — | — | — | — | Y(1) | $\overline{MCLR}$/VPP |
| — | 7 | — | — | — | — | — | — | — | — | VDD |
| — | 8 | — | — | — | — | — | — | — | — | VDD |
| — | 28 | — | — | — | — | — | — | — | — | VDD |
| — | 6 | — | — | — | — | — | — | — | — | VSS |
| — | 30 | — | — | — | — | — | — | — | — | VSS |
| — | 31 | — | — | — | — | — | — | — | — | VSS |
| — | 13 | — | — | — | — | — | — | — | — | NC (no connect) |
| — | 29 | — | — | — | — | — | — | — | — | NC (no connect) |

**Note 1:** Pull-up activated only with external $\overline{MCLR}$ configuration.


© 2006-2012 Microchip Technology Inc.

## Pin Diagrams – PIC16F884/887, 44-Pin TQFP

**44-Pin TQFP**

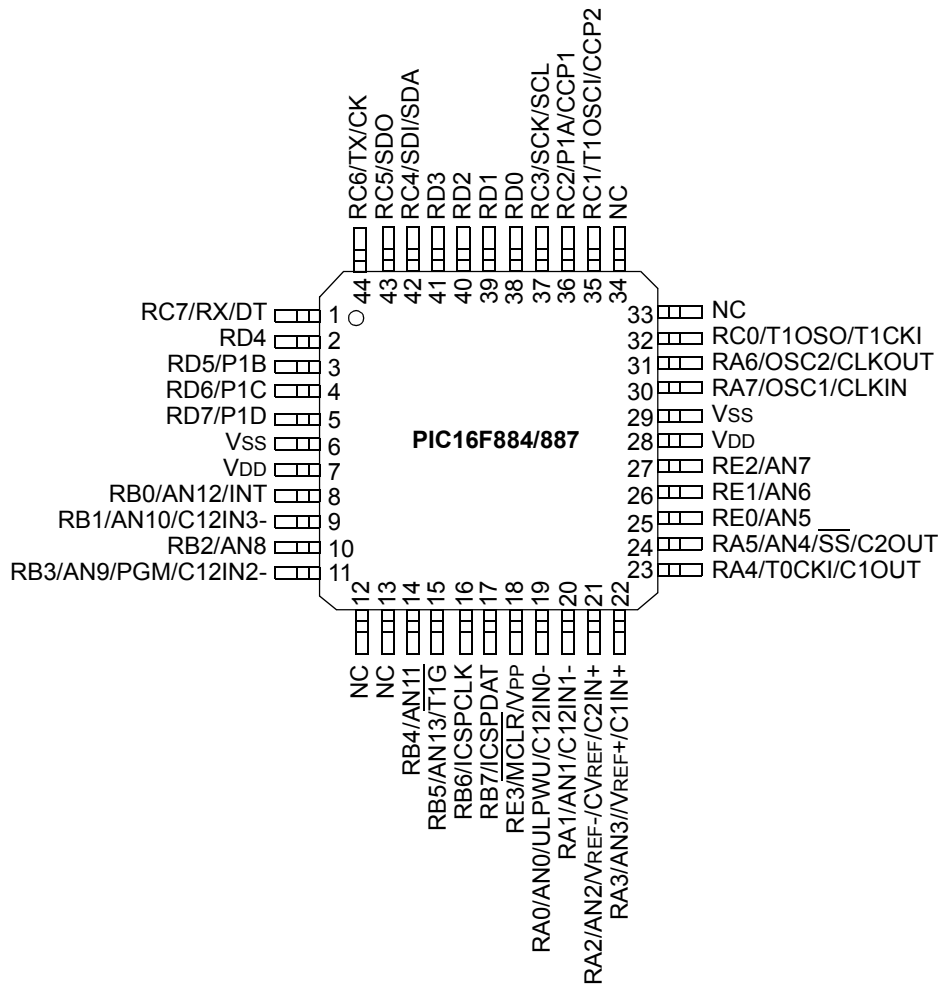


**TABLE 5: 44-PIN TQFP ALLOCATION TABLE (PIC16F884/887)**

| I/O | 44-Pin TQFP | Analog | Comparators | Timers | ECCP | EUSART | MSSP | Interrupt | Pull-up | Basic |
|---|---|---|---|---|---|---|---|---|---|---|
| RA0 | 19 | AN0/ULPWU | C12IN0- | — | — | — | — | — | — | — |
| RA1 | 20 | AN1 | C12IN1- | — | — | — | — | — | — | — |
| RA2 | 21 | AN2 | C2IN+ | — | — | — | — | — | — | VREF-/CVREF |
| RA3 | 22 | AN3 | C1IN+ | — | — | — | — | — | — | VREF+ |
| RA4 | 23 | — | C1OUT | T0CKI | — | — | — | — | — | — |
| RA5 | 24 | AN4 | C2OUT | — | — | — | $\overline{SS}$ | — | — | — |
| RA6 | 31 | — | — | — | — | — | — | — | — | OSC2/CLKOUT |
| RA7 | 30 | — | — | — | — | — | — | — | — | OSC1/CLKIN |
| RB0 | 8 | AN12 | — | — | — | — | — | IOC/INT | Y | — |
| RB1 | 9 | AN10 | C12IN3- | — | — | — | — | IOC | Y | — |
| RB2 | 10 | AN8 | — | — | — | — | — | IOC | Y | — |
| RB3 | 11 | AN9 | C12IN2- | — | — | — | — | IOC | Y | PGM |
| RB4 | 14 | AN11 | — | — | — | — | — | IOC | Y | — |
| RB5 | 15 | AN13 | — | $\overline{T1G}$ | — | — | — | IOC | Y | — |
| RB6 | 16 | — | — | — | — | — | — | IOC | Y | ICSPCLK |
| RB7 | 17 | — | — | — | — | — | — | IOC | Y | ICSPDAT |
| RC0 | 32 | — | — | T1OSO/T1CKI | — | — | — | — | — | — |
| RC1 | 35 | — | — | T1OSI | CCP2 | — | — | — | — | — |
| RC2 | 36 | — | — | — | CCP1/P1A | — | — | — | — | — |
| RC3 | 37 | — | — | — | — | — | SCK/SCL | — | — | — |
| RC4 | 42 | — | — | — | — | — | SDI/SDA | — | — | — |
| RC5 | 43 | — | — | — | — | — | SDO | — | — | — |
| RC6 | 44 | — | — | — | — | TX/CK | — | — | — | — |
| RC7 | 1 | — | — | — | — | RX/DT | — | — | — | — |
| RD0 | 38 | — | — | — | — | — | — | — | — | — |
| RD1 | 39 | — | — | — | — | — | — | — | — | — |
| RD2 | 40 | — | — | — | — | — | — | — | — | — |
| RD3 | 41 | — | — | — | — | — | — | — | — | — |
| RD4 | 2 | — | — | — | — | — | — | — | — | — |
| RD5 | 3 | — | — | — | P1B | — | — | — | — | — |
| RD6 | 4 | — | — | — | P1C | — | — | — | — | — |
| RD7 | 5 | — | — | — | P1D | — | — | — | — | — |
| RE0 | 25 | AN5 | — | — | — | — | — | — | — | — |
| RE1 | 26 | AN6 | — | — | — | — | — | — | — | — |
| RE2 | 27 | AN7 | — | — | — | — | — | — | — | — |
| RE3 | 18 | — | — | — | — | — | — | — | Y(1) | $\overline{MCLR}$/VPP |
| — | 7 | — | — | — | — | — | — | — | — | VDD |
| — | 28 | — | — | — | — | — | — | — | — | VDD |
| — | 6 | — | — | — | — | — | — | — | — | VSS |
| — | 13 | — | — | — | — | — | — | — | — | NC (no connect) |
| — | 29 | — | — | — | — | — | — | — | — | VSS |
| — | 34 | — | — | — | — | — | — | — | — | NC (no connect) |
| — | 33 | — | — | — | — | — | — | — | — | NC (no connect) |
| — | 12 | — | — | — | — | — | — | — | — | NC (no connect) |

**Note 1:** Pull-up activated only with external $\overline{MCLR}$ configuration.


© 2006-2012 Microchip Technology Inc.

## Table of Contents

1.0 Device Overview ..... 15
2.0 Memory Organization ..... 23
3.0 I/O Ports ..... 41
4.0 Oscillator Module (With Fail-Safe Clock Monitor) ..... 65
5.0 Timer0 Module ..... 77
6.0 Timer1 Module with Gate Control ..... 81
7.0 Timer2 Module ..... 87
8.0 Comparator Module ..... 89
9.0 Analog-to-Digital Converter (ADC) Module ..... 103
10.0 Data EEPROM and Flash Program Memory Control ..... 115
11.0 Enhanced Capture/Compare/PWM Module ..... 127
12.0 Enhanced Universal Synchronous Asynchronous Receiver Transmitter (EUSART) ..... 155
13.0 Master Synchronous Serial Port (MSSP) Module ..... 183
14.0 Special Features of the CPU ..... 213
15.0 Instruction Set Summary ..... 235
16.0 Development Support ..... 245
17.0 Electrical Specifications ..... 249
18.0 DC and AC Characteristics Graphs and Tables ..... 277
19.0 Packaging Information ..... 305
Appendix A: Data Sheet Revision History ..... 323
Appendix B: Migrating from other PIC® Devices ..... 324
Index ..... 325
The Microchip Web Site ..... 333
Customer Change Notification Service ..... 333
Customer Support ..... 333
Reader Response ..... 334
Product Identification System ..... 335

## TO OUR VALUED CUSTOMERS

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

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

### Most Current Data Sheet

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

**http://www.microchip.com**

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

### Errata

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

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

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

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

### Customer Notification System

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

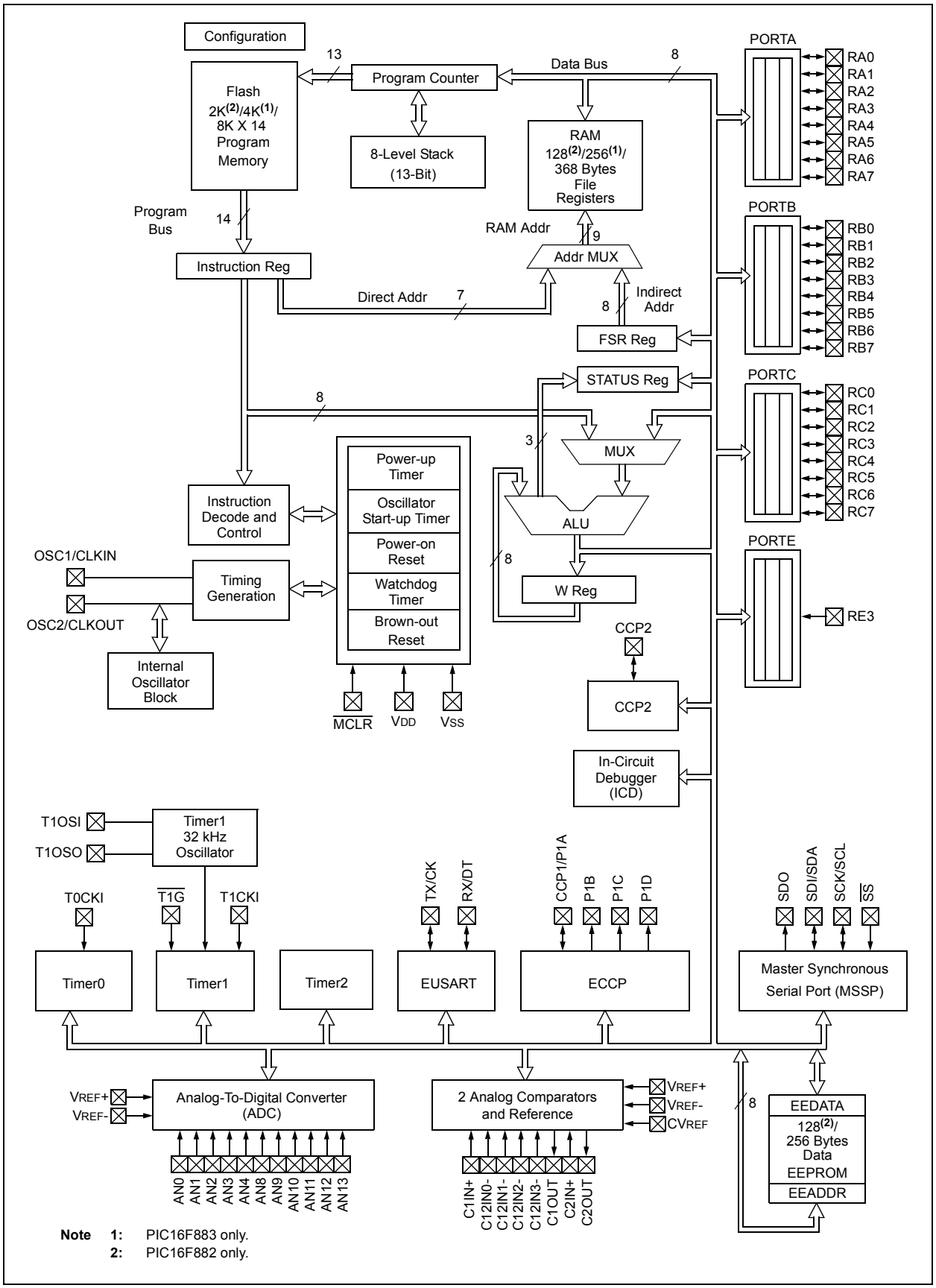
**NOTES:**


© 2006-2012 Microchip Technology Inc.

## 1.0 DEVICE OVERVIEW

The PIC16F882/883/884/886/887 devices are covered by this data sheet. The PIC16F882/883/886 devices are available in 28-pin PDIP, SOIC, SSOP and QFN packages. The PIC16F884/887 are available in a 40-pin PDIP and 44-pin QFN and TQFP packages. Figure 1-1 shows the block diagram of the PIC16F882/883/886 devices and Figure 1-2 shows a block diagram of the PIC16F884/887 devices. Table 1-1 and Table 1-2 show the corresponding pinout descriptions.

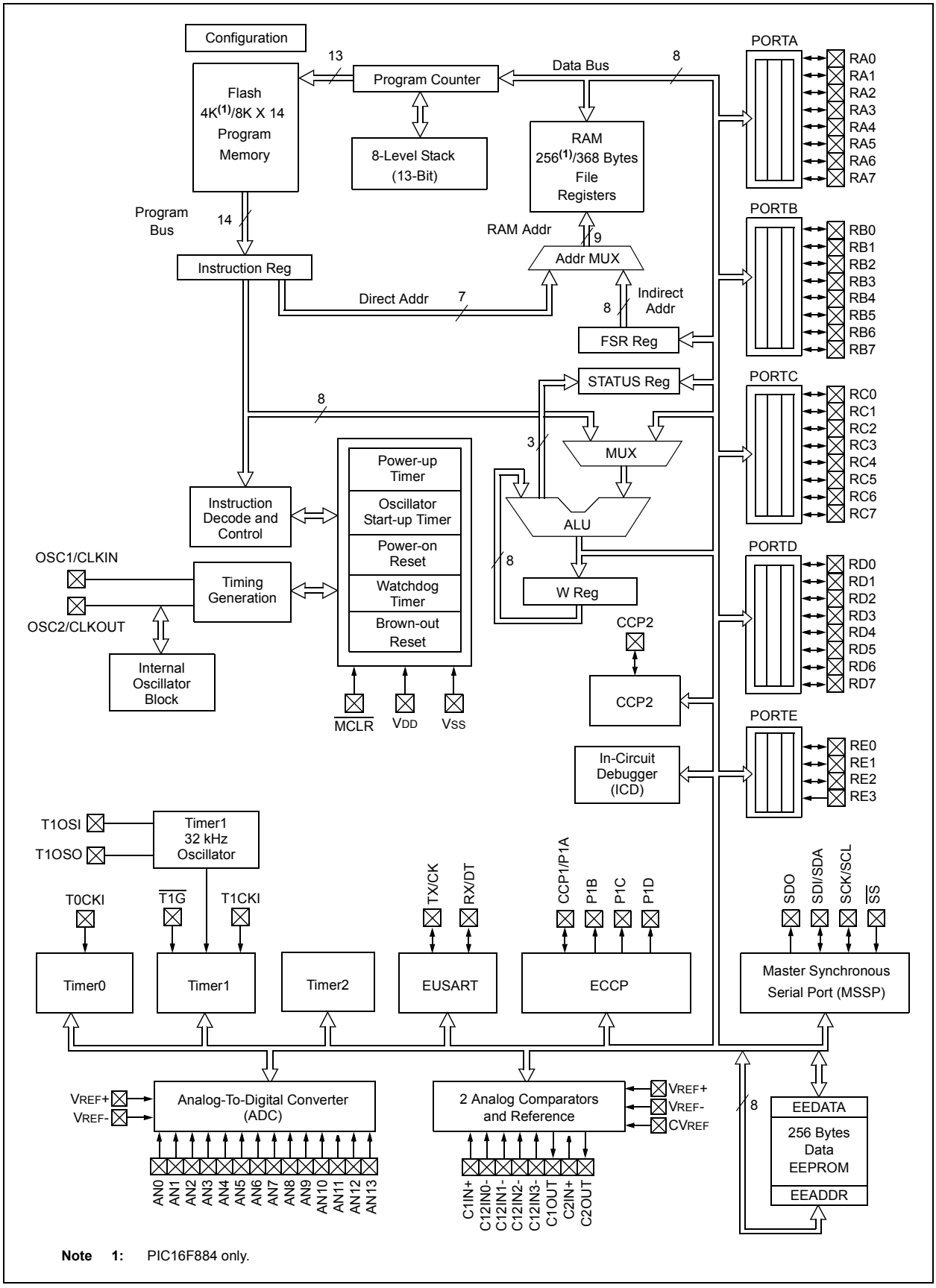
**FIGURE 1-1: PIC16F882/883/886 BLOCK DIAGRAM**




**Note 1:** PIC16F883 only.
**2:** PIC16F882 only.


© 2006-2012 Microchip Technology Inc.

**FIGURE 1-2: PIC16F884/PIC16F887 BLOCK DIAGRAM**




**Note 1:** PIC16F884 only.

**TABLE 1-1: PIC16F882/883/886 PINOUT DESCRIPTION**

| Name | Function | Input Type | Output Type | Description |
|---|---|---|---|---|
| RA0/AN0/ULPWU/C12IN0- | RA0 | TTL | CMOS | General purpose I/O. |
| | AN0 | AN | — | A/D Channel 0 input. |
| | ULPWU | AN | — | Ultra Low-Power Wake-up input. |
| | C12IN0- | AN | — | Comparator C1 or C2 negative input. |
| RA1/AN1/C12IN1- | RA1 | TTL | CMOS | General purpose I/O. |
| | AN1 | AN | — | A/D Channel 1 input. |
| | C12IN1- | AN | — | Comparator C1 or C2 negative input. |
| RA2/AN2/VREF-/CVREF/C2IN+ | RA2 | TTL | CMOS | General purpose I/O. |
| | AN2 | AN | — | A/D Channel 2. |
| | VREF- | AN | — | A/D Negative Voltage Reference input. |
| | CVREF | — | AN | Comparator Voltage Reference output. |
| | C2IN+ | AN | — | Comparator C2 positive input. |
| RA3/AN3/VREF+/C1IN+ | RA3 | TTL | — | General purpose I/O. |
| | AN3 | AN | — | A/D Channel 3. |
| | VREF+ | AN | — | Programming voltage. |
| | C1IN+ | AN | — | Comparator C1 positive input. |
| RA4/T0CKI/C1OUT | RA4 | TTL | CMOS | General purpose I/O. |
| | T0CKI | ST | — | Timer0 clock input. |
| | C1OUT | — | CMOS | Comparator C1 output. |
| RA5/AN4/$\overline{SS}$/C2OUT | RA5 | TTL | CMOS | General purpose I/O. |
| | AN4 | AN | — | A/D Channel 4. |
| | $\overline{SS}$ | ST | — | Slave Select input. |
| | C2OUT | — | CMOS | Comparator C2 output. |
| RA6/OSC2/CLKOUT | RA6 | TTL | CMOS | General purpose I/O. |
| | OSC2 | — | XTAL | Master Clear with internal pull-up. |
| | CLKOUT | — | CMOS | FOSC/4 output. |
| RA7/OSC1/CLKIN | RA7 | TTL | CMOS | General purpose I/O. |
| | OSC1 | XTAL | — | Crystal/Resonator. |
| | CLKIN | ST | — | External clock input/RC oscillator connection. |
| RB0/AN12/INT | RB0 | TTL | CMOS | General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up. |
| | AN12 | AN | — | A/D Channel 12. |
| | INT | ST | — | External interrupt. |
| RB1/AN10/P1C/C12IN3- | RB1 | TTL | CMOS | General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up. |
| | AN10 | AN | — | A/D Channel 10. |
| | P1C | — | CMOS | PWM output. |
| | C12IN3- | AN | — | Comparator C1 or C2 negative input. |
| RB2/AN8/P1B | RB2 | TTL | CMOS | General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up. |
| | AN8 | AN | — | A/D Channel 8. |
| | P1B | — | CMOS | PWM output. |

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


© 2006-2012 Microchip Technology Inc.

**TABLE 1-1: PIC16F882/883/886 PINOUT DESCRIPTION (CONTINUED)**

| Name | Function | Input Type | Output Type | Description |
|---|---|---|---|---|
| RB3/AN9/PGM/C12IN2- | RB3 | TTL | CMOS | General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up. |
| | AN9 | AN | — | A/D Channel 9. |
| | PGM | ST | — | Low-voltage ICSP™ Programming enable pin. |
| | C12IN2- | AN | — | Comparator C1 or C2 negative input. |
| RB4/AN11/P1D | RB4 | TTL | CMOS | General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up. |
| | AN11 | AN | — | A/D Channel 11. |
| | P1D | — | CMOS | PWM output. |
| RB5/AN13/$\overline{T1G}$ | RB5 | TTL | CMOS | General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up. |
| | AN13 | AN | — | A/D Channel 13. |
| | $\overline{T1G}$ | ST | — | Timer1 Gate input. |
| RB6/ICSPCLK | RB6 | TTL | CMOS | General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up. |
| | ICSPCLK | ST | — | Serial Programming Clock. |
| RB7/ICSPDAT | RB7 | TTL | CMOS | General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up. |
| | ICSPDAT | ST | CMOS | ICSP™ Data I/O. |
| RC0/T1OSO/T1CKI | RC0 | ST | CMOS | General purpose I/O. |
| | T1OSO | — | CMOS | Timer1 oscillator output. |
| | T1CKI | ST | — | Timer1 clock input. |
| RC1/T1OSI/CCP2 | RC1 | ST | CMOS | General purpose I/O. |
| | T1OSI | ST | — | Timer1 oscillator input. |
| | CCP2 | ST | CMOS | Capture/Compare/PWM2. |
| RC2/P1A/CCP1 | RC2 | ST | CMOS | General purpose I/O. |
| | P1A | — | CMOS | PWM output. |
| | CCP1 | ST | CMOS | Capture/Compare/PWM1. |
| RC3/SCK/SCL | RC3 | ST | CMOS | General purpose I/O. |
| | SCK | ST | CMOS | SPI clock. |
| | SCL | ST | OD | $I^2C$™ clock. |
| RC4/SDI/SDA | RC4 | ST | CMOS | General purpose I/O. |
| | SDI | ST | — | SPI data input. |
| | SDA | ST | OD | $I^2C$ data input/output. |
| RC5/SDO | RC5 | ST | CMOS | General purpose I/O. |
| | SDO | — | CMOS | SPI data output. |
| RC6/TX/CK | RC6 | ST | CMOS | General purpose I/O. |
| | TX | — | CMOS | EUSART asynchronous transmit. |
| | CK | ST | CMOS | EUSART synchronous clock. |
| RC7/RX/DT | RC7 | ST | CMOS | General purpose I/O. |
| | RX | ST | — | EUSART asynchronous input. |
| | DT | ST | CMOS | EUSART synchronous data. |
| RE3/$\overline{MCLR}$/$V_{PP}$ | RE3 | TTL | — | General purpose input. |
| | $\overline{MCLR}$ | ST | — | Master Clear with internal pull-up. |
| | $V_{PP}$ | HV | — | Programming voltage. |
| $V_{SS}$ | $V_{SS}$ | Power | — | Ground reference. |
| $V_{DD}$ | $V_{DD}$ | Power | — | Positive supply. |

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

**TABLE 1-2: PIC16F884/887 PINOUT DESCRIPTION**

| Name | Function | Input Type | Output Type | Description |
|---|---|---|---|---|
| RA0/AN0/ULPWU/C12IN0- | RA0 | TTL | CMOS | General purpose I/O. |
| | AN0 | AN | — | A/D Channel 0 input. |
| | ULPWU | AN | — | Ultra Low-Power Wake-up input. |
| | C12IN0- | AN | — | Comparator C1 or C2 negative input. |
| RA1/AN1/C12IN1- | RA1 | TTL | CMOS | General purpose I/O. |
| | AN1 | AN | — | A/D Channel 1 input. |
| | C12IN1- | AN | — | Comparator C1 or C2 negative input. |
| RA2/AN2/VREF-/CVREF/C2IN+ | RA2 | TTL | CMOS | General purpose I/O. |
| | AN2 | AN | — | A/D Channel 2. |
| | VREF- | AN | — | A/D Negative Voltage Reference input. |
| | CVREF | — | AN | Comparator Voltage Reference output. |
| | C2IN+ | AN | — | Comparator C2 positive input. |
| RA3/AN3/VREF+/C1IN+ | RA3 | TTL | CMOS | General purpose I/O. |
| | AN3 | AN | — | A/D Channel 3. |
| | VREF+ | AN | — | A/D Positive Voltage Reference input. |
| | C1IN+ | AN | — | Comparator C1 positive input. |
| RA4/T0CKI/C1OUT | RA4 | TTL | CMOS | General purpose I/O. |
| | T0CKI | ST | — | Timer0 clock input. |
| | C1OUT | — | CMOS | Comparator C1 output. |
| RA5/AN4/$\overline{SS}$/C2OUT | RA5 | TTL | CMOS | General purpose I/O. |
| | AN4 | AN | — | A/D Channel 4. |
| | $\overline{SS}$ | ST | — | Slave Select input. |
| | C2OUT | — | CMOS | Comparator C2 output. |
| RA6/OSC2/CLKOUT | RA6 | TTL | CMOS | General purpose I/O. |
| | OSC2 | — | XTAL | Crystal/Resonator. |
| | CLKOUT | — | CMOS | FOSC/4 output. |
| RA7/OSC1/CLKIN | RA7 | TTL | CMOS | General purpose I/O. |
| | OSC1 | XTAL | — | Crystal/Resonator. |
| | CLKIN | ST | — | External clock input/RC oscillator connection. |
| RB0/AN12/INT | RB0 | TTL | CMOS | General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up. |
| | AN12 | AN | — | A/D Channel 12. |
| | INT | ST | — | External interrupt. |
| RB1/AN10/C12IN3- | RB1 | TTL | CMOS | General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up. |
| | AN10 | AN | — | A/D Channel 10. |
| | C12IN3- | AN | — | Comparator C1 or C2 negative input. |
| RB2/AN8 | RB2 | TTL | CMOS | General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up. |
| | AN8 | AN | — | A/D Channel 8. |
| RB3/AN9/PGM/C12IN2- | RB3 | TTL | CMOS | General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up. |
| | AN9 | AN | — | A/D Channel 9. |
| | PGM | ST | — | Low-voltage ICSP™ Programming enable pin. |
| | C12IN2- | AN | — | Comparator C1 or C2 negative input. |

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


© 2006-2012 Microchip Technology Inc.

**TABLE 1-2: PIC16F884/887 PINOUT DESCRIPTION (CONTINUED)**

| Name | Function | Input Type | Output Type | Description |
|---|---|---|---|---|
| RB4/AN11 | RB4 | TTL | CMOS | General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up. |
| | AN11 | AN | — | A/D Channel 11. |
| RB5/AN13/$\overline{T1G}$ | RB5 | TTL | CMOS | General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up. |
| | AN13 | AN | — | A/D Channel 13. |
| | $\overline{T1G}$ | ST | — | Timer1 Gate input. |
| RB6/ICSPCLK | RB6 | TTL | CMOS | General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up. |
| | ICSPCLK | ST | — | Serial Programming Clock. |
| RB7/ICSPDAT | RB7 | TTL | CMOS | General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up. |
| | ICSPDAT | ST | TTL | ICSP™ Data I/O. |
| RC0/T1OSO/T1CKI | RC0 | ST | CMOS | General purpose I/O. |
| | T1OSO | — | XTAL | Timer1 oscillator output. |
| | T1CKI | ST | — | Timer1 clock input. |
| RC1/T1OSI/CCP2 | RC1 | ST | CMOS | General purpose I/O. |
| | T1OSI | XTAL | — | Timer1 oscillator input. |
| | CCP2 | ST | CMOS | Capture/Compare/PWM2. |
| RC2/P1A/CCP1 | RC2 | ST | CMOS | General purpose I/O. |
| | P1A | ST | CMOS | PWM output. |
| | CCP1 | — | CMOS | Capture/Compare/PWM1. |
| RC3/SCK/SCL | RC3 | ST | CMOS | General purpose I/O. |
| | SCK | ST | CMOS | SPI clock. |
| | SCL | ST | OD | $I^2C$™ clock. |
| RC4/SDI/SDA | RC4 | ST | CMOS | General purpose I/O. |
| | SDI | ST | — | SPI data input. |
| | SDA | ST | OD | $I^2C$ data input/output. |
| RC5/SDO | RC5 | ST | CMOS | General purpose I/O. |
| | SDO | — | CMOS | SPI data output. |
| RC6/TX/CK | RC6 | ST | CMOS | General purpose I/O. |
| | TX | — | CMOS | EUSART asynchronous transmit. |
| | CK | ST | CMOS | EUSART synchronous clock. |
| RC7/RX/DT | RC7 | ST | CMOS | General purpose I/O. |
| | RX | ST | — | EUSART asynchronous input. |
| | DT | ST | CMOS | EUSART synchronous data. |
| RD0 | RD0 | TTL | CMOS | General purpose I/O. |
| RD1 | RD1 | TTL | CMOS | General purpose I/O. |
| RD2 | RD2 | TTL | CMOS | General purpose I/O. |
| RD3 | RD3 | TTL | CMOS | General purpose I/O. |
| RD4 | RD4 | TTL | CMOS | General purpose I/O. |
| RD5/P1B | RD5 | TTL | CMOS | General purpose I/O. |
| | P1B | — | CMOS | PWM output. |
| RD6/P1C | RD6 | TTL | CMOS | General purpose I/O. |
| | P1C | — | CMOS | PWM output. |

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

**TABLE 1-2: PIC16F884/887 PINOUT DESCRIPTION (CONTINUED)**

| Name | Function | Input Type | Output Type | Description |
|---|---|---|---|---|
| RD7/P1D | RD7 | TTL | CMOS | General purpose I/O. |
| | P1D | AN | — | PWM output. |
| RE0/AN5 | RE0 | TTL | CMOS | General purpose I/O. |
| | AN5 | AN | — | A/D Channel 5. |
| RE1/AN6 | RE1 | TTL | CMOS | General purpose I/O. |
| | AN6 | AN | — | A/D Channel 6. |
| RE2/AN7 | RE2 | TTL | CMOS | General purpose I/O. |
| | AN7 | AN | — | A/D Channel 7. |
| RE3/$\overline{MCLR}$/$V_{PP}$ | RE3 | TTL | — | General purpose input. |
| | $\overline{MCLR}$ | ST | — | Master Clear with internal pull-up. |
| | $V_{PP}$ | HV | — | Programming voltage. |
| $V_{SS}$ | $V_{SS}$ | Power | — | Ground reference. |
| $V_{DD}$ | $V_{DD}$ | Power | — | Positive supply. |

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


© 2006-2012 Microchip Technology Inc.

## 2.0 MEMORY ORGANIZATION

### 2.1 Program Memory Organization

The PIC16F882/883/884/886/887 devices have a 13-bit program counter capable of addressing a 2K x 14 (0000h-07FFh) for the PIC16F882, 4K x 14 (0000h-0FFFh) for the PIC16F883/PIC16F884, and 8K x 14 (0000h-1FFFh) for the PIC16F886/PIC16F887 program memory space. Accessing a location above these boundaries will cause a wrap-around within the first 8K x 14 space. The Reset vector is at 0000h and the interrupt vector is at 0004h (see Figures 2-2 and 2-3).

**FIGURE 2-1: PROGRAM MEMORY MAP AND STACK FOR THE PIC16F882**

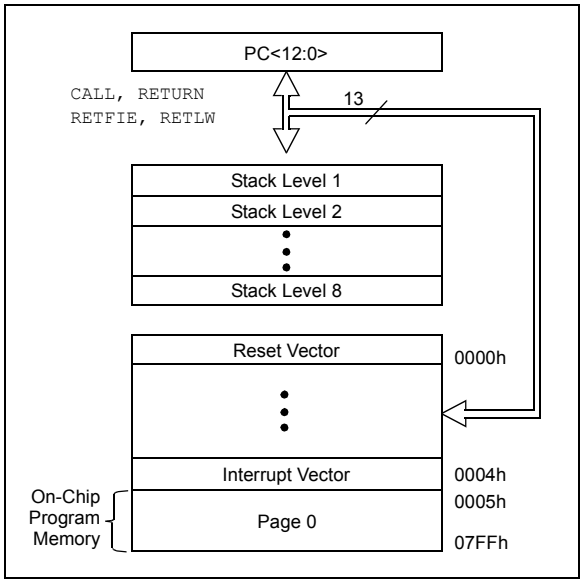



**FIGURE 2-2: PROGRAM MEMORY MAP AND STACK FOR THE PIC16F883/PIC16F884**

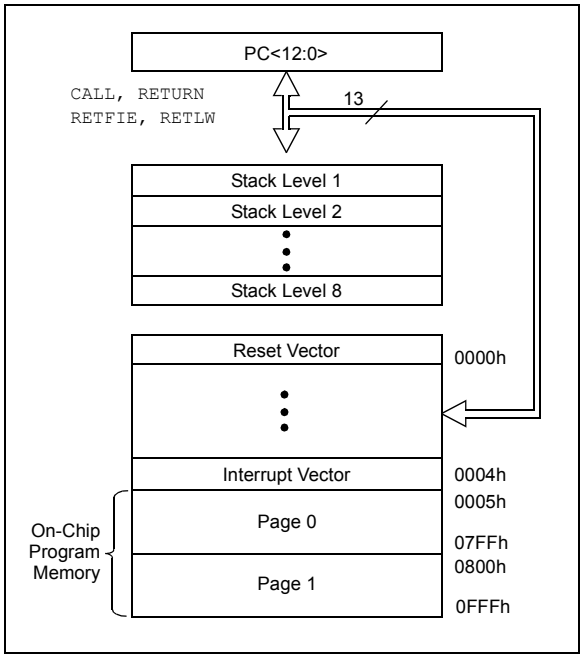
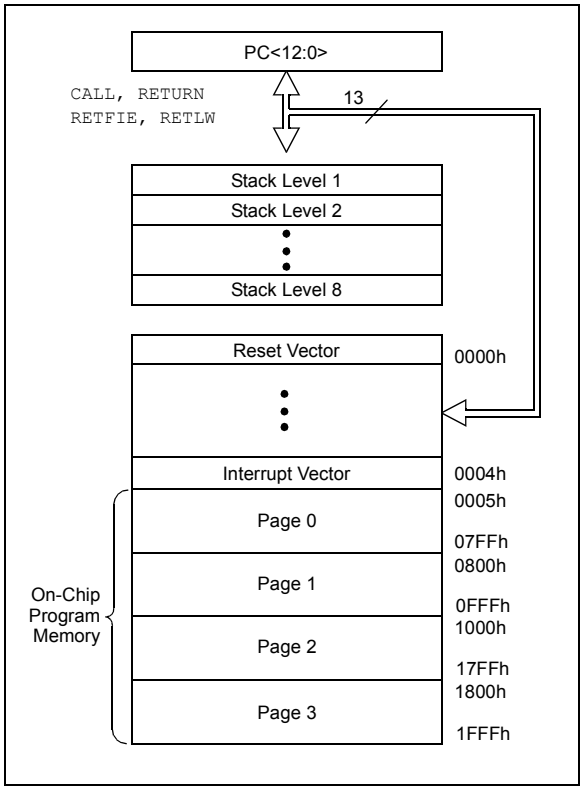



**FIGURE 2-3: PROGRAM MEMORY MAP AND STACK FOR THE PIC16F886/PIC16F887**



## 2.2 Data Memory Organization

The data memory (see Figures 2-2 and 2-3) 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. The General Purpose Registers, implemented as static RAM, are located in the last 96 locations of each Bank. Register locations F0h-FFh in Bank 1, 170h-17Fh in Bank 2 and 1F0h-1FFh in Bank 3, point to addresses 70h-7Fh in Bank 0. The actual number of General Purpose Resisters (GPR) implemented in each Bank depends on the device. Details are shown in Figures 2-5 and 2-6. All other RAM is unimplemented and returns '0' when read. RP<1:0> 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 128 x 8 in the PIC16F882, 256 x 8 in the PIC16F883/PIC16F884, and 368 x 8 in the PIC16F886/PIC16F887. Each register is accessed, either directly or indirectly, through the File Select Register (FSR) (see **Section 2.4 "Indirect Addressing, INDF and FSR Registers"**).

### 2.2.2 SPECIAL FUNCTION REGISTERS

The Special Function Registers are registers used by the CPU and peripheral functions for controlling the desired operation of the device (see 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.


© 2006-2012 Microchip Technology Inc.

**FIGURE 2-4: PIC16F882 SPECIAL FUNCTION REGISTERS**

| | File Address | | File Address | | File Address | | File Address |
|---|---|---|---|---|---|---|---|
| Indirect addr. (1) | 00h | Indirect addr. (1) | 80h | Indirect addr. (1) | 100h | Indirect addr. (1) | 180h |
| TMR0 | 01h | OPTION_REG | 81h | TMR0 | 101h | OPTION_REG | 181h |
| PCL | 02h | PCL | 82h | PCL | 102h | PCL | 182h |
| STATUS | 03h | STATUS | 83h | STATUS | 103h | STATUS | 183h |
| FSR | 04h | FSR | 84h | FSR | 104h | FSR | 184h |
| PORTA | 05h | TRISA | 85h | WDTCON | 105h | SRCON | 185h |
| PORTB | 06h | TRISB | 86h | PORTB | 106h | TRISB | 186h |
| PORTC | 07h | TRISC | 87h | CM1CON0 | 107h | BAUDCTL | 187h |
| | 08h | | 88h | CM2CON0 | 108h | ANSEL | 188h |
| PORTE | 09h | TRISE | 89h | CM2CON1 | 109h | ANSELH | 189h |
| PCLATH | 0Ah | PCLATH | 8Ah | PCLATH | 10Ah | PCLATH | 18Ah |
| INTCON | 0Bh | INTCON | 8Bh | INTCON | 10Bh | INTCON | 18Bh |
| PIR1 | 0Ch | PIE1 | 8Ch | EEDAT | 10Ch | EECON1 | 18Ch |
| PIR2 | 0Dh | PIE2 | 8Dh | EEADR | 10Dh | EECON2(1) | 18Dh |
| TMR1L | 0Eh | PCON | 8Eh | EEDATH | 10Eh | Reserved | 18Eh |
| TMR1H | 0Fh | OSCCON | 8Fh | EEADRH | 10Fh | Reserved | 18Fh |
| T1CON | 10h | OSCTUNE | 90h | | 110h | | 190h |
| TMR2 | 11h | SSPCON2 | 91h | | 111h | | 191h |
| T2CON | 12h | PR2 | 92h | | 112h | | 192h |
| SSPBUF | 13h | SSPADD | 93h | | 113h | | 193h |
| SSPCON | 14h | SSPSTAT | 94h | | 114h | | 194h |
| CCPR1L | 15h | WPUB | 95h | | 115h | | 195h |
| CCPR1H | 16h | IOCB | 96h | | 116h | | 196h |
| CCP1CON | 17h | VRCON | 97h | | 117h | | 197h |
| RCSTA | 18h | TXSTA | 98h | | 118h | | 198h |
| TXREG | 19h | SPBRG | 99h | | 119h | | 199h |
| RCREG | 1Ah | SPBRGH | 9Ah | | 11Ah | | 19Ah |
| CCPR2L | 1Bh | PWM1CON | 9Bh | | 11Bh | | 19Bh |
| CCPR2H | 1Ch | ECCPAS | 9Ch | | 11Ch | | 19Ch |
| CCP2CON | 1Dh | PSTRCON | 9Dh | | 11Dh | | 19Dh |
| ADRESH | 1Eh | ADRESL | 9Eh | | 11Eh | | 19Eh |
| ADCON0 | 1Fh | ADCON1 | 9Fh | | 11Fh | | 19Fh |
| | 20h | General Purpose Registers | A0h | | 120h | | 1A0h |
| General Purpose Registers | | 32 Bytes | BFh | | | | |
| | | | C0h | | | | |
| 96 Bytes | | | EFh | | 16Fh | | 1EFh |
| | | accesses 70h-7Fh | F0h | accesses 70h-7Fh | 170h | accesses 70h-7Fh | 1F0h |
| | 7Fh | | FFh | | 17Fh | | 1FFh |
| Bank 0 | | Bank 1 | | Bank 2 | | Bank 3 | |

Unimplemented data memory locations, read as '0'.

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

**FIGURE 2-5: PIC16F883/PIC16F884 SPECIAL FUNCTION REGISTERS**

| | File Address | | File Address | | File Address | | File Address |
|---|---|---|---|---|---|---|---|
| Indirect addr. (1) | 00h | Indirect addr. (1) | 80h | Indirect addr. (1) | 100h | Indirect addr. (1) | 180h |
| TMR0 | 01h | OPTION_REG | 81h | TMR0 | 101h | OPTION_REG | 181h |
| PCL | 02h | PCL | 82h | PCL | 102h | PCL | 182h |
| STATUS | 03h | STATUS | 83h | STATUS | 103h | STATUS | 183h |
| FSR | 04h | FSR | 84h | FSR | 104h | FSR | 184h |
| PORTA | 05h | TRISA | 85h | WDTCON | 105h | SRCON | 185h |
| PORTB | 06h | TRISB | 86h | PORTB | 106h | TRISB | 186h |
| PORTC | 07h | TRISC | 87h | CM1CON0 | 107h | BAUDCTL | 187h |
| PORTD(2) | 08h | TRISD(2) | 88h | CM2CON0 | 108h | ANSEL | 188h |
| PORTE | 09h | TRISE | 89h | CM2CON1 | 109h | ANSELH | 189h |
| PCLATH | 0Ah | PCLATH | 8Ah | PCLATH | 10Ah | PCLATH | 18Ah |
| INTCON | 0Bh | INTCON | 8Bh | INTCON | 10Bh | INTCON | 18Bh |
| PIR1 | 0Ch | PIE1 | 8Ch | EEDAT | 10Ch | EECON1 | 18Ch |
| PIR2 | 0Dh | PIE2 | 8Dh | EEADR | 10Dh | EECON2(1) | 18Dh |
| TMR1L | 0Eh | PCON | 8Eh | EEDATH | 10Eh | Reserved | 18Eh |
| TMR1H | 0Fh | OSCCON | 8Fh | EEADRH | 10Fh | Reserved | 18Fh |
| T1CON | 10h | OSCTUNE | 90h | | 110h | | 190h |
| TMR2 | 11h | SSPCON2 | 91h | | 111h | | 191h |
| T2CON | 12h | PR2 | 92h | | 112h | | 192h |
| SSPBUF | 13h | SSPADD | 93h | | 113h | | 193h |
| SSPCON | 14h | SSPSTAT | 94h | | 114h | | 194h |
| CCPR1L | 15h | WPUB | 95h | | 115h | | 195h |
| CCPR1H | 16h | IOCB | 96h | | 116h | | 196h |
| CCP1CON | 17h | VRCON | 97h | | 117h | | 197h |
| RCSTA | 18h | TXSTA | 98h | | 118h | | 198h |
| TXREG | 19h | SPBRG | 99h | | 119h | | 199h |
| RCREG | 1Ah | SPBRGH | 9Ah | | 11Ah | | 19Ah |
| CCPR2L | 1Bh | PWM1CON | 9Bh | | 11Bh | | 19Bh |
| CCPR2H | 1Ch | ECCPAS | 9Ch | | 11Ch | | 19Ch |
| CCP2CON | 1Dh | PSTRCON | 9Dh | | 11Dh | | 19Dh |
| ADRESH | 1Eh | ADRESL | 9Eh | | 11Eh | | 19Eh |
| ADCON0 | 1Fh | ADCON1 | 9Fh | | 11Fh | | 19Fh |
| General Purpose Registers 96 Bytes | 20h – 7Fh | General Purpose Registers 80 Bytes; accesses 70h-7Fh | A0h – FFh | General Purpose Registers 80 Bytes; accesses 70h-7Fh | 120h – 17Fh | accesses 70h-7Fh | 1A0h – 1FFh |
| Bank 0 | | Bank 1 | | Bank 2 | | Bank 3 | |

Unimplemented data memory locations, read as '0'.

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

**2:** PIC16F884 only.


© 2006-2012 Microchip Technology Inc.

**FIGURE 2-6: PIC16F886/PIC16F887 SPECIAL FUNCTION REGISTERS**

| Bank 0 | File Address | Bank 1 | File Address | Bank 2 | File Address | Bank 3 | File Address |
|---|---|---|---|---|---|---|---|
| Indirect addr. (1) | 00h | Indirect addr. (1) | 80h | Indirect addr. (1) | 100h | Indirect addr. (1) | 180h |
| TMR0 | 01h | OPTION_REG | 81h | TMR0 | 101h | OPTION_REG | 181h |
| PCL | 02h | PCL | 82h | PCL | 102h | PCL | 182h |
| STATUS | 03h | STATUS | 83h | STATUS | 103h | STATUS | 183h |
| FSR | 04h | FSR | 84h | FSR | 104h | FSR | 184h |
| PORTA | 05h | TRISA | 85h | WDTCON | 105h | SRCON | 185h |
| PORTB | 06h | TRISB | 86h | PORTB | 106h | TRISB | 186h |
| PORTC | 07h | TRISC | 87h | CM1CON0 | 107h | BAUDCTL | 187h |
| PORTD(2) | 08h | TRISD(2) | 88h | CM2CON0 | 108h | ANSEL | 188h |
| PORTE | 09h | TRISE | 89h | CM2CON1 | 109h | ANSELH | 189h |
| PCLATH | 0Ah | PCLATH | 8Ah | PCLATH | 10Ah | PCLATH | 18Ah |
| INTCON | 0Bh | INTCON | 8Bh | INTCON | 10Bh | INTCON | 18Bh |
| PIR1 | 0Ch | PIE1 | 8Ch | EEDAT | 10Ch | EECON1 | 18Ch |
| PIR2 | 0Dh | PIE2 | 8Dh | EEADR | 10Dh | EECON2(1) | 18Dh |
| TMR1L | 0Eh | PCON | 8Eh | EEDATH | 10Eh | Reserved | 18Eh |
| TMR1H | 0Fh | OSCCON | 8Fh | EEADRH | 10Fh | Reserved | 18Fh |
| T1CON | 10h | OSCTUNE | 90h | | 110h | | 190h |
| TMR2 | 11h | SSPCON2 | 91h | | 111h | | 191h |
| T2CON | 12h | PR2 | 92h | | 112h | | 192h |
| SSPBUF | 13h | SSPADD | 93h | | 113h | | 193h |
| SSPCON | 14h | SSPSTAT | 94h | | 114h | | 194h |
| CCPR1L | 15h | WPUB | 95h | | 115h | | 195h |
| CCPR1H | 16h | IOCB | 96h | General Purpose Registers | 116h | General Purpose Registers | 196h |
| CCP1CON | 17h | VRCON | 97h | | 117h | | 197h |
| RCSTA | 18h | TXSTA | 98h | | 118h | | 198h |
| TXREG | 19h | SPBRG | 99h | 16 Bytes | 119h | 16 Bytes | 199h |
| RCREG | 1Ah | SPBRGH | 9Ah | | 11Ah | | 19Ah |
| CCPR2L | 1Bh | PWM1CON | 9Bh | | 11Bh | | 19Bh |
| CCPR2H | 1Ch | ECCPAS | 9Ch | | 11Ch | | 19Ch |
| CCP2CON | 1Dh | PSTRCON | 9Dh | | 11Dh | | 19Dh |
| ADRESH | 1Eh | ADRESL | 9Eh | | 11Eh | | 19Eh |
| ADCON0 | 1Fh | ADCON1 | 9Fh | | 11Fh | | 19Fh |
| General Purpose Registers 96 Bytes | 20h – 6Fh | General Purpose Registers 80 Bytes | A0h – EFh | General Purpose Registers 80 Bytes | 120h – 16Fh | General Purpose Registers 80 Bytes | 1A0h – 1EFh |
| | 70h – 7Fh | accesses 70h-7Fh | F0h – FFh | accesses 70h-7Fh | 170h – 17Fh | accesses 70h-7Fh | 1F0h – 1FFh |

Unimplemented data memory locations, read as '0'.

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

**2:** PIC16F887 only.

**TABLE 2-1: PIC16F882/883/884/886/887 SPECIAL FUNCTION 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 | Addressing this location uses contents of FSR to address data memory (not a physical register) | | | | | | | | xxxx xxxx | xxxx xxxx |
| 01h | TMR0 | Timer0 Module Register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 02h | PCL | Program Counter's (PC) Least Significant Byte | | | | | | | | 0000 0000 | 0000 0000 |
| 03h | STATUS | IRP | RP1 | RP0 | $\overline{TO}$ | $\overline{PD}$ | Z | DC | C | 0001 1xxx | 000q quuu(5) |
| 04h | FSR | Indirect Data Memory Address Pointer | | | | | | | | xxxx xxxx | uuuu uuuu |
| 05h | PORTA(3) | RA7 | RA6 | RA5 | RA4 | RA3 | RA2 | RA1 | RA0 | xxxx xxxx | 0000 0000 |
| 06h | PORTB(3) | RB7 | RB6 | RB5 | RB4 | RB3 | RB2 | RB1 | RB0 | xxxx xxxx | 0000 0000 |
| 07h | PORTC(3) | RC7 | RC6 | RC5 | RC4 | RC3 | RC2 | RC1 | RC0 | xxxx xxxx | 0000 0000 |
| 08h | PORTD(3,4) | RD7 | RD6 | RD5 | RD4 | RD3 | RD2 | RD1 | RD0 | xxxx xxxx | 0000 0000 |
| 09h | PORTE(3) | — | — | — | — | RE3 | RE2(4) | RE1(4) | RE0(4) | ---- xxxx | ---- 0000 |
| 0Ah | PCLATH | — | — | — | Write Buffer for upper 5 bits of Program Counter | | | | | ---0 0000 | ---0 0000 |
| 0Bh | INTCON | GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | RBIF(1) | 0000 000x | 0000 000u |
| 0Ch | PIR1 | — | ADIF | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | -000 0000 | 0000 0000 |
| 0Dh | PIR2 | OSFIF | C2IF | C1IF | EEIF | BCLIF | ULPWUIF | — | CCP2IF | 0000 00-0 | 0000 0000 |
| 0Eh | TMR1L | Holding Register for the Least Significant Byte of the 16-bit TMR1 Register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 0Fh | TMR1H | Holding Register for the Most Significant Byte of the 16-bit TMR1 Register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 10h | T1CON | T1GINV | TMR1GE | T1CKPS1 | T1CKPS0 | T1OSCEN | $\overline{T1SYNC}$ | TMR1CS | TMR1ON | 0000 0000 | uuuu uuuu |
| 11h | TMR2 | Timer2 Module Register | | | | | | | | 0000 0000 | 0000 0000 |
| 12h | T2CON | — | TOUTPS3 | TOUTPS2 | TOUTPS1 | TOUTPS0 | TMR2ON | T2CKPS1 | T2CKPS0 | -000 0000 | -000 0000 |
| 13h | SSPBUF | Synchronous Serial Port Receive Buffer/Transmit Register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 14h | SSPCON(2) | WCOL | SSPOV | SSPEN | CKP | SSPM3 | SSPM2 | SSPM1 | SSPM0 | 0000 0000 | 0000 0000 |
| 15h | CCPR1L | Capture/Compare/PWM Register 1 Low Byte (LSB) | | | | | | | | xxxx xxxx | uuuu uuuu |
| 16h | CCPR1H | Capture/Compare/PWM Register 1 High Byte (MSB) | | | | | | | | xxxx xxxx | uuuu uuuu |
| 17h | CCP1CON | P1M1 | P1M0 | DC1B1 | DC1B0 | CCP1M3 | CCP1M2 | CCP1M1 | CCP1M0 | 0000 0000 | 0000 0000 |
| 18h | RCSTA | SPEN | RX9 | SREN | CREN | ADDEN | FERR | OERR | RX9D | 0000 000x | 0000 0000 |
| 19h | TXREG | EUSART Transmit Data Register | | | | | | | | 0000 0000 | 0000 0000 |
| 1Ah | RCREG | EUSART Receive Data Register | | | | | | | | 0000 0000 | 0000 0000 |
| 1Bh | CCPR2L | Capture/Compare/PWM Register 2 Low Byte (LSB) | | | | | | | | xxxx xxxx | uuuu uuuu |
| 1Ch | CCPR2H | Capture/Compare/PWM Register 2 High Byte (MSB) | | | | | | | | xxxx xxxx | uuuu uuuu |
| 1Dh | CCP2CON | — | — | DC2B1 | DC2B0 | CCP2M3 | CCP2M2 | CCP2M1 | CCP2M0 | --00 0000 | --00 000 |
| 1Eh | ADRESH | A/D Result Register High Byte | | | | | | | | xxxx xxxx | uuuu uuuu |
| 1Fh | ADCON0 | ADCS1 | ADCS0 | CHS3 | CHS2 | CHS1 | CHS0 | $\overline{GO}$/DONE | ADON | 0000 0000 | 00-0 0000 |

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

**Note 1:** $\overline{MCLR}$ and WDT Reset do not affect the previous value data latch. The RBIF bit will be cleared upon Reset but will set again if the mismatch exists.

**2:** When SSPCON register bits SSPM<3:0> = 1001, any reads or writes to the SSPADD SFR address are accessed through the SSPMSK register. See Registers 13-2 and 13-4 for more details.

**3:** Port pins with analog functions controlled by the ANSEL and ANSELH registers will read '0' immediately after a Reset even though the data latches are either undefined (POR) or unchanged (other Resets).

**4:** PIC16F884/PIC16F887 only.

**5:** See Table 14-5 for Reset value for specific condition.


© 2006-2012 Microchip Technology Inc.

**TABLE 2-2: PIC16F882/883/884/886/887 SPECIAL FUNCTION REGISTERS SUMMARY BANK 1**

| Addr | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR, BOR | Value on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Bank 1** | | | | | | | | | | | |
| 80h | INDF | Addressing this location uses contents of FSR to address data memory (not a physical register) | | | | | | | | xxxx xxxx | xxxx xxxx |
| 81h | OPTION_REG | $\overline{RBPU}$ | INTEDG | T0CS | T0SE | PSA | PS2 | PS1 | PS0 | 1111 1111 | 1111 1111 |
| 82h | PCL | Program Counter's (PC) Least Significant Byte | | | | | | | | 0000 0000 | 0000 0000 |
| 83h | STATUS | IRP | RP1 | RP0 | $\overline{TO}$ | $\overline{PD}$ | Z | DC | C | 0001 1xxx | 000q quuu(5) |
| 84h | FSR | Indirect Data Memory Address Pointer | | | | | | | | xxxx xxxx | uuuu uuuu |
| 85h | TRISA | TRISA7 | TRISA6 | TRISA5 | TRISA4 | TRISA3 | TRISA2 | TRISA1 | TRISA0 | 1111 1111 | 1111 1111 |
| 86h | TRISB | TRISB7 | TRISB6 | TRISB5 | TRISB4 | TRISB3 | TRISB2 | TRISB1 | TRISB0 | 1111 1111 | 1111 1111 |
| 87h | TRISC | TRISC7 | TRISC6 | TRISC5 | TRISC4 | TRISC3 | TRISC2 | TRISC1 | TRISC0 | 1111 1111 | 1111 1111 |
| 88h | TRISD(3) | TRISD7 | TRISD6 | TRISD5 | TRISD4 | TRISD3 | TRISD2 | TRISD1 | TRISD0 | 1111 1111 | 1111 1111 |
| 89h | TRISE | — | — | — | — | TRISE3 | TRISE2(3) | TRISE1(3) | TRISE0(3) | ---- 1111 | ---- 1111 |
| 8Ah | PCLATH | — | — | — | Write Buffer for the upper 5 bits of the Program Counter | | | | | ---0 0000 | ---0 0000 |
| 8Bh | INTCON | GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | RBIF(1) | 0000 000x | 0000 000u |
| 8Ch | PIE1 | — | ADIE | RCIE | TXIE | SSPIE | CCP1IE | TMR2IE | TMR1IE | -000 0000 | 0000 0000 |
| 8Dh | PIE2 | OSFIE | C2IE | C1IE | EEIE | BCLIE | ULPWUIE | — | CCP2IE | 0000 00-0 | 0000 0000 |
| 8Eh | PCON | — | — | ULPWUE | SBOREN | — | — | $\overline{POR}$ | $\overline{BOR}$ | --01 --qq | --0u --uu(4,6) |
| 8Fh | OSCCON | — | IRCF2 | IRCF1 | IRCF0 | OSTS | HTS | LTS | SCS | -110 q000 | -110 q000 |
| 90h | OSCTUNE | — | — | — | TUN4 | TUN3 | TUN2 | TUN1 | TUN0 | ---0 0000 | ---u uuuu |
| 91h | SSPCON2 | GCEN | ACKSTAT | ACKDT | ACKEN | RCEN | PEN | RSEN | SEN | 0000 0000 | 0000 0000 |
| 92h | PR2 | Timer2 Period Register | | | | | | | | 1111 1111 | 1111 1111 |
| 93h | SSPADD(2) | Synchronous Serial Port ($I^2C$ mode) Address Register | | | | | | | | 0000 0000 | 0000 0000 |
| 93h | SSPMSK(2) | MSK7 | MSK6 | MSK5 | MSK4 | MSK3 | MSK2 | MSK1 | MSK0 | 1111 1111 | 1111 1111 |
| 94h | SSPSTAT | SMP | CKE | D/$\overline{A}$ | P | S | R/$\overline{W}$ | UA | BF | 0000 0000 | 0000 0000 |
| 95h | WPUB | WPUB7 | WPUB6 | WPUB5 | WPUB4 | WPUB3 | WPUB2 | WPUB1 | WPUB0 | 1111 1111 | 1111 1111 |
| 96h | IOCB | IOCB7 | IOCB6 | IOCB5 | IOCB4 | IOCB3 | IOCB2 | IOCB1 | IOCB0 | 0000 0000 | 0000 0000 |
| 97h | VRCON | VREN | VROE | VRR | VRSS | VR3 | VR2 | VR1 | VR0 | 0000 0000 | 0000 0000 |
| 98h | TXSTA | CSRC | TX9 | TXEN | SYNC | SENDB | BRGH | TRMT | TX9D | 0000 0010 | 0000 -010 |
| 99h | SPBRG | BRG7 | BRG6 | BRG5 | BRG4 | BRG3 | BRG2 | BRG1 | BRG0 | 0000 0000 | 0000 0000 |
| 9Ah | SPBRGH | BRG15 | BRG14 | BRG13 | BRG12 | BRG11 | BRG10 | BRG9 | BRG8 | 0000 0000 | 0000 0000 |
| 9Bh | PWM1CON | PRSEN | PDC6 | PDC5 | PDC4 | PDC3 | PDC2 | PDC1 | PDC0 | 0000 0000 | 0000 0000 |
| 9Ch | ECCPAS | ECCPASE | ECCPAS2 | ECCPAS1 | ECCPAS0 | PSSAC1 | PSSAC0 | PSSBD1 | PSSBD0 | 0000 0000 | 0000 0000 |
| 9Dh | PSTRCON | — | — | — | STRSYNC | STRD | STRC | STRB | STRA | ---0 0001 | ---0 0001 |
| 9Eh | ADRESL | A/D Result Register Low Byte | | | | | | | | xxxx xxxx | uuuu uuuu |
| 9Fh | ADCON1 | ADFM | — | VCFG1 | VCFG0 | — | — | — | — | 0-00 ---- | 0-00 ---- |

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

**Note 1:** $\overline{MCLR}$ and WDT Reset do not affect the previous value data latch. The RBIF bit will be cleared upon Reset but will set again if the mismatch exists.

**2:** Accessible only when SSPCON register bits SSPM<3:0> = 1001.

**3:** PIC16F884/PIC16F887 only.

**4:** If VDD goes too low, Power-on Reset will be activated and registers will be affected differently.

**5:** See Table 14-5 for Reset value for specific condition.

**6:** If Reset was due to brown-out, then bit 0 = 0. All other Resets will cause bit 0 = u.

**TABLE 2-3: PIC16F882/883/884/886/887 SPECIAL FUNCTION 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 | Addressing this location uses contents of FSR to address data memory (not a physical register) | | | | | | | | xxxx xxxx | xxxx xxxx |
| 101h | TMR0 | Timer0 Module Register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 102h | PCL | Program Counter's (PC) Least Significant Byte | | | | | | | | 0000 0000 | 0000 0000 |
| 103h | STATUS | IRP | RP1 | RP0 | $\overline{TO}$ | $\overline{PD}$ | Z | DC | C | 0001 1xxx | 000q quuu(3) |
| 104h | FSR | Indirect Data Memory Address Pointer | | | | | | | | xxxx xxxx | uuuu uuuu |
| 105h | WDTCON | — | — | — | WDTPS3 | WDTPS2 | WDTPS1 | WDTPS0 | SWDTEN | ---0 1000 | ---0 1000 |
| 106h | PORTB | RB7 | RB6 | RB5 | RB4 | RB3 | RB2 | RB1 | RB0 | xxxx xxxx | 0000 0000 |
| 107h | CM1CON0 | C1ON | C1OUT | C1OE | C1POL | — | C1R | C1CH1 | C1CH0 | 0000 -000 | 0000 0-00 |
| 108h | CM2CON0 | C2ON | C2OUT | C2OE | C2POL | — | C2R | C2CH1 | C2CH0 | 0000 -000 | 0000 0-00 |
| 109h | CM2CON1 | MC1OUT | MC2OUT | C1RSEL | C2RSEL | — | — | T1GSS | C2SYNC | 0000 --10 | 0000 0--0 |
| 10Ah | PCLATH | — | — | — | Write Buffer for the upper 5 bits of the Program Counter | | | | | ---0 0000 | ---0 0000 |
| 10Bh | INTCON | GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | RBIF(1) | 0000 000x | 0000 000u |
| 10Ch | EEDAT | EEDAT7 | EEDAT6 | EEDAT5 | EEDAT4 | EEDAT3 | EEDAT2 | EEDAT1 | EEDAT0 | 0000 0000 | 0000 0000 |
| 10Dh | EEADR | EEADR7 | EEADR6 | EEADR5 | EEADR4 | EEADR3 | EEADR2 | EEADR1 | EEADR0 | 0000 0000 | 0000 0000 |
| 10Eh | EEDATH | — | — | EEDATH5 | EEDATH4 | EEDATH3 | EEDATH2 | EEDATH1 | EEDATH0 | --00 0000 | --00 0000 |
| 10Fh | EEADRH | — | — | — | EEADRH4(2) | EEADRH3 | EEADRH2 | EEADRH1 | EEADRH0 | ---- 0000 | ---0 0000 |

**Legend:** – = Unimplemented locations read as '0', u = unchanged, x = unknown, q = value depends on condition, shaded = unimplemented
**Note 1:** $\overline{MCLR}$ and WDT Reset does not affect the previous value data latch. The RBIF bit will be cleared upon Reset but will set again if the mismatch exists.
**2:** PIC16F886/PIC16F887 only.
**3:** See Table 14-5 for Reset value for specific condition.

**TABLE 2-4: PIC16F882/883/884/886/887 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 | Value on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Bank 3** | | | | | | | | | | | |
| 180h | INDF | Addressing this location uses contents of FSR to address data memory (not a physical register) | | | | | | | | xxxx xxxx | xxxx xxxx |
| 181h | OPTION_REG | $\overline{RBPU}$ | INTEDG | T0CS | T0SE | PSA | PS2 | PS1 | PS0 | 1111 1111 | 1111 1111 |
| 182h | PCL | Program Counter's (PC) Least Significant Byte | | | | | | | | 0000 0000 | 0000 0000 |
| 183h | STATUS | IRP | RP1 | RP0 | $\overline{TO}$ | $\overline{PD}$ | Z | DC | C | 0001 1xxx | 000q quuu(3) |
| 184h | FSR | Indirect Data Memory Address Pointer | | | | | | | | xxxx xxxx | uuuu uuuu |
| 185h | SRCON | SR1 | SR0 | C1SEN | C2REN | PULSS | PULSR | — | FVREN | 0000 00-0 | 0000 00-0 |
| 186h | TRISB | TRISB7 | TRISB6 | TRISB5 | TRISB4 | TRISB3 | TRISB2 | TRISB1 | TRISB0 | 1111 1111 | 1111 1111 |
| 187h | BAUDCTL | ABDOVF | RCIDL | — | SCKP | BRG16 | — | WUE | ABDEN | 01-0 0-00 | 01-0 0-00 |
| 188h | ANSEL | ANS7(2) | ANS6(2) | ANS5(2) | ANS4 | ANS3 | ANS2 | ANS1 | ANS0 | 1111 1111 | 1111 1111 |
| 189h | ANSELH | — | — | ANS13 | ANS12 | ANS11 | ANS10 | ANS9 | ANS8 | --11 1111 | 1111 1111 |
| 18Ah | PCLATH | — | — | — | Write Buffer for the upper 5 bits of the Program Counter | | | | | ---0 0000 | ---0 0000 |
| 18Bh | INTCON | GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | RBIF(1) | 0000 000x | 0000 000u |
| 18Ch | EECON1 | EEPGD | — | — | — | WRERR | WREN | WR | RD | x--- x000 | ---- q000 |
| 18Dh | EECON2 | EEPROM Control Register 2 (not a physical register) | | | | | | | | ---- ---- | ---- ---- |

**Legend:** – = Unimplemented locations read as '0', u = unchanged, x = unknown, q = value depends on condition, shaded = unimplemented
**Note 1:** $\overline{MCLR}$ and WDT Reset does not affect the previous value data latch. The RBIF bit will be cleared upon Reset but will set again if the mismatch exists.
**2:** PIC16F884/PIC16F887 only.
**3:** See Table 14-5 for Reset value for specific condition.


© 2006-2012 Microchip Technology Inc.

#### 2.2.2.1 STATUS Register

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

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

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

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

It is recommended, therefore, that only `BCF`, `BSF`, `SWAPF` and `MOVWF` instructions are used to alter the STATUS register, because these instructions do not affect any Status bits. For other instructions not affecting any Status bits, see **Section 15.0 "Instruction Set Summary"**

**Note 1:** The C and DC bits operate as a $\overline{Borrow}$ and $\overline{Digit\ Borrow}$ out bit, respectively, in subtraction.

## REGISTER DEFINITIONS: STATUS

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

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

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

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

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

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

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

#### 2.2.2.2 OPTION Register

The OPTION register, shown in Register 2-2, is a readable and writable register, which contains various control bits to configure:

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

**Note:** To achieve a 1:1 prescaler assignment for Timer0, assign the prescaler to the WDT by setting PSA bit of the OPTION register to '1'. See **Section 6.3 "Timer1 Prescaler"**.

## REGISTER DEFINITIONS: OPTION REGISTER

### 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 |
|---|---|---|---|---|---|---|---|
| $\overline{\text{RBPU}}$ | INTEDG | T0CS | T0SE | PSA | PS2 | PS1 | PS0 |
| 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 **$\overline{\text{RBPU}}$:** PORTB Pull-up Enable bit
1 = PORTB pull-ups are disabled
0 = PORTB pull-ups are enabled by individual PORT latch values

bit 6 **INTEDG:** Interrupt Edge Select bit
1 = Interrupt on rising edge of 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 |


© 2006-2012 Microchip Technology Inc.

### 2.2.2.3 INTCON Register

The INTCON register, shown in Register 2-3, is a readable and writable register, which contains the various enable and flag bits for TMR0 register overflow, PORTB change and external 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 DEFINITIONS: INTERRUPT CONTROL

### 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-x |
|---|---|---|---|---|---|---|---|
| GIE | PEIE | T0IE | INTE | RBIE(1) | T0IF(2) | INTF | RBIF |
| 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:** INT External Interrupt Enable bit
1 = Enables the INT external interrupt
0 = Disables the INT external interrupt

bit 3 **RBIE:** PORTB Change Interrupt Enable bit(1)
1 = Enables the PORTB change interrupt
0 = Disables the PORTB change interrupt

bit 2 **T0IF:** Timer0 Overflow Interrupt Flag bit(2)
1 = TMR0 register has overflowed (must be cleared in software)
0 = TMR0 register did not overflow

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

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

**Note 1:** IOCB register must also be enabled.
**2:** T0IF bit is set when Timer0 rolls over. Timer0 is unchanged on Reset and should be initialized before clearing T0IF bit.

#### 2.2.2.4 PIE1 Register

The PIE1 register contains the 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 DEFINITIONS: PIE1

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

| U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | ADIE | RCIE | TXIE | SSPIE | CCP1IE | 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 **Unimplemented:** Read as '0'

bit 6 **ADIE:** A/D Converter (ADC) Interrupt Enable bit
1 = Enables the ADC interrupt
0 = Disables the ADC interrupt

bit 5 **RCIE:** EUSART Receive Interrupt Enable bit
1 = Enables the EUSART receive interrupt
0 = Disables the EUSART receive interrupt

bit 4 **TXIE:** EUSART Transmit Interrupt Enable bit
1 = Enables the EUSART transmit interrupt
0 = Disables the EUSART transmit interrupt

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

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

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

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


© 2006-2012 Microchip Technology Inc.

### 2.2.2.5 PIE2 Register

The PIE2 register contains the 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 DEFINITIONS: PIE2

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

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | U-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| OSFIE | C2IE | C1IE | EEIE | BCLIE | ULPWUIE | — | CCP2IE |
| 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 **OSFIE:** Oscillator Fail Interrupt Enable bit
- 1 = Enables oscillator fail interrupt
- 0 = Disables oscillator fail interrupt

bit 6 **C2IE:** Comparator C2 Interrupt Enable bit
- 1 = Enables Comparator C2 interrupt
- 0 = Disables Comparator C2 interrupt

bit 5 **C1IE:** Comparator C1 Interrupt Enable bit
- 1 = Enables Comparator C1 interrupt
- 0 = Disables Comparator C1 interrupt

bit 4 **EEIE:** EEPROM Write Operation Interrupt Enable bit
- 1 = Enables EEPROM write operation interrupt
- 0 = Disables EEPROM write operation interrupt

bit 3 **BCLIE:** Bus Collision Interrupt Enable bit
- 1 = Enables Bus Collision interrupt
- 0 = Disables Bus Collision interrupt

bit 2 **ULPWUIE:** Ultra Low-Power Wake-up Interrupt Enable bit
- 1 = Enables Ultra Low-Power Wake-up interrupt
- 0 = Disables Ultra Low-Power Wake-up interrupt

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

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

### 2.2.2.6 PIR1 Register

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

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

## REGISTER DEFINITIONS: PIR1

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

| U-0 | R/W-0 | R-0 | R-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | ADIF | RCIF | TXIF | SSPIF | CCP1IF | 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 **Unimplemented:** Read as '0'

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

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

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

bit 3 **SSPIF:** Master Synchronous Serial Port (MSSP) Interrupt Flag bit
1 = The MSSP interrupt condition has occurred, and must be cleared in software before returning from the Interrupt Service Routine. The conditions that will set this bit are:
SPI
A transmission/reception has taken place
$I^2C$ Slave/Master
A transmission/reception has taken place
$I^2C$ Master
The initiated Start condition was completed by the MSSP module
The initiated Stop condition was completed by the MSSP module
The initiated restart condition was completed by the MSSP module
The initiated Acknowledge condition was completed by the MSSP module
A Start condition occurred while the MSSP module was idle (Multi-master system)
A Stop condition occurred while the MSSP module was idle (Multi-master system)
0 = No MSSP interrupt condition has occurred

bit 2 **CCP1IF:** CCP1 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

bit 1 **TMR2IF:** Timer2 to PR2 Interrupt Flag bit
1 = A Timer2 to PR2 match occurred (must be cleared in software)
0 = No Timer2 to PR2 match occurred

bit 0 **TMR1IF:** Timer1 Overflow Interrupt Flag bit
1 = The TMR1 register overflowed (must be cleared in software)
0 = The TMR1 register did not overflow


© 2006-2012 Microchip Technology Inc.

### 2.2.2.7 PIR2 Register

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

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

## REGISTER DEFINITIONS: PIR2

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

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | U-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| OSFIF | C2IF | C1IF | EEIF | BCLIF | ULPWUIF | — | CCP2IF |
| 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 **OSFIF:** Oscillator Fail Interrupt Flag bit
1 = System oscillator failed, clock input has changed to INTOSC (must be cleared in software)
0 = System clock operating

bit 6 **C2IF:** Comparator C2 Interrupt Flag bit
1 = Comparator output (C2OUT bit) has changed (must be cleared in software)
0 = Comparator output (C2OUT bit) has not changed

bit 5 **C1IF:** Comparator C1 Interrupt Flag bit
1 = Comparator output (C1OUT bit) has changed (must be cleared in software)
0 = Comparator output (C1OUT bit) has not changed

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

bit 3 **BCLIF:** Bus Collision Interrupt Flag bit
1 = A bus collision has occurred in the MSSP when configured for $I^2C$ Master mode
0 = No bus collision has occurred

bit 2 **ULPWUIF:** Ultra Low-Power Wake-up Interrupt Flag bit
1 = Wake-up condition has occurred (must be cleared in software)
0 = No Wake-up condition has occurred

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

bit 0 **CCP2IF:** CCP2 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

### 2.2.2.8 PCON Register

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

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

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

## REGISTER DEFINITIONS: PCON

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

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

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

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

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

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

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

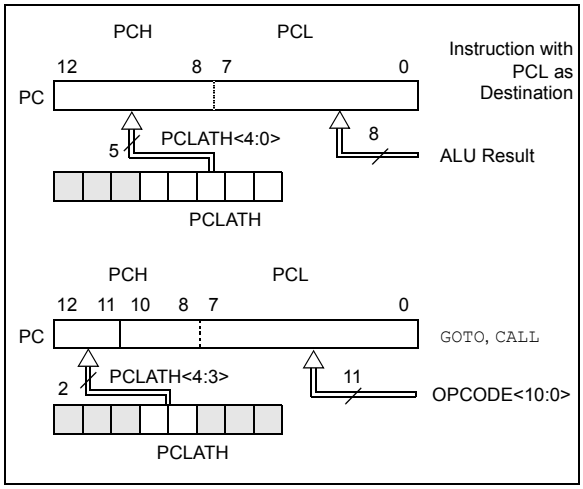
**Note 1:** BOREN<1:0> = 01 in the Configuration Word Register 1 for this bit to control the $\overline{BOR}$.


© 2006-2012 Microchip Technology Inc.

## 2.3 PCL and PCLATH

The Program Counter (PC) is 13 bits wide. The low byte comes from the PCL register, which is a readable and writable register. The high byte (PC<12:8>) is not directly readable or writable and comes from PCLATH. On any Reset, the PC is cleared. Figure 2-7 shows the two situations for the loading of the PC. The upper example in Figure 2-7 shows how the PC is loaded on a write to PCL (PCLATH<4:0> → PCH). The lower example in Figure 2-7 shows how the PC is loaded during a CALL or GOTO instruction (PCLATH<4:3> → PCH).

**FIGURE 2-7: LOADING OF PC IN DIFFERENT SITUATIONS**




### 2.3.1 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 5 bits to the PCLATH register. When the lower 8 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 8 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).

### 2.3.2 STACK

The PIC16F882/883/884/886/887 devices have an 8-level x 13-bit wide hardware stack (see Figures 2-2 and 2-3). The stack space is not part of either program or data space and the Stack Pointer is not readable or writable. The PC is PUSHed onto the stack when a CALL instruction is executed or an interrupt causes a branch. The stack is POPed in the event of a RETURN, RETLW or a RETFIE instruction execution. PCLATH is not affected by a PUSH or POP operation.

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

**Note 1:** There are no Status bits to indicate stack overflow or stack underflow conditions.

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

## 2.4 Indirect Addressing, INDF and FSR Registers

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

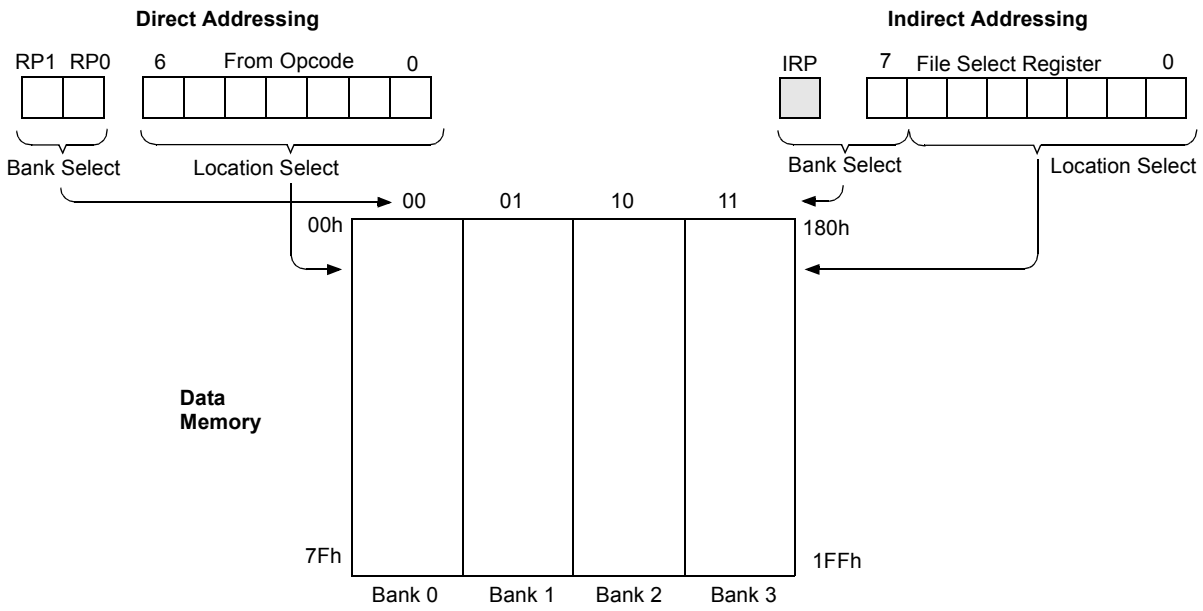
Indirect addressing is possible by using the INDF register. Any instruction using the INDF register actually accesses data pointed to by the File Select Register (FSR). Reading INDF itself indirectly will produce 00h. Writing to the INDF register indirectly results in a no operation (although Status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

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

**EXAMPLE 2-1: INDIRECT ADDRESSING**

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

**FIGURE 2-8: DIRECT/INDIRECT ADDRESSING PIC16F882/883/884/886/887**




**Note:** For memory map detail, see Figures 2-2 and 2-3.


© 2006-2012 Microchip Technology Inc.

## 3.0 I/O PORTS

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

### 3.1 PORTA and the TRISA Registers

PORTA is a 8-bit wide, bidirectional port. The corresponding data direction register is TRISA (Register 3-2). 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). Example 3-1 shows how to initialize PORTA.

Reading the PORTA register (Register 3-1) reads the status of the pins, whereas writing to it will write to the PORT latch. All write operations are read-modify-write operations. Therefore, a write to a port implies that the port pins are read, this value is modified and then written to the PORT data latch.

The TRISA register (Register 3-2) controls the PORTA pin output drivers, even when they are being used as analog inputs. The user should ensure the bits in the TRISA register are maintained set when using them as analog inputs. I/O pins configured as analog 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'. |
|---|---|

**EXAMPLE 3-1: INITIALIZING PORTA**

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

**REGISTER 3-1: PORTA: PORTA REGISTER**

| R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x |
|---|---|---|---|---|---|---|---|
| RA7 | RA6 | RA5 | RA4 | RA3 | RA2 | RA1 | RA0 |
| 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 **RA<7:0>**: PORTA I/O Pin bit
1 = Port pin is > $V_{IH}$
0 = Port pin is < $V_{IL}$

**REGISTER 3-2: TRISA: PORTA TRI-STATE REGISTER**

| R/W-1(1) | R/W-1(1) | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 |
|---|---|---|---|---|---|---|---|
| TRISA7 | TRISA6 | TRISA5 | TRISA4 | TRISA3 | TRISA2 | TRISA1 | TRISA0 |
| 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 **TRISA<7:0>:** PORTA Tri-State Control bit
1 = PORTA pin configured as an input (tri-stated)
0 = PORTA pin configured as an output

**Note 1:** TRISA<7:6> always reads '1' in XT, HS and LP Oscillator modes.

## 3.2 Additional Pin Functions

RA0 also has an Ultra Low-Power Wake-up option. The next three sections describe these functions.

### 3.2.1 ANSEL REGISTER

The ANSEL register (Register 3-3) is used to configure the Input mode of an I/O pin to analog. Setting the appropriate ANSEL 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 ANSEL bits has no affect 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.

**REGISTER 3-3: ANSEL: ANALOG SELECT 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 |
|---|---|---|---|---|---|---|---|
| ANS7(2) | ANS6(2) | ANS5(2) | ANS4 | ANS3 | ANS2 | ANS1 | ANS0 |
| 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 **ANS<7:0>**: Analog Select bits
Analog select between analog or digital function on pins AN<7:0>, respectively.
1 = Analog input. Pin is assigned as analog input(1).
0 = Digital I/O. Pin is assigned to port or special function.

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

**2:** Not implemented on PIC16F883/886.


© 2006-2012 Microchip Technology Inc.

### 3.2.2 ULTRA LOW-POWER WAKE-UP

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

Follow these steps to use this feature:

a) Charge the capacitor on RA0 by configuring the RA0 pin to output (= 1).
b) Configure RA0 as an input.
c) Set the ULPWUIE bit of the PIE2 register to enable interrupt.
d) Set the ULPWUE bit of the PCON register to begin the capacitor discharge.
e) Execute a SLEEP instruction.

When the voltage on RA0 drops below $V_{IL}$, an interrupt will be generated which will cause the device to wake-up and execute the next instruction. If the GIE bit of the INTCON register is set, the device will then call the interrupt vector (0004h).

This feature provides a low-power technique for periodically waking up the device from Sleep. The time-out is dependent on the discharge time of the RC circuit on RA0. See Example 3-2 for initializing the Ultra Low-Power Wake-up module.

A series resistor between RA0 and the external capacitor provides overcurrent protection for the RA0/AN0/ULPWU/C12IN0- pin and can allow for software calibration of the time-out (see Figure 3-1). A timer can be used to measure the charge time and discharge time of the capacitor. The charge time can then be adjusted to provide the desired interrupt delay. This technique will compensate for the affects of temperature, voltage and component accuracy. The Ultra Low-Power Wake-up peripheral can also be configured as a simple Programmable Low Voltage Detect or temperature sensor.

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

**EXAMPLE 3-2: ULTRA LOW-POWER WAKE-UP INITIALIZATION**

```
BANKSEL PORTA           ;
BSF     PORTA,0         ;Set RA0 data latch
BANKSEL ANSEL           ;
BCF     ANSEL,0         ;RA0 to digital I/O
BANKSEL TRISA           ;
BCF     TRISA,0         ;Output high to
CALL    CapDelay        ;charge capacitor
BANKSEL PIR2            ;
BCF     PIR2,ULPWUIF    ;Clear flag
BANKSEL PCON
BSF     PCON,ULPWUE     ;Enable ULP Wake-up
BSF     TRISA,0         ;RA0 to input
BSF     PIE2, ULPWUIE   ;Enable interrupt
MOVLW   B'11000000'     ;Enable peripheral
MOVWF   INTCON          ;interrupt
SLEEP                   ;Wait for IOC
NOP                     ;
```

### 3.2.3 PIN DESCRIPTIONS AND DIAGRAMS

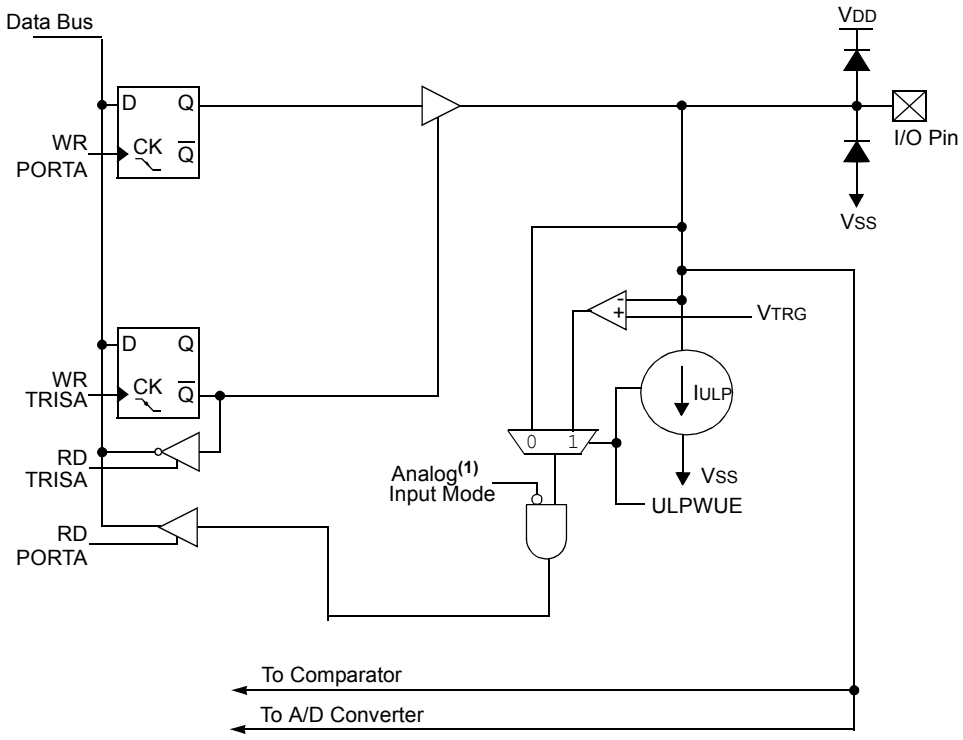
Each PORTA pin is multiplexed with other functions. The pins and their combined functions are briefly described here. For specific information about individual functions such as the comparator or the A/D Converter (ADC), refer to the appropriate section in this data sheet.

#### 3.2.3.1 RA0/AN0/ULPWU/C12IN0-

Figure 3-1 shows the diagram for this pin. This pin is configurable to function as one of the following:

- a general purpose I/O
- an analog input for the ADC
- a negative analog input to Comparator C1 or C2
- an analog input for the Ultra Low-Power Wake-up

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




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

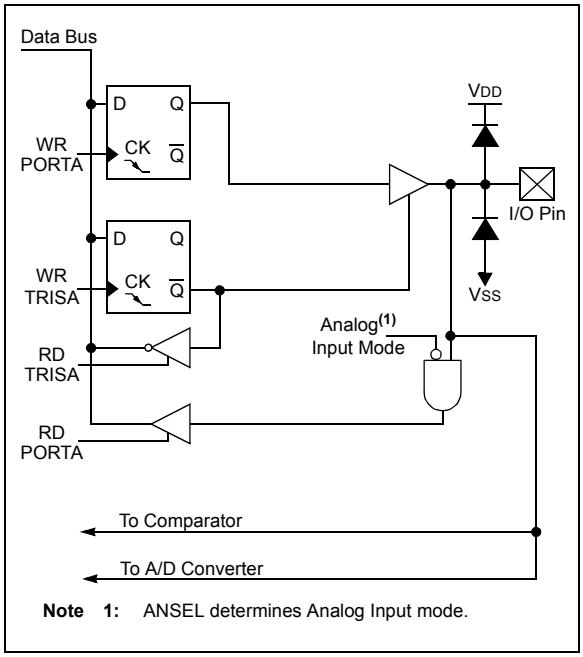

© 2006-2012 Microchip Technology Inc.

### 3.2.3.2 RA1/AN1/C12IN1-

Figure 3-2 shows the diagram for this pin. This pin is configurable to function as one of the following:

- a general purpose I/O
- an analog input for the ADC
- a negative analog input to Comparator C1 or C2

**FIGURE 3-2: BLOCK DIAGRAM OF RA1**



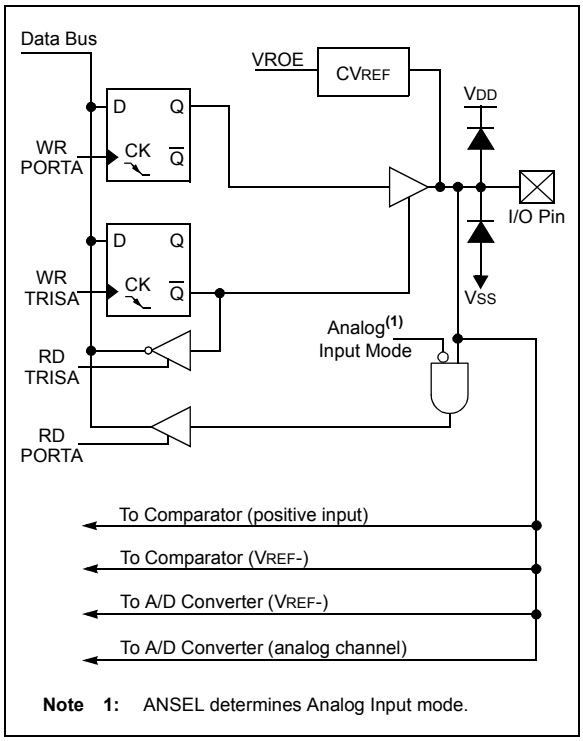

**Note 1:** ANSEL determines Analog Input mode.

### 3.2.3.3 RA2/AN2/VREF-/CVREF/C2IN+

Figure 3-3 shows the diagram for this pin. This pin is configurable to function as one of the following:

- a general purpose I/O
- an analog input for the ADC
- a negative voltage reference input for the ADC and CVREF
- a comparator voltage reference output
- a positive analog input to Comparator C2

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



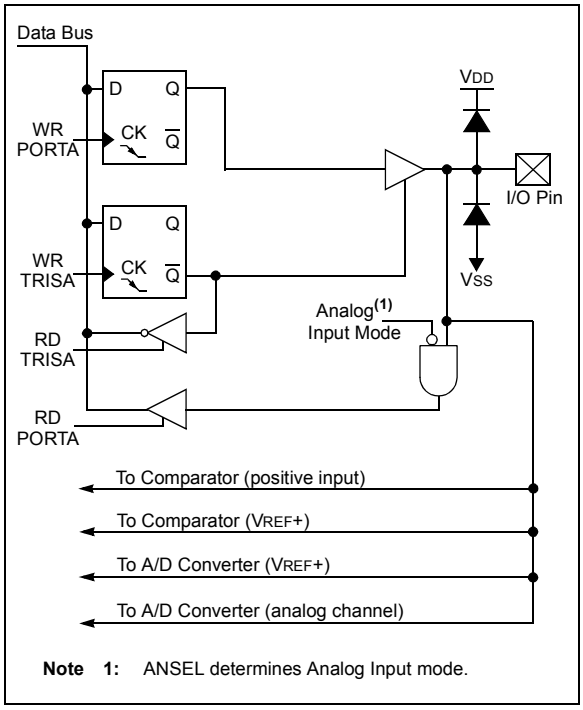

**Note 1:** ANSEL determines Analog Input mode.

### 3.2.3.4 RA3/AN3/VREF+/C1IN+

Figure 3-4 shows the diagram for this pin. This pin is configurable to function as one of the following:

- a general purpose input
- an analog input for the ADC
- a positive voltage reference input for the ADC and CVREF
- a positive analog input to Comparator C1

**FIGURE 3-4: BLOCK DIAGRAM OF RA3**



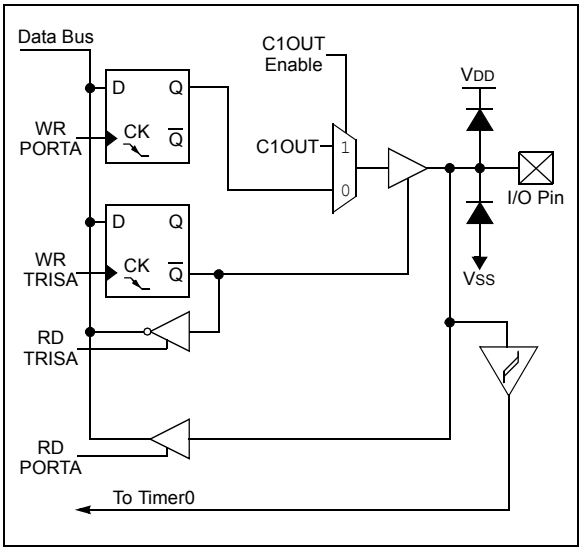

**Note 1:** ANSEL determines Analog Input mode.

### 3.2.3.5 RA4/T0CKI/C1OUT

Figure 3-5 shows the diagram for this pin. This pin is configurable to function as one of the following:

- a general purpose I/O
- a clock input for Timer0
- a digital output from Comparator C1

**FIGURE 3-5: BLOCK DIAGRAM OF RA4**





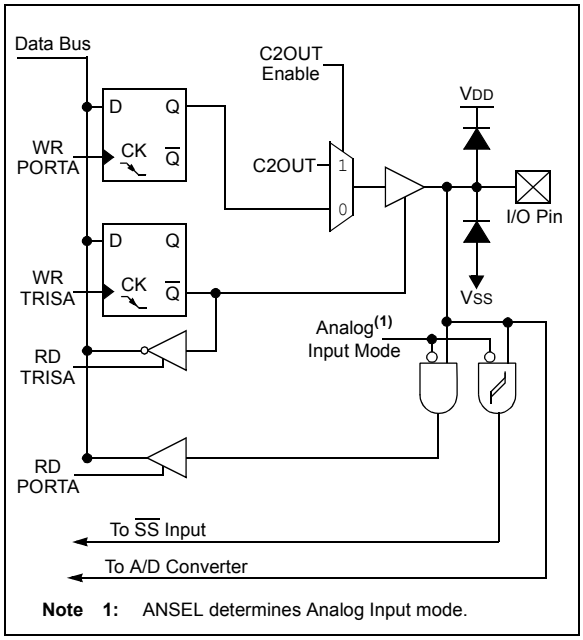
© 2006-2012 Microchip Technology Inc.

### 3.2.3.6 RA5/AN4/$\overline{SS}$/C2OUT

Figure 3-6 shows the diagram for this pin. This pin is configurable to function as one of the following:

- a general purpose I/O
- an analog input for the ADC
- a slave select input
- a digital output from Comparator C2

**FIGURE 3-6: BLOCK DIAGRAM OF RA5**



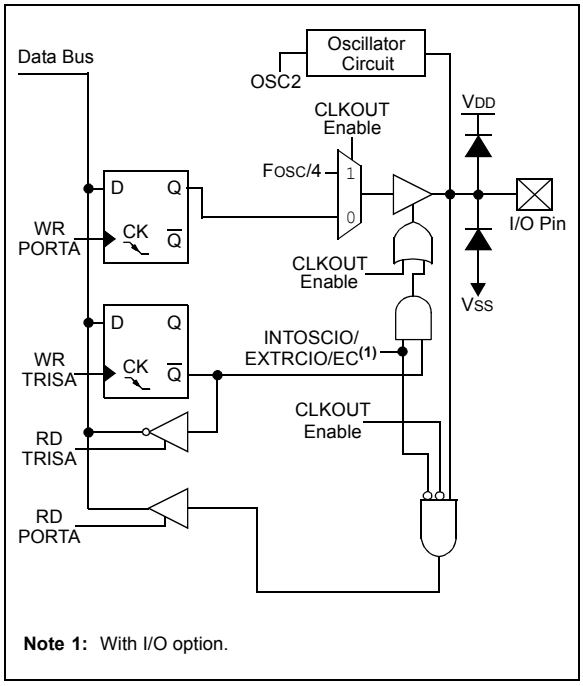

**Note 1:** ANSEL determines Analog Input mode.

### 3.2.3.7 RA6/OSC2/CLKOUT

Figure 3-7 shows the diagram for this pin. This pin is configurable to function as one of the following:

- a general purpose I/O
- a crystal/resonator connection
- a clock output

**FIGURE 3-7: BLOCK DIAGRAM OF RA6**




**Note 1:** With I/O option.

### 3.2.3.8 RA7/OSC1/CLKIN

Figure 3-8 shows the diagram for this pin. This pin is configurable to function as one of the following:

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

**FIGURE 3-8: BLOCK DIAGRAM OF RA7**

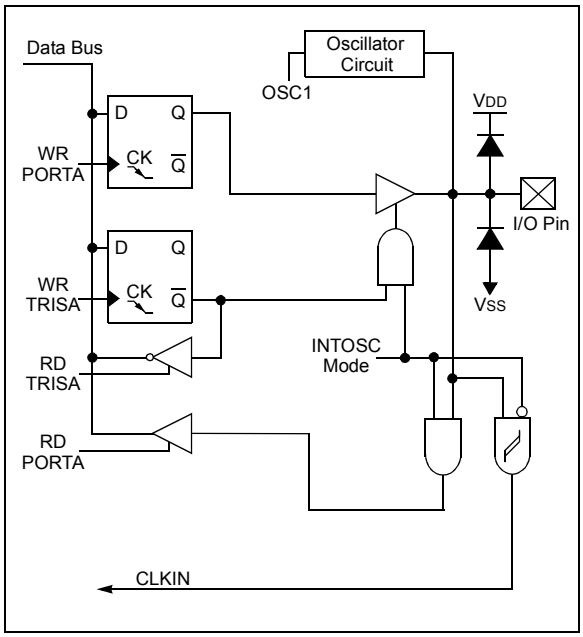



**TABLE 3-1: SUMMARY OF REGISTERS ASSOCIATED WITH PORTA**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|
| ADCON0 | ADCS1 | ADCS0 | CHS3 | CHS2 | CHS1 | CHS0 | GO/$\overline{\text{DONE}}$ | ADON | 108 |
| ANSEL | ANS7 | ANS6 | ANS5 | ANS4 | ANS3 | ANS2 | ANS1 | ANS0 | 42 |
| CM1CON0 | C1ON | C1OUT | C1OE | C1POL | — | C1R | C1CH1 | C1CH0 | 93 |
| CM2CON0 | C2ON | C2OUT | C2OE | C2POL | — | C2R | C2CH1 | C2CH0 | 94 |
| CM2CON1 | MC1OUT | MC2OUT | C1RSEL | C2RSEL | — | — | T1GSS | C2SYNC | 96 |
| PCON | — | — | ULPWUE | SBOREN | — | — | $\overline{\text{POR}}$ | $\overline{\text{BOR}}$ | 38 |
| OPTION_REG | $\overline{\text{RBPU}}$ | INTEDG | T0CS | T0SE | PSA | PS2 | PS1 | PS0 | 32 |
| PORTA | RA7 | RA6 | RA5 | RA4 | RA3 | RA2 | RA1 | RA0 | 41 |
| SSPCON | WCOL | SSPOV | SSPEN | CKP | SSPM3 | SSPM2 | SSPM1 | SSPM0 | 185 |
| TRISA | TRISA7 | TRISA6 | TRISA5 | TRISA4 | TRISA3 | TRISA2 | TRISA1 | TRISA0 | 41 |

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


© 2006-2012 Microchip Technology Inc.

## 3.3 PORTB and TRISB Registers

PORTB is an 8-bit wide, bidirectional port. The corresponding data direction register is TRISB (Register 3-6). Setting a TRISB bit (= 1) will make the corresponding PORTB pin an input (i.e., put the corresponding output driver in a High-Impedance mode). Clearing a TRISB bit (= 0) will make the corresponding PORTB pin an output (i.e., enable the output driver and put the contents of the output latch on the selected pin). Example 3-3 shows how to initialize PORTB.

Reading the PORTB register (Register 3-5) reads the status of the pins, whereas writing to it will write to the PORT latch. All write operations are read-modify-write operations. Therefore, a write to a port implies that the port pins are read, this value is modified and then written to the PORT data latch.

The TRISB register (Register 3-6) controls the PORTB pin output drivers, even when they are being used as analog inputs. The user should ensure the bits in the TRISB register are maintained set when using them as analog inputs. I/O pins configured as analog input always read '0'. Example 3-3 shows how to initialize PORTB.

**EXAMPLE 3-3: INITIALIZING PORTB**

```
BANKSEL PORTB       ;
CLRF    PORTB       ;Init PORTB
BANKSEL TRISB       ;
MOVLW   B'11110000' ;Set RB<7:4> as inputs
                    ;and RB<3:0> as outputs
MOVWF   TRISB       ;
```

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

## 3.4 Additional PORTB Pin Functions

PORTB pins RB<7:0> on the device family device have an interrupt-on-change option and a weak pull-up option. The following three sections describe these PORTB pin functions.

Every PORTB pin on this device family has an interrupt-on-change option and a weak pull-up option.

### 3.4.1 ANSELH REGISTER

The ANSELH register (Register 3-4) is used to configure the Input mode of an I/O pin to analog. Setting the appropriate ANSELH 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 ANSELH bits has no affect on digital output functions. A pin with TRIS clear and ANSELH 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.

### 3.4.2 WEAK PULL-UPS

Each of the PORTB pins has an individually configurable internal weak pull-up. Control bits WPUB<7:0> enable or disable each pull-up (see Register 3-7). Each weak pull-up is automatically turned off when the port pin is configured as an output. All pull-ups are disabled on a Power-on Reset by the $\overline{\text{RBPU}}$ bit of the OPTION register.

### 3.4.3 INTERRUPT-ON-CHANGE

All of the PORTB pins are individually configurable as an interrupt-on-change pin. Control bits IOCB<7:0> enable or disable the interrupt function for each pin. Refer to Register 3-8. The interrupt-on-change feature is disabled on a Power-on Reset.

For enabled interrupt-on-change pins, the present value is compared with the old value latched on the last read of PORTB to determine which bits have changed or mismatched the old value. The 'mismatch' outputs of the last read are OR'd together to set the PORTB Change Interrupt flag bit (RBIF) in the INTCON register.

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

a) Any read or write of PORTB. This will end the mismatch condition.
b) Clear the flag bit RBIF.

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

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

**REGISTER 3-4: ANSELH: ANALOG SELECT HIGH REGISTER**

| U-0 | U-0 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 |
|---|---|---|---|---|---|---|---|
| — | — | ANS13 | ANS12 | ANS11 | ANS10 | ANS9 | ANS8 |
| 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 **ANS<13:8>**: Analog Select bits
Analog select between analog or digital function on pins AN<13:8>, respectively.
1 = Analog input. Pin is assigned as analog input(1).
0 = Digital I/O. Pin is assigned to port or special function.

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

**REGISTER 3-5: PORTB: PORTB REGISTER**

| R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x |
|---|---|---|---|---|---|---|---|
| RB7 | RB6 | RB5 | RB4 | RB3 | RB2 | RB1 | RB0 |
| 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 **RB<7:0>**: PORTB I/O Pin bit
1 = Port pin is > $V_{IH}$
0 = Port pin is < $V_{IL}$

**REGISTER 3-6: TRISB: PORTB TRI-STATE 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 |
|---|---|---|---|---|---|---|---|
| TRISB7 | TRISB6 | TRISB5 | TRISB4 | TRISB3 | TRISB2 | TRISB1 | TRISB0 |
| 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 **TRISB<7:0>:** PORTB Tri-State Control bit
1 = PORTB pin configured as an input (tri-stated)
0 = PORTB pin configured as an output


© 2006-2012 Microchip Technology Inc.

**REGISTER 3-7: WPUB: WEAK PULL-UP PORTB 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 |
|---|---|---|---|---|---|---|---|
| WPUB7 | WPUB6 | WPUB5 | WPUB4 | WPUB3 | WPUB2 | WPUB1 | WPUB0 |
| 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 **WPUB<7:0>**: Weak Pull-up Register bit
1 = Pull-up enabled
0 = Pull-up disabled

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

**REGISTER 3-8: IOCB: INTERRUPT-ON-CHANGE PORTB 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 |
|---|---|---|---|---|---|---|---|
| IOCB7 | IOCB6 | IOCB5 | IOCB4 | IOCB3 | IOCB2 | IOCB1 | IOCB0 |
| 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 **IOCB<7:0>:** Interrupt-on-Change PORTB Control bit
1 = Interrupt-on-change enabled
0 = Interrupt-on-change disabled

### 3.4.4 PIN DESCRIPTIONS AND DIAGRAMS

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

#### 3.4.4.1 RB0/AN12/INT

Figure 3-9 shows the diagram for this pin. This pin is configurable to function as one of the following:

- a general purpose I/O
- an analog input for the ADC
- an external edge triggered interrupt

#### 3.4.4.2 RB1/AN10/P1C(1)/C12IN3-

Figure 3-9 shows the diagram for this pin. This pin is configurable to function as one of the following:

- a general purpose I/O
- an analog input for the ADC
- a PWM output(1)
- an analog input to Comparator C1 or C2

| | |
|---|---|
| **Note 1:** | P1C is available on PIC16F882/883/886 only. |

#### 3.4.4.3 RB2/AN8/P1B(1)

Figure 3-9 shows the diagram for this pin. This pin is configurable to function as one of the following:

- a general purpose I/O
- an analog input for the ADC
- a PWM output(1)

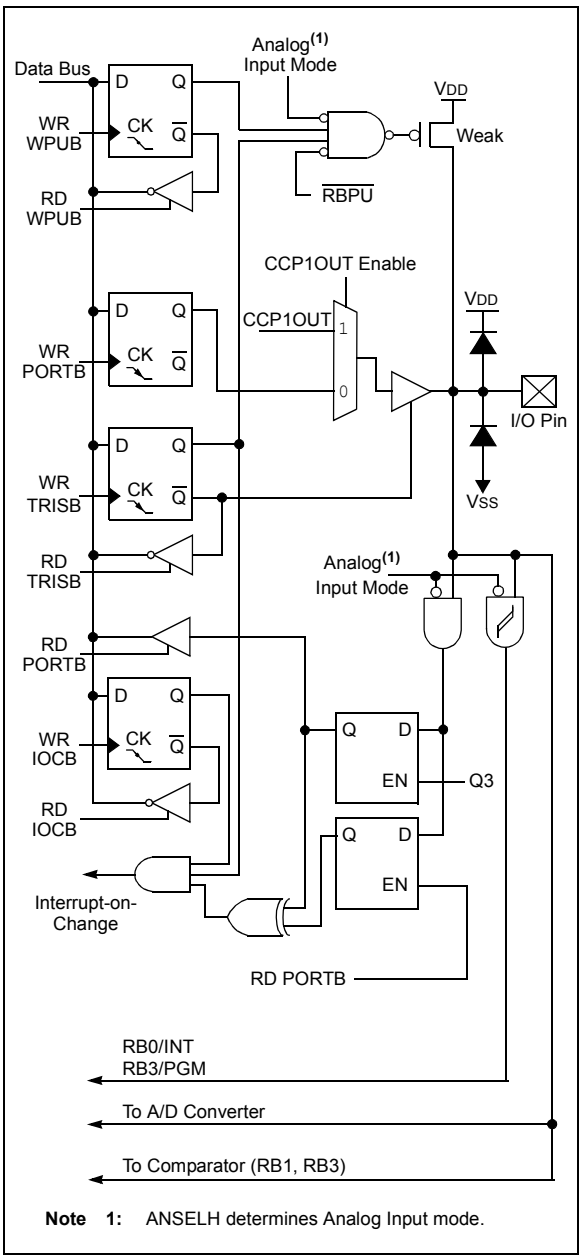
| | |
|---|---|
| **Note 1:** | P1B is available on PIC16F882/883/886 only. |

#### 3.4.4.4 RB3/AN9/PGM/C12IN2-

Figure 3-9 shows the diagram for this pin. This pin is configurable to function as one of the following:

- a general purpose I/O
- an analog input for the ADC
- Low-voltage In-Circuit Serial Programming enable pin
- an analog input to Comparator C1 or C2

**FIGURE 3-9: BLOCK DIAGRAM OF RB<3:0>**





© 2006-2012 Microchip Technology Inc.

#### 3.4.4.5 RB4/AN11/P1D(1)

Figure 3-10 shows the diagram for this pin. This pin is configurable to function as one of the following:

- a general purpose I/O
- an analog input for the ADC
- a PWM output(1)

**Note 1:** P1D is available on PIC16F882/883/886 only.

#### 3.4.4.6 RB5/AN13/$\overline{T1G}$

Figure 3-10 shows the diagram for this pin. This pin is configurable to function as one of the following:

- a general purpose I/O
- an analog input for the ADC
- a Timer1 gate input

#### 3.4.4.7 RB6/ICSPCLK

Figure 3-10 shows the diagram for this pin. This pin is configurable to function as one of the following:

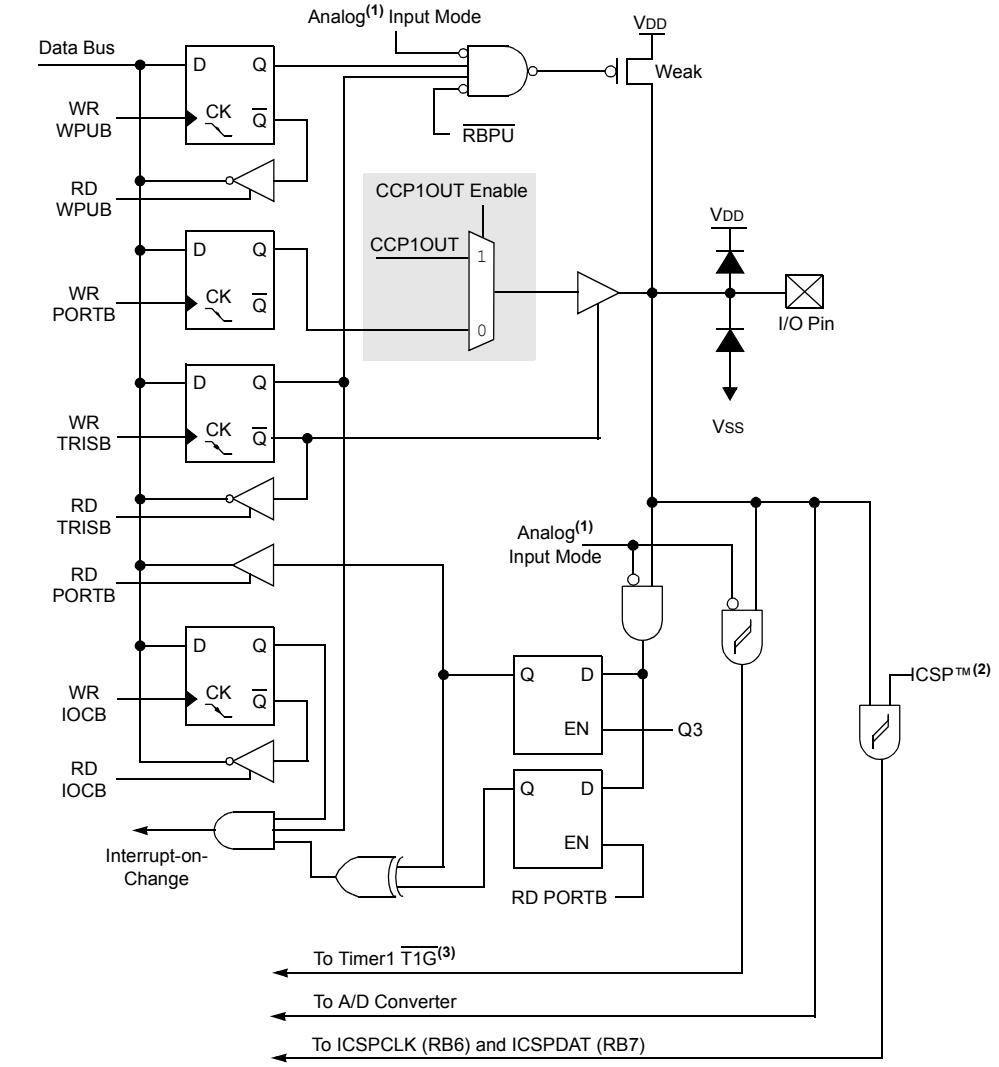
- a general purpose I/O
- In-Circuit Serial Programming clock

#### 3.4.4.8 RB7/ICSPDAT

Figure 3-10 shows the diagram for this pin. This pin is configurable to function as one of the following:

- a general purpose I/O
- In-Circuit Serial Programming data

**FIGURE 3-10: BLOCK DIAGRAM OF RB<7:4>**




**Note 1:** ANSELH determines Analog Input mode.
**2:** Applies to RB<7:6> pins only).
**3:** Applies to RB5 pin only.

**TABLE 3-2: SUMMARY OF REGISTERS ASSOCIATED WITH PORTB**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|
| ANSELH | — | — | ANS13 | ANS12 | ANS11 | ANS10 | ANS9 | ANS8 | 50 |
| CCP1CON | P1M1 | P1M0 | DC1B1 | DC1B0 | CCP1M3 | CCP1M2 | CCP1M1 | CCP1M0 | 128 |
| CM2CON1 | MC1OUT | MC2OUT | C1RSEL | C2RSEL | — | — | T1GSS | C2SYNC | 96 |
| IOCB | IOCB7 | IOCB6 | IOCB5 | IOCB4 | IOCB3 | IOCB2 | IOCB1 | IOCB0 | 51 |
| INTCON | GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | RBIF | 33 |
| OPTION_REG | $\overline{\text{RBPU}}$ | INTEDG | T0CS | T0SE | PSA | PS2 | PS1 | PS0 | 32 |
| PORTB | RB7 | RB6 | RB5 | RB4 | RB3 | RB2 | RB1 | RB0 | 50 |
| TRISB | TRISB7 | TRISB6 | TRISB5 | TRISB4 | TRISB3 | TRISB2 | TRISB1 | TRISB0 | 50 |
| WPUB | WPUB7 | WPUB6 | WPUB5 | WPUB4 | WPUB3 | WPUB2 | WPUB1 | WPUB0 | 51 |

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


© 2006-2012 Microchip Technology Inc.

## 3.5 PORTC and TRISC Registers

PORTC is a 8-bit wide, bidirectional port. The corresponding data direction register is TRISC (Register 3-10). Setting a TRISC bit (= 1) will make the corresponding PORTC pin an input (i.e., put the corresponding output driver in a High-Impedance mode). Clearing a TRISC bit (= 0) will make the corresponding PORTC pin an output (i.e., enable the output driver and put the contents of the output latch on the selected pin). Example 3-4 shows how to initialize PORTC.

Reading the PORTC register (Register 3-9) reads the status of the pins, whereas writing to it will write to the PORT latch. All write operations are read-modify-write operations. Therefore, a write to a port implies that the port pins are read, this value is modified and then written to the PORT data latch.

The TRISC register (Register 3-10) controls the PORTC pin output drivers, even when they are being used as analog inputs. The user should ensure the bits in the TRISC register are maintained set when using them as analog inputs. I/O pins configured as analog input always read '0'.

**EXAMPLE 3-4: INITIALIZING PORTC**

```
BANKSEL PORTC          ;
CLRF    PORTC          ;Init PORTC
BANKSEL TRISC          ;
MOVLW   B'00001100'    ;Set RC<3:2> as inputs
MOVWF   TRISC          ;and set RC<7:4,1:0>
                       ;as outputs
```

**REGISTER 3-9: PORTC: PORTC REGISTER**

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

| **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 **RC<7:0>**: PORTC General Purpose I/O Pin bit
1 = Port pin is > VIH
0 = Port pin is < VIL

**REGISTER 3-10: TRISC: PORTC TRI-STATE REGISTER**

| R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1(1) | R/W-1(1) |
|---|---|---|---|---|---|---|---|
| TRISC7 | TRISC6 | TRISC5 | TRISC4 | TRISC3 | TRISC2 | TRISC1 | TRISC0 |
| 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 **TRISC<7:0>:** PORTC Tri-State Control bit
1 = PORTC pin configured as an input (tri-stated)
0 = PORTC pin configured as an output

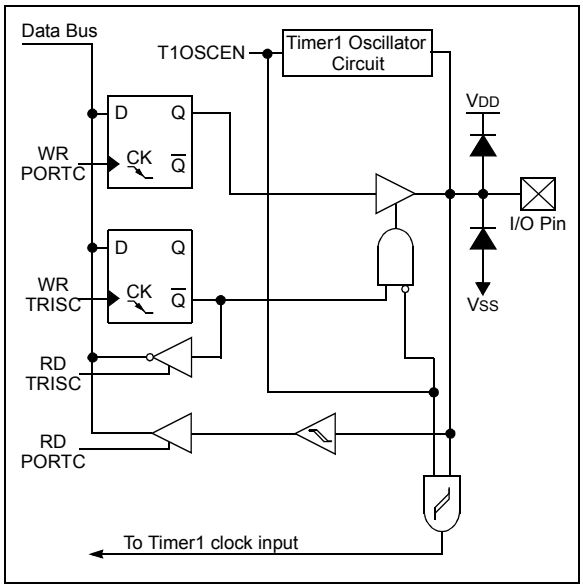
**Note 1:** TRISC<1:0> always reads '1' in LP Oscillator mode.

### 3.5.1 RC0/T1OSO/T1CKI

Figure 3-11 shows the diagram for this pin. This pin is configurable to function as one of the following:

- a general purpose I/O
- a Timer1 oscillator output
- a Timer1 clock input

**FIGURE 3-11: BLOCK DIAGRAM OF RC0**

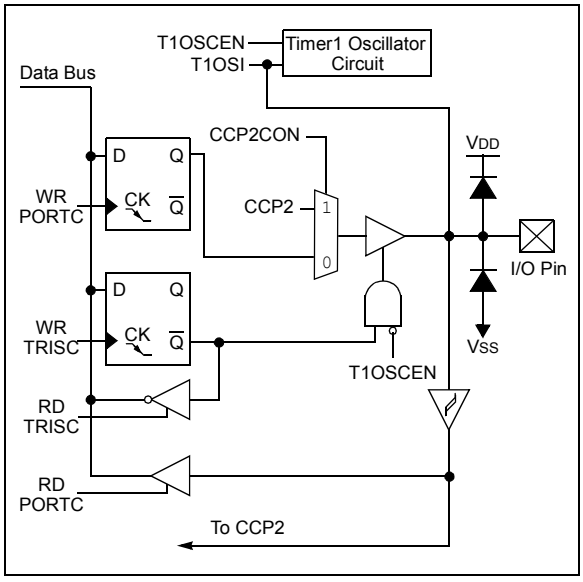



### 3.5.2 RC1/T1OSI/CCP2

Figure 3-12 shows the diagram for this pin. This pin is configurable to function as one of the following:

- a general purpose I/O
- a Timer1 oscillator input
- a Capture input and Compare/PWM output for Comparator C2

**FIGURE 3-12: BLOCK DIAGRAM OF RC1**

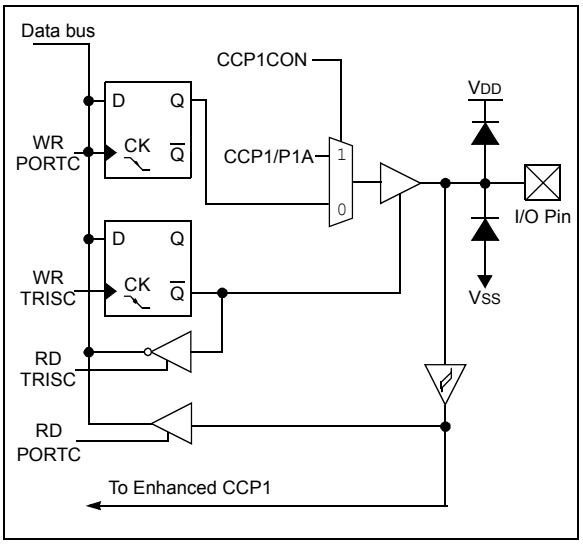



### 3.5.3 RC2/P1A/CCP1

Figure 3-13 shows the diagram for this pin. This pin is configurable to function as one of the following:

- a general purpose I/O
- a PWM output
- a Capture input and Compare output for Comparator C1

**FIGURE 3-13: BLOCK DIAGRAM OF RC2**





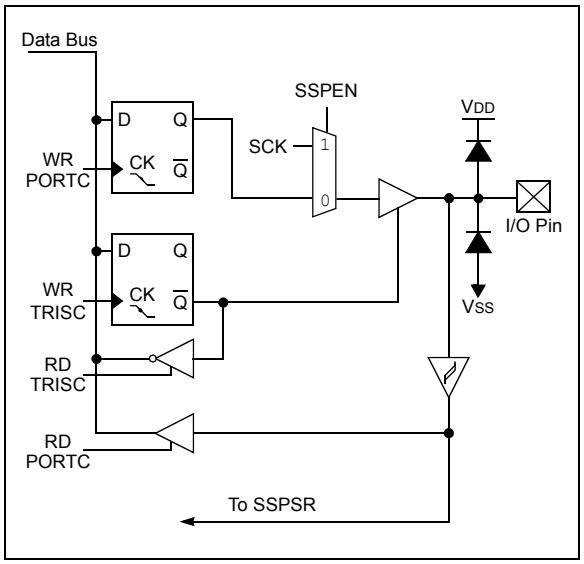
© 2006-2012 Microchip Technology Inc.

### 3.5.4 RC3/SCK/SCL

Figure 3-14 shows the diagram for this pin. This pin is configurable to function as one of the following:

- a general purpose I/O
- a SPI clock
- an $I^2C$™ clock

**FIGURE 3-14: BLOCK DIAGRAM OF RC3**

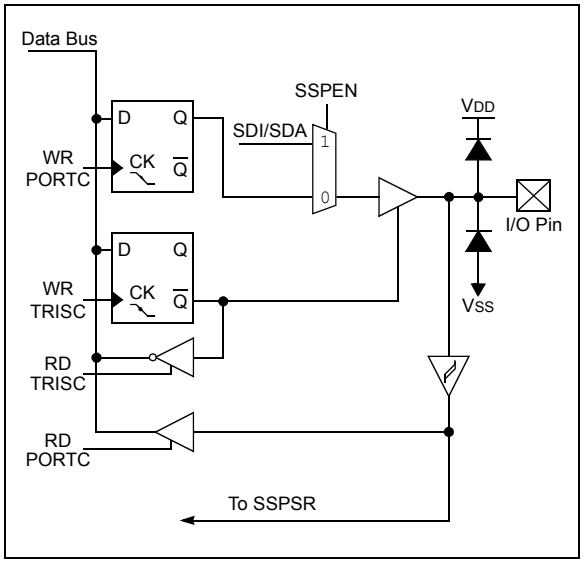



### 3.5.5 RC4/SDI/SDA

Figure 3-15 shows the diagram for this pin. This pin is configurable to function as one of the following:

- a general purpose I/O
- a SPI data I/O
- an $I^2C$ data I/O

**FIGURE 3-15: BLOCK DIAGRAM OF RC4**

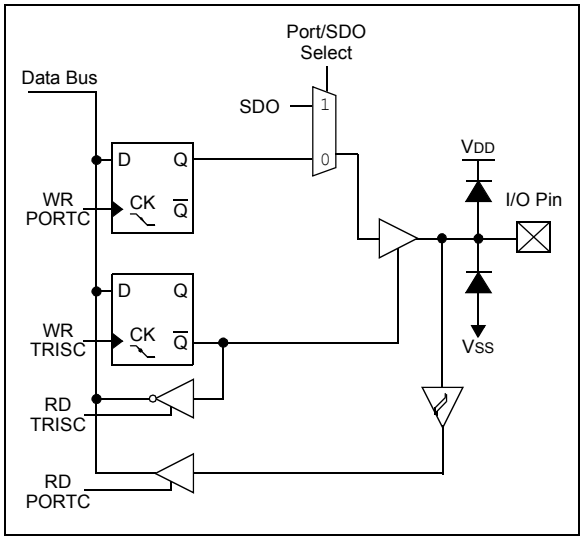



### 3.5.6 RC5/SDO

Figure 3-16 shows the diagram for this pin. This pin is configurable to function as one of the following:

- a general purpose I/O
- a serial data output

**FIGURE 3-16: BLOCK DIAGRAM OF RC5**

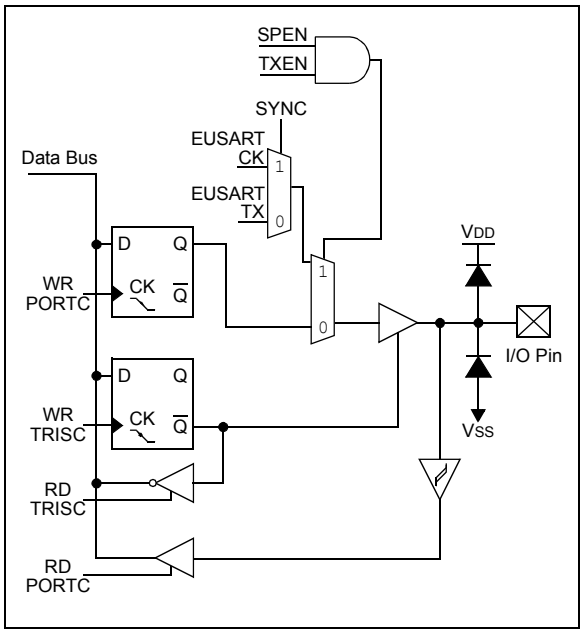


### 3.5.7 RC6/TX/CK

Figure 3-17 shows the diagram for this pin. This pin is configurable to function as one of the following:

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

**FIGURE 3-17: BLOCK DIAGRAM OF RC6**




### 3.5.8 RC7/RX/DT

Figure 3-18 shows the diagram for this pin. This pin is configurable to function as one of the following:

- a general purpose I/O
- an asynchronous serial input
- a synchronous serial data I/O

**FIGURE 3-18: BLOCK DIAGRAM OF RC7**

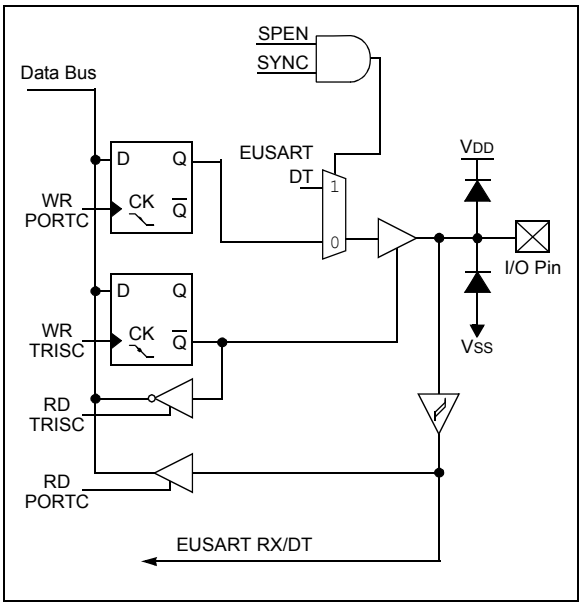



**TABLE 3-3: SUMMARY OF REGISTERS ASSOCIATED WITH PORTC**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|
| CCP1CON | P1M1 | P1M0 | DC1B1 | DC1B0 | CCP1M3 | CCP1M2 | CCP1M1 | CCP1M0 | 128 |
| CCP2CON | — | — | DC2B1 | DC2B0 | CCP2M3 | CCP2M2 | CCP2M1 | CCP2M0 | 129 |
| PORTC | RC7 | RC6 | RC5 | RC4 | RC3 | RC2 | RC1 | RC0 | 55 |
| PSTRCON | — | — | — | STRSYNC | STRD | STRC | STRB | STRA | 150 |
| RCSTA | SPEN | RX9 | SREN | CREN | ADDEN | FERR | OERR | RX9D | 165 |
| SSPCON | WCOL | SSPOV | SSPEN | CKP | SSPM3 | SSPM2 | SSPM1 | SSPM0 | 185 |
| T1CON | T1GINV | TMR1GE | T1CKPS1 | T1CKPS0 | T1OSCEN | $\overline{\text{T1SYNC}}$ | TMR1CS | TMR1ON | 84 |
| TRISC | TRISC7 | TRISC6 | TRISC5 | TRISC4 | TRISC3 | TRISC2 | TRISC1 | TRISC0 | 55 |

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


© 2006-2012 Microchip Technology Inc.

## 3.6 PORTD and TRISD Registers

PORTD(1) is a 8-bit wide, bidirectional port. The corresponding data direction register is TRISD (Register 3-12). Setting a TRISD bit (= 1) will make the corresponding PORTD pin an input (i.e., put the corresponding output driver in a High-Impedance mode). Clearing a TRISD bit (= 0) will make the corresponding PORTD pin an output (i.e., enable the output driver and put the contents of the output latch on the selected pin). Example 3-5 shows how to initialize PORTD.

Reading the PORTD register (Register 3-11) 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.

**Note 1:** PORTD is available on PIC16F884/887 only.

The TRISD register (Register 3-12) controls the PORTD pin output drivers, even when they are being used as analog inputs. The user should ensure the bits in the TRISD register are maintained set when using them as analog inputs. I/O pins configured as analog input always read '0'.

**EXAMPLE 3-5: INITIALIZING PORTD**

```
BANKSEL PORTD        ;
CLRF    PORTD        ;Init PORTD
BANKSEL TRISD        ;
MOVLW   B'00001100'  ;Set RD<3:2> as inputs
MOVWF   TRISD        ;and set RD<7:4,1:0>
                     ;as outputs
```

**REGISTER 3-11: PORTD: PORTD REGISTER**

| R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x |
|---|---|---|---|---|---|---|---|
| RD7 | RD6 | RD5 | RD4 | RD3 | RD2 | RD1 | RD0 |
| 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 **RD<7:0>**: PORTD General Purpose I/O Pin bit
1 = Port pin is > VIH
0 = Port pin is < VIL

**REGISTER 3-12: TRISD: PORTD TRI-STATE 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 |
|---|---|---|---|---|---|---|---|
| TRISD7 | TRISD6 | TRISD5 | TRISD4 | TRISD3 | TRISD2 | TRISD1 | TRISD0 |
| 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 |

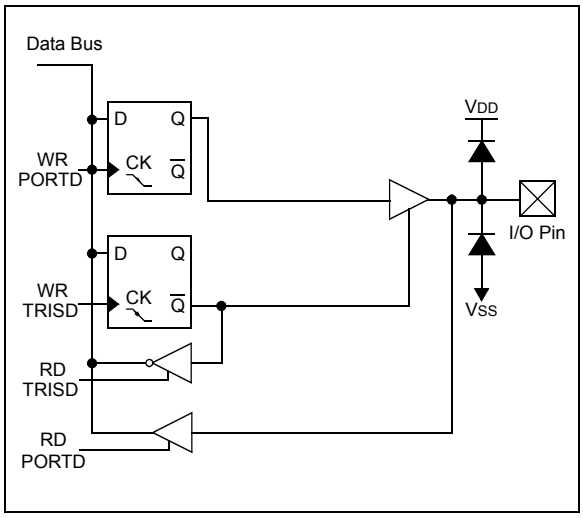
bit 7-0 **TRISD<7:0>:** PORTD Tri-State Control bit
1 = PORTD pin configured as an input (tri-stated)
0 = PORTD pin configured as an output

### 3.6.1 RD<4:0>

Figure 3-19 shows the diagram for these pins. These pins are configured to function as general purpose I/O's.

| | |
|---|---|
| **Note:** | RD<4:0> is available on PIC16F884/887 only. |

**FIGURE 3-19: BLOCK DIAGRAM OF RD<4:0>**




### 3.6.2 RD5/P1B(1)

Figure 3-20 shows the diagram for this pin. This pin is configurable to function as one of the following:

- a general purpose I/O
- a PWM output

| | |
|---|---|
| **Note 1:** | RD5/P1B is available on PIC16F884/887 only. See RB2/AN8/P1B for this function on PIC16F882/883/886. |

### 3.6.3 RD6/P1C(1)

Figure 3-20 shows the diagram for this pin. This pin is configurable to function as one of the following:

- a general purpose I/O
- a PWM output

| | |
|---|---|
| **Note 1:** | RD6/P1C is available on PIC16F884/887 only. See RB1/AN10/P1C/C12IN3- for this function on PIC16F882/883/886. |

### 3.6.4 RD7/P1D(1)

Figure 3-20 shows the diagram for this pin. This pin is configurable to function as one of the following:

- a general purpose I/O
- a PWM output

| | |
|---|---|
| **Note 1:** | RD7/P1D is available on PIC16F884/887 only. See RB4/AN11/P1D for this function on PIC16F882/883/886. |

**FIGURE 3-20: BLOCK DIAGRAM OF RD<7:5>**

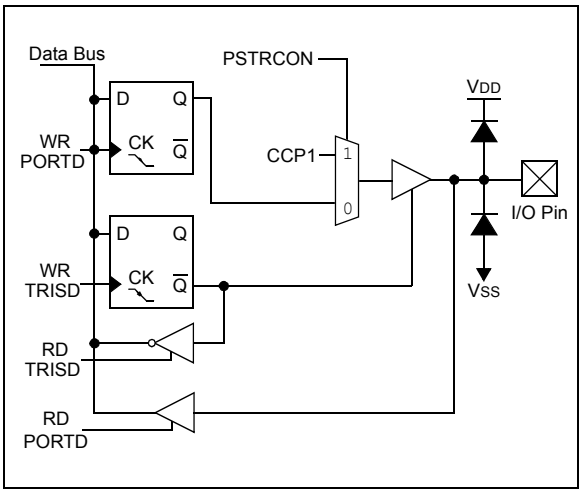



**TABLE 3-4: SUMMARY OF REGISTERS ASSOCIATED WITH PORTD**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|
| PORTD | RD7 | RD6 | RD5 | RD4 | RD3 | RD2 | RD1 | RD0 | 59 |
| PSTRCON | — | — | — | STRSYNC | STRD | STRC | STRB | STRA | 150 |
| TRISD | TRISD7 | TRISD6 | TRISD5 | TRISD4 | TRISD3 | TRISD2 | TRISD1 | TRISD0 | 59 |

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


© 2006-2012 Microchip Technology Inc.

## 3.7 PORTE and TRISE Registers

PORTE(1) is a 4-bit wide, bidirectional port. The corresponding data direction register is TRISE. Setting a TRISE bit (= 1) will make the corresponding PORTE pin an input (i.e., put the corresponding output driver in a High-Impedance mode). Clearing a TRISE bit (= 0) will make the corresponding PORTE pin an output (i.e., enable the output driver and put the contents of the output latch on the selected pin). The exception is RE3, which is input only and its TRIS bit will always read as '1'. Example 3-6 shows how to initialize PORTE.

Reading the PORTE register (Register 3-13) 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. RE3 reads '0' when MCLRE = 1.

**Note 1:** RE<2:0> pins are available on PIC16F884/887 only.

The TRISE register (Register 3-14) controls the PORTE pin output drivers, even when they are being used as analog inputs. The user should ensure the bits in the TRISE 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'.

**EXAMPLE 3-6: INITIALIZING PORTE**

```
BANKSEL PORTE          ;
CLRF    PORTE          ;Init PORTE
BANKSEL ANSEL          ;
CLRF    ANSEL          ;digital I/O
BCF     STATUS,RP1     ;Bank 1
BANKSEL TRISE          ;
MOVLW   B'00001100'    ;Set RE<3:2> as inputs
MOVWF   TRISE          ;and set RE<1:0>
                       ;as outputs
```

**REGISTER 3-13: PORTE: PORTE REGISTER**

| U-0 | U-0 | U-0 | U-0 | R-x | R/W-x | R/W-x | R/W-x |
|---|---|---|---|---|---|---|---|
| — | — | — | — | RE3 | RE2 | RE1 | RE0 |
| 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 **RD<3:0>**: PORTE General Purpose I/O Pin bit
1 = Port pin is > VIH
0 = Port pin is < VIL

**REGISTER 3-14: TRISE: PORTE TRI-STATE REGISTER**

| U-0 | U-0 | U-0 | U-0 | R-1(1) | R/W-1 | R/W-1 | R/W-1 |
|---|---|---|---|---|---|---|---|
| — | — | — | — | TRISE3 | TRISE2 | TRISE1 | TRISE0 |
| 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 **TRISE<3:0>:** PORTE Tri-State Control bit
1 = PORTE pin configured as an input (tri-stated)
0 = PORTE pin configured as an output

**Note 1:** TRISE<3> always reads '1'.

### 3.7.1 RE0/AN5(1)

This pin is configurable to function as one of the following:

- a general purpose I/O
- an analog input for the ADC

**Note 1:** RE0/AN5 is available on PIC16F884/887 only.

### 3.7.2 RE1/AN6(1)

This pin is configurable to function as one of the following:

- a general purpose I/O
- an analog input for the ADC

**Note 1:** RE1/AN6 is available on PIC16F884/887 only.

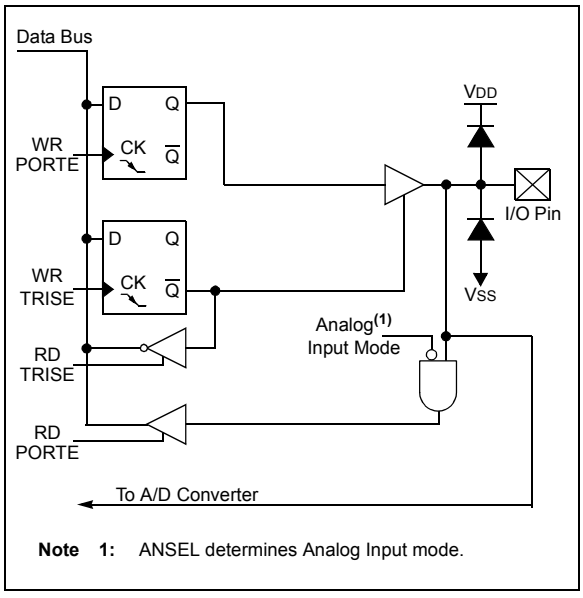
### 3.7.3 RE2/AN7(1)

This pin is configurable to function as one of the following:

- a general purpose I/O
- an analog input for the ADC

**Note 1:** RE2/AN7 is available on PIC16F884/887 only.

**FIGURE 3-21: BLOCK DIAGRAM OF RE<2:0>**



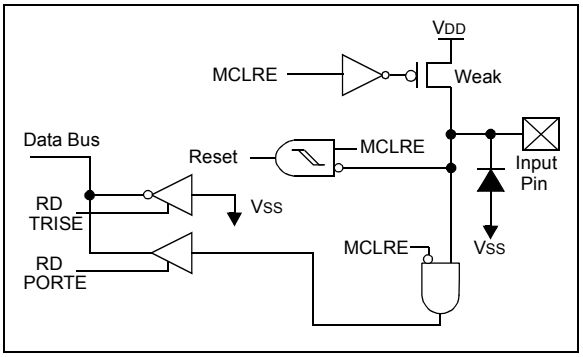

**Note 1:** ANSEL determines Analog Input mode.

### 3.7.4 RE3/$\overline{MCLR}$/VPP

Figure 3-22 shows the diagram for this pin. This pin is configurable to function as one of the following:

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

**FIGURE 3-22: BLOCK DIAGRAM OF RE3**





© 2006-2012 Microchip Technology Inc.

**TABLE 3-5: SUMMARY OF REGISTERS ASSOCIATED WITH PORTE**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|
| ANSEL | ANS7 | ANS6 | ANS5 | ANS4 | ANS3 | ANS2 | ANS1 | ANS0 | 42 |
| PORTE | — | — | — | — | RE3 | RE2 | RE1 | RE0 | 61 |
| TRISE | — | — | — | — | TRISE3 | TRISE2 | TRISE1 | TRISE0 | 61 |

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

**NOTES:**


© 2006-2012 Microchip Technology Inc.

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

### 4.1 Overview

The oscillator module has a wide 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.

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

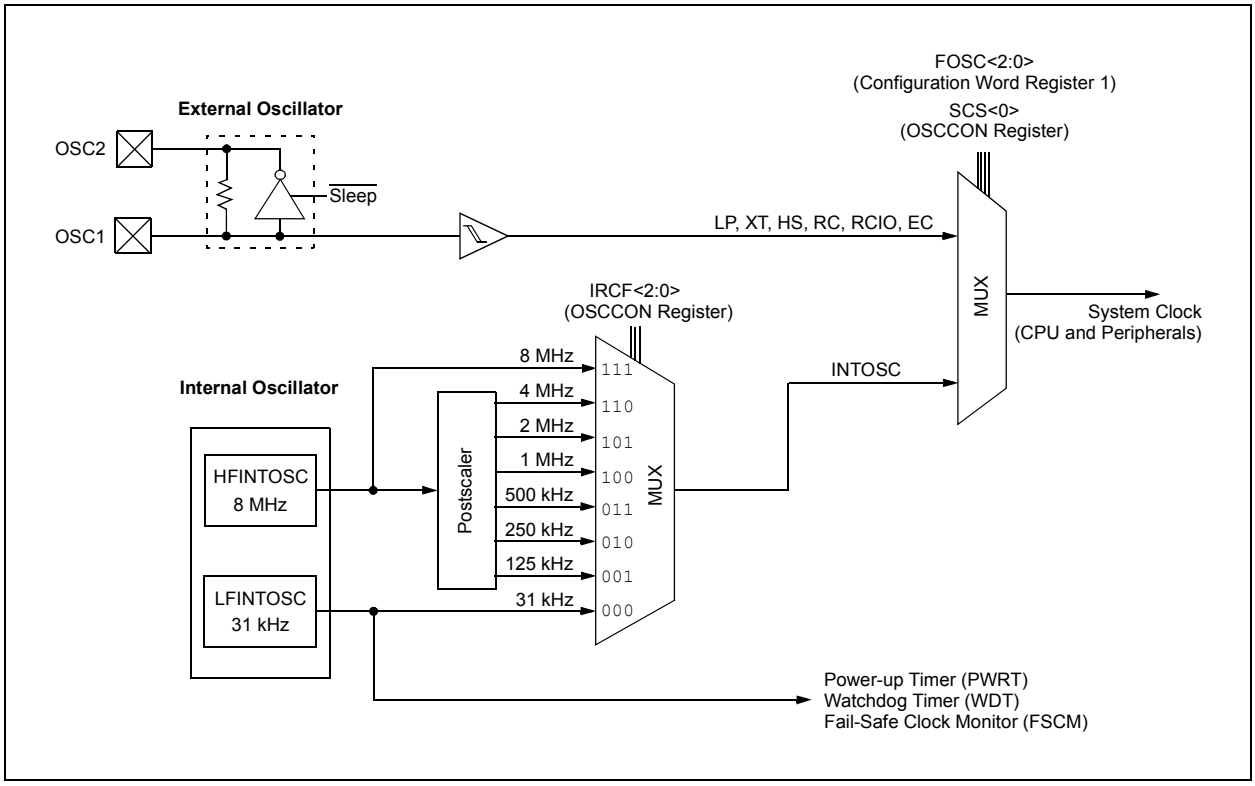
- Selectable system clock source between external or internal via software.
- Two-Speed Start-up mode, which minimizes latency between external oscillator start-up and code execution.
- Fail-Safe Clock Monitor (FSCM) designed to detect a failure of the external clock source (LP, XT, HS, EC or RC modes)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

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

1. EC – External clock with I/O on OSC2/CLKOUT.
2. LP – 32 kHz Low-Power Crystal mode.
3. XT – Medium Gain Crystal or Ceramic Resonator Oscillator mode.
4. HS – High Gain Crystal or Ceramic Resonator mode.
5. RC – External Resistor-Capacitor (RC) with FOSC/4 output on OSC2/CLKOUT.
6. RCIO – External Resistor-Capacitor (RC) with I/O on OSC2/CLKOUT.
7. INTOSC – Internal oscillator with FOSC/4 output on OSC2 and I/O on OSC1/CLKIN.
8. INTOSCIO – Internal oscillator with I/O on OSC1/CLKIN and OSC2/CLKOUT.

Clock Source modes are configured by the FOSC<2:0> bits in the Configuration Word Register 1 (CONFIG1). The internal clock can be generated from two internal oscillators. The HFINTOSC is a calibrated high-frequency oscillator. The LFINTOSC is an uncalibrated low-frequency oscillator.

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



## 4.2 Oscillator Control

The Oscillator Control (OSCCON) register (Figure 4-1) controls the system clock and frequency selection options. The OSCCON register contains the following bits:

- Frequency selection bits (IRCF)
- Frequency Status bits (HTS, LTS)
- System clock control bits (OSTS, SCS)

## REGISTER DEFINITIONS: OSCILLATOR CONTROL

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

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

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

bit 3 **OSTS:** Oscillator Start-up Time-out Status bit(1)
1 = Device is running from the clock defined by FOSC<2:0> of the CONFIG1 register
0 = Device is running from the internal oscillator (HFINTOSC or LFINTOSC)

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

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

bit 0 **SCS:** System Clock Select bit
1 = Internal oscillator is used for system clock
0 = Clock source defined by FOSC<2:0> of the CONFIG1 register

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


© 2006-2012 Microchip Technology Inc.

## 4.3 Clock Source Modes

Clock Source modes can be classified as external or internal.

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

The system clock can be selected between external or internal clock sources via the System Clock Select (SCS) bit of the OSCCON register. See **Section 4.6 "Clock Switching"** for additional information.

## 4.4 External Clock Modes

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

If the oscillator module is configured for LP, XT or HS modes, the Oscillator Start-up Timer (OST) counts 1024 oscillations from OSC1. This occurs following a Power-on Reset (POR) and when the Power-up Timer (PWRT) has expired (if configured), or a wake-up from Sleep. During this time, the program counter does not increment and program execution is suspended. The OST ensures that the oscillator circuit, using a quartz crystal resonator or ceramic resonator, has started and is providing a stable system clock to the oscillator module. When switching between clock sources, a delay is required to allow the new clock to stabilize. These oscillator delays are shown in Table 4-1.

In order to minimize latency between external oscillator start-up and code execution, the Two-Speed Clock Start-up mode can be selected (see **Section 4.7 "Two-Speed Clock Start-up Mode"**).

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

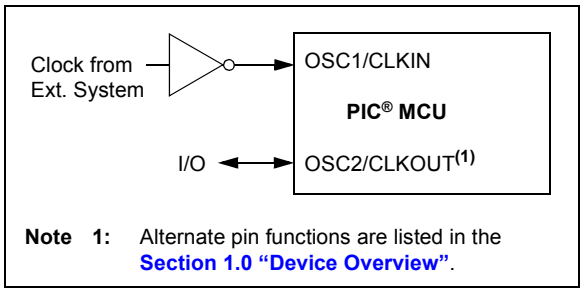
| Switch From | Switch To | Frequency | Oscillator Delay |
|---|---|---|---|
| Sleep/POR | LFINTOSC<br>HFINTOSC | 31 kHz<br>125 kHz to 8 MHz | Oscillator Warm-up Delay (TWARM) |
| Sleep/POR | EC, RC | DC – 20 MHz | 2 cycles |
| LFINTOSC (31 kHz) | EC, RC | DC – 20 MHz | 1 cycle of each |
| Sleep/POR | LP, XT, HS | 32 kHz to 20 MHz | 1024 Clock Cycles (OST) |
| LFINTOSC (31 kHz) | HFINTOSC | 125 kHz to 8 MHz | 1 μs (approx.) |

### 4.4.2 EC MODE

The External Clock (EC) mode allows an externally generated logic level as the system clock source. When operating in this mode, an external clock source is connected to the OSC1 input and the OSC2 is available for general purpose I/O. Figure 4-2 shows the pin connections for EC mode.

The Oscillator Start-up Timer (OST) is disabled when EC mode is selected. Therefore, there is no delay in operation after a Power-on Reset (POR) or wake-up from Sleep. Because the 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 4-2: EXTERNAL CLOCK (EC) MODE OPERATION**




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

### 4.4.3 LP, XT, HS MODES

The LP, XT and HS modes support the use of quartz crystal resonators or ceramic resonators connected to OSC1 and OSC2 (Figure 4-3). The mode selects a low, medium or high gain setting of the internal inverter-amplifier to support various resonator types and speed.

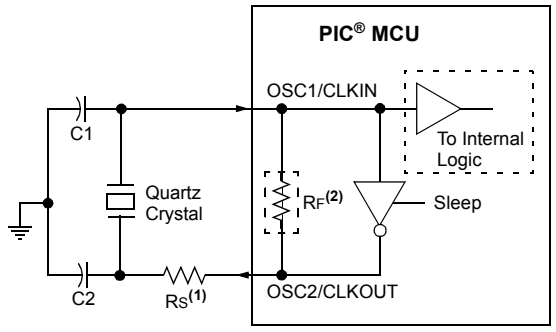
**LP** Oscillator mode selects the lowest gain setting of the internal inverter-amplifier. LP mode current consumption is the least of the three modes. This mode is designed to drive only 32.768 kHz tuning-fork type crystals (watch crystals).

**XT** Oscillator mode selects the intermediate gain setting of the internal inverter-amplifier. XT mode current consumption is the medium of the three modes. This mode is best suited to drive resonators with a medium drive level specification.

**HS** Oscillator mode selects the highest gain setting of the internal inverter-amplifier. HS mode current consumption is the highest of the three modes. This mode is best suited for resonators that require a high drive setting.

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

**FIGURE 4-3: QUARTZ CRYSTAL OPERATION (LP, XT OR HS MODE)**




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

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

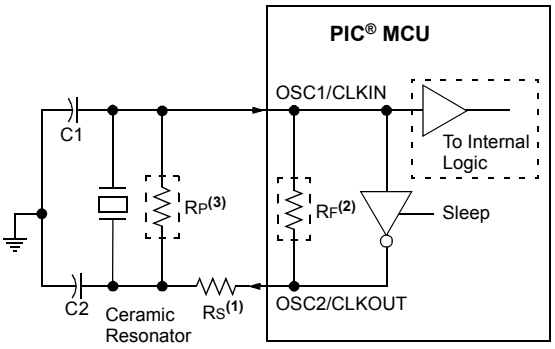
**Note 1:** Quartz crystal characteristics vary according to type, package and manufacturer. The user should consult the manufacturer data sheets for specifications and recommended application.

**2:** Always verify oscillator performance over the VDD and temperature range that is expected for the application.

**3:** For oscillator design assistance, reference the following Microchip Applications Notes:

- AN826, "*Crystal Oscillator Basics and Crystal Selection for rfPIC® and PIC® Devices*" (DS00826)
- AN849, "*Basic PIC® Oscillator Design*" (DS00849)
- AN943, "*Practical PIC® Oscillator Analysis and Design*" (DS00943)
- AN949, "*Making Your Oscillator Work*" (DS00949)

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




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

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

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

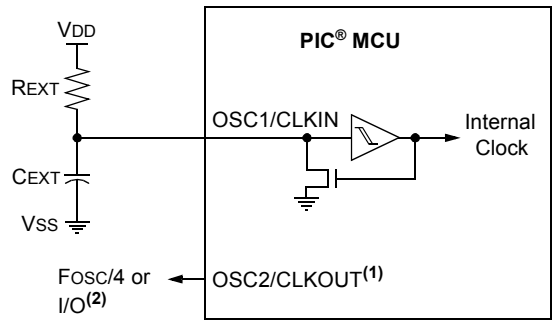

© 2006-2012 Microchip Technology Inc.

### 4.4.4 EXTERNAL RC MODES

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

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

**FIGURE 4-5: EXTERNAL RC MODES**




Recommended values: 10 kΩ ≤ REXT ≤ 100 kΩ, <3V
3 kΩ ≤ REXT ≤ 100 kΩ, 3-5V
CEXT > 20 pF, 2-5V

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

**2:** Output depends upon RC or RCIO Clock mode.

In RCIO mode, the RC circuit is connected to OSC1. OSC2 becomes an additional general purpose I/O pin.

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

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

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

## 4.5 Internal Clock Modes

The oscillator module has two independent, internal oscillators that can be configured or selected as the system clock source.

1. The **HFINTOSC** (High-Frequency Internal Oscillator) is factory calibrated and operates at 8 MHz. The frequency of the HFINTOSC can be user-adjusted via software using the OSCTUNE register (Register 4-2).
2. The **LFINTOSC** (Low-Frequency Internal Oscillator) is uncalibrated and operates at 31 kHz.

The system clock speed can be selected via software using the Internal Oscillator Frequency Select bits IRCF<2:0> of the OSCCON register.

The system clock can be selected between external or internal clock sources via the System Clock Selection (SCS) bit of the OSCCON register. See **Section 4.6 "Clock Switching"** for more information.

### 4.5.1 INTOSC AND INTOSCIO MODES

The INTOSC and INTOSCIO modes configure the internal oscillators as the system clock source when the device is programmed using the oscillator selection or the FOSC<2:0> bits in the Configuration Word Register 1 (CONFIG1).

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

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

### 4.5.2 HFINTOSC

The High-Frequency Internal Oscillator (HFINTOSC) is a factory calibrated 8 MHz internal clock source. The frequency of the HFINTOSC can be altered via software using the OSCTUNE register (Register 4-2).

The output of the HFINTOSC connects to a postscaler and multiplexer (see Figure 4-1). One of seven frequencies can be selected via software using the IRCF<2:0> bits of the OSCCON register. See **Section 4.5.4 "Frequency Select Bits (IRCF)"** for more information.

The HFINTOSC is enabled by selecting any frequency between 8 MHz and 125 kHz by setting the IRCF<2:0> bits of the OSCCON register ≠ 000. Then, set the System Clock Source (SCS) bit of the OSCCON register to '1' or enable Two-Speed Start-up by setting the IESO bit in the Configuration Word Register 1 (CONFIG1) to '1'.

The HF Internal Oscillator (HTS) bit of the OSCCON register indicates whether the HFINTOSC is stable or not.

#### 4.5.2.1 OSCTUNE Register

The HFINTOSC 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 HFINTOSC frequency will begin shifting to the new frequency. Code execution continues during this shift. There is no indication that the shift has occurred.

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

**REGISTER 4-2: OSCTUNE: OSCILLATOR TUNING REGISTER**

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

| **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 factory-calibrated frequency.
11111 =
•
•
•
10000 = Minimum frequency


© 2006-2012 Microchip Technology Inc.

### 4.5.3 LFINTOSC

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

The output of the LFINTOSC connects to a postscaler and multiplexer (see Figure 4-1). Select 31 kHz, via software, using the IRCF<2:0> bits of the OSCCON register. See **Section 4.5.4 "Frequency Select Bits (IRCF)"** for more information. The LFINTOSC is also the frequency for the Power-up Timer (PWRT), Watchdog Timer (WDT) and Fail-Safe Clock Monitor (FSCM).

The LFINTOSC is enabled by selecting 31 kHz (IRCF<2:0> bits of the OSCCON register = 000) as the system clock source (SCS bit of the OSCCON register = 1), or when any of the following are enabled:

- Two-Speed Start-up IESO bit of the Configuration Word Register 1 = 1 and IRCF<2:0> bits of the OSCCON register = 000
- Power-up Timer (PWRT)
- Watchdog Timer (WDT)
- Fail-Safe Clock Monitor (FSCM)

The LF Internal Oscillator (LTS) bit of the OSCCON register indicates whether the LFINTOSC is stable or not.

### 4.5.4 FREQUENCY SELECT BITS (IRCF)

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

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

| | |
|---|---|
| **Note:** | Following any Reset, the IRCF<2:0> bits of the OSCCON register are set to '110' and the frequency selection is set to 4 MHz. The user can modify the IRCF bits to select a different frequency. |

### 4.5.5 HFINTOSC AND LFINTOSC CLOCK SWITCH TIMING

When switching between the LFINTOSC and the HFINTOSC, the new oscillator may already be shut down to save power (see Figure 4-6). If this is the case, there is a delay after the IRCF<2:0> bits of the OSCCON register are modified before the frequency selection takes place. The LTS and HTS bits of the OSCCON register will reflect the current active status of the LFINTOSC and HFINTOSC oscillators. The timing of a frequency selection is as follows:

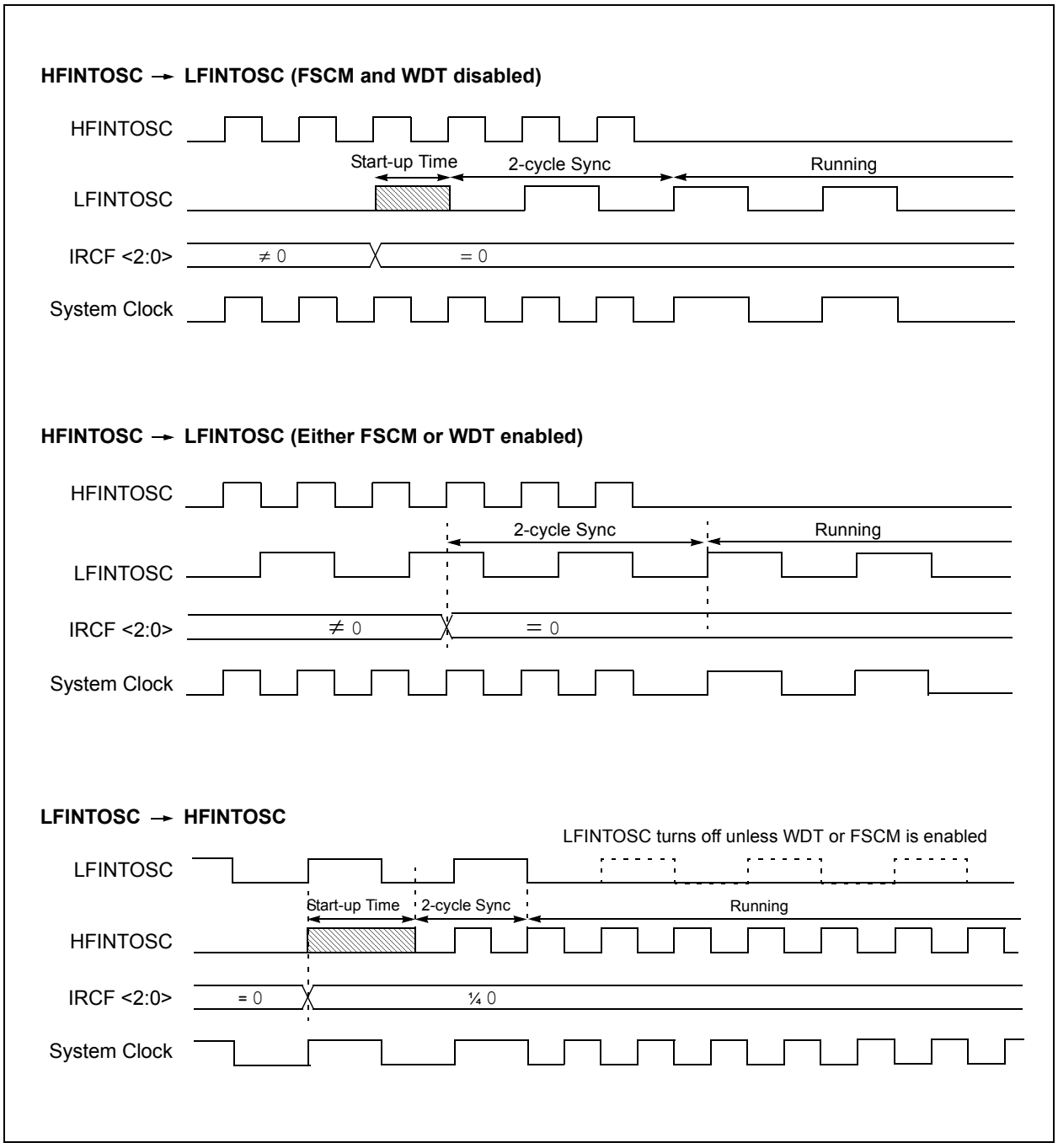
1. IRCF<2:0> bits of the OSCCON register are modified.
2. If the new clock is shut down, a clock start-up delay is started.
3. Clock switch circuitry waits for a falling edge of the current clock.
4. CLKOUT is held low and the clock switch circuitry waits for a rising edge in the new clock.
5. CLKOUT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

See Figure 4-1 for more details.

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

Start-up delay specifications are located in the oscillator tables of **Section 17.0 "Electrical Specifications"**.

**FIGURE 4-6: INTERNAL OSCILLATOR SWITCH TIMING**





© 2006-2012 Microchip Technology Inc.

## 4.6 Clock Switching

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

### 4.6.1 SYSTEM CLOCK SELECT (SCS) BIT

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

- When the SCS bit of the OSCCON register = 0, the system clock source is determined by configuration of the FOSC<2:0> bits in the Configuration Word Register 1 (CONFIG1).
- When the SCS bit of the OSCCON register = 1, the system clock source is chosen by the internal oscillator frequency selected by the IRCF<2:0> bits of the OSCCON register. After a Reset, the SCS bit of the OSCCON register is always cleared.

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

### 4.6.2 OSCILLATOR START-UP TIME-OUT STATUS (OSTS) BIT

The Oscillator Start-up Time-out Status (OSTS) bit of the OSCCON register indicates whether the system clock is running from the external clock source, as defined by the FOSC<2:0> bits in the Configuration Word Register 1 (CONFIG1), or from the internal clock source. In particular, OSTS indicates that the Oscillator Start-up Timer (OST) has timed out for LP, XT or HS modes.

## 4.7 Two-Speed Clock Start-up Mode

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

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

| | |
|---|---|
| **Note:** | Executing a SLEEP instruction will abort the oscillator start-up time and will cause the OSTS bit of the OSCCON register to remain clear. |

When the oscillator module is configured for LP, XT or HS modes, the Oscillator Start-up Timer (OST) is enabled (see **Section 4.4.1 "Oscillator Start-up Timer (OST)"**). The OST will suspend program execution until 1024 oscillations are counted. Two-Speed Start-up mode minimizes the delay in code execution by operating from the internal oscillator as the OST is counting. When the OST count reaches 1024 and the OSTS bit of the OSCCON register is set, program execution switches to the external oscillator.

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

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

- IESO (of the Configuration Word Register 1) = 1; Internal/External Switchover bit (Two-Speed Start-up mode enabled).
- SCS (of the OSCCON register) = 0.
- FOSC<2:0> bits in the Configuration Word Register 1 (CONFIG1) configured for LP, XT or HS mode.

Two-Speed Start-up mode is entered after:

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

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

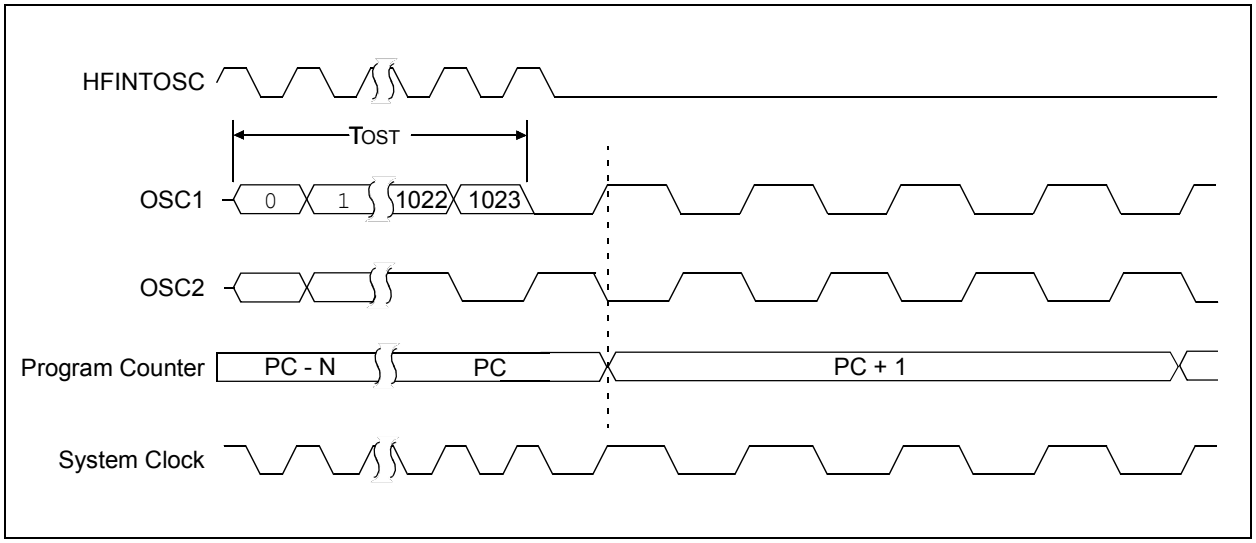
### 4.7.2 TWO-SPEED START-UP SEQUENCE

1. Wake-up from Power-on Reset or Sleep.
2. Instructions begin execution by the internal oscillator at the frequency set in the IRCF<2:0> bits of the OSCCON register.
3. OST enabled to count 1024 clock cycles.
4. OST timed out, wait for falling edge of the internal oscillator.
5. OSTS is set.
6. System clock held low until the next falling edge of new clock (LP, XT or HS mode).
7. System clock is switched to external clock source.

### 4.7.3 CHECKING TWO-SPEED CLOCK STATUS

Checking the state of the OSTS bit of the OSCCON register will confirm if the microcontroller is running from the external clock source, as defined by the FOSC<2:0> bits in the Configuration Word Register 1 (CONFIG1), or the internal oscillator.

**FIGURE 4-7: TWO-SPEED START-UP**



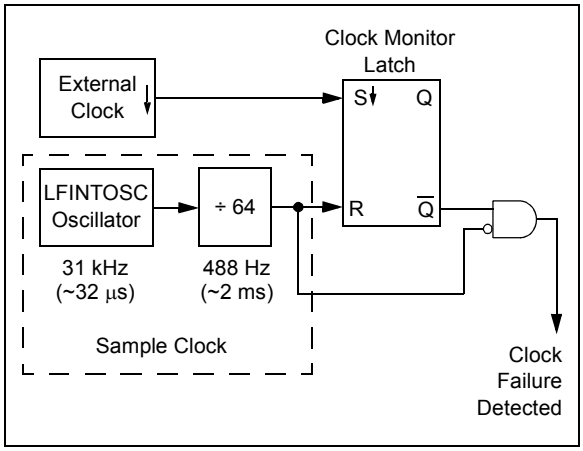


© 2006-2012 Microchip Technology Inc.

## 4.8 Fail-Safe Clock Monitor

The Fail-Safe Clock Monitor (FSCM) allows the device to continue operating should the external oscillator fail. The FSCM can detect oscillator failure any time after the Oscillator Start-up Timer (OST) has expired. The FSCM is enabled by setting the FCMEN bit in the Configuration Word Register 1 (CONFIG1). The FSCM is applicable to all external Oscillator modes (LP, XT, HS, EC, RC and RCIO).

**FIGURE 4-8: FSCM BLOCK DIAGRAM**




### 4.8.1 FAIL-SAFE DETECTION

The FSCM module detects a failed oscillator by comparing the external oscillator to the FSCM sample clock. The sample clock is generated by dividing the LFINTOSC by 64. See Figure 4-8. Inside the fail detector block is a latch. The external clock sets the latch on each falling edge of the external clock. The sample clock clears the latch on each rising edge of the sample clock. A failure is detected when an entire half-cycle of the sample clock elapses before the primary clock goes low.

### 4.8.2 FAIL-SAFE OPERATION

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

The internal clock source chosen by the FSCM is determined by the IRCF<2:0> bits of the OSCCON register. This allows the internal oscillator to be configured before a failure occurs.

### 4.8.3 FAIL-SAFE CONDITION CLEARING

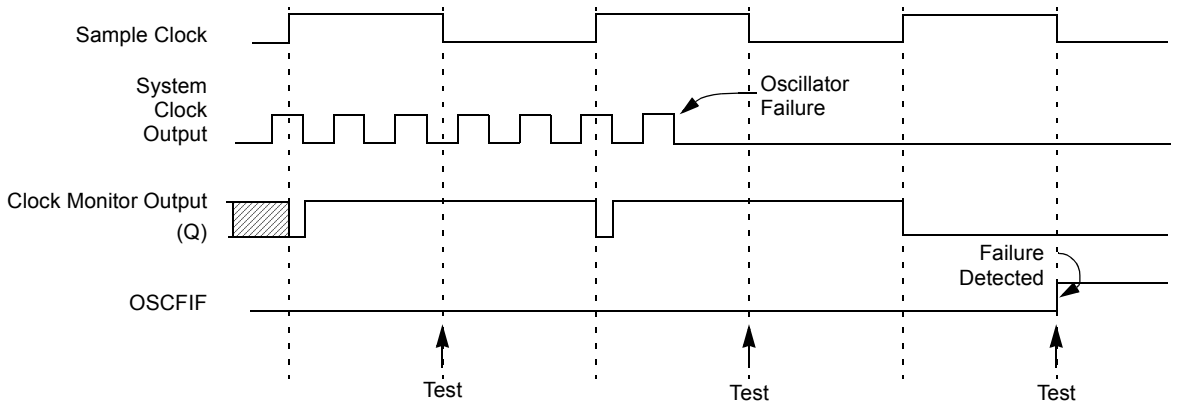
The Fail-Safe condition is cleared after a Reset, executing a SLEEP instruction or toggling the SCS bit of the OSCCON register. When the SCS bit is toggled, the OST is restarted. While the OST is running, the device continues to operate from the INTOSC selected in OSCCON. When the OST times out, the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

### 4.8.4 RESET OR WAKE-UP FROM SLEEP

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

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

**FIGURE 4-9: FSCM TIMING DIAGRAM**




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

**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 | — | IRCF2 | IRCF1 | IRCF0 | OSTS | HTS | LTS | SCS | 66 |
| OSCTUNE | — | — | — | TUN4 | TUN3 | TUN2 | TUN1 | TUN0 | 70 |
| PIE2 | OSFIE | C2IE | C1IE | EEIE | BCLIE | ULPWUIE | — | CCP2IE | 35 |
| PIR2 | OSFIF | C2IF | C1IF | EEIF | BCLIF | ULPWUIF | — | CCP2IF | 37 |

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

**TABLE 4-3: SUMMARY OF CONFIGURATION WORD ASSOCIATED WITH CLOCK SOURCES**

| Name | Bits | Bit -/7 | Bit -/6 | Bit 13/5 | Bit 12/4 | Bit 11/3 | Bit 10/2 | Bit 9/1 | Bit 8/0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|---|
| CONFIG1(1) | 13:8 | — | — | $\overline{\text{DEBUG}}$ | LVP | FCMEN | IESO | BOREN 1 | BOREN0 | 214 |
| | 7:0 | $\overline{\text{CPD}}$ | $\overline{\text{CP}}$ | MCLRE | $\overline{\text{PWRTE}}$ | WDTE | FOSC 2 | FOSC 1 | FOSC 0 | |

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

**Note 1:** See Configuration Word Register 1 (Register 14-1) for operation of all register bits.


© 2006-2012 Microchip Technology Inc.

## 5.0 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 5-1 is a block diagram of the Timer0 module.

### 5.1 Timer0 Operation

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

#### 5.1.1 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'.

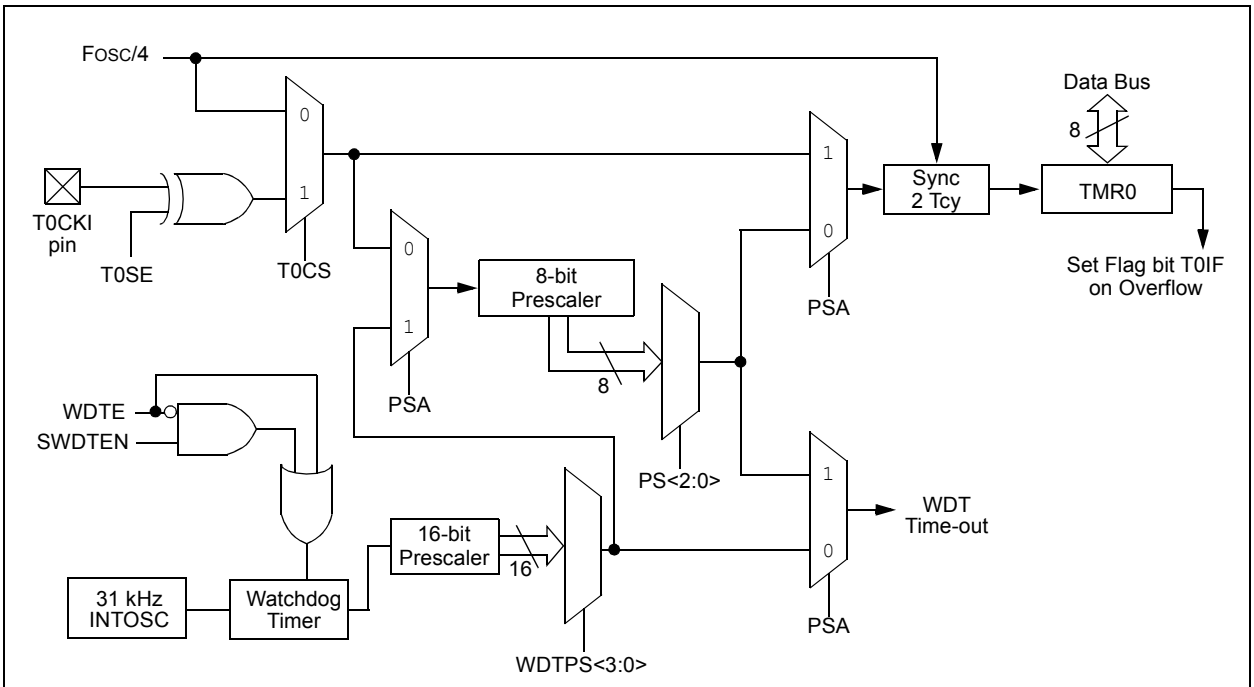
When TMR0 is written, the increment is inhibited for two instruction cycles immediately following the write.

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

#### 5.1.2 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 register. Counter mode is selected by setting the T0CS bit of the OPTION register to '1'.

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




**Note 1:** T0SE, T0CS, PSA, PS<2:0> are bits in the OPTION register.
**2:** SWDTEN and WDTPS<3:0> are bits in the WDTCON register.
**3:** WDTE bit is in the Configuration Word Register1.

### 5.1.3 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 8 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, a CLRWDT instruction will clear the prescaler along with the WDT.

#### 5.1.3.1 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 5-1, must be executed.

**EXAMPLE 5-1: 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 5-2).

**EXAMPLE 5-2: 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       ;
```

### 5.1.4 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. |
|---|---|

### 5.1.5 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 the **Section 17.0 "Electrical Specifications"**.


© 2006-2012 Microchip Technology Inc.

## REGISTER DEFINITIONS: OPTION REGISTER

### REGISTER 5-1: 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 |
|---|---|---|---|---|---|---|---|
| $\overline{\text{RBPU}}$ | INTEDG | T0CS | T0SE | PSA | PS2 | PS1 | PS0 |
| 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 **$\overline{\text{RBPU}}$:** PORTB Pull-up Enable bit
1 = PORTB pull-ups are disabled
0 = PORTB pull-ups are enabled by individual PORT latch values

bit 6 **INTEDG:** Interrupt Edge Select bit
1 = Interrupt on rising edge of 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 |

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

### TABLE 5-1: 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 | Timer0 Module Register | | | | | | | | 77 |
| INTCON | GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | RBIF | 33 |
| OPTION_REG | $\overline{\text{RBPU}}$ | INTEDG | T0CS | T0SE | PSA | PS2 | PS1 | PS0 | 79 |
| TRISA | TRISA7 | TRISA6 | TRISA5 | TRISA4 | TRISA3 | TRISA2 | TRISA1 | TRISA0 | 41 |

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

© 2006-2012 Microchip Technology Inc.

**NOTES:**


© 2006-2012 Microchip Technology Inc.

## 6.0 TIMER1 MODULE WITH GATE CONTROL

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

- 16-bit timer/counter register pair (TMR1H:TMR1L)
- Programmable internal or external clock source
- 3-bit prescaler
- Optional LP oscillator
- Synchronous or asynchronous operation
- Timer1 gate (count enable) via comparator or T1G pin
- Interrupt on overflow
- Wake-up on overflow (external clock, Asynchronous mode only)
- Time base for the Capture/Compare function
- Special Event Trigger (with ECCP)
- Comparator output synchronization to Timer1 clock

Figure 6-1 is a block diagram of the Timer1 module.

### 6.1 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.

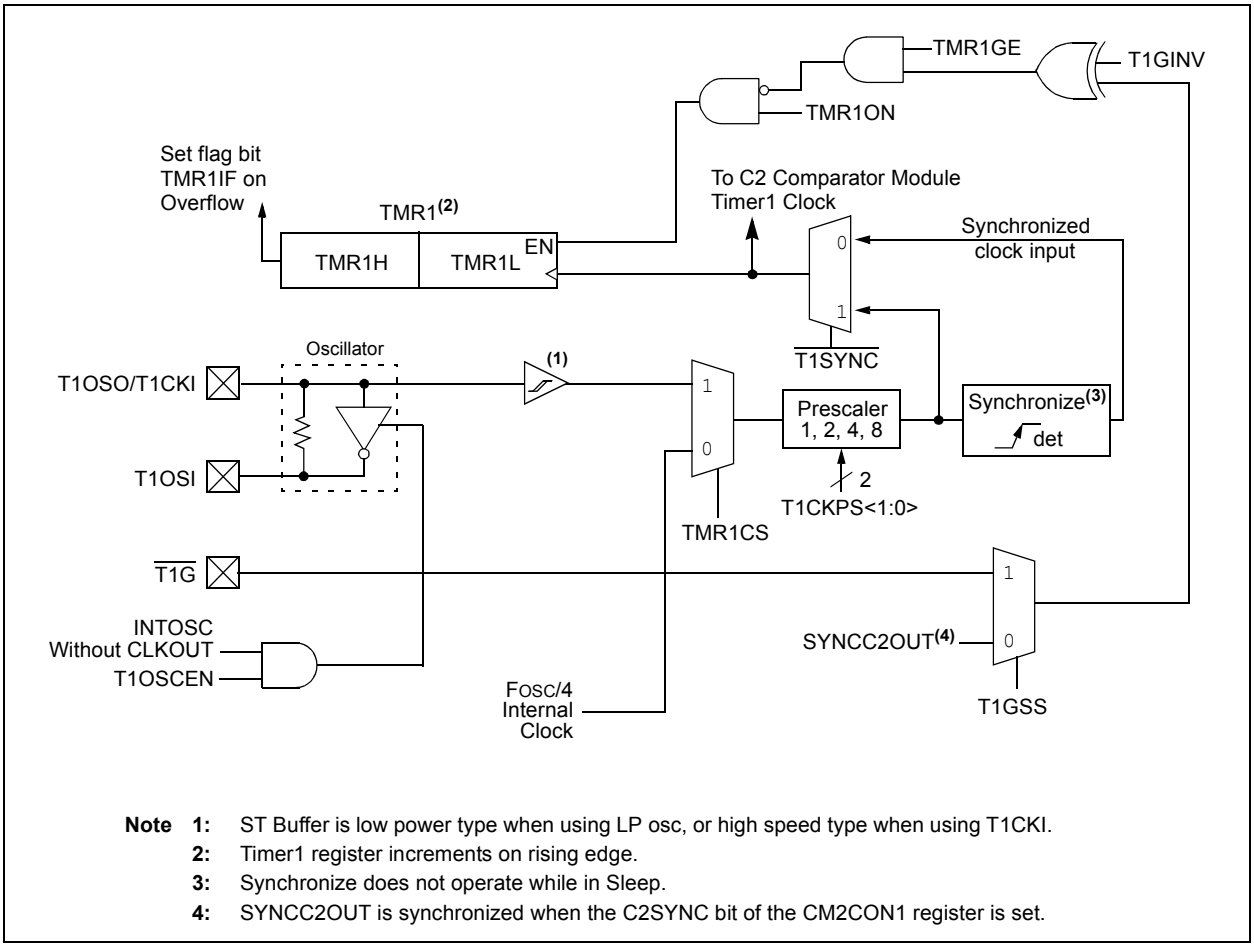
When used with an internal clock source, the module is a timer. When used with an external clock source, the module can be used as either a timer or counter.

### 6.2 Clock Source Selection

The TMR1CS bit of the T1CON register is used to select the clock source. When TMR1CS = 0, the clock source is FOSC/4. When TMR1CS = 1, the clock source is supplied externally.

| Clock Source | TMR1CS |
|---|---|
| FOSC/4 | 0 |
| T1CKI pin | 1 |

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




**Note 1:** ST Buffer is low power type when using LP osc, or high speed type when using T1CKI.
**2:** Timer1 register increments on rising edge.
**3:** Synchronize does not operate while in Sleep.
**4:** SYNCC2OUT is synchronized when the C2SYNC bit of the CM2CON1 register is set.

### 6.2.1 INTERNAL CLOCK SOURCE

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

### 6.2.2 EXTERNAL CLOCK SOURCE

When the external clock source is selected, the Timer1 module may work as a timer or a counter.

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

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

In Counter mode, a falling edge must be registered by the counter prior to the first incrementing rising edge after one or more of the following conditions (see Figure 6-2):

- Timer1 is enabled after POR or BOR Reset
- A write to TMR1H or TMR1L
- T1CKI is high when Timer1 is disabled and when Timer1 is re-enabled T1CKI is low.

## 6.3 Timer1 Prescaler

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

## 6.4 Timer1 Oscillator

A low-power 32.768 kHz oscillator is built-in between pins T1OSI (input) and T1OSO (amplifier output). The oscillator is enabled by setting the T1OSCEN control bit of the T1CON register. The oscillator will continue to run during Sleep.

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

TRISC0 and TRISC1 bits are set when the Timer1 oscillator is enabled. RC0 and RC1 bits read as '0' and TRISC0 and TRISC1 bits read as '1'.

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

## 6.5 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 continues to increment asynchronous to the internal phase clocks. The timer will continue to run during Sleep and can generate an interrupt on overflow, which will wake-up the processor. However, special precautions in software are needed to read/write the timer (see **Section 6.5.1 "Reading and Writing Timer1 in Asynchronous Counter Mode"**).

| | |
|---|---|
| **Note:** | 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 a single spurious increment. |

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

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

For writes, it is recommended that the user simply stop the timer and write the desired values. A write contention may occur by writing to the timer registers, while the register is incrementing. This may produce an unpredictable value in the TMR1H:TTMR1L register pair.

## 6.6 Timer1 Gate

Timer1 gate source is software configurable to be the $\overline{\text{T1G}}$ pin or the output of Comparator C2. This allows the device to directly time external events using $\overline{\text{T1G}}$ or analog events using Comparator C2. See the CM2CON1 register (Register 8-3) for selecting the Timer1 gate source. This feature can simplify the software for a Delta-Sigma A/D converter and many other applications. For more information on Delta-Sigma A/D converters, see the Microchip web site (www.microchip.com).

| | |
|---|---|
| **Note:** | TMR1GE bit of the T1CON register must be set to use the Timer1 gate. |

Timer1 gate can be inverted using the T1GINV bit of the T1CON register, whether it originates from the $\overline{\text{T1G}}$ pin or Comparator C2 output. This configures Timer1 to measure either the active-high or active-low time between events.


© 2006-2012 Microchip Technology Inc.

## 6.7 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:

- Timer1 interrupt enable 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:TTMR1L register pair and the TMR1IF bit should be cleared before enabling interrupts. |
|---|---|

## 6.8 Timer1 Operation During Sleep

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

- 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

The device will wake-up on an overflow and execute the next instruction. If the GIE bit of the INTCON register is set, the device will call the Interrupt Service Routine (0004h).

## 6.9 ECCP Capture/Compare Time Base

The ECCP 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 CCPRxH:CCPRxL register pair on a configured event.

In Compare mode, an event is triggered when the value CCPRxH:CCPRxL register pair matches the value in the TMR1H:TMR1L register pair. This event can be a Special Event Trigger.

See **Section 11.0 “Capture/Compare/PWM Modules (CCP1 and CCP2)”** for more information.

## 6.10 ECCP Special Event Trigger

If an ECCP 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 ECCP module may still be configured to generate a ECCP interrupt.

In this mode of operation, the CCPRxH:CCPRxL register pair effectively becomes the period register for Timer1.

Timer1 should be synchronized to the FOSC 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 ECCP, the write will take precedence.

For more information, see **Section 11.0 “Capture/Compare/PWM Modules (CCP1 and CCP2)”**.

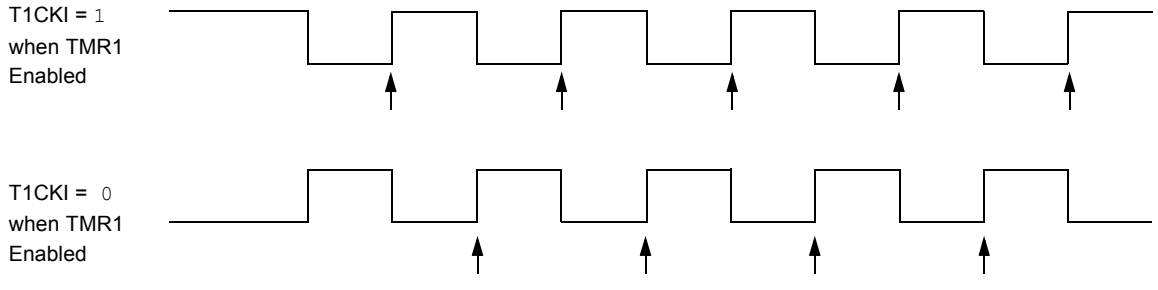
## 6.11 Comparator Synchronization

The same clock used to increment Timer1 can also be used to synchronize the comparator output. This feature is enabled in the Comparator module.

When using the comparator for Timer1 gate, the comparator output should be synchronized to Timer1. This ensures Timer1 does not miss an increment if the comparator changes.

For more information, see **Section 8.0 “Comparator Module”**.

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




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

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

## 6.12 Timer1 Control Register

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

## REGISTER DEFINITIONS: TIMER1 CONTROL

### REGISTER 6-1: T1CON: TIMER1 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 |
|---|---|---|---|---|---|---|---|
| T1GINV(1) | TMR1GE(2) | T1CKPS1 | T1CKPS0 | T1OSCEN | $\overline{\text{T1SYNC}}$ | TMR1CS | 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 **T1GINV:** Timer1 Gate Invert bit(1)
- 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 6 **TMR1GE:** Timer1 Gate Enable bit(2)

If TMR1ON = 0:
This bit is ignored
If TMR1ON = 1:
- 1 = Timer1 counting is controlled by the Timer1 Gate function
- 0 = Timer1 is always counting

bit 5-4 **T1CKPS<1:0>:** Timer1 Input Clock Prescale Select bits
- 11 = 1:8 Prescale Value
- 10 = 1:4 Prescale Value
- 01 = 1:2 Prescale Value
- 00 = 1:1 Prescale Value

bit 3 **T1OSCEN:** LP Oscillator Enable Control bit
- 1 = LP oscillator is enabled for Timer1 clock
- 0 = LP oscillator is off

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

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

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

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

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

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

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


© 2006-2012 Microchip Technology Inc.

**TABLE 6-1: 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 |
|---|---|---|---|---|---|---|---|---|---|
| CM2CON1 | MC1OUT | MC2OUT | C1RSEL | C2RSEL | — | — | T1GSS | C2SYNC | 96 |
| INTCON | GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | RBIF | 33 |
| PIE1 | — | ADIE | RCIE | TXIE | SSPIE | CCP1IE | TMR2IE | TMR1IE | 34 |
| PIR1 | — | ADIF | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | 36 |
| TMR1H | Holding Register for the Most Significant Byte of the 16-bit TMR1 Register | | | | | | | | 81 |
| TMR1L | Holding Register for the Least Significant Byte of the 16-bit TMR1 Register | | | | | | | | 81 |
| T1CON | T1GINV | TMR1GE | T1CKPS1 | T1CKPS0 | T1OSCEN | $\overline{\text{T1SYNC}}$ | TMR1CS | TMR1ON | 84 |

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

**NOTES:**


© 2006-2012 Microchip Technology Inc.

## 7.0 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)
- Software programmable postscaler (1:1 to 1:16)

See Figure 7-1 for a block diagram of Timer2.

### 7.1 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.

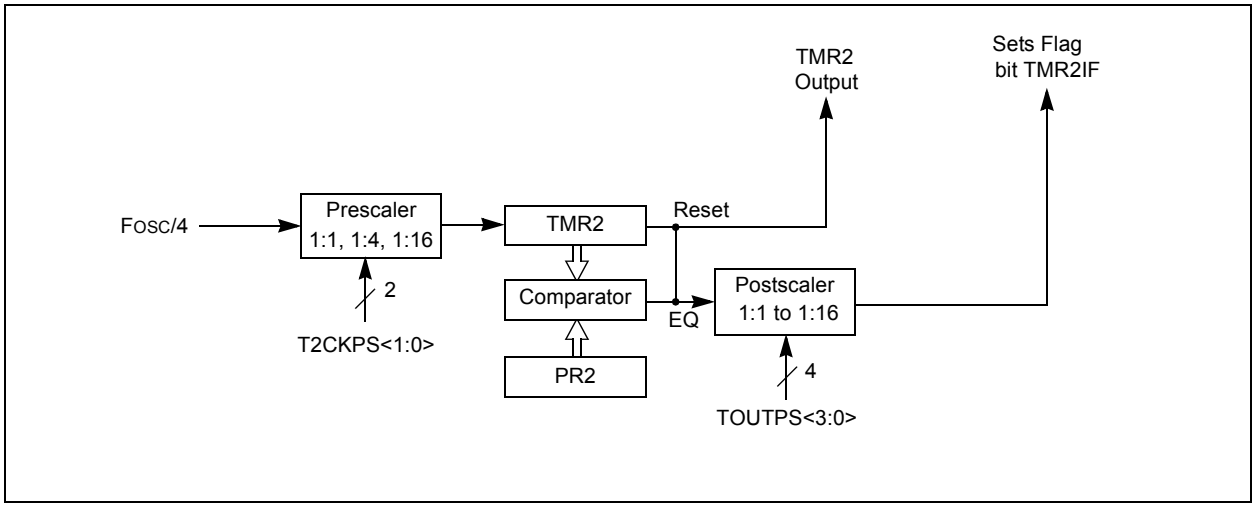
Timer2 is turned on by setting the TMR2ON bit in the T2CON register to a '1'. Timer2 is turned off by clearing the TMR2ON bit to a '0'.

The Timer2 prescaler is controlled by the T2CKPS bits in the T2CON register. The Timer2 postscaler is controlled by the TOUTPS 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 7-1: TIMER2 BLOCK DIAGRAM**



## REGISTER DEFINITIONS: TIMER2 CONTROL

### REGISTER 7-1: 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 |
|---|---|---|---|---|---|---|---|
| — | TOUTPS3 | TOUTPS2 | TOUTPS1 | TOUTPS0 | TMR2ON | T2CKPS1 | T2CKPS0 |
| 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 **TOUTPS<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
1x = Prescaler is 16

### TABLE 7-1: SUMMARY OF ASSOCIATED TIMER2 REGISTERS

| 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 | RBIE | T0IF | INTF | RBIF | 33 |
| PIE1 | — | ADIE | RCIE | TXIE | SSPIE | CCP1IE | TMR2IE | TMR1IE | 34 |
| PIR1 | — | ADIF | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | 36 |
| PR2 | Timer2 Module Period Register | | | | | | | | 87 |
| TMR2 | Holding Register for the 8-bit TMR2 Register | | | | | | | | 87 |
| T2CON | — | TOUTPS3 | TOUTPS2 | TOUTPS1 | TOUTPS0 | TMR2ON | T2CKPS1 | T2CKPS0 | 88 |

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


© 2006-2012 Microchip Technology Inc.

## 8.0 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. The comparators are very useful mixed signal building blocks because they provide analog functionality independent of the program execution. The analog comparator module includes the following features:

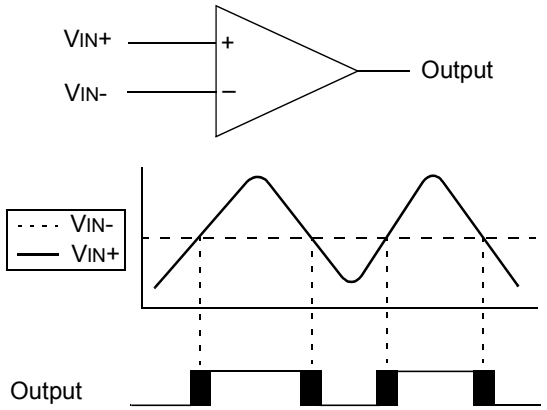
- Independent comparator control
- Programmable input selection
- Comparator output is available internally/externally
- Programmable output polarity
- Interrupt-on-change
- Wake-up from Sleep
- PWM shutdown
- Timer1 gate (count enable)
- Output synchronization to Timer1 clock input
- SR Latch
- Programmable and fixed voltage reference

**Note:** Only Comparator C2 can be linked to Timer1.

### 8.1 Comparator Overview

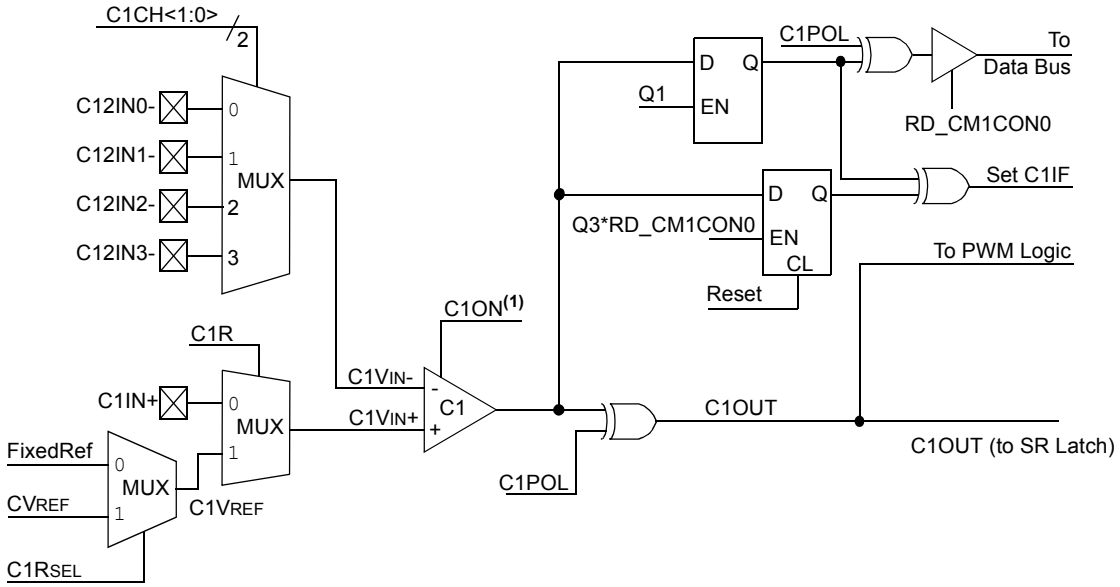
A single comparator is shown in Figure 8-1 along with the relationship between the analog input levels and the digital output. When the analog voltage at VIN+ is less than the analog voltage at VIN-, the output of the comparator is a digital low level. When the analog voltage at VIN+ is greater than the analog voltage at VIN-, the output of the comparator is a digital high level.

**FIGURE 8-1: SINGLE COMPARATOR**



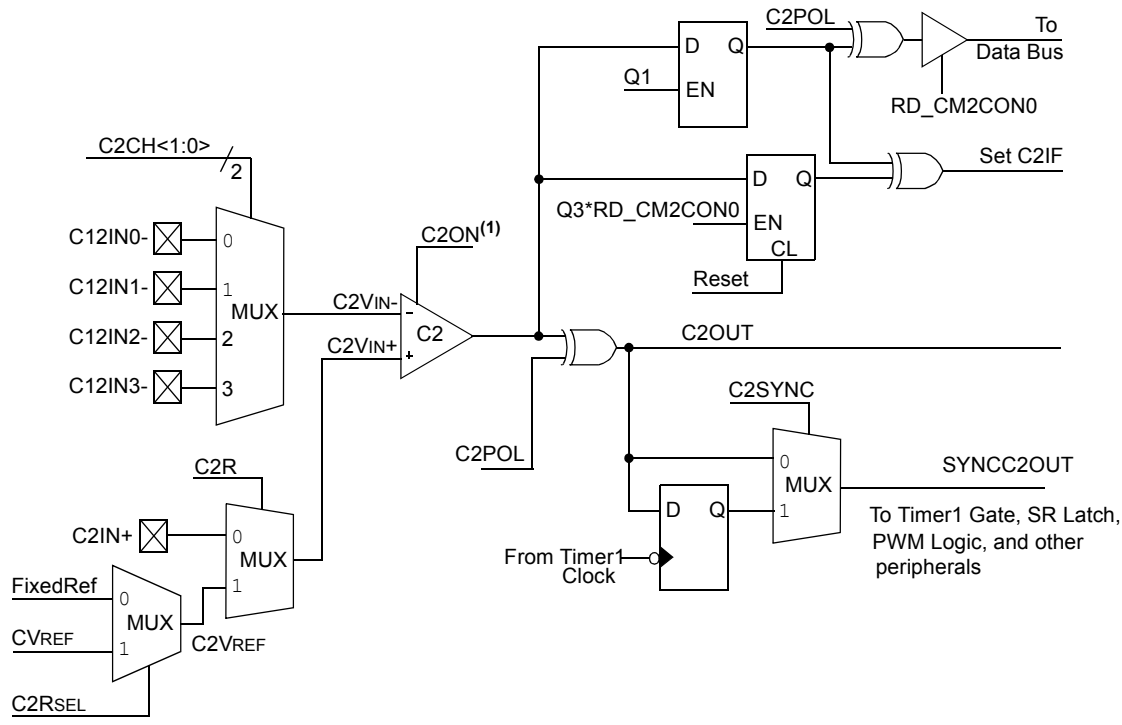

**Note:** The black areas of the output of the comparator represents the uncertainty due to input offsets and response time.

**FIGURE 8-2: COMPARATOR C1 SIMPLIFIED BLOCK DIAGRAM**




**Note 1:** When C1ON = 0, the C1 comparator will produce a '0' output to the XOR Gate.
**2:** Q1 and Q3 are phases of the four-phase system clock (FOSC).
**3:** Q1 is held high during Sleep mode.

**FIGURE 8-3: COMPARATOR C2 SIMPLIFIED BLOCK DIAGRAM**




**Note 1:** When C2ON = 0, the C2 comparator will produce a '0' output to the XOR Gate.
**2:** Q1 and Q3 are phases of the four-phase system clock (FOSC).
**3:** Q1 is held high during Sleep mode.


© 2006-2012 Microchip Technology Inc.

## 8.2 Comparator Control

Each comparator has a separate control and Configuration register: CM1CON0 for Comparator C1 and CM2CON0 for Comparator C2. In addition, Comparator C2 has a second control register, CM2CON1, for controlling the interaction with Timer1 and simultaneous reading of both comparator outputs.

The CM1CON0 and CM2CON0 registers (see Registers 8-1 and 8-2, respectively) contain the control and Status bits for the following:

- Enable
- Input selection
- Reference selection
- Output selection
- Output polarity

### 8.2.1 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.

### 8.2.2 COMPARATOR INPUT SELECTION

The CxCH<1:0> bits of the CMxCON0 register direct one of four analog input pins to the comparator inverting input.

| | |
|---|---|
| **Note:** | To use C**x**IN+ and C**x**IN- pins as analog inputs, the appropriate bits must be set in the ANSEL and ANSELH registers and the corresponding TRIS bits must also be set to disable the output drivers. |

### 8.2.3 COMPARATOR REFERENCE SELECTION

Setting the CxR bit of the CMxCON0 register directs an internal voltage reference or an analog input pin to the non-inverting input of the comparator. See **Section 8.10 "Comparator Voltage Reference"** for more information on the internal voltage reference module.

### 8.2.4 COMPARATOR OUTPUT SELECTION

The output of the comparator can be monitored by reading either the CxOUT bit of the CMxCON0 register or the MCxOUT bit of the CM2CON1 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 overrides the PORT data latch. Setting the CxON 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. |

### 8.2.5 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 8-1 shows the output state versus input conditions, including polarity control.

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

| Input Condition | CxPOL | CxOUT |
|---|---|---|
| CxVIN- > CxVIN+ | 0 | 0 |
| CxVIN- < CxVIN+ | 0 | 1 |
| CxVIN- > CxVIN+ | 1 | 1 |
| CxVIN- < CxVIN+ | 1 | 0 |

## 8.3 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 17.0 "Electrical Specifications"** for more details.

## 8.4 Comparator Interrupt Operation

The comparator interrupt flag can be set whenever there is a change in the output value of the comparator. Changes are recognized by means of a mismatch circuit which consists of two latches and an exclusive-or gate (see Figures 8-2 and 8-3). One latch is updated with the comparator output level when the CMxCON0 register is read. This latch retains the value until the next read of the CMxCON0 register or the occurrence of a Reset. The other latch of the mismatch circuit is updated on every Q1 system clock. A mismatch condition will occur when a comparator output change is clocked through the second latch on the Q1 clock cycle. At this point the two mismatch latches have opposite output levels which is detected by the exclusive-or gate and fed to the interrupt circuitry. The mismatch condition persists until either the CMxCON0 register is read or the comparator output returns to the previous state.

**Note 1:** A write operation to the CMxCON0 register will also clear the mismatch condition because all writes include a read operation at the beginning of the write cycle.

**2:** Comparator interrupts will operate correctly regardless of the state of CxOE.

The comparator interrupt is set by the mismatch edge and not the mismatch level. This means that the interrupt flag can be reset without the additional step of reading or writing the CMxCON0 register to clear the mismatch registers. When the mismatch registers are cleared, an interrupt will occur upon the comparator's return to the previous state, otherwise no interrupt will be generated.

Software will need to maintain information about the status of the comparator output, as read from the CMxCON0 register, or CM2CON1 register, to determine the actual change that has occurred.

The CxIF bit of the PIR2 register is the comparator interrupt flag. This bit must be reset in software by clearing it to '0'. Since it is also possible to write a '1' to this register, an interrupt can be generated.

The CxIE bit of the PIE2 register and the PEIE and GIE bits of the INTCON register must all be set to enable comparator interrupts. If any of these bits are cleared, the interrupt is not enabled, although the CxIF bit of the PIR2 register will still be set if an interrupt condition occurs.

**FIGURE 8-4: COMPARATOR INTERRUPT TIMING W/O CMxCON0 READ**

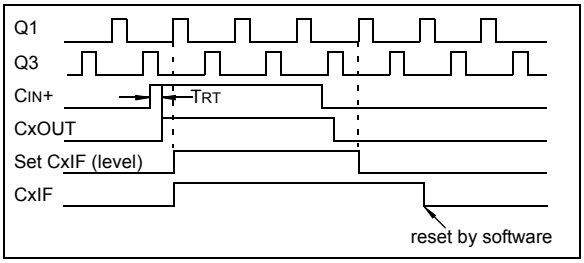
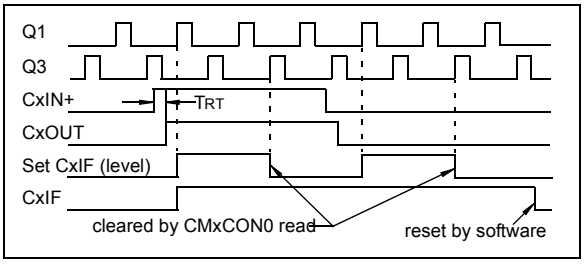



**FIGURE 8-5: COMPARATOR INTERRUPT TIMING WITH CMxCON0 READ**




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

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


© 2006-2012 Microchip Technology Inc.

## 8.5 Operation During Sleep

The comparator, if enabled before entering Sleep mode, remains active during Sleep. The additional current consumed by the comparator is shown separately in the **Section 17.0 "Electrical Specifications"**. If the comparator is not used to wake the device, power consumption can be minimized while in Sleep mode by turning off the comparator. Each comparator is turned off by clearing the CxON bit of the CMxCON0 register.

A change to the comparator output can wake-up the device from Sleep. To enable the comparator to wake the device from Sleep, the CxIE bit of the PIE2 register and the PEIE bit of the INTCON register must be set. The instruction following the Sleep instruction always executes following a wake from Sleep. If the GIE bit of the INTCON register is also set, the device will then execute the Interrupt Service Routine.

## 8.6 Effects of a Reset

A device Reset forces the CMxCON0 and CM2CON1 registers to their Reset states. This forces both comparators and the voltage references to their Off states.

## REGISTER DEFINITIONS: COMPARATOR C1

### REGISTER 8-1: CM1CON0: COMPARATOR C1 CONTROL REGISTER 0

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

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

bit 6 **C1OUT:** Comparator C1 Output bit
If C1POL = 1 (inverted polarity):
C1OUT = 0 when C1VIN+ > C1VIN-
C1OUT = 1 when C1VIN+ < C1VIN-
If C1POL = 0 (non-inverted polarity):
C1OUT = 1 when C1VIN+ > C1VIN-
C1OUT = 0 when C1VIN+ < C1VIN-

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

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

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

bit 2 **C1R:** Comparator C1 Reference Select bit (non-inverting input)
1 = C1VIN+ connects to C1VREF output
0 = C1VIN+ connects to C1IN+ pin

bit 1-0 **C1CH<1:0>:** Comparator C1 Channel Select bit
00 = C12IN0- pin of C1 connects to C1VIN-
01 = C12IN1- pin of C1 connects to C1VIN-
10 = C12IN2- pin of C1 connects to C1VIN-
11 = C12IN3- pin of C1 connects to C1VIN-

**Note 1:** Comparator output requires the following three conditions: C1OE = 1, C1ON = 1 and corresponding port TRIS bit = 0.

© 2006-2012 Microchip Technology Inc.

## REGISTER DEFINITIONS: COMPARATOR C2

**REGISTER 8-2: CM2CON0: COMPARATOR C2 CONTROL REGISTER 0**

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

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

bit 6 **C2OUT:** Comparator C2 Output bit
If C2POL = 1 (inverted polarity):
C2OUT = 0 when C2VIN+ > C2VIN-
C2OUT = 1 when C2VIN+ < C2VIN-
If C2POL = 0 (non-inverted polarity):
C2OUT = 1 when C2VIN+ > C2VIN-
C2OUT = 0 when C2VIN+ < C2VIN-

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

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

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

bit 2 **C2R:** Comparator C2 Reference Select bits (non-inverting input)
1 = C2VIN+ connects to C2VREF
0 = C2VIN+ connects to C2IN+ pin

bit 1-0 **C2CH<1:0>:** Comparator C2 Channel Select bits
00 = C12IN0- pin of C2 connects to C2VIN-
01 = C12IN1- pin of C2 connects to C2VIN-
10 = C12IN2- pin of C2 connects to C2VIN-
11 = C12IN3- pin of C2 connects to C2VIN-

**Note 1:** Comparator output requires the following three conditions: C2OE = 1, C2ON = 1 and corresponding port TRIS bit = 0.


© 2006-2012 Microchip Technology Inc.

## 8.7 Analog Input Connection Considerations

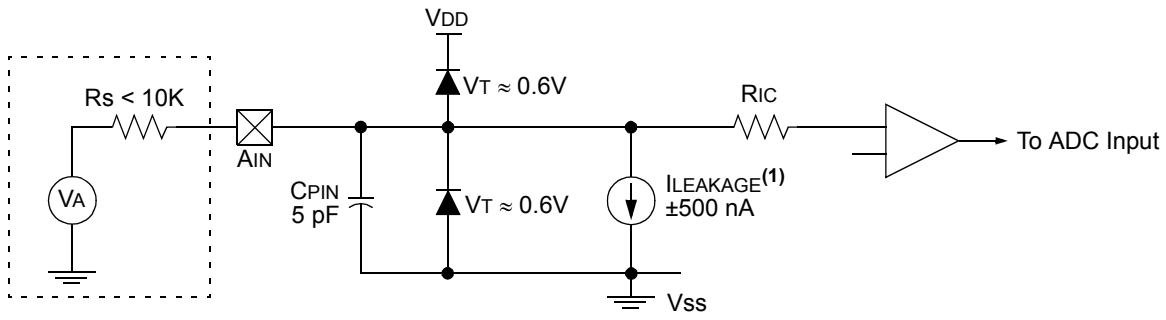
A simplified circuit for an analog input is shown in Figure 8-6. Since the analog input pins share their connection with a digital input, they have reverse biased ESD protection diodes to VDD and VSS. The analog input, therefore, must be between VSS and VDD. If the input voltage deviates from this range 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 8-6: 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 17.0 "Electrical Specifications"**.

## 8.8 Additional Comparator Features

There are three additional comparator features:

- Timer1 count enable (gate)
- Synchronizing output with Timer1
- Simultaneous read of comparator outputs

### 8.8.1 COMPARATOR C2 GATING TIMER1

This feature can be used to time the duration or interval of analog events. Clearing the T1GSS bit of the CM2CON1 register will enable Timer1 to increment based on the output of Comparator C2. This requires that Timer1 is on and gating is enabled. See **Section 6.0 "Timer1 Module with Gate Control"** for details.

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

### 8.8.2 SYNCHRONIZING COMPARATOR C2 OUTPUT TO TIMER1

The Comparator C2 output can be synchronized with Timer1 by setting the C2SYNC bit of the CM2CON1 register. When enabled, the C2 output is latched on the falling edge of the Timer1 clock source. 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 (Figures 8-2 and 8-3) and the Timer1 Block Diagram (Figure 6-1) for more information.

### 8.8.3 SIMULTANEOUS COMPARATOR OUTPUT READ

The MC1OUT and MC2OUT bits of the CM2CON1 register are mirror copies of both comparator outputs. The ability to read both outputs simultaneously from a single register eliminates the timing skew of reading separate registers.

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

**REGISTER 8-3: CM2CON1: COMPARATOR C2 CONTROL REGISTER 1**

| R-0 | R-0 | R/W-0 | R/W-0 | U-0 | U-0 | R/W-1 | R/W-0 |
|---|---|---|---|---|---|---|---|
| MC1OUT | MC2OUT | C1RSEL | C2RSEL | — | — | T1GSS | C2SYNC |
| 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 **MC1OUT:** Mirror Copy of C1OUT bit

bit 6 **MC2OUT:** Mirror Copy of C2OUT bit

bit 5 **C1RSEL:** Comparator C1 Reference Select bit
1 = CVREF routed to C1VREF input of Comparator C1
0 = Absolute voltage reference (0.6) routed to C1VREF input of Comparator C1 (or 1.2V precision reference on parts so equipped)

bit 4 **C2RSEL:** Comparator C2 Reference Select bit
1 = CVREF routed to C2VREF input of Comparator C2
0 = Absolute voltage reference (0.6) routed to C2VREF input of Comparator C2 (or 1.2V precision reference on parts so equipped)

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

bit 1 **T1GSS:** Timer1 Gate Source Select bit
1 = Timer1 gate source is $\overline{T1G}$
0 = Timer1 gate source is SYNCC2OUT.

bit 0 **C2SYNC:** Comparator C2 Output Synchronization bit
1 = Output is synchronous to falling edge of Timer1 clock
0 = Output is asynchronous


© 2006-2012 Microchip Technology Inc.

## 8.9 Comparator SR Latch

The SR latch module provides additional control of the comparator outputs. The module consists of a single SR latch and output multiplexers. The SR latch can be set, reset or toggled by the comparator outputs. The SR latch may also be set or reset, independent of comparator output, by control bits in the SRCON control register. The SR latch output multiplexers select whether the latch outputs or the comparator outputs are directed to the I/O port logic for eventual output to a pin.

### 8.9.1 LATCH OPERATION

The latch is a Set-Reset latch that does not depend on a clock source. Each of the Set and Reset inputs are active-high. Each latch input is connected to a comparator output and a software controlled pulse generator. The latch can be set by C1OUT or the PULSS bit of the SRCON register. The latch can be reset by C2OUT or the PULSR bit of the SRCON register. The latch is reset-dominant, therefore, if both Set and Reset inputs are high the latch will go to the Reset state. Both the PULSS and PULSR bits are self resetting which means that a single write to either of the bits is all that is necessary to complete a latch set or Reset operation.

### 8.9.2 LATCH OUTPUT

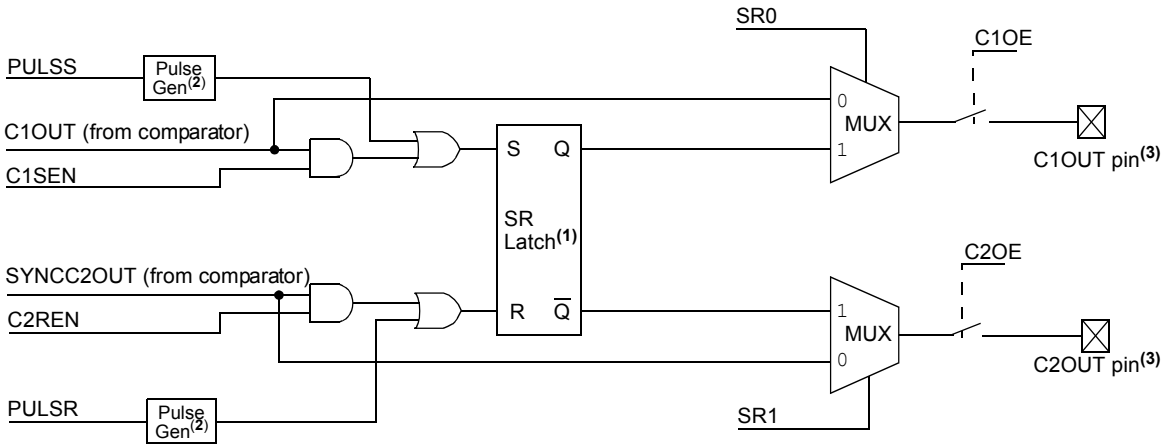
The SR<1:0> bits of the SRCON register control the latch output multiplexers and determine four possible output configurations. In these four configurations, the CxOUT I/O port logic is connected to:

- C1OUT and C2OUT
- C1OUT and SR latch $\overline{Q}$
- C2OUT and SR latch Q
- SR latch Q and $\overline{Q}$

After any Reset, the default output configuration is the unlatched C1OUT and C2OUT mode. This maintains compatibility with devices that do not have the SR latch feature.

The applicable TRIS bits of the corresponding ports must be cleared to enable the port pin output drivers. Additionally, the CxOE comparator output enable bits of the CMxCON0 registers must be set in order to make the comparator or latch outputs available on the output pins. The latch configuration enable states are completely independent of the enable states for the comparators.

**FIGURE 8-7: SR LATCH SIMPLIFIED BLOCK DIAGRAM**




**Note 1:** If R = 1 and S = 1 simultaneously, Q = 0, $\overline{Q}$ = 1
**2:** Pulse generator causes a 1/2 Q-state (1 Tosc) pulse width.
**3:** Output shown for reference only. See I/O port pin block diagram for more detail.

## REGISTER DEFINITIONS: SR LATCH

**REGISTER 8-4: SRCON: SR LATCH CONTROL REGISTER**

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

| **Legend:** | | S = Bit is set only - | |
|---|---|---|---|
| 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 **SR1:** SR Latch Configuration bit(2)
- 1 = C2OUT pin is the latch $\overline{Q}$ output
- 0 = C2OUT pin is the C2 comparator output

bit 6 **SR0:** SR Latch Configuration bits(2)
- 1 = C1OUT pin is the latch Q output
- 0 = C1OUT pin is the C1 Comparator output

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

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

bit 3 **PULSS:** Pulse the SET Input of the SR Latch bit
- 1 = Triggers pulse generator to set SR latch. Bit is immediately reset by hardware.
- 0 = Does not trigger pulse generator

bit 2 **PULSR:** Pulse the Reset Input of the SR Latch bit
- 1 = Triggers pulse generator to reset SR latch. Bit is immediately reset by hardware.
- 0 = Does not trigger pulse generator

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

bit 0 **FVREN:** Fixed Voltage Reference Enable bit
- 1 = 0.6V Reference FROM INTOSC LDO is enabled
- 0 = 0.6V Reference FROM INTOSC LDO is disabled

**Note 1:** The CxOUT bit in the CMxCON0 register will always reflect the actual comparator output (not the level on the pin), regardless of the SR latch operation.

**2:** To enable an SR Latch output to the pin, the appropriate CxOE and TRIS bits must be properly configured.


© 2006-2012 Microchip Technology Inc.

## 8.10 Comparator Voltage Reference

The comparator voltage reference module provides an internally generated voltage reference for the comparators. The following features are available:

- Independent from Comparator operation
- Two 16-level voltage ranges
- Output clamped to VSS
- Ratiometric with VDD
- Fixed Reference (0.6V)

The VRCON register (Register 8-5) controls the voltage reference module shown in Figure 8-8.

The voltage source is selectable through both ends of the 16 connection resistor ladder network. Bit VRSS of the VRCON register selects either the internal or external voltage source.

The PIC16F882/883/884/886/887 allows the CVREF signal to be output to the RA2 pin of PORTA under certain configurations only. For more details, see Figure 8-9.

### 8.10.1 INDEPENDENT OPERATION

The comparator voltage reference is independent of the comparator configuration. Setting the VREN bit of the VRCON register will enable the voltage reference.

### 8.10.2 OUTPUT VOLTAGE SELECTION

The CVREF voltage reference has 2 ranges with 16 voltage levels in each range. Range selection is controlled by the VRR bit of the VRCON register. The 16 levels are set with the VR<3:0> bits of the VRCON register.

The CVREF output voltage is determined by the following equations:

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

$$VRR = 1 \text{ (low range):}$$
$$CVREF = (VR<3:0>/24) \times VLADDER$$
$$VRR = 0 \text{ (high range):}$$
$$CVREF = (VLADDER/4) + (VR<3:0> \times VLADDER/32)$$
$$VLADDER = VDD \text{ or } ([VREF+] - [VREF-]) \text{ or } VREF+$$

The full range of VSS to VDD cannot be realized due to the construction of the module. See Figure 8-8.

### 8.10.3 OUTPUT CLAMPED TO VSS

The CVREF output voltage can be set to Vss with no power consumption by clearing the FVREN bit of the VRCON register.

This allows the comparator to detect a zero-crossing while not consuming additional CVREF module current.

| | |
|---|---|
| **Note:** | Depending on the application, additional components may be required for a zero cross circuit. Reference TB3013, *"Using the ESD Parasitic Diodes on Mixed Signal Microcontrollers"* (DS93013), for more information. |

### 8.10.4 OUTPUT RATIOMETRIC TO VDD

The comparator voltage reference is VDD derived and therefore, the CVREF output changes with fluctuations in VDD. The tested absolute accuracy of the Comparator Voltage Reference can be found in **Section 17.0 "Electrical Specifications"**.

### 8.10.5 FIXED VOLTAGE REFERENCE

The fixed voltage reference is independent of VDD, with a nominal output voltage of 0.6V. This reference can be enabled by setting the FVREN bit of the SRCON register to '1'. This reference is always enabled when the HFINTOSC oscillator is active.

### 8.10.6 FIXED VOLTAGE REFERENCE STABILIZATION PERIOD

When the fixed voltage reference module is enabled, it will require some time for the reference and its amplifier circuits to stabilize. The user program must include a small delay routine to allow the module to settle. See **Section 17.0 "Electrical Specifications"** for the minimum delay requirement.

### 8.10.7 VOLTAGE REFERENCE SELECTION

Multiplexers on the output of the voltage reference module enable selection of either the CVREF or fixed voltage reference for use by the comparators.

Setting the C1RSEL bit of the CM2CON1 register enables current to flow in the CVREF voltage divider and selects the CVREF voltage for use by C1. Clearing the C1RSEL bit selects the fixed voltage for use by C1.

Setting the C2RSEL bit of the CM2CON1 register enables current to flow in the CVREF voltage divider and selects the CVREF voltage for use by C2. Clearing the C2RSEL bit selects the fixed voltage for use by C2.

When both the C1RSEL and C2RSEL bits are cleared, current flow in the CVREF voltage divider is disabled minimizing the power drain of the voltage reference peripheral.

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

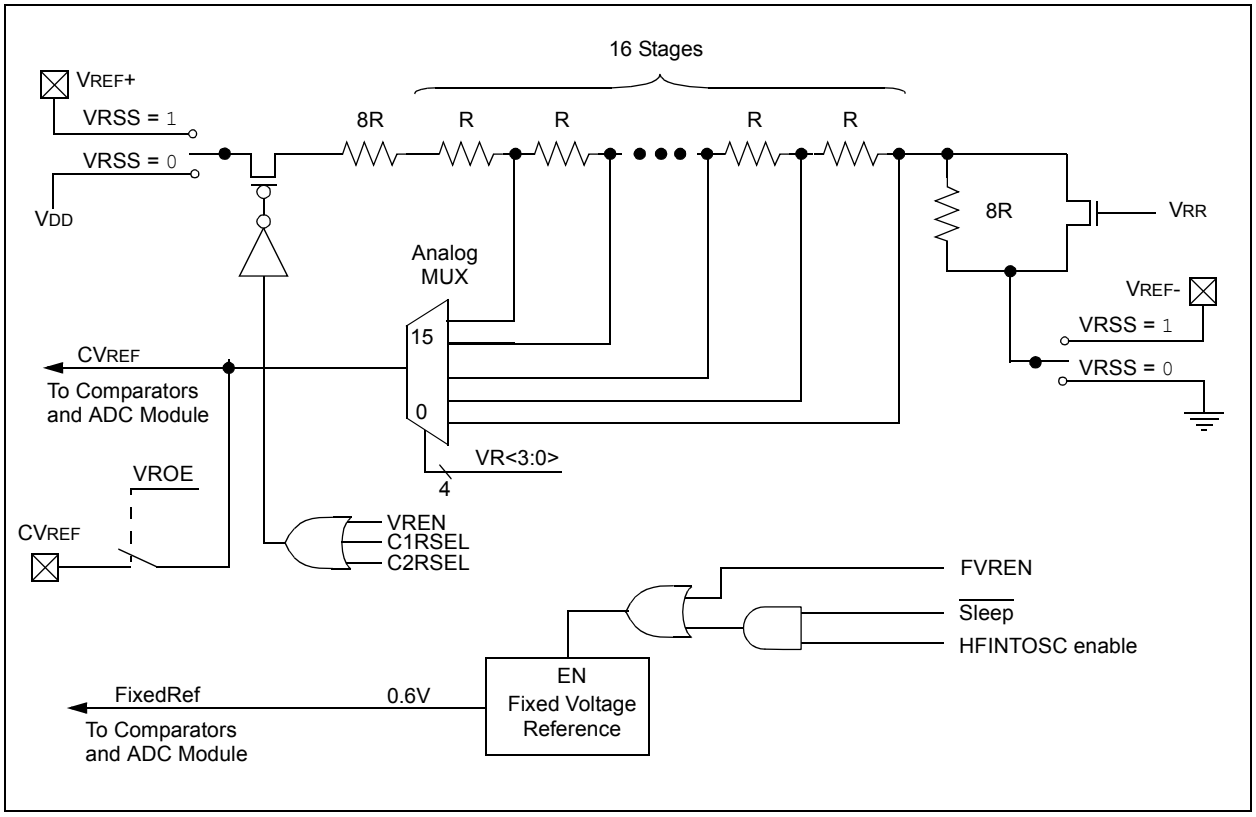
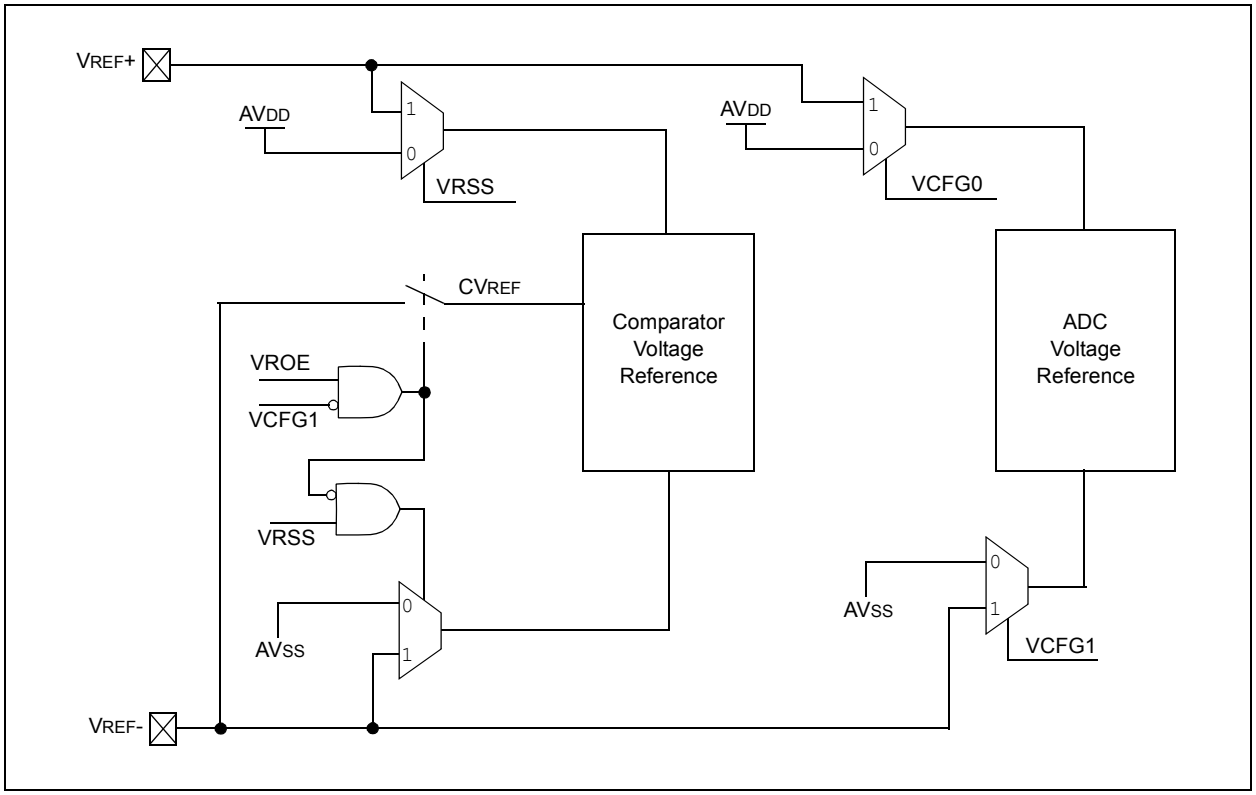



**FIGURE 8-9: COMPARATOR AND ADC VOLTAGE REFERENCE BLOCK DIAGRAM**





© 2006-2012 Microchip Technology Inc.

**TABLE 8-2: COMPARATOR AND ADC VOLTAGE REFERENCE PRIORITY**

| RA3 | RA2 | Comp. Reference (+) | Comp. Reference (-) | ADC Reference (+) | ADC Reference (-) | CFG1 | CFG0 | VRSS | VROE |
|---|---|---|---|---|---|---|---|---|---|
| I/O | I/O | AVDD | AVSS | AVDD | AVSS | 0 | 0 | 0 | 0 |
| I/O | CVREF | AVDD | AVSS | AVDD | AVSS | 0 | 0 | 0 | 1 |
| VREF+ | VREF- | VREF+ | VREF- | AVDD | AVSS | 0 | 0 | 1 | 0 |
| VREF+ | CVREF | VREF+ | AVSS | AVDD | AVSS | 0 | 0 | 1 | 1 |
| VREF+ | I/O | AVDD | AVSS | VREF+ | AVSS | 0 | 1 | 0 | 0 |
| VREF+ | CVREF | AVDD | AVSS | VREF+ | AVSS | 0 | 1 | 0 | 1 |
| VREF+ | VREF- | VREF+ | VREF- | VREF+ | AVSS | 0 | 1 | 1 | 0 |
| VREF+ | CVREF | VREF+ | AVSS | VREF+ | AVSS | 0 | 1 | 1 | 1 |
| I/O | VREF- | AVDD | AVSS | AVDD | VREF- | 1 | 0 | 0 | 0 |
| I/O | VREF- | AVDD | AVSS | AVDD | VREF- | 1 | 0 | 0 | 1 |
| VREF+ | VREF- | VREF+ | VREF- | AVDD | VREF- | 1 | 0 | 1 | 0 |
| VREF+ | VREF- | VREF+ | VREF- | AVDD | VREF- | 1 | 0 | 1 | 1 |
| VREF+ | VREF- | AVDD | AVSS | VREF+ | VREF- | 1 | 1 | 0 | 0 |
| VREF+ | VREF- | AVDD | AVSS | VREF+ | VREF- | 1 | 1 | 0 | 1 |
| VREF+ | VREF- | VREF+ | VREF- | VREF+ | VREF- | 1 | 1 | 1 | 0 |
| VREF+ | VREF- | VREF+ | VREF- | VREF+ | VREF- | 1 | 1 | 1 | 1 |

## REGISTER DEFINITIONS: VOLTAGE REFERENCE CONTROL

### REGISTER 8-5: VRCON: VOLTAGE REFERENCE 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 |
|---|---|---|---|---|---|---|---|
| VREN | VROE | VRR | VRSS | VR3 | VR2 | VR1 | VR0 |
| 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 **VREN:** Comparator C1 Voltage Reference Enable bit
1 = CVREF circuit powered on
0 = CVREF circuit powered down

bit 6 **VROE:** Comparator C2 Voltage Reference Enable bit
1 = CVREF voltage level is also output on the RA2/AN2/VREF-/CVREF/C2IN+ pin
0 = CVREF voltage is disconnected from the RA2/AN2/VREF-/CVREF/C2IN+ pin

bit 5 **VRR:** CVREF Range Selection bit
1 = Low range
0 = High range

bit 4 **VRSS:** Comparator VREF Range Selection bit
1 = Comparator Reference Source, CVRSRC = (VREF+) - (VREF-)
0 = Comparator Reference Source, CVRSRC = VDD - VSS

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

### TABLE 8-3: SUMMARY OF REGISTERS ASSOCIATED WITH THE COMPARATOR AND VOLTAGE REFERENCE MODULES

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|
| ANSEL | ANS7 | ANS6 | ANS5 | ANS4 | ANS3 | ANS2 | ANS1 | ANS0 | 42 |
| ANSELH | — | — | ANS13 | ANS12 | ANS11 | ANS10 | ANS9 | ANS8 | 50 |
| CM1CON0 | C1ON | C1OUT | C1OE | C1POL | — | C1R | C1CH1 | C1CH0 | 93 |
| CM2CON0 | C2ON | C2OUT | C2OE | C2POL | — | C2R | C2CH1 | C2CH0 | 94 |
| CM2CON1 | MC1OUT | MC2OUT | C1RSEL | C2RSEL | — | — | T1GSS | C2SYNC | 96 |
| INTCON | GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | RBIF | 33 |
| PIE2 | OSFIE | C2IE | C1IE | EEIE | BCLIE | ULPWUIE | — | CCP2IE | 35 |
| PIR2 | OSFIF | C2IF | C1IF | EEIF | BCLIF | ULPWUIF | — | CCP2IF | 37 |
| PORTA | RA7 | RA6 | RA5 | RA4 | RA3 | RA2 | RA1 | RA0 | 41 |
| PORTB | RB7 | RB6 | RB5 | RB4 | RB3 | RB2 | RB1 | RB0 | 50 |
| SRCON | SR1 | SR0 | C1SEN | C2SEN | PULSS | PULSR | — | FVREN | 98 |
| TRISA | TRISA7 | TRISA6 | TRISA5 | TRISA4 | TRISA3 | TRISA2 | TRISA1 | TRISA0 | 41 |
| TRISB | TRISB7 | TRISB6 | TRISB5 | TRISB4 | TRISB3 | TRISB2 | TRISB1 | TRISB0 | 50 |
| VRCON | VREN | VROE | VRR | VRSS | VR3 | VR2 | VR1 | VR0 | 102 |

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


© 2006-2012 Microchip Technology Inc.

## 9.0 ANALOG-TO-DIGITAL CONVERTER (ADC) MODULE

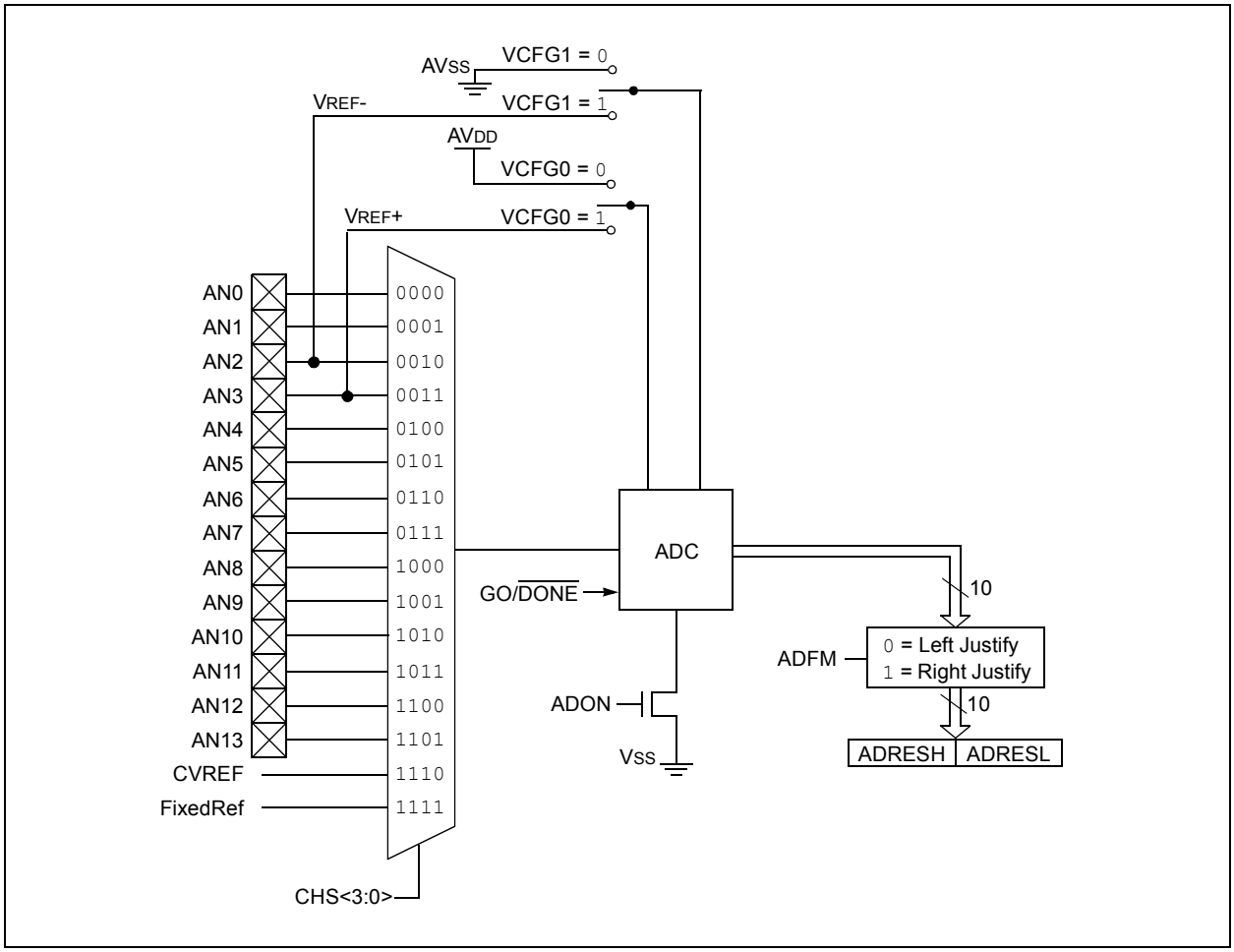
The Analog-to-Digital Converter (ADC) allows conversion of an analog input signal to a 10-bit binary representation of that signal. This device uses analog inputs, which are multiplexed into a single sample and hold circuit. The output of the sample and hold is connected to the input of the converter. 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 be either internally generated or externally supplied.

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

Figure 9-1 shows the block diagram of the ADC.

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



## 9.1 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

### 9.1.1 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. |

### 9.1.2 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 9.2 "ADC Operation"** for more information.

### 9.1.3 ADC VOLTAGE REFERENCE

The VCFG bits of the ADCON1 register provide independent control of the positive and negative voltage references. The positive voltage reference can be either VDD or an external voltage source. Likewise, the negative voltage reference can be either VSS or an external voltage source.

### 9.1.4 CONVERSION CLOCK

The source of the conversion clock is software selectable via the ADCS bits of the ADCON0 register. There are four possible clock options:

- FOSC/2
- FOSC/8
- FOSC/32
- 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 9-2.

For correct conversion, the appropriate TAD specification must be met. See A/D conversion requirements in **Section 17.0 "Electrical Specifications"** for more information. Table 9-1 gives examples of appropriate ADC clock selections.

| | |
|---|---|
| **Note:** | Unless using the FRC, any changes in the system clock frequency will change the ADC clock frequency, which may adversely affect the ADC result. |


© 2006-2012 Microchip Technology Inc.

**TABLE 9-1: ADC CLOCK PERIOD (TAD) VS. DEVICE OPERATING FREQUENCIES (VDD ≥ 3.0V)**

| ADC Clock Period (TAD) | | Device Frequency (FOSC) | | | |
|---|---|---|---|---|---|
| **ADC Clock Source** | **ADCS<1:0>** | **20 MHz** | **8 MHz** | **4 MHz** | **1 MHz** |
| FOSC/2 | 00 | 100 ns(2) | 250 ns(2) | 500 ns(2) | 2.0 μs |
| FOSC/8 | 01 | 400 ns(2) | 1.0 μs(2) | 2.0 μs | 8.0 μs(3) |
| FOSC/32 | 10 | 1.6 μs | 4.0 μs | 8.0 μs(3) | 32.0 μs(3) |
| FRC | 11 | 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.

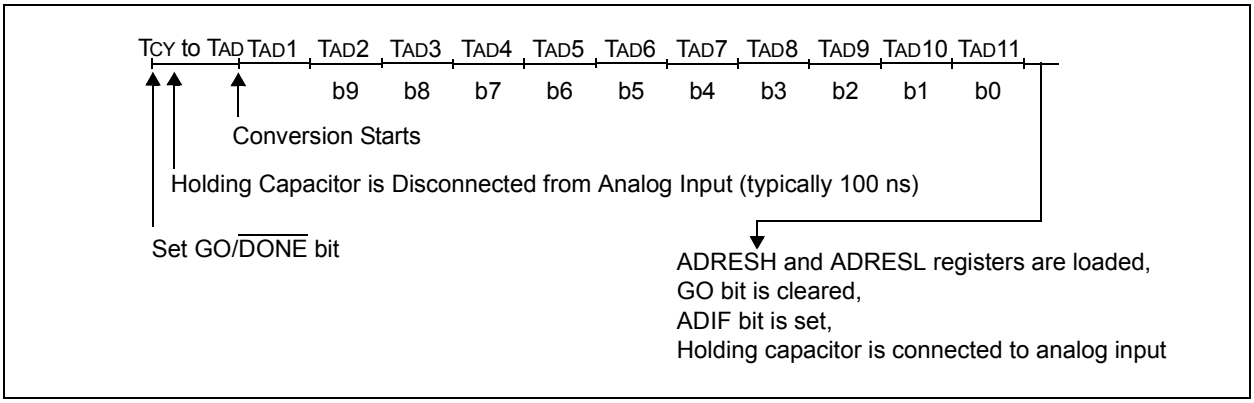
**Note 1:** The FRC source has a typical TAD time of 4 μs for VDD > 3.0V.

**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 9-2: ANALOG-TO-DIGITAL CONVERSION TAD CYCLES**




### 9.1.5 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.

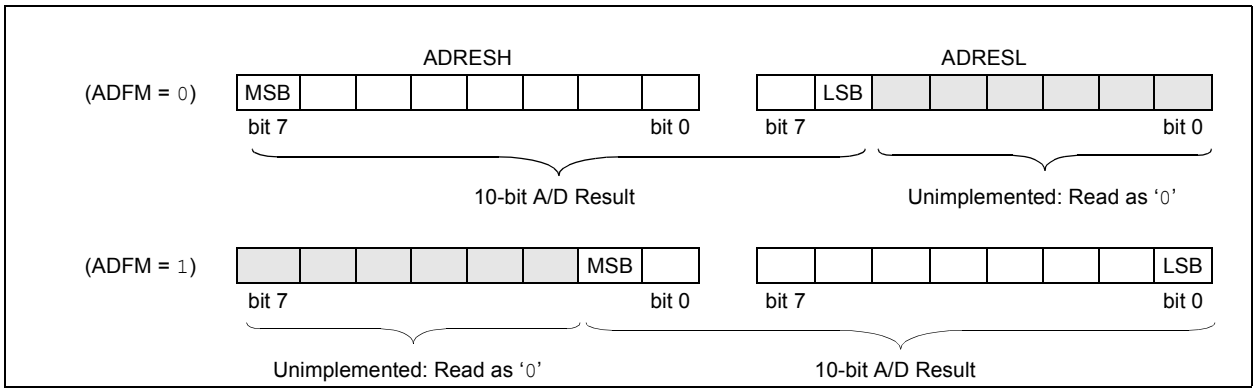
Please see **Section 14.3 "Interrupts"** for more information.

### 9.1.6 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 9-3 shows the two output formats.

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




## 9.2 ADC Operation

### 9.2.1 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 9.2.6 "A/D Conversion Procedure"**. |
|---|---|

### 9.2.2 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

### 9.2.3 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. |
|---|---|

### 9.2.4 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.

### 9.2.5 SPECIAL EVENT TRIGGER

The ECCP 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 11.0 "Capture/Compare/PWM Modules (CCP1 and CCP2)"** for more information.


© 2006-2012 Microchip Technology Inc.

### 9.2.6 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 9.3 "A/D Acquisition Requirements"**.

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

```
;This code block configures the ADC
;for polling, Vdd and Vss as reference, Frc
clock and AN0 input.
;
;Conversion start & polling for completion
; are included.
;
BANKSEL   ADCON1       ;
MOVLW     B'10000000'  ;right justify
MOVWF     ADCON1       ;Vdd and Vss as Vref
BANKSEL   TRISA        ;
BSF       TRISA,0      ;Set RA0 to input
BANKSEL   ANSEL        ;
BSF       ANSEL,0      ;Set RA0 to analog
BANKSEL   ADCON0       ;
MOVLW     B'11000001'  ;ADC Frc clock,
MOVWF     ADCON0       ;AN0, On
CALL      SampleTime   ;Acquisiton delay
BSF       ADCON0,GO    ;Start conversion
BTFSC     ADCON0,GO    ;Is conversion done?
GOTO      $-1          ;No, test again
BANKSEL   ADRESH       ;
MOVF      ADRESH,W     ;Read upper 2 bits
MOVWF     RESULTHI     ;store in GPR space
BANKSEL   ADRESL       ;
MOVF      ADRESL,W     ;Read lower 8 bits
MOVWF     RESULTLO     ;Store in GPR space
```

### 9.2.7 ADC REGISTER DEFINITIONS

The following registers are used to control the operation of the ADC.

**Note:** For ANSEL and ANSELH registers, see Register 3-3 and Register 3-4, respectively.

## REGISTER DEFINITIONS: ADC CONTROL

**REGISTER 9-1: ADCON0: A/D CONTROL REGISTER 0**

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| ADCS1 | ADCS0 | CHS3 | CHS2 | CHS1 | CHS0 | 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 7-6 **ADCS<1:0>:** A/D Conversion Clock Select bits
00 = FOSC/2
01 = FOSC/8
10 = FOSC/32
11 = FRC (clock derived from a dedicated internal oscillator = 500 kHz max)

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
1000 = AN8
1001 = AN9
1010 = AN10
1011 = AN11
1100 = AN12
1101 = AN13
1110 = CVREF
1111 = Fixed Ref (0.6V 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


© 2006-2012 Microchip Technology Inc.

**REGISTER 9-2: ADCON1: A/D CONTROL REGISTER 1**

| R/W-0 | U-0 | R/W-0 | R/W-0 | U-0 | U-0 | U-0 | U-0 |
|---|---|---|---|---|---|---|---|
| ADFM | — | VCFG1 | VCFG0 | — | — | — | — |
| 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 **VCFG1:** Voltage Reference bit
1 = $V_{REF-}$ pin
0 = $V_{SS}$

bit 4 **VCFG0:** Voltage Reference bit
1 = $V_{REF+}$ pin
0 = $V_{DD}$

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

**REGISTER 9-3: ADRESH: ADC RESULT REGISTER HIGH (ADRESH) ADFM = 0**

| R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x |
|---|---|---|---|---|---|---|---|
| ADRES9 | ADRES8 | ADRES7 | ADRES6 | ADRES5 | ADRES4 | ADRES3 | ADRES2 |
| 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 **ADRES<9:2>**: ADC Result Register bits
Upper 8 bits of 10-bit conversion result

**REGISTER 9-4: ADRESL: ADC RESULT REGISTER LOW (ADRESL) ADFM = 0**

| R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x |
|---|---|---|---|---|---|---|---|
| ADRES1 | ADRES0 | — | — | — | — | — | — |
| 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 **ADRES<1:0>**: ADC Result Register bits
Lower 2 bits of 10-bit conversion result

bit 5-0 **Reserved**: Do not use.

**REGISTER 9-5: ADRESH: ADC RESULT REGISTER HIGH (ADRESH) ADFM = 1**

| R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x |
|---|---|---|---|---|---|---|---|
| — | — | — | — | — | — | ADRES9 | ADRES8 |
| 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 **Reserved**: Do not use.

bit 1-0 **ADRES<9:8>**: ADC Result Register bits
Upper 2 bits of 10-bit conversion result

**REGISTER 9-6: ADRESL: ADC RESULT REGISTER LOW (ADRESL) ADFM = 1**

| R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x |
|---|---|---|---|---|---|---|---|
| ADRES7 | ADRES6 | ADRES5 | ADRES4 | ADRES3 | ADRES2 | ADRES1 | ADRES0 |
| 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 **ADRES<7:0>**: ADC Result Register bits
Lower 8 bits of 10-bit conversion result


© 2006-2012 Microchip Technology Inc.

## 9.3 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 9-4. 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 9-4. 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 9-1 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 9-1: 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:*

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

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

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

*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:*

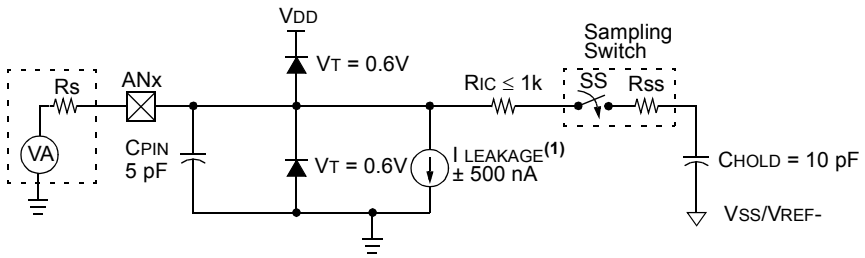
$$\begin{aligned} T_{ACQ} &= 2MS + 1.37MS + [(50°C - 25°C)(0.05MS/°C)] \\ &= 4.67MS \end{aligned}$$

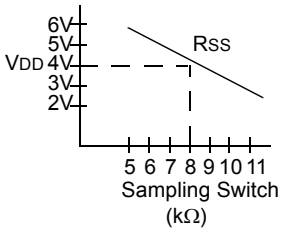
**Note 1:** The reference voltage (VREF) has no effect on the equation, since it cancels itself out.

**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 9-4: ANALOG INPUT MODEL**

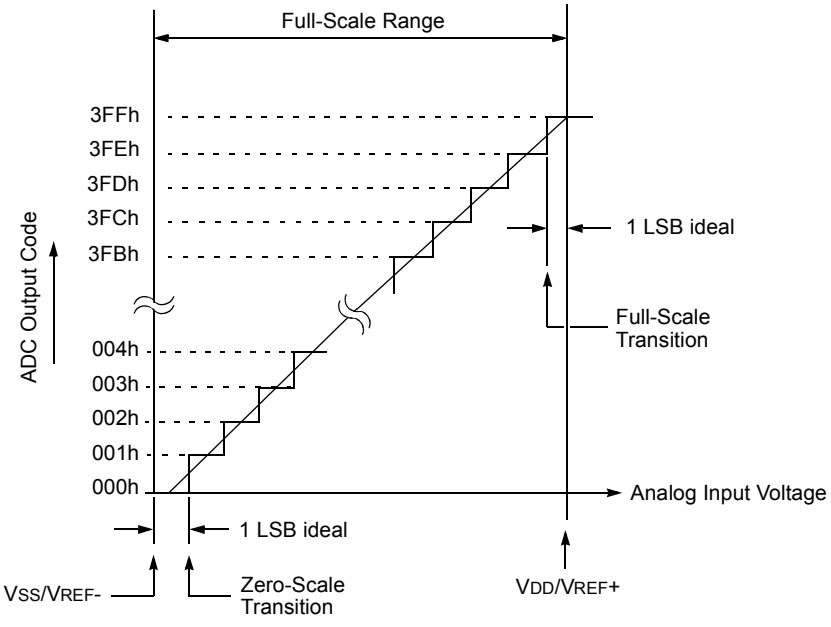



**Legend:** CPIN = Input Capacitance
VT = Threshold Voltage
I LEAKAGE = Leakage current at the pin due to various junctions
RIC = Interconnect Resistance
SS = Sampling Switch
CHOLD = Sample/Hold Capacitance

**Note 1:** See **Section 17.0 "Electrical Specifications"**.

**FIGURE 9-5: ADC TRANSFER FUNCTION**





© 2006-2012 Microchip Technology Inc.

**TABLE 9-2: 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 | ADCS1 | ADCS0 | CHS3 | CHS2 | CHS1 | CHS0 | GO/$\overline{\text{DONE}}$ | ADON | 108 |
| ADCON1 | ADFM | — | VCFG1 | VCFG0 | — | — | — | — | 109 |
| ANSEL | ANS7 | ANS6 | ANS5 | ANS4 | ANS3 | ANS2 | ANS1 | ANS0 | 42 |
| ANSELH | — | — | ANS13 | ANS12 | ANS11 | ANS10 | ANS9 | ANS8 | 50 |
| ADRESH | A/D Result Register High Byte | | | | | | | | 110 |
| ADRESL | A/D Result Register Low Byte | | | | | | | | 110 |
| INTCON | GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | RBIF | 33 |
| PIE1 | — | ADIE | RCIE | TXIE | SSPIE | CCP1IE | TMR2IE | TMR1IE | 34 |
| PIR1 | — | ADIF | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | 36 |
| PORTA | RA7 | RA6 | RA5 | RA4 | RA3 | RA2 | RA1 | RA0 | 41 |
| PORTB | RB7 | RB6 | RB5 | RB4 | RB3 | RB2 | RB1 | RB0 | 50 |
| PORTE | — | — | — | — | RE3 | RE2 | RE1 | RE0 | 61 |
| TRISA | TRISA7 | TRISA6 | TRISA5 | TRISA4 | TRISA3 | TRISA2 | TRISA1 | TRISA0 | 41 |
| TRISB | TRISB7 | TRISB6 | TRISB5 | TRISB4 | TRISB3 | TRISB2 | TRISB1 | TRISB0 | 50 |
| TRISE | — | — | — | — | TRISE3 | TRISE2 | TRISE1 | TRISE0 | 61 |

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

**NOTES:**


© 2006-2012 Microchip Technology Inc.

## 10.0 DATA EEPROM AND FLASH PROGRAM MEMORY CONTROL

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

- EECON1
- EECON2
- EEDAT
- EEDATH
- EEADR
- EEADRH (bit 4 on PIC16F886/PIC16F887 only)

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

When accessing the program memory block of the PIC16F886/PIC16F887 devices, the EEDAT and EEDATH registers form a 2-byte word that holds the 14-bit data for read/write, and the EEADR and EEADRH registers form a 2-byte word that holds the 12-bit address of the EEPROM location being read. The PIC16F882 devices have 2K words of program EEPROM with an address range from 0h to 07FFh. The PIC16F883/PIC16F884 devices have 4K words of program EEPROM with an address range from 0h to 0FFFh. The program memory allows one-word reads.

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

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

Depending on the setting of the Flash Program Memory Self Write Enable bits WRT<1:0> of the Configuration Word Register 2, the device may or may not be able to write certain blocks of the program memory. However, reads from the program memory are allowed.

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

### 10.1 EEADR and EEADRH Registers

The EEADR and EEADRH registers can address up to a maximum of 256 bytes of data EEPROM or up to a maximum of 8K words of program EEPROM.

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

#### 10.1.1 EECON1 AND EECON2 REGISTERS

EECON1 is the control register for EE memory accesses.

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

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

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

Interrupt flag bit EEIF of the PIR2 register is set when write is complete. It must be cleared in the software.

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

## REGISTER DEFINITIONS: DATA EEPROM CONTROL

### REGISTER 10-1: EEDAT: EEPROM DATA 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 |
|---|---|---|---|---|---|---|---|
| EEDAT7 | EEDAT6 | EEDAT5 | EEDAT4 | EEDAT3 | EEDAT2 | EEDAT1 | EEDAT0 |
| 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 **EEDAT<7:0>**: 8 Least Significant Address bits to Write to or Read from data EEPROM or Read from program memory

### REGISTER 10-2: EEADR: EEPROM ADDRESS 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 |
|---|---|---|---|---|---|---|---|
| EEADR7 | EEADR6 | EEADR5 | EEADR4 | EEADR3 | EEADR2 | EEADR1 | EEADR0 |
| 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 **EEADR<7:0>**: 8 Least Significant Address bits for EEPROM Read/Write Operation(1) or Read from program memory

### REGISTER 10-3: EEDATH: EEPROM 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 |
|---|---|---|---|---|---|---|---|
| — | — | EEDATH5 | EEDATH4 | EEDATH3 | EEDATH2 | EEDATH1 | EEDATH0 |
| 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 **EEDATH<5:0>**: 6 Most Significant Data bits from program memory

### REGISTER 10-4: EEADRH: EEPROM ADDRESS HIGH BYTE REGISTER

| U-0 | U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | — | — | EEADRH4(1) | EEADRH3 | EEADRH2 | EEADRH1 | EEADRH0 |
| 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 **EEADRH<4:0>**: Specifies the 4 Most Significant Address bits or high bits for program memory reads

**Note 1:** PIC16F886/PIC16F887 only.


© 2006-2012 Microchip Technology Inc.

**REGISTER 10-5: EECON1: EEPROM CONTROL REGISTER**

| R/W-x | U-0 | U-0 | U-0 | R/W-x | R/W-0 | R/S-0 | R/S-0 |
|---|---|---|---|---|---|---|---|
| EEPGD | — | — | — | WRERR | WREN | WR | RD |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | | |
|---|---|---|---|
| S = Bit can only be set | | | |
| 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 **EEPGD:** Program/Data EEPROM Select bit
- 1 = Accesses program memory
- 0 = Accesses data memory

bit 6-4 **Unimplemented:** Read as '0'

bit 3 **WRERR:** EEPROM Error Flag bit
- 1 = A write operation is prematurely terminated (any $\overline{\text{MCLR}}$ Reset, any WDT Reset during normal operation or BOR Reset)
- 0 = The write operation completed

bit 2 **WREN:** EEPROM Write Enable bit
- 1 = Allows write cycles
- 0 = Inhibits write to the data EEPROM

bit 1 **WR:** Write Control bit
- 1 = Initiates a write cycle (The bit is cleared by hardware once write is complete. The WR bit can only be set, not cleared, in software.)
- 0 = Write cycle to the data EEPROM is complete

bit 0 **RD:** Read Control bit
- 1 = Initiates a memory read (the RD is cleared in hardware and can only be set, not cleared, in software.)
- 0 = Does not initiate a memory read

### 10.1.2 READING THE DATA EEPROM MEMORY

To read a data memory location, the user must write the address to the EEADR register, clear the EEPGD control bit of the EECON1 register, and then set control bit RD. The data is available at the very next cycle, in the EEDAT register; therefore, it can be read in the next instruction. EEDAT will hold this value until another read or until it is written to by the user (during a write operation).

**EXAMPLE 10-1: DATA EEPROM READ**

```
BANKSEL  EEADR          ;
MOVLW    DATA_EE_ADDR   ;
MOVWF    EEADR          ;Data Memory
                        ;Address to read
BANKSEL  EECON1         ;
BCF      EECON1, EEPGD  ;Point to DATA memory
BSF      EECON1, RD     ;EE Read
BANKSEL  EEDAT          ;
MOVF     EEDAT, W       ;W = EEDAT
BCF      STATUS, RP1    ;Bank 0
```

### 10.1.3 WRITING TO THE DATA EEPROM MEMORY

To write an EEPROM data location, the user must first write the address to the EEADR register and the data to the EEDAT register. Then the user must follow a specific sequence to initiate the write for each byte.

The write will not initiate if the above sequence is not followed exactly (write 55h to EECON2, write AAh to EECON2, then set WR bit) for each byte. Interrupts should be disabled during this code segment.

Additionally, the WREN bit in EECON1 must be set to enable write. This mechanism prevents accidental writes to data EEPROM due to errant (unexpected) code execution (i.e., lost programs). The user should keep the WREN bit clear at all times, except when updating EEPROM. The WREN bit is not cleared by hardware.

After a write sequence has been initiated, clearing the WREN bit will not affect this write cycle. The WR bit will be inhibited from being set unless the WREN bit is set.

At the completion of the write cycle, the WR bit is cleared in hardware and the EE Write Complete Interrupt Flag bit (EEIF) is set. The user can either enable this interrupt or poll this bit. EEIF must be cleared by software.

**EXAMPLE 10-2: DATA EEPROM WRITE**

```
        BANKSEL  EEADR          ;
        MOVLW    DATA_EE_ADDR   ;
        MOVWF    EEADR          ;Data Memory Address to write
        MOVLW    DATA_EE_DATA   ;
        MOVWF    EEDAT          ;Data Memory Value to write
        BANKSEL  EECON1         ;
        BCF      EECON1, EEPGD  ;Point to DATA memory
        BSF      EECON1, WREN   ;Enable writes

        BCF      INTCON, GIE    ;Disable INTs.
        BTFSC    INTCON, GIE    ;SEE AN576
        GOTO     $-2
Required
Sequence
        MOVLW    55h            ;
        MOVWF    EECON2         ;Write 55h
        MOVLW    AAh            ;
        MOVWF    EECON2         ;Write AAh
        BSF      EECON1, WR     ;Set WR bit to begin write
        BSF      INTCON, GIE    ;Enable INTs.

        SLEEP                   ;Wait for interrupt to signal write complete
        BCF      EECON1, WREN   ;Disable writes
        BCF      STATUS, RP0    ;Bank 0
        BCF      STATUS, RP1
```


© 2006-2012 Microchip Technology Inc.

### 10.1.4 READING THE FLASH PROGRAM MEMORY

To read a program memory location, the user must write the Least and Most Significant address bits to the EEADR and EEADRH registers, set the EEPGD control bit of the EECON1 register, and then set control bit RD. Once the read control bit is set, the program memory Flash controller will use the second instruction cycle to read the data. This causes the second instruction immediately following the "BSF EECON1,RD" instruction to be ignored. The data is available in the very next cycle, in the EEDAT and EEDATH registers; therefore, it can be read as two bytes in the following instructions.

EEDAT and EEDATH registers will hold this value until another read or until it is written to by the user.

**Note 1:** The two instructions following a program memory read are required to be NOPs. This prevents the user from executing a two-cycle instruction on the next instruction after the RD bit is set.

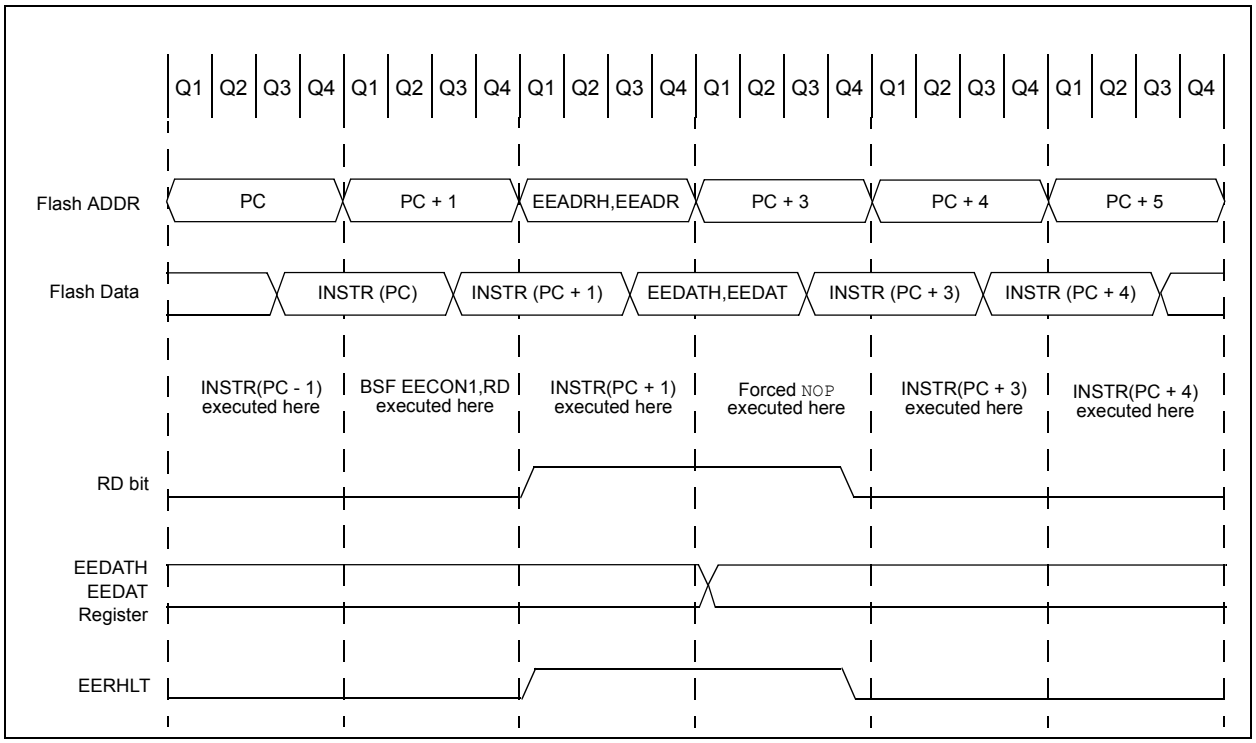
**2:** If the WR bit is set when EEPGD = 1, it will be immediately reset to '0' and no operation will take place.

**EXAMPLE 10-3: FLASH PROGRAM READ**

```
        BANKSEL EEADR                   ;
        MOVLW   MS_PROG_EE_ADDR         ;
        MOVWF   EEADRH                  ;MS Byte of Program Address to read
        MOVLW   LS_PROG_EE_ADDR         ;
        MOVWF   EEADR                   ;LS Byte of Program Address to read
        BANKSEL EECON1                  ;
        BSF     EECON1, EEPGD           ;Point to PROGRAM memory
        BSF     EECON1, RD              ;EE Read
                                        ;
                                        ;First instruction after BSF EECON1,RD executes normally
        NOP
        NOP                             ;Any instructions here are ignored as program
                                        ;memory is read in second cycle after BSF EECON1,RD
;
        BANKSEL EEDAT                   ;
        MOVF    EEDAT, W                ;W = LS Byte of Program Memory
        MOVWF   LOWPMBYTE               ;
        MOVF    EEDATH, W               ;W = MS Byte of Program EEDAT
        MOVWF   HIGHPMBYTE              ;
        BCF     STATUS, RP1             ;Bank 0
```

Required Sequence

**FIGURE 10-1: FLASH PROGRAM MEMORY READ CYCLE EXECUTION**





© 2006-2012 Microchip Technology Inc.

## 10.2 Writing to Flash Program Memory

Flash program memory may only be written to if the destination address is in a segment of memory that is not write-protected, as defined in bits WRT<1:0> of the Configuration Word Register 2. Flash program memory must be written in eight-word blocks (four-word blocks for 4K memory devices). See Figures 10-2 and 10-3 for more details. A block consists of eight words with sequential addresses, with a lower boundary defined by an address, where EEADR<2:0> = 000. All block writes to program memory are done as 16-word erase by eight-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 10-2). This is accomplished by first writing the destination address to EEADR and EEADRH and then writing the data to EEDATA and EEDATH. After the address and data have been set up, then the following sequence of events must be executed:

1. Set the EEPGD control bit of the EECON1 register.
2. Write 55h, then AAh, to EECON2 (Flash programming sequence).
3. Set the WR control bit of the EECON1 register.

All eight buffer register locations should be written to with correct data. If less than eight words are being written to in the block of eight 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 EEDATA and EEDATH 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 EEADR and EEADRH must point to the last location in the eight-word block (EEADR<2:0> = 111). Then the following sequence of events must be executed:

1. Set the EEPGD control bit of the EECON1 register.
2. Write 55h, then AAh, to EECON2 (Flash programming sequence).
3. Set control bit WR of the EECON1 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, 100, 101, 110, 111). When the write is performed on the last word (EEADR<2:0> = 111), a block of sixteen words is automatically erased and the content of the eight word buffer registers are written into the program memory.

After the "BSF EECON1,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 seven 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 eight-word write cycle, the processor will resume operation with the third instruction after the EECON1 write instruction. The above sequence must be repeated for the higher eight words.

**FIGURE 10-2: BLOCK WRITES TO 2K AND 4K FLASH PROGRAM MEMORY**

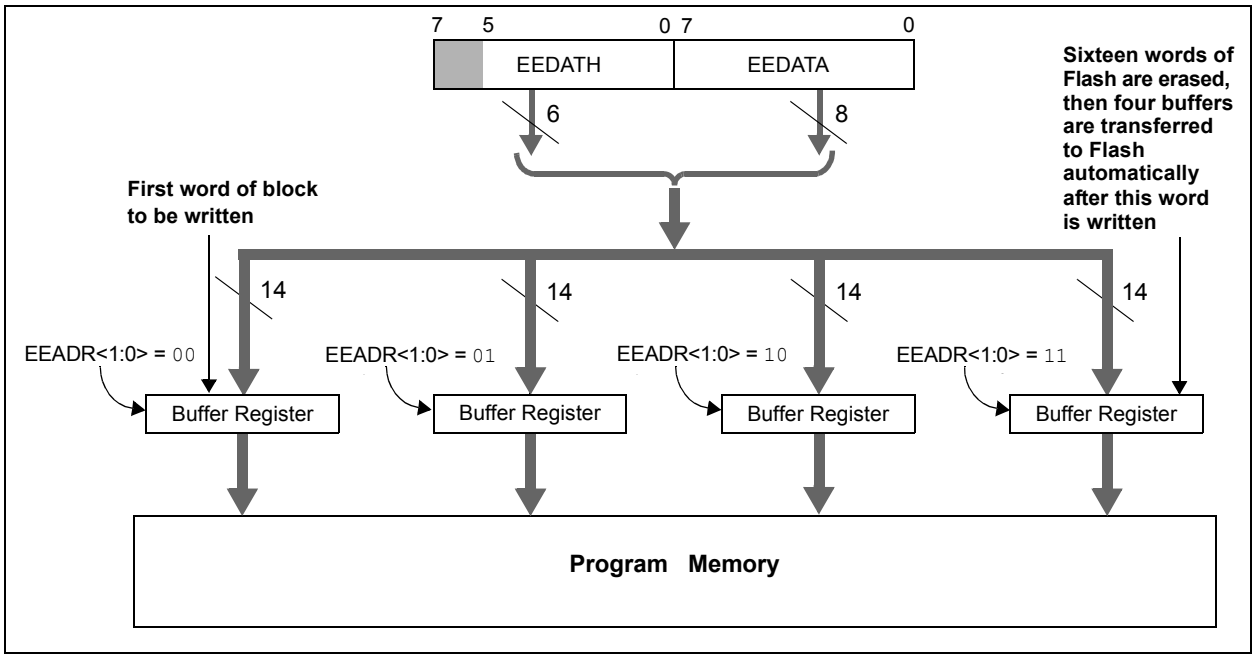
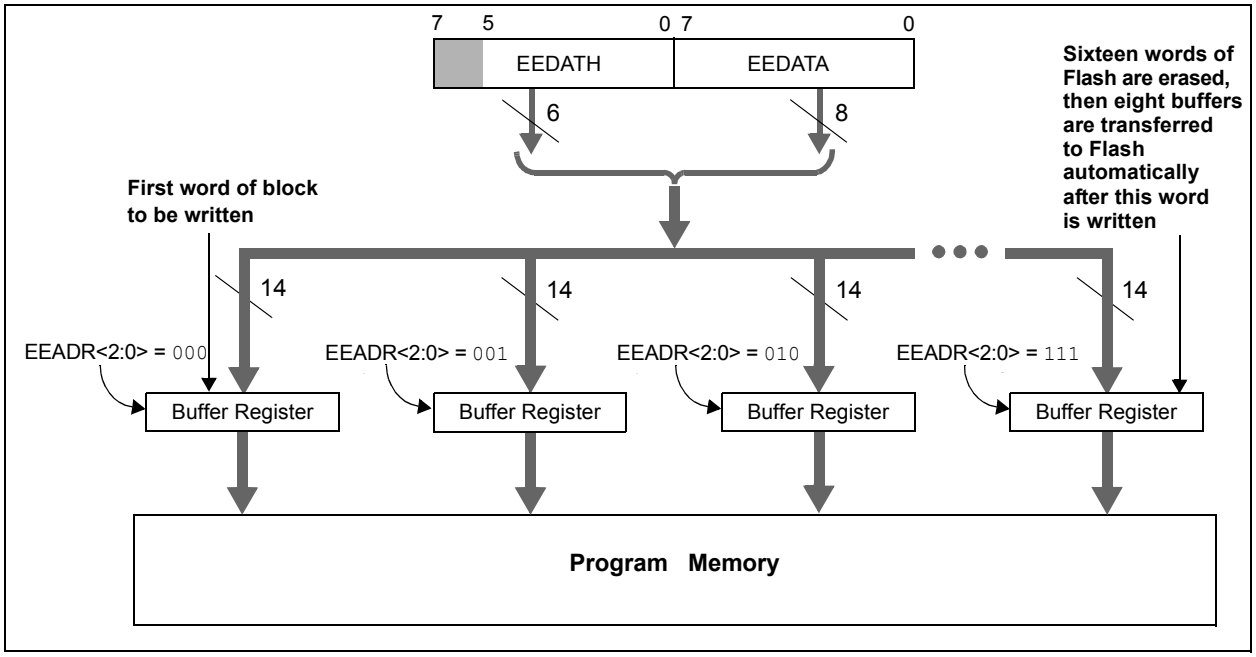



**FIGURE 10-3: BLOCK WRITES TO 8K FLASH PROGRAM MEMORY**





© 2006-2012 Microchip Technology Inc.

An example of the complete eight-word write sequence is shown in Example 10-4. The initial address is loaded into the EEADRH and EEADR register pair; the eight words of data are loaded using indirect addressing.

**EXAMPLE 10-4: WRITING TO FLASH PROGRAM MEMORY**

```
        ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
        ; This write routine assumes the following:
        ;     A valid starting address (the least significant bits = '000')
        ;     is loaded in ADDRH:ADDRL
        ;     ADDRH, ADDRL and DATADDR are all located in data memory
        ;
        BANKSEL  EEADRH
        MOVF     ADDRH,W      ; Load initial address
        MOVWF    EEADRH       ;
        MOVF     ADDRL,W      ;
        MOVWF    EEADR        ;
        MOVF     DATAADDR,W   ; Load initial data address
        MOVWF    FSR          ;
LOOP    MOVF     INDF,W       ; Load first data byte into lower
        MOVWF    EEDATA       ;
        INCF     FSR,F        ; Next byte
        MOVF     INDF,W       ; Load second data byte into upper
        MOVWF    EEDATH       ;
        INCF     FSR,F        ;
        BANKSEL  EECON1
        BSF      EECON1,EEPGD ; Point to program memory
        BSF      EECON1,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    EECON2       ; Write 55h
        MOVLW    0AAh         ;
        MOVWF    EECON2       ; Write 0AAh
        BSF      EECON1,WR    ; Set WR bit to begin write
        NOP                   ; Required to transfer data to the buffer
        NOP                   ; registers
        ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
        BCF      EECON1,WREN  ; Disable writes
        BSF      INTCON,GIE   ; Enable interrupts (comment out if not using interrupts)
        BANKSEL  EEADR
        MOVF     EEADR, W
        INCF     EEADR,F      ; Increment address
        ANDLW    0x0F         ; Indicates when sixteen words have been programmed
        SUBLW    0x0F         ;   0x0F = 16 words
                              ;   0x0B = 12 words (PIC16F884/883/882 only)
                              ;   0x07 = 8 words
                              ;   0x03 = 4 words(PIC16F884/883/882 only)
        BTFSS    STATUS,Z     ; Exit on a match,
        GOTO     LOOP         ; Continue if more data needs to be written
```

## 10.3 Write Verify

Depending on the application, good programming practice may dictate that the value written to the data EEPROM should be verified (see Example 10-5) to the desired value to be written.

**EXAMPLE 10-5: WRITE VERIFY**

```
BANKSEL EEDAT          ;
MOVF    EEDAT, W       ;EEDAT not changed
                       ;from previous write
BANKSEL EECON1         ;
BSF     EECON1, RD     ;YES, Read the
                       ;value written
BANKSEL EEDAT          ;
XORWF   EEDAT, W       ;
BTFSS   STATUS, Z      ;Is data the same
GOTO    WRITE_ERR      ;No, handle error
:                      ;Yes, continue
BCF     STATUS, RP1    ;Bank 0
```

### 10.3.1 USING THE DATA EEPROM

The data EEPROM is a high-endurance, byte addressable array that has been optimized for the storage of frequently changing information (e.g., program variables or other data that are updated often). When variables in one section change frequently, while variables in another section do not change, it is possible to exceed the total number of write cycles to the EEPROM (specification D124) without exceeding the total number of write cycles to a single byte (specifications D120 and D120A). If this is the case, then a refresh of the array must be performed. For this reason, variables that change infrequently (such as constants, IDs, calibration, etc.) should be stored in Flash program memory.

## 10.4 Protection Against Spurious Write

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

The write initiate sequence and the WREN bit together help prevent an accidental write during:

- Brown-out
- Power Glitch
- Software Malfunction

## 10.5 Data EEPROM Operation During Code-Protect

Data memory can be code-protected by programming the $\overline{\text{CPD}}$ bit in the Configuration Word Register 1 (Register 14-1) to '0'.

When the data memory is code-protected, only the CPU is able to read and write data to the data EEPROM. It is recommended to code-protect the program memory when code-protecting data memory. This prevents anyone from programming zeroes over the existing code (which will execute as NOPs) to reach an added routine, programmed in unused program memory, which outputs the contents of data memory. Programming unused locations in program memory to '0' will also help prevent data memory code protection from becoming breached.


© 2006-2012 Microchip Technology Inc.

**TABLE 10-1: SUMMARY OF REGISTERS ASSOCIATED WITH DATA EEPROM**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|
| EECON1 | EEPGD | — | — | — | WRERR | WREN | WR | RD | 117 |
| EECON2 | EEPROM Control Register 2 (not a physical register) | | | | | | | | — |
| EEADR | EEADR7 | EEADR6 | EEADR5 | EEADR4 | EEADR3 | EEADR2 | EEADR1 | EEADR0 | 116 |
| EEADRH | — | — | — | EEADRH4(1) | EEADRH3 | EEADRH2 | EEADRH1 | EEADRH0 | 116 |
| EEDAT | EEDAT7 | EEDAT6 | EEDAT5 | EEDAT4 | EEDAT3 | EEDAT2 | EEDAT1 | EEDAT0 | 116 |
| EEDATH | — | — | EEDATH5 | EEDATH4 | EEDATH3 | EEDATH2 | EEDATH1 | EEDATH0 | 116 |
| INTCON | GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | RBIF | 33 |
| PIE2 | OSFIE | C2IE | C1IE | EEIE | BCLIE | ULPWUIE | — | CCP2IE | 35 |
| PIR2 | OSFIF | C2IF | C1IF | EEIF | BCLIF | ULPWUIF | — | CCP2IF | 37 |

**Legend:** x = unknown, u = unchanged, – = unimplemented read as '0', q = value depends upon condition. Shaded cells are not used by data EEPROM module.

**Note 1:** PIC16F886/PIC16F887 only.

**NOTES:**


© 2006-2012 Microchip Technology Inc.

## 11.0 CAPTURE/COMPARE/PWM MODULES (CCP1 AND CCP2)

This device contains one Enhanced Capture/Compare/PWM (CCP1) and Capture/Compare/PWM module (CCP2). The CCP1 and CCP2 modules are identical in operation, with the exception of the Enhanced PWM features available on CCP1 only. See **Section 11.6 "PWM (Enhanced Mode)"** for more information.

| | |
|---|---|
| **Note:** | CCPRx and CCPx throughout this document refer to CCPR1 or CCPR2 and CCP1 or CCP2, respectively. |

## 11.1 Enhanced Capture/Compare/PWM (CCP1)

The Enhanced Capture/Compare/PWM module is a peripheral which allows the user to time and control different events. 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 a Pulse-Width Modulated signal of varying frequency and duty cycle.

Table 11-1 shows the timer resources required by the ECCP module.

**TABLE 11-1: ECCP MODE – TIMER RESOURCES REQUIRED**

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

## REGISTER DEFINITIONS: CCP CONTROL

### REGISTER 11-1: CCP1CON: ENHANCED CCP1 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 |
|---|---|---|---|---|---|---|---|
| P1M1 | P1M0 | DC1B1 | DC1B0 | CCP1M3 | CCP1M2 | CCP1M1 | CCP1M0 |
| 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 **P1M<1:0>:** PWM Output Configuration bits

If CCP1M<3:2> = 00, 01, 10:

xx = P1A assigned as Capture/Compare input; P1B, P1C, P1D assigned as port pins

If CCP1M<3:2> = 11:

00 = Single output; P1A modulated; P1B, P1C, P1D assigned as port pins
01 = Full-Bridge output forward; P1D modulated; P1A active; P1B, P1C inactive
10 = Half-Bridge output; P1A, P1B modulated with dead-band control; P1C, P1D assigned as port pins
11 = Full-Bridge output reverse; P1B modulated; P1C active; P1A, P1D inactive

bit 5-4 **DC1B<1:0>:** PWM Duty Cycle Least Significant bits

Capture mode:

Unused.

Compare mode:

Unused.

PWM mode:

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

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

0000 = Capture/Compare/PWM off (resets ECCP module)
0001 = Unused (reserved)
0010 = Compare mode, toggle output on match (CCP1IF bit is set)
0011 = Unused (reserved)
0100 = Capture mode, every falling edge
0101 = Capture mode, every rising edge
0110 = Capture mode, every 4th rising edge
0111 = Capture mode, every 16th rising edge
1000 = Compare mode, set output on match (CCP1IF bit is set)
1001 = Compare mode, clear output on match (CCP1IF bit is set)
1010 = Compare mode, generate software interrupt on match (CCP1IF bit is set, CCP1 pin is unaffected)
1011 = Compare mode, trigger special event (CCP1IF bit is set; CCP1 resets TMR1 or TMR2
1100 = PWM mode; P1A, P1C active-high; P1B, P1D active-high
1101 = PWM mode; P1A, P1C active-high; P1B, P1D active-low
1110 = PWM mode; P1A, P1C active-low; P1B, P1D active-high
1111 = PWM mode; P1A, P1C active-low; P1B, P1D active-low


© 2006-2012 Microchip Technology Inc.

## 11.2 Capture/Compare/PWM (CCP2)

The Capture/Compare/PWM module is a peripheral which allows the user to time and control different events. 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 a Pulse-Width Modulated signal of varying frequency and duty cycle.

The timer resources used by the module are shown in Table 11-2.

Additional information on CCP modules is available in the Application Note AN594, *"Using the CCP Modules"* (DS00594).

**TABLE 11-2: CCP MODE – TIMER RESOURCES REQUIRED**

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

**REGISTER 11-2: CCP2CON: CCP2 CONTROL REGISTER**

| U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | — | DC2B1 | DC2B0 | CCP2M3 | CCP2M2 | CCP2M1 | CCP2M0 |
| 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 **DC2B<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 CCPR2L.

bit 3-0 **CCP2M<3:0>:** CCP2 Mode Select bits

0000 = Capture/Compare/PWM off (resets CCP2 module)
0001 = Unused (reserved)
0010 = Unused (reserved)
0011 = Unused (reserved)
0100 = Capture mode, every falling edge
0101 = Capture mode, every rising edge
0110 = Capture mode, every 4th rising edge
0111 = Capture mode, every 16th rising edge
1000 = Compare mode, set output on match (CCP2IF bit is set)
1001 = Compare mode, clear output on match (CCP2IF bit is set)
1010 = Compare mode, generate software interrupt on match (CCP2IF bit is set, CCP2 pin is unaffected)
1011 = Compare mode, trigger special event (CCP2IF bit is set, TMR1 is reset and A/D conversion is started if the ADC module is enabled. CCP2 pin is unaffected.)
11xx = PWM mode.

## 11.3 Capture Mode

In Capture mode, the CCPRxH, CCPRxL register pair captures the 16-bit value of the TMR1 register when an event occurs on pin CCPx. 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

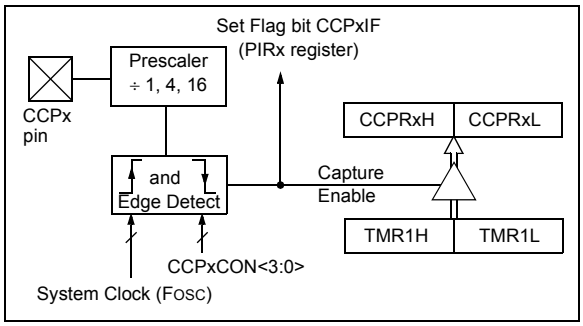
When a capture is made, the Interrupt Request Flag bit CCPxIF of the PIRx register is set. The interrupt flag must be cleared in software. If another capture occurs before the value in the CCPRxH, CCPRxL register pair is read, the old captured value is overwritten by the new captured value (see Figure 11-1).

### 11.3.1 CCP PIN CONFIGURATION

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

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

**FIGURE 11-1: CAPTURE MODE OPERATION BLOCK DIAGRAM**




### 11.3.2 TIMER1 MODE SELECTION

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

### 11.3.3 SOFTWARE INTERRUPT

When the Capture mode is changed, a false capture interrupt may be generated. The user should keep the CCPxIE interrupt enable bit of the PIEx register clear to avoid false interrupts. Additionally, the user should clear the CCPxIF interrupt flag bit of the PIRx register following any change in Operating mode.

### 11.3.4 CCP PRESCALER

There are four prescaler settings specified by the CCPxM<3:0> bits of the CCPxCON register. Whenever the CCP module is turned off, or the CCP module is not in Capture mode, the prescaler counter is cleared. Any Reset will clear the prescaler counter.

Switching from one capture prescaler to another does not clear the prescaler and may generate a false interrupt. To avoid this unexpected operation, turn the module off by clearing the CCPxCON register before changing the prescaler (see Example 11-1).

**EXAMPLE 11-1: CHANGING BETWEEN CAPTURE PRESCALERS**

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


© 2006-2012 Microchip Technology Inc.

## 11.4 Compare Mode

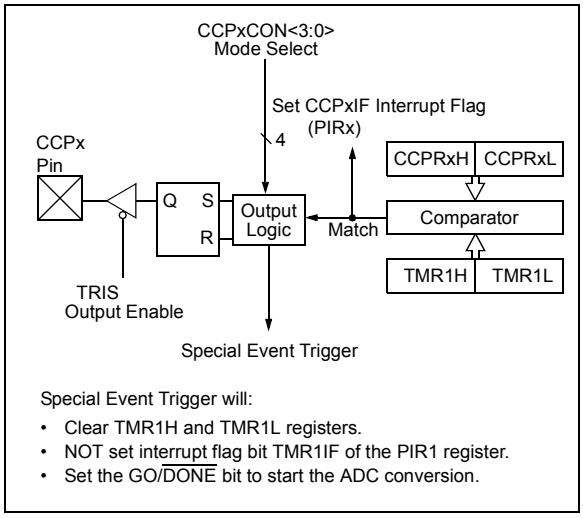
In Compare mode, the 16-bit CCPRx register value is constantly compared against the TMR1 register pair value. When a match occurs, the CCPx module may:

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

The action on the pin is based on the value of the CCPxM<3:0> control bits of the CCPx1CON register.

All Compare modes can generate an interrupt.

**FIGURE 11-2: COMPARE MODE OPERATION BLOCK DIAGRAM**




### 11.4.1 CCP PIN CONFIGURATION

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

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

### 11.4.2 TIMER1 MODE SELECTION

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.

### 11.4.3 SOFTWARE INTERRUPT MODE

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

### 11.4.4 SPECIAL EVENT TRIGGER

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

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

The CCPx module does not assert control of the CCPx pin in this mode (see the CCPxCON register).

The Special Event Trigger output of the CCP occurs immediately upon a match between the TMR1H, TMR1L register pair and the CCPRxH, CCPRxL register pair. The TMR1H, TMR1L register pair is not reset until the next rising edge of the Timer1 clock. This allows the CCPRxH, CCPRxL register pair to effectively provide a 16-bit programmable period register for Timer1.

| | |
|---|---|
| **Note 1:** | The Special Event Trigger from the CCP module does not set interrupt flag bit TMRxIF of the PIR1 register. |
| **2:** | Removing the match condition by changing the contents of the CCPRxH and CCPRxL register pair, between the clock edge that generates the Special Event Trigger and the clock edge that generates the Timer1 Reset, will preclude the Reset from occurring. |

## 11.5 PWM Mode

The PWM mode generates a Pulse-Width Modulated signal on the CCPx pin. The duty cycle, period and resolution are determined by the following registers:

- PR2
- T2CON
- CCPRxL
- CCPxCON

In Pulse-Width Modulation (PWM) mode, the CCP module produces up to a 10-bit resolution PWM output on the CCPx pin. Since the CCPx pin is multiplexed with the PORT data latch, the TRIS for that pin must be cleared to enable the CCPx pin output driver.

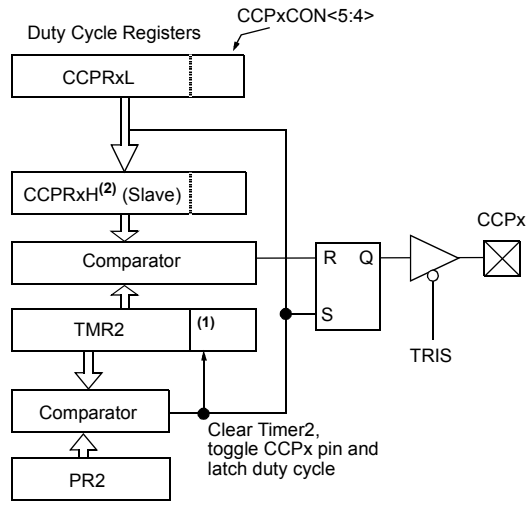
| **Note:** | Clearing the CCPxCON register will relinquish CCPx control of the CCPx pin. |
|---|---|

Figure 11-3 shows a simplified block diagram of PWM operation.

Figure 11-4 shows a typical waveform of the PWM signal.

For a step-by-step procedure on how to set up the CCP module for PWM operation, see **Section 11.5.7 "Setup for PWM Operation"**.

**FIGURE 11-3: SIMPLIFIED PWM BLOCK DIAGRAM**

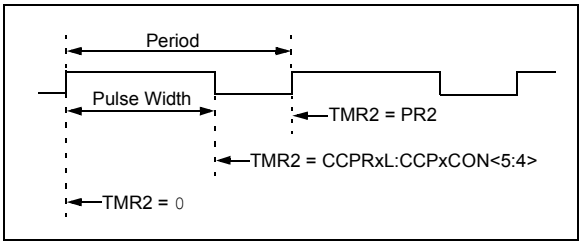



**Note 1:** The 8-bit timer TMR2 register is concatenated with the 2-bit internal system clock (FOSC), or 2 bits of the prescaler, to create the 10-bit time base.

**2:** In PWM mode, CCPRxH is a read-only register.

The PWM output (Figure 11-4) has a time base (period) and a time that the output stays high (duty cycle).

**FIGURE 11-4: CCP PWM OUTPUT**




### 11.5.1 PWM PERIOD

The PWM period is specified by the PR2 register of Timer2. The PWM period can be calculated using the formula of Equation 11-1.

**EQUATION 11-1: PWM PERIOD**

$$PWM\ Period = [(PR2) + 1] \bullet 4 \bullet TOSC \bullet (TMR2\ Prescale\ Value)$$

**Note:** TOSC = 1/FOSC

When TMR2 is equal to PR2, the following three events occur on the next increment cycle:

- TMR2 is cleared
- The CCPx pin is set. (Exception: If the PWM duty cycle = 0%, the pin will not be set.)
- The PWM duty cycle is latched from CCPRxL into CCPRxH.

| **Note:** | The Timer2 postscaler (see **Section 7.1 "Timer2 Operation"**) is not used in the determination of the PWM frequency. |
|---|---|


© 2006-2012 Microchip Technology Inc.

### 11.5.2 PWM DUTY CYCLE

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

Equation 11-2 is used to calculate the PWM pulse width.

Equation 11-3 is used to calculate the PWM duty cycle ratio.

**EQUATION 11-2: PULSE WIDTH**

$$\text{Pulse Width} = (\text{CCPRxL:CCPxCON<5:4>}) \bullet T_{OSC} \bullet (\text{TMR2 Prescale Value})$$

**EQUATION 11-3: DUTY CYCLE RATIO**

$$\text{Duty Cycle Ratio} = \frac{(\text{CCPRxL:CCPxCON<5:4>})}{4(PR2 + 1)}$$

The CCPRxH register and a 2-bit internal latch are used to double buffer the PWM duty cycle. This double buffering is essential for glitchless PWM operation.

The 8-bit timer TMR2 register is concatenated with either the 2-bit internal system clock (FOSC), or 2 bits of the prescaler, to create the 10-bit time base. The system clock is used if the Timer2 prescaler is set to 1:1.

When the 10-bit time base matches the CCPRxH and 2-bit latch, then the CCPx pin is cleared (see Figure 11-3).

### 11.5.3 PWM RESOLUTION

The resolution determines the number of available duty cycles for a given period. For example, a 10-bit resolution will result in 1024 discrete duty cycles, whereas an 8-bit resolution will result in 256 discrete duty cycles.

The maximum PWM resolution is 10 bits when PR2 is 255. The resolution is a function of the PR2 register value as shown by Equation 11-4.

**EQUATION 11-4: PWM RESOLUTION**

$$Resolution = \frac{log[4(PR2 + 1)]}{log(2)} \ bits$$

Note: If the pulse width value is greater than the period the assigned PWM pin(s) will remain unchanged.

**TABLE 11-3: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS (Fosc = 20 MHz)**

| PWM Frequency | 1.22 kHz | 4.88 kHz | 19.53 kHz | 78.12 kHz | 156.3 kHz | 208.3 kHz |
|---|---|---|---|---|---|---|
| Timer Prescale (1, 4, 16) | 16 | 4 | 1 | 1 | 1 | 1 |
| PR2 Value | 0xFF | 0xFF | 0xFF | 0x3F | 0x1F | 0x17 |
| Maximum Resolution (bits) | 10 | 10 | 10 | 8 | 7 | 6.6 |

**TABLE 11-4: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS (Fosc = 8 MHz)**

| PWM Frequency | 1.22 kHz | 4.90 kHz | 19.61 kHz | 76.92 kHz | 153.85 kHz | 200.0 kHz |
|---|---|---|---|---|---|---|
| Timer Prescale (1, 4, 16) | 16 | 4 | 1 | 1 | 1 | 1 |
| PR2 Value | 0x65 | 0x65 | 0x65 | 0x19 | 0x0C | 0x09 |
| Maximum Resolution (bits) | 8 | 8 | 8 | 6 | 5 | 5 |


© 2006-2012 Microchip Technology Inc.

### 11.5.4 OPERATION IN SLEEP MODE

In Sleep mode, the TMR2 register will not increment and the state of the module will not change. If the CCPx pin is driving a value, it will continue to drive that value. When the device wakes up, TMR2 will continue from its previous state.

### 11.5.5 CHANGES IN SYSTEM CLOCK FREQUENCY

The PWM frequency is derived from the system clock frequency. Any changes in the system clock frequency will result in changes to the PWM frequency. See **Section 4.0 "Oscillator Module (With Fail-Safe Clock Monitor)"** for additional details.

### 11.5.6 EFFECTS OF RESET

Any Reset will force all ports to Input mode and the CCP registers to their Reset states.

### 11.5.7 SETUP FOR PWM OPERATION

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

1. Disable the PWM pin (CCPx) output drivers as an input by setting the associated TRIS bit.
2. Set the PWM period by loading the PR2 register.
3. Configure the CCP module for the PWM mode by loading the CCPxCON register with the appropriate values.
4. Set the PWM duty cycle by loading the CCPRxL register and DCxB<1:0> bits of the CCPxCON register.
5. Configure and start Timer2:
   - Clear the TMR2IF interrupt flag bit of the PIR1 register.
   - Set the Timer2 prescale value by loading the T2CKPS bits of the T2CON register.
   - Enable Timer2 by setting the TMR2ON bit of the T2CON register.
6. Enable PWM output after a new PWM cycle has started:
   - Wait until Timer2 overflows (TMR2IF bit of the PIR1 register is set).
   - Enable the CCPx pin output driver by clearing the associated TRIS bit.

## 11.6 PWM (Enhanced Mode)

The Enhanced PWM Mode can generate a PWM signal on up to four different output pins with up to 10-bits of resolution. It can do this through four different PWM output modes:

- Single PWM
- Half-Bridge PWM
- Full-Bridge PWM, Forward mode
- Full-Bridge PWM, Reverse mode

To select an Enhanced PWM mode, the P1M bits of the CCP1CON register must be set appropriately.

| | |
|---|---|
| **Note:** | The PWM Enhanced mode is available on the Enhanced Capture/Compare/PWM module (CCP1) only. |

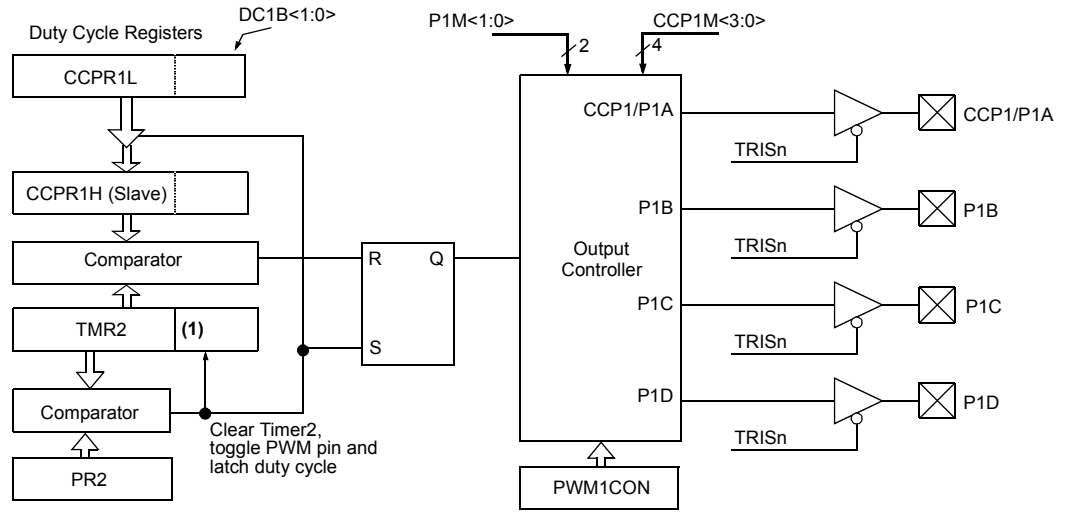
The PWM outputs are multiplexed with I/O pins and are designated P1A, P1B, P1C and P1D. The polarity of the PWM pins is configurable and is selected by setting the CCP1M bits in the CCP1CON register appropriately.

Table 11-5 shows the pin assignments for each Enhanced PWM mode.

Figure 11-5 shows an example of a simplified block diagram of the Enhanced PWM module.

| | |
|---|---|
| **Note:** | To prevent the generation of an incomplete waveform when the PWM is first enabled, the ECCP module waits until the start of a new PWM period before generating a PWM signal. |

**FIGURE 11-5: EXAMPLE SIMPLIFIED BLOCK DIAGRAM OF THE ENHANCED PWM MODE**




**Note 1:** The 8-bit timer TMR2 register is concatenated with the 2-bit internal Q clock, or 2 bits of the prescaler to create the 10-bit time base.

| | |
|---|---|
| **Note 1:** | The TRIS register value for each PWM output must be configured appropriately. |
| **2:** | Clearing the CCPxCON register will relinquish ECCP control of all PWM output pins. |
| **3:** | Any pin not used by an Enhanced PWM mode is available for alternate pin functions. |

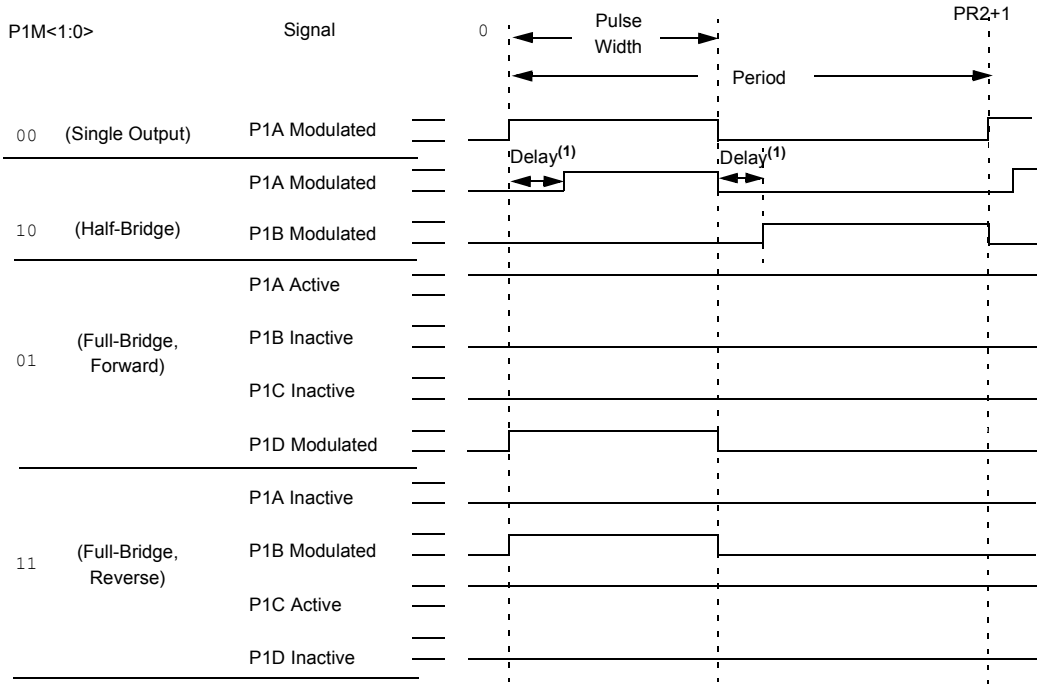
**TABLE 11-5: EXAMPLE PIN ASSIGNMENTS FOR VARIOUS PWM ENHANCED MODES**

| ECCP Mode | P1M<1:0> | CCP1/P1A | P1B | P1C | P1D |
|---|---|---|---|---|---|
| Single | 00 | Yes(1) | Yes(1) | Yes(1) | Yes(1) |
| Half-Bridge | 10 | Yes | Yes | No | No |
| Full-Bridge, Forward | 01 | Yes | Yes | Yes | Yes |
| Full-Bridge, Reverse | 11 | Yes | Yes | Yes | Yes |

**Note 1:** Pulse Steering enables outputs in Single mode.


© 2006-2012 Microchip Technology Inc.

**FIGURE 11-6: EXAMPLE PWM (ENHANCED MODE) OUTPUT RELATIONSHIPS (ACTIVE-HIGH STATE)**

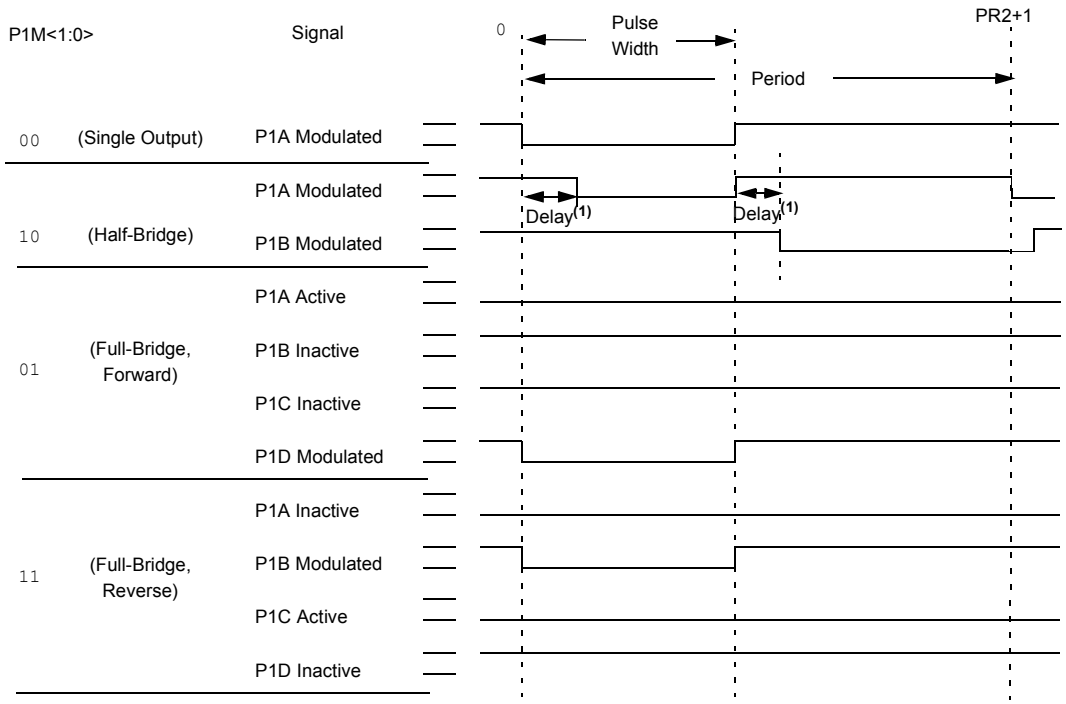



Relationships:

- Period = 4 * Tosc * (PR2 + 1) * (TMR2 Prescale Value)
- Pulse Width = Tosc * (CCPR1L<7:0>:CCP1CON<5:4>) * (TMR2 Prescale Value)
- Delay = 4 * Tosc * (PWM1CON<6:0>)

**Note 1:** Dead-band delay is programmed using the PWM1CON register (**Section 11.6.6 "Programmable Dead-Band Delay Mode"**).

**FIGURE 11-7: EXAMPLE ENHANCED PWM OUTPUT RELATIONSHIPS (ACTIVE-LOW STATE)**




Relationships:

- Period = 4 * Tosc * (PR2 + 1) * (TMR2 Prescale Value)
- Pulse Width = Tosc * (CCPR1L<7:0>:CCP1CON<5:4>) * (TMR2 Prescale Value)
- Delay = 4 * Tosc * (PWM1CON<6:0>)

**Note 1:** Dead-band delay is programmed using the PWM1CON register (**Section 11.6.6 "Programmable Dead-Band Delay Mode"**).


© 2006-2012 Microchip Technology Inc.

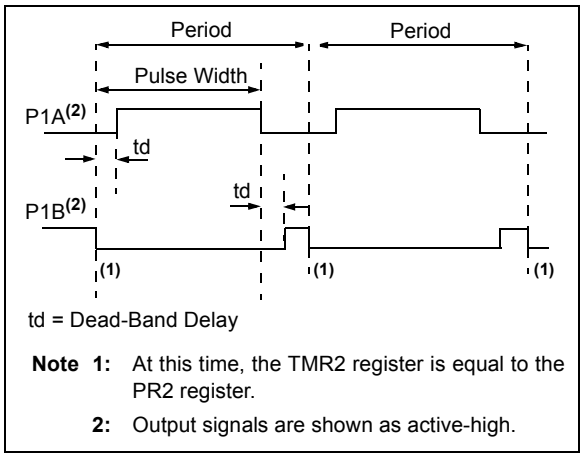
### 11.6.1 HALF-BRIDGE MODE

In Half-Bridge mode, two pins are used as outputs to drive push-pull loads. The PWM output signal is output on the CCPx/P1A pin, while the complementary PWM output signal is output on the P1B pin (see Figure 11-9). This mode can be used for Half-Bridge applications, as shown in Figure 11-9, or for Full-Bridge applications, where four power switches are being modulated with two PWM signals.

In Half-Bridge mode, the programmable dead-band delay can be used to prevent shoot-through current in Half-Bridge power devices. The value of the PDC<6:0> bits of the PWM1CON register sets the number of instruction cycles before the output is driven active. If the value is greater than the duty cycle, the corresponding output remains inactive during the entire cycle. See **Section 11.6.6 "Programmable Dead-Band Delay Mode"** for more details of the dead-band delay operations.

Since the P1A and P1B outputs are multiplexed with the PORT data latches, the associated TRIS bits must be cleared to configure P1A and P1B as outputs.

**FIGURE 11-8: EXAMPLE OF HALF-BRIDGE PWM OUTPUT**

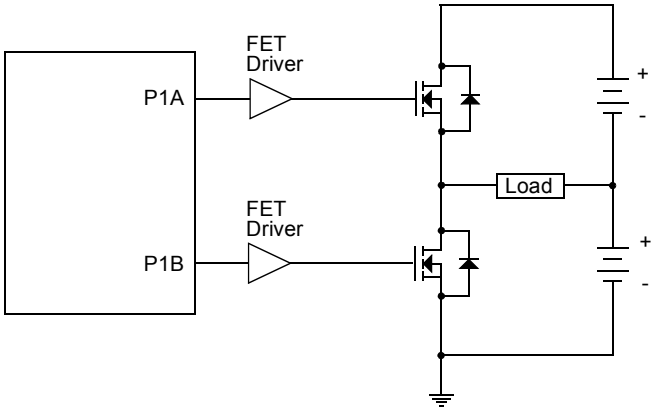



**Note 1:** At this time, the TMR2 register is equal to the PR2 register.

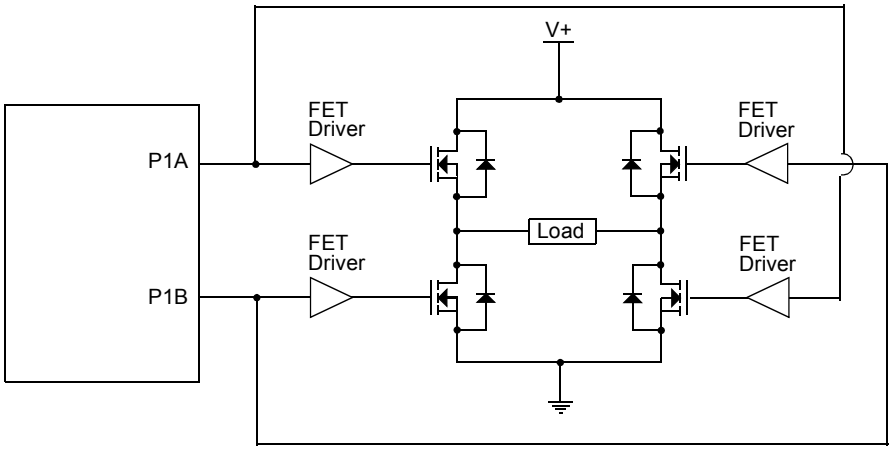
**2:** Output signals are shown as active-high.

**FIGURE 11-9: EXAMPLE OF HALF-BRIDGE APPLICATIONS**

**Standard Half-Bridge Circuit ("Push-Pull")**




**Half-Bridge Output Driving a Full-Bridge Circuit**



### 11.6.2 FULL-BRIDGE MODE

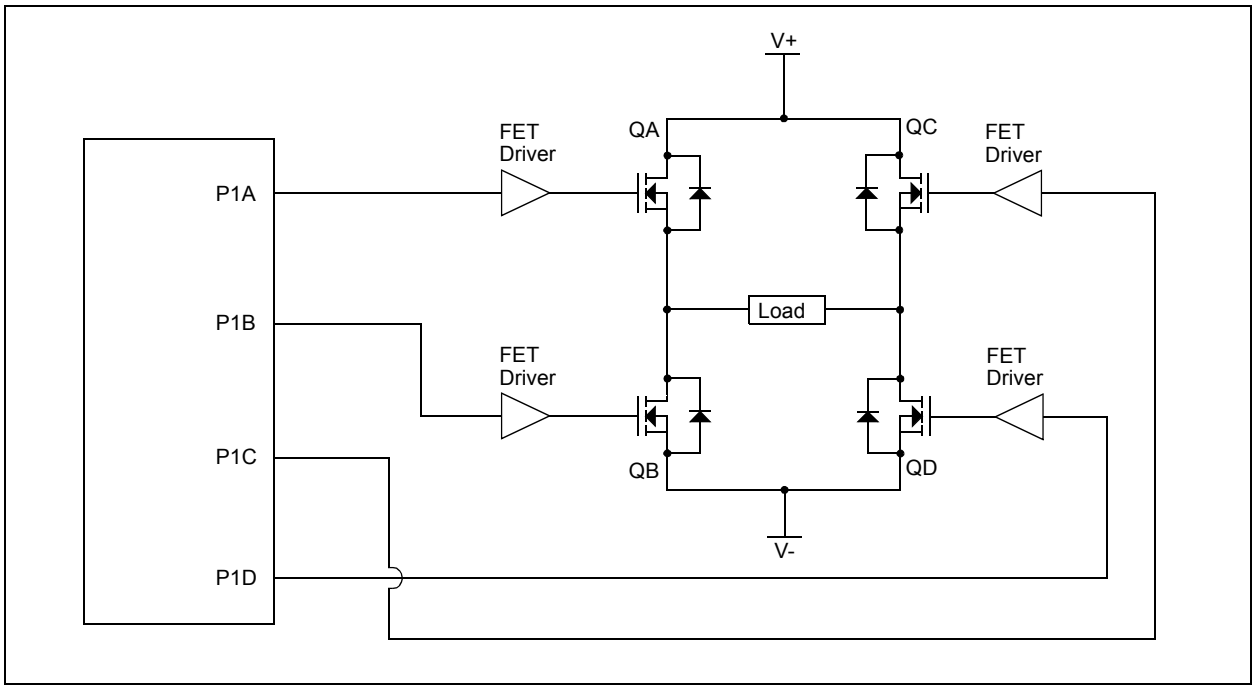
In Full-Bridge mode, all four pins are used as outputs. An example of Full-Bridge application is shown in Figure 11-10.

In the Forward mode, pin CCP1/P1A is driven to its active state, pin P1D is modulated, while P1B and P1C will be driven to their inactive state as shown in Figure 11-11.

In the Reverse mode, P1C is driven to its active state, pin P1B is modulated, while P1A and P1D will be driven to their inactive state as shown Figure 11-11.

P1A, P1B, P1C and P1D outputs are multiplexed with the PORT data latches. The associated TRIS bits must be cleared to configure the P1A, P1B, P1C and P1D pins as outputs.

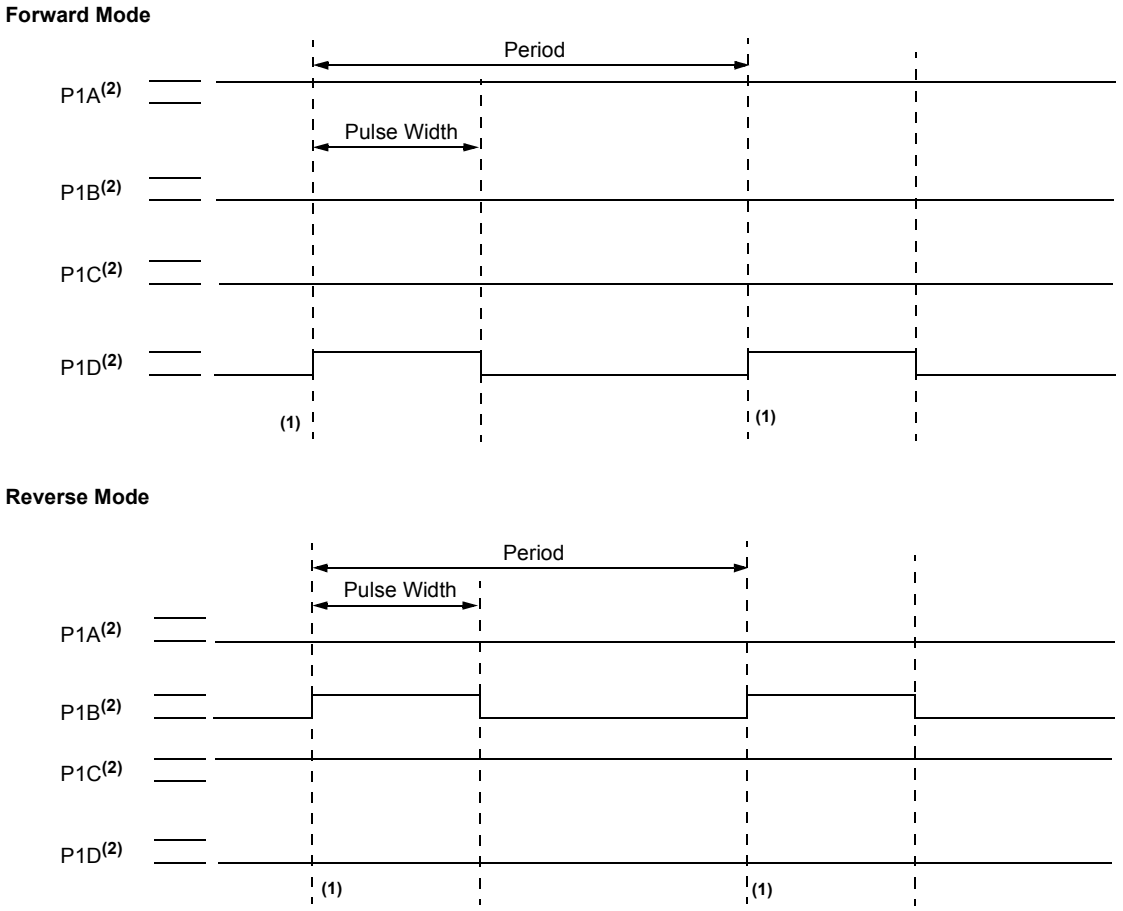
**FIGURE 11-10: EXAMPLE OF FULL-BRIDGE APPLICATION**





© 2006-2012 Microchip Technology Inc.

**FIGURE 11-11: EXAMPLE OF FULL-BRIDGE PWM OUTPUT**




**Note 1:** At this time, the TMR2 register is equal to the PR2 register.

**2:** Output signal is shown as active-high.

### 11.6.2.1 Direction Change in Full-Bridge Mode

In the Full-Bridge mode, the P1M1 bit in the CCP1CON register allows users to control the forward/reverse direction. When the application firmware changes this direction control bit, the module will change to the new direction on the next PWM cycle.

A direction change is initiated in software by changing the P1M1 bit of the CCP1CON register. The following sequence occurs prior to the end of the current PWM period:

- The modulated outputs (P1B and P1D) are placed in their inactive state.
- The associated unmodulated outputs (P1A and P1C) are switched to drive in the opposite direction.
- PWM modulation resumes at the beginning of the next period.

See Figure 11-12 for an illustration of this sequence.

The Full-Bridge mode does not provide dead-band delay. As one output is modulated at a time, dead-band delay is generally not required. There is a situation where dead-band delay is required. This situation occurs when both of the following conditions are true:

1. The direction of the PWM output changes when the duty cycle of the output is at or near 100%.
2. The turn off time of the power switch, including the power device and driver circuit, is greater than the turn on time.

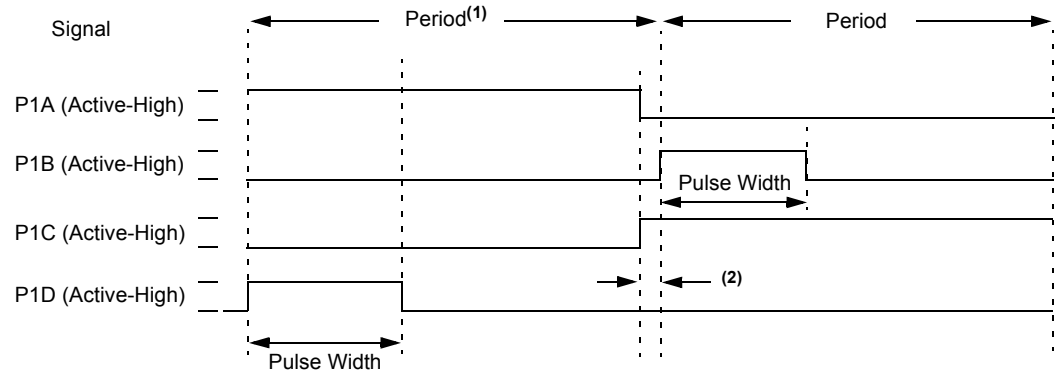
Figure 11-13 shows an example of the PWM direction changing from forward to reverse, at a near 100% duty cycle. In this example, at time t1, the output P1A and P1D become inactive, while output P1C becomes active. Since the turn off time of the power devices is longer than the turn on time, a shoot-through current will flow through power devices QC and QD (see Figure 11-10) for the duration of 't'. The same phenomenon will occur to power devices QA and QB for PWM direction change from reverse to forward.

If changing PWM direction at high duty cycle is required for an application, two possible solutions for eliminating the shoot-through current are:

1. Reduce PWM duty cycle for one PWM period before changing directions.
2. Use switch drivers that can drive the switches off faster than they can drive them on.

Other options to prevent shoot-through current may exist.

**FIGURE 11-12: EXAMPLE OF PWM DIRECTION CHANGE**

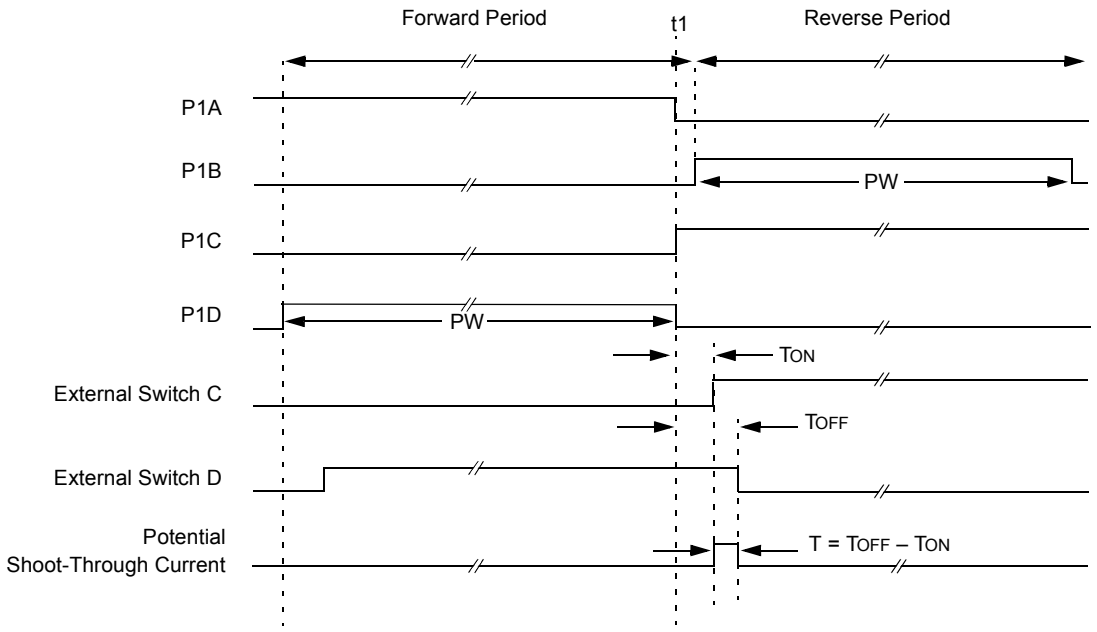



**Note 1:** The direction bit P1M1 of the CCP1CON register is written any time during the PWM cycle.

**2:** When changing directions, the P1A and P1C signals switch before the end of the current PWM cycle. The modulated P1B and P1D signals are inactive at this time. The length of this time is (1/Fosc) • TMR2 prescale value.


© 2006-2012 Microchip Technology Inc.

**FIGURE 11-13: EXAMPLE OF PWM DIRECTION CHANGE AT NEAR 100% DUTY CYCLE**




**Note 1:** All signals are shown as active-high.

**2:** TON is the turn on delay of power switch QC and its driver.

**3:** TOFF is the turn off delay of power switch QD and its driver.

### 11.6.3 START-UP CONSIDERATIONS

When any PWM mode is used, the application hardware must use the proper external pull-up and/or pull-down resistors on the PWM output pins.

| | |
|---|---|
| **Note:** | When the microcontroller is released from Reset, all of the I/O pins are in the high-impedance state. The external circuits must keep the power switch devices in the Off state until the microcontroller drives the I/O pins with the proper signal levels or activates the PWM output(s). |

The CCP1M<1:0> bits of the CCP1CON register allow the user to choose whether the PWM output signals are active-high or active-low for each pair of PWM output pins (P1A/P1C and P1B/P1D). The PWM output polarities must be selected before the PWM pin output drivers are enabled. Changing the polarity configuration while the PWM pin output drivers are enable is not recommended since it may result in damage to the application circuits.

The P1A, P1B, P1C and P1D output latches may not be in the proper states when the PWM module is initialized. Enabling the PWM pin output drivers at the same time as the Enhanced PWM modes may cause damage to the application circuit. The Enhanced PWM modes must be enabled in the proper Output mode and complete a full PWM cycle before enabling the PWM pin output drivers. The completion of a full PWM cycle is indicated by the TMR2IF bit of the PIR1 register being set as the second PWM period begins.


© 2006-2012 Microchip Technology Inc.

### 11.6.4 ENHANCED PWM AUTO-SHUTDOWN MODE

The PWM mode supports an Auto-Shutdown mode that will disable the PWM outputs when an external shutdown event occurs. Auto-Shutdown mode places the PWM output pins into a predetermined state. This mode is used to help prevent the PWM from damaging the application.

The auto-shutdown sources are selected using the ECCPAS<2:0> bits of the ECCPAS register. A shutdown event may be generated by:

- A logic '0' on the INT pin
- Comparator C1
- Comparator C2
- Setting the ECCPASE bit in firmware

A shutdown condition is indicated by the ECCPASE (Auto-Shutdown Event Status) bit of the ECCPAS register. If the bit is a '0', the PWM pins are operating normally. If the bit is a '1', the PWM outputs are in the shutdown state.

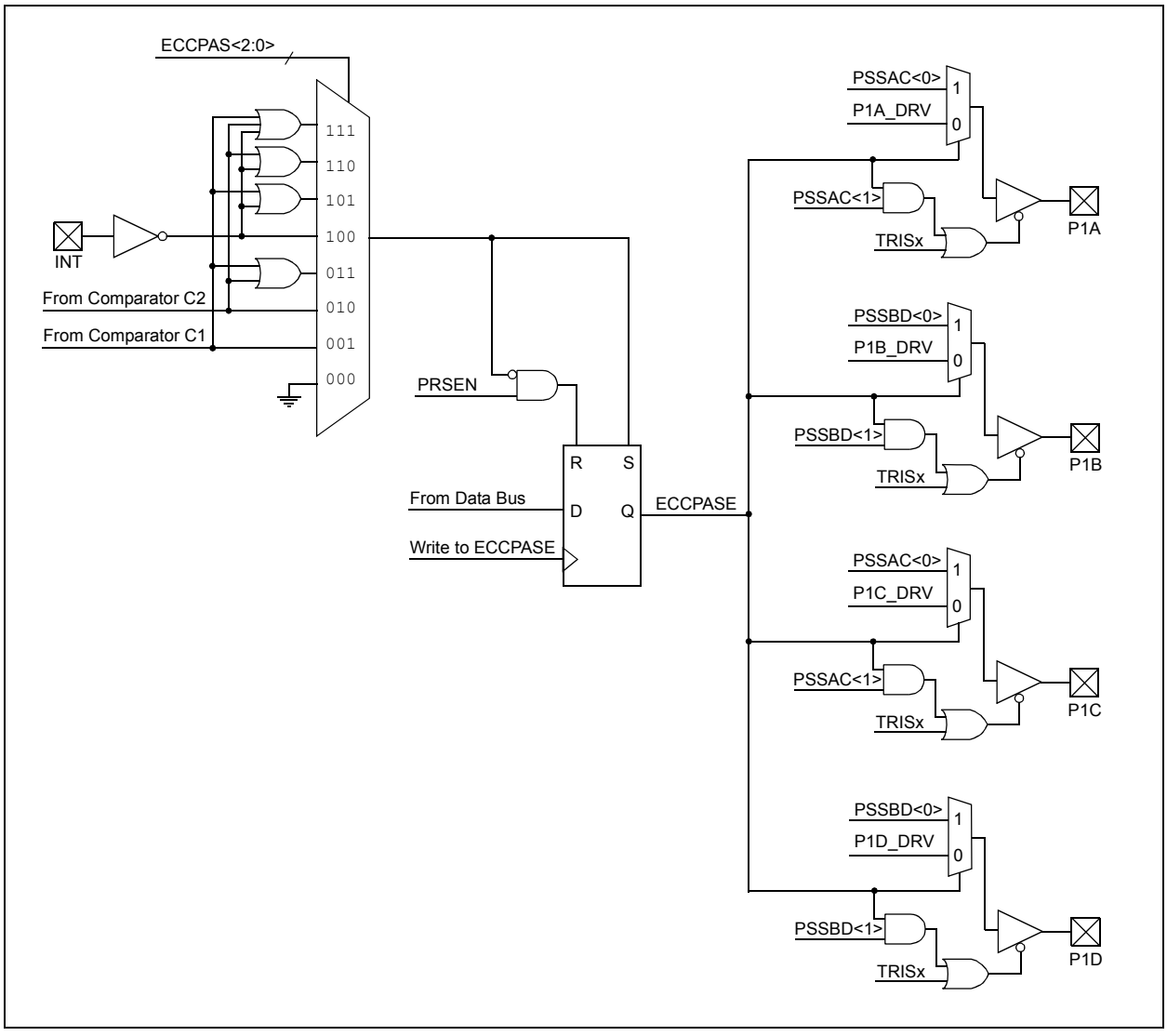
When a shutdown event occurs, two things happen:

The ECCPASE bit is set to '1'. The ECCPASE will remain set until cleared in firmware or an auto-restart occurs (see **Section 11.6.5 "Auto-Restart Mode"**).

The enabled PWM pins are asynchronously placed in their shutdown states. The PWM output pins are grouped into pairs [P1A/P1C] and [P1B/P1D]. The state of each pin pair is determined by the PSSAC and PSSBD bits of the ECCPAS register. Each pin pair may be placed into one of three states:

- Drive logic '1'
- Drive logic '0'
- Tri-state (high-impedance)

**FIGURE 11-14: AUTO-SHUTDOWN BLOCK DIAGRAM**



**REGISTER 11-3: ECCPAS: ENHANCED CAPTURE/COMPARE/PWM AUTO-SHUTDOWN 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 |
|---|---|---|---|---|---|---|---|
| ECCPASE | ECCPAS2 | ECCPAS1 | ECCPAS0 | PSSAC1 | PSSAC0 | PSSBD1 | PSSBD0 |
| 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 **ECCPASE:** ECCP Auto-Shutdown Event Status bit
1 = A shutdown event has occurred; ECCP outputs are in shutdown state
0 = ECCP outputs are operating

bit 6-4 **ECCPAS<2:0>:** ECCP Auto-shutdown Source Select bits
000 = Auto-Shutdown is disabled
001 = Comparator C1 output high
010 = Comparator C2 output high(1)
011 = Either Comparators output is high
100 = VIL on INT pin
101 = VIL on INT pin or Comparator C1 output high
110 = VIL on INT pin or Comparator C2 output high
111 =VIL on INT pin or either Comparators output is high

bit 3-2 **PSSACn:** Pins P1A and P1C Shutdown State Control bits
00 = Drive pins P1A and P1C to '0'
01 = Drive pins P1A and P1C to '1'
1x = Pins P1A and P1C tri-state

bit 1-0 **PSSBDn:** Pins P1B and P1D Shutdown State Control bits
00 = Drive pins P1B and P1D to '0'
01 = Drive pins P1B and P1D to '1'
1x = Pins P1B and P1D tri-state

**Note 1:** If C2SYNC is enabled, the shutdown will be delayed by Timer1.

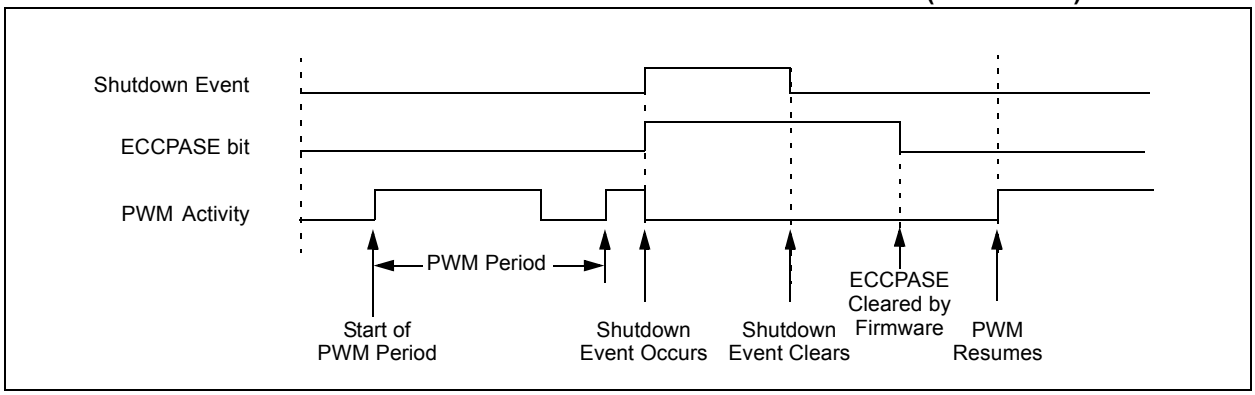
**Note 1:** The auto-shutdown condition is a level-based signal, not an edge-based signal. As long as the level is present, the auto-shutdown will persist.

**2:** Writing to the ECCPASE bit is disabled while an auto-shutdown condition persists.

**3:** Once the auto-shutdown condition has been removed and the PWM restarted (either through firmware or auto-restart) the PWM signal will always restart at the beginning of the next PWM period.


© 2006-2012 Microchip Technology Inc.

**FIGURE 11-15: PWM AUTO-SHUTDOWN WITH FIRMWARE RESTART (PRSEN = 0)**

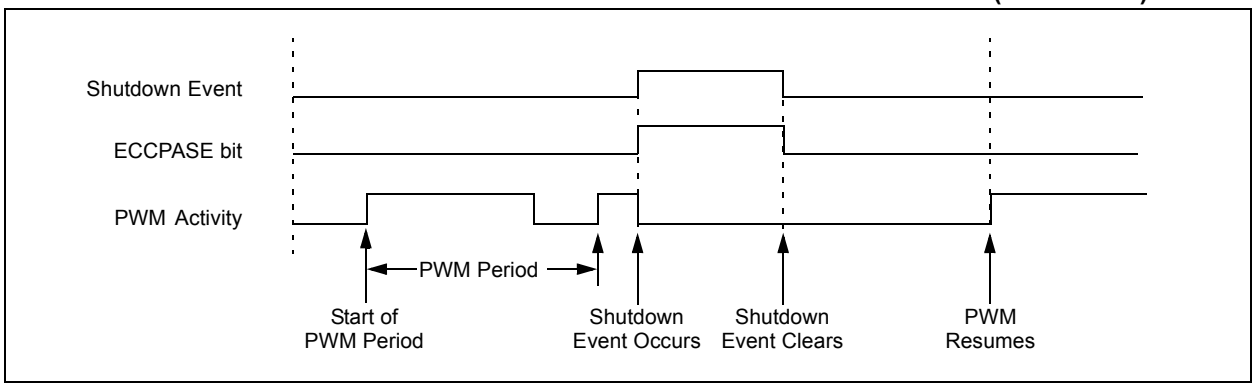



### 11.6.5 AUTO-RESTART MODE

The Enhanced PWM can be configured to automatically restart the PWM signal once the auto-shutdown condition has been removed. Auto-restart is enabled by setting the PRSEN bit in the PWM1CON register.

If auto-restart is enabled, the ECCPASE bit will remain set as long as the auto-shutdown condition is active. When the auto-shutdown condition is removed, the ECCPASE bit will be cleared via hardware and normal operation will resume.

**FIGURE 11-16: PWM AUTO-SHUTDOWN WITH AUTO-RESTART ENABLED (PRSEN = 1)**

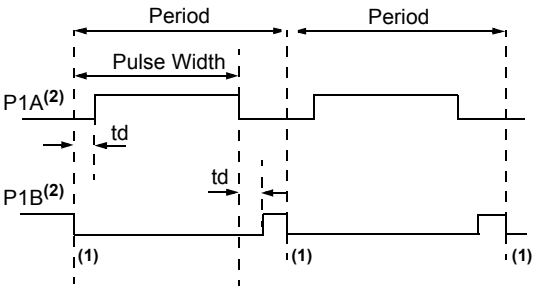


### 11.6.6 PROGRAMMABLE DEAD-BAND DELAY MODE

In Half-Bridge applications where all power switches are modulated at the PWM frequency, the power switches normally require more time to turn off than to turn on. If both the upper and lower power switches are switched at the same time (one turned on, and the other turned off), both switches may be on for a short period of time until one switch completely turns off. During this brief interval, a very high current (*shoot-through current*) will flow through both power switches, shorting the bridge supply. To avoid this potentially destructive shoot-through current from flowing during switching, turning on either of the power switches is normally delayed to allow the other switch to completely turn off.

In Half-Bridge mode, a digitally programmable dead-band delay is available to avoid shoot-through current from destroying the bridge power switches. The delay occurs at the signal transition from the non-active state to the active state. See Figure 11-17 for illustration. The lower seven bits of the associated PWM1CON register (Register 11-4) sets the delay period in terms of microcontroller instruction cycles (TCY or 4 TOSC).

**FIGURE 11-17: EXAMPLE OF HALF-BRIDGE PWM OUTPUT**

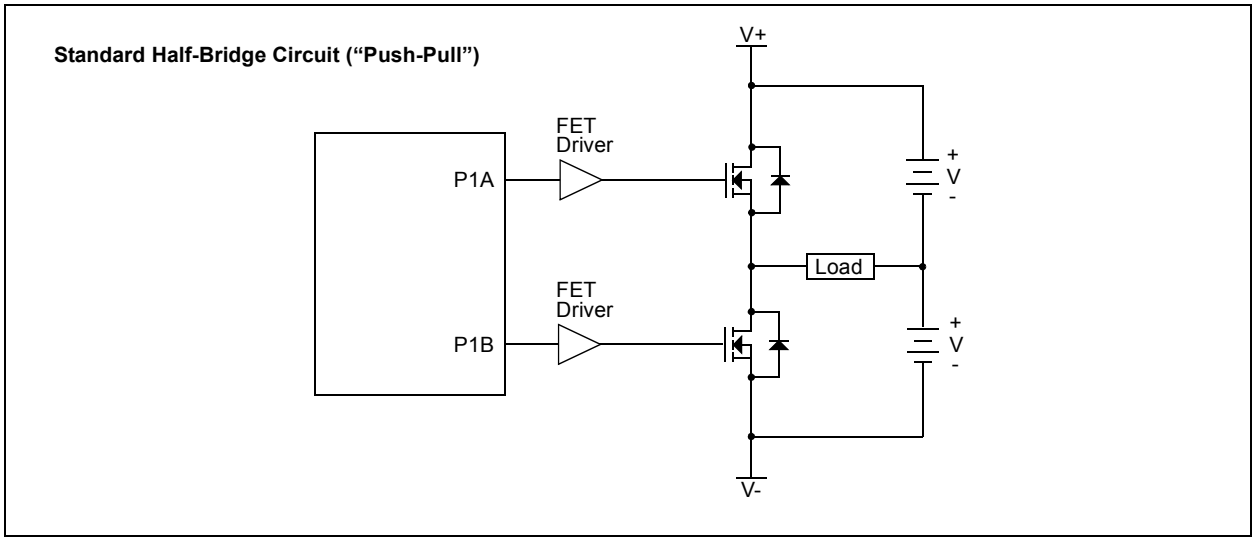



td = Dead-Band Delay

**Note 1:** At this time, the TMR2 register is equal to the PR2 register.

**2:** Output signals are shown as active-high.

**FIGURE 11-18: EXAMPLE OF HALF-BRIDGE APPLICATIONS**





© 2006-2012 Microchip Technology Inc.

## REGISTER DEFINITIONS: PWM CONTROL

### REGISTER 11-4: PWM1CON: ENHANCED PWM 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 |
|---|---|---|---|---|---|---|---|
| PRSEN | PDC6 | PDC5 | PDC4 | PDC3 | PDC2 | PDC1 | PDC0 |
| 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 **PRSEN:** PWM Restart Enable bit

1 = Upon auto-shutdown, the ECCPASE bit clears automatically once the shutdown event goes away; the PWM restarts automatically

0 = Upon auto-shutdown, ECCPASE must be cleared in software to restart the PWM

bit 6-0 **PDC<6:0>:** PWM Delay Count bits

PDCn = Number of FOSC/4 (4 * TOSC) cycles between the scheduled time when a PWM signal **should** transition active and the **actual** time it transitions active.

### 11.6.7 PULSE STEERING MODE

In Single Output mode, pulse steering allows any of the PWM pins to be the modulated signal. Additionally, the same PWM signal can be simultaneously available on multiple pins.

Once the Single Output mode is selected (CCP1M<3:2> = 11 and P1M<1:0> = 00 of the CCP1CON register), the user firmware can bring out the same PWM signal to one, two, three or four output pins by setting the appropriate STR<D:A> bits of the PSTRCON register, as shown in Table 11-5.

| **Note:** | The associated TRIS bits must be set to output ('0') to enable the pin output driver in order to see the PWM signal on the pin. |
|---|---|

While the PWM Steering mode is active, CCP1M<1:0> bits of the CCP1CON register select the PWM output polarity for the P1<D:A> pins.

The PWM auto-shutdown operation also applies to PWM Steering mode as described in **Section 11.6.4 "Enhanced PWM Auto-Shutdown Mode"**. An auto-shutdown event will only affect pins that have PWM outputs enabled.

## REGISTER DEFINITIONS: PULSE STEERING CONTROL

**REGISTER 11-5: PSTRCON: PULSE STEERING CONTROL REGISTER(1)**

| U-0 | U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-1 |
|---|---|---|---|---|---|---|---|
| — | — | — | STRSYNC | STRD | STRC | STRB | STRA |
| bit 7 | | | | | | | bit 0 |

| **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 **STRSYNC:** Steering Sync bit
1 = Output steering update occurs on next PWM period
0 = Output steering update occurs at the beginning of the instruction cycle boundary

bit 3 **STRD:** Steering Enable bit D
1 = P1D pin has the PWM waveform with polarity control from CCPxM<1:0>
0 = P1D pin is assigned to port pin

bit 2 **STRC:** Steering Enable bit C
1 = P1C pin has the PWM waveform with polarity control from CCPxM<1:0>
0 = P1C pin is assigned to port pin

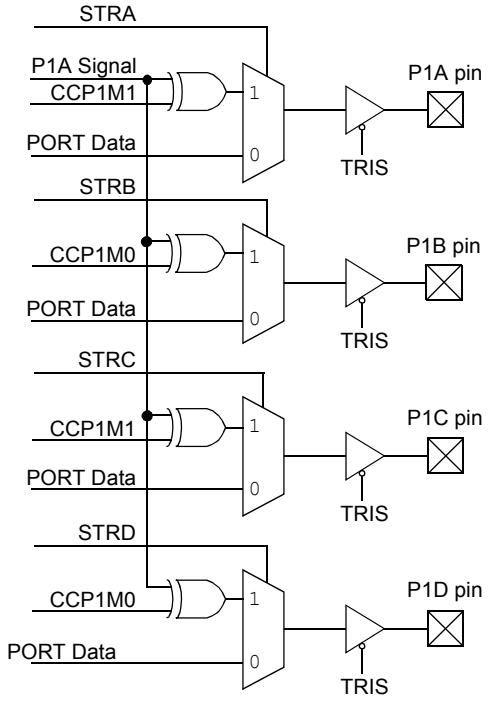
bit 1 **STRB:** Steering Enable bit B
1 = P1B pin has the PWM waveform with polarity control from CCPxM<1:0>
0 = P1B pin is assigned to port pin

bit 0 **STRA:** Steering Enable bit A
1 = P1A pin has the PWM waveform with polarity control from CCPxM<1:0>
0 = P1A pin is assigned to port pin

**Note 1:** The PWM Steering mode is available only when the CCP1CON register bits CCP1M<3:2> = 11 and P1M<1:0> = 00.


© 2006-2012 Microchip Technology Inc.

**FIGURE 11-19: SIMPLIFIED STEERING BLOCK DIAGRAM**




**Note 1:** Port outputs are configured as shown when the CCP1CON register bits P1M<1:0> = 00 and CCP1M<3:2> = 11.

**2:** Single PWM output requires setting at least one of the STRx bits.

#### 11.6.7.1 Steering Synchronization

The STRSYNC bit of the PSTRCON register gives the user two selections of when the steering event will happen. When the STRSYNC bit is '0', the steering event will happen at the end of the instruction that writes to the PSTRCON register. In this case, the output signal at the P1<D:A> pins may be an incomplete PWM waveform. This operation is useful when the user firmware needs to immediately remove a PWM signal from the pin.

When the STRSYNC bit is '1', the effective steering update will happen at the beginning of the next PWM period. In this case, steering on/off the PWM output will always produce a complete PWM waveform.

Figures 11-20 and 11-21 illustrate the timing diagrams of the PWM steering depending on the STRSYNC setting.

**FIGURE 11-20: EXAMPLE OF STEERING EVENT AT END OF INSTRUCTION (STRSYNC = 0)**

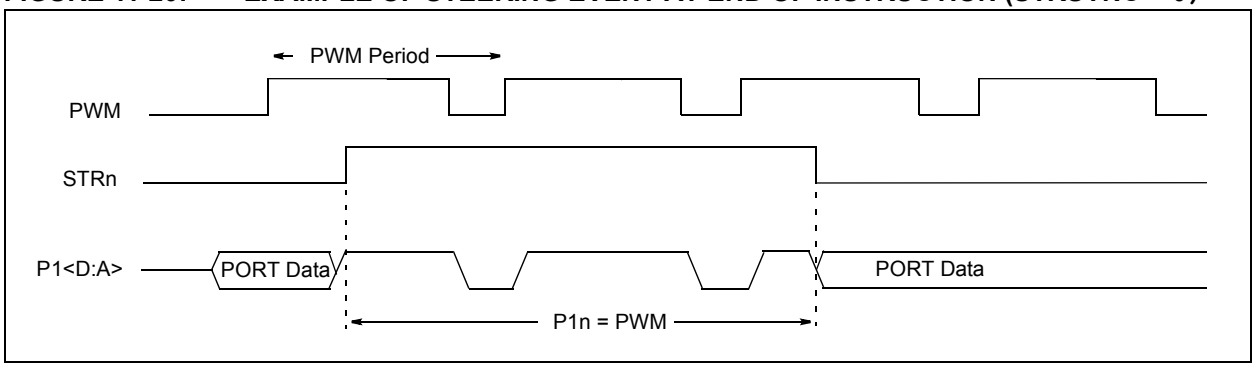
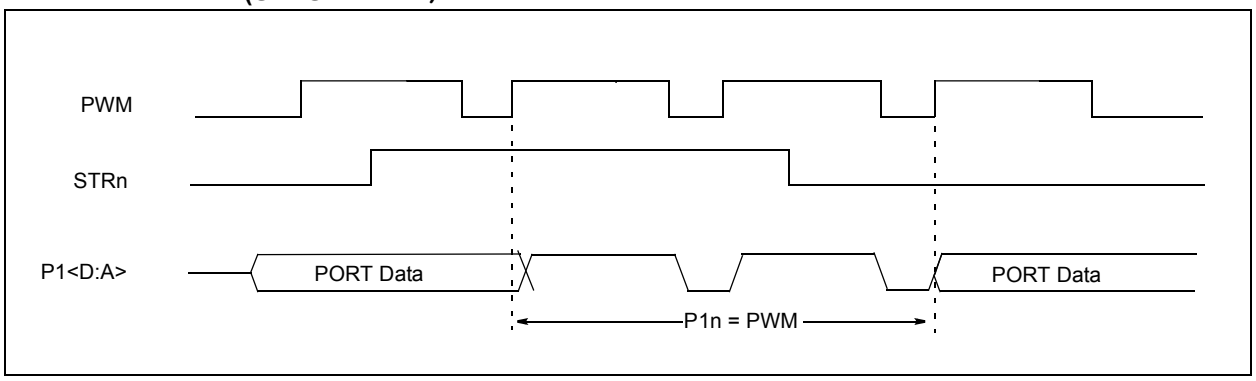



**FIGURE 11-21: EXAMPLE OF STEERING EVENT AT BEGINNING OF INSTRUCTION (STRSYNC = 1)**





© 2006-2012 Microchip Technology Inc.

**TABLE 11-6: REGISTERS ASSOCIATED WITH CAPTURE, COMPARE AND TIMER1**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|
| CCP1CON | P1M1 | P1M0 | DC1B1 | DC1B0 | CCP1M3 | CCP1M2 | CCP1M1 | CCP1M0 | 128 |
| CCP2CON | — | — | DC2B1 | DC2B0 | CCP2M3 | CCP2M2 | CCP2M1 | CCP2M0 | 129 |
| CCPR1L | Capture/Compare/PWM Register 1 Low Byte (LSB) | | | | | | | | 130 |
| CCPR1H | Capture/Compare/PWM Register 1 High Byte (MSB) | | | | | | | | 130 |
| CCPR2L | Capture/Compare/PWM Register 2 Low Byte (LSB) | | | | | | | | 130 |
| CCPR2H | Capture/Compare/PWM Register 2 High Byte (MSB) | | | | | | | | 130 |
| CM2CON1 | MC1OUT | MC2OUT | C1RSEL | C2RSEL | — | — | T1GSS | C2SYNC | 96 |
| INTCON | GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | RBIF | 33 |
| PIE1 | — | ADIE | RCIE | TXIE | SSPIE | CCP1IE | TMR2IE | TMR1IE | 34 |
| PIE2 | OSFIE | C2IE | C1IE | EEIE | BCLIE | ULPWUIE | — | CCP2IE | 35 |
| PIR1 | — | ADIF | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | 36 |
| PIR2 | OSFIF | C2IF | C1IF | EEIF | BCLIF | ULPWUIF | — | CCP2IF | 37 |
| T1CON | T1GINV | TMR1GE | T1CKPS1 | T1CKPS0 | T1OSCEN | $\overline{\text{T1SYNC}}$ | TMR1CS | TMR1ON | 84 |
| TMR1L | Holding Register for the Least Significant Byte of the 16-bit TMR1 Register | | | | | | | | 81 |
| TMR1H | Holding Register for the Most Significant Byte of the 16-bit TMR1 Register | | | | | | | | 81 |
| TRISC | TRISC7 | TRISC6 | TRISC5 | TRISC4 | TRISC3 | TRISC2 | TRISC1 | TRISC0 | 55 |

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

**TABLE 11-7: REGISTERS ASSOCIATED WITH PWM AND TIMER2**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|
| CCP1CON | P1M1 | P1M0 | DC1B1 | DC1B0 | CCP1M3 | CCP1M2 | CCP1M1 | CCP1M0 | 128 |
| CCP2CON | — | — | DC2B1 | DC2B0 | CCP2M3 | CCP2M2 | CCP2M1 | CCP2M0 | 129 |
| ECCPAS | ECCPASE | ECCPAS2 | ECCPAS1 | ECCPAS0 | PSSAC1 | PSSAC0 | PSSBD1 | PSSBD0 | 146 |
| INTCON | GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | RBIF | 33 |
| PR2 | Timer2 Period Register | | | | | | | | 87 |
| PSTRCON | — | — | — | STRSYNC | STRD | STRC | STRB | STRA | 150 |
| PWM1CON | PRSEN | PDC6 | PDC5 | PDC4 | PDC3 | PDC2 | PDC1 | PDC0 | 149 |
| T2CON | — | TOUTPS3 | TOUTPS2 | TOUTPS1 | TOUTPS0 | TMR2ON | T2CKPS1 | T2CKPS0 | 88 |
| TMR2 | Timer2 Module Register | | | | | | | | 87 |
| TRISB | TRISB7 | TRISB6 | TRISB5 | TRISB4 | TRISB3 | TRISB2 | TRISB1 | TRISB0 | 50 |
| TRISC | TRISC7 | TRISC6 | TRISC5 | TRISC4 | TRISC3 | TRISC2 | TRISC1 | TRISC0 | 55 |
| TRISD | TRISD7 | TRISD6 | TRISD5 | TRISD4 | TRISD3 | TRISD2 | TRISD1 | TRISD0 | 59 |

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

**NOTES:**


© 2006-2012 Microchip Technology Inc.

## 12.0 ENHANCED UNIVERSAL SYNCHRONOUS ASYNCHRONOUS RECEIVER TRANSMITTER (EUSART)

The Enhanced Universal Synchronous Asynchronous Receiver Transmitter (EUSART) module is a serial I/O communications peripheral. It contains all the clock generators, shift registers and data buffers necessary to perform an input or output serial data transfer independent of device program execution. The EUSART, also known as a Serial Communications Interface (SCI), can be configured as a full-duplex asynchronous system or half-duplex synchronous system. Full-Duplex mode is useful for communications with peripheral systems, such as CRT terminals and personal computers. Half-Duplex Synchronous mode is intended for communications with peripheral devices, such as A/D or D/A integrated circuits, serial EEPROMs or other microcontrollers. These devices typically do not have internal clocks for baud rate generation and require the external clock signal provided by a master synchronous device.

The EUSART module includes the following capabilities:

- Full-duplex asynchronous transmit and receive
- Two-character input buffer
- One-character output buffer
- Programmable 8-bit or 9-bit character length
- Address detection in 9-bit mode
- Input buffer overrun error detection
- Received character framing error detection
- Half-duplex synchronous master
- Half-duplex synchronous slave
- Programmable clock polarity in synchronous modes
- Sleep operation

The EUSART module implements the following additional features, making it ideally suited for use in Local Interconnect Network (LIN) bus systems:

- Automatic detection and calibration of the baud rate
- Wake-up on Break reception
- 13-bit Break character transmit

Block diagrams of the EUSART transmitter and receiver are shown in Figure 12-1 and Figure 12-2.

**FIGURE 12-1: EUSART TRANSMIT BLOCK DIAGRAM**

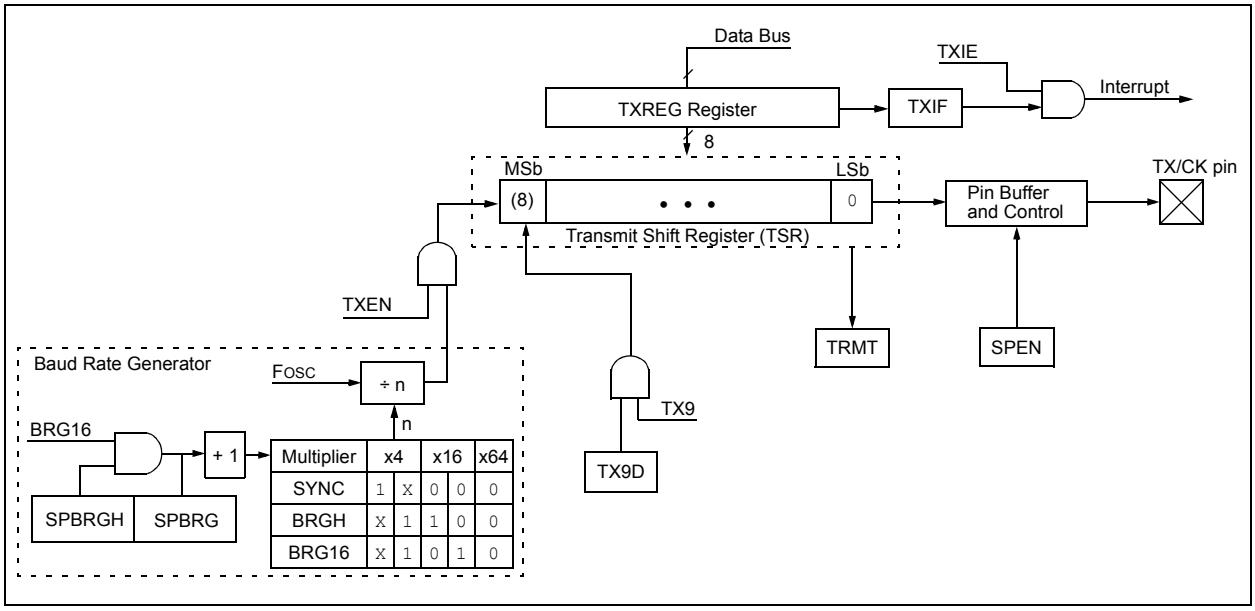
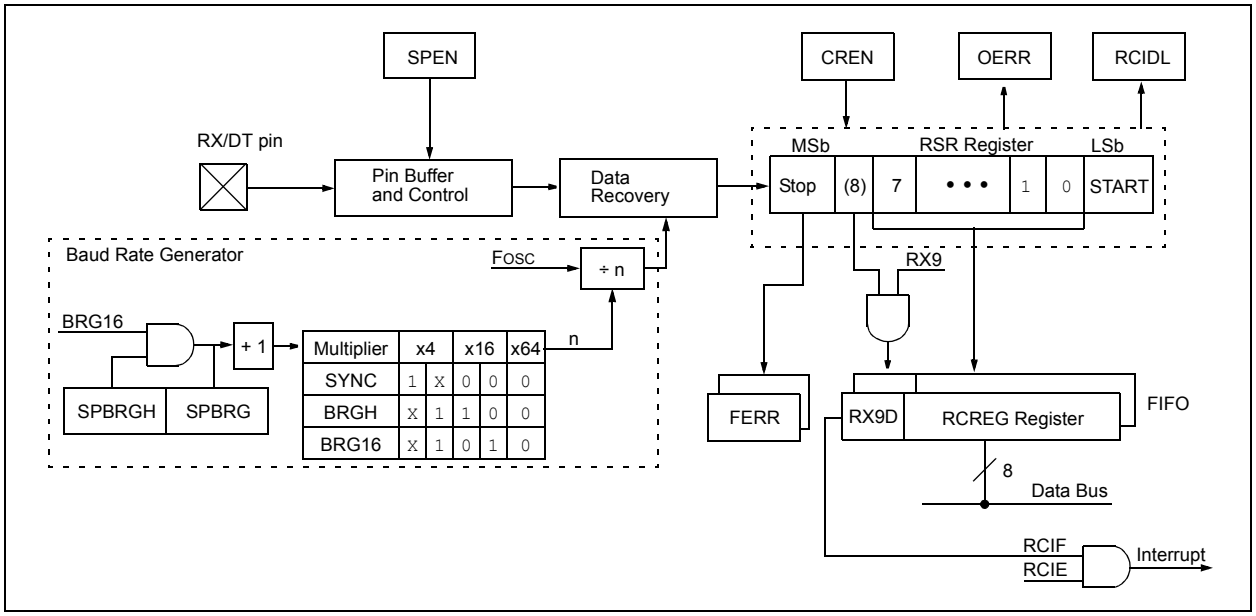


**FIGURE 12-2: EUSART RECEIVE BLOCK DIAGRAM**




The operation of the EUSART module is controlled through three registers:

- Transmit Status and Control (TXSTA)
- Receive Status and Control (RCSTA)
- Baud Rate Control (BAUDCTL)

These registers are detailed in Register 12-1, Register 12-2 and Register 12-3, respectively.


© 2006-2012 Microchip Technology Inc.

## 12.1 EUSART Asynchronous Mode

The EUSART transmits and receives data using the standard non-return-to-zero (NRZ) format. NRZ is implemented with two levels: a VOH mark state which represents a '1' data bit, and a VOL space state which represents a '0' data bit. NRZ refers to the fact that consecutively transmitted data bits of the same value stay at the output level of that bit without returning to a neutral level between each bit transmission. An NRZ transmission port idles in the mark state. Each character transmission consists of one Start bit followed by eight or nine data bits and is always terminated by one or more Stop bits. The Start bit is always a space and the Stop bits are always marks. The most common data format is 8 bits. Each transmitted bit persists for a period of 1/(Baud Rate). An on-chip dedicated 8-bit/16-bit Baud Rate Generator is used to derive standard baud rate frequencies from the system oscillator. See Table 12-5 for examples of baud rate configurations.

The EUSART transmits and receives the LSb first. The EUSART's transmitter and receiver are functionally independent, but share the same data format and baud rate. Parity is not supported by the hardware, but can be implemented in software and stored as the ninth data bit.

### 12.1.1 EUSART ASYNCHRONOUS TRANSMITTER

The EUSART transmitter block diagram is shown in Figure 12-1. The heart of the transmitter is the serial Transmit Shift Register (TSR), which is not directly accessible by software. The TSR obtains its data from the transmit buffer, which is the TXREG register.

#### 12.1.1.1 Enabling the Transmitter

The EUSART transmitter is enabled for asynchronous operations by configuring the following three control bits:

- TXEN = 1
- SYNC = 0
- SPEN = 1

All other EUSART control bits are assumed to be in their default state.

Setting the TXEN bit of the TXSTA register enables the transmitter circuitry of the EUSART. Clearing the SYNC bit of the TXSTA register configures the EUSART for asynchronous operation. Setting the SPEN bit of the RCSTA register enables the EUSART and automatically configures the TX/CK I/O pin as an output. If the TX/CK pin is shared with an analog peripheral the analog I/O function must be disabled by clearing the corresponding ANSEL bit.

**Note 1:** When the SPEN bit is set the RX/DT I/O pin is automatically configured as an input, regardless of the state of the corresponding TRIS bit and whether or not the EUSART receiver is enabled. The RX/DT pin data can be read via a normal PORT read but PORT latch data output is precluded.

**2:** The TXIF transmitter interrupt flag is set when the TXEN enable bit is set.

#### 12.1.1.2 Transmitting Data

A transmission is initiated by writing a character to the TXREG register. If this is the first character, or the previous character has been completely flushed from the TSR, the data in the TXREG is immediately transferred to the TSR register. If the TSR still contains all or part of a previous character, the new character data is held in the TXREG until the Stop bit of the previous character has been transmitted. The pending character in the TXREG is then transferred to the TSR in one TCY immediately following the Stop bit transmission. The transmission of the Start bit, data bits and Stop bit sequence commences immediately following the transfer of the data to the TSR from the TXREG.

#### 12.1.1.3 Transmit Interrupt Flag

The TXIF interrupt flag bit of the PIR1 register is set whenever the EUSART transmitter is enabled and no character is being held for transmission in the TXREG. In other words, the TXIF bit is only clear when the TSR is busy with a character and a new character has been queued for transmission in the TXREG. The TXIF flag bit is not cleared immediately upon writing TXREG. TXIF becomes valid in the second instruction cycle following the write execution. Polling TXIF immediately following the TXREG write will return invalid results. The TXIF bit is read-only, it cannot be set or cleared by software.

The TXIF interrupt can be enabled by setting the TXIE interrupt enable bit of the PIE1 register. However, the TXIF flag bit will be set whenever the TXREG is empty, regardless of the state of TXIE enable bit.

To use interrupts when transmitting data, set the TXIE bit only when there is more data to send. Clear the TXIE interrupt enable bit upon writing the last character of the transmission to the TXREG.

#### 12.1.1.4 TSR Status

The TRMT bit of the TXSTA register indicates the status of the TSR register. This is a read-only bit. The TRMT bit is set when the TSR register is empty and is cleared when a character is transferred to the TSR register from the TXREG. The TRMT bit remains clear until all bits have been shifted out of the TSR register. No interrupt logic is tied to this bit, so the user has to poll this bit to determine the TSR status.

| **Note:** | The TSR register is not mapped in data memory, so it is not available to the user. |
|---|---|

#### 12.1.1.5 Transmitting 9-Bit Characters

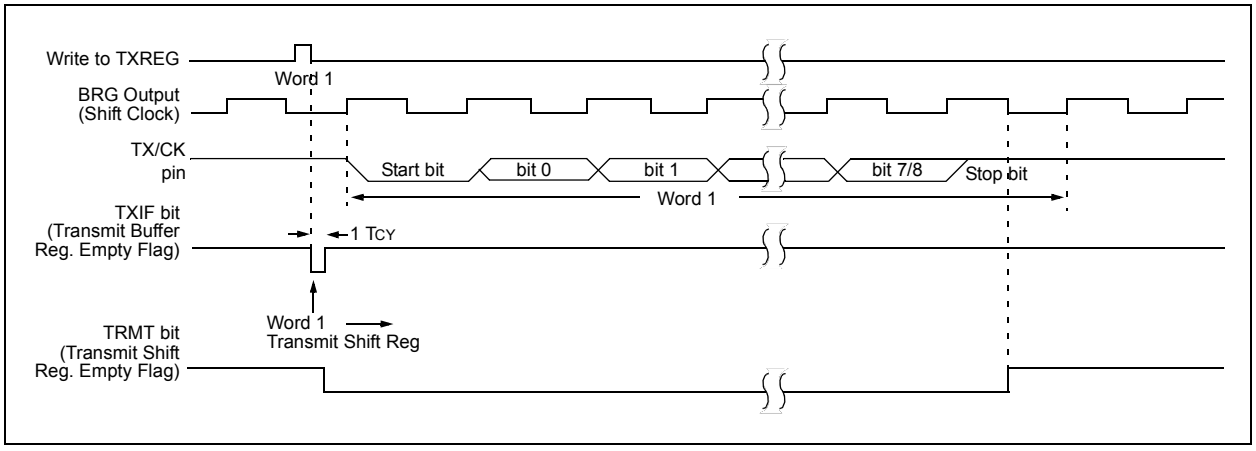
The EUSART supports 9-bit character transmissions. When the TX9 bit of the TXSTA register is set the EUSART will shift 9 bits out for each character transmitted. The TX9D bit of the TXSTA register is the ninth, and Most Significant, data bit. When transmitting 9-bit data, the TX9D data bit must be written before writing the 8 Least Significant bits into the TXREG. All nine bits of data will be transferred to the TSR shift register immediately after the TXREG is written.

A special 9-bit Address mode is available for use with multiple receivers. See **Section 12.1.2.7 "Address Detection"** for more information on the Address mode.

#### 12.1.1.6 Asynchronous Transmission Setup:

1. Initialize the SPBRGH, SPBRG register pair and the BRGH and BRG16 bits to achieve the desired baud rate (see **Section 12.3 "EUSART Baud Rate Generator (BRG)"**).
2. Enable the asynchronous serial port by clearing the SYNC bit and setting the SPEN bit.
3. If 9-bit transmission is desired, set the TX9 control bit. A set ninth data bit will indicate that the 8 Least Significant data bits are an address when the receiver is set for address detection.
4. Enable the transmission by setting the TXEN control bit. This will cause the TXIF interrupt bit to be set.
5. If interrupts are desired, set the TXIE interrupt enable bit of the PIE1 register. An interrupt will occur immediately provided that the GIE and PEIE bits of the INTCON register are also set.
6. If 9-bit transmission is selected, the ninth bit should be loaded into the TX9D data bit.
7. Load 8-bit data into the TXREG register. This will start the transmission.

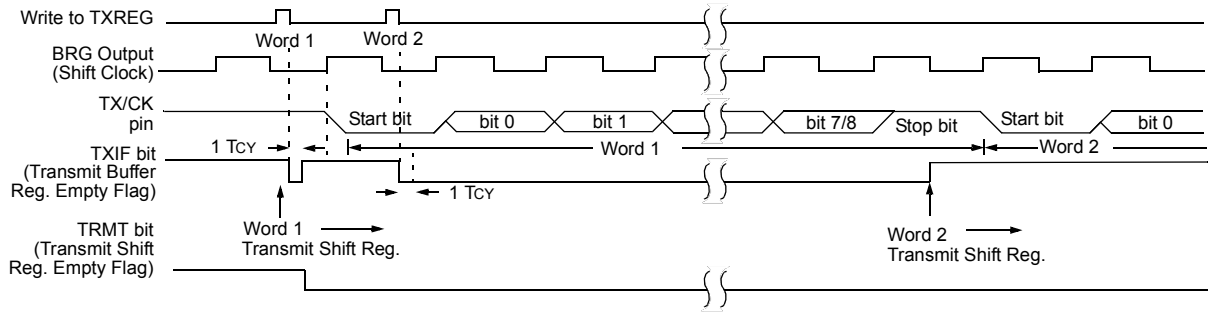
**FIGURE 12-3: ASYNCHRONOUS TRANSMISSION**





© 2006-2012 Microchip Technology Inc.

**FIGURE 12-4: ASYNCHRONOUS TRANSMISSION (BACK-TO-BACK)**




**Note:** This timing diagram shows two consecutive transmissions.

**TABLE 12-1: REGISTERS ASSOCIATED WITH ASYNCHRONOUS TRANSMISSION**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|
| BAUDCTL | ABDOVF | RCIDL | — | SCKP | BRG16 | — | WUE | ABDEN | 166 |
| INTCON | GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | RBIF | 33 |
| PIE1 | — | ADIE | RCIE | TXIE | SSPIE | CCP1IE | TMR2IE | TMR1IE | 34 |
| PIR1 | — | ADIF | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | 36 |
| RCREG | EUSART Receive Data Register | | | | | | | | 162 |
| RCSTA | SPEN | RX9 | SREN | CREN | ADDEN | FERR | OERR | RX9D | 165 |
| SPBRG | BRG7 | BRG6 | BRG5 | BRG4 | BRG3 | BRG2 | BRG1 | BRG0 | 167 |
| SPBRGH | BRG15 | BRG14 | BRG13 | BRG12 | BRG11 | BRG10 | BRG9 | BRG8 | 167 |
| TRISC | TRISC7 | TRISC6 | TRISC5 | TRISC4 | TRISC3 | TRISC2 | TRISC1 | TRISC0 | 55 |
| TXREG | EUSART Transmit Data Register | | | | | | | | 157 |
| TXSTA | CSRC | TX9 | TXEN | SYNC | SENDB | BRGH | TRMT | TX9D | 164 |

**Legend:** x = unknown, – = unimplemented read as '0'. Shaded cells are not used for Asynchronous Transmission.

### 12.1.2 EUSART ASYNCHRONOUS RECEIVER

The Asynchronous mode is typically used in RS-232 systems. The receiver block diagram is shown in Figure 12-2. The data is received on the RX/DT pin and drives the data recovery block. The data recovery block is actually a high-speed shifter operating at 16 times the baud rate, whereas the serial Receive Shift Register (RSR) operates at the bit rate. When all 8 or 9 bits of the character have been shifted in, they are immediately transferred to a two character First-In-First-Out (FIFO) memory. The FIFO buffering allows reception of two complete characters and the start of a third character before software must start servicing the EUSART receiver. The FIFO and RSR registers are not directly accessible by software. Access to the received data is via the RCREG register.

#### 12.1.2.1 Enabling the Receiver

The EUSART receiver is enabled for asynchronous operation by configuring the following three control bits:

- CREN = 1
- SYNC = 0
- SPEN = 1

All other EUSART control bits are assumed to be in their default state.

Setting the CREN bit of the RCSTA register enables the receiver circuitry of the EUSART. Clearing the SYNC bit of the TXSTA register configures the EUSART for asynchronous operation. Setting the SPEN bit of the RCSTA register enables the EUSART and automatically configures the RX/DT I/O pin as an input. If the RX/DT pin is shared with an analog peripheral the analog I/O function must be disabled by clearing the corresponding ANSEL bit.

| | |
|---|---|
| **Note:** | When the SPEN bit is set the TX/CK I/O pin is automatically configured as an output, regardless of the state of the corresponding TRIS bit and whether or not the EUSART transmitter is enabled. The PORT latch is disconnected from the output driver so it is not possible to use the TX/CK pin as a general purpose output. |

#### 12.1.2.2 Receiving Data

The receiver data recovery circuit initiates character reception on the falling edge of the first bit. The first bit, also known as the Start bit, is always a zero. The data recovery circuit counts one-half bit time to the center of the Start bit and verifies that the bit is still a zero. If it is not a zero then the data recovery circuit aborts character reception, without generating an error, and resumes looking for the falling edge of the Start bit. If the Start bit zero verification succeeds then the data recovery circuit counts a full bit time to the center of the next bit. The bit is then sampled by a majority detect circuit and the resulting '0' or '1' is shifted into the RSR. This repeats until all data bits have been sampled and shifted into the RSR. One final bit time is measured and the level sampled. This is the Stop bit, which is always a '1'. If the data recovery circuit samples a '0' in the Stop bit position then a framing error is set for this character, otherwise the framing error is cleared for this character. See **Section 12.1.2.4 "Receive Framing Error"** for more information on framing errors.

Immediately after all data bits and the Stop bit have been received, the character in the RSR is transferred to the EUSART receive FIFO and the RCIF interrupt flag bit of the PIR1 register is set. The top character in the FIFO is transferred out of the FIFO by reading the RCREG register.

| | |
|---|---|
| **Note:** | If the receive FIFO is overrun, no additional characters will be received until the overrun condition is cleared. See **Section 12.1.2.5 "Receive Overrun Error"** for more information on overrun errors. |

#### 12.1.2.3 Receive Interrupts

The RCIF interrupt flag bit of the PIR1 register is set whenever the EUSART receiver is enabled and there is an unread character in the receive FIFO. The RCIF interrupt flag bit is read-only, it cannot be set or cleared by software.

RCIF interrupts are enabled by setting all of the following bits:

- RCIE interrupt enable bit of the PIE1 register
- PEIE peripheral interrupt enable bit of the INTCON register
- GIE global interrupt enable bit of the INTCON register

The RCIF interrupt flag bit will be set when there is an unread character in the FIFO, regardless of the state of interrupt enable bits.


© 2006-2012 Microchip Technology Inc.

#### 12.1.2.4 Receive Framing Error

Each character in the receive FIFO buffer has a corresponding framing error Status bit. A framing error indicates that a Stop bit was not seen at the expected time. The framing error status is accessed via the FERR bit of the RCSTA register. The FERR bit represents the status of the top unread character in the receive FIFO. Therefore, the FERR bit must be read before reading the RCREG.

The FERR bit is read-only and only applies to the top unread character in the receive FIFO. A framing error (FERR = 1) does not preclude reception of additional characters. It is not necessary to clear the FERR bit. Reading the next character from the FIFO buffer will advance the FIFO to the next character and the next corresponding framing error.

The FERR bit can be forced clear by clearing the SPEN bit of the RCSTA register which resets the EUSART. Clearing the CREN bit of the RCSTA register does not affect the FERR bit. A framing error by itself does not generate an interrupt.

| | |
|---|---|
| **Note:** | If all receive characters in the receive FIFO have framing errors, repeated reads of the RCREG will not clear the FERR bit. |

#### 12.1.2.5 Receive Overrun Error

The receive FIFO buffer can hold two characters. An overrun error will be generated If a third character, in its entirety, is received before the FIFO is accessed. When this happens the OERR bit of the RCSTA register is set. The characters already in the FIFO buffer can be read but no additional characters will be received until the error is cleared. The error must be cleared by either clearing the CREN bit of the RCSTA register or by resetting the EUSART by clearing the SPEN bit of the RCSTA register.

#### 12.1.2.6 Receiving 9-Bit Characters

The EUSART supports 9-bit character reception. When the RX9 bit of the RCSTA register is set the EUSART will shift 9 bits into the RSR for each character received. The RX9D bit of the RCSTA register is the ninth and Most Significant data bit of the top unread character in the receive FIFO. When reading 9-bit data from the receive FIFO buffer, the RX9D data bit must be read before reading the 8 Least Significant bits from the RCREG.

#### 12.1.2.7 Address Detection

A special Address Detection mode is available for use when multiple receivers share the same transmission line, such as in RS-485 systems. Address detection is enabled by setting the ADDEN bit of the RCSTA register.

Address detection requires 9-bit character reception. When address detection is enabled, only characters with the ninth data bit set will be transferred to the receive FIFO buffer, thereby setting the RCIF interrupt bit. All other characters will be ignored.

Upon receiving an address character, user software determines if the address matches its own. Upon address match, user software must disable address detection by clearing the ADDEN bit before the next Stop bit occurs. When user software detects the end of the message, determined by the message protocol used, software places the receiver back into the Address Detection mode by setting the ADDEN bit.

#### 12.1.2.8 Asynchronous Reception Setup:

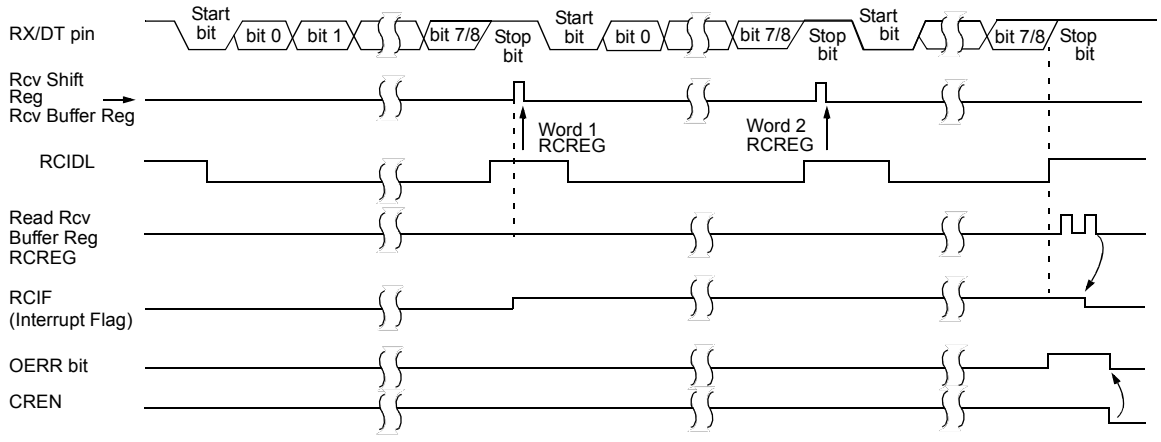
1. Initialize the SPBRGH, SPBRG register pair and the BRGH and BRG16 bits to achieve the desired baud rate (see **Section 12.3 "EUSART Baud Rate Generator (BRG)"**).
2. Enable the serial port by setting the SPEN bit. The SYNC bit must be clear for asynchronous operation.
3. If interrupts are desired, set the RCIE bit of the PIE1 register and the GIE and PEIE bits of the INTCON register.
4. If 9-bit reception is desired, set the RX9 bit.
5. Enable reception by setting the CREN bit.
6. The RCIF interrupt flag bit will be set when a character is transferred from the RSR to the receive buffer. An interrupt will be generated if the RCIE interrupt enable bit was also set.
7. Read the RCSTA register to get the error flags and, if 9-bit data reception is enabled, the ninth data bit.
8. Get the received 8 Least Significant data bits from the receive buffer by reading the RCREG register.
9. If an overrun occurred, clear the OERR flag by clearing the CREN receiver enable bit.

#### 12.1.2.9 9-bit Address Detection Mode Setup

This mode would typically be used in RS-485 systems. To set up an Asynchronous Reception with Address Detect Enable:

1. Initialize the SPBRGH, SPBRG register pair and the BRGH and BRG16 bits to achieve the desired baud rate (see **Section 12.3 "EUSART Baud Rate Generator (BRG)"**).
2. Enable the serial port by setting the SPEN bit. The SYNC bit must be clear for asynchronous operation.
3. If interrupts are desired, set the RCIE bit of the PIE1 register and the GIE and PEIE bits of the INTCON register.
4. Enable 9-bit reception by setting the RX9 bit.
5. Enable address detection by setting the ADDEN bit.
6. Enable reception by setting the CREN bit.
7. The RCIF interrupt flag bit will be set when a character with the ninth bit set is transferred from the RSR to the receive buffer. An interrupt will be generated if the RCIE interrupt enable bit was also set.
8. Read the RCSTA register to get the error flags. The ninth data bit will always be set.
9. Get the received 8 Least Significant data bits from the receive buffer by reading the RCREG register. Software determines if this is the device's address.
10. If an overrun occurred, clear the OERR flag by clearing the CREN receiver enable bit.
11. If the device has been addressed, clear the ADDEN bit to allow all received data into the receive buffer and generate interrupts.

**FIGURE 12-5: ASYNCHRONOUS RECEPTION**




**Note:** This timing diagram shows three words appearing on the RX input. The RCREG (receive buffer) is read after the third word, causing the OERR (overrun) bit to be set.


© 2006-2012 Microchip Technology Inc.

**TABLE 12-2: REGISTERS ASSOCIATED WITH ASYNCHRONOUS RECEPTION**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|
| BAUDCTL | ABDOVF | RCIDL | — | SCKP | BRG16 | — | WUE | ABDEN | 166 |
| INTCON | GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | RBIF | 33 |
| PIE1 | — | ADIE | RCIE | TXIE | SSPIE | CCP1IE | TMR2IE | TMR1IE | 34 |
| PIR1 | — | ADIF | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | 36 |
| RCREG | EUSART Receive Data Register | | | | | | | | 162 |
| RCSTA | SPEN | RX9 | SREN | CREN | ADDEN | FERR | OERR | RX9D | 165 |
| SPBRG | BRG7 | BRG6 | BRG5 | BRG4 | BRG3 | BRG2 | BRG1 | BRG0 | 167 |
| SPBRGH | BRG15 | BRG14 | BRG13 | BRG12 | BRG11 | BRG10 | BRG9 | BRG8 | 167 |
| TRISC | TRISC7 | TRISC6 | TRISC5 | TRISC4 | TRISC3 | TRISC2 | TRISC1 | TRISC0 | 55 |
| TXREG | EUSART Transmit Data Register | | | | | | | | 157 |
| TXSTA | CSRC | TX9 | TXEN | SYNC | SENDB | BRGH | TRMT | TX9D | 164 |

**Legend:** x = unknown, – = unimplemented read as '0'. Shaded cells are not used for Asynchronous Reception.

## 12.2 Clock Accuracy with Asynchronous Operation

The factory calibrates the Internal Oscillator block output (INTOSC). However, the INTOSC frequency may drift as VDD or temperature changes, and this directly affects the asynchronous baud rate. Two methods may be used to adjust the baud rate clock, but both require a reference clock source of some kind.

The first (preferred) method uses the OSCTUNE register to adjust the INTOSC output. Adjusting the value in the OSCTUNE register allows for fine resolution changes to the system clock source. See **Section 4.5 "Internal Clock Modes"** for more information.

The other method adjusts the value in the Baud Rate Generator. This can be done automatically with the Auto-Baud Detect feature (see **Section 12.3.1 "Auto-Baud Detect"**). There may not be fine enough resolution when adjusting the Baud Rate Generator to compensate for a gradual change in the peripheral clock frequency.

## REGISTER DEFINITIONS: EUSART CONTROL

### REGISTER 12-1: TXSTA: TRANSMIT STATUS AND CONTROL REGISTER

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R-1 | R/W-0 |
|---|---|---|---|---|---|---|---|
| CSRC | TX9 | TXEN(1) | SYNC | SENDB | BRGH | TRMT | TX9D |
| 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 **CSRC:** Clock Source Select bit
Asynchronous mode:
Don't care
Synchronous mode:
1 = Master mode (clock generated internally from BRG)
0 = Slave mode (clock from external source)

bit 6 **TX9:** 9-bit Transmit Enable bit
1 = Selects 9-bit transmission
0 = Selects 8-bit transmission

bit 5 **TXEN:** Transmit Enable bit(1)
1 = Transmit enabled
0 = Transmit disabled

bit 4 **SYNC:** EUSART Mode Select bit
1 = Synchronous mode
0 = Asynchronous mode

bit 3 **SENDB:** Send Break Character bit
Asynchronous mode:
1 = Send Sync Break on next transmission (cleared by hardware upon completion)
0 = Sync Break transmission completed
Synchronous mode:
Don't care

bit 2 **BRGH:** High Baud Rate Select bit
Asynchronous mode:
1 = High speed
0 = Low speed
Synchronous mode:
Unused in this mode

bit 1 **TRMT:** Transmit Shift Register Status bit
1 = TSR empty
0 = TSR full

bit 0 **TX9D:** Ninth bit of Transmit Data
Can be address/data bit or a parity bit.

**Note 1:** SREN/CREN overrides TXEN in Sync mode.


© 2006-2012 Microchip Technology Inc.

**REGISTER 12-2: RCSTA: RECEIVE STATUS AND CONTROL REGISTER(1)**

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R-0 | R-0 | R-x |
|---|---|---|---|---|---|---|---|
| SPEN | RX9 | SREN | CREN | ADDEN | FERR | OERR | RX9D |
| bit 7 | | | | | | | bit 0 |

| **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 **SPEN:** Serial Port Enable bit
1 = Serial port enabled (configures RX/DT and TX/CK pins as serial port pins)
0 = Serial port disabled (held in Reset)

bit 6 **RX9:** 9-bit Receive Enable bit
1 = Selects 9-bit reception
0 = Selects 8-bit reception

bit 5 **SREN:** Single Receive Enable bit
Asynchronous mode:
Don't care
Synchronous mode – Master:
1 = Enables single receive
0 = Disables single receive
This bit is cleared after reception is complete.
Synchronous mode – Slave
Don't care

bit 4 **CREN:** Continuous Receive Enable bit
Asynchronous mode:
1 = Enables receiver
0 = Disables receiver
Synchronous mode:
1 = Enables continuous receive until enable bit CREN is cleared (CREN overrides SREN)
0 = Disables continuous receive

bit 3 **ADDEN:** Address Detect Enable bit
Asynchronous mode 9-bit (RX9 = 1):
1 = Enables address detection, enable interrupt and load the receive buffer when RSR<8> is set
0 = Disables address detection, all bytes are received and ninth bit can be used as parity bit
Asynchronous mode 8-bit (RX9 = 0):
Don't care

bit 2 **FERR:** Framing Error bit
1 = Framing error (can be updated by reading RCREG register and receive next valid byte)
0 = No framing error

bit 1 **OERR:** Overrun Error bit
1 = Overrun error (can be cleared by clearing bit CREN)
0 = No overrun error

bit 0 **RX9D:** Ninth bit of Received Data
This can be address/data bit or a parity bit and must be calculated by user firmware.

**REGISTER 12-3: BAUDCTL: BAUD RATE CONTROL REGISTER**

| R-0 | R-1 | U-0 | R/W-0 | R/W-0 | U-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| ABDOVF | RCIDL | — | SCKP | BRG16 | — | WUE | ABDEN |
| bit 7 | | | | | | | bit 0 |

| **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 **ABDOVF:** Auto-Baud Detect Overflow bit
Asynchronous mode:
1 = Auto-baud timer overflowed
0 = Auto-baud timer did not overflow
Synchronous mode:
Don't care

bit 6 **RCIDL**: Receive Idle Flag bit
Asynchronous mode:
1 = Receiver is Idle
0 = Start bit has been received and the receiver is receiving
Synchronous mode:
Don't care

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

bit 4 **SCKP**: Synchronous Clock Polarity Select bit
Asynchronous mode:
1 = Transmit inverted data to the RB7/TX/CK pin
0 = Transmit non-inverted data to the RB7/TX/CK pin
Synchronous mode:
1 = Data is clocked on rising edge of the clock
0 = Data is clocked on falling edge of the clock

bit 3 **BRG16:** 16-bit Baud Rate Generator bit
1 = 16-bit Baud Rate Generator is used
0 = 8-bit Baud Rate Generator is used

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

bit 1 **WUE:** Wake-up Enable bit
Asynchronous mode:
1 = Receiver is waiting for a falling edge. No character will be received byte RCIF will be set. WUE will automatically clear after RCIF is set.
0 = Receiver is operating normally
Synchronous mode:
Don't care

bit 0 **ABDEN**: Auto-Baud Detect Enable bit
Asynchronous mode:
1 = Auto-Baud Detect mode is enabled (clears when auto-baud is complete)
0 = Auto-Baud Detect mode is disabled
Synchronous mode:
Don't care


© 2006-2012 Microchip Technology Inc.

## 12.3 EUSART Baud Rate Generator (BRG)

The Baud Rate Generator (BRG) is an 8-bit or 16-bit timer that is dedicated to the support of both the asynchronous and synchronous EUSART operation. By default, the BRG operates in 8-bit mode. Setting the BRG16 bit of the BAUDCTL register selects 16-bit mode.

The SPBRGH, SPBRG register pair determines the period of the free running baud rate timer. In Asynchronous mode the multiplier of the baud rate period is determined by both the BRGH bit of the TXSTA register and the BRG16 bit of the BAUDCTL register. In Synchronous mode, the BRGH bit is ignored.

Table 12-3 contains the formulas for determining the baud rate. Example 12-1 provides a sample calculation for determining the baud rate and baud rate error.

Typical baud rates and error values for various asynchronous modes have been computed for your convenience and are shown in Table 12-3. It may be advantageous to use the high baud rate (BRGH = 1), or the 16-bit BRG (BRG16 = 1) to reduce the baud rate error. The 16-bit BRG mode is used to achieve slow baud rates for fast oscillator frequencies.

Writing a new value to the SPBRGH, SPBRG register pair causes the BRG timer to be reset (or cleared). This ensures that the BRG does not wait for a timer overflow before outputting the new baud rate.

If the system clock is changed during an active receive operation, a receive error or data loss may result. To avoid this problem, check the status of the RCIDL bit to make sure that the receive operation is Idle before changing the system clock.

### EXAMPLE 12-1: CALCULATING BAUD RATE ERROR

For a device with FOSC of 16 MHz, desired baud rate of 9600, Asynchronous mode, 8-bit BRG:

$$Desired\ Baud\ Rate = \frac{FOSC}{64([SPBRGH:SPBRG] + 1)}$$

Solving for SPBRGH:SPBRG:

$$X = \frac{\frac{FOSC}{Desired\ Baud\ Rate}}{64} - 1$$

$$= \frac{\frac{16000000}{9600}}{64} - 1$$

$$= [25.042] = 25$$

$$Calculated\ Baud\ Rate = \frac{16000000}{64(25 + 1)}$$

$$= 9615$$

$$Error = \frac{Calc.\ Baud\ Rate - Desired\ Baud\ Rate}{Desired\ Baud\ Rate}$$

$$= \frac{(9615 - 9600)}{9600} = 0.16\%$$

### TABLE 12-3: BAUD RATE FORMULAS

| Configuration Bits | | | BRG/EUSART Mode | Baud Rate Formula |
|---|---|---|---|---|
| SYNC | BRG16 | BRGH | | |
| 0 | 0 | 0 | 8-bit/Asynchronous | FOSC/[64 (n+1)] |
| 0 | 0 | 1 | 8-bit/Asynchronous | FOSC/[16 (n+1)] |
| 0 | 1 | 0 | 16-bit/Asynchronous | |
| 0 | 1 | 1 | 16-bit/Asynchronous | FOSC/[4 (n+1)] |
| 1 | 0 | x | 8-bit/Synchronous | |
| 1 | 1 | x | 16-bit/Synchronous | |

**Legend:** x = Don't care, n = value of SPBRGH, SPBRG register pair

### TABLE 12-4: REGISTERS ASSOCIATED WITH THE BAUD RATE GENERATOR

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|
| BAUDCTL | ABDOVF | RCIDL | — | SCKP | BRG16 | — | WUE | ABDEN | 166 |
| RCSTA | SPEN | RX9 | SREN | CREN | ADDEN | FERR | OERR | RX9D | 165 |
| SPBRG | BRG7 | BRG6 | BRG5 | BRG4 | BRG3 | BRG2 | BRG1 | BRG0 | 167 |
| SPBRGH | BRG15 | BRG14 | BRG13 | BRG12 | BRG11 | BRG10 | BRG9 | BRG8 | 167 |
| TXSTA | CSRC | TX9 | TXEN | SYNC | SENDB | BRGH | TRMT | TX9D | 164 |

**Legend:** x = unknown, – = unimplemented read as '0'. Shaded cells are not used for the Baud Rate Generator.

© 2006-2012 Microchip Technology Inc.

**TABLE 12-5: BAUD RATES FOR ASYNCHRONOUS MODES**

| BAUD RATE | SYNC = 0, BRGH = 0, BRG16 = 0 | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Fosc = 20.000 MHz | | | Fosc = 18.432 MHz | | | Fosc = 11.0592 MHz | | | Fosc = 8.000 MHz | | |
| | Actual Rate | % Error | SPBRG value (decimal) | Actual Rate | % Error | SPBRG value (decimal) | Actual Rate | % Error | SPBRG value (decimal) | Actual Rate | % Error | SPBRG value (decimal) |
| 300 | — | — | — | — | — | — | — | — | — | — | — | — |
| 1200 | 1221 | 1.73 | 255 | 1200 | 0.00 | 239 | 1200 | 0.00 | 143 | 1202 | 0.16 | 103 |
| 2400 | 2404 | 0.16 | 129 | 2400 | 0.00 | 119 | 2400 | 0.00 | 71 | 2404 | 0.16 | 51 |
| 9600 | 9470 | -1.36 | 32 | 9600 | 0.00 | 29 | 9600 | 0.00 | 17 | 9615 | 0.16 | 12 |
| 10417 | 10417 | 0.00 | 29 | 10286 | -1.26 | 27 | 10165 | -2.42 | 16 | 10417 | 0.00 | 11 |
| 19.2k | 19.53k | 1.73 | 15 | 19.20k | 0.00 | 14 | 19.20k | 0.00 | 8 | — | — | — |
| 57.6k | — | — | — | 57.60k | 0.00 | 7 | 57.60k | 0.00 | 2 | — | — | — |
| 115.2k | — | — | — | — | — | — | — | — | — | — | — | — |

| BAUD RATE | SYNC = 0, BRGH = 0, BRG16 = 0 | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Fosc = 4.000 MHz | | | Fosc = 3.6864 MHz | | | Fosc = 2.000 MHz | | | Fosc = 1.000 MHz | | |
| | Actual Rate | % Error | SPBRG value (decimal) | Actual Rate | % Error | SPBRG value (decimal) | Actual Rate | % Error | SPBRG value (decimal) | Actual Rate | % Error | SPBRG value (decimal) |
| 300 | 300 | 0.16 | 207 | 300 | 0.00 | 191 | 300 | 0.16 | 103 | 300 | 0.16 | 51 |
| 1200 | 1202 | 0.16 | 51 | 1200 | 0.00 | 47 | 1202 | 0.16 | 25 | 1202 | 0.16 | 12 |
| 2400 | 2404 | 0.16 | 25 | 2400 | 0.00 | 23 | 2404 | 0.16 | 12 | — | — | — |
| 9600 | — | — | — | 9600 | 0.00 | 5 | — | — | — | — | — | — |
| 10417 | 10417 | 0.00 | 5 | — | — | — | 10417 | 0.00 | 2 | — | — | — |
| 19.2k | — | — | — | 19.20k | 0.00 | 2 | — | — | — | — | — | — |
| 57.6k | — | — | — | 57.60k | 0.00 | 0 | — | — | — | — | — | — |
| 115.2k | — | — | — | — | — | — | — | — | — | — | — | — |

| BAUD RATE | SYNC = 0, BRGH = 1, BRG16 = 0 | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Fosc = 20.000 MHz | | | Fosc = 18.432 MHz | | | Fosc = 11.0592 MHz | | | Fosc = 8.000 MHz | | |
| | Actual Rate | % Error | SPBRG value (decimal) | Actual Rate | % Error | SPBRG value (decimal) | Actual Rate | % Error | SPBRG value (decimal) | Actual Rate | % Error | SPBRG value (decimal) |
| 300 | — | — | — | — | — | — | — | — | — | — | — | — |
| 1200 | — | — | — | — | — | — | — | — | — | — | — | — |
| 2400 | — | — | — | — | — | — | — | — | — | 2404 | 0.16 | 207 |
| 9600 | 9615 | 0.16 | 129 | 9600 | 0.00 | 119 | 9600 | 0.00 | 71 | 9615 | 0.16 | 51 |
| 10417 | 10417 | 0.00 | 119 | 10378 | -0.37 | 110 | 10473 | 0.53 | 65 | 10417 | 0.00 | 47 |
| 19.2k | 19.23k | 0.16 | 64 | 19.20k | 0.00 | 59 | 19.20k | 0.00 | 35 | 19231 | 0.16 | 25 |
| 57.6k | 56.82k | -1.36 | 21 | 57.60k | 0.00 | 19 | 57.60k | 0.00 | 11 | 55556 | -3.55 | 8 |
| 115.2k | 113.64k | -1.36 | 10 | 115.2k | 0.00 | 9 | 115.2k | 0.00 | 5 | — | — | — |


© 2006-2012 Microchip Technology Inc.

**TABLE 12-5: BAUD RATES FOR ASYNCHRONOUS MODES (CONTINUED)**

| BAUD RATE | SYNC = 0, BRGH = 1, BRG16 = 0 | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Fosc = 4.000 MHz | | | Fosc = 3.6864 MHz | | | Fosc = 2.000 MHz | | | Fosc = 1.000 MHz | | |
| | Actual Rate | % Error | SPBRG value (decimal) | Actual Rate | % Error | SPBRG value (decimal) | Actual Rate | % Error | SPBRG value (decimal) | Actual Rate | % Error | SPBRG value (decimal) |
| 300 | — | — | — | — | — | — | — | — | — | 300 | 0.16 | 207 |
| 1200 | 1202 | 0.16 | 207 | 1200 | 0.00 | 191 | 1202 | 0.16 | 103 | 1202 | 0.16 | 51 |
| 2400 | 2404 | 0.16 | 103 | 2400 | 0.00 | 95 | 2404 | 0.16 | 51 | 2404 | 0.16 | 25 |
| 9600 | 9615 | 0.16 | 25 | 9600 | 0.00 | 23 | 9615 | 0.16 | 12 | — | — | — |
| 10417 | 10417 | 0.00 | 23 | 10473 | 0.53 | 21 | 10417 | 0.00 | 11 | 10417 | 0.00 | 5 |
| 19.2k | 19.23k | 0.16 | 12 | 19.2k | 0.00 | 11 | — | — | — | — | — | — |
| 57.6k | — | — | — | 57.60k | 0.00 | 3 | — | — | — | — | — | — |
| 115.2k | — | — | — | 115.2k | 0.00 | 1 | — | — | — | — | — | — |

| BAUD RATE | SYNC = 0, BRGH = 0, BRG16 = 1 | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Fosc = 20.000 MHz | | | Fosc = 18.432 MHz | | | Fosc = 11.0592 MHz | | | Fosc = 8.000 MHz | | |
| | Actual Rate | % Error | SPBRG value (decimal) | Actual Rate | % Error | SPBRG value (decimal) | Actual Rate | % Error | SPBRG value (decimal) | Actual Rate | % Error | SPBRG value (decimal) |
| 300 | 300.0 | -0.01 | 4166 | 300.0 | 0.00 | 3839 | 300.0 | 0.00 | 2303 | 299.9 | -0.02 | 1666 |
| 1200 | 1200 | -0.03 | 1041 | 1200 | 0.00 | 959 | 1200 | 0.00 | 575 | 1199 | -0.08 | 416 |
| 2400 | 2399 | -0.03 | 520 | 2400 | 0.00 | 479 | 2400 | 0.00 | 287 | 2404 | 0.16 | 207 |
| 9600 | 9615 | 0.16 | 129 | 9600 | 0.00 | 119 | 9600 | 0.00 | 71 | 9615 | 0.16 | 51 |
| 10417 | 10417 | 0.00 | 119 | 10378 | -0.37 | 110 | 10473 | 0.53 | 65 | 10417 | 0.00 | 47 |
| 19.2k | 19.23k | 0.16 | 64 | 19.20k | 0.00 | 59 | 19.20k | 0.00 | 35 | 19.23k | 0.16 | 25 |
| 57.6k | 56.818 | -1.36 | 21 | 57.60k | 0.00 | 19 | 57.60k | 0.00 | 11 | 55556 | -3.55 | 8 |
| 115.2k | 113.636 | -1.36 | 10 | 115.2k | 0.00 | 9 | 115.2k | 0.00 | 5 | — | — | — |

| BAUD RATE | SYNC = 0, BRGH = 0, BRG16 = 1 | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Fosc = 4.000 MHz | | | Fosc = 3.6864 MHz | | | Fosc = 2.000 MHz | | | Fosc = 1.000 MHz | | |
| | Actual Rate | % Error | SPBRG value (decimal) | Actual Rate | % Error | SPBRG value (decimal) | Actual Rate | % Error | SPBRG value (decimal) | Actual Rate | % Error | SPBRG value (decimal) |
| 300 | 300.1 | 0.04 | 832 | 300.0 | 0.00 | 767 | 299.8 | -0.108 | 416 | 300.5 | 0.16 | 207 |
| 1200 | 1202 | 0.16 | 207 | 1200 | 0.00 | 191 | 1202 | 0.16 | 103 | 1202 | 0.16 | 51 |
| 2400 | 2404 | 0.16 | 103 | 2400 | 0.00 | 95 | 2404 | 0.16 | 51 | 2404 | 0.16 | 25 |
| 9600 | 9615 | 0.16 | 25 | 9600 | 0.00 | 23 | 9615 | 0.16 | 12 | — | — | — |
| 10417 | 10417 | 0.00 | 23 | 10473 | 0.53 | 21 | 10417 | 0.00 | 11 | 10417 | 0.00 | 5 |
| 19.2k | 19.23k | 0.16 | 12 | 19.20k | 0.00 | 11 | — | — | — | — | — | — |
| 57.6k | — | — | — | 57.60k | 0.00 | 3 | — | — | — | — | — | — |
| 115.2k | — | — | — | 115.2k | 0.00 | 1 | — | — | — | — | — | — |

**TABLE 12-5: BAUD RATES FOR ASYNCHRONOUS MODES (CONTINUED)**

| BAUD RATE | SYNC = 0, BRGH = 1, BRG16 = 1 or SYNC = 1, BRG16 = 1 | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Fosc = 20.000 MHz | | | Fosc = 18.432 MHz | | | Fosc = 11.0592 MHz | | | Fosc = 8.000 MHz | | |
| | Actual Rate | % Error | SPBRG value (decimal) | Actual Rate | % Error | SPBRG value (decimal) | Actual Rate | % Error | SPBRG value (decimal) | Actual Rate | % Error | SPBRG value (decimal) |
| 300 | 300.0 | 0.00 | 16665 | 300.0 | 0.00 | 15359 | 300.0 | 0.00 | 9215 | 300.0 | 0.00 | 6666 |
| 1200 | 1200 | -0.01 | 4166 | 1200 | 0.00 | 3839 | 1200 | 0.00 | 2303 | 1200 | -0.02 | 1666 |
| 2400 | 2400 | 0.02 | 2082 | 2400 | 0.00 | 1919 | 2400 | 0.00 | 1151 | 2401 | 0.04 | 832 |
| 9600 | 9597 | -0.03 | 520 | 9600 | 0.00 | 479 | 9600 | 0.00 | 287 | 9615 | 0.16 | 207 |
| 10417 | 10417 | 0.00 | 479 | 10425 | 0.08 | 441 | 10433 | 0.16 | 264 | 10417 | 0 | 191 |
| 19.2k | 19.23k | 0.16 | 259 | 19.20k | 0.00 | 239 | 19.20k | 0.00 | 143 | 19.23k | 0.16 | 103 |
| 57.6k | 57.47k | -0.22 | 86 | 57.60k | 0.00 | 79 | 57.60k | 0.00 | 47 | 57.14k | -0.79 | 34 |
| 115.2k | 116.3k | 0.94 | 42 | 115.2k | 0.00 | 39 | 115.2k | 0.00 | 23 | 117.6k | 2.12 | 16 |

| BAUD RATE | SYNC = 0, BRGH = 1, BRG16 = 1 or SYNC = 1, BRG16 = 1 | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Fosc = 4.000 MHz | | | Fosc = 3.6864 MHz | | | Fosc = 2.000 MHz | | | Fosc = 1.000 MHz | | |
| | Actual Rate | % Error | SPBRG value (decimal) | Actual Rate | % Error | SPBRG value (decimal) | Actual Rate | % Error | SPBRG value (decimal) | Actual Rate | % Error | SPBRG value (decimal) |
| 300 | 300.0 | 0.01 | 3332 | 300.0 | 0.00 | 3071 | 299.9 | -0.02 | 1666 | 300.1 | 0.04 | 832 |
| 1200 | 1200 | 0.04 | 832 | 1200 | 0.00 | 767 | 1199 | -0.08 | 416 | 1202 | 0.16 | 207 |
| 2400 | 2398 | 0.08 | 416 | 2400 | 0.00 | 383 | 2404 | 0.16 | 207 | 2404 | 0.16 | 103 |
| 9600 | 9615 | 0.16 | 103 | 9600 | 0.00 | 95 | 9615 | 0.16 | 51 | 9615 | 0.16 | 25 |
| 10417 | 10417 | 0.00 | 95 | 10473 | 0.53 | 87 | 10417 | 0.00 | 47 | 10417 | 0.00 | 23 |
| 19.2k | 19.23k | 0.16 | 51 | 19.20k | 0.00 | 47 | 19.23k | 0.16 | 25 | 19.23k | 0.16 | 12 |
| 57.6k | 58.82k | 2.12 | 16 | 57.60k | 0.00 | 15 | 55.56k | -3.55 | 8 | — | — | — |
| 115.2k | 111.1k | -3.55 | 8 | 115.2k | 0.00 | 7 | — | — | — | — | — | — |


© 2006-2012 Microchip Technology Inc.

### 12.3.1 AUTO-BAUD DETECT

The EUSART module supports automatic detection and calibration of the baud rate.

In the Auto-Baud Detect (ABD) mode, the clock to the BRG is reversed. Rather than the BRG clocking the incoming RX signal, the RX signal is timing the BRG. The Baud Rate Generator is used to time the period of a received 55h (ASCII "U") which is the Sync character for the LIN bus. The unique feature of this character is that it has five rising edges including the Stop bit edge.

Setting the ABDEN bit of the BAUDCTL register starts the auto-baud calibration sequence (Figure 12-6). While the ABD sequence takes place, the EUSART state machine is held in Idle. On the first rising edge of the receive line, after the Start bit, the SPBRG begins counting up using the BRG counter clock as shown in Table 12-6. The fifth rising edge will occur on the RX pin at the end of the eighth bit period. At that time, an accumulated value totaling the proper BRG period is left in the SPBRGH, SPBRG register pair, the ABDEN bit is automatically cleared and the RCIF interrupt flag is set. The value in the RCREG needs to be read to clear the RCIF interrupt. RCREG content should be discarded. When calibrating for modes that do not use the SPBRGH register the user can verify that the SPBRG register did not overflow by checking for 00h in the SPBRGH register.

The BRG auto-baud clock is determined by the BRG16 and BRGH bits as shown in Table 12-6. During ABD, both the SPBRGH and SPBRG registers are used as a 16-bit counter, independent of the BRG16 bit setting. While calibrating the baud rate period, the SPBRGH and SPBRG registers are clocked at 1/8th the BRG base clock rate. The resulting byte measurement is the average bit time when clocked at full speed.

**Note 1:** If the WUE bit is set with the ABDEN bit, auto-baud detection will occur on the byte following the Break character (see **Section 12.3.2 "Auto-Wake-up on Break"**).

**2:** It is up to the user to determine that the incoming character baud rate is within the range of the selected BRG clock source. Some combinations of oscillator frequency and EUSART baud rates are not possible.

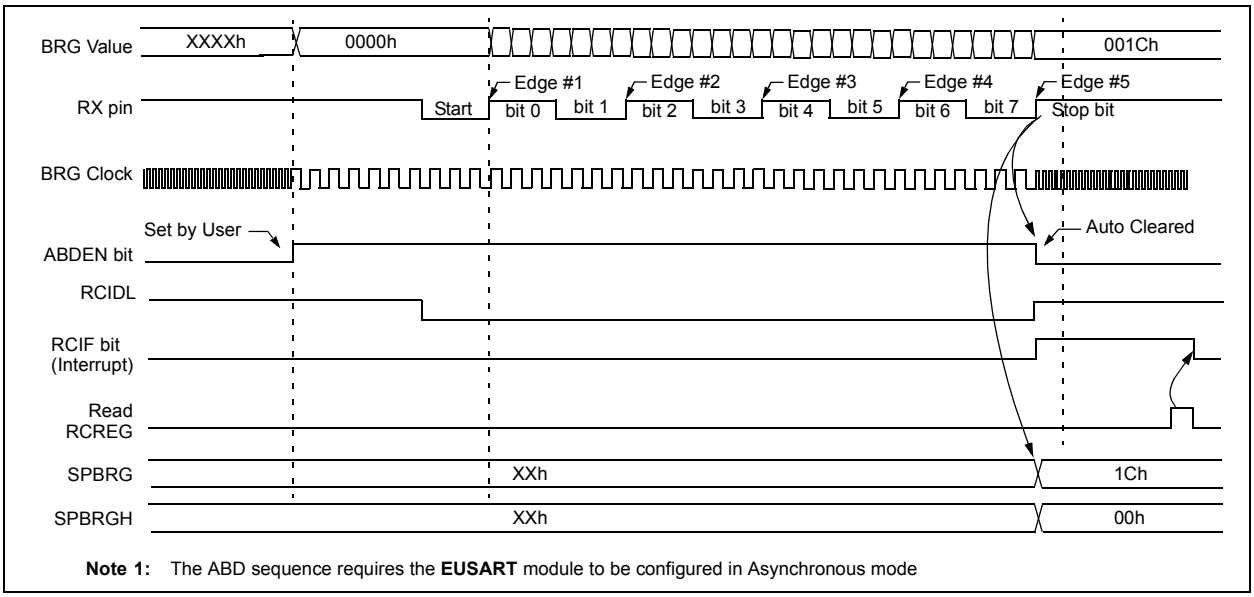
**3:** During the auto-baud process, the auto-baud counter starts counting at 1. Upon completion of the auto-baud sequence, to achieve maximum accuracy, subtract 1 from the SPBRGH:SPBRG register pair.

**TABLE 12-6: BRG COUNTER CLOCK RATES**

| BRG16 | BRGH | BRG Base Clock | BRG ABD Clock |
|---|---|---|---|
| 0 | 0 | FOSC/64 | FOSC/512 |
| 0 | 1 | FOSC/16 | FOSC/128 |
| 1 | 0 | FOSC/16 | FOSC/128 |
| 1 | 1 | FOSC/4 | FOSC/32 |

**Note:** During the ABD sequence, SPBRG and SPBRGH registers are both used as a 16-bit counter, independent of BRG16 setting.

**FIGURE 12-6: AUTOMATIC BAUD RATE CALIBRATION**




**Note 1:** The ABD sequence requires the **EUSART** module to be configured in Asynchronous mode

### 12.3.2 AUTO-WAKE-UP ON BREAK

During Sleep mode, all clocks to the EUSART are suspended. Because of this, the Baud Rate Generator is inactive and a proper character reception cannot be performed. The Auto-Wake-up feature allows the controller to wake-up due to activity on the RX/DT line. This feature is available only in Asynchronous mode.

The Auto-Wake-up feature is enabled by setting the WUE bit of the BAUDCTL register. Once set, the normal receive sequence on RX/DT is disabled, and the EUSART remains in an Idle state, monitoring for a wake-up event independent of the CPU mode. A wake-up event consists of a high-to-low transition on the RX/DT line. (This coincides with the start of a Sync Break or a wake-up signal character for the LIN protocol.)

The EUSART module generates an RCIF interrupt coincident with the wake-up event. The interrupt is generated synchronously to the Q clocks in normal CPU operating modes (Figure 12-7), and asynchronously if the device is in Sleep mode (Figure 12-8). The interrupt condition is cleared by reading the RCREG register.

The WUE bit is automatically cleared by the low-to-high transition on the RX line at the end of the Break. This signals to the user that the Break event is over. At this point, the EUSART module is in Idle mode waiting to receive the next character.

#### 12.3.2.1 Special Considerations

Break Character

To avoid character errors or character fragments during a wake-up event, the wake-up character must be all zeros.

When the wake-up is enabled the function works independent of the low time on the data stream. If the WUE bit is set and a valid non-zero character is received, the low time from the Start bit to the first rising edge will be interpreted as the wake-up event. The remaining bits in the character will be received as a fragmented character and subsequent characters can result in framing or overrun errors.

Therefore, the initial character in the transmission must be all '0's. This must be 10 or more bit times, 13-bit times recommended for LIN bus, or any number of bit times for standard RS-232 devices.

Oscillator Startup Time

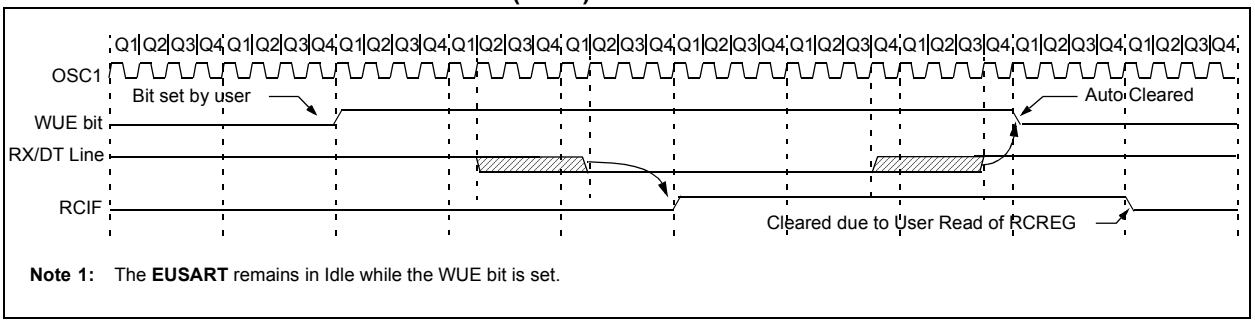
Oscillator start-up time must be considered, especially in applications using oscillators with longer start-up intervals (i.e., LP, XT or HS/PLL mode). The Sync Break (or wake-up signal) character must be of sufficient length, and be followed by a sufficient interval, to allow enough time for the selected oscillator to start and provide proper initialization of the EUSART.

WUE Bit

The wake-up event causes a receive interrupt by setting the RCIF bit. The WUE bit is cleared in hardware by a rising edge on RX/DT. The interrupt condition is then cleared in software by reading the RCREG register and discarding its contents.

To ensure that no actual data is lost, check the RCIDL bit to verify that a receive operation is not in process before setting the WUE bit. If a receive operation is not occurring, the WUE bit may then be set just prior to entering the Sleep mode.

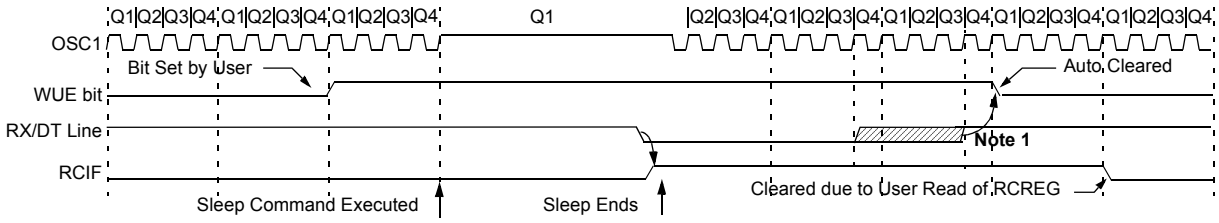
**FIGURE 12-7: AUTO-WAKE-UP BIT (WUE) TIMING DURING NORMAL OPERATION**




**Note 1:** The **EUSART** remains in Idle while the WUE bit is set.


© 2006-2012 Microchip Technology Inc.

**FIGURE 12-8: AUTO-WAKE-UP BIT (WUE) TIMINGS DURING SLEEP**




**Note 1:** If the wake-up event requires long oscillator warm-up time, the automatic clearing of the WUE bit can occur while the `stposc` signal is still active. This sequence should not depend on the presence of Q clocks.

**2:** The **EUSART** remains in Idle while the WUE bit is set.

### 12.3.3 BREAK CHARACTER SEQUENCE

The EUSART module has the capability of sending the special Break character sequences that are required by the LIN bus standard. A Break character consists of a Start bit, followed by 12 '0' bits and a Stop bit.

To send a Break character, set the SENDB and TXEN bits of the TXSTA register. The Break character transmission is then initiated by a write to the TXREG. The value of data written to TXREG will be ignored and all '0's will be transmitted.

The SENDB bit is automatically reset by hardware after the corresponding Stop bit is sent. This allows the user to preload the transmit FIFO with the next transmit byte following the Break character (typically, the Sync character in the LIN specification).

The TRMT bit of the TXSTA register indicates when the transmit operation is active or Idle, just as it does during normal transmission. See Figure 12-9 for the timing of the Break character sequence.

#### 12.3.3.1 Break and Sync Transmit Sequence

The following sequence will start a message frame header made up of a Break, followed by an auto-baud Sync byte. This sequence is typical of a LIN bus master.

1. Configure the EUSART for the desired mode.
2. Set the TXEN and SENDB bits to enable the Break sequence.
3. Load the TXREG with a dummy character to initiate transmission (the value is ignored).
4. Write '55h' to TXREG to load the Sync character into the transmit FIFO buffer.
5. After the Break has been sent, the SENDB bit is reset by hardware and the Sync character is then transmitted.

When the TXREG becomes empty, as indicated by the TXIF, the next data byte can be written to TXREG.

### 12.3.4 RECEIVING A BREAK CHARACTER

The Enhanced EUSART module can receive a Break character in two ways.

The first method to detect a Break character uses the FERR bit of the RCSTA register and the Received data as indicated by RCREG. The Baud Rate Generator is assumed to have been initialized to the expected baud rate.

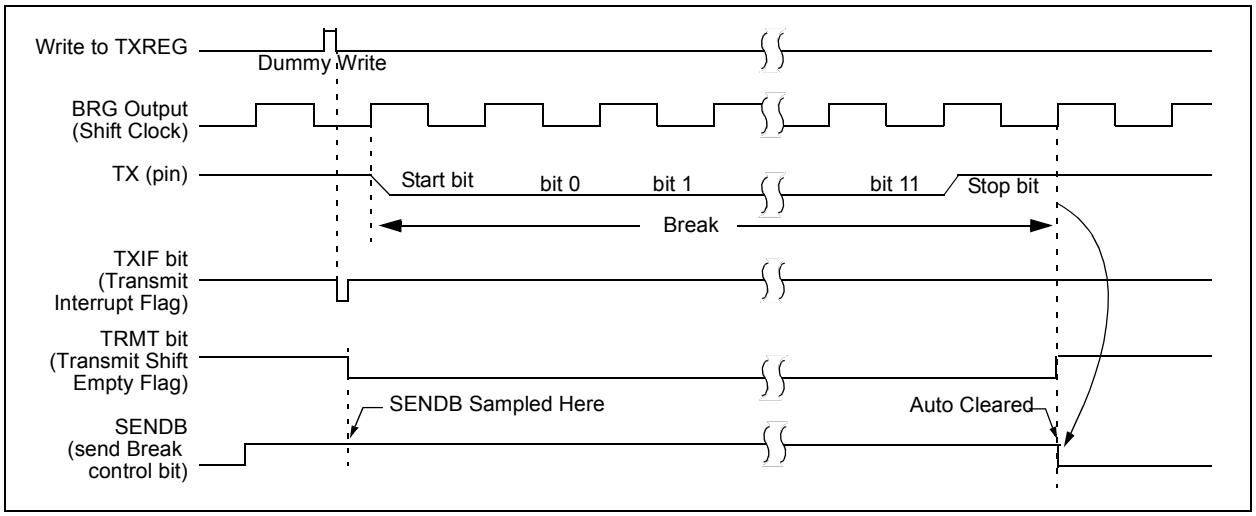
A Break character has been received when;

- RCIF bit is set
- FERR bit is set
- RCREG = 00h

The second method uses the Auto-Wake-up feature described in **Section 12.3.2 "Auto-Wake-up on Break"**. By enabling this feature, the EUSART will sample the next two transitions on RX/DT, cause an RCIF interrupt, and receive the next data byte followed by another interrupt.

Note that following a Break character, the user will typically want to enable the Auto-Baud Detect feature. For both methods, the user can set the ABDEN bit of the BAUDCTL register before placing the EUSART in Sleep mode.

**FIGURE 12-9: SEND BREAK CHARACTER SEQUENCE**





© 2006-2012 Microchip Technology Inc.

## 12.4 EUSART Synchronous Mode

Synchronous serial communications are typically used in systems with a single master and one or more slaves. The master device contains the necessary circuitry for baud rate generation and supplies the clock for all devices in the system. Slave devices can take advantage of the master clock by eliminating the internal clock generation circuitry.

There are two signal lines in Synchronous mode: a bidirectional data line and a clock line. Slaves use the external clock supplied by the master to shift the serial data into and out of their respective receive and transmit shift registers. Since the data line is bidirectional, synchronous operation is half-duplex only. Half-duplex refers to the fact that master and slave devices can receive and transmit data but not both simultaneously. The EUSART can operate as either a master or slave device.

Start and Stop bits are not used in synchronous transmissions.

### 12.4.1 SYNCHRONOUS MASTER MODE

The following bits are used to configure the EUSART for Synchronous Master operation:

- SYNC = 1
- CSRC = 1
- SREN = 0 (for transmit); SREN = 1 (for receive)
- CREN = 0 (for transmit); CREN = 1 (for receive)
- SPEN = 1

Setting the SYNC bit of the TXSTA register configures the device for synchronous operation. Setting the CSRC bit of the TXSTA register configures the device as a master. Clearing the SREN and CREN bits of the RCSTA register ensures that the device is in the Transmit mode, otherwise the device will be configured to receive. Setting the SPEN bit of the RCSTA register enables the EUSART. If the RX/DT or TX/CK pins are shared with an analog peripheral the analog I/O functions must be disabled by clearing the corresponding ANSEL bits.

#### 12.4.1.1 Master Clock

Synchronous data transfers use a separate clock line, which is synchronous with the data. A device configured as a master transmits the clock on the TX/CK line. The TX/CK pin output driver is automatically enabled when the EUSART is configured for synchronous transmit or receive operation. Serial data bits change on the leading edge to ensure they are valid at the trailing edge of each clock. One clock cycle is generated for each data bit. Only as many clock cycles are generated as there are data bits.

#### 12.4.1.2 Clock Polarity

A clock polarity option is provided for Microwire compatibility. Clock polarity is selected with the SCKP bit of the BAUDCTL register. Setting the SCKP bit sets the clock Idle state as high. When the SCKP bit is set, the data changes on the falling edge of each clock. Clearing the SCKP bit sets the Idle state as low. When the SCKP bit is cleared, the data changes on the rising edge of each clock.

#### 12.4.1.3 Synchronous Master Transmission

Data is transferred out of the device on the RX/DT pin. The RX/DT and TX/CK pin output drivers are automatically enabled when the EUSART is configured for synchronous master transmit operation.

A transmission is initiated by writing a character to the TXREG register. If the TSR still contains all or part of a previous character the new character data is held in the TXREG until the last bit of the previous character has been transmitted. If this is the first character, or the previous character has been completely flushed from the TSR, the data in the TXREG is immediately transferred to the TSR. The transmission of the character commences immediately following the transfer of the data to the TSR from the TXREG.

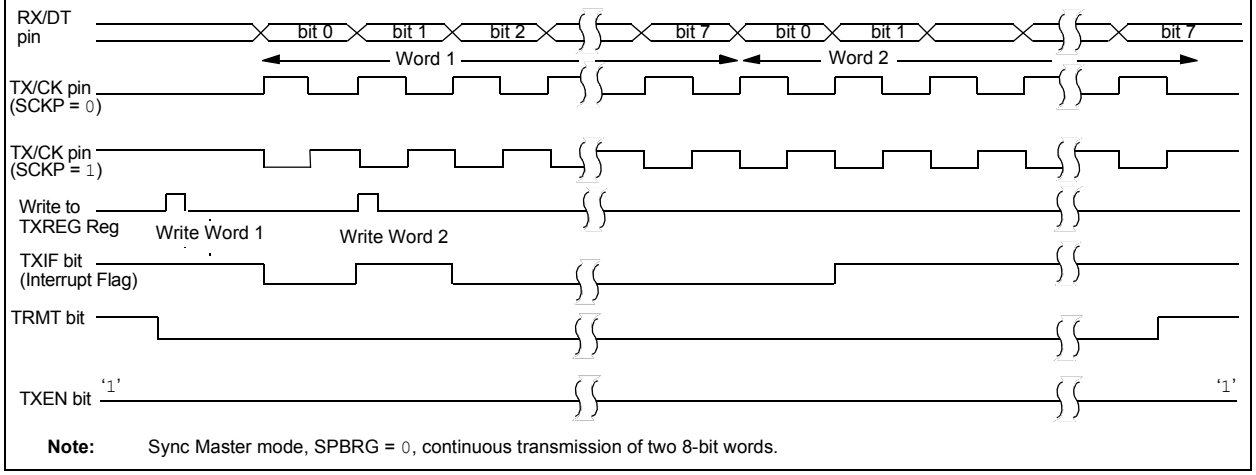
Each data bit changes on the leading edge of the master clock and remains valid until the subsequent leading clock edge.

| **Note:** | The TSR register is not mapped in data memory, so it is not available to the user. |
|---|---|

#### 12.4.1.4 Synchronous Master Transmission Setup:

1. Initialize the SPBRGH, SPBRG register pair and the BRGH and BRG16 bits to achieve the desired baud rate (see **Section 12.3 "EUSART Baud Rate Generator (BRG)"**).
2. Enable the synchronous master serial port by setting bits SYNC, SPEN, and CSRC.
3. Disable Receive mode by clearing bits SREN and CREN.
4. Enable Transmit mode by setting the TXEN bit.
5. If 9-bit transmission is desired, set the TX9 bit.
6. If interrupts are desired, set the TXIE bit of the PIE1 register and the GIE and PEIE bits of the INTCON register.
7. If 9-bit transmission is selected, the ninth bit should be loaded in the TX9D bit.
8. Start transmission by loading data to the TXREG register.

**FIGURE 12-10: SYNCHRONOUS TRANSMISSION**




**Note:** Sync Master mode, SPBRG = 0, continuous transmission of two 8-bit words.

**FIGURE 12-11: SYNCHRONOUS TRANSMISSION (THROUGH TXEN)**

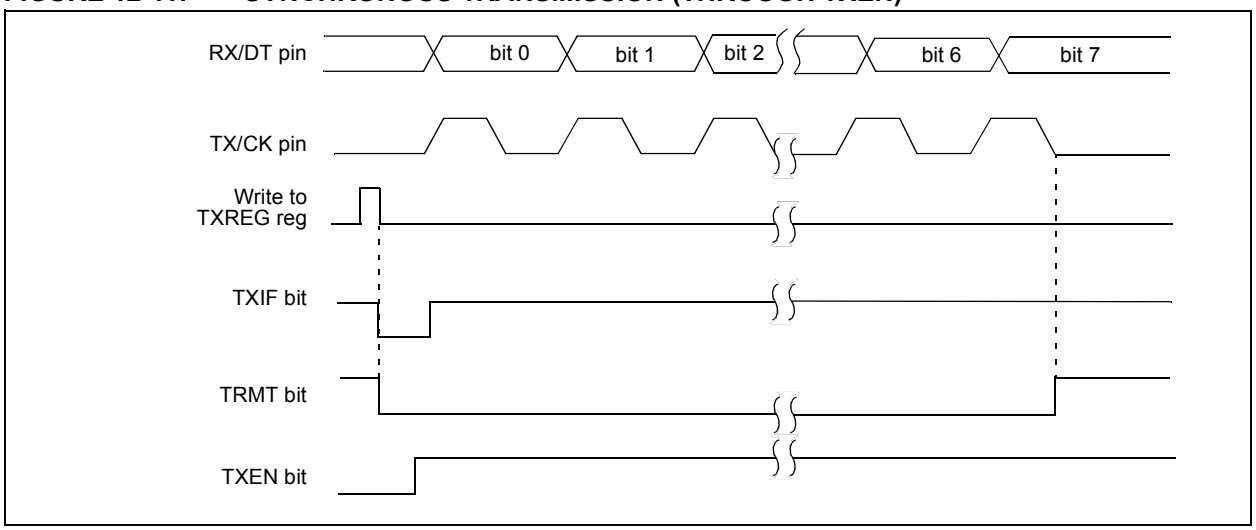



**TABLE 12-7: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER TRANSMISSION**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|
| BAUDCTL | ABDOVF | RCIDL | — | SCKP | BRG16 | — | WUE | ABDEN | 166 |
| INTCON | GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | RBIF | 33 |
| PIE1 | — | ADIE | RCIE | TXIE | SSPIE | CCP1IE | TMR2IE | TMR1IE | 34 |
| PIR1 | — | ADIF | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | 36 |
| RCREG | EUSART Receive Data Register | | | | | | | | 162 |
| RCSTA | SPEN | RX9 | SREN | CREN | ADDEN | FERR | OERR | RX9D | 165 |
| SPBRG | BRG7 | BRG6 | BRG5 | BRG4 | BRG3 | BRG2 | BRG1 | BRG0 | 167 |
| SPBRGH | BRG15 | BRG14 | BRG13 | BRG12 | BRG11 | BRG10 | BRG9 | BRG8 | 167 |
| TRISC | TRISC7 | TRISC6 | TRISC5 | TRISC4 | TRISC3 | TRISC2 | TRISC1 | TRISC0 | 55 |
| TXREG | EUSART Transmit Data Register | | | | | | | | 157 |
| TXSTA | CSRC | TX9 | TXEN | SYNC | SENDB | BRGH | TRMT | TX9D | 164 |

**Legend:** x = unknown, – = unimplemented read as '0'. Shaded cells are not used for Synchronous Master Transmission.


© 2006-2012 Microchip Technology Inc.

#### 12.4.1.5 Synchronous Master Reception

Data is received at the RX/DT pin. The RX/DT pin output driver is automatically disabled when the EUSART is configured for synchronous master receive operation.

In Synchronous mode, reception is enabled by setting either the Single Receive Enable bit (SREN of the RCSTA register) or the Continuous Receive Enable bit (CREN of the RCSTA register).

When SREN is set and CREN is clear, only as many clock cycles are generated as there are data bits in a single character. The SREN bit is automatically cleared at the completion of one character. When CREN is set, clocks are continuously generated until CREN is cleared. If CREN is cleared in the middle of a character the CK clock stops immediately and the partial character is discarded. If SREN and CREN are both set, then SREN is cleared at the completion of the first character and CREN takes precedence.

To initiate reception, set either SREN or CREN. Data is sampled at the RX/DT pin on the trailing edge of the TX/CK clock pin and is shifted into the Receive Shift Register (RSR). When a complete character is received into the RSR, the RCIF bit is set and the character is automatically transferred to the two character receive FIFO. The Least Significant eight bits of the top character in the receive FIFO are available in RCREG. The RCIF bit remains set as long as there are un-read characters in the receive FIFO.

#### 12.4.1.6 Slave Clock

Synchronous data transfers use a separate clock line, which is synchronous with the data. A device configured as a slave receives the clock on the TX/CK line. The TX/CK pin output driver is automatically disabled when the device is configured for synchronous slave transmit or receive operation. Serial data bits change on the leading edge to ensure they are valid at the trailing edge of each clock. One data bit is transferred for each clock cycle. Only as many clock cycles should be received as there are data bits.

#### 12.4.1.7 Receive Overrun Error

The receive FIFO buffer can hold two characters. An overrun error will be generated if a third character, in its entirety, is received before RCREG is read to access the FIFO. When this happens the OERR bit of the RCSTA register is set. Previous data in the FIFO will not be overwritten. The two characters in the FIFO buffer can be read, however, no additional characters will be received until the error is cleared. The OERR bit can only be cleared by clearing the overrun condition. If the overrun error occurred when the SREN bit is set and CREN is clear then the error is cleared by reading RCREG. If the overrun occurred when the CREN bit is set then the error condition is cleared by either clearing the CREN bit of the RCSTA register or by clearing the SPEN bit which resets the EUSART.

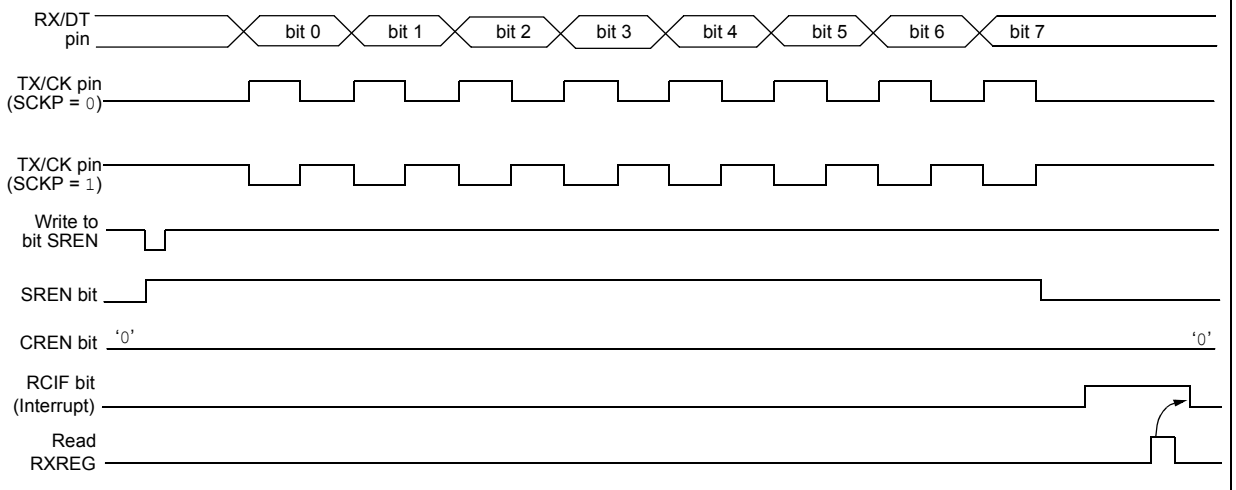
#### 12.4.1.8 Receiving 9-Bit Characters

The EUSART supports 9-bit character reception. When the RX9 bit of the RCSTA register is set the EUSART will shift 9-bits into the RSR for each character received. The RX9D bit of the RCSTA register is the ninth, and Most Significant, data bit of the top unread character in the receive FIFO. When reading 9-bit data from the receive FIFO buffer, the RX9D data bit must be read before reading the 8 Least Significant bits from the RCREG.

#### 12.4.1.9 Synchronous Master Reception Setup:

1. Initialize the SPBRGH, SPBRG register pair for the appropriate baud rate. Set or clear the BRGH and BRG16 bits, as required, to achieve the desired baud rate.
2. Enable the synchronous master serial port by setting bits SYNC, SPEN and CSRC.
3. Ensure bits CREN and SREN are clear.
4. If interrupts are desired, set the RCIE bit of the PIE1 register and the GIE and PEIE bits of the INTCON register.
5. If 9-bit reception is desired, set bit RX9.
6. Start reception by setting the SREN bit or for continuous reception, set the CREN bit.
7. Interrupt flag bit RCIF will be set when reception of a character is complete. An interrupt will be generated if the enable bit RCIE was set.
8. Read the RCSTA register to get the ninth bit (if enabled) and determine if any error occurred during reception.
9. Read the 8-bit received data by reading the RCREG register.
10. If an overrun error occurs, clear the error by either clearing the CREN bit of the RCSTA register or by clearing the SPEN bit which resets the EUSART.

**FIGURE 12-12: SYNCHRONOUS RECEPTION (MASTER MODE, SREN)**




**Note:** Timing diagram demonstrates Sync Master mode with bit SREN = 1 and bit BRGH = 0.

**TABLE 12-8: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER RECEPTION**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|
| BAUDCTL | ABDOVF | RCIDL | — | SCKP | BRG16 | — | WUE | ABDEN | 166 |
| INTCON | GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | RBIF | 33 |
| PIE1 | — | ADIE | RCIE | TXIE | SSPIE | CCP1IE | TMR2IE | TMR1IE | 34 |
| PIR1 | — | ADIF | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | 36 |
| RCREG | EUSART Receive Data Register | | | | | | | | 162 |
| RCSTA | SPEN | RX9 | SREN | CREN | ADDEN | FERR | OERR | RX9D | 165 |
| SPBRG | BRG7 | BRG6 | BRG5 | BRG4 | BRG3 | BRG2 | BRG1 | BRG0 | 167 |
| SPBRGH | BRG15 | BRG14 | BRG13 | BRG12 | BRG11 | BRG10 | BRG9 | BRG8 | 167 |
| TRISC | TRISC7 | TRISC6 | TRISC5 | TRISC4 | TRISC3 | TRISC2 | TRISC1 | TRISC0 | 55 |
| TXREG | EUSART Transmit Data Register | | | | | | | | 157 |
| TXSTA | CSRC | TX9 | TXEN | SYNC | SENDB | BRGH | TRMT | TX9D | 164 |

**Legend:** x = unknown, – = unimplemented read as '0'. Shaded cells are not used for Synchronous Master Reception.


© 2006-2012 Microchip Technology Inc.

### 12.4.2 SYNCHRONOUS SLAVE MODE

The following bits are used to configure the EUSART for Synchronous slave operation:

- SYNC = 1
- CSRC = 0
- SREN = 0 (for transmit); SREN = 1 (for receive)
- CREN = 0 (for transmit); CREN = 1 (for receive)
- SPEN = 1

Setting the SYNC bit of the TXSTA register configures the device for synchronous operation. Clearing the CSRC bit of the TXSTA register configures the device as a slave. Clearing the SREN and CREN bits of the RCSTA register ensures that the device is in the Transmit mode, otherwise the device will be configured to receive. Setting the SPEN bit of the RCSTA register enables the EUSART. If the RX/DT or TX/CK pins are shared with an analog peripheral the analog I/O functions must be disabled by clearing the corresponding ANSEL bits.

#### 12.4.2.1 EUSART Synchronous Slave Transmit

The operation of the Synchronous Master and Slave modes are identical (see **Section 12.4.1.3 "Synchronous Master Transmission"**), except in the case of the Sleep mode.

If two words are written to the TXREG and then the SLEEP instruction is executed, the following will occur:

1. The first character will immediately transfer to the TSR register and transmit.
2. The second word will remain in TXREG register.
3. The TXIF bit will not be set.
4. After the first character has been shifted out of TSR, the TXREG register will transfer the second character to the TSR and the TXIF bit will now be set.
5. If the PEIE and TXIE bits are set, the interrupt will wake the device from Sleep and execute the next instruction. If the GIE bit is also set, the program will call the Interrupt Service Routine.

#### 12.4.2.2 Synchronous Slave Transmission Setup:

1. Set the SYNC and SPEN bits and clear the CSRC bit.
2. Clear the CREN and SREN bits.
3. If interrupts are desired, set the TXIE bit of the PIE1 register and the GIE and PEIE bits of the INTCON register.
4. If 9-bit transmission is desired, set the TX9 bit.
5. Enable transmission by setting the TXEN bit.
6. If 9-bit transmission is selected, insert the Most Significant bit into the TX9D bit.
7. Start transmission by writing the Least Significant 8 bits to the TXREG register.

**TABLE 12-9: REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE TRANSMISSION**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|
| BAUDCTL | ABDOVF | RCIDL | — | SCKP | BRG16 | — | WUE | ABDEN | 166 |
| INTCON | GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | RBIF | 33 |
| PIE1 | — | ADIE | RCIE | TXIE | SSPIE | CCP1IE | TMR2IE | TMR1IE | 34 |
| PIR1 | — | ADIF | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | 36 |
| RCREG | EUSART Receive Data Register | | | | | | | | 162 |
| RCSTA | SPEN | RX9 | SREN | CREN | ADDEN | FERR | OERR | RX9D | 165 |
| SPBRG | BRG7 | BRG6 | BRG5 | BRG4 | BRG3 | BRG2 | BRG1 | BRG0 | 167 |
| SPBRGH | BRG15 | BRG14 | BRG13 | BRG12 | BRG11 | BRG10 | BRG9 | BRG8 | 167 |
| TRISC | TRISC7 | TRISC6 | TRISC5 | TRISC4 | TRISC3 | TRISC2 | TRISC1 | TRISC0 | 55 |
| TXREG | EUSART Transmit Data Register | | | | | | | | 157 |
| TXSTA | CSRC | TX9 | TXEN | SYNC | SENDB | BRGH | TRMT | TX9D | 164 |

**Legend:** x = unknown, – = unimplemented read as '0'. Shaded cells are not used for Synchronous Slave Transmission.

#### 12.4.2.3 EUSART Synchronous Slave Reception

The operation of the Synchronous Master and Slave modes is identical (**Section 12.4.1.5 "Synchronous Master Reception"**), with the following exceptions:

- Sleep
- CREN bit is always set, therefore the receiver is never Idle
- SREN bit, which is a "don't care" in Slave mode

A character may be received while in Sleep mode by setting the CREN bit prior to entering Sleep. Once the word is received, the RSR register will transfer the data to the RCREG register. If the RCIE enable bit is set, the interrupt generated will wake the device from Sleep and execute the next instruction. If the GIE bit is also set, the program will branch to the interrupt vector.

#### 12.4.2.4 Synchronous Slave Reception Setup:

1. Set the SYNC and SPEN bits and clear the CSRC bit.
2. If interrupts are desired, set the RCIE bit of the PIE1 register and the GIE and PEIE bits of the INTCON register.
3. If 9-bit reception is desired, set the RX9 bit.
4. Set the CREN bit to enable reception.
5. The RCIF bit will be set when reception is complete. An interrupt will be generated if the RCIE bit was set.
6. If 9-bit mode is enabled, retrieve the Most Significant bit from the RX9D bit of the RCSTA register.
7. Retrieve the 8 Least Significant bits from the receive FIFO by reading the RCREG register.
8. If an overrun error occurs, clear the error by either clearing the CREN bit of the RCSTA register or by clearing the SPEN bit which resets the EUSART.

**TABLE 12-10: REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE RECEPTION**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|
| BAUDCTL | ABDOVF | RCIDL | — | SCKP | BRG16 | — | WUE | ABDEN | 166 |
| INTCON | GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | RBIF | 33 |
| PIE1 | — | ADIE | RCIE | TXIE | SSPIE | CCP1IE | TMR2IE | TMR1IE | 34 |
| PIR1 | — | ADIF | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | 36 |
| RCREG | EUSART Receive Data Register | | | | | | | | 162 |
| RCSTA | SPEN | RX9 | SREN | CREN | ADDEN | FERR | OERR | RX9D | 165 |
| SPBRG | BRG7 | BRG6 | BRG5 | BRG4 | BRG3 | BRG2 | BRG1 | BRG0 | 167 |
| SPBRGH | BRG15 | BRG14 | BRG13 | BRG12 | BRG11 | BRG10 | BRG9 | BRG8 | 167 |
| TRISC | TRISC7 | TRISC6 | TRISC5 | TRISC4 | TRISC3 | TRISC2 | TRISC1 | TRISC0 | 55 |
| TXREG | EUSART Transmit Data Register | | | | | | | | 157 |
| TXSTA | CSRC | TX9 | TXEN | SYNC | SENDB | BRGH | TRMT | TX9D | 164 |

**Legend:** x = unknown, – = unimplemented read as '0'. Shaded cells are not used for Synchronous Slave Reception.


© 2006-2012 Microchip Technology Inc.

## 12.5 EUSART Operation During Sleep

The EUSART WILL remain active during Sleep only in the Synchronous Slave mode. All other modes require the system clock and therefore cannot generate the necessary signals to run the Transmit or Receive Shift registers during Sleep.

Synchronous Slave mode uses an externally generated clock to run the Transmit and Receive Shift registers.

### 12.5.1 SYNCHRONOUS RECEIVE DURING SLEEP

To receive during Sleep, all the following conditions must be met before entering Sleep mode:

- RCSTA and TXSTA Control registers must be configured for Synchronous Slave Reception (see **Section 12.4.2.4 "Synchronous Slave Reception Setup:"**).
- If interrupts are desired, set the RCIE bit of the PIE1 register and the GIE and PEIE bits of the INTCON register.
- The RCIF interrupt flag must be cleared by reading RCREG to unload any pending characters in the receive buffer.

Upon entering Sleep mode, the device will be ready to accept data and clocks on the RX/DT and TX/CK pins, respectively. When the data word has been completely clocked in by the external device, the RCIF interrupt flag bit of the PIR1 register will be set. Thereby, waking the processor from Sleep.

Upon waking from Sleep, the instruction following the SLEEP instruction will be executed. If the GIE global interrupt enable bit of the INTCON register is also set, then the Interrupt Service Routine at address 004h will be called.

### 12.5.2 SYNCHRONOUS TRANSMIT DURING SLEEP

To transmit during Sleep, all the following conditions must be met before entering Sleep mode:

- RCSTA and TXSTA Control registers must be configured for Synchronous Slave Transmission (see **Section 12.4.2.2 "Synchronous Slave Transmission Setup:"**).
- The TXIF interrupt flag must be cleared by writing the output data to the TXREG, thereby filling the TSR and transmit buffer.

9. If interrupts are desired, set the TXIE bit of the PIE1 register and the PEIE bit of the INTCON register.

- Interrupt enable bits TXIE of the PIE1 register and PEIE of the INTCON register must set.

Upon entering Sleep mode, the device will be ready to accept clocks on TX/CK pin and transmit data on the RX/DT pin. When the data word in the TSR has been completely clocked out by the external device, the pending byte in the TXREG will transfer to the TSR and the TXIF flag will be set. Thereby, waking the processor from Sleep. At this point, the TXREG is available to accept another character for transmission, which will clear the TXIF flag.

Upon waking from Sleep, the instruction following the SLEEP instruction will be executed. If the GIE global interrupt enable bit is also set then the Interrupt Service Routine at address 0004h will be called.

**NOTES:**


© 2006-2012 Microchip Technology Inc.

# 13.0 MASTER SYNCHRONOUS SERIAL PORT (MSSP) MODULE

## 13.1 Master SSP (MSSP) Module Overview

The Master Synchronous Serial Port (MSSP) module is a serial interface useful for communicating with other peripheral or microcontroller devices. These peripheral devices may be Serial EEPROMs, shift registers, display drivers, A/D converters, etc. The MSSP module can operate in one of two modes:

- Serial Peripheral Interface (SPI)
- Inter-Integrated Circuit™ ($I^2C$™)
  - Full Master mode
  - Slave mode (with general address call).

The $I^2C$ interface supports the following modes in hardware:

- Master mode
- Multi-Master mode
- Slave mode.

## 13.2 Control Registers

The MSSP module has three associated registers. These include a STATUS register and two control registers.

Register 13-1 shows the MSSP STATUS register (SSPSTAT), Register 13-2 shows the MSSP Control Register 1 (SSPCON), and Register 13-3 shows the MSSP Control Register 2 (SSPCON2).

**REGISTER 13-1: SSPSTAT: SSP STATUS REGISTER**

| R/W-0 | R/W-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |
|---|---|---|---|---|---|---|---|
| SMP | CKE | D/$\overline{A}$ | P | S | R/$\overline{W}$ | UA | BF |
| bit 7 | | | | | | | bit 0 |

| **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 **SMP:** Sample bit

SPI Master mode:
1 = Input data sampled at end of data output time
0 = Input data sampled at middle of data output time

SPI Slave mode:
SMP must be cleared when SPI is used in Slave mode

In $I^2C$ Master or Slave mode:
1 = Slew rate control disabled for standard speed mode (100 kHz and 1 MHz)
0 = Slew rate control enabled for high speed mode (400 kHz)

bit 6 **CKE:** SPI Clock Edge Select bit

CKP = 0:
1 = Data transmitted on falling edge of SCK
0 = Data transmitted on rising edge of SCK

CKP = 1:
1 = Data transmitted on rising edge of SCK
0 = Data transmitted on falling edge of SCK

bit 5 **D/$\overline{A}$:** Data/$\overline{Address}$ bit ($I^2C$ mode only)
1 = Indicates that the last byte received or transmitted was data
0 = Indicates that the last byte received or transmitted was address

bit 4 **P:** Stop bit

($I^2C$ mode only. This bit is cleared when the MSSP module is disabled, SSPEN is cleared.)
1 = Indicates that a Stop bit has been detected last (this bit is '0' on Reset)
0 = Stop bit was not detected last

bit 3 **S:** Start bit

($I^2C$ mode only. This bit is cleared when the MSSP module is disabled, SSPEN is cleared.)
1 = Indicates that a Start bit has been detected last (this bit is '0' on Reset)
0 = Start bit was not detected last

bit 2 **R/$\overline{W}$:** Read/Write bit information ($I^2C$ mode only)

This bit holds the R/$\overline{W}$ bit information following the last address match. This bit is only valid from the address match to the next Start bit, Stop bit, or not ACK bit.

In $I^2C$ Slave mode:
1 = Read
0 = Write

In $I^2C$ Master mode:
1 = Transmit is in progress
0 = Transmit is not in progress
OR-ing this bit with SEN, RSEN, PEN, RCEN, or ACKEN will indicate if the MSSP is in Idle mode.

bit 1 **UA:** Update Address bit (10-bit $I^2C$ mode only)
1 = Indicates that the user needs to update the address in the SSPADD register
0 = Address does not need to be updated

bit 0 **BF:** Buffer Full Status bit

Receive (SPI and $I^2C$ modes):
1 = Receive complete, SSPBUF is full
0 = Receive not complete, SSPBUF is empty

Transmit ($I^2C$ mode only):
1 = Data transmit in progress (does not include the $\overline{ACK}$ and Stop bits), SSPBUF is full
0 = Data transmit complete (does not include the $\overline{ACK}$ and Stop bits), SSPBUF is empty


© 2006-2012 Microchip Technology Inc.

**REGISTER 13-2: SSPCON: SSP CONTROL REGISTER 1**

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| WCOL | SSPOV | SSPEN | CKP | SSPM3 | SSPM2 | SSPM1 | SSPM0 |
| 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 **WCOL:** Write Collision Detect bit
Master mode:
1 = A write to the SSPBUF register was attempted while the $I^2C$ conditions were not valid for a transmission to be started
0 = No collision
Slave mode:
1 = The SSPBUF register is written while it is still transmitting the previous word (must be cleared in software)
0 = No collision

bit 6 **SSPOV:** Receive Overflow Indicator bit
In SPI mode:
1 = A new byte is received while the SSPBUF register is still holding the previous data. In case of overflow, the data in SSPSR is lost. Overflow can only occur in Slave mode. In Slave mode, the user must read the SSPBUF, even if only transmitting data, to avoid setting overflow. In Master mode, the overflow bit is not set since each new reception (and transmission) is initiated by writing to the SSPBUF register (must be cleared in software).
0 = No overflow
In $I^2C$ mode:
1 = A byte is received while the SSPBUF register is still holding the previous byte. SSPOV is a "don't care" in Transmit mode (must be cleared in software).
0 = No overflow

bit 5 **SSPEN:** Synchronous Serial Port Enable bit
In both modes, when enabled, these pins must be properly configured as input or output
In SPI mode:
1 = Enables serial port and configures SCK, SDO, SDI and $\overline{SS}$ as the source of the serial port pins
0 = Disables serial port and configures these pins as I/O port pins
In $I^2C$ mode:
1 = Enables the serial port and configures the SDA and SCL pins as the source of the serial port pins
0 = Disables serial port and configures these pins as I/O port pins

bit 4 **CKP:** Clock Polarity Select bit
In SPI mode:
1 = Idle state for clock is a high level
0 = Idle state for clock is a low level
In $I^2C$ Slave mode:
SCK release control
1 = Release clock
0 = Holds clock low (clock stretch). (Used to ensure data setup time.)
In $I^2C$ Master mode:
Unused in this mode

bit 3-0 **SSPM<3:0>:** Synchronous Serial Port Mode Select bits
0000 = SPI Master mode, clock = FOSC/4
0001 = SPI Master mode, clock = FOSC/16
0010 = SPI Master mode, clock = FOSC/64
0011 = SPI Master mode, clock = TMR2 output/2
0100 = SPI Slave mode, clock = SCK pin, $\overline{SS}$ pin control enabled
0101 = SPI Slave mode, clock = SCK pin, $\overline{SS}$ pin control disabled, $\overline{SS}$ can be used as I/O pin
0110 = $I^2C$ Slave mode, 7-bit address
0111 = $I^2C$ Slave mode, 10-bit address
1000 = $I^2C$ Master mode, clock = FOSC / (4 * (SSPADD+1))
1001 = Load Mask function
1010 = Reserved
1011 = $I^2C$ firmware controlled Master mode (Slave idle)
1100 = Reserved
1101 = Reserved
1110 = $I^2C$ Slave mode, 7-bit address with Start and Stop bit interrupts enabled
1111 = $I^2C$ Slave mode, 10-bit address with Start and Stop bit interrupts enabled

© 2006-2012 Microchip Technology Inc.

**REGISTER 13-3: SSPCON2: SSP CONTROL REGISTER 2**

| R/W-0 | R-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| GCEN | ACKSTAT | ACKDT | ACKEN | RCEN | PEN | RSEN | SEN |
| bit 7 | | | | | | | bit 0 |

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

bit 7 **GCEN:** General Call Enable bit (in $I^2C$ Slave mode only)
1 = Enable interrupt when a general call address (0000h) is received in the SSPSR
0 = General call address disabled

bit 6 **ACKSTAT:** Acknowledge Status bit (in $I^2C$ Master mode only)
In Master Transmit mode:
1 = Acknowledge was not received from slave
0 = Acknowledge was received from slave

bit 5 **ACKDT:** Acknowledge Data bit (in $I^2C$ Master mode only)
In Master Receive mode:
Value transmitted when the user initiates an Acknowledge sequence at the end of a receive
1 = Not Acknowledge
0 = Acknowledge

bit 4 **ACKEN:** Acknowledge Sequence Enable bit (in $I^2C$ Master mode only)
In Master Receive mode:
1 = Initiate Acknowledge sequence on SDA and SCL pins, and transmit ACKDT data bit. Automatically cleared by hardware.
0 = Acknowledge sequence idle

bit 3 **RCEN:** Receive Enable bit (in $I^2C$ Master mode only)
1 = Enables Receive mode for $I^2C$
0 = Receive idle

bit 2 **PEN:** Stop Condition Enable bit (in $I^2C$ Master mode only)
SCK Release Control:
1 = Initiate Stop condition on SDA and SCL pins. Automatically cleared by hardware.
0 = Stop condition Idle

bit 1 **RSEN:** Repeated Start Condition Enabled bit (in $I^2C$ Master mode only)
1 = Initiate Repeated Start condition on SDA and SCL pins. Automatically cleared by hardware.
0 = Repeated Start condition Idle

bit 0 **SEN:** Start Condition Enabled bit (in $I^2C$ Master mode only)
In Master mode:
1 = Initiate Start condition on SDA and SCL pins. Automatically cleared by hardware.
0 = Start condition Idle
In Slave mode:
1 = Clock stretching is enabled for both slave transmit and slave receive (stretch enabled)
0 = Clock stretching is disabled

**Note 1:** For bits ACKEN, RCEN, PEN, RSEN, SEN: If the $I^2C$ module is not in the Idle mode, this bit may not be set (no spooling) and the SSPBUF may not be written (or writes to the SSPBUF are disabled).


© 2006-2012 Microchip Technology Inc.

## 13.3 SPI Mode

The SPI mode allows 8 bits of data to be synchronously transmitted and received, simultaneously. All four modes of SPI are supported. To accomplish communication, typically three pins are used:

- Serial Data Out (SDO) – RC5/SDO
- Serial Data In (SDI) – RC4/SDI/SDA
- Serial Clock (SCK) – RC3/SCK/SCL

Additionally, a fourth pin may be used when in any Slave mode of operation:

- Slave Select ($\overline{SS}$) – RA5/$\overline{SS}$/AN4

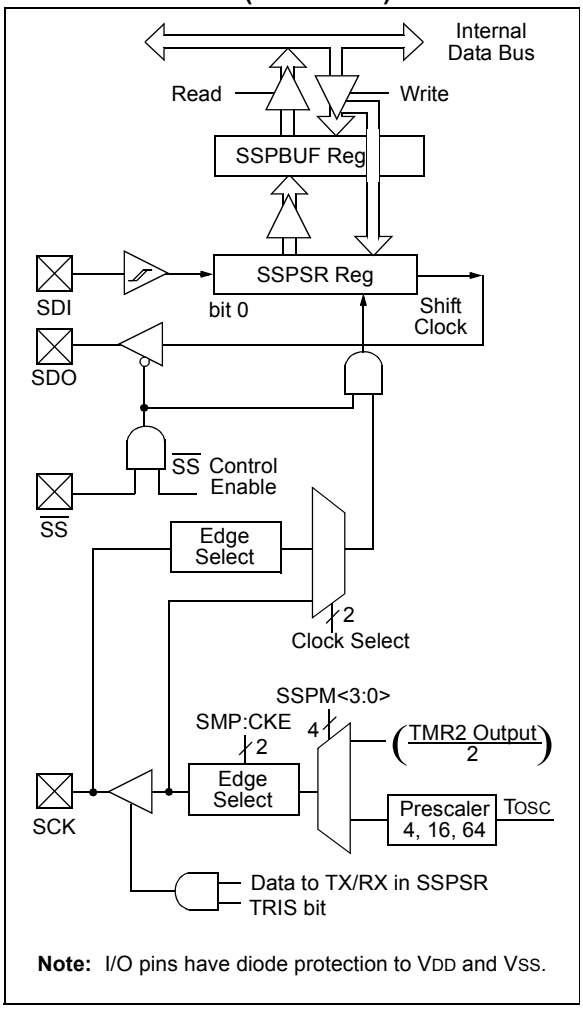
### 13.3.1 OPERATION

When initializing the SPI, several options need to be specified. This is done by programming the appropriate control bits SSPCON<5:0> and SSPSTAT<7:6>. These control bits allow the following to be specified:

- Master mode (SCK is the clock output)
- Slave mode (SCK is the clock input)
- Clock polarity (Idle state of SCK)
- Data input sample phase (middle or end of data output time)
- Clock edge (output data on rising/falling edge of SCK)
- Clock rate (Master mode only)
- Slave Select mode (Slave mode only)

Figure 13-1 shows the block diagram of the MSSP module, when in SPI mode.

**FIGURE 13-1: MSSP BLOCK DIAGRAM (SPI MODE)**




**Note:** I/O pins have diode protection to VDD and VSS.

The MSSP consists of a transmit/receive shift register (SSPSR) and a buffer register (SSPBUF). The SSPSR shifts the data in and out of the device, MSb first. The SSPBUF holds the data that was written to the SSPSR, until the received data is ready. Once the 8 bits of data have been received, that byte is moved to the SSPBUF register. Then, the buffer full-detect bit BF of the SSP-STAT register and the interrupt flag bit SSPIF of the PIR1 register are set. This double buffering of the received data (SSPBUF) allows the next byte to start reception before reading the data that was just received. Any write to the SSPBUF register during transmission/reception of data will be ignored, and the write collision detect bit WCOL of the SSPCON register will be set. User software must clear the WCOL bit so that it can be determined if the following write(s) to the SSPBUF register completed successfully.

When the application software is expecting to receive valid data, the SSPBUF should be read before the next byte of data to transfer is written to the SSPBUF. The buffer full bit BF of the SSPSTAT register indicates when SSPBUF has been loaded with the received data (transmission is complete). When the SSPBUF is read, the BF bit is cleared. This data may be irrelevant if the SPI is only a transmitter. Generally, the MSSP Interrupt is used to determine when the transmission/reception has completed. The SSPBUF must be read and/or written. If the interrupt method is not going to be used, then software polling can be done to ensure that a write collision does not occur. Example 13-1 shows the loading of the SSPBUF (SSPSR) for data transmission.

The SSPSR is not directly readable or writable, and can only be accessed by addressing the SSPBUF register. Additionally, the MSSP STATUS register (SSPSTAT register) indicates the various status conditions.

### 13.3.2 ENABLING SPI I/O

To enable the serial port, SSP Enable bit SSPEN of the SSPCON register must be set. To reset or reconfigure SPI mode, clear the SSPEN bit, re-initialize the SSPCON registers, and then set the SSPEN bit. This configures the SDI, SDO, SCK and $\overline{SS}$ pins as serial port pins. For the pins to behave as the serial port function, some must have their data direction bits (in the TRIS register) appropriately programmed. That is:

- SDI is automatically controlled by the SPI module
- SDO must have TRISC<5> bit cleared
- SCK (Master mode) must have TRISC<3> bit cleared
- SCK (Slave mode) must have TRISC<3> bit set
- $\overline{SS}$ must have TRISA<5> bit set

Any serial port function that is not desired may be overridden by programming the corresponding data direction (TRIS) register to the opposite value.

**EXAMPLE 13-1: LOADING THE SSPBUF (SSPSR) REGISTER**

```
LOOP BTFSS SSPSTAT, BF    ;Has data been received (transmit complete)?
     GOTO  LOOP           ;No
     MOVF  SSPBUF, W      ;WREG reg = contents of SSPBUF
     MOVWF RXDATA         ;Save in user RAM, if data is meaningful
     MOVF  TXDATA, W      ;W reg = contents of TXDATA
     MOVWF SSPBUF         ;New data to xmit
```


© 2006-2012 Microchip Technology Inc.

### 13.3.3 MASTER MODE

The master can initiate the data transfer at any time because it controls the SCK. The master determines when the slave is to broadcast data by the software protocol.

In Master mode, the data is transmitted/received as soon as the SSPBUF register is written to. If the SPI is only going to receive, the SDO output could be disabled (programmed as an input). The SSPSR register will continue to shift in the signal present on the SDI pin at the programmed clock rate. As each byte is received, it will be loaded into the SSPBUF register as a normal received byte (interrupts and Status bits appropriately set). This could be useful in receiver applications as a “Line Activity Monitor” mode.

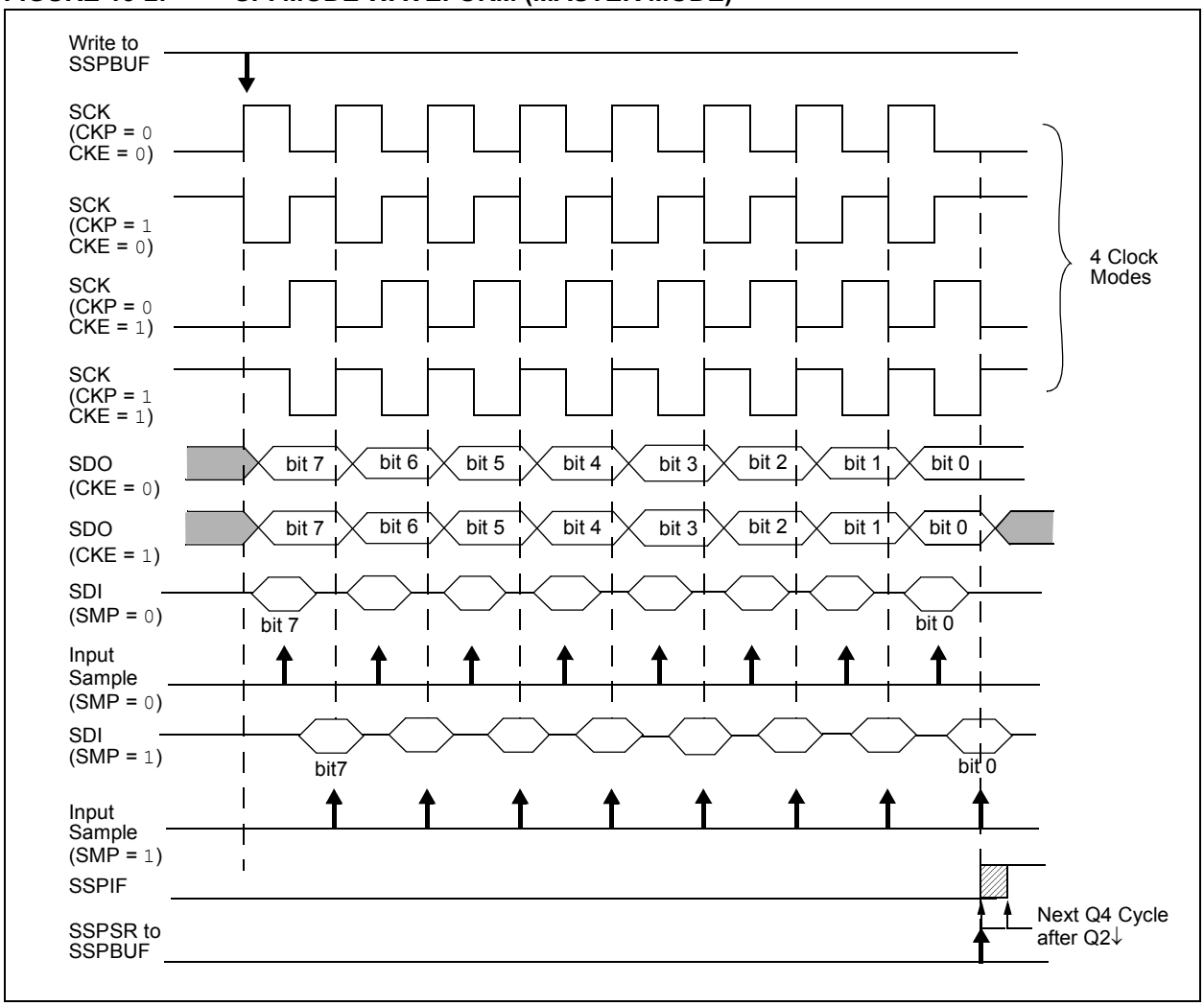
The clock polarity is selected by appropriately programming the CKP bit of the SSPCON register. This, then, would give waveforms for SPI communication as shown in Figure 13-2, Figure 13-4 and Figure 13-5, where the MSb is transmitted first. In Master mode, the SPI clock rate (bit rate) is user programmable to be one of the following:

- FOSC/4 (or TCY)
- FOSC/16 (or 4 • TCY)
- FOSC/64 (or 16 • TCY)
- Timer2 output/2

This allows a maximum data rate (at 40 MHz) of 10.00 Mbps.

Figure 13-2 shows the waveforms for Master mode. When the CKE bit of the SSPSTAT register is set, the SDO data is valid before there is a clock edge on SCK. The change of the input sample is shown based on the state of the SMP bit of the SSPSTAT register. The time when the SSPBUF is loaded with the received data is shown.

**FIGURE 13-2: SPI MODE WAVEFORM (MASTER MODE)**



### 13.3.4 SLAVE MODE

In Slave mode, the data is transmitted and received as the external clock pulses appear on SCK. When the last bit is latched, the SSPIF interrupt flag bit of the PIR1 register is set.

While in Slave mode, the external clock is supplied by the external clock source on the SCK pin. This external clock must meet the minimum high and low times, as specified in the electrical specifications.

While in Sleep mode, the slave can transmit/receive data. When a byte is received, the device will wake-up from Sleep.

### 13.3.5 SLAVE SELECT SYNCHRONIZATION

The $\overline{SS}$ pin allows a Synchronous Slave mode. The SPI must be in Slave mode with $\overline{SS}$ pin control enabled (SSPCON<3:0> = 04h). The pin must not be driven low for the $\overline{SS}$ pin to function as an input. The Data Latch must be high. When the $\overline{SS}$ pin is low, transmission and reception are enabled and the SDO pin is driven. When the $\overline{SS}$ pin goes high, the SDO pin is no longer driven, even if in the middle of a transmitted byte, and becomes a floating output. External pull-up/pull-down resistors may be desirable, depending on the application.

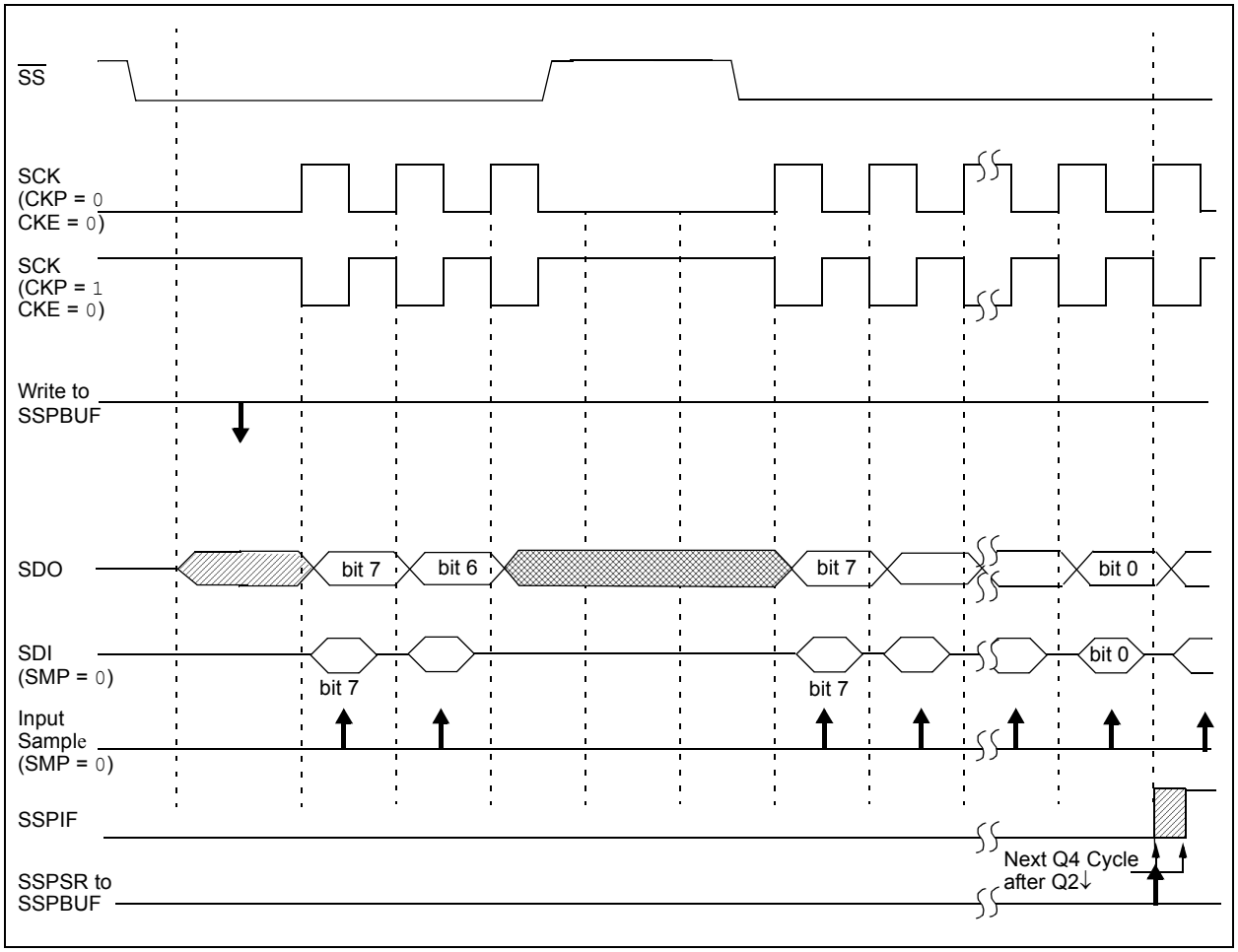
**Note 1:** When the SPI is in Slave mode with $\overline{SS}$ pin control enabled (SSPCON<3:0> = 0100), the SPI module will reset if the $\overline{SS}$ pin is set to $V_{DD}$.

**2:** If the SPI is used in Slave mode with CKE set (SSPSTAT register), then the $\overline{SS}$ pin control must be enabled.

When the SPI module resets, the bit counter is forced to '0'. This can be done by either forcing the $\overline{SS}$ pin to a high level, or clearing the SSPEN bit.

To emulate two-wire communication, the SDO pin can be connected to the SDI pin. When the SPI needs to operate as a receiver, the SDO pin can be configured as an input. This disables transmissions from the SDO. The SDI can always be left as an input (SDI function), since it cannot create a bus conflict.

**FIGURE 13-3: SLAVE SYNCHRONIZATION WAVEFORM**





© 2006-2012 Microchip Technology Inc.

**FIGURE 13-4: SPI MODE WAVEFORM (SLAVE MODE WITH CKE = 0)**

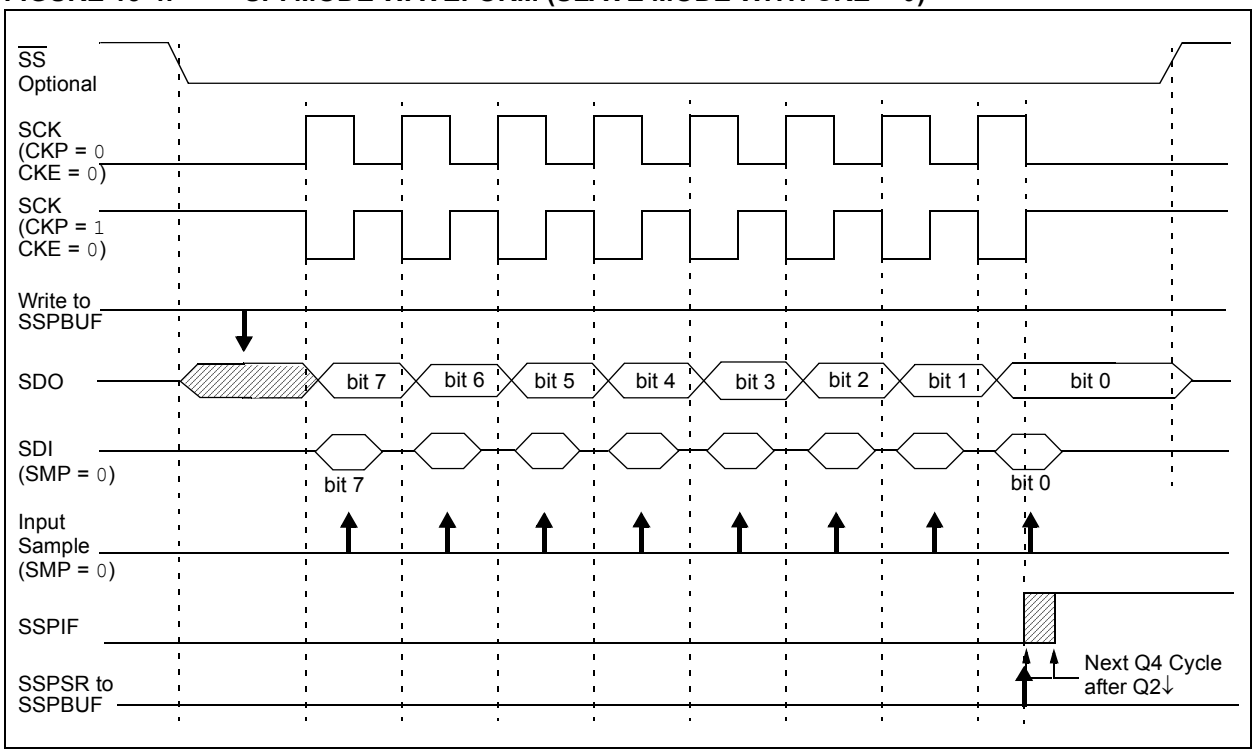
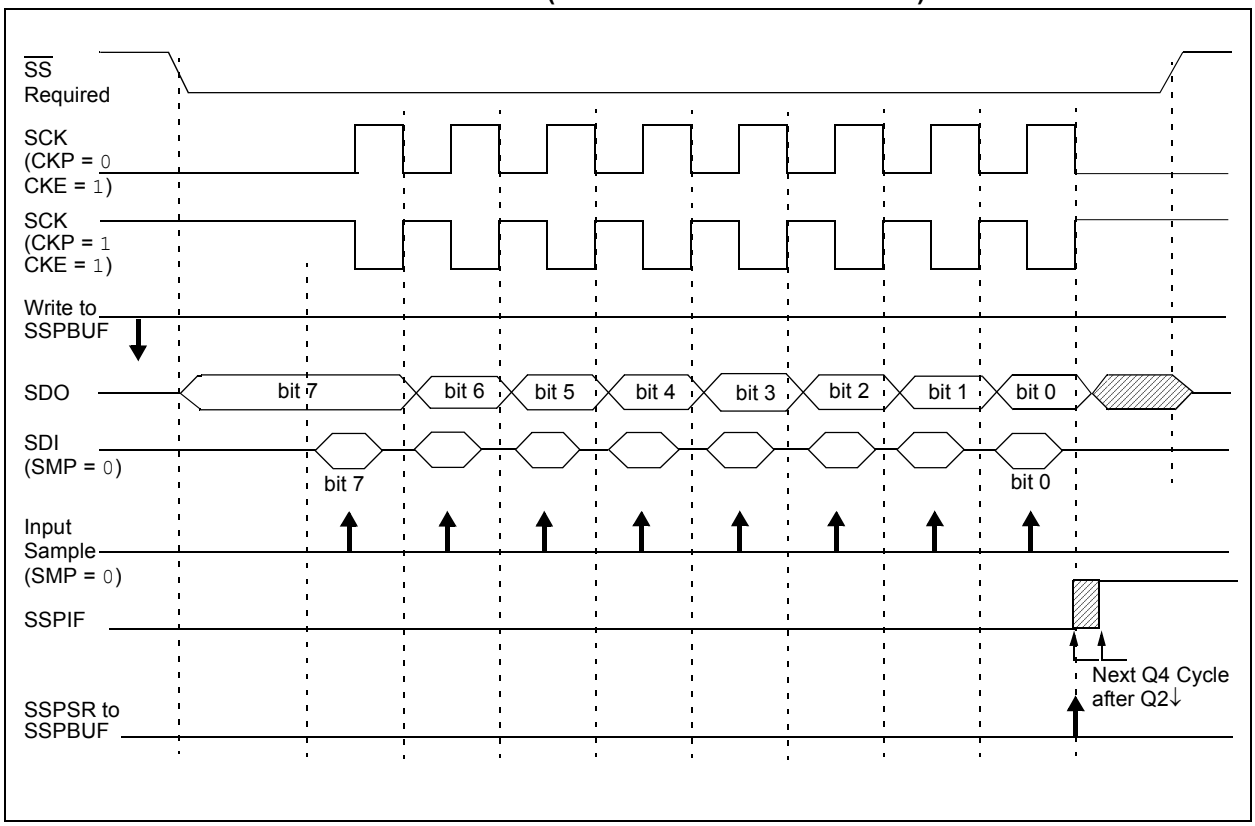



**FIGURE 13-5: SPI MODE WAVEFORM (SLAVE MODE WITH CKE = 1)**



### 13.3.6 SLEEP OPERATION

In Master mode, all module clocks are halted, and the transmission/reception will remain in that state until the device wakes from Sleep. After the device returns to normal mode, the module will continue to transmit/receive data.

In Slave mode, the SPI transmit/receive shift register operates asynchronously to the device. This allows the device to be placed in Sleep mode and data to be shifted into the SPI transmit/receive shift register. When all eight bits have been received, the MSSP interrupt flag bit will be set and, if enabled, will wake the device from Sleep.

### 13.3.7 EFFECTS OF A RESET

A Reset disables the MSSP module and terminates the current transfer.

### 13.3.8 BUS MODE COMPATIBILITY

Table 13-1 shows the compatibility between the standard SPI modes and the states of the CKP and CKE control bits.

**TABLE 13-1: SPI BUS MODES**

| Standard SPI Mode Terminology | Control Bits State | |
|---|---|---|
| | CKP | CKE |
| 0, 0 | 0 | 1 |
| 0, 1 | 0 | 0 |
| 1, 0 | 1 | 1 |
| 1, 1 | 1 | 0 |

There is also a SMP bit that controls when the data will be sampled.

**TABLE 13-2: REGISTERS ASSOCIATED WITH SPI OPERATION**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|
| INTCON | GIE/GIEH | PEIE/GIEL | T0IE | INTE | RBIE | T0IF | INTF | RBIF | 33 |
| PIE1 | — | ADIE | RCIE | TXIE | SSPIE | CCP1IE | TMR2IE | TMR1IE | 34 |
| PIR1 | — | ADIF | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | 36 |
| SSPBUF | Synchronous Serial Port Receive Buffer/Transmit Register | | | | | | | | 187 |
| SSPCON | WCOL | SSPOV | SSPEN | CKP | SSPM3 | SSPM2 | SSPM1 | SSPM0 | 185 |
| SSPSTAT | SMP | CKE | D/$\overline{A}$ | P | S | R/$\overline{W}$ | UA | BF | 184 |
| TRISA | TRISA7 | TRISA6 | TRISA5 | TRISA4 | TRISA3 | TRISA2 | TRISA1 | TRISA0 | 41 |
| TRISC | TRISC7 | TRISC6 | TRISC5 | TRISC4 | TRISC3 | TRISC2 | TRISC1 | TRISC0 | 55 |

**Legend:** x = unknown, u = unchanged, – = unimplemented, read as '0'. Shaded cells are not used by the MSSP in SPI mode.

**Note 1:** Bit 6 of PORTA, LATA and TRISA are enabled in ECIO and RCIO Oscillator modes only. In all other oscillator modes, they are disabled and read '0'.


© 2006-2012 Microchip Technology Inc.

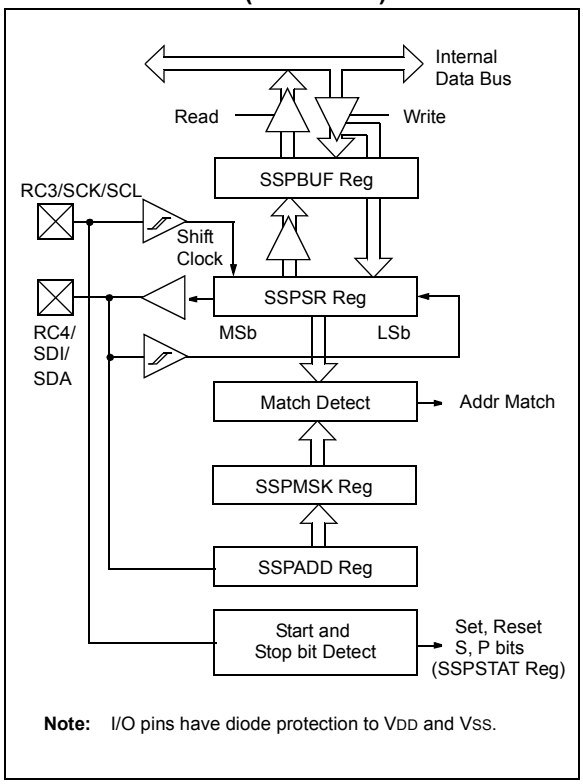
## 13.4 MSSP I²C Operation

The MSSP module in I²C mode, fully implements all master and slave functions (including general call support) and provides interrupts on Start and Stop bits in hardware, to determine a free bus (Multi-Master mode). The MSSP module implements the standard mode specifications, as well as 7-bit and 10-bit addressing.

Two pins are used for data transfer. These are the RC3/SCK/SCL pin, which is the clock (SCL), and the RC4/SDI/SDA pin, which is the data (SDA). The user must configure these pins as inputs or outputs through the TRISC<4:3> bits.

The MSSP module functions are enabled by setting MSSP Enable bit SSPEN of the SSPCON register.

**FIGURE 13-6: MSSP BLOCK DIAGRAM (I²C MODE)**




**Note:** I/O pins have diode protection to VDD and VSS.

The MSSP module has these six registers for I²C operation:

- MSSP Control Register 1 (SSPCON)
- MSSP Control Register 2 (SSPCON2)
- MSSP STATUS register (SSPSTAT)
- Serial Receive/Transmit Buffer (SSPBUF)
- MSSP Shift Register (SSPSR) – Not directly accessible
- MSSP Address register (SSPADD)
- MSSP Mask register (SSPMSK)

The SSPCON register allows control of the I²C operation. The SSPM<3:0> mode selection bits (SSPCON register) allow one of the following I²C modes to be selected:

- I²C Master mode, clock = OSC/4 (SSPADD +1)
- I²C Slave mode (7-bit address)
- I²C Slave mode (10-bit address)
- I²C Slave mode (7-bit address), with Start and Stop bit interrupts enabled
- I²C Slave mode (10-bit address), with Start and Stop bit interrupts enabled
- I²C firmware controlled master operation, slave is idle

Selection of any I²C mode with the SSPEN bit set, forces the SCL and SDA pins to be open drain, provided these pins are programmed to be inputs by setting the appropriate TRISC bits.

### 13.4.1 SLAVE MODE

In Slave mode, the SCL and SDA pins must be configured as inputs (TRISC<4:3> set). The MSSP module will override the input state with the output data when required (slave-transmitter).

When an address is matched, or the data transfer after an address match is received, the hardware automatically will generate the Acknowledge ($\overline{ACK}$) pulse and load the SSPBUF register with the received value currently in the SSPSR register.

If either or both of the following conditions are true, the MSSP module will not give this $\overline{ACK}$ pulse:

a) The buffer full bit BF (SSPCON register) was set before the transfer was received.

b) The overflow bit SSPOV (SSPCON register) was set before the transfer was received.

In this event, the SSPSR register value is not loaded into the SSPBUF, but bit SSPIF of the PIR1 register is set. The BF bit is cleared by reading the SSPBUF register, while bit SSPOV is cleared through software.

The SCL clock input must have a minimum high and low for proper operation. The high and low times of the I²C specification, as well as the requirement of the MSSP module, are shown in timing parameter #100 and parameter #101.

#### 13.4.1.1 Addressing

Once the MSSP module has been enabled, it waits for a Start condition to occur. Following the Start condition, the eight bits are shifted into the SSPSR register. All incoming bits are sampled with the rising edge of the clock (SCL) line. The value of register SSPSR<7:1> is compared to the value of the SSPADD register. The address is compared on the falling edge of the eighth clock (SCL) pulse. If the addresses match, and the BF and SSPOV bits are clear, the following events occur:

a) The SSPSR register value is loaded into the SSPBUF register.
b) The buffer full bit BF is set.
c) An $\overline{ACK}$ pulse is generated.
d) MSSP interrupt flag bit, SSPIF of the PIR1 register, is set on the falling edge of the ninth SCL pulse (interrupt is generated, if enabled).

In 10-bit address mode, two address bytes need to be received by the slave. The five Most Significant bits (MSb) of the first address byte specify if this is a 10-bit address. The R/$\overline{W}$ bit (SSPSTAT register) must specify a write so the slave device will receive the second address byte. For a 10-bit address, the first byte would equal '1111 0 A9 A8 0', where A9 and A8 are the two MSb's of the address.

The sequence of events for 10-bit addressing is as follows, with steps 7-9 for slave-transmitter:

1. Receive first (high) byte of address (bit SSPIF of the PIR1 register and bits BF and UA of the SSPSTAT register are set).
2. Update the SSPADD register with second (low) byte of address (clears bit UA and releases the SCL line).
3. Read the SSPBUF register (clears bit BF) and clear flag bit SSPIF.
4. Receive second (low) byte of address (bits SSPIF, BF, and UA are set).
5. Update the SSPADD register with the first (high) byte of address. If match releases SCL line, this will clear bit UA.
6. Read the SSPBUF register (clears bit BF) and clear flag bit SSPIF.
7. Receive Repeated Start condition.
8. Receive first (high) byte of address (bits SSPIF and BF are set).
9. Read the SSPBUF register (clears bit BF) and clear flag bit SSPIF.

#### 13.4.1.2 Reception

When the R/$\overline{W}$ bit of the address byte is clear and an address match occurs, the R/$\overline{W}$ bit of the SSPSTAT register is cleared. The received address is loaded into the SSPBUF register.

When the address byte overflow condition exists, then no Acknowledge ($\overline{ACK}$) pulse is given. An overflow condition is defined as either bit BF (SSPSTAT register) is set, or bit SSPOV (SSPCON register) is set.

An MSSP interrupt is generated for each data transfer byte. Flag bit SSPIF of the PIR1 register must be cleared in software. The SSPSTAT register is used to determine the status of the byte.

#### 13.4.1.3 Transmission

When the R/$\overline{W}$ bit of the incoming address byte is set and an address match occurs, the R/$\overline{W}$ bit of the SSPSTAT register is set. The received address is loaded into the SSPBUF register. The $\overline{ACK}$ pulse will be sent on the ninth bit and pin RC3/SCK/SCL is held low. The transmit data must be loaded into the SSPBUF register, which also loads the SSPSR register. Then pin RC3/SCK/SCL should be enabled by setting bit CKP (SSPCON register). The master must monitor the SCL pin prior to asserting another clock pulse. The slave devices may be holding off the master by stretching the clock. The eight data bits are shifted out on the falling edge of the SCL input. This ensures that the SDA signal is valid during the SCL high time (Figure 13-8).

An MSSP interrupt is generated for each data transfer byte. The SSPIF bit must be cleared in software and the SSPSTAT register is used to determine the status of the byte. The SSPIF bit is set on the falling edge of the ninth clock pulse.

As a slave-transmitter, the $\overline{ACK}$ pulse from the master-receiver is latched on the rising edge of the ninth SCL input pulse. If the SDA line is high (not $\overline{ACK}$), then the data transfer is complete. When the $\overline{ACK}$ is latched by the slave, the slave logic is reset and the slave monitors for another occurrence of the Start bit. If the SDA line was low ($\overline{ACK}$), the transmit data must be loaded into the SSPBUF register, which also loads the SSPSR register. Pin RC3/SCK/SCL should be enabled by setting bit CKP.


© 2006-2012 Microchip Technology Inc.

**FIGURE 13-7: I²C™ SLAVE MODE WAVEFORMS FOR RECEPTION (7-BIT ADDRESS)**

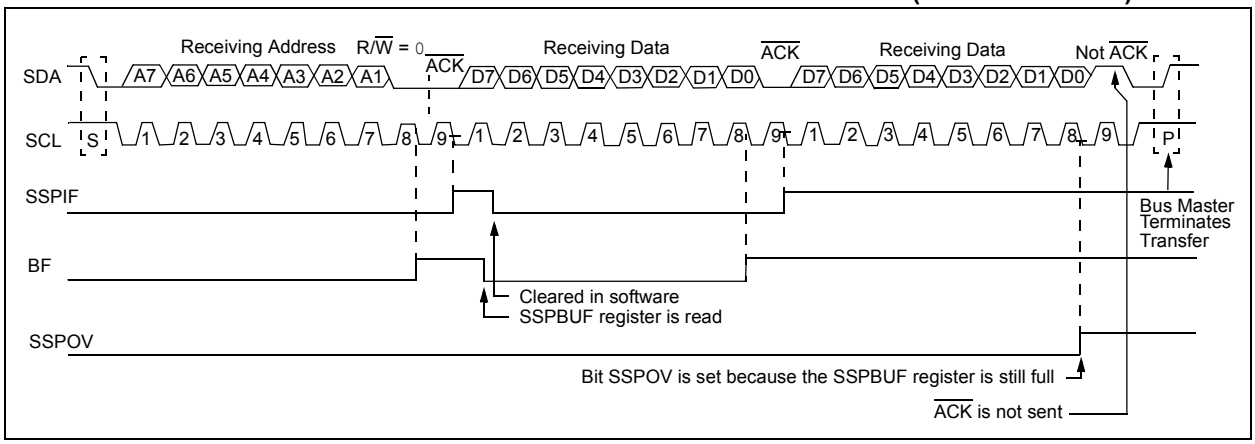
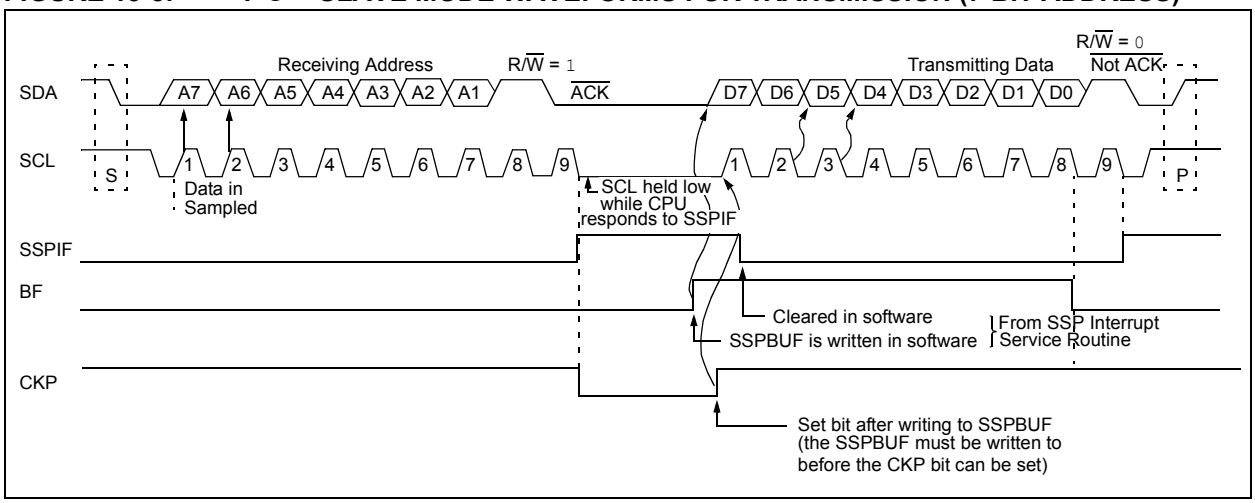



**FIGURE 13-8: I²C™ SLAVE MODE WAVEFORMS FOR TRANSMISSION (7-BIT ADDRESS)**



### 13.4.2 GENERAL CALL ADDRESS SUPPORT

The addressing procedure for the $I^2C$ bus is such that, the first byte after the Start condition usually determines which device will be the slave addressed by the master. The exception is the general call address, which can address all devices. When this address is used, all devices should, in theory, respond with an Acknowledge.

The general call address is one of eight addresses reserved for specific purposes by the $I^2C$ protocol. It consists of all 0's with R/$\overline{W}$ = 0.

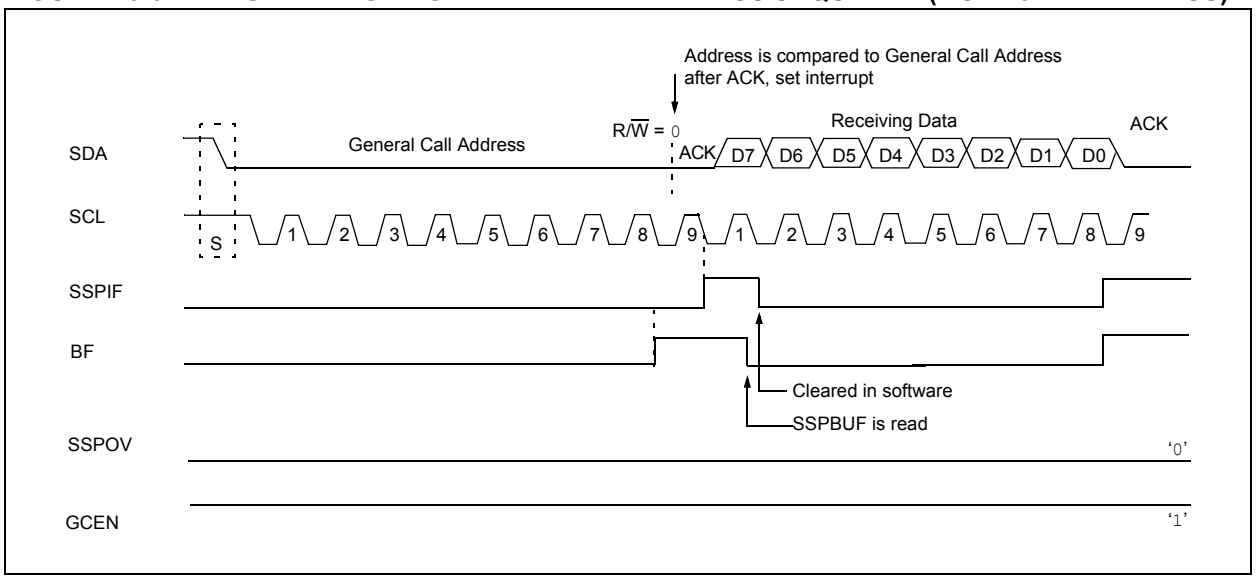
The general call address is recognized (enabled) when the General Call Enable (GCEN) bit is set (SSPCON2 register). Following a Start bit detect, eight bits are shifted into the SSPSR and the address is compared against the SSPADD. It is also compared to the general call address and fixed in hardware.

If the general call address matches, the SSPSR is transferred to the SSPBUF, the BF bit is set (eighth bit), and on the falling edge of the ninth bit ($\overline{ACK}$ bit), the SSPIF interrupt flag bit is set.

When the interrupt is serviced, the source for the interrupt can be checked by reading the contents of the SSPBUF. The value can be used to determine if the address was device specific or a general call address.

In 10-bit mode, the SSPADD is required to be updated for the second half of the address to match, and the UA bit is set (SSPSTAT register). If the general call address is sampled when the GCEN bit is set, and while the slave is configured in 10-bit address mode, then the second half of the address is not necessary. The UA bit will not be set, and the slave will begin receiving data after the Acknowledge (Figure 13-9).

**FIGURE 13-9: SLAVE MODE GENERAL CALL ADDRESS SEQUENCE (7 OR 10-BIT ADDRESS)**





© 2006-2012 Microchip Technology Inc.

### 13.4.3 MASTER MODE

Master mode of operation is supported by interrupt generation on the detection of the Start and Stop conditions. The Stop (P) and Start (S) bits are cleared from a Reset, or when the MSSP module is disabled. Control of the $I^2C$ bus may be taken when the P bit is set, or the bus is idle, with both the S and P bits clear.

In Master mode, the SCL and SDA lines are manipulated by the MSSP hardware.

The following events will cause SSP Interrupt Flag bit, SSPIF, to be set (SSP Interrupt if enabled):

- Start condition
- Stop condition
- Data transfer byte transmitted/received
- Acknowledge transmit
- Repeated Start condition

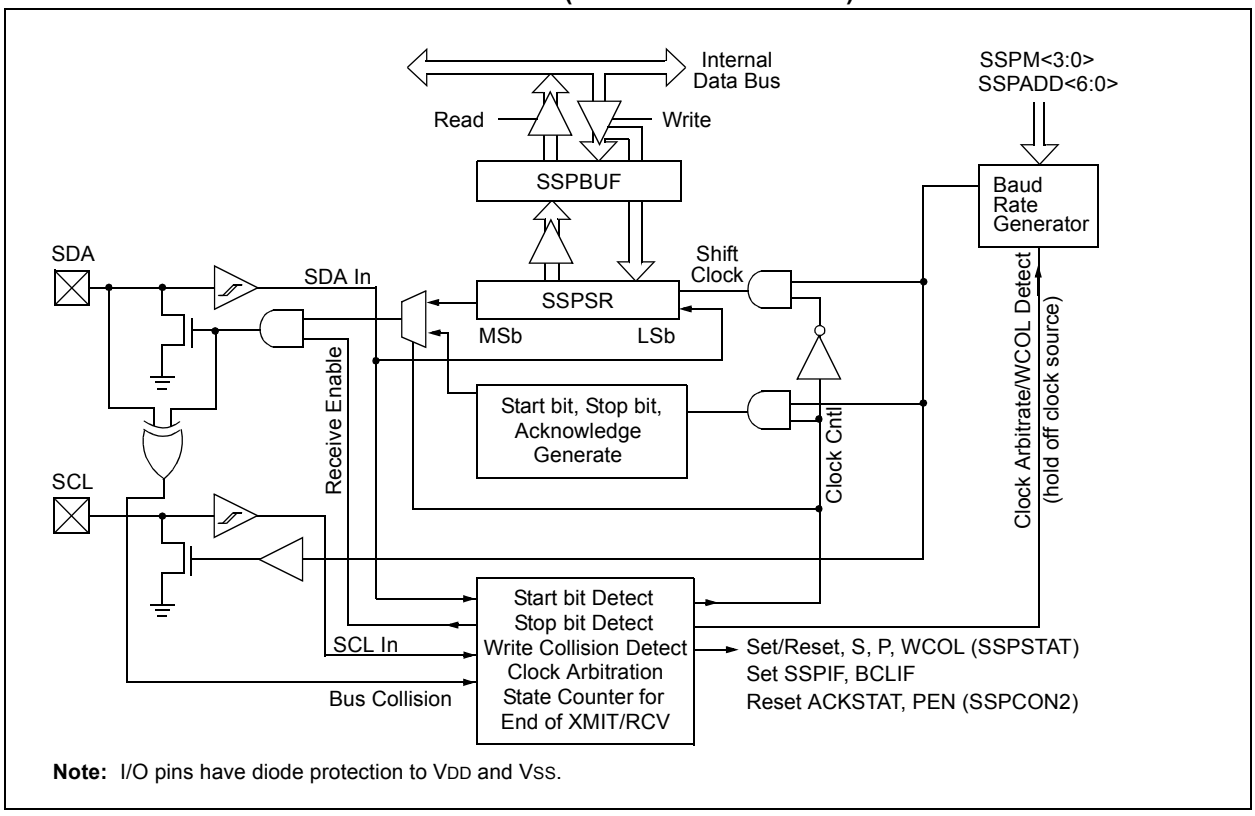
### 13.4.4 $I^2C$™ MASTER MODE SUPPORT

Master mode is enabled by setting and clearing the appropriate SSPM bits in SSPCON and by setting the SSPEN bit. Once Master mode is enabled, the user has the following six options:

1. Assert a Start condition on SDA and SCL.
2. Assert a Repeated Start condition on SDA and SCL.
3. Write to the SSPBUF register initiating transmission of data/address.
4. Generate a Stop condition on SDA and SCL.
5. Configure the $I^2C$ port to receive data.
6. Generate an Acknowledge condition at the end of a received byte of data.

**Note:** The MSSP module, when configured in $I^2C$ Master mode, does not allow queueing of events. For instance, the user is not allowed to initiate a Start condition and immediately write the SSPBUF register to imitate transmission, before the Start condition is complete. In this case, the SSPBUF will not be written to and the WCOL bit will be set, indicating that a write to the SSPBUF did not occur.

**FIGURE 13-10: MSSP BLOCK DIAGRAM ($I^2C$™ MASTER MODE)**




**Note:** I/O pins have diode protection to VDD and VSS.

#### 13.4.4.1 I²C™ Master Mode Operation

The master device generates all of the serial clock pulses and the Start and Stop conditions. A transfer is ended with a Stop condition or with a Repeated Start condition. Since the Repeated Start condition is also the beginning of the next serial transfer, the I²C bus will not be released.

In Master Transmitter mode, serial data is output through SDA, while SCL outputs the serial clock. The first byte transmitted contains the slave address of the receiving device (7 bits) and the Read/Write (R/$\overline{W}$) bit. In this case, the R/$\overline{W}$ bit will be logic '0'. Serial data is transmitted eight bits at a time. After each byte is transmitted, an Acknowledge bit is received. Start and Stop conditions are output to indicate the beginning and the end of a serial transfer.

In Master Receive mode, the first byte transmitted contains the slave address of the transmitting device (7 bits) and the R/$\overline{W}$ bit. In this case, the R/$\overline{W}$ bit will be logic '1'. Thus, the first byte transmitted is a 7-bit slave address followed by a '1' to indicate receive bit. Serial data is received via SDA, while SCL outputs the serial clock. Serial data is received eight bits at a time. After each byte is received, an Acknowledge bit is transmitted. Start and Stop conditions indicate the beginning and end of transmission.

The Baud Rate Generator used for the SPI mode operation is now used to set the SCL clock frequency for either 100 kHz, 400 kHz, or 1 MHz I²C operation. The Baud Rate Generator reload value is contained in the lower 7 bits of the SSPADD register. The Baud Rate Generator will automatically begin counting on a write to the SSPBUF. Once the given operation is complete (i.e., transmission of the last data bit is followed by ACK), the internal clock will automatically stop counting and the SCL pin will remain in its last state.

A typical transmit sequence would go as follows:

a) The user generates a Start condition by setting the Start Enable (SEN) bit (SSPCON2 register).
b) SSPIF is set. The MSSP module will wait the required start time before any other operation takes place.
c) The user loads the SSPBUF with the address to transmit.
d) Address is shifted out the SDA pin until all eight bits are transmitted.
e) The MSSP module shifts in the ACK bit from the slave device and writes its value into the ACKSTAT bit (SSPCON2 register).
f) The MSSP module generates an interrupt at the end of the ninth clock cycle by setting the SSPIF bit.
g) The user loads the SSPBUF with eight bits of data.
h) Data is shifted out the SDA pin until all eight bits are transmitted.
i) The MSSP module shifts in the ACK bit from the slave device and writes its value into the ACKSTAT bit (SSPCON2 register).
j) The MSSP module generates an interrupt at the end of the ninth clock cycle by setting the SSPIF bit.
k) The user generates a Stop condition by setting the Stop Enable bit PEN (SSPCON2 register).
l) Interrupt is generated once the Stop condition is complete.


© 2006-2012 Microchip Technology Inc.

### 13.4.5 BAUD RATE GENERATOR

In $I^2C$ Master mode, the reload value for the BRG is located in the lower 7 bits of the SSPADD register (Figure 13-11). When the BRG is loaded with this value, the BRG counts down to 0 and stops until another reload has taken place. The BRG count is decremented twice per instruction cycle (TCY) on the Q2 and Q4 clocks. In $I^2C$ Master mode, the BRG is reloaded automatically. If clock arbitration is taking place, for instance, the BRG will be reloaded when the SCL pin is sampled high (Figure 13-12).

**FIGURE 13-11: BAUD RATE GENERATOR BLOCK DIAGRAM**

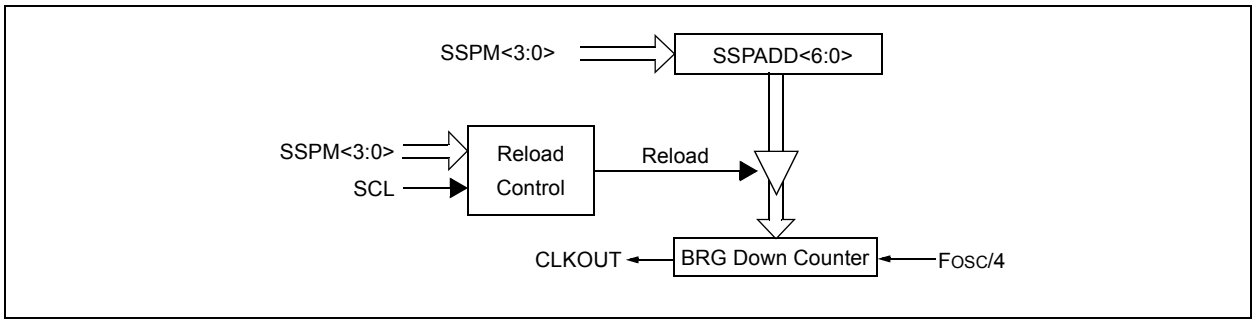
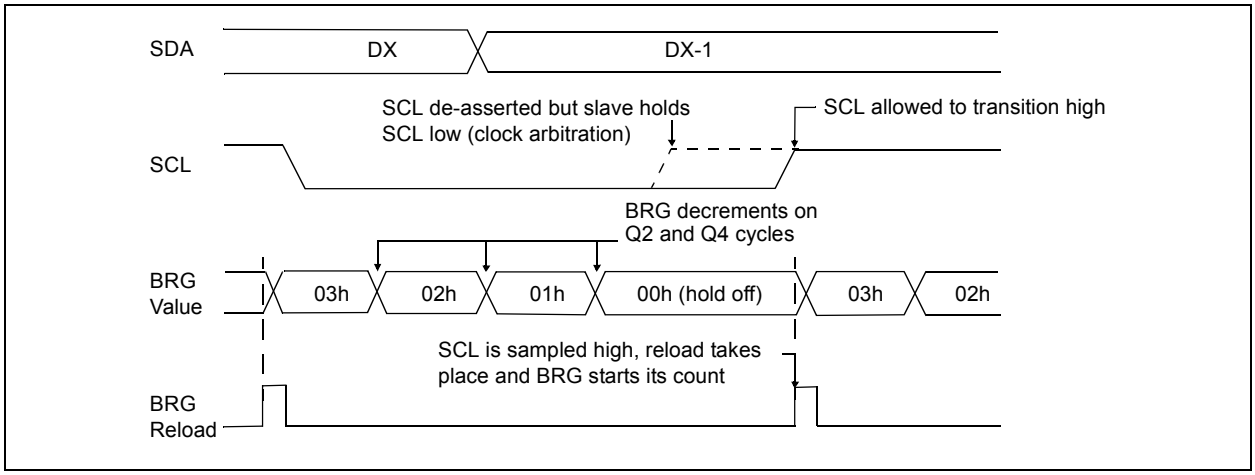



**FIGURE 13-12: BAUD RATE GENERATOR TIMING WITH CLOCK ARBITRATION**



### 13.4.6 I²C™ MASTER MODE START CONDITION TIMING

To initiate a Start condition, the user sets the Start Condition Enable bit SEN of the SSPCON2 register. If the SDA and SCL pins are sampled high, the Baud Rate Generator is reloaded with the contents of SSPADD<6:0> and starts its count. If SCL and SDA are both sampled high when the Baud Rate Generator times out (TBRG), the SDA pin is driven low. The action of the SDA being driven low, while SCL is high, is the Start condition, and causes the S bit of the SSPSTAT register to be set. Following this, the Baud Rate Generator is reloaded with the contents of SSPADD<6:0> and resumes its count. When the Baud Rate Generator times out (TBRG), the SEN bit of the SSPCON2 register will be automatically cleared by hardware, the Baud Rate Generator is suspended leaving the SDA line held low and the Start condition is complete.

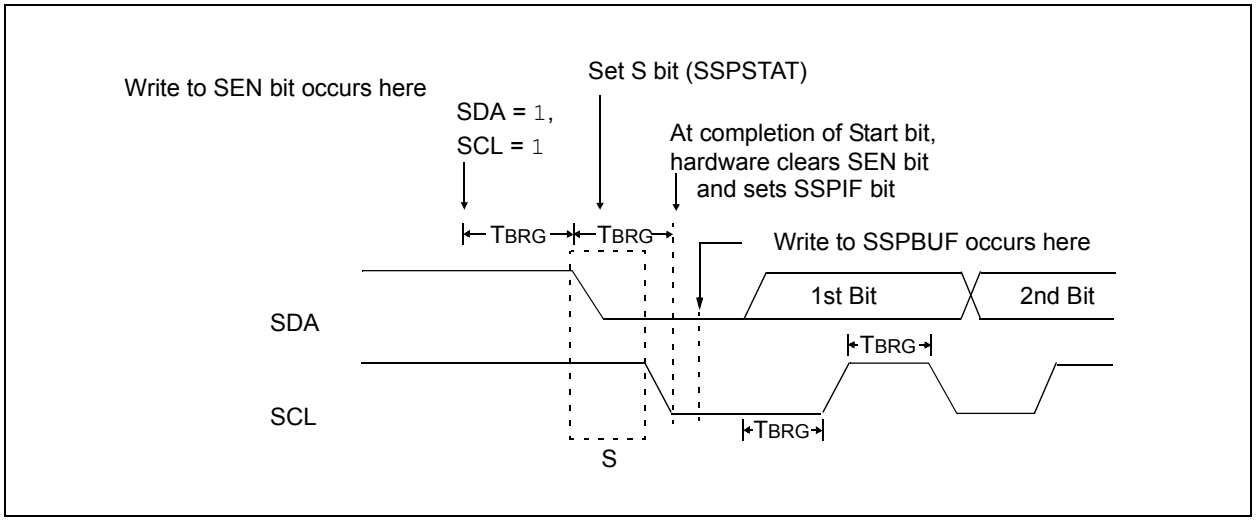
| | |
|---|---|
| **Note:** | If, at the beginning of the Start condition, the SDA and SCL pins are already sampled low, or if during the Start condition the SCL line is sampled low before the SDA line is driven low, a bus collision occurs, the Bus Collision Interrupt Flag, BCLIF, is set, the Start condition is aborted, and the I²C module is reset into its Idle state. |

#### 13.4.6.1 WCOL Status Flag

If the user writes the SSPBUF when a Start sequence is in progress, the WCOL is set and the contents of the buffer are unchanged (the write doesn't occur).

| | |
|---|---|
| **Note:** | Because queueing of events is not allowed, writing to the lower 5 bits of SSPCON2 is disabled until the Start condition is complete. |

**FIGURE 13-13: FIRST START BIT TIMING**





© 2006-2012 Microchip Technology Inc.

### 13.4.7 I²C™ MASTER MODE REPEATED START CONDITION TIMING

A Repeated Start condition occurs when the RSEN bit (SSPCON2 register) is programmed high and the I²C logic module is in the Idle state. When the RSEN bit is set, the SCL pin is asserted low. When the SCL pin is sampled low, the Baud Rate Generator is loaded with the contents of SSPADD<5:0> and begins counting. The SDA pin is released (brought high) for one Baud Rate Generator count (TBRG). When the Baud Rate Generator times out, if SDA is sampled high, the SCL pin will be de-asserted (brought high). When SCL is sampled high, the Baud Rate Generator is reloaded with the contents of SSPADD<6:0> and begins counting. SDA and SCL must be sampled high for one TBRG. This action is then followed by assertion of the SDA pin (SDA = 0) for one TBRG, while SCL is high. Following this, the RSEN bit (SSPCON2 register) will be automatically cleared and the Baud Rate Generator will not be reloaded, leaving the SDA pin held low. As soon as a Start condition is detected on the SDA and SCL pins, the S bit (SSPSTAT register) will be set. The SSPIF bit will not be set until the Baud Rate Generator has timed out.

**Note 1:** If RSEN is programmed while any other event is in progress, it will not take effect.

**2:** A bus collision during the Repeated Start condition occurs if:

- SDA is sampled low when SCL goes from low-to-high.
- SCL goes low before SDA is asserted low. This may indicate that another master is attempting to transmit a data "1".

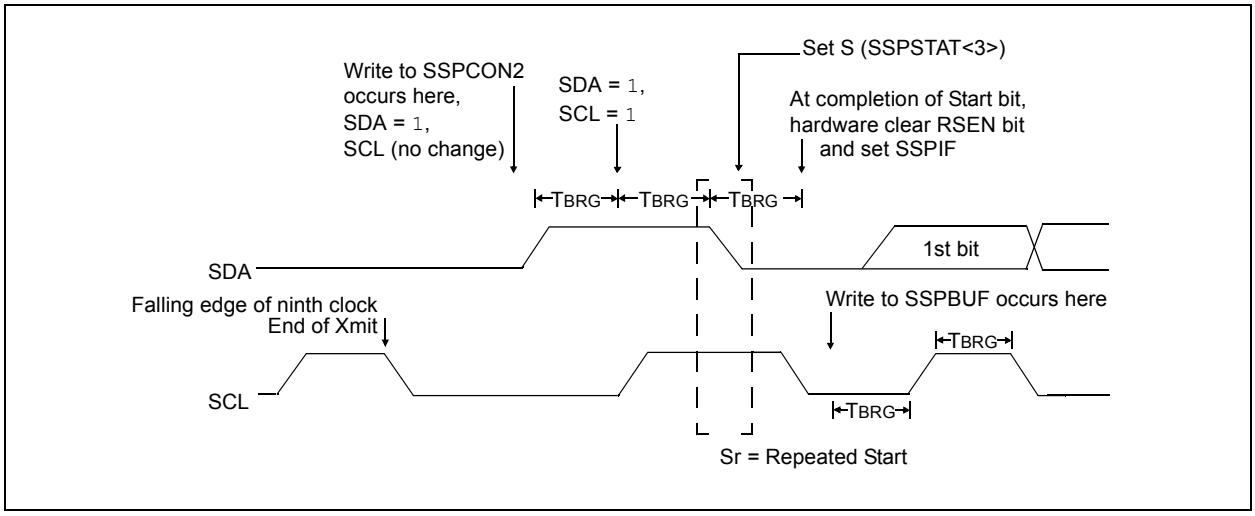
Immediately following the SSPIF bit getting set, the user may write the SSPBUF with the 7-bit address in 7-bit mode, or the default first address in 10-bit mode. After the first eight bits are transmitted and an ACK is received, the user may then transmit an additional eight bits of address (10-bit mode), or eight bits of data (7-bit mode).

#### 13.4.7.1 WCOL Status Flag

If the user writes the SSPBUF when a Repeated Start sequence is in progress, the WCOL is set and the contents of the buffer are unchanged (the write doesn't occur).

**Note:** Because queueing of events is not allowed, writing of the lower 5 bits of SSPCON2 is disabled until the Repeated Start condition is complete.

**FIGURE 13-14: REPEAT START CONDITION WAVEFORM**



### 13.4.8 I²C™ MASTER MODE TRANSMISSION

Transmission of a data byte, a 7-bit address, or the other half of a 10-bit address, is accomplished by simply writing a value to the SSPBUF register. This action will set the Buffer Full bit, BF, and allow the Baud Rate Generator to begin counting and start the next transmission. Each bit of address/data will be shifted out onto the SDA pin after the falling edge of SCL is asserted (see data hold time specification, parameter 106). SCL is held low for one Baud Rate Generator rollover count (TBRG). Data should be valid before SCL is released high (see data setup time specification, parameter 107). When the SCL pin is released high, it is held that way for TBRG. The data on the SDA pin must remain stable for that duration and some hold time after the next falling edge of SCL. After the eighth bit is shifted out (the falling edge of the eighth clock), the BF bit is cleared and the master releases SDA, allowing the slave device being addressed to respond with an ACK bit during the ninth bit time, if an address match occurs, or if data was received properly. The status of ACK is written into the ACKDT bit on the falling edge of the ninth clock. If the master receives an Acknowledge, the Acknowledge Status bit, ACKSTAT, is cleared. If not, the bit is set. After the ninth clock, the SSPIF bit is set and the master clock (Baud Rate Generator) is suspended until the next data byte is loaded into the SSPBUF, leaving SCL low and SDA unchanged (Figure 13-15).

After the write to the SSPBUF, each bit of the address will be shifted out on the falling edge of SCL, until all seven address bits and the R/W bit, are completed. On the falling edge of the eighth clock, the master will deassert the SDA pin, allowing the slave to respond with an Acknowledge. On the falling edge of the ninth clock, the master will sample the SDA pin to see if the address was recognized by a slave. The status of the ACK bit is loaded into the ACKSTAT Status bit (SSPCON2 register). Following the falling edge of the ninth clock transmission of the address, the SSPIF is set, the BF bit is cleared and the Baud Rate Generator is turned off, until another write to the SSPBUF takes place, holding SCL low and allowing SDA to float.

#### 13.4.8.1 BF Status Flag

In Transmit mode, the BF bit (SSPSTAT register) is set when the CPU writes to SSPBUF, and is cleared when all eight bits are shifted out.

#### 13.4.8.2 WCOL Status Flag

If the user writes the SSPBUF when a transmit is already in progress (i.e., SSPSR is still shifting out a data byte), the WCOL is set and the contents of the buffer are unchanged (the write doesn't occur). WCOL must be cleared in software.

#### 13.4.8.3 ACKSTAT Status Flag

In Transmit mode, the ACKSTAT bit (SSPCON2 register) is cleared when the slave has sent an Acknowledge (ACK = 0), and is set when the slave does not Acknowledge (ACK = 1). A slave sends an Acknowledge when it has recognized its address (including a general call), or when the slave has properly received its data.

### 13.4.9 I²C™ MASTER MODE RECEPTION

Master mode reception is enabled by programming the Receive Enable bit, RCEN (SSPCON2 register).

| **Note:** | The MSSP module must be in an Idle state before the RCEN bit is set, or the RCEN bit will be disregarded. |
|---|---|

The Baud Rate Generator begins counting, and on each rollover, the state of the SCL pin changes (high-to-low/low-to-high) and data is shifted into the SSPSR. After the falling edge of the eighth clock, the RCEN bit is automatically cleared, the contents of the SSPSR are loaded into the SSPBUF, the BF bit is set, the SSPIF flag bit is set and the Baud Rate Generator is suspended from counting, holding SCL low. The MSSP is now in Idle state, awaiting the next command. When the buffer is read by the CPU, the BF bit is automatically cleared. The user can then send an Acknowledge bit at the end of reception, by setting the Acknowledge Sequence Enable bit ACKEN (SSPCON2 register).

#### 13.4.9.1 BF Status Flag

In receive operation, the BF bit is set when an address or data byte is loaded into SSPBUF from SSPSR. It is cleared when the SSPBUF register is read.

#### 13.4.9.2 SSPOV Status Flag

In receive operation, the SSPOV bit is set when eight bits are received into the SSPSR and the BF bit is already set from a previous reception.

#### 13.4.9.3 WCOL Status Flag

If the user writes the SSPBUF when a receive is already in progress (i.e., SSPSR is still shifting in a data byte), the WCOL bit is set and the contents of the buffer are unchanged (the write doesn't occur).


© 2006-2012 Microchip Technology Inc.

**FIGURE 13-15: I²C™ MASTER MODE WAVEFORM (TRANSMISSION, 7 OR 10-BIT ADDRESS)**

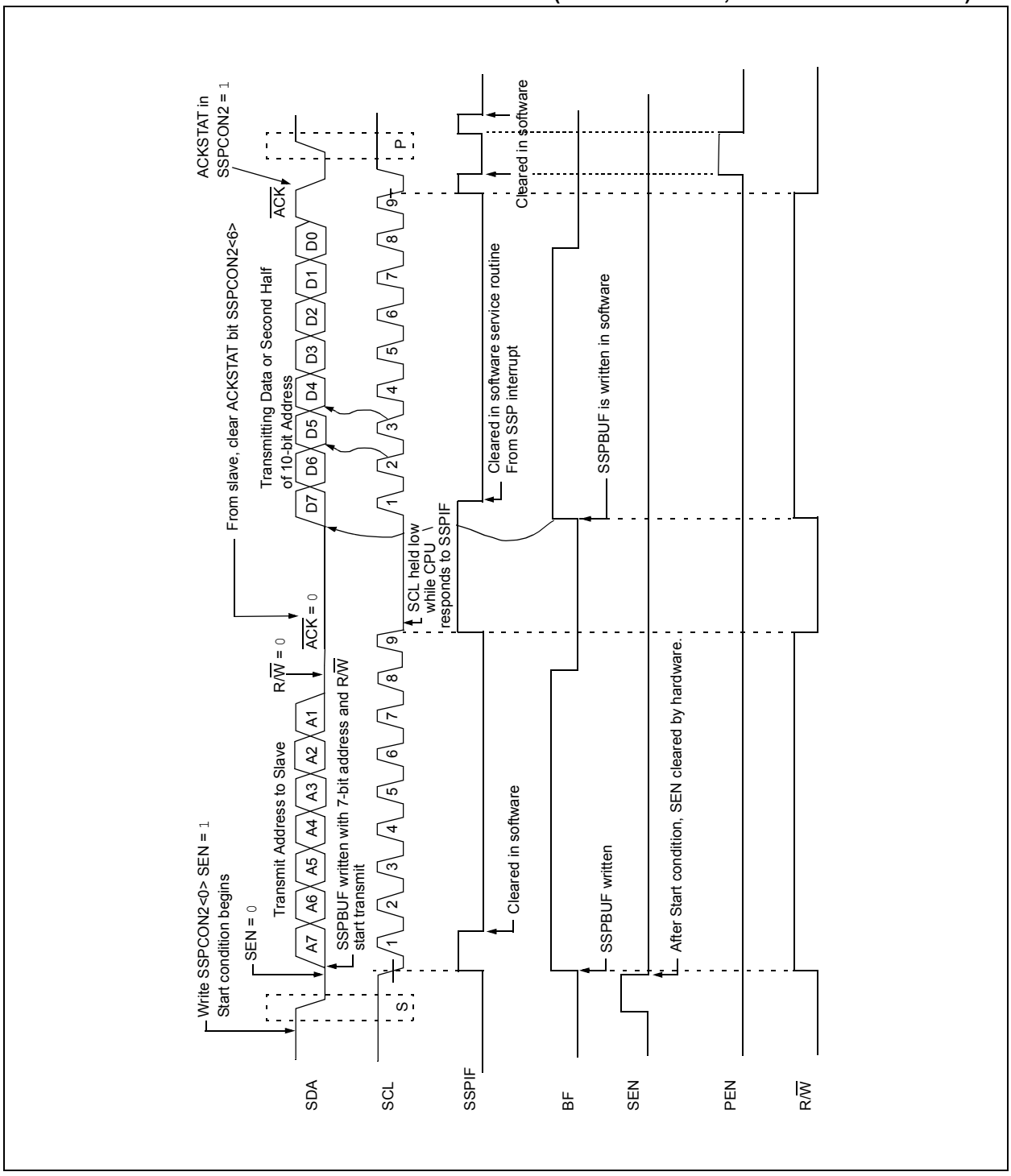
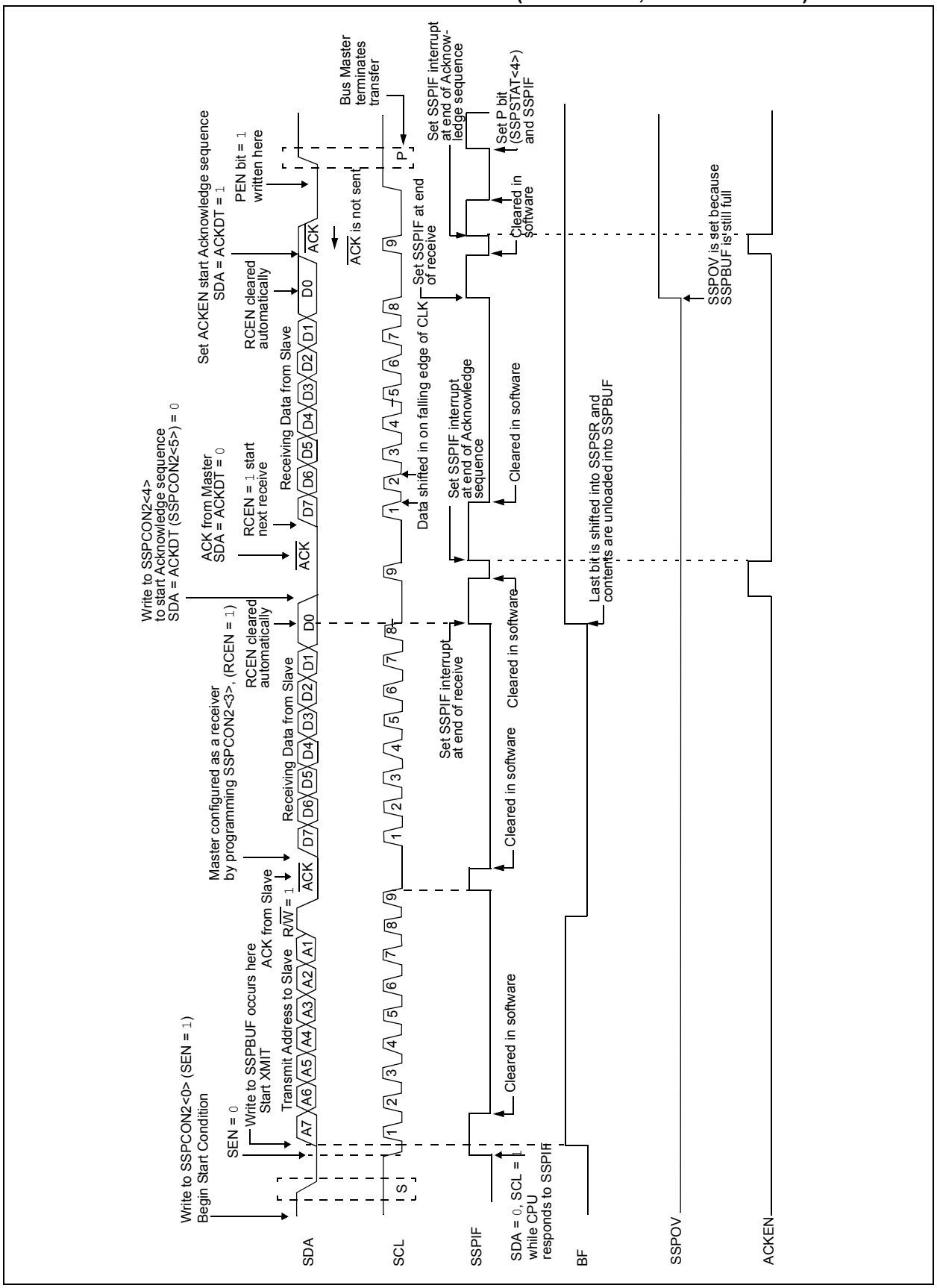


**FIGURE 13-16: I²C™ MASTER MODE WAVEFORM (RECEPTION, 7-BIT ADDRESS)**





© 2006-2012 Microchip Technology Inc.

### 13.4.10 ACKNOWLEDGE SEQUENCE TIMING

An Acknowledge sequence is enabled by setting the Acknowledge Sequence Enable bit, ACKEN (SSPCON2 register). When this bit is set, the SCL pin is pulled low and the contents of the Acknowledge Data bit (ACKDT) is presented on the SDA pin. If the user wishes to generate an Acknowledge, then the ACKDT bit should be cleared. If not, the user should set the ACKDT bit before starting an Acknowledge sequence. The Baud Rate Generator then counts for one rollover period (TBRG) and the SCL pin is de-asserted (pulled high). When the SCL pin is sampled high (clock arbitration), the Baud Rate Generator counts for TBRG. The SCL pin is then pulled low. Following this, the ACKEN bit is automatically cleared, the Baud Rate Generator is turned off and the MSSP module then goes into Idle mode (Figure 13-17).

#### 13.4.10.1 WCOL Status Flag

If the user writes the SSPBUF when an Acknowledge sequence is in progress, then WCOL is set and the contents of the buffer are unchanged (the write doesn't occur).

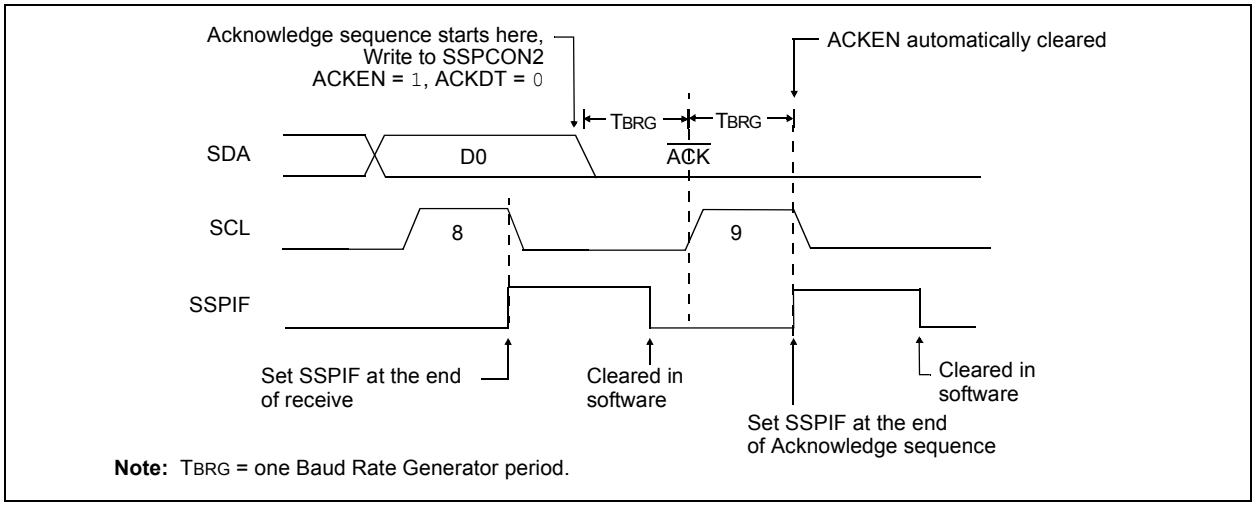
### 13.4.11 STOP CONDITION TIMING

A Stop bit is asserted on the SDA pin at the end of a receive/transmit by setting the Stop Sequence Enable bit, PEN (SSPCON2 register). At the end of a receive/transmit, the SCL line is held low after the falling edge of the ninth clock. When the PEN bit is set, the master will assert the SDA line low. When the SDA line is sampled low, the Baud Rate Generator is reloaded and counts down to 0. When the Baud Rate Generator times out, the SCL pin will be brought high, and one TBRG (Baud Rate Generator rollover count) later, the SDA pin will be de-asserted. When the SDA pin is sampled high while SCL is high, the P bit (SSPSTAT register) is set. A TBRG later, the PEN bit is cleared and the SSPIF bit is set (Figure 13-18).

#### 13.4.11.1 WCOL Status Flag

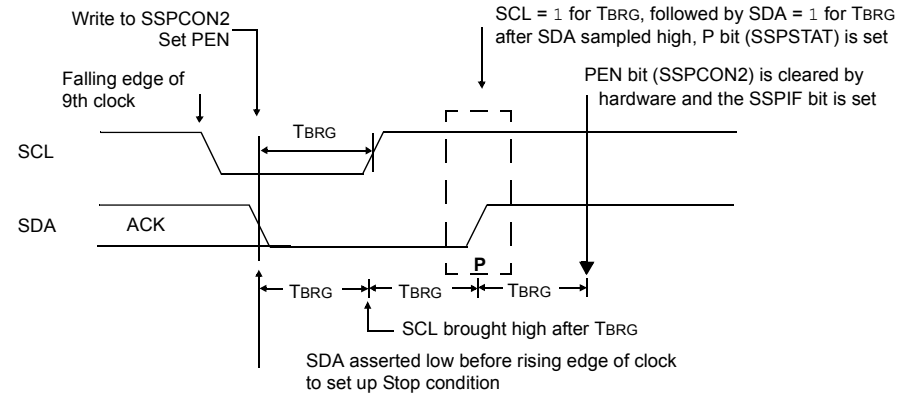
If the user writes the SSPBUF when a Stop sequence is in progress, then the WCOL bit is set and the contents of the buffer are unchanged (the write doesn't occur).

**FIGURE 13-17: ACKNOWLEDGE SEQUENCE WAVEFORM**




**Note:** TBRG = one Baud Rate Generator period.

**FIGURE 13-18: STOP CONDITION RECEIVE OR TRANSMIT MODE**




**Note:** TBRG = one Baud Rate Generator period.

### 13.4.12 CLOCK ARBITRATION

Clock arbitration occurs when the master, during any receive, transmit or Repeated Start/Stop condition, de-asserts the SCL pin (SCL allowed to float high). When the SCL pin is allowed to float high, the Baud Rate Generator (BRG) is suspended from counting until the SCL pin is actually sampled high. When the SCL pin is sampled high, the Baud Rate Generator is reloaded with the contents of SSPADD<6:0> and begins counting. This ensures that the SCL high time will always be at least one BRG rollover count, in the event that the clock is held low by an external device (Figure 13-19).

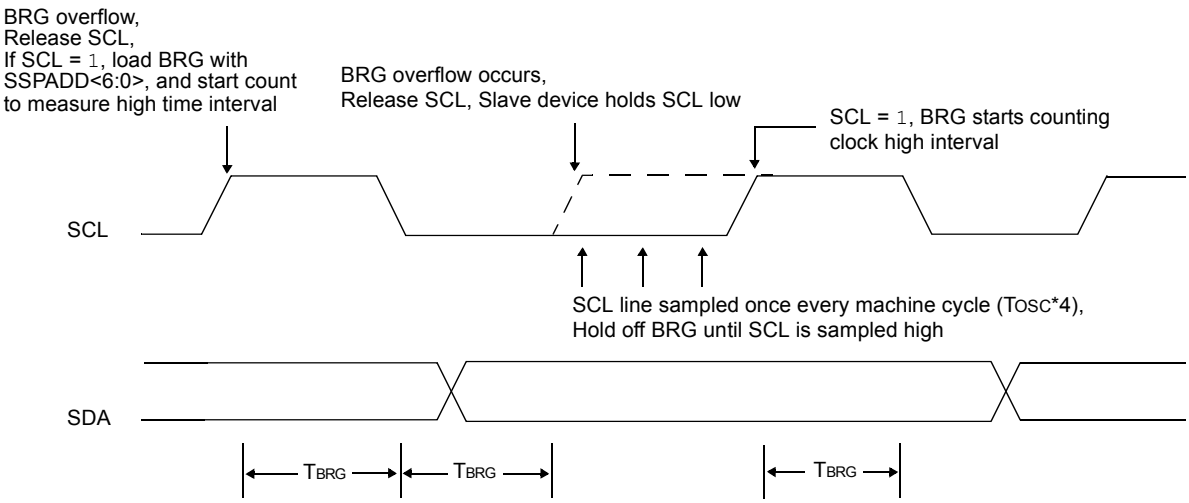
### 13.4.13 SLEEP OPERATION

While in Sleep mode, the $I^2C$ module can receive addresses or data, and when an address match or complete byte transfer occurs, wake the processor from Sleep (if the MSSP interrupt is enabled).

### 13.4.14 EFFECT OF A RESET

A Reset disables the MSSP module and terminates the current transfer.

**FIGURE 13-19: CLOCK ARBITRATION TIMING IN MASTER TRANSMIT MODE**





© 2006-2012 Microchip Technology Inc.

### 13.4.15 MULTI-MASTER MODE

In Multi-Master mode, the interrupt generation on the detection of the Start and Stop conditions allows the determination of when the bus is free. The Stop (P) and Start (S) bits are cleared from a Reset, or when the MSSP module is disabled. Control of the $I^2C$ bus may be taken when the P bit (SSPSTAT register) is set, or the bus is idle with both the S and P bits clear. When the bus is busy, enabling the SSP Interrupt will generate the interrupt when the Stop condition occurs.

In Multi-Master operation, the SDA line must be monitored for arbitration, to see if the signal level is the expected output level. This check is performed in hardware, with the result placed in the BCLIF bit.

Arbitration can be lost in the following states:

- Address transfer
- Data transfer
- A Start condition
- A Repeated Start condition
- An Acknowledge condition

### 13.4.16 MULTI -MASTER COMMUNICATION, BUS COLLISION, AND BUS ARBITRATION

Multi-Master mode support is achieved by bus arbitration. When the master outputs address/data bits onto the SDA pin, arbitration takes place when the master outputs a '1' on SDA, by letting SDA float high and another master asserts a '0'. When the SCL pin floats high, data should be stable. If the expected data on SDA is a '1' and the data sampled on the SDA pin = 0, then a bus collision has taken place. The master will set the Bus Collision Interrupt Flag (BCLIF) and reset the $I^2C$ port to its Idle state (Figure 13-20).

If a transmit was in progress when the bus collision occurred, the transmission is halted, the BF bit is cleared, the SDA and SCL lines are de-asserted, and the SSPBUF can be written to. When the user services the bus collision interrupt service routine, and if the $I^2C$ bus is free, the user can resume communication by asserting a Start condition.

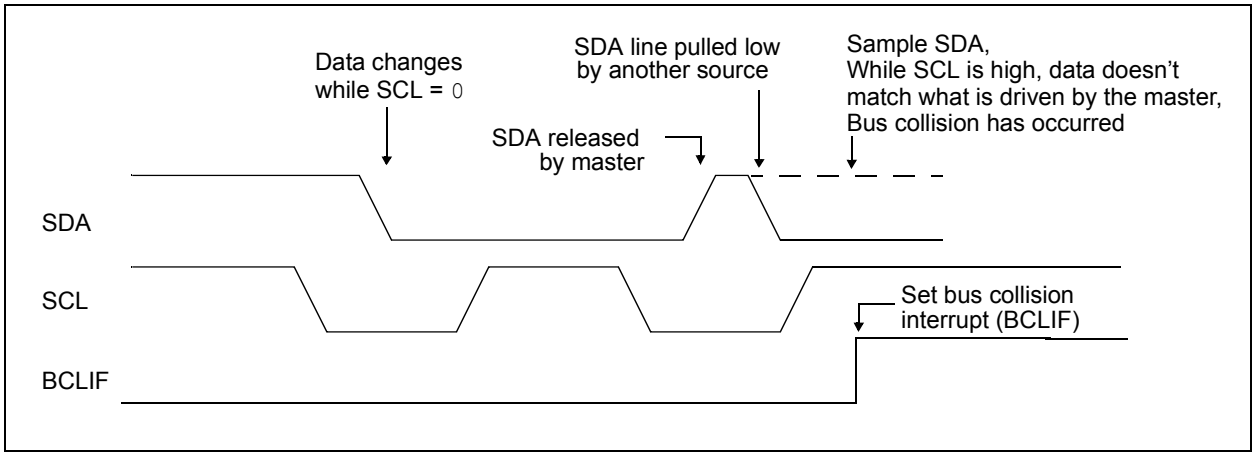
If a Start, Repeated Start, Stop, or Acknowledge condition was in progress when the bus collision occurred, the condition is aborted, the SDA and SCL lines are de-asserted, and the respective control bits in the SSPCON2 register are cleared. When the user services the bus collision interrupt service routine, and if the $I^2C$ bus is free, the user can resume communication by asserting a Start condition.

The master will continue to monitor the SDA and SCL pins. If a Stop condition occurs, the SSPIF bit will be set.

A write to the SSPBUF will start the transmission of data at the first data bit, regardless of where the transmitter left off when the bus collision occurred.

In Multi-Master mode, the interrupt generation on the detection of Start and Stop conditions allows the determination of when the bus is free. Control of the $I^2C$ bus can be taken when the P bit is set in the SSPSTAT register, or the bus is idle and the S and P bits are cleared.

**FIGURE 13-20: BUS COLLISION TIMING FOR TRANSMIT AND ACKNOWLEDGE**



#### 13.4.16.1 Bus Collision During a Start Condition

During a Start condition, a bus collision occurs if:

a) SDA or SCL are sampled low at the beginning of the Start condition (Figure 13-21).

b) SCL is sampled low before SDA is asserted low (Figure 13-22).

During a Start condition, both the SDA and the SCL pins are monitored, if:

the SDA pin is already low,
or the SCL pin is already low,

then:

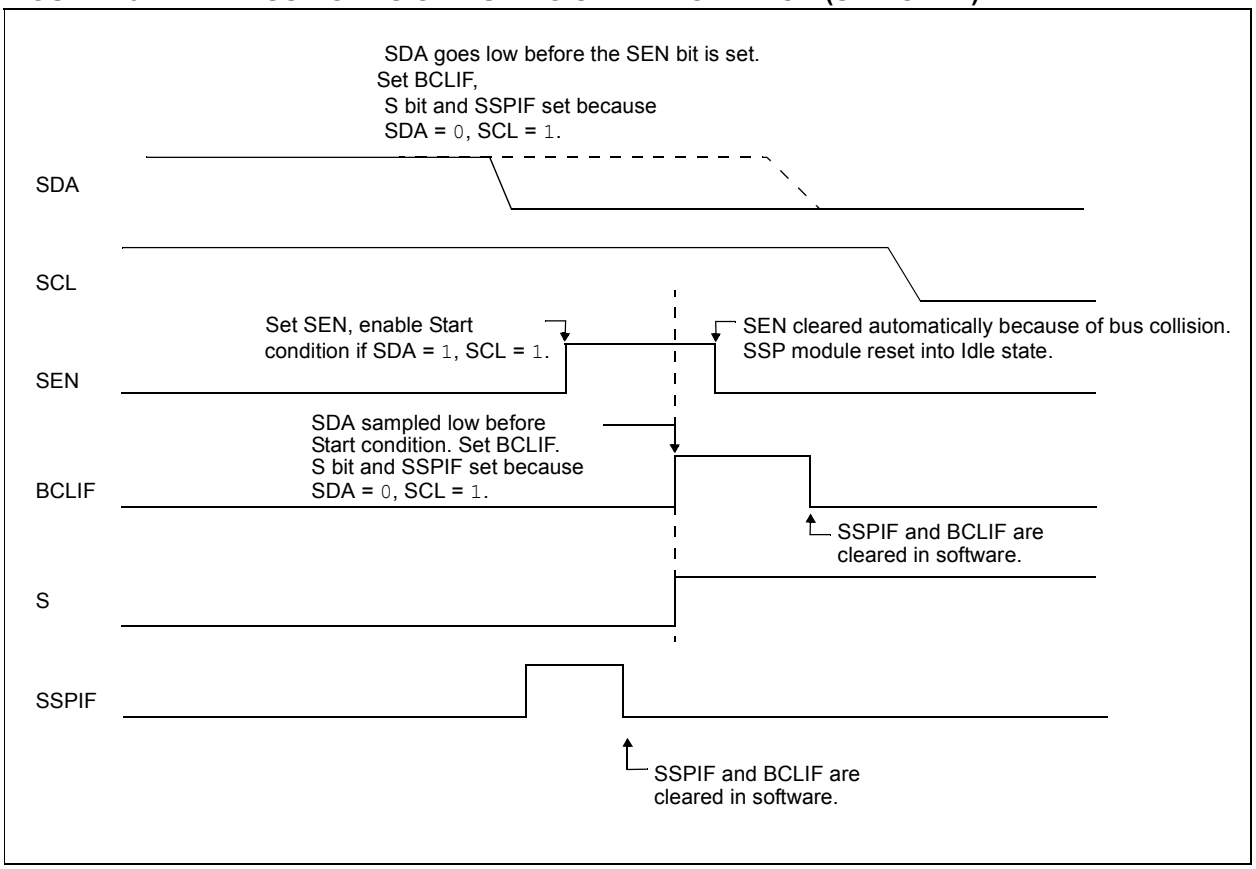
the Start condition is aborted,
and the BCLIF flag is set,
and the MSSP module is reset to its Idle state (Figure 13-21).

The Start condition begins with the SDA and SCL pins de-asserted. When the SDA pin is sampled high, the Baud Rate Generator is loaded from SSPADD<6:0> and counts down to 0. If the SCL pin is sampled low while SDA is high, a bus collision occurs, because it is assumed that another master is attempting to drive a data '1' during the Start condition.

If the SDA pin is sampled low during this count, the BRG is reset and the SDA line is asserted early (Figure 13-23). If, however, a '1' is sampled on the SDA pin, the SDA pin is asserted low at the end of the BRG count. The Baud Rate Generator is then reloaded and counts down to 0, and during this time, if the SCL pin is sampled as '0', a bus collision does not occur. At the end of the BRG count, the SCL pin is asserted low.

**Note:** The reason that bus collision is not a factor during a Start condition, is that no two bus masters can assert a Start condition at the exact same time. Therefore, one master will always assert SDA before the other. This condition does not cause a bus collision, because the two masters must be allowed to arbitrate the first address following the Start condition. If the address is the same, arbitration must be allowed to continue into the data portion, Repeated Start or Stop conditions.

**FIGURE 13-21: BUS COLLISION DURING START CONDITION (SDA ONLY)**





© 2006-2012 Microchip Technology Inc.

**FIGURE 13-22: BUS COLLISION DURING START CONDITION (SCL = 0)**

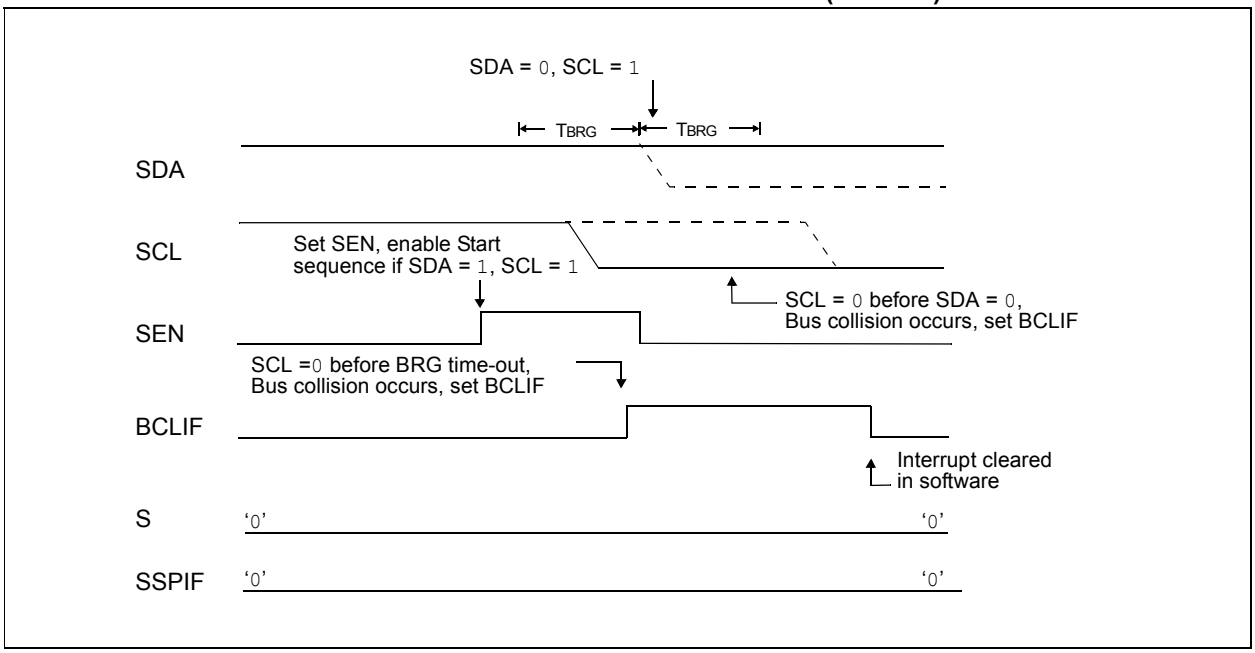
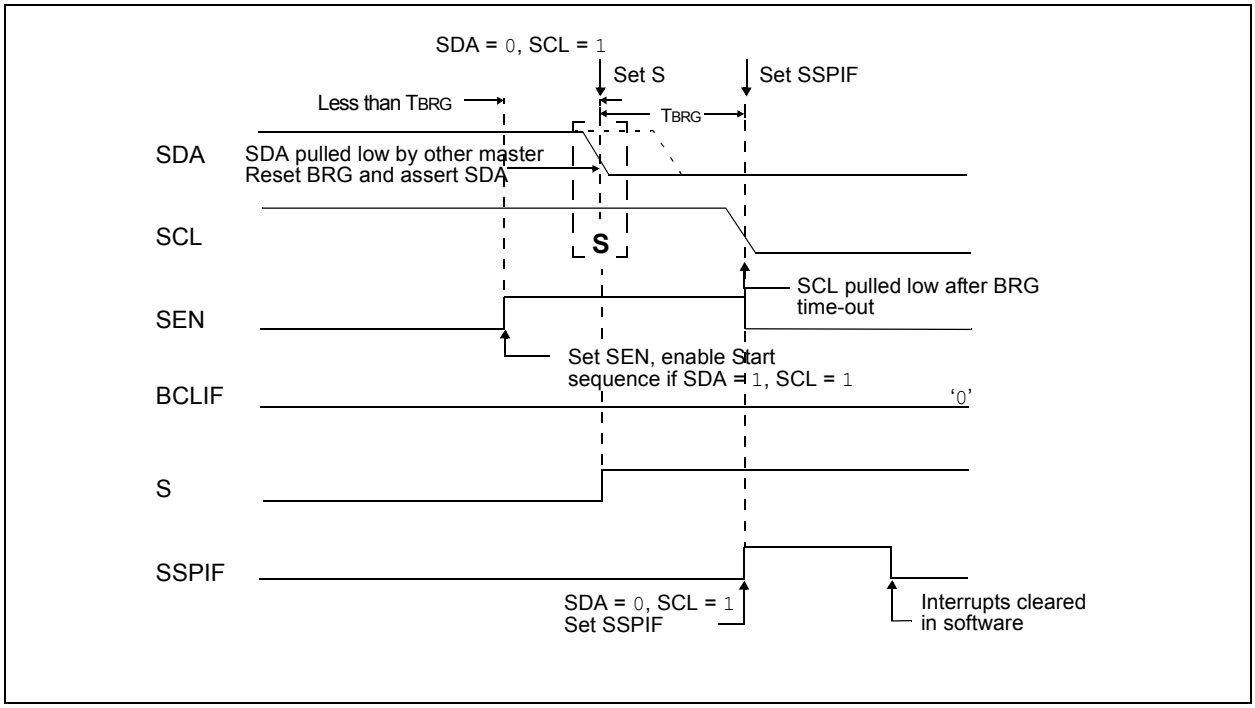



**FIGURE 13-23: BRG RESET DUE TO SDA ARBITRATION DURING START CONDITION**



### 13.4.16.2 Bus Collision During a Repeated Start Condition

During a Repeated Start condition, a bus collision occurs if:

a) A low level is sampled on SDA when SCL goes from low level to high level.

b) SCL goes low before SDA is asserted low, indicating that another master is attempting to transmit a data '1'.

When the user de-asserts SDA and the pin is allowed to float high, the BRG is loaded with SSPADD<6:0> and counts down to 0. The SCL pin is then de-asserted, and when sampled high, the SDA pin is sampled.

If SDA is low, a bus collision has occurred (i.e, another master is attempting to transmit a data '0', see Figure 13-24). If SDA is sampled high, the BRG is reloaded and begins counting. If SDA goes from high-to-low before the BRG times out, no bus collision occurs because no two masters can assert SDA at exactly the same time.

If SCL goes from high-to-low before the BRG times out and SDA has not already been asserted, a bus collision occurs. In this case, another master is attempting to transmit a data '1' during the Repeated Start condition (Figure 13-25).

If at the end of the BRG time-out, both SCL and SDA are still high, the SDA pin is driven low and the BRG is reloaded and begins counting. At the end of the count, regardless of the status of the SCL pin, the SCL pin is driven low and the Repeated Start condition is complete.

**FIGURE 13-24: BUS COLLISION DURING A REPEATED START CONDITION (CASE 1)**

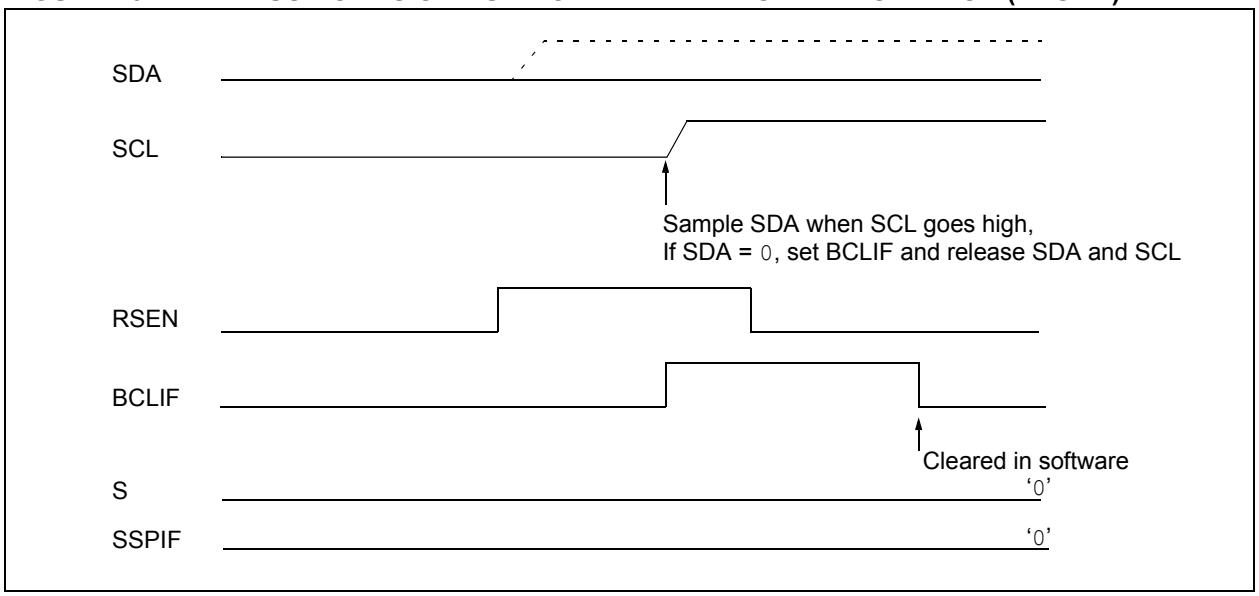
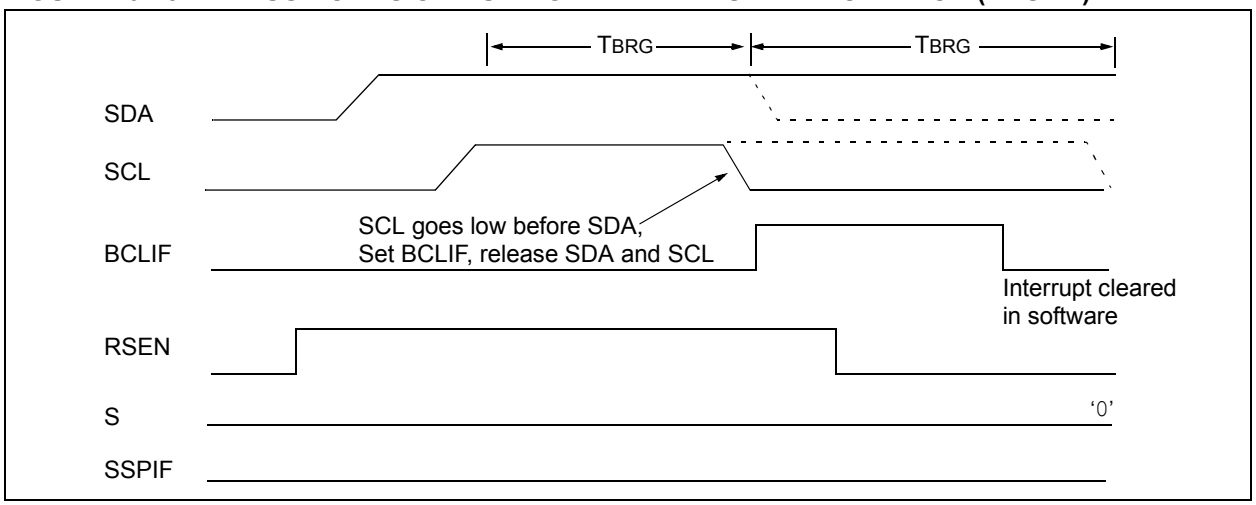



**FIGURE 13-25: BUS COLLISION DURING REPEATED START CONDITION (CASE 2)**





© 2006-2012 Microchip Technology Inc.

### 13.4.16.3 Bus Collision During a Stop Condition

Bus collision occurs during a Stop condition if:

a) After the SDA pin has been de-asserted and allowed to float high, SDA is sampled low after the BRG has timed out.

b) After the SCL pin is de-asserted, SCL is sampled low before SDA goes high.

The Stop condition begins with SDA asserted low. When SDA is sampled low, the SCL pin is allowed to float. When the pin is sampled high (clock arbitration), the Baud Rate Generator is loaded with SSPADD<6:0> and counts down to 0. After the BRG times out, SDA is sampled. If SDA is sampled low, a bus collision has occurred. This is due to another master attempting to drive a data '0' (Figure 13-26). If the SCL pin is sampled low before SDA is allowed to float high, a bus collision occurs. This is another case of another master attempting to drive a data '0' (Figure 13-27).

**FIGURE 13-26: BUS COLLISION DURING A STOP CONDITION (CASE 1)**

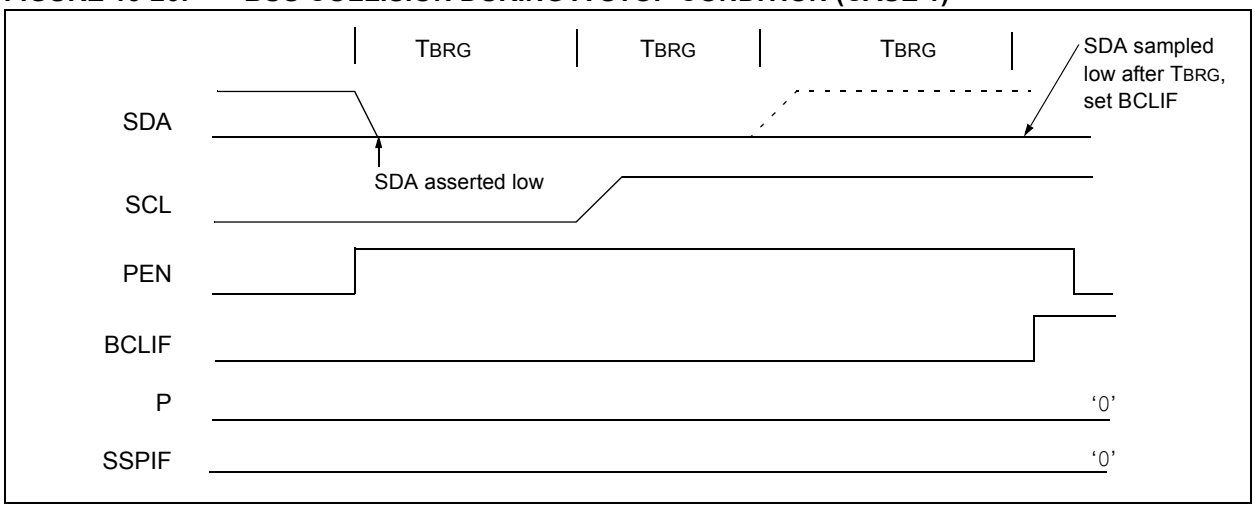
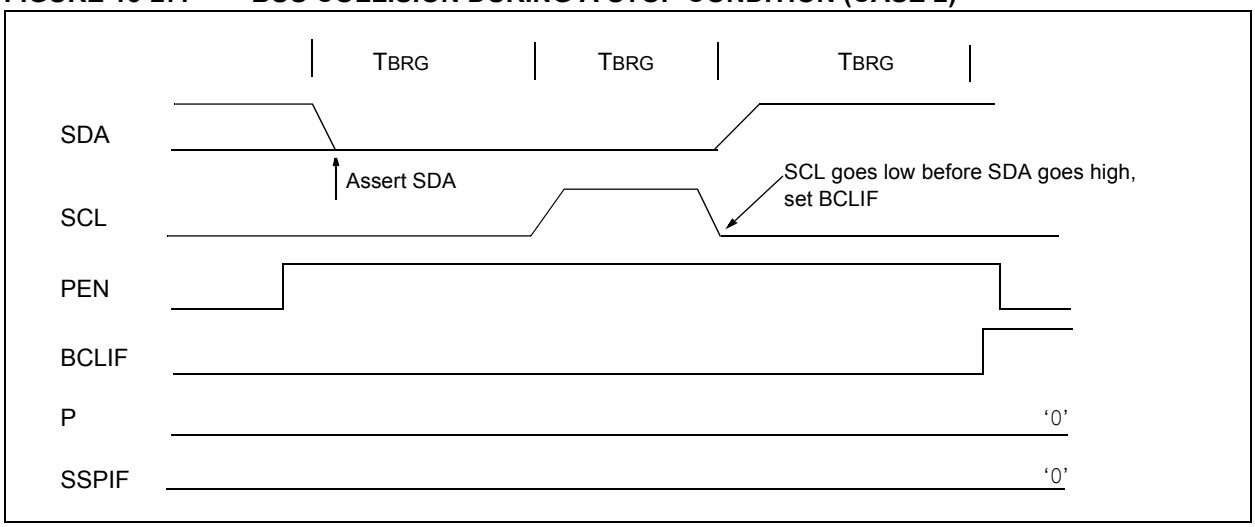



**FIGURE 13-27: BUS COLLISION DURING A STOP CONDITION (CASE 2)**



### 13.4.17 SSP MASK REGISTER

An SSP Mask (SSPMSK) register is available in $I^2C$ Slave mode as a mask for the value held in the SSPSR register during an address comparison operation. A zero ('0') bit in the SSPMSK register has the effect of making the corresponding bit in the SSPSR register a "don't care".

This register is reset to all '1's upon any Reset condition and, therefore, has no effect on standard SSP operation until written with a mask value.

This register must be initiated prior to setting SSPM<3:0> bits to select the $I^2C$ Slave mode (7-bit or 10-bit address).

This register can only be accessed when the appropriate mode is selected by bits (SSPM<3:0> of SSPCON).

The SSP Mask register is active during:

- 7-bit Address mode: address compare of A<7:1>.
- 10-bit Address mode: address compare of A<7:0> only. The SSP mask has no effect during the reception of the first (high) byte of the address.

**REGISTER 13-4: SSPMSK: SSP MASK REGISTER(1)**

| R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 |
|---|---|---|---|---|---|---|---|
| MSK7 | MSK6 | MSK5 | MSK4 | MSK3 | MSK2 | MSK1 | MSK0(2) |
| bit 7 | | | | | | | bit 0 |

| **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 7-1 **MSK<7:1>:** Mask bits
1 = The received address bit n is compared to SSPADD<n> to detect $I^2C$ address match
0 = The received address bit n is not used to detect $I^2C$ address match

bit 0 **MSK<0>:** Mask bit for $I^2C$ Slave mode, 10-bit Address(2)
$I^2C$ Slave mode, 10-bit Address (SSPM<3:0> = 0111):
1 = The received address bit 0 is compared to SSPADD<0> to detect $I^2C$ address match
0 = The received address bit 0 is not used to detect $I^2C$ address match

**Note 1:** When SSPCON bits SSPM<3:0> = 1001, any reads or writes to the SSPADD SFR address are accessed through the SSPMSK register.

**2:** In all other SSP modes, this bit has no effect.


© 2006-2012 Microchip Technology Inc.

## 14.0 SPECIAL FEATURES OF THE CPU

The PIC16F882/883/884/886/887 devices have a host of features intended to maximize system reliability, minimize cost through elimination of external components, provide power-saving features and offer code protection.

These features are:

- Reset
  - Power-on Reset (POR)
  - Power-up Timer (PWRT)
  - Oscillator Start-up Timer (OST)
  - Brown-out Reset (BOR)
- Interrupts
- Watchdog Timer (WDT)
- Oscillator selection
- Sleep
- Code protection
- ID Locations
- In-Circuit Serial Programming™
- Low-voltage In-Circuit Serial Programming™

The PIC16F882/883/884/886/887 devices have two timers that offer necessary delays on power-up. One is the Oscillator Start-up Timer (OST), intended to keep the chip in Reset until the crystal oscillator is stable. The other is the Power-up Timer (PWRT), which provides a fixed delay of 64 ms (nominal) on power-up only, designed to keep the part in Reset while the power supply stabilizes. There is also circuitry to reset the device if a brown-out occurs, which can use the Power-up Timer to provide at least a 64 ms Reset. With these three functions-on-chip, most applications need no external Reset circuitry.

The Sleep mode is designed to offer a very low-current Power-Down mode. The user can wake-up from Sleep through:

- External Reset
- Watchdog Timer Wake-up
- An interrupt

Several oscillator options are also made available to allow the part to fit the application. The INTOSC option saves system cost while the LP crystal option saves power. A set of Configuration bits are used to select various options (see Register 14-3).

## 14.1 Configuration Bits

The Configuration bits can be programmed (read as '0'), or left unprogrammed (read as '1') to select various device configurations as shown in Register 14-1. These bits are mapped in program memory location 2007h and 2008h, respectively.

**Note:** Address 2007h and 2008h are beyond the user program memory space. It belongs to the special configuration memory space (2000h-3FFFh), which can be accessed only during programming. See "*PIC16F88X Memory Programming Specification*" (DS41287) for more information.

## REGISTER DEFINITIONS: CONFIGURATION WORDS

### REGISTER 14-1: CONFIG1: CONFIGURATION WORD REGISTER 1

| | | | | |
|---|---|---|---|---|
| $\overline{\text{DEBUG}}$ | LVP | FCMEN | IESO | BOREN<1:0> |
| bit 13 | | | | bit 8 |

| | | | | | |
|---|---|---|---|---|---|
| $\overline{\text{CPD}}$ | $\overline{\text{CP}}$ | MCLRE | $\overline{\text{PWRTE}}$ | WDTE | FOSC<2:0> |
| bit 7 | | | | | bit 0 |

bit 13 **$\overline{\text{DEBUG}}$:** In-Circuit Debugger Mode bit
1 = In-Circuit Debugger disabled, RB6/ICSPCLK and RB7/ICSPDAT are general purpose I/O pins
0 = In-Circuit Debugger enabled, RB6/ICSPCLK and RB7/ICSPDAT are dedicated to the debugger

bit 12 **LVP:** Low Voltage Programming Enable bit
1 = RB3/PGM pin has PGM function, low voltage programming enabled
0 = RB3 pin is digital I/O, HV on $\overline{\text{MCLR}}$ must be used for programming

bit 11 **FCMEN:** Fail-Safe Clock Monitor Enabled bit
1 = Fail-Safe Clock Monitor is enabled
0 = Fail-Safe Clock Monitor is disabled

bit 10 **IESO:** Internal External Switchover bit
1 = Internal/External Switchover mode is enabled
0 = Internal/External Switchover mode is disabled

bit 9-8 **BOREN<1:0>:** Brown-out Reset Selection bits(1)
11 = BOR enabled
10 = BOR enabled during operation and disabled in Sleep
01 = BOR controlled by SBOREN bit of the PCON register
00 = BOR disabled

bit 7 **$\overline{\text{CPD}}$:** Data Code Protection bit(2)
1 = Data memory code protection is disabled
0 = Data memory code protection is enabled

bit 6 **$\overline{\text{CP}}$:** Code Protection bit(3)
1 = Program memory code protection is disabled
0 = Program memory code protection is enabled

bit 5 **MCLRE:** RE3/$\overline{\text{MCLR}}$ pin function select bit(4)
1 = RE3/$\overline{\text{MCLR}}$ pin function is $\overline{\text{MCLR}}$
0 = RE3/$\overline{\text{MCLR}}$ pin function is digital input, $\overline{\text{MCLR}}$ internally tied to $V_{DD}$

bit 4 **$\overline{\text{PWRTE}}$:** Power-up Timer Enable bit
1 = PWRT disabled
0 = PWRT enabled

bit 3 **WDTE:** Watchdog Timer Enable bit
1 = WDT enabled
0 = WDT disabled and can be enabled by SWDTEN bit of the WDTCON register

bit 2-0 **FOSC<2:0>:** Oscillator Selection bits
111 = RC oscillator: CLKOUT function on RA6/OSC2/CLKOUT pin, RC on RA7/OSC1/CLKIN
110 = RCIO oscillator: I/O function on RA6/OSC2/CLKOUT pin, RC on RA7/OSC1/CLKIN
101 = INTOSC oscillator: CLKOUT function on RA6/OSC2/CLKOUT pin, I/O function on RA7/OSC1/CLKIN
100 = INTOSCIO oscillator: I/O function on RA6/OSC2/CLKOUT pin, I/O function on RA7/OSC1/CLKIN
011 = EC: I/O function on RA6/OSC2/CLKOUT pin, CLKIN on RA7/OSC1/CLKIN
010 = HS oscillator: High-speed crystal/resonator on RA6/OSC2/CLKOUT and RA7/OSC1/CLKIN
001 = XT oscillator: Crystal/resonator on RA6/OSC2/CLKOUT and RA7/OSC1/CLKIN
000 = LP oscillator: Low-power crystal on RA6/OSC2/CLKOUT and RA7/OSC1/CLKIN

**Note 1:** Enabling Brown-out Reset does not automatically enable Power-up Timer.
**2:** The entire data EEPROM will be erased when the code protection is turned off.
**3:** The entire program memory will be erased when the code protection is turned off.
**4:** When $\overline{\text{MCLR}}$ is asserted in INTOSC or RC mode, the internal clock oscillator is disabled.


© 2006-2012 Microchip Technology Inc.

**REGISTER 14-2: CONFIG2: CONFIGURATION WORD REGISTER 2**

| | | | | |
|---|---|---|---|---|
| — | — | — | WRT<1:0> | BOR4V |
| bit 13 | | | | bit 8 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| — | — | — | — | — | — | — | — |
| bit 7 | | | | | | | bit 0 |

bit 13-11 **Unimplemented**: Read as '1'

bit 10-9 **WRT<1:0>:** Flash Program Memory Self Write Enable bits

PIC16F883/PIC16F884

00 = 0000h to 07FFh write protected, 0800h to 0FFFh may be modified by EECON control
01 = 0000h to 03FFh write protected, 0400h to 0FFFh may be modified by EECON control
10 = 0000h to 00FFh write protected, 0100h to 0FFFh may be modified by EECON control
11 = Write protection off

PIC16F886/PIC16F887

00 = 0000h to 0FFFh write protected, 1000h to 1FFFh may be modified by EECON control
01 = 0000h to 07FFh write protected, 0800h to 1FFFh may be modified by EECON control
10 = 0000h to 00FFh write protected, 0100h to 1FFFh may be modified by EECON control
11 = Write protection off

PIC16F882

00 = 0000h to 03FFh write protected, 0400h to 07FFh may be modified by EECON control
01 = 0000h to 00FFh write protected, 0100h to 07FFh may be modified by EECON control
11 = Write protection off

bit 8 **BOR4V**: Brown-out Reset Selection bit

0 = Brown-out Reset set to 2.1V
1 = Brown-out Reset set to 4.0V

bit 7-0 **Unimplemented**: Read as '1'

## 14.2 Reset

The PIC16F882/883/884/886/887 devices differentiate between various kinds of Reset:

a) Power-on Reset (POR)
b) WDT Reset during normal operation
c) WDT Reset during Sleep
d) $\overline{\text{MCLR}}$ Reset during normal operation
e) $\overline{\text{MCLR}}$ Reset during Sleep
f) Brown-out Reset (BOR)

Some registers are not affected in any Reset condition; their status is unknown on POR and unchanged in any other Reset. Most other registers are reset to a "Reset state" on:

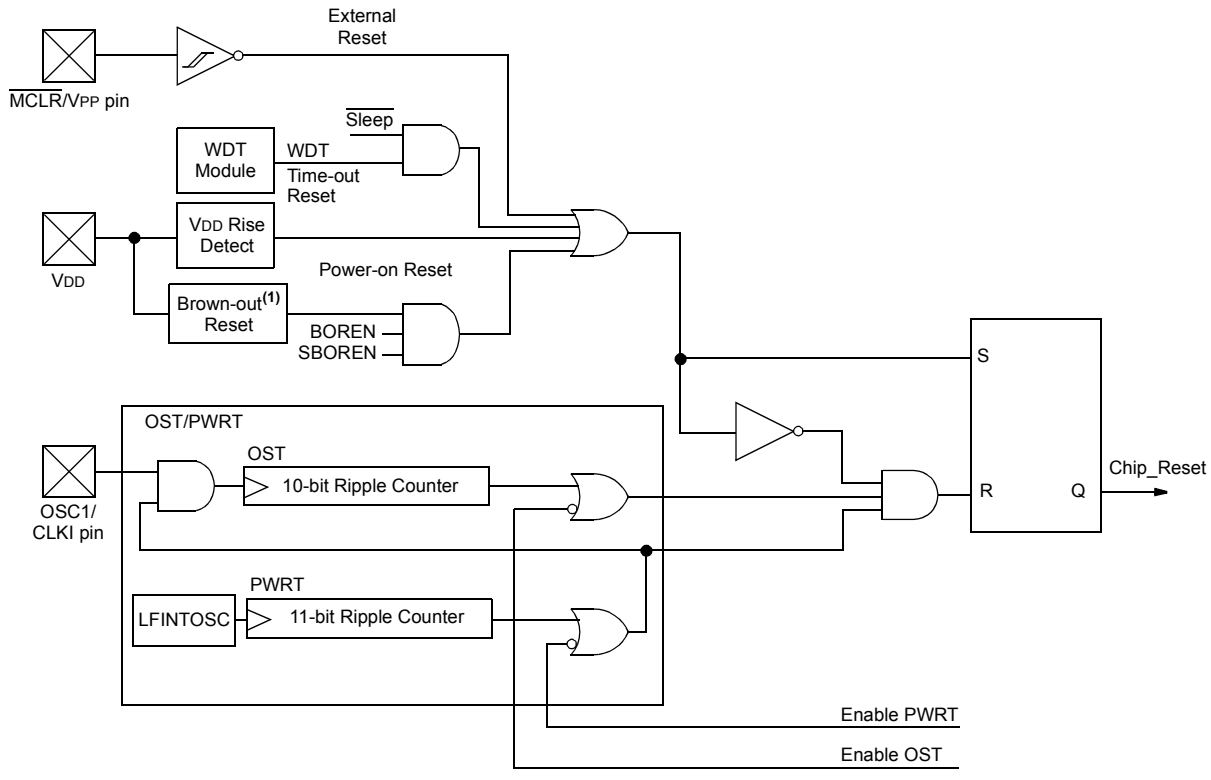
- Power-on Reset
- $\overline{\text{MCLR}}$ Reset
- $\overline{\text{MCLR}}$ Reset during Sleep
- WDT Reset
- Brown-out Reset (BOR)

They are not affected by a WDT Wake-up since this is viewed as the resumption of normal operation. $\overline{\text{TO}}$ and $\overline{\text{PD}}$ bits are set or cleared differently in different Reset situations, as indicated in Table 14-2. These bits are used in software to determine the nature of the Reset. See Table 14-5 for a full description of Reset states of all registers.

A simplified block diagram of the On-Chip Reset Circuit is shown in Figure 14-1.

The $\overline{\text{MCLR}}$ Reset path has a noise filter to detect and ignore small pulses. See **Section 17.0 "Electrical Specifications"** for pulse-width specifications.

**FIGURE 14-1: SIMPLIFIED BLOCK DIAGRAM OF ON-CHIP RESET CIRCUIT**




**Note 1:** Refer to the Configuration Word Register 1 (Register 14-1).


© 2006-2012 Microchip Technology Inc.

### 14.2.1 POWER-ON RESET (POR)

The on-chip POR circuit holds the chip in Reset until VDD has reached a high enough level for proper operation. A maximum rise time for VDD is required. See **Section 17.0 "Electrical Specifications"** for details. If the BOR is enabled, the maximum rise time specification does not apply. The BOR circuitry will keep the device in Reset until VDD reaches VBOR (see **Section 14.2.4 "Brown-out Reset (BOR)"**).

| **Note:** | The POR circuit does not produce an internal Reset when VDD declines. To re-enable the POR, VDD must reach Vss for a minimum of 100 μs. |
|---|---|

When the device starts normal operation (exits the Reset condition), device operating parameters (i.e., voltage, frequency, temperature, etc.) must be met to ensure operation. If these conditions are not met, the device must be held in Reset until the operating conditions are met.

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

### 14.2.2 $\overline{\text{MCLR}}$

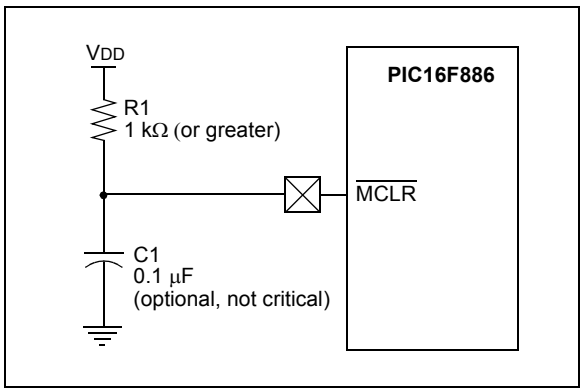
PIC16F882/883/884/886/887 have a noise filter in the $\overline{\text{MCLR}}$ Reset path. The filter will detect and ignore small pulses.

It should be noted that a WDT Reset does not drive $\overline{\text{MCLR}}$ pin low.

The behavior of the ESD protection on the $\overline{\text{MCLR}}$ pin has been altered from early devices of this family. Voltages applied to the pin that exceed its specification can result in both $\overline{\text{MCLR}}$ Resets and excessive current beyond the device specification during the ESD event. For this reason, Microchip recommends that the $\overline{\text{MCLR}}$ pin no longer be tied directly to VDD. The use of an RC network, as shown in Figure 14-2, is suggested.

An internal $\overline{\text{MCLR}}$ option is enabled by clearing the MCLRE bit in the Configuration Word Register 1. When $\overline{\text{MCLRE}}$ = 0, the Reset signal to the chip is generated internally. When the $\overline{\text{MCLRE}}$ = 1, the RA3/$\overline{\text{MCLR}}$ pin becomes an external Reset input. In this mode, the RA3/$\overline{\text{MCLR}}$ pin has a weak pull-up to VDD.

**FIGURE 14-2: RECOMMENDED $\overline{\text{MCLR}}$ CIRCUIT**




### 14.2.3 POWER-UP TIMER (PWRT)

The Power-up Timer provides a fixed 64 ms (nominal) time-out on power-up only, from POR or Brown-out Reset. The Power-up Timer operates from the 31 kHz LFINTOSC oscillator. For more information, see **Section 4.5 "Internal Clock Modes"**. The chip is kept in Reset as long as PWRT is active. The PWRT delay allows the VDD to rise to an acceptable level. A Configuration bit, $\overline{\text{PWRTE}}$, can disable (if set) or enable (if cleared or programmed) the Power-up Timer. The Power-up Timer should be enabled when Brown-out Reset is enabled, although it is not required.

The Power-up Timer delay will vary from chip-to-chip and vary due to:

- VDD variation
- Temperature variation
- Process variation

See DC parameters for details (**Section 17.0 "Electrical Specifications"**).

### 14.2.4 BROWN-OUT RESET (BOR)

The BOREN0 and BOREN1 bits in the Configuration Word Register 1 select one of four BOR modes. Two modes have been added to allow software or hardware control of the BOR enable. When BOREN<1:0> = 01, the SBOREN bit (PCON<4>) enables/disables the BOR allowing it to be controlled in software. By selecting BOREN<1:0>, the BOR is automatically disabled in Sleep to conserve power and enabled on wake-up. In this mode, the SBOREN bit is disabled. See Register 14-3 for the Configuration Word definition.

The BOR4V bit in the Configuration Word Register 2 selects one of two Brown-out Reset voltages. When BOR4B = 1, VBOR is set to 4V. When BOR4V = 0, VBOR is set to 2.1V.

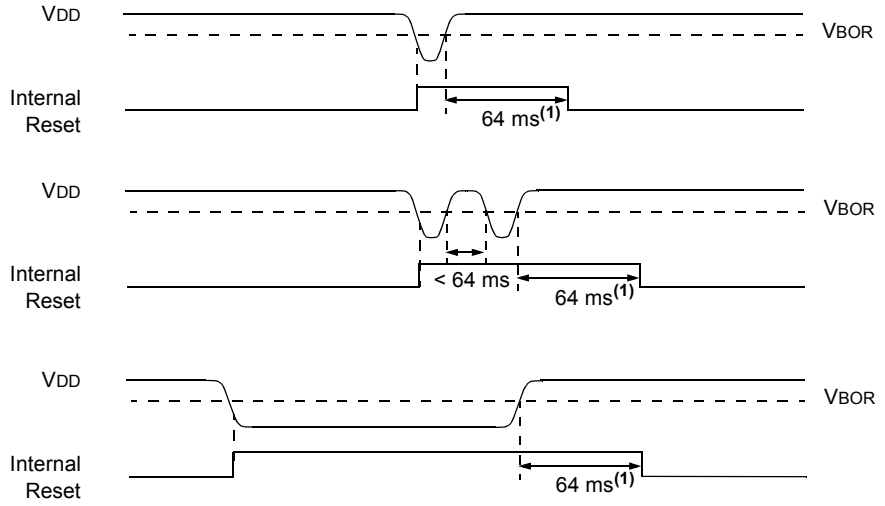
If VDD falls below VBOR for greater than parameter (TBOR) (see **Section 17.0 "Electrical Specifications"**), the Brown-out situation will reset the device. This will occur regardless of VDD slew rate. A Reset is not insured to occur if VDD falls below VBOR for less than parameter (TBOR).

On any Reset (Power-on, Brown-out Reset, Watchdog Timer, etc.), the chip will remain in Reset until VDD rises above VBOR (see Figure 14-3). The Power-up Timer will now be invoked, if enabled and will keep the chip in Reset an additional 64 ms.

| **Note:** | The Power-up Timer is enabled by the $\overline{\text{PWRTE}}$ bit in the Configuration Word Register 1. |
|---|---|

If VDD drops below VBOR while the Power-up Timer is running, the chip will go back into a Brown-out Reset and the Power-up Timer will be re-initialized. Once VDD rises above VBOR, the Power-up Timer will execute a 64 ms Reset.

**FIGURE 14-3: BROWN-OUT SITUATIONS**




**Note 1:** 64 ms delay only if $\overline{\text{PWRTE}}$ bit is programmed to '0'.


© 2006-2012 Microchip Technology Inc.

### 14.2.5 TIME-OUT SEQUENCE

On power-up, the time-out sequence is as follows: first, PWRT time-out is invoked after POR has expired, then OST is activated after the PWRT time-out has expired. The total time-out will vary based on oscillator configuration and $\overline{\text{PWRTE}}$ bit status. For example, in EC mode with $\overline{\text{PWRTE}}$ bit erased (PWRT disabled), there will be no time-out at all. Figures 14-4, 14-5 and 14-6 depict time-out sequences. The device can execute code from the INTOSC while OST is active by enabling Two-Speed Start-up or Fail-Safe Monitor (see **Section 4.7.2 "Two-Speed Start-up Sequence"** and **Section 4.8 "Fail-Safe Clock Monitor"**).

Since the time-outs occur from the POR pulse, if $\overline{\text{MCLR}}$ is kept low long enough, the time-outs will expire. Then, bringing $\overline{\text{MCLR}}$ high will begin execution immediately (see Figure 14-5). This is useful for testing purposes or to synchronize more than one PIC16F882/883/884/886/887 device operating in parallel.

Table 14-5 shows the Reset conditions for some special registers, while Table 14-4 shows the Reset conditions for all the registers.

### 14.2.6 POWER CONTROL (PCON) REGISTER

The Power Control register PCON (address 8Eh) has two Status bits to indicate what type of Reset that last occurred.

Bit 0 is $\overline{\text{BOR}}$ (Brown-out Reset). $\overline{\text{BOR}}$ is unknown on Power-on Reset. It must then be set by the user and checked on subsequent Resets to see if $\overline{\text{BOR}}$ = 0, indicating that a Brown-out has occurred. The $\overline{\text{BOR}}$ 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 1).

Bit 1 is $\overline{\text{POR}}$ (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 $\overline{\text{POR}}$ 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 3.2.2 "Ultra Low-Power Wake-up"** and **Section 14.2.4 "Brown-out Reset (BOR)"**.

**TABLE 14-1: TIME-OUT IN VARIOUS SITUATIONS**

| Oscillator Configuration | Power-up | | Brown-out Reset | | Wake-up from Sleep |
|---|---|---|---|---|---|
| | $\overline{\text{PWRTE}}$ = 0 | $\overline{\text{PWRTE}}$ = 1 | $\overline{\text{PWRTE}}$ = 0 | $\overline{\text{PWRTE}}$ = 1 | |
| XT, HS, LP | $T_{PWRT}$ + 1024 • $T_{OSC}$ | 1024 • $T_{OSC}$ | $T_{PWRT}$ + 1024 • $T_{OSC}$ | 1024 • $T_{OSC}$ | 1024 • $T_{OSC}$ |
| LP, T1OSCIN = 1 | $T_{PWRT}$ | — | $T_{PWRT}$ | — | — |
| RC, EC, INTOSC | $T_{PWRT}$ | — | $T_{PWRT}$ | — | — |

**TABLE 14-2: STATUS/PCON BITS AND THEIR SIGNIFICANCE**

| POR | BOR | $\overline{\text{TO}}$ | $\overline{\text{PD}}$ | 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 | $\overline{\text{MCLR}}$ Reset during normal operation |
| u | u | 1 | 0 | $\overline{\text{MCLR}}$ Reset during Sleep |

**Legend:** u = unchanged, x = unknown

**TABLE 14-3: SUMMARY OF REGISTERS ASSOCIATED WITH BROWN-OUT**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|
| PCON | — | — | ULPWUE | SBOREN | — | — | $\overline{\text{POR}}$ | $\overline{\text{BOR}}$ | 38 |
| STATUS | IRP | RP1 | RPO | $\overline{\text{TO}}$ | $\overline{\text{PD}}$ | Z | DC | C | 31 |

**Legend:** u = unchanged, x = unknown, – = unimplemented bit, reads as '0', q = value depends on condition. Shaded cells are not used by BOR.

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

© 2006-2012 Microchip Technology Inc.

**FIGURE 14-4: TIME-OUT SEQUENCE ON POWER-UP (DELAYED $\overline{\text{MCLR}}$): CASE 1**

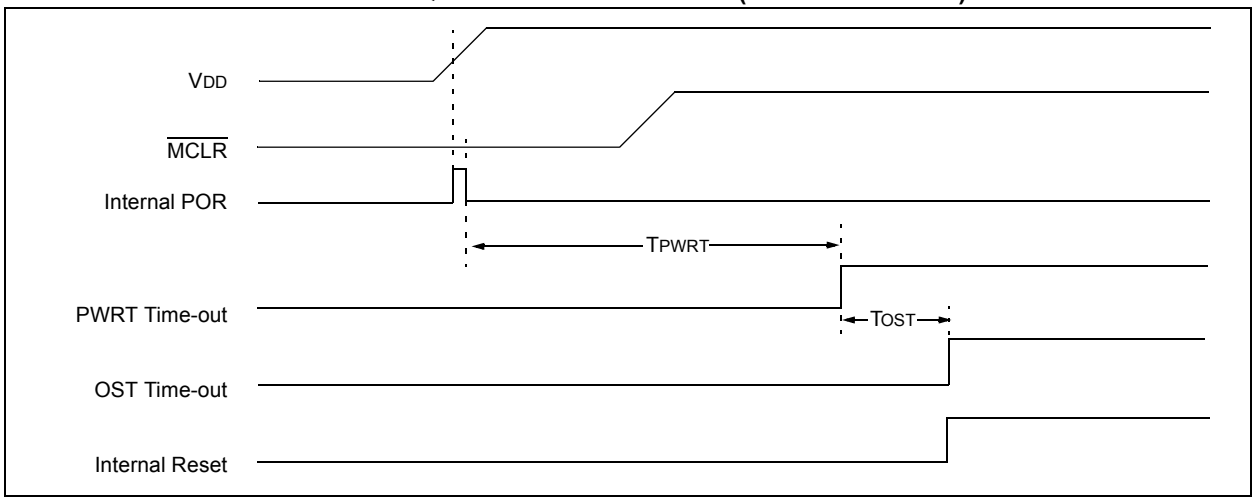



**FIGURE 14-5: TIME-OUT SEQUENCE ON POWER-UP (DELAYED $\overline{\text{MCLR}}$): CASE 2**

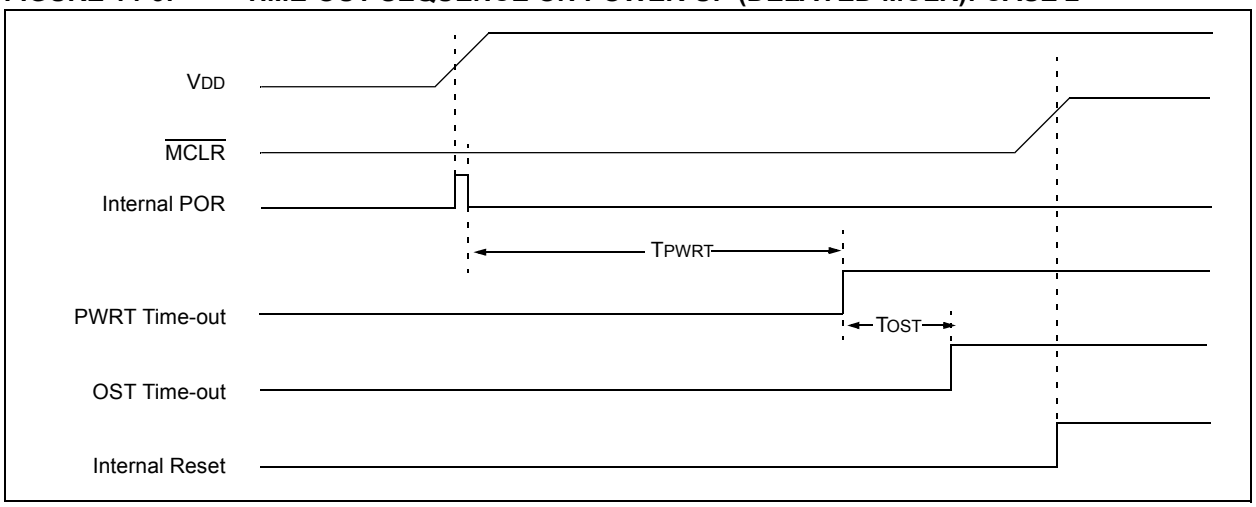
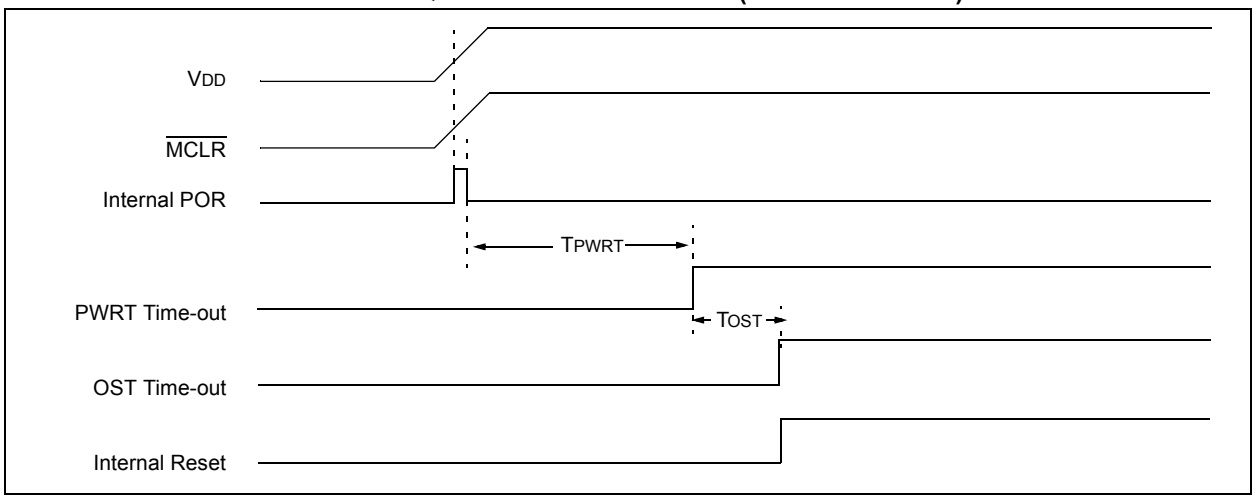



**FIGURE 14-6: TIME-OUT SEQUENCE ON POWER-UP ($\overline{\text{MCLR}}$ WITH VDD)**





© 2006-2012 Microchip Technology Inc.

**TABLE 14-4: INITIALIZATION CONDITION FOR REGISTER**

| Register | Address | Power-on Reset | $\overline{\text{MCLR}}$ Reset<br>WDT Reset<br>Brown-out Reset(1) | Wake-up from Sleep through Interrupt<br>Wake-up from Sleep through WDT Time-out |
|---|---|---|---|---|
| W | — | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| INDF | 00h/80h/100h/180h | xxxx xxxx | xxxx xxxx | uuuu uuuu |
| TMR0 | 01h/101h | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| PCL | 02h/82h/102h/182h | 0000 0000 | 0000 0000 | PC + 1(3) |
| STATUS | 03h/83h/103h/183h | 0001 1xxx | 000q quuu(4) | uuuq quuu(4) |
| FSR | 04h/84h/104h/184h | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| PORTA | 05h | xxxx xxxx | 0000 0000 | uuuu uuuu |
| PORTB | 06h/106h | xxxx xxxx | 0000 0000 | uuuu uuuu |
| PORTC | 07h | xxxx xxxx | 0000 0000 | uuuu uuuu |
| PORTD | 08h | xxxx xxxx | 0000 0000 | uuuu uuuu |
| PORTE | 09h | ---- xxxx | ---- 0000 | ---- uuuu |
| PCLATH | 0Ah/8Ah/10Ah/18Ah | ---0 0000 | ---0 0000 | ---u uuuu |
| INTCON | 0Bh/8Bh/10Bh/18Bh | 0000 000x | 0000 000u | uuuu uuuu(2) |
| PIR1 | 0Ch | 0000 0000 | 0000 0000 | uuuu uuuu(2) |
| PIR2 | 0Dh | 0000 0000 | 0000 0000 | uuuu uuuu(2) |
| TMR1L | 0Eh | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| TMR1H | 0Fh | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| T1CON | 10h | 0000 0000 | uuuu uuuu | -uuu uuuu |
| TMR2 | 11h | 0000 0000 | 0000 0000 | uuuu uuuu |
| T2CON | 12h | -000 0000 | -000 0000 | -uuu uuuu |
| SSPBUF | 13h | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| SSPCON | 14h | 0000 0000 | 0000 0000 | uuuu uuuu |
| CCPR1L | 15h | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| CCPR1H | 16h | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| CCP1CON | 17h | 0000 0000 | 0000 0000 | uuuu uuuu |
| RCSTA | 18h | 0000 000x | 0000 0000 | uuuu uuuu |
| TXREG | 19h | 0000 0000 | 0000 0000 | uuuu uuuu |
| RCREG | 1Ah | 0000 0000 | 0000 0000 | uuuu uuuu |
| CCPR2L | 1Bh | xxxx xxxx | uuuu uuuu | uuuu uuuu |

**Legend:** u = unchanged, x = unknown, – = unimplemented bit, reads as '0', q = value depends on condition.

**Note 1:** If VDD goes too low, Power-on Reset will be activated and registers will be affected differently.

**2:** One or more bits in INTCON and/or PIR1 will be affected (to cause wake-up).

**3:** When the wake-up is due to an interrupt and the GIE bit is set, the PC is loaded with the interrupt vector (0004h).

**4:** See Table 14-5 for Reset value for specific condition.

**5:** If Reset was due to brown-out, then bit 0 = 0. All other Resets will cause bit 0 = u.

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

**TABLE 14-4: INITIALIZATION CONDITION FOR REGISTER (CONTINUED)**

| Register | Address | Power-on Reset | $\overline{\text{MCLR}}$ Reset<br>WDT Reset (Continued)<br>Brown-out Reset(1) | Wake-up from Sleep through Interrupt<br>Wake-up from Sleep through WDT Time-out (Continued) |
|---|---|---|---|---|
| CCPR2H | 1Ch | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| CCP2CON | 1Dh | --00 0000 | --00 0000 | --uu uuuu |
| ADRESH | 1Eh | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| ADCON0 | 1Fh | 00-0 0000 | 00-0 0000 | uu-u uuuu |
| OPTION_REG | 81h/181h | 1111 1111 | 1111 1111 | uuuu uuuu |
| TRISA | 85h | 1111 1111 | 1111 1111 | uuuu uuuu |
| TRISB | 86h/186h | 1111 1111 | 1111 1111 | uuuu uuuu |
| TRISC | 87h | 1111 1111 | 1111 1111 | uuuu uuuu |
| TRISD | 88h | 1111 1111 | 1111 1111 | uuuu uuuu |
| TRISE | 89h | ---- 1111 | ---- 1111 | ---- uuuu |
| PIE1 | 8Ch | 0000 0000 | 0000 0000 | uuuu uuuu |
| PIE2 | 8Dh | 0000 0000 | 0000 0000 | uuuu uuuu |
| PCON | 8Eh | --01 --0x | --0u --uu(1, 5) | --uu --uu |
| OSCCON | 8Fh | -110 q000 | -110 q000 | -uuu uuuu |
| OSCTUNE | 90h | ---0 0000 | ---u uuuu | ---u uuuu |
| SSPCON2 | 91h | 0000 0000 | 0000 0000 | uuuu uuuu |
| PR2 | 92h | 1111 1111 | 1111 1111 | 1111 1111 |
| SSPADD(6) | 93h | 0000 0000 | 0000 0000 | uuuu uuuu |
| SSPMSK(6) | 93h | 1111 1111 | 1111 1111 | 1111 1111 |
| SSPSTAT | 94h | 0000 0000 | 0000 0000 | uuuu uuuu |
| WPUB | 95h | 1111 1111 | 1111 1111 | uuuu uuuu |
| IOCB | 96h | 0000 0000 | 0000 0000 | uuuu uuuu |
| VRCON | 97h | 0000 0000 | 0000 0000 | uuuu uuuu |
| TXSTA | 98h | 0000 -010 | 0000 -010 | uuuu -uuu |
| SPBRG | 99h | 0000 0000 | 0000 0000 | uuuu uuuu |
| SPBRGH | 9Ah | 0000 0000 | 0000 0000 | uuuu uuuu |
| PWM1CON | 9Bh | 0000 0000 | 0000 0000 | uuuu uuuu |
| ECCPAS | 9Ch | 0000 0000 | 0000 0000 | uuuu uuuu |
| PSTRCON | 9Dh | ---0 0001 | ---0 0001 | ---u uuuu |
| ADRESL | 9Eh | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| ADCON1 | 9Fh | 0-00 ---- | 0-00 ---- | u-uu ---- |
| WDTCON | 105h | ---0 1000 | ---0 1000 | ---u uuuu |
| CM1CON0 | 107h | 0000 0-00 | 0000 0-00 | uuuu u-uu |
| CM2CON0 | 108h | 0000 0-00 | 0000 0-00 | uuuu u-uu |

**Legend:** u = unchanged, x = unknown, – = unimplemented bit, reads as '0', q = value depends on condition.

**Note 1:** If VDD goes too low, Power-on Reset will be activated and registers will be affected differently.

**2:** One or more bits in INTCON and/or PIR1 will be affected (to cause wake-up).

**3:** When the wake-up is due to an interrupt and the GIE bit is set, the PC is loaded with the interrupt vector (0004h).

**4:** See Table 14-5 for Reset value for specific condition.

**5:** If Reset was due to brown-out, then bit 0 = 0. All other Resets will cause bit 0 = u.

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


© 2006-2012 Microchip Technology Inc.

**TABLE 14-4: INITIALIZATION CONDITION FOR REGISTER (CONTINUED)**

| Register | Address | Power-on Reset | $\overline{\text{MCLR}}$ Reset<br>WDT Reset (Continued)<br>Brown-out Reset(1) | Wake-up from Sleep through Interrupt<br>Wake-up from Sleep through WDT Time-out (Continued) |
|---|---|---|---|---|
| CM2CON1 | 109h | 0000 0--0 | 0000 0--0 | uuuu u--u |
| EEDAT | 10Ch | 0000 0000 | 0000 0000 | uuuu uuuu |
| EEADR | 10Dh | 0000 0000 | 0000 0000 | uuuu uuuu |
| EEDATH | 10Eh | --00 0000 | --00 0000 | --uu uuuu |
| EEADRH | 10Fh | ---0 0000 | ---0 0000 | ---u uuuu |
| SRCON | 185h | 0000 00-0 | 0000 00-0 | uuuu uu-u |
| BAUDCTL | 187h | 01-0 0-00 | 01-0 0-00 | uu-u u-uu |
| ANSEL | 188h | 1111 1111 | 1111 1111 | uuuu uuuu |
| ANSELH | 189h | 1111 1111 | 1111 1111 | uuuu uuuu |
| EECON1 | 18Ch | ---- x000 | ---- q000 | ---- uuuu |
| EECON2 | 18Dh | ---- ---- | ---- ---- | ---- ---- |

**Legend:** u = unchanged, x = unknown, – = unimplemented bit, reads as '0', q = value depends on condition.

**Note 1:** If VDD goes too low, Power-on Reset will be activated and registers will be affected differently.

**2:** One or more bits in INTCON and/or PIR1 will be affected (to cause wake-up).

**3:** When the wake-up is due to an interrupt and the GIE bit is set, the PC is loaded with the interrupt vector (0004h).

**4:** See Table 14-5 for Reset value for specific condition.

**5:** If Reset was due to brown-out, then bit 0 = 0. All other Resets will cause bit 0 = u.

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

**TABLE 14-5: INITIALIZATION CONDITION FOR SPECIAL REGISTERS**

| Condition | Program Counter | Status Register | PCON Register |
|---|---|---|---|
| Power-on Reset | 000h | 0001 1xxx | --01 --0x |
| $\overline{\text{MCLR}}$ Reset during normal operation | 000h | 000u uuuu | --0u --uu |
| $\overline{\text{MCLR}}$ Reset during Sleep | 000h | 0001 0uuu | --0u --uu |
| WDT Reset | 000h | 0000 uuuu | --0u --uu |
| WDT Wake-up | PC + 1 | uuu0 0uuu | --uu --uu |
| Brown-out Reset | 000h | 0001 1uuu | --01 --u0 |
| Interrupt Wake-up from Sleep | PC + 1(1) | uuu1 0uuu | --uu --uu |

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

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

## 14.3 Interrupts

The PIC16F882/883/884/886/887 devices have multiple interrupt sources:

- External Interrupt RB0/INT
- Timer0 Overflow Interrupt
- PORTB Change Interrupts
- 2 Comparator Interrupts
- A/D Interrupt
- Timer1 Overflow Interrupt
- Timer2 Match Interrupt
- EEPROM Data Write Interrupt
- Fail-Safe Clock Monitor Interrupt
- Enhanced CCP Interrupt
- EUSART Receive and Transmit Interrupts
- Ultra Low-Power Wake-up Interrupt
- MSSP Interrupt

The Interrupt Control register (INTCON) and Peripheral Interrupt Request Register 1 (PIR1) record individual interrupt requests in flag bits. The INTCON register also has individual and global interrupt enable bits.

A Global Interrupt Enable bit, GIE (INTCON<7>), enables (if set) all unmasked interrupts, or disables (if cleared) all interrupts. Individual interrupts can be disabled through their corresponding enable bits in the INTCON, PIE1 and PIE2 registers, respectively. GIE is cleared on Reset.

The Return from Interrupt instruction, `RETFIE`, exits the interrupt routine, as well as sets the GIE bit, which re-enables unmasked interrupts.

The following interrupt flags are contained in the INTCON register:

- INT Pin Interrupt
- PORTB Change Interrupts
- Timer0 Overflow Interrupt

The peripheral interrupt flags are contained in the PIR1 and PIR2 registers. The corresponding interrupt enable bits are contained in PIE1 and PIE2 registers.

The following interrupt flags are contained in the PIR1 register:

- A/D Interrupt
- EUSART Receive and Transmit Interrupts
- Timer1 Overflow Interrupt
- Synchronous Serial Port (SSP) Interrupt
- Enhanced CCP1 Interrupt
- Timer1 Overflow Interrupt
- Timer2 Match Interrupt

The following interrupt flags are contained in the PIR2 register:

- Fail-Safe Clock Monitor Interrupt
- 2 Comparator Interrupts
- EEPROM Data Write Interrupt
- Ultra Low-Power Wake-up Interrupt
- CCP2 Interrupt

When an interrupt is serviced:

- The GIE is cleared to disable any further interrupt.
- The return address is pushed onto the stack.
- The PC is loaded with 0004h.

For external interrupt events, such as the INT pin, PORTB change interrupts, the interrupt latency will be three or four instruction cycles. The exact latency depends upon when the interrupt event occurs (see Figure 14-8). 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, A/D, data EEPROM, EUSART, MSSP or Enhanced CCP modules, refer to the respective peripheral section.

### 14.3.1 RB0/INT INTERRUPT

External interrupt on RB0/INT pin is edge-triggered; either rising if the INTEDG bit (OPTION_REG<6>) is set, or falling, if the INTEDG bit is clear. When a valid edge appears on the RB0/INT pin, the INTF bit (INTCON<1>) is set. This interrupt can be disabled by clearing the INTE control bit (INTCON<4>). The INTF bit must be cleared in software in the Interrupt Service Routine before re-enabling this interrupt. The RB0/INT interrupt can wake-up the processor from Sleep, if the INTE bit was set prior to going into Sleep. The status of the GIE bit decides whether or not the processor branches to the interrupt vector following wake-up (0004h). See **Section 14.6 "Power-Down Mode (Sleep)"** for details on Sleep and Figure 14-10 for timing of wake-up from Sleep through RB0/INT interrupt.


© 2006-2012 Microchip Technology Inc.

### 14.3.2 TIMER0 INTERRUPT

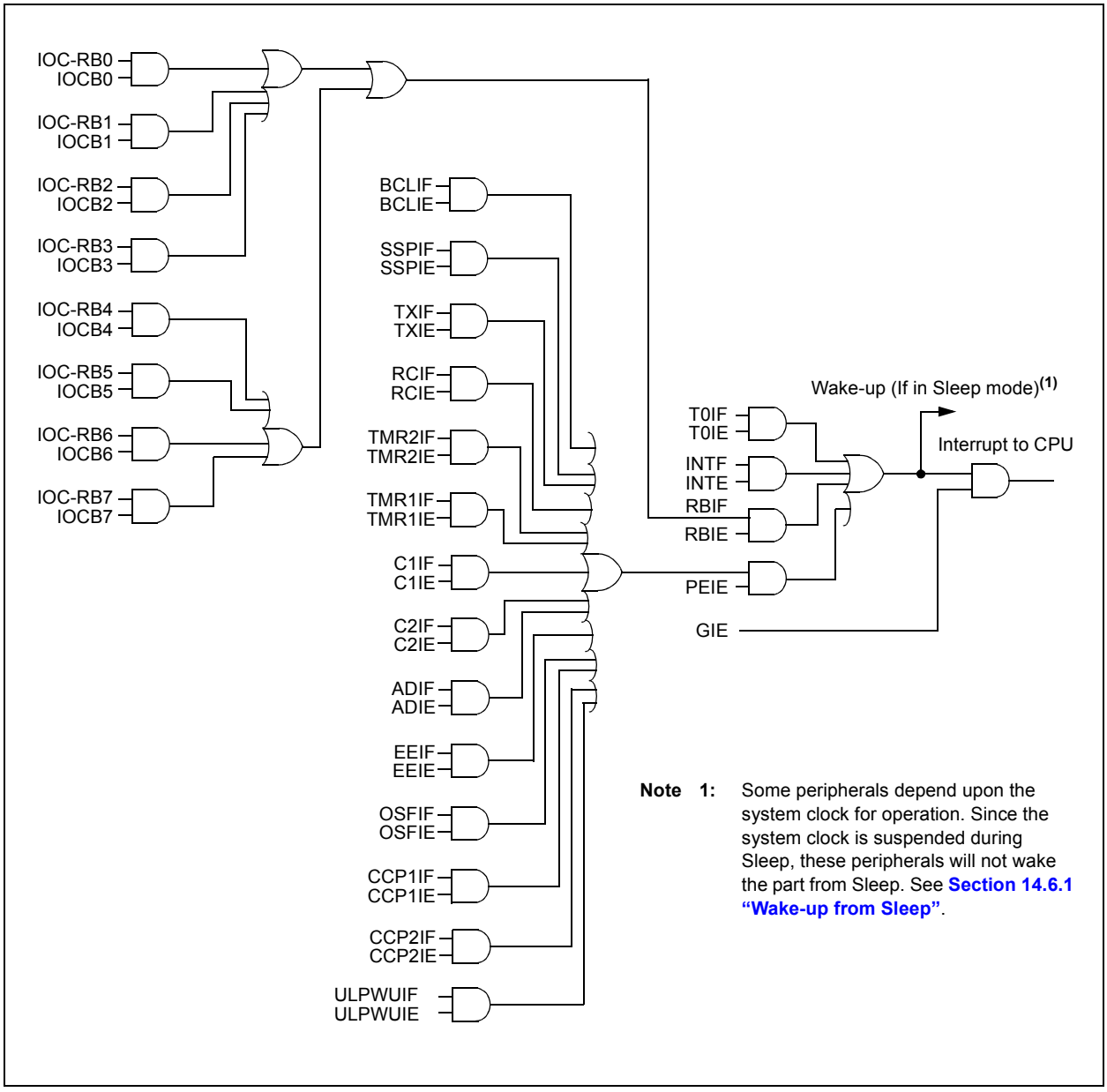
An overflow (FFh → 00h) in the TMR0 register will set the T0IF (INTCON<2>) bit. The interrupt can be enabled/disabled by setting/clearing T0IE (INTCON<5>) bit. See **Section 5.0 "Timer0 Module"** for operation of the Timer0 module.

### 14.3.3 PORTB INTERRUPT

An input change on PORTB change sets the RBIF (INTCON<0>) bit. The interrupt can be enabled/disabled by setting/clearing the RBIE (INTCON<3>) bit. Plus, individual pins can be configured through the IOCB register.

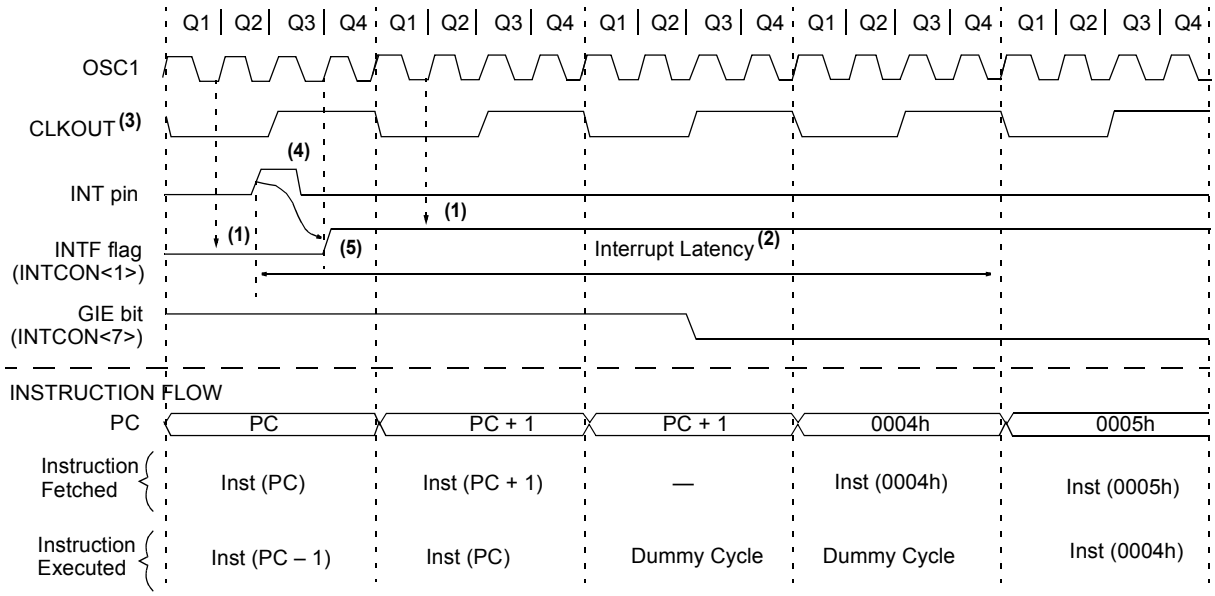
| **Note:** | If a change on the I/O pin should occur when the read operation is being executed (start of the Q2 cycle), then the RBIF interrupt flag may not get set. See **Section 3.4.3 "Interrupt-on-Change"** for more information. |
|---|---|

**FIGURE 14-7: INTERRUPT LOGIC**




**Note 1:** Some peripherals depend upon the system clock for operation. Since the system clock is suspended during Sleep, these peripherals will not wake the part from Sleep. See **Section 14.6.1 "Wake-up from Sleep"**.

**FIGURE 14-8: 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 17.0 "Electrical Specifications"**.

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

**TABLE 14-6: SUMMARY OF INTERRUPT REGISTERS**

| 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 | RBIE | T0IF | INTF | RBIF | 33 |
| PIE1 | — | ADIE | RCIE | TXIE | SSPIE | CCP1IE | TMR2IE | TMR1IE | 34 |
| PIE2 | OSFIE | C2IE | C1IE | EEIE | BCLIE | ULPWUIE | — | CCP2IE | 35 |
| PIR1 | — | ADIF | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | 36 |
| PIR2 | OSFIF | C2IF | C1IF | EEIF | BCLIF | ULPWUIF | — | CCP2IF | 37 |

**Legend:** x = unknown, u = unchanged, – = unimplemented read as '0', q = value depends upon condition. Shaded cells are not used by the interrupt module.


© 2006-2012 Microchip Technology Inc.

## 14.4 Context Saving During Interrupts

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

Since the upper 16 bytes of all GPR banks are common in the PIC16F882/883/884/886/887 (see Figures 2-2 and 2-3), temporary holding registers, W_TEMP and STATUS_TEMP, should be placed in here. These 16 locations do not require banking and therefore, make it easier to context save and restore. The same code shown in Example 14-1 can be used to:

- Store the W register
- Store the STATUS register
- Execute the ISR code
- Restore the Status (and Bank Select Bit register)
- Restore the W register

**Note:** The PIC16F882/883/884/886/887 devices normally do not require saving the PCLATH. However, if computed GOTOs are used in the ISR and the main code, the PCLATH must be saved and restored in the ISR.

**EXAMPLE 14-1: SAVING STATUS AND W REGISTERS IN RAM**

```
MOVWF  W_TEMP           ;Copy W to TEMP register
SWAPF  STATUS,W         ;Swap status to be saved into W
                        ;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
```

## 14.5 Watchdog Timer (WDT)

The WDT has the following features:

- Operates from the LFINTOSC (31 kHz)
- Contains a 16-bit prescaler
- Shares an 8-bit prescaler with Timer0
- Time-out period is from 1 ms to 268 seconds
- Configuration bit and software controlled

WDT is cleared under certain conditions described in Table 14-7.

### 14.5.1 WDT OSCILLATOR

The WDT derives its time base from the 31 kHz LFINTOSC. The LTS bit of the OSCCON register does not reflect that the LFINTOSC is enabled.

The value of WDTCON is '---0 1000' on all Resets. This gives a nominal time base of 17 ms.

| **Note:** | When the Oscillator Start-up Timer (OST) is invoked, the WDT is held in Reset, because the WDT Ripple Counter is used by the OST to perform the oscillator delay count. When the OST count has expired, the WDT will begin counting (if enabled). |
|---|---|

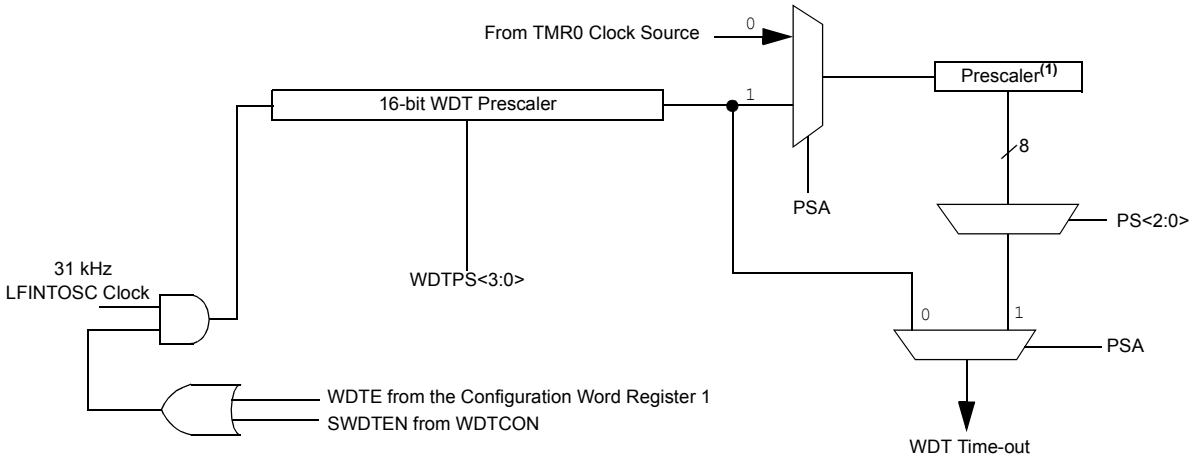
### 14.5.2 WDT CONTROL

The WDTE bit is located in the Configuration Word Register 1. When set, the WDT runs continuously.

When the WDTE bit in the Configuration Word Register 1 is set, the SWDTEN bit of the WDTCON register has no effect. If WDTE is clear, then the SWDTEN bit can be used to enable and disable the WDT. Setting the bit will enable it and clearing the bit will disable it.

The PSA and PS<2:0> bits of the OPTION register have the same function as in previous versions of the PIC16F882/883/884/886/887 family of microcontrollers. See **Section 5.0 "Timer0 Module"** for more information.

**FIGURE 14-9: WATCHDOG TIMER BLOCK DIAGRAM**




**Note 1:** This is the shared Timer0/WDT prescaler. See **Section 5.1.3 "Software Programmable Prescaler"** for more information.

**TABLE 14-7: WDT STATUS**

| Conditions | WDT |
|---|---|
| WDTE = 0 | Cleared |
| CLRWDT Command | |
| Oscillator Fail Detected | |
| Exit Sleep + System Clock = T1OSC, EXTRC, INTOSC, EXTCLK | |
| Exit Sleep + System Clock = XT, HS, LP | Cleared until the end of OST |


© 2006-2012 Microchip Technology Inc.

**REGISTER 14-3: WDTCON: WATCHDOG TIMER CONTROL REGISTER**

| U-0 | U-0 | U-0 | R/W-0 | R/W-1 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | — | — | WDTPS3 | WDTPS2 | WDTPS1 | WDTPS0 | SWDTEN(1) |
| bit 7 | | | | | | | bit 0 |

| 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 4-1 **WDTPS<3:0>:** Watchdog Timer Period Select bits

Bit Value = Prescale Rate

0000 = 1:32
0001 = 1:64
0010 = 1:128
0011 = 1:256
0100 = 1:512 (Reset value)
0101 = 1:1024
0110 = 1:2048
0111 = 1:4096
1000 = 1:8192
1001 = 1:16384
1010 = 1:32768
1011 = 1:65536
1100 = reserved
1101 = reserved
1110 = reserved
1111 = reserved

bit 0 **SWDTEN:** Software Enable or Disable the Watchdog Timer(1)

1 = WDT is turned on
0 = WDT is turned off (Reset value)

**Note 1:** If WDTE Configuration bit = 1, then WDT is always enabled, irrespective of this control bit. If WDTE Configuration bit = 0, then it is possible to turn WDT on/off with this control bit.

**TABLE 14-8: 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 | $\overline{\text{RBPU}}$ | INTEDG | T0CS | T0SE | PSA | PS2 | PS1 | PS0 | 32 |
| WDTCON | — | — | — | WDTPS3 | WDTPS2 | WSTPS1 | WDTPS0 | SWDTEN | 229 |

**Legend:** Shaded cells are not used by the Watchdog Timer.

**TABLE 14-9: SUMMARY OF CONFIGURATION WORD ASSOCIATED WITH WATCHDOG TIMER**

| Name | Bits | Bit -/7 | Bit -/6 | Bit 13/5 | Bit 12/4 | Bit 11/3 | Bit 10/2 | Bit 9/1 | Bit 8/0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|---|
| CONFIG1(1) | 13:8 | — | — | $\overline{\text{DEBUG}}$ | LVP | FCMEN | IESO | BOREN 1 | BOREN0 | 214 |
| | 7:0 | $\overline{\text{CPD}}$ | $\overline{\text{CP}}$ | MCLRE | $\overline{\text{PWRTE}}$ | WDTE | FOSC 2 | FOSC 1 | FOSC 0 | |

**Legend:** – = unimplemented locations read as '0'. Shaded cells are not used by the Watchdog Timer.

**Note 1:** See Configuration Word Register 1 (Register 14-1) for operation of all register bits.

## 14.6 Power-Down Mode (Sleep)

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

If the Watchdog Timer is enabled:

- WDT will be cleared but keeps running.
- $\overline{PD}$ bit in the STATUS register is cleared.
- $\overline{TO}$ bit is set.
- Oscillator driver is turned off.
- I/O ports maintain the status they had before SLEEP was executed (driving high, low or high-impedance).

For lowest current consumption in this mode, all I/O pins should be either at VDD or VSS, with no external circuitry drawing current from the I/O pin and the comparators and CVREF 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. |
|---|---|

### 14.6.1 WAKE-UP FROM SLEEP

The device can wake-up from Sleep through one of the following events:

1. External Reset input on $\overline{MCLR}$ pin.
2. Watchdog Timer Wake-up (if WDT was enabled).
3. Interrupt from RB0/INT pin, PORTB change or a peripheral interrupt.

The first event will cause a device Reset. The two latter events are considered a continuation of program execution. The $\overline{TO}$ and $\overline{PD}$ bits in the STATUS register can be used to determine the cause of device Reset. The $\overline{PD}$ bit, which is set on power-up, is cleared when Sleep is invoked. $\overline{TO}$ bit is cleared if WDT Wake-up occurred.

The following peripheral interrupts can wake the device from Sleep:

1. TMR1 interrupt. Timer1 must be operating as an asynchronous counter.
2. ECCP Capture mode interrupt.
3. A/D conversion (when A/D clock source is FRC).
4. EEPROM write operation completion.
5. Comparator output changes state.
6. Interrupt-on-change.
7. External Interrupt from INT pin.
8. EUSART Break detect, $I^2C$ slave.

Other peripherals cannot generate interrupts since during Sleep, no on-chip clocks are present.

When the SLEEP instruction is being executed, the next instruction (PC + 1) is prefetched. For the device to wake-up through an interrupt event, the corresponding interrupt enable bit must be set (enabled). Wake-up occurs regardless of the state of the GIE bit. If the GIE bit is clear (disabled), the device continues execution at the instruction after the SLEEP instruction. If the GIE bit is set (enabled), the device executes the instruction after the SLEEP instruction, then branches to the interrupt address (0004h). In cases where the execution of the instruction following SLEEP is not desirable, the user should have a NOP after the SLEEP instruction.

| **Note:** | If the global interrupts are disabled (GIE is cleared), but any interrupt source has both its interrupt enable bit and the corresponding interrupt flag bits set, the device will immediately wake-up from Sleep. The SLEEP instruction is completely executed. |
|---|---|

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

### 14.6.2 WAKE-UP USING INTERRUPTS

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

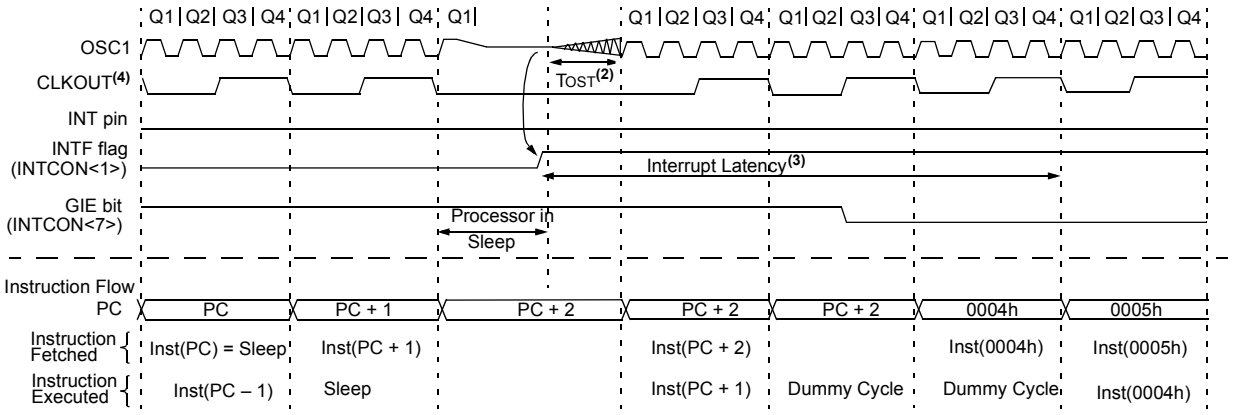
- If the interrupt occurs **before** the execution of a SLEEP instruction, the SLEEP instruction will complete as a NOP. Therefore, the WDT and WDT prescaler and postscaler (if enabled) will not be cleared, the $\overline{TO}$ bit will not be set and the $\overline{PD}$ bit will not be cleared.
- If the interrupt occurs **during or after** the execution of a SLEEP instruction, the device will immediately wake-up from Sleep. The SLEEP instruction will be completely executed before the wake-up. Therefore, the WDT and WDT prescaler and postscaler (if enabled) will be cleared, the $\overline{TO}$ bit will be set and the $\overline{PD}$ bit will be cleared.

Even if the flag bits were checked before executing a SLEEP instruction, it may be possible for flag bits to become set before the SLEEP instruction completes. To determine whether a SLEEP instruction executed, test the $\overline{PD}$ bit. If the $\overline{PD}$ bit is set, the SLEEP instruction was executed as a NOP.

To ensure that the WDT is cleared, a CLRWDT instruction should be executed before a SLEEP instruction.


© 2006-2012 Microchip Technology Inc.

**FIGURE 14-10: WAKE-UP FROM SLEEP THROUGH INTERRUPT**




**Note 1:** XT, HS or LP Oscillator mode assumed.

**2:** TOST = 1024 TOSC (drawing not to scale). This delay does not apply to EC and RC Oscillator modes.

**3:** GIE = 1 assumed. In this case after wake-up, the processor jumps to 0004h. If GIE = 0, execution will continue in-line.

**4:** CLKOUT is not available in XT, HS, LP or EC Oscillator modes, but shown here for timing reference.

## 14.7 Code Protection

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

**Note:** The entire data EEPROM and Flash program memory will be erased when the code protection is switched from on to off. See the "*PIC16F88X Memory Programming Specification*" (DS41287) for more information.

## 14.8 ID Locations

Four memory locations (2000h-2003h) are designated as ID locations where the user can store checksum or other code identification numbers. These locations are not accessible during normal execution but are readable and writable during Program/Verify mode. Only the Least Significant 7 bits of the ID locations are used.

## 14.9 In-Circuit Serial Programming™

The PIC16F882/883/884/886/887 microcontrollers can be serially programmed while in the end application circuit. This is simply done with two lines for clock and data and three other lines for:

- power
- ground
- programming voltage

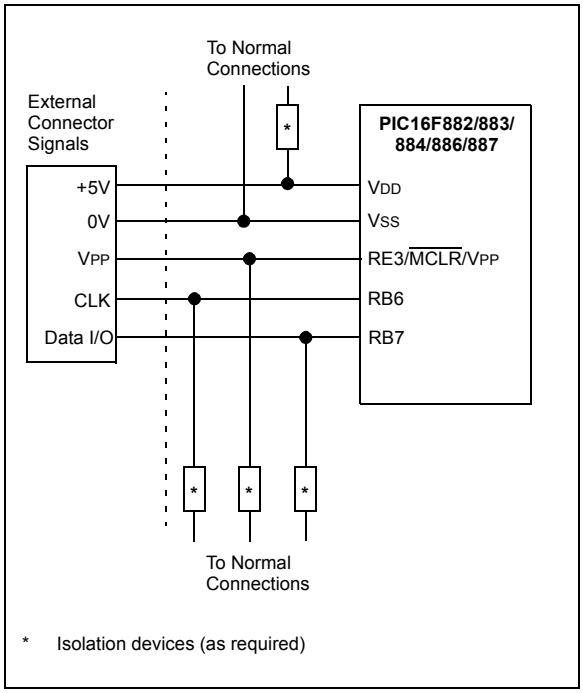
This allows customers to manufacture boards with unprogrammed devices and then program the microcontroller just before shipping the product. This also allows the most recent firmware or a custom firmware to be programmed.

The device is placed into a Program/Verify mode by holding the RB6/ICSPCLK and RB7/ICSPDAT pins low, while raising the MCLR (VPP) pin from VIL to VIHH. See the "*PIC16F88X Memory Programming Specification*" (DS41287) for more information. RB7 becomes the programming data and RB6 becomes the programming clock. Both RB7 and RB6 are Schmitt Trigger inputs in this mode.

After Reset, to place the device into Program/Verify mode, the Program Counter (PC) is at location 00h. A 6-bit command is then supplied to the device. Depending on the command, 14 bits of program data are then supplied to or from the device, depending on whether the command was a Load or a Read. For complete details of serial programming, please refer to the "*PIC16F88X Memory Programming Specification*" (DS41287).

A typical In-Circuit Serial Programming connection is shown in Figure 14-11.

**FIGURE 14-11: TYPICAL IN-CIRCUIT SERIAL PROGRAMMING™ CONNECTION**




## 14.10 Low-Voltage (Single-Supply) ICSP Programming

The LVP bit of the Configuration Word enables low-voltage ICSP programming. This mode allows the microcontroller to be programmed via ICSP using a VDD source in the operating voltage range. This only means that VPP does not have to be brought to VIHH but can instead be left at the normal operating voltage. In this mode, the RB3/PGM pin is dedicated to the programming function and ceases to be a general purpose I/O pin. During programming, VDD is applied to the MCLR pin. To enter Programming mode, VDD must be applied to the RB3/PGM provided the LVP bit is set. The LVP bit defaults to on ('1') from the factory.

**Note 1:** The High-Voltage Programming mode is always available, regardless of the state of the LVP bit, by applying VIHH to the MCLR pin.

**2:** While in Low-Voltage ICSP mode, the RB3 pin can no longer be used as a general purpose I/O pin.

**3:** When using Low-Voltage ICSP Programming (LVP) and the pull-ups on PORTB are enabled, bit 3 in the TRISB register must be cleared to disable the pull-up on RB3 and ensure the proper operation of the device.

**4:** RB3 should not be allowed to float if LVP is enabled. An external pull-down device should be used to default the device to normal operating mode. If RB3 floats high, the PIC16F882/883/884/886/887 devices will enter Programming mode.

**5:** LVP mode is enabled by default on all devices shipped from Microchip. It can be disabled by clearing the LVP bit in the CONFIG register.

If Low-Voltage Programming mode is not used, the LVP bit can be programmed to a '0' and RB3/PGM becomes a digital I/O pin. However, the LVP bit may only be programmed when programming is entered with VIHH on MCLR. The LVP bit can only be charged when using high voltage on MCLR.

It should be noted, that once the LVP bit is programmed to '0', only the High-Voltage Programming mode is available and only High-Voltage Programming mode can be used to program the device.

When using low-voltage ICSP, the part must be supplied at 4.5V to 5.5V if a bulk erase will be executed. This includes reprogramming of the code-protect bits from an on state to an off state. For all other cases of low-voltage ICSP, the part may be programmed at the normal operating voltage. This means calibration values, unique user IDs or user code can be reprogrammed or added.


© 2006-2012 Microchip Technology Inc.

## 14.11 In-Circuit Debugger

The PIC16F882/883/884/886/887-ICD can be used in any of the package types. The devices will be mounted on the target application board, which in turn has a 3 or 4-wire connection to the ICD tool.

When the debug bit in the Configuration Word (CONFIG<13>) is programmed to a '0', the In-Circuit Debugger functionality is enabled. This function allows simple debugging functions when used with MPLAB® ICD 2. When the microcontroller has this feature enabled, some of the resources are not available for general use. See Table 14-10 for more detail.

**Note:** The user's application must have the circuitry required to support ICD functionality. Once the ICD circuitry is enabled, normal device pin functions on RB6/ICSPCLK and RB7/ICSPDAT will not be usable. The ICD circuitry uses these pins for communication with the ICD2 external debugger.

For more information, see "Using *MPLAB® ICD 2*" (DS51265), available on Microchip's web site (www.microchip.com).

### 14.11.1 ICD PINOUT

The devices in the PIC16F88X family carry the circuitry for the In-Circuit Debugger on-chip and on existing device pins. This eliminates the need for a separate die or package for the ICD device. The pinout for the ICD device is the same as the devices (see **Section 1.0 "Device Overview"** for complete pinout and pin descriptions). Table 14-10 shows the location and function of the ICD related pins on the 28 and 40 pin devices.

**TABLE 14-10: PIC16F883/884/886/887-ICD PIN DESCRIPTIONS**

| Pin (PDIP) | | Name | Type | Pull-up | Description |
|---|---|---|---|---|---|
| PIC16F884/887 | PIC16F882/883/886 | | | | |
| 40 | 28 | ICDDATA | TTL | — | In-Circuit Debugger Bidirectional data |
| 39 | 27 | ICDCLK | ST | — | In-Circuit Debugger Bidirectional clock |
| 1 | 1 | $\overline{MCLR}$/VPP | HV | — | Programming voltage |
| 11,32 | 20 | VDD | P | — | |
| 12,31 | 8,19 | VSS | P | — | |

**Legend:** TTL = TTL input buffer, ST = Schmitt Trigger input buffer, P = Power, HV = High Voltage

**NOTES:**


© 2006-2012 Microchip Technology Inc.

## 15.0 INSTRUCTION SET SUMMARY

The PIC16F882/883/884/886/887 instruction set is highly orthogonal and is comprised of three basic categories:

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

Each PIC16 instruction is a 14-bit word divided into an **opcode,** which specifies the instruction type and one or more **operands,** which further specify the operation of the instruction. The formats for each of the categories is presented in Figure 15-1, while the various opcode fields are summarized in Table 15-1.

Table 15-2 lists the instructions recognized by the MPASM™ assembler.

For **byte-oriented** instructions, 'f' represents a file register designator and 'd' represents a destination designator. The file register designator specifies which file register is to be used by the instruction.

The destination designator specifies where the result of the operation is to be placed. If 'd' is zero, the result is placed in the W register. If 'd' is one, the result is placed in the file register specified in the instruction.

For **bit-oriented** instructions, 'b' represents a bit field designator, which selects the bit affected by the operation, while 'f' represents the address of the file in which the bit is located.

For **literal and control** operations, 'k' represents an 8-bit or 11-bit constant, or literal value.

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

All instruction examples use the format '0xhh' to represent a hexadecimal number, where 'h' signifies a hexadecimal digit.

### 15.1 Read-Modify-Write Operations

Any instruction that specifies a file register as part of the instruction performs a Read-Modify-Write (RMW) operation. The register is read, the data is modified, and the result is stored according to either the instruction, or the destination designator 'd'. A read operation is performed on a register even if the instruction writes to that register.

For example, a CLRF PORTA instruction will read PORTA, clear all the data bits, then write the result back to PORTA. This example would have the unintended consequence of clearing the condition that set the RAIF flag.

**TABLE 15-1: OPCODE FIELD DESCRIPTIONS**

| Field | Description |
|---|---|
| f | Register file address (0x00 to 0x7F) |
| W | Working register (accumulator) |
| b | Bit address within an 8-bit file register |
| k | Literal field, constant data or label |
| x | Don't care location (= 0 or 1).<br>The assembler will generate code with x = 0.<br>It is the recommended form of use for compatibility with all Microchip software tools. |
| d | Destination select; d = 0: store result in W,<br>d = 1: store result in file register f.<br>Default is d = 1. |
| PC | Program Counter |
| $\overline{TO}$ | Time-out bit |
| C | Carry bit |
| DC | Digit carry bit |
| Z | Zero bit |
| $\overline{PD}$ | Power-down bit |

**FIGURE 15-1: GENERAL FORMAT FOR INSTRUCTIONS**

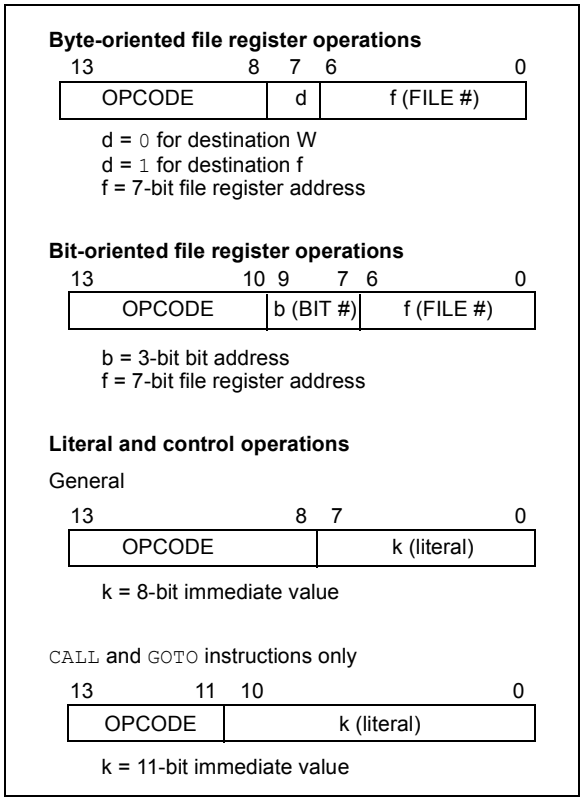


**TABLE 15-2: PIC16F882/883/884/886/887 INSTRUCTION SET**

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

**Note 1:** When an I/O register is modified as a function of itself (e.g., `MOVF GPIO, 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.


© 2006-2012 Microchip Technology Inc.

## 15.2 Instruction Descriptions

**ADDLW** — **Add literal and W**

| | |
|---|---|
| Syntax: | [ *label* ] ADDLW k |
| Operands: | 0 ≤ k ≤ 255 |
| Operation: | (W) + k → (W) |
| Status Affected: | C, DC, Z |
| Description: | The contents of the W register are added to the eight-bit literal 'k' and the result is placed in the W register. |

**ADDWF** — **Add W and f**

| | |
|---|---|
| Syntax: | [ *label* ] ADDWF f,d |
| Operands: | 0 ≤ f ≤ 127<br>d ∈ [0,1] |
| Operation: | (W) + (f) → (destination) |
| Status Affected: | C, DC, Z |
| Description: | Add the contents of the W register with register 'f'. If 'd' is '0', the result is stored in the W register. If 'd' is '1', the result is stored back in register 'f'. |

**ANDLW** — **AND literal with W**

| | |
|---|---|
| Syntax: | [ *label* ] ANDLW k |
| Operands: | 0 ≤ k ≤ 255 |
| Operation: | (W) .AND. (k) → (W) |
| Status Affected: | Z |
| Description: | The contents of W register are AND'ed with the eight-bit literal 'k'. The result is placed in the W register. |

**ANDWF** — **AND W with f**

| | |
|---|---|
| Syntax: | [ *label* ] ANDWF f,d |
| Operands: | 0 ≤ f ≤ 127<br>d ∈ [0,1] |
| Operation: | (W) .AND. (f) → (destination) |
| Status Affected: | Z |
| Description: | AND the W register with register 'f'. If 'd' is '0', the result is stored in the W register. If 'd' is '1', the result is stored back in register 'f'. |

**BCF** — **Bit Clear f**

| | |
|---|---|
| Syntax: | [ *label* ] BCF f,b |
| Operands: | 0 ≤ f ≤ 127<br>0 ≤ b ≤ 7 |
| Operation: | 0 → (f<b>) |
| Status Affected: | None |
| Description: | Bit 'b' in register 'f' is cleared. |

**BSF** — **Bit Set f**

| | |
|---|---|
| Syntax: | [ *label* ] BSF f,b |
| Operands: | 0 ≤ f ≤ 127<br>0 ≤ b ≤ 7 |
| Operation: | 1 → (f<b>) |
| Status Affected: | None |
| Description: | Bit 'b' in register 'f' is set. |

**BTFSC** — **Bit Test f, Skip if Clear**

| | |
|---|---|
| Syntax: | [ *label* ] BTFSC f,b |
| Operands: | 0 ≤ f ≤ 127<br>0 ≤ b ≤ 7 |
| Operation: | skip if (f<b>) = 0 |
| Status Affected: | None |
| Description: | If bit 'b' in register 'f' is '1', the next instruction is executed.<br>If bit 'b' in register 'f' is '0', the next instruction is discarded, and a NOP is executed instead, making this a two-cycle instruction. |

| BTFSS | Bit Test f, Skip if Set |
|---|---|
| Syntax: | [ *label* ] BTFSS f,b |
| Operands: | $0 \le f \le 127$<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 two-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 eleven-bit immediate address is loaded into PC bits <10:0>. The upper bits of the PC are loaded from PCLATH. CALL is a two-cycle instruction. |

| CLRF | Clear f |
|---|---|
| Syntax: | [ *label* ] CLRF f |
| Operands: | $0 \le f \le 127$ |
| 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'. |


© 2006-2012 Microchip Technology Inc.

| DECFSZ | Decrement f, Skip if 0 |
|---|---|
| Syntax: | [ *label* ] DECFSZ f,d |
| Operands: | 0 ≤ f ≤ 127<br>d ∈ [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 two-cycle instruction. |

| GOTO | Unconditional Branch |
|---|---|
| Syntax: | [ *label* ] GOTO k |
| Operands: | 0 ≤ k ≤ 2047 |
| Operation: | k → PC<10:0><br>PCLATH<4:3> → PC<12:11> |
| Status Affected: | None |
| Description: | GOTO is an unconditional branch. The eleven-bit immediate value is loaded into PC bits <10:0>. The upper bits of PC are loaded from PCLATH<4:3>. GOTO is a two-cycle instruction. |

| INCF | Increment f |
|---|---|
| Syntax: | [ *label* ] INCF f,d |
| Operands: | 0 ≤ f ≤ 127<br>d ∈ [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 ≤ f ≤ 127<br>d ∈ [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 two-cycle instruction. |

| IORLW | Inclusive OR literal with W |
|---|---|
| Syntax: | [ *label* ] IORLW k |
| Operands: | 0 ≤ k ≤ 255 |
| Operation: | (W) .OR. k → (W) |
| Status Affected: | Z |
| Description: | The contents of the W register are OR'ed with the eight-bit literal 'k'. The result is placed in the W register. |

| IORWF | Inclusive OR W with f |
|---|---|
| Syntax: | [ *label* ] IORWF f,d |
| Operands: | 0 ≤ f ≤ 127<br>d ∈ [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 ≤ f ≤ 127<br>d ∈ [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 ≤ f ≤ 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 ≤ k ≤ 255 |
| Operation: | k → (W) |
| Status Affected: | None |
| Description: | The eight-bit literal 'k' is loaded into W register. The "don't cares" will assemble as '0's. |
| Words: | 1 |
| Cycles: | 1 |
| Example: | MOVLW 0x5A<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 |


© 2006-2012 Microchip Technology Inc.

| RETFIE | Return from Interrupt |
|---|---|
| Syntax: | [ *label* ] RETFIE |
| Operands: | None |
| Operation: | TOS → PC,<br>1 → GIE |
| Status Affected: | None |
| Description: | Return from Interrupt. Stack is POPed and Top-of-Stack (TOS) is loaded in the PC. Interrupts are enabled by setting Global Interrupt Enable bit, GIE (INTCON<7>). This is a two-cycle instruction. |
| Words: | 1 |
| Cycles: | 2 |
| Example: | RETFIE<br>After Interrupt<br>PC = TOS<br>GIE = 1 |

| RETLW | Return with literal in W |
|---|---|
| Syntax: | [ *label* ] RETLW k |
| Operands: | 0 ≤ k ≤ 255 |
| Operation: | k → (W);<br>TOS → PC |
| Status Affected: | None |
| Description: | The W register is loaded with the eight-bit literal 'k'. The program counter is loaded from the top of the stack (the return address). This is a two-cycle instruction. |
| Words: | 1 |
| Cycles: | 2 |
| Example: | CALL TABLE;W contains table<br>;offset value<br>TABLE • ;W now has<br>• ;table value<br>•<br>•<br>ADDWF PC ;W = offset<br>RETLW k1 ;Begin table<br>RETLW k2 ;<br>•<br>•<br>•<br>RETLW kn ;End of table<br><br>Before Instruction<br>W = 0x07<br>After Instruction<br>W = value of k8 |

| RETURN | Return from Subroutine |
|---|---|
| Syntax: | [ *label* ] RETURN |
| Operands: | None |
| Operation: | TOS → PC |
| Status Affected: | None |
| Description: | Return from subroutine. The stack is POPed and the top of the stack (TOS) is loaded into the program counter. This is a two-cycle instruction. |

**RLF** **Rotate Left f through Carry**

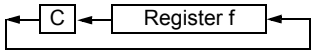
Syntax: [ *label* ] RLF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: See description below

Status Affected: C

Description: The contents of register 'f' are rotated one bit to the left through the Carry flag. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is stored back in register 'f'.




Words: 1

Cycles: 1

Example:

```
RLF     REG1,0
```

Before Instruction

```
REG1    =   1110 0110
C       =   0
```

After Instruction

```
REG1    =   1110 0110
W       =   1100 1100
C       =   1
```

**RRF** **Rotate Right f through Carry**

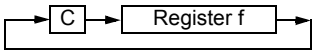
Syntax: [ *label* ] RRF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: See description below

Status Affected: C

Description: The contents of register 'f' are rotated one bit to the right through the Carry flag. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'.




**SLEEP** **Enter Sleep mode**

Syntax: [ *label* ] SLEEP

Operands: None

Operation: 00h → WDT,
0 → WDT prescaler,
1 → $\overline{TO}$,
0 → $\overline{PD}$

Status Affected: $\overline{TO}$, $\overline{PD}$

Description: The power-down Status bit, $\overline{PD}$ is cleared. Time-out Status bit, $\overline{TO}$ is set. Watchdog Timer and its prescaler are cleared.
The processor is put into Sleep mode with the oscillator stopped.

**SUBLW** **Subtract W from literal**

Syntax: [ *label* ] SUBLW k

Operands: $0 \le k \le 255$

Operation: k - (W) → (W)

Status Affected: C, DC, Z

Description: The W register is subtracted (2's complement method) from the eight-bit literal 'k'. The result is placed in the W register.

| | |
|---|---|
| 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> |


© 2006-2012 Microchip Technology Inc.

| SWAPF | Swap Nibbles in f |
|---|---|
| Syntax: | [ *label* ] SWAPF f,d |
| Operands: | $0 \le f \le 127$<br>$d \in [0,1]$ |
| Operation: | (f<3:0>) → (destination<7:4>),<br>(f<7:4>) → (destination<3:0>) |
| Status Affected: | None |
| Description: | The upper and lower nibbles of register 'f' are exchanged. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed in register 'f'. |

| XORLW | Exclusive OR literal with W |
|---|---|
| Syntax: | [ *label* ] XORLW k |
| Operands: | $0 \le k \le 255$ |
| Operation: | (W) .XOR. k → (W) |
| Status Affected: | Z |
| Description: | The contents of the W register are XOR'ed with the eight-bit literal 'k'. The result is placed in the W register. |

| XORWF | Exclusive OR W with f |
|---|---|
| Syntax: | [ *label* ] XORWF f,d |
| Operands: | $0 \le f \le 127$<br>$d \in [0,1]$ |
| Operation: | (W) .XOR. (f) → (destination) |
| Status Affected: | Z |
| Description: | Exclusive OR the contents of the W register with register 'f'. If 'd' is '0', the result is stored in the W register. If 'd' is '1', the result is stored back in register 'f'. |

**NOTES:**


© 2006-2012 Microchip Technology Inc.

## 16.0 DEVELOPMENT SUPPORT

The PIC® microcontrollers and dsPIC® digital signal controllers are supported with a full range of software and hardware development tools:

- Integrated Development Environment
  - MPLAB® IDE Software
- Compilers/Assemblers/Linkers
  - MPLAB C Compiler for Various Device Families
  - HI-TECH C® for Various Device Families
  - MPASM™ Assembler
  - MPLINK™ Object Linker/ MPLIB™ Object Librarian
  - MPLAB Assembler/Linker/Librarian for Various Device Families
- Simulators
  - MPLAB SIM Software Simulator
- Emulators
  - MPLAB REAL ICE™ In-Circuit Emulator
- In-Circuit Debuggers
  - MPLAB ICD 3
  - PICkit™ 3 Debug Express
- Device Programmers
  - PICkit™ 2 Programmer
  - MPLAB PM3 Device Programmer
- Low-Cost Demonstration/Development Boards, Evaluation Kits, and Starter Kits

### 16.1 MPLAB Integrated Development Environment Software

The MPLAB IDE software brings an ease of software development previously unseen in the 8/16/32-bit microcontroller market. The MPLAB IDE is a Windows® operating system-based application that contains:

- A single graphical interface to all debugging tools
  - Simulator
  - Programmer (sold separately)
  - In-Circuit Emulator (sold separately)
  - In-Circuit Debugger (sold separately)
- A full-featured editor with color-coded context
- A multiple project manager
- Customizable data windows with direct edit of contents
- High-level source code debugging
- Mouse over variable inspection
- Drag and drop variables from source to watch windows
- Extensive on-line help
- Integration of select third party tools, such as IAR C Compilers

The MPLAB IDE allows you to:

- Edit your source files (either C or assembly)
- One-touch compile or assemble, and download to emulator and simulator tools (automatically updates all project information)
- Debug using:
  - Source files (C or assembly)
  - Mixed C and assembly
  - Machine code

MPLAB IDE supports multiple debugging tools in a single development paradigm, from the cost-effective simulators, through low-cost in-circuit debuggers, to full-featured emulators. This eliminates the learning curve when upgrading to tools with increased flexibility and power.

## 16.2 MPLAB C Compilers for Various Device Families

The MPLAB C Compiler code development systems are complete ANSI C compilers for Microchip's PIC18, PIC24 and PIC32 families of microcontrollers and the dsPIC30 and dsPIC33 families of digital signal controllers. These compilers provide powerful integration capabilities, superior code optimization and ease of use.

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

## 16.3 HI-TECH C for Various Device Families

The HI-TECH C Compiler code development systems are complete ANSI C compilers for Microchip's PIC family of microcontrollers and the dsPIC family of digital signal controllers. These compilers provide powerful integration capabilities, omniscient code generation and ease of use.

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

The compilers include a macro assembler, linker, preprocessor, and one-step driver, and can run on multiple platforms.

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

## 16.5 MPLINK Object Linker/ MPLIB Object Librarian

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

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

The object linker/library features include:

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

## 16.6 MPLAB Assembler, Linker and Librarian for Various Device Families

MPLAB Assembler produces relocatable machine code from symbolic assembly language for PIC24, PIC32 and dsPIC devices. MPLAB C Compiler uses the assembler to produce its object file. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. Notable features of the assembler include:

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


© 2006-2012 Microchip Technology Inc.

## 16.7 MPLAB SIM Software Simulator

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

The MPLAB SIM Software Simulator fully supports symbolic debugging using the MPLAB C 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.

## 16.8 MPLAB REAL ICE In-Circuit Emulator System

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 PIC® Flash MCUs and dsPIC® Flash DSCs with the easy-to-use, powerful graphical user interface of the MPLAB Integrated Development Environment (IDE), included with each kit.

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 (RJ11) 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 IDE. In upcoming releases of MPLAB IDE, new devices will be supported, and new features will be added. MPLAB REAL ICE offers significant advantages over competitive emulators including low-cost, full-speed emulation, run-time variable watches, trace analysis, complex breakpoints, a ruggedized probe interface and long (up to three meters) interconnection cables.

## 16.9 MPLAB ICD 3 In-Circuit Debugger System

MPLAB ICD 3 In-Circuit Debugger System is Microchip's most cost effective high-speed hardware debugger/programmer for Microchip Flash Digital Signal Controller (DSC) and microcontroller (MCU) devices. It debugs and programs PIC® Flash microcontrollers and dsPIC® DSCs with the powerful, yet easy-to-use graphical user interface of MPLAB Integrated Development Environment (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.

## 16.10 PICkit 3 In-Circuit Debugger/ Programmer and PICkit 3 Debug Express

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 Integrated Development Environment (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 an 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™.

The PICkit 3 Debug Express include the PICkit 3, demo board and microcontroller, hookup cables and CDROM with user's guide, lessons, tutorial, compiler and MPLAB IDE software.

© 2006-2012 Microchip Technology Inc.

## 16.11 PICkit 2 Development Programmer/Debugger and PICkit 2 Debug Express

The PICkit™ 2 Development Programmer/Debugger is a low-cost development tool with an easy to use interface for programming and debugging Microchip's Flash families of microcontrollers. The full featured Windows® programming interface supports baseline (PIC10F, PIC12F5xx, PIC16F5xx), midrange (PIC12F6xx, PIC16F), PIC18F, PIC24, dsPIC30, dsPIC33, and PIC32 families of 8-bit, 16-bit, and 32-bit microcontrollers, and many Microchip Serial EEPROM products. With Microchip's powerful MPLAB Integrated Development Environment (IDE) the PICkit™ 2 enables in-circuit debugging on most PIC® microcontrollers. In-Circuit-Debugging runs, halts and single steps the program while the PIC microcontroller is embedded in the application. When halted at a breakpoint, the file registers can be examined and modified.

The PICkit 2 Debug Express include the PICkit 2, demo board and microcontroller, hookup cables and CDROM with user's guide, lessons, tutorial, compiler and MPLAB IDE software.

## 16.12 MPLAB PM3 Device Programmer

The MPLAB PM3 Device Programmer is a universal, CE compliant device programmer with programmable voltage verification at VDDMIN and VDDMAX 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.

## 16.13 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.


© 2006-2012 Microchip Technology Inc.

## 17.0 ELECTRICAL SPECIFICATIONS

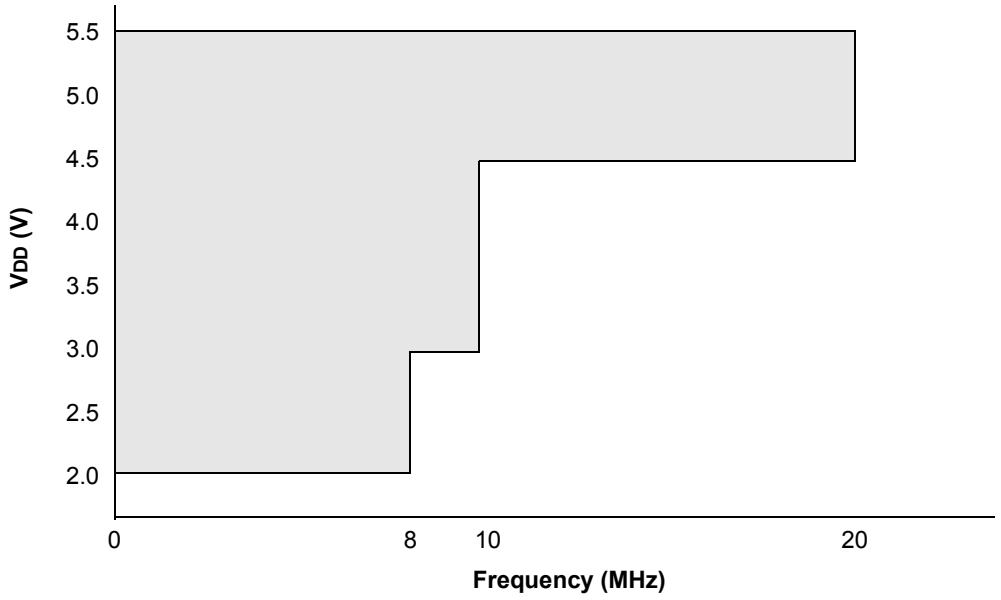
### Absolute Maximum Ratings(†)

Ambient temperature under bias ..... -40° to +125°C
Storage temperature ..... -65°C to +150°C
Voltage on $V_{DD}$ with respect to $V_{SS}$ ..... -0.3V to +6.5V
Voltage on $\overline{MCLR}$ with respect to Vss ..... -0.3V to +13.5V
Voltage on all other pins with respect to $V_{SS}$ ..... -0.3V to ($V_{DD}$ + 0.3V)
Total power dissipation(1) ..... 800 mW
Maximum current out of $V_{SS}$ pin ..... 95 mA
Maximum current into $V_{DD}$ pin ..... 95 mA
Input clamp current, $I_{IK}$ ($V_I$ < 0 or $V_I$ > $V_{DD}$) ..... ± 20 mA
Output clamp current, $I_{OK}$ (Vo < 0 or Vo > $V_{DD}$) ..... ± 20 mA
Maximum output current sunk by any I/O pin ..... 25 mA
Maximum output current sourced by any I/O pin ..... 25 mA
Maximum output current sunk by any I/O PIN ..... 25 mA
Maximum output current sourced by any I/O pin ..... 25 mA

**Note 1:** Power dissipation is calculated as follows: $P_{DIS} = V_{DD} \times \{I_{DD} - \sum I_{OH}\} + \sum \{(V_{DD} - V_{OH}) \times I_{OH}\} + \sum (Vol \times I_{OL})$.

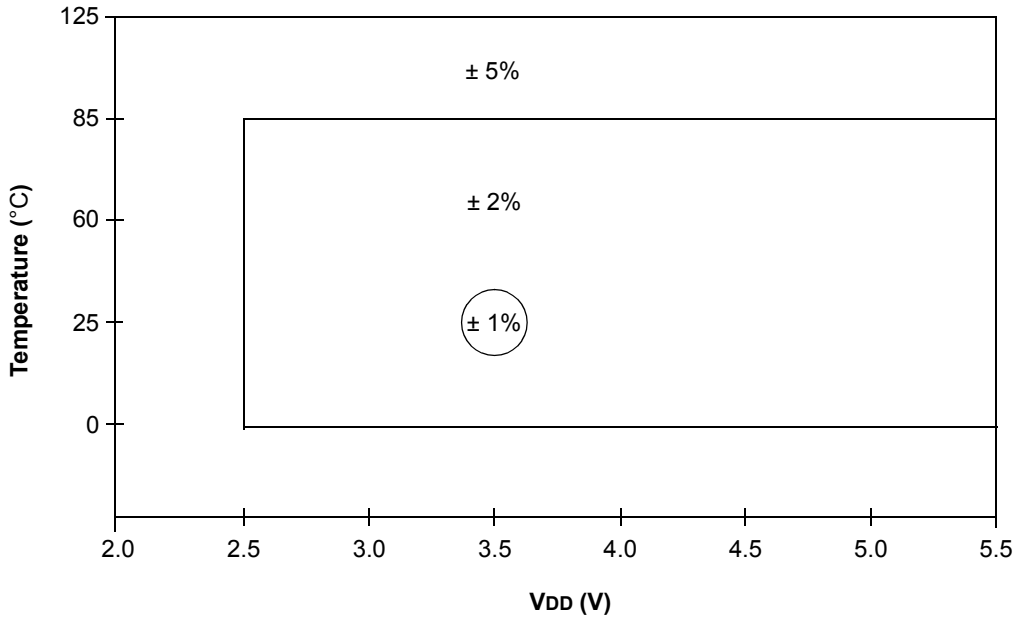
† NOTICE: Stresses above those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure above maximum rating conditions for extended periods may affect device reliability.

**FIGURE 17-1: PIC16F882/883/884/886/887 VOLTAGE-FREQUENCY GRAPH, -40°C ≤ TA ≤ +125°C**




**Note 1:** The shaded region indicates the permissible combinations of voltage and frequency.

**FIGURE 17-2: HFINTOSC FREQUENCY ACCURACY OVER DEVICE VDD AND TEMPERATURE**





© 2006-2012 Microchip Technology Inc.

## 17.1 DC Characteristics: PIC16F882/883/884/886/887-I (Industrial) PIC16F882/883/884/886/887-E (Extended)

| DC CHARACTERISTICS | | | Standard Operating Conditions (unless otherwise stated)<br>Operating temperature -40°C ≤ TA ≤ +85°C for industrial<br>-40°C ≤ TA ≤ +125°C for extended | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Sym.** | **Characteristic** | **Min.** | **Typ†** | **Max.** | **Units** | **Conditions** |
| | VDD | **Supply Voltage** | 2.0 | — | 5.5 | V | FOSC < = 8 MHz: HFINTOSC, EC |
| D001 | | | 2.0 | — | 5.5 | V | FOSC < = 4 MHz |
| D001C | | | 3.0 | — | 5.5 | V | FOSC < = 10 MHz |
| D001D | | | 4.5 | — | 5.5 | V | FOSC < = 20 MHz |
| D002* | VDR | **RAM Data Retention Voltage**(1) | 1.5 | — | — | V | Device in Sleep mode |
| D003 | VPOR | **VDD Start Voltage** to ensure internal Power-on Reset signal | — | VSS | — | V | See **Section 14.2.1 "Power-on Reset (POR)"** for details. |
| D004* | SVDD | **VDD Rise Rate** to ensure internal Power-on Reset signal | 0.05 | — | — | V/ms | See **Section 14.2.1 "Power-on Reset (POR)"** for details. |

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** This is the limit to which VDD can be lowered in Sleep mode without losing RAM data.

## 17.2 DC Characteristics: PIC16F882/883/884/886/887-I (Industrial) PIC16F882/883/884/886/887-E (Extended)

| DC CHARACTERISTICS | | Standard Operating Conditions (unless otherwise stated)<br>Operating temperature -40°C ≤ TA ≤ +85°C for industrial<br>-40°C ≤ TA ≤ +125°C for extended | | | | | |
|---|---|---|---|---|---|---|---|
| Param No. | Device Characteristics | Min. | Typ† | Max. | Units | Conditions | |
| | | | | | | VDD | Note |
| D010 | Supply Current (IDD)(1, 2) | — | 13 | 19 | µA | 2.0 | FOSC = 32 kHz<br>LP Oscillator mode |
| | | — | 22 | 30 | µA | 3.0 | |
| | | — | 33 | 60 | µA | 5.0 | |
| D011* | | — | 180 | 250 | µA | 2.0 | FOSC = 1 MHz<br>XT Oscillator mode |
| | | — | 290 | 400 | µA | 3.0 | |
| | | — | 490 | 650 | µA | 5.0 | |
| D012 | | — | 280 | 380 | µA | 2.0 | FOSC = 4 MHz<br>XT Oscillator mode |
| | | — | 480 | 670 | µA | 3.0 | |
| | | — | 0.9 | 1.4 | mA | 5.0 | |
| D013* | | — | 170 | 295 | µA | 2.0 | FOSC = 1 MHz<br>EC Oscillator mode |
| | | — | 280 | 480 | µA | 3.0 | |
| | | — | 470 | 690 | µA | 5.0 | |
| D014 | | — | 290 | 450 | µA | 2.0 | FOSC = 4 MHz<br>EC Oscillator mode |
| | | — | 490 | 720 | µA | 3.0 | |
| | | — | 0.85 | 1.3 | mA | 5.0 | |
| D015 | | — | 8 | 20 | µA | 2.0 | FOSC = 31 kHz<br>LFINTOSC mode |
| | | — | 16 | 40 | µA | 3.0 | |
| | | — | 31 | 65 | µA | 5.0 | |
| D016* | | — | 416 | 520 | µA | 2.0 | FOSC = 4 MHz<br>HFINTOSC mode |
| | | — | 640 | 840 | µA | 3.0 | |
| | | — | 1.13 | 1.6 | mA | 5.0 | |
| D017 | | — | 0.65 | 0.9 | mA | 2.0 | FOSC = 8 MHz<br>HFINTOSC mode |
| | | — | 1.01 | 1.3 | mA | 3.0 | |
| | | — | 1.86 | 2.3 | mA | 5.0 | |
| D018 | | — | 340 | 580 | µA | 2.0 | FOSC = 4 MHz<br>EXTRC mode(3) |
| | | — | 550 | 900 | µA | 3.0 | |
| | | — | 0.92 | 1.4 | mA | 5.0 | |
| D019 | | — | 3.8 | 4.7 | mA | 4.5 | FOSC = 20 MHz<br>HS Oscillator mode |
| | | — | 4.0 | 4.8 | mA | 5.0 | |

* 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 IDD measurements in active operation mode are: OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to VDD; $\overline{MCLR}$ = VDD; WDT disabled.

**2:** The supply current is mainly a function of the operating voltage and frequency. Other factors, such as I/O pin loading and switching rate, oscillator type, internal code execution pattern and temperature, also have an impact on the current consumption.

**3:** For RC oscillator configurations, current through REXT is not included. The current through the resistor can be extended by the formula IR = VDD/2REXT (mA) with REXT in kΩ.


© 2006-2012 Microchip Technology Inc.

## 17.3 DC Characteristics: PIC16F882/883/884/886/887-I (Industrial)

| DC CHARACTERISTICS | | Standard Operating Conditions (unless otherwise stated)<br>Operating temperature -40°C ≤ TA ≤ +85°C for industrial | | | | | |
|---|---|---|---|---|---|---|---|
| Param No. | Device Characteristics | Min. | Typ† | Max. | Units | Conditions | |
| | | | | | | VDD | Note |
| D020 | **Power-down Base Current(IPD)(2)** | — | 0.05 | 1.2 | µA | 2.0 | WDT, BOR, Comparators, VREF and T1OSC disabled |
| | | — | 0.15 | 1.5 | µA | 3.0 | |
| | | — | 0.35 | 1.8 | µA | 5.0 | |
| | | — | 150 | 500 | nA | 3.0 | -40°C ≤ TA ≤ +25°C |
| D021 | | — | 1.0 | 2.2 | µA | 2.0 | WDT Current(1) |
| | | — | 2.0 | 4.0 | µA | 3.0 | |
| | | — | 3.0 | 7.0 | µA | 5.0 | |
| D022 | | — | 42 | 60 | µA | 3.0 | BOR Current(1) |
| | | — | 85 | 122 | µA | 5.0 | |
| D023 | | — | 32 | 45 | µA | 2.0 | Comparator Current(1), both comparators enabled |
| | | — | 60 | 78 | µA | 3.0 | |
| | | — | 120 | 160 | µA | 5.0 | |
| D024 | | — | 30 | 36 | µA | 2.0 | CVREF Current(1) (high range) |
| | | — | 45 | 55 | µA | 3.0 | |
| | | — | 75 | 95 | µA | 5.0 | |
| D025* | | — | 39 | 47 | µA | 2.0 | CVREF Current(1) (low range) |
| | | — | 59 | 72 | µA | 3.0 | |
| | | — | 98 | 124 | µA | 5.0 | |
| D026 | | — | 2.0 | 5.0 | µA | 2.0 | T1OSC Current(1), 32.768 kHz |
| | | — | 2.5 | 5.5 | µA | 3.0 | |
| | | — | 3.0 | 7.0 | µA | 5.0 | |
| D027 | | — | 0.30 | 1.6 | µA | 3.0 | A/D Current(1), no conversion in progress |
| | | — | 0.36 | 1.9 | µA | 5.0 | |
| D028 | | — | 90 | 125 | µA | 3.0 | VP6 Reference Current |
| | | — | 125 | 162 | µA | 5.0 | |

* 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 current is the sum of the base IDD or IPD and the additional current consumed when this peripheral is enabled. The peripheral Δ current can be determined by subtracting the base IDD or IPD current from this limit. Max values should be used when calculating total current consumption.

**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 VDD.

### 17.4 DC Characteristics: PIC16F882/883/884/886/887-E (Extended)

| DC CHARACTERISTICS | | Standard Operating Conditions (unless otherwise stated)<br>Operating temperature -40°C ≤ TA ≤ +125°C for extended | | | | | |
|---|---|---|---|---|---|---|---|
| Param No. | Device Characteristics | Min. | Typ† | Max. | Units | Conditions | |
| | | | | | | VDD | Note |
| D020E | Power-down Base Current (IPD)(2) | — | 0.05 | 9 | µA | 2.0 | WDT, BOR, Comparators, VREF and T1OSC disabled |
| | | — | 0.15 | 11 | µA | 3.0 | |
| | | — | 0.35 | 15 | µA | 5.0 | |
| D021E | | — | 1 | 28 | µA | 2.0 | WDT Current(1) |
| | | — | 2 | 30 | µA | 3.0 | |
| | | — | 3 | 35 | µA | 5.0 | |
| D022E | | — | 42 | 65 | µA | 3.0 | BOR Current(1) |
| | | — | 85 | 127 | µA | 5.0 | |
| D023E | | — | 32 | 45 | µA | 2.0 | Comparator Current(1), both comparators enabled |
| | | — | 60 | 78 | µA | 3.0 | |
| | | — | 120 | 160 | µA | 5.0 | |
| D024E | | — | 30 | 70 | µA | 2.0 | CVREF Current(1) (high range) |
| | | — | 45 | 90 | µA | 3.0 | |
| | | — | 75 | 120 | µA | 5.0 | |
| D025E* | | — | 39 | 91 | µA | 2.0 | CVREF Current(1) (low range) |
| | | — | 59 | 117 | µA | 3.0 | |
| | | — | 98 | 156 | µA | 5.0 | |
| D026E | | — | 3.5 | 18 | µA | 2.0 | T1OSC Current(1), 32.768 kHz |
| | | — | 4.0 | 21 | µA | 3.0 | |
| | | — | 5.0 | 24 | µA | 5.0 | |
| D027E | | — | 0.30 | 12 | µA | 3.0 | A/D Current(1), no conversion in progress |
| | | — | 0.36 | 16 | µA | 5.0 | |
| D028E | | — | 90 | 130 | µA | 3.0 | VP6 Reference Current |
| | | — | 125 | 170 | µA | 5.0 | |

* 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 current is the sum of the base IDD or IPD and the additional current consumed when this peripheral is enabled. The peripheral ∆ current can be determined by subtracting the base IDD or IPD current from this limit. Max values should be used when calculating total current consumption.

**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 VDD.


© 2006-2012 Microchip Technology Inc.

## 17.5 DC Characteristics: PIC16F882/883/884/886/887-I (Industrial) PIC16F882/883/884/886/887-E (Extended)

| DC CHARACTERISTICS | | | Standard Operating Conditions (unless otherwise stated)<br>Operating temperature -40°C ≤ TA ≤ +85°C for industrial<br>-40°C ≤ TA ≤ +125°C for extended | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Sym.** | **Characteristic** | **Min.** | **Typ†** | **Max.** | **Units** | **Conditions** |
| | VIL | **Input Low Voltage** | | | | | |
| | | I/O Port: | | | | | |
| D030 | | with TTL buffer | Vss | — | 0.8 | V | 4.5V ≤ VDD ≤ 5.5V |
| D030A | | | Vss | — | 0.15 VDD | V | 2.0V ≤ VDD ≤ 4.5V |
| D031 | | with Schmitt Trigger buffer | Vss | — | 0.2 VDD | V | 2.0V ≤ VDD ≤ 5.5V |
| D032 | | $\overline{MCLR}$, OSC1 (RC mode)(1) | VSS | — | 0.2 VDD | V | |
| D033 | | OSC1 (XT and LP modes) | VSS | — | 0.3 | V | |
| D033A | | OSC1 (HS mode) | VSS | — | 0.3 VDD | V | |
| | VIH | Input High Voltage | | | | | |
| | | I/O ports: | | — | | | |
| D040 | | with TTL buffer | 2.0 | — | VDD | V | 4.5V ≤ VDD ≤ 5.5V |
| D040A | | | 0.25 VDD + 0.8 | — | VDD | V | 2.0V ≤ VDD ≤ 4.5V |
| D041 | | with Schmitt Trigger buffer | 0.8 VDD | — | VDD | V | 2.0V ≤ VDD ≤ 5.5V |
| D042 | | $\overline{MCLR}$ | 0.8 VDD | — | VDD | V | |
| D043 | | OSC1 (XT and LP modes) | 1.6 | — | VDD | V | |
| D043A | | OSC1 (HS mode) | 0.7 VDD | — | VDD | V | |
| D043B | | OSC1 (RC mode) | 0.9 VDD | — | VDD | V | **(Note 1)** |
| | IIL | Input Leakage Current(2) | | | | | |
| D060 | | I/O ports | — | ± 0.1 | ± 1 | µA | VSS ≤ VPIN ≤ VDD,<br>Pin at high-impedance |
| D061 | | $\overline{MCLR}$(3) | — | ± 0.1 | ± 5 | µA | VSS ≤ VPIN ≤ VDD |
| D063 | | OSC1 | — | ± 0.1 | ± 5 | µA | VSS ≤ VPIN ≤ VDD, XT, HS and LP oscillator configuration |
| D070* | IPUR | PORTB Weak Pull-up Current | 50 | 250 | 400 | µA | VDD = 5.0V, VPIN = VSS |
| | VOL | Output Low Voltage(5) | | | | | |
| D080 | | I/O ports | — | — | 0.6 | V | IOL = 8.5 mA, VDD = 4.5V (Ind.) |
| | VOH | Output High Voltage(5) | | | | | |
| D090 | | I/O ports | VDD – 0.7 | — | — | V | IOH = -3.0 mA, VDD = 4.5V (Ind.) |

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** In RC oscillator configuration, the OSC1/CLKIN pin is a Schmitt Trigger input. It is not recommended to use an external clock in RC mode.

**2:** Negative current is defined as current sourced by the pin.

**3:** The leakage current on the $\overline{MCLR}$ pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.

**4:** See **Section 10.3.1 "Using the Data EEPROM"** for additional information.

**5:** Including OSC2 in CLKOUT mode.

## 17.5 DC Characteristics: PIC16F882/883/884/886/887-I (Industrial) PIC16F882/883/884/886/887-E (Extended) (Continued)

| DC CHARACTERISTICS | | | Standard Operating Conditions (unless otherwise stated)<br>Operating temperature -40°C ≤ TA ≤ +85°C for industrial<br>-40°C ≤ TA ≤ +125°C for extended | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Sym.** | **Characteristic** | **Min.** | **Typ†** | **Max.** | **Units** | **Conditions** |
| D100 | IULP | Ultra Low-Power Wake-Up Current | — | 200 | — | nA | See Application Note AN879, "*Using the Microchip Ultra Low-Power Wake-up Module*" (DS00879) |
| | | **Capacitive Loading Specs on Output Pins** | | | | | |
| D101* | COSC2 | OSC2 pin | — | — | 15 | pF | In XT, HS and LP modes when external clock is used to drive OSC1 |
| D101A* | CIO | All I/O pins | — | — | 50 | pF | |
| | | **Data EEPROM Memory** | | | | | |
| D120 | ED | Byte Endurance | 100K | 1M | — | E/W | -40°C ≤ TA ≤ +85°C |
| D120A | ED | Byte Endurance | 10K | 100K | — | E/W | +85°C ≤ TA ≤ +125°C |
| D121 | VDRW | VDD for Read/Write | VMIN | — | 5.5 | V | Using EECON1 to read/write VMIN = Minimum operating voltage |
| D122 | TDEW | Erase/Write Cycle Time | — | 5 | 6 | ms | |
| D123 | TRETD | Characteristic Retention | 40 | — | — | Year | Provided no other specifications are violated |
| D124 | TREF | Number of Total Erase/Write Cycles before Refresh(4) | 1M | 10M | — | E/W | -40°C ≤ TA ≤ +85°C |
| | | **Program Flash Memory** | | | | | |
| D130 | EP | Cell Endurance | 10K | 100K | — | E/W | -40°C ≤ TA ≤ +85°C |
| D130A | ED | Cell Endurance | 1K | 10K | — | E/W | +85°C ≤ TA ≤ +125°C |
| D131 | VPR | VDD for Read | VMIN | — | 5.5 | V | VMIN = Minimum operating voltage |
| D132 | VPEW | VDD for Row Erase/Write | VMIN | — | 5.5 | V | |
| | | VDD for Bulk Erase Operations | 4.5 | — | 5.5 | V | |
| D133 | TPEW | Erase/Write cycle time | — | 2 | 2.5 | ms | |
| D134 | TRETD | Characteristic Retention | 40 | — | — | Year | Provided no other specifications are violated |

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** In RC oscillator configuration, the OSC1/CLKIN pin is a Schmitt Trigger input. It is not recommended to use an external clock in RC mode.

**2:** Negative current is defined as current sourced by the pin.

**3:** The leakage current on the $\overline{MCLR}$ pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.

**4:** See **Section 10.3.1 "Using the Data EEPROM"** for additional information.

**5:** Including OSC2 in CLKOUT mode.


© 2006-2012 Microchip Technology Inc.

## 17.6 Thermal Considerations

| **Standard Operating Conditions (unless otherwise stated)**<br>Operating temperature -40°C ≤ TA ≤ +125°C | | | | | |
|---|---|---|---|---|---|
| **Param No.** | **Sym.** | **Characteristic** | **Typ.** | **Units** | **Conditions** |
| TH01 | θJA | Thermal Resistance Junction to Ambient | 47.2 | C/W | 40-pin PDIP package |
| | | | 24.4 | C/W | 44-pin QFN package |
| | | | 45.8 | C/W | 44-pin TQFP package |
| | | | 60.2 | C/W | 28-pin PDIP package |
| | | | 80.2 | C/W | 28-pin SOIC package |
| | | | 89.4 | C/W | 28-pin SSOP package |
| | | | 29 | C/W | 28-pin QFN package |
| TH02 | θJC | Thermal Resistance Junction to Case | 24.7 | C/W | 40-pin PDIP package |
| | | | 20.0 | C/W | 44-pin QFN package |
| | | | 14.5 | C/W | 44-pin TQFP package |
| | | | 29 | C/W | 28-pin PDIP package |
| | | | 23.8 | C/W | 28-pin SOIC package |
| | | | 23.9 | C/W | 28-pin SSOP package |
| | | | 20.0 | C/W | 28-pin QFN package |
| TH03 | TJ | Junction Temperature | 150 | C | For derated power calculations |
| TH04 | PD | Power Dissipation | — | W | PD = PINTERNAL + PI/O |
| TH05 | PINTERNAL | Internal Power Dissipation | — | W | PINTERNAL = IDD x VDD **(NOTE 1)** |
| TH06 | PI/O | I/O Power Dissipation | — | W | PI/O = Σ (IOL * VOL) + Σ (IOH * (VDD - VOH)) |
| TH07 | PDER | Derated Power | — | W | PDER = (TJ - TA)/θJA **(NOTE 2, 3)** |

**Note 1:** IDD is current to run the chip alone without driving any load on the output pins.

**2:** TA = Ambient Temperature.

**3:** Maximum allowable power dissipation is the lower value of either the absolute maximum total power dissipation or derated power (PDER).

## 17.7 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 |

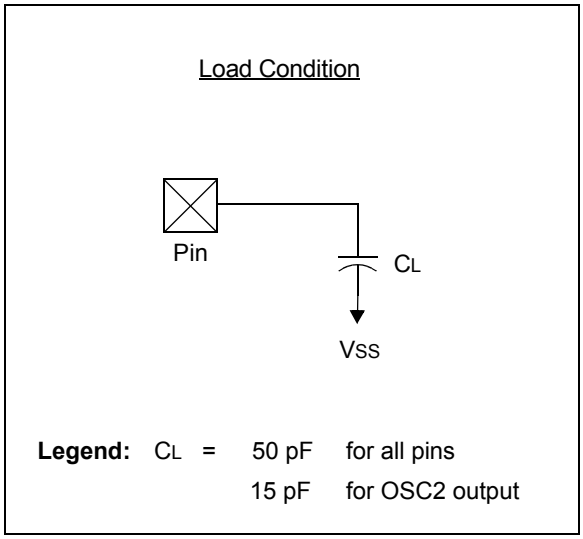
Lowercase letters (pp) and their meanings:

| 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 17-3: LOAD CONDITIONS**





© 2006-2012 Microchip Technology Inc.

## 17.8 AC Characteristics: PIC16F882/883/884/886/887 (Industrial, Extended)

**FIGURE 17-4: CLOCK TIMING**

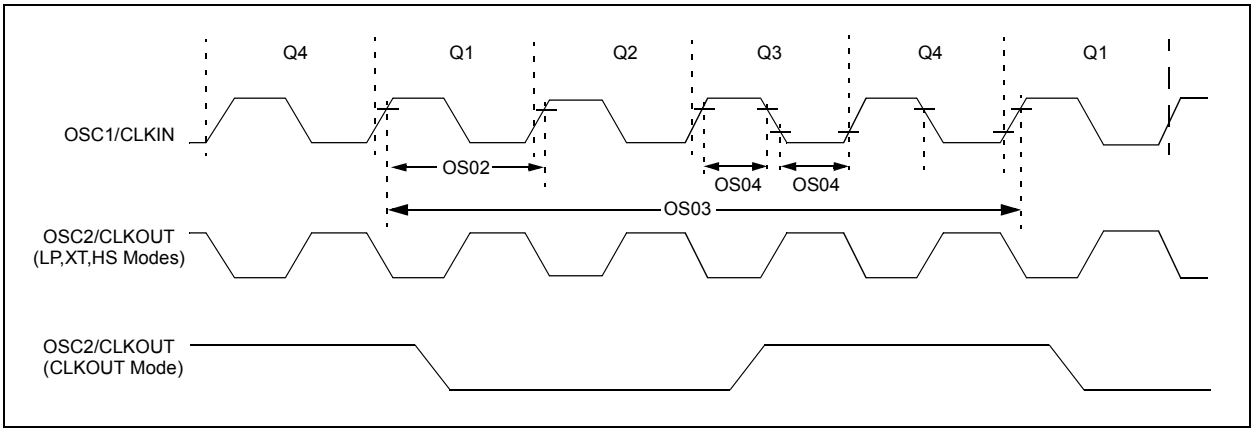



**TABLE 17-1: CLOCK OSCILLATOR TIMING REQUIREMENTS**

| **Standard Operating Conditions (unless otherwise stated)**<br>Operating temperature -40°C ≤ TA ≤ +125°C | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Sym.** | **Characteristic** | **Min.** | **Typ†** | **Max.** | **Units** | **Conditions** |
| OS01 | FOSC | External CLKIN Frequency(1) | DC | — | 37 | kHz | LP Oscillator mode |
| | | | DC | — | 4 | MHz | XT Oscillator mode |
| | | | DC | — | 20 | MHz | HS Oscillator mode |
| | | | DC | — | 20 | MHz | EC Oscillator mode |
| | | Oscillator Frequency(1) | — | 32.768 | — | kHz | LP Oscillator mode |
| | | | 0.1 | — | 4 | MHz | XT Oscillator mode |
| | | | 1 | — | 20 | MHz | HS Oscillator mode |
| | | | DC | — | 4 | MHz | RC Oscillator mode |
| OS02 | TOSC | External CLKIN Period(1) | 27 | — | • | μs | LP Oscillator mode |
| | | | 250 | — | • | ns | XT Oscillator mode |
| | | | 50 | — | • | ns | HS Oscillator mode |
| | | | 50 | — | • | ns | EC Oscillator mode |
| | | Oscillator Period(1) | — | 30.5 | — | μs | LP Oscillator mode |
| | | | 250 | — | 10,000 | ns | XT Oscillator mode |
| | | | 50 | — | 1,000 | ns | HS Oscillator mode |
| | | | 250 | — | — | ns | RC Oscillator mode |
| OS03 | TCY | Instruction Cycle Time(1) | 200 | TCY | DC | ns | TCY = 4/FOSC |
| OS04* | TosH, TosL | External CLKIN High, External CLKIN Low | 2 | — | — | μs | LP oscillator |
| | | | 100 | — | — | ns | XT oscillator |
| | | | 20 | — | — | ns | HS oscillator |
| OS05* | TosR, TosF | External CLKIN Rise, External CLKIN Fall | 0 | — | • | ns | LP oscillator |
| | | | 0 | — | • | ns | XT oscillator |
| | | | 0 | — | • | ns | HS oscillator |

* These parameters are characterized but not tested.

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

**Note 1:** Instruction cycle period (TCY) equals four times the input oscillator time base period. All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at "min" values with an external clock applied to OSC1 pin. When an external clock input is used, the "max" cycle time limit is "DC" (no clock) for all devices.

**TABLE 17-2: OSCILLATOR PARAMETERS**

| **Standard Operating Conditions (unless otherwise stated)**<br>Operating Temperature -40°C ≤ TA ≤ +125°C | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Param No.** | **Sym.** | **Characteristic** | **Freq. Tolerance** | **Min.** | **Typ†** | **Max.** | **Units** | **Conditions** |
| OS06 | TWARM | Internal Oscillator Switch when running(3) | — | — | — | 2 | TOSC | Slowest clock |
| OS07 | TSC | Fail-Safe Sample Clock Period(1) | — | — | 21 | — | ms | LFINTOSC/64 |
| OS08 | HFOSC | Internal Calibrated HFINTOSC Frequency(2) | ±1% | 7.92 | 8.0 | 8.08 | MHz | VDD = 3.5V, 25°C |
| | | | ±2% | 7.84 | 8.0 | 8.16 | MHz | 2.5V ≤ VDD ≤ 5.5V, 0°C ≤ TA ≤ +85°C |
| | | | ±5% | 7.60 | 8.0 | 8.40 | MHz | 2.0V ≤ VDD ≤ 5.5V, -40°C ≤ TA ≤ +85°C (Ind.), -40°C ≤ TA ≤ +125°C (Ext.) |
| OS09* | LFOSC | Internal Uncalibrated LFINTOSC Frequency | — | 15 | 31 | 45 | kHz | |
| OS10* | TIOSC ST | HFINTOSC Oscillator Wake-up from Sleep Start-up Time | — | 5.5 | 12 | 24 | µs | VDD = 2.0V, -40°C to +85°C |
| | | | — | 3.5 | 7 | 14 | µs | VDD = 3.0V, -40°C to +85°C |
| | | | — | 3 | 6 | 11 | µs | VDD = 5.0V, -40°C to +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:** 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 the OSC1 pin. When an external clock input is used, the "max" cycle time limit is "DC" (no clock) for all devices.

**2:** 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.

**3:** By design.


© 2006-2012 Microchip Technology Inc.

**FIGURE 17-5: CLKOUT AND I/O TIMING**

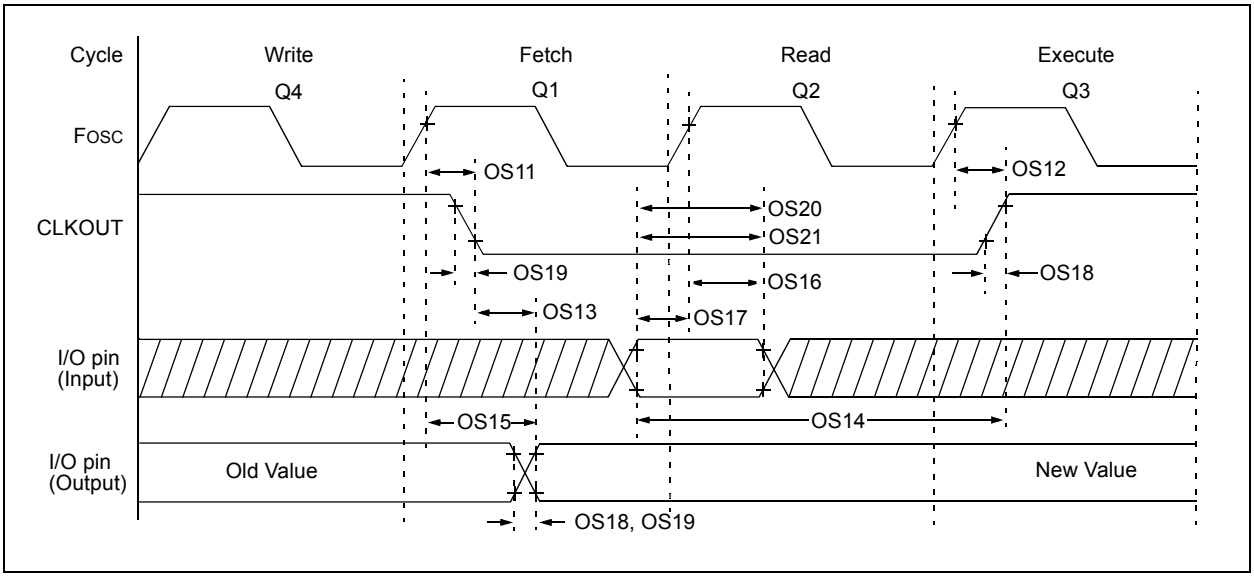



**TABLE 17-3: CLKOUT AND I/O TIMING PARAMETERS**

| **Standard Operating Conditions (unless otherwise stated)**<br>Operating Temperature -40°C ≤ TA ≤ +125°C | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Sym.** | **Characteristic** | **Min.** | **Typ†** | **Max.** | **Units** | **Conditions** |
| OS11 | TOSH2CKL | FOSC↑ to CLKOUT↓ (1) | — | — | 70 | ns | VDD = 5.0V |
| OS12 | TOSH2CKH | FOSC↑ to CLKOUT↑ (1) | — | — | 72 | ns | VDD = 5.0V |
| OS13 | TCKL2IOV | CLKOUT↓ to Port out valid(1) | — | — | 20 | ns | |
| OS14 | TIOV2CKH | Port input valid before CLKOUT↑(1) | TOSC + 200 ns | — | — | ns | |
| OS15* | TOSH2IOV | FOSC↑ (Q1 cycle) to Port out valid | — | 50 | 70 | ns | VDD = 5.0V |
| OS16 | TOSH2IOI | FOSC↑ (Q2 cycle) to Port input invalid (I/O in hold 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(2) | —<br>— | 15<br>40 | 72<br>32 | ns | VDD = 2.0V<br>VDD = 5.0V |
| OS19 | TIOF | Port output fall time(2) | —<br>— | 28<br>15 | 55<br>30 | ns | VDD = 2.0V<br>VDD = 5.0V |
| OS20* | TINP | INT pin input high or low time | 25 | — | — | ns | |
| OS21* | TRAP | PORTA interrupt-on-change new input level time | TCY | — | — | ns | |

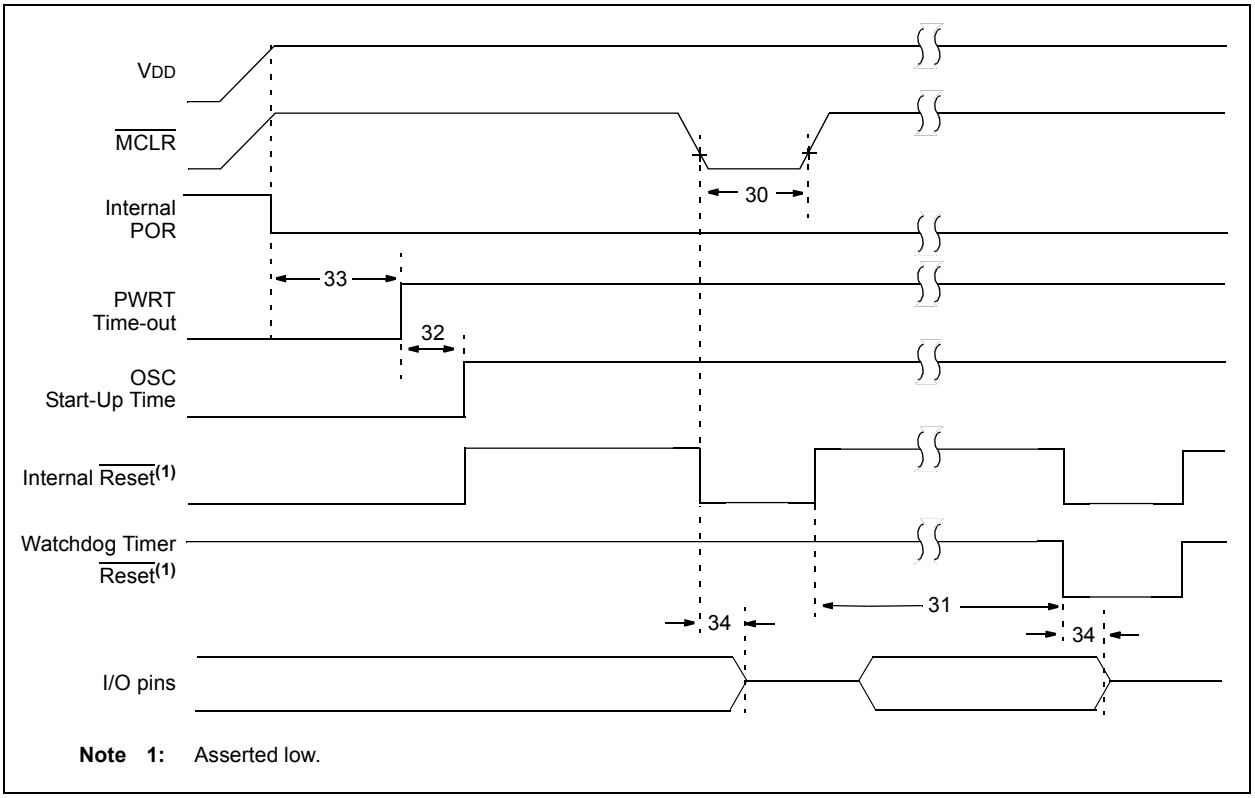
* These parameters are characterized but not tested.

† 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.

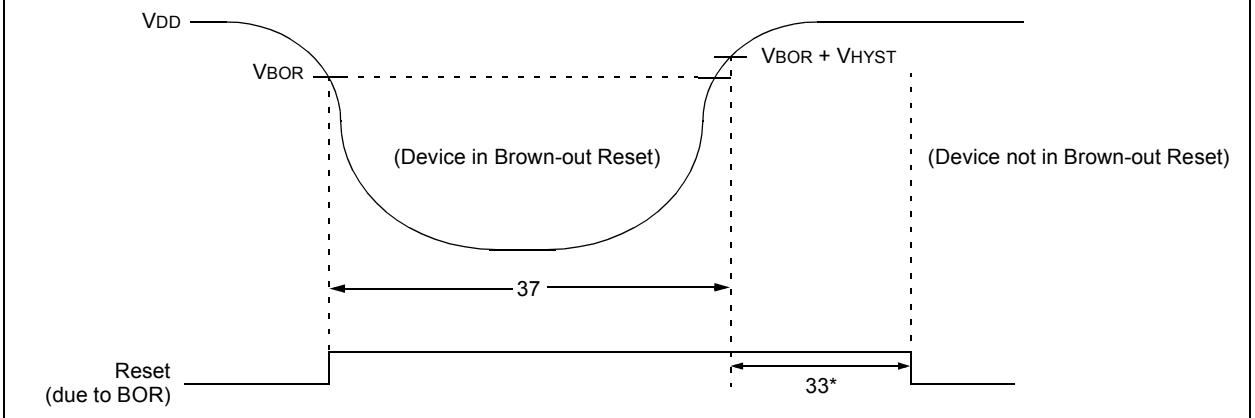
**2:** Includes OSC2 in CLKOUT mode.

**FIGURE 17-6: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING**




**Note 1:** Asserted low.

**FIGURE 17-7: BROWN-OUT RESET TIMING AND CHARACTERISTICS**




* 64 ms delay only if PWRTE bit in the Configuration Word Register 1 is programmed to '0'.


© 2006-2012 Microchip Technology Inc.

**TABLE 17-4: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER, POWER-UP TIMER AND BROWN-OUT RESET PARAMETERS**

| **Standard Operating Conditions (unless otherwise stated)**<br>Operating Temperature -40°C ≤ TA ≤ +125°C | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Sym.** | **Characteristic** | **Min.** | **Typ†** | **Max.** | **Units** | **Conditions** |
| 30 | TMCL | $\overline{\text{MCLR}}$ Pulse Width (low) | 2<br>5 | —<br>— | —<br>— | μs<br>μs | VDD = 5V, -40°C to +85°C<br>VDD = 5V |
| 31 | TWDT | Watchdog Timer Time-out Period (No Prescaler) | 10<br>10 | 16<br>16 | 29<br>31 | ms<br>ms | VDD = 5V, -40°C to +85°C<br>VDD = 5V |
| 32 | TOST | Oscillation Start-up Timer Period**(1, 2)** | — | 1024 | — | TOSC | **(NOTE 3)** |
| 33* | TPWRT | Power-up Timer Period | 40 | 65 | 140 | ms | |
| 34* | TIOZ | I/O High-impedance from $\overline{\text{MCLR}}$ Low or Watchdog Timer Reset | — | — | 2.0 | μs | |
| 35 | VBOR | Brown-out Reset Voltage | 2.0 | — | 2.2 | V | BOR4V bit = 0 **(NOTE 4)** |
| | | | 3.6 | 4.0 | 4.4 | V | BOR4V bit = 1, -40°C to +85°C **(NOTE 4)** |
| | | | 3.6 | 4.0 | 4.5 | V | BOR4V bit = 1, -40°C to +125°C **(NOTE 4)** |
| 36* | VHYST | Brown-out Reset Hysteresis | — | 50 | — | mV | |
| 37* | TBOR | Brown-out Reset 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:** 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 the OSC1 pin. When an external clock input is used, the "max" cycle time limit is "DC" (no clock) for all devices.

**2:** By design.

**3:** Period of the slower clock.

**4:** 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 17-8: TIMER0 AND TIMER1 EXTERNAL CLOCK TIMINGS**

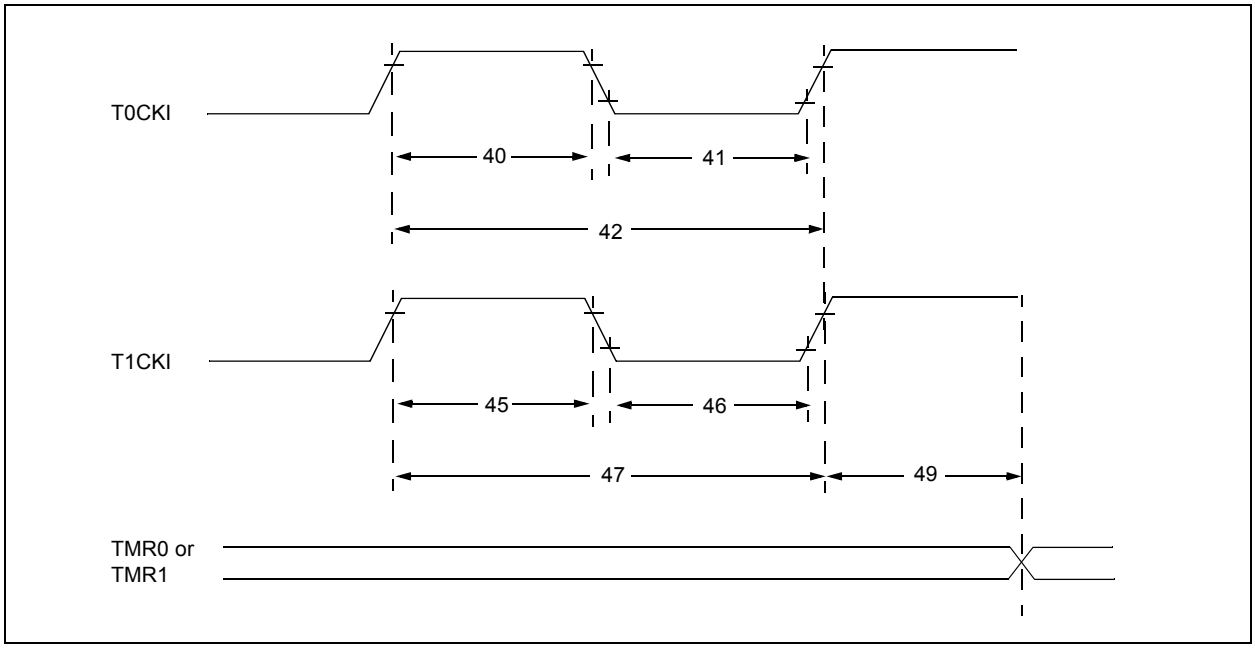



**TABLE 17-5: TIMER0 AND TIMER1 EXTERNAL CLOCK REQUIREMENTS**

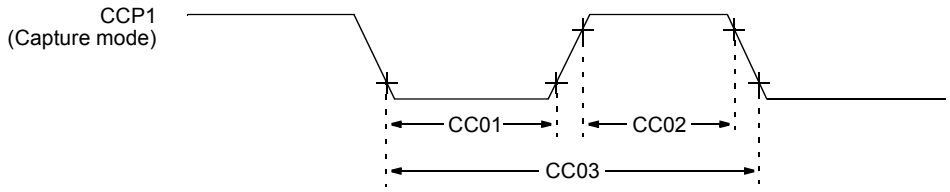
| **Standard Operating Conditions (unless otherwise stated)**<br>Operating Temperature -40°C ≤ TA ≤ +125°C | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Param No.** | **Sym.** | **Characteristic** | | **Min.** | **Typ†** | **Max.** | **Units** | **Conditions** |
| 40* | TT0H | T0CKI High Pulse Width | No Prescaler | 0.5 TCY + 20 | — | — | ns | |
| | | | With Prescaler | 10 | — | — | ns | |
| 41* | TT0L | T0CKI Low Pulse Width | No Prescaler | 0.5 TCY + 20 | — | — | ns | |
| | | | With Prescaler | 10 | — | — | ns | |
| 42* | TT0P | T0CKI Period | | Greater of: 20 or TCY + 40 / N | — | — | ns | N = prescale value (2, 4, ..., 256) |
| 45* | TT1H | T1CKI High Time | Synchronous, No Prescaler | 0.5 TCY + 20 | — | — | ns | |
| | | | Synchronous, with Prescaler | 15 | — | — | ns | |
| | | | Asynchronous | 30 | — | — | ns | |
| 46* | TT1L | T1CKI Low Time | Synchronous, No Prescaler | 0.5 TCY + 20 | — | — | ns | |
| | | | Synchronous, with Prescaler | 15 | — | — | ns | |
| | | | Asynchronous | 30 | — | — | ns | |
| 47* | TT1P | T1CKI Input Period | Synchronous | Greater of: 30 or TCY + 40 / N | — | — | ns | N = prescale value (1, 2, 4, 8) |
| | | | Asynchronous | 60 | — | — | ns | |
| 48 | FT1 | Timer1 Oscillator Input Frequency Range (oscillator enabled by setting bit T1OSCEN) | | — | 32.768 | — | kHz | |
| 49* | TCKEZTMR1 | Delay from External Clock Edge to Timer Increment | | 2 TOSC | — | 7 TOSC | — | Timers in Sync mode |

* These parameters are characterized but not tested.

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


© 2006-2012 Microchip Technology Inc.

**FIGURE 17-9: CAPTURE/COMPARE/PWM TIMINGS (ECCP)**




**Note:** Refer to Figure 17-3 for load conditions.

**TABLE 17-6: CAPTURE/COMPARE/PWM REQUIREMENTS (ECCP)**

| **Standard Operating Conditions (unless otherwise stated)**<br>Operating Temperature -40°C ≤ TA ≤ +125°C | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Param No.** | **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 17-7: COMPARATOR SPECIFICATIONS**

| **Standard Operating Conditions (unless otherwise stated)**<br>Operating Temperature -40°C ≤ TA ≤ +125°C | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Param No.** | **Sym.** | **Characteristics** | | **Min.** | **Typ†** | **Max.** | **Units** | **Comments** |
| CM01 | VOS | Input Offset Voltage | | — | ± 5.0 | ± 10 | mV | (VDD - 1.5)/2 |
| CM02 | VCM | Input Common Mode Voltage | | 0 | — | VDD - 1.5 | V | |
| CM03* | CMRR | Common Mode Rejection Ratio | | +55 | — | — | dB | |
| CM04* | TRT | Response Time | Falling | — | 150 | 600 | ns | **(NOTE 1)** |
| | | | Rising | — | 200 | 1000 | ns | |
| CM05* | TMC2COV | Comparator Mode Change to Output Valid | | — | — | 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:** Response time is measured with one comparator input at (VDD - 1.5)/2 - 100 mV to (VDD - 1.5)/2 + 20 mV.

**TABLE 17-8: COMPARATOR VOLTAGE REFERENCE (CVREF) SPECIFICATIONS**

| **Standard Operating Conditions (unless otherwise stated)**<br>Operating temperature -40°C ≤ TA ≤ +125°C | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Sym.** | **Characteristics** | **Min.** | **Typ†** | **Max.** | **Units** | **Comments** |
| CV01* | CLSB | Step Size(2) | —<br>— | VDD/24<br>VDD/32 | —<br>— | V<br>V | Low Range (VRR = 1)<br>High Range (VRR = 0) |
| CV02* | CACC | Absolute Accuracy | —<br>— | —<br>— | ± 1/2<br>± 1/2 | LSb<br>LSb | Low Range (VRR = 1)<br>High Range (VRR = 0) |
| CV03* | CR | Unit Resistor Value (R) | — | 2k | — | Ω | |
| CV04* | CST | Settling Time(1) | — | — | 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:** Settling time measured while VRR = 1 and VR<3:0> transitions from '0000' to '1111'.

**2:** See **Section 8.10 "Comparator Voltage Reference"** for more information.

**TABLE 17-9: VOLTAGE (VR) REFERENCE SPECIFICATIONS**

| **VR Voltage Reference Specifications** | | | **Standard Operating Conditions (unless otherwise stated)**<br>Operating temperature -40°C ≤ TA ≤ +125°C | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Symbol** | **Characteristics** | **Min.** | **Typ.** | **Max.** | **Units** | **Comments** |
| VR01 | VROUT | VR voltage output | 0.5 | 0.6 | 0.7 | V | |
| VR02* | TSTABLE | Settling Time | — | 10 | 100* | µs | |

* These parameters are characterized but not tested.


© 2006-2012 Microchip Technology Inc.

**TABLE 17-10: PIC16F882/883/884/886/887 A/D CONVERTER (ADC) CHARACTERISTICS**

| **Standard Operating Conditions (unless otherwise stated)**<br>Operating temperature -40°C ≤ TA ≤ +125°C | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Sym.** | **Characteristic** | **Min.** | **Typ†** | **Max.** | **Units** | **Conditions** |
| AD01 | NR | Resolution | — | — | 10 bits | bit | |
| AD02 | EIL | Integral Error | — | — | ±1 | LSb | VREF = 5.12V |
| AD03 | EDL | Differential Error | — | — | ±1 | LSb | No missing codes to 10 bits<br>VREF = 5.12V |
| AD04 | EOFF | Offset Error | 0 | +1.5 | +3.0 | LSb | VREF = 5.12V |
| AD07 | EGN | Gain Error | — | — | ±1 | LSb | VREF = 5.12V |
| AD06<br>AD06A | VREF | Reference Voltage(3) | 2.2<br>2.7 | — | —<br>VDD | V | <br>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 | |
| AD09* | IREF | VREF Input Current(3) | 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 5.0V, 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 A/D conversion result never decreases with an increase in the input voltage and has no missing codes.

**3:** ADC VREF is from external VREF or VDD pin, whichever is selected as reference input.

**4:** When ADC is off, it will not consume any current other than leakage current. The power-down current specification includes any such leakage from the ADC module.

**TABLE 17-11: PIC16F882/883/884/886/887 A/D CONVERSION REQUIREMENTS**

| **Standard Operating Conditions (unless otherwise stated)**<br>Operating temperature -40°C ≤ TA ≤ +125°C | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Sym.** | **Characteristic** | **Min.** | **Typ†** | **Max.** | **Units** | **Conditions** |
| AD130* | TAD | A/D Clock Period | 1.6 | — | 9.0 | µs | TOSC-based, VREF ≥ 3.0V |
| | | | 3.0 | — | 9.0 | µs | TOSC-based, VREF full range |
| | | A/D Internal RC Oscillator Period | 3.0 | 6.0 | 9.0 | µs | ADCS<1:0> = 11 (ADRC mode)<br>At VDD = 2.5V |
| | | | 1.6 | 4.0 | 6.0 | µs | At VDD = 5.0V |
| AD131 | TCNV | Conversion Time (not including Acquisition Time)(1) | — | 11 | — | TAD | Set GO/$\overline{\text{DONE}}$ bit to new data in A/D Result register |
| AD132* | TACQ | Acquisition Time | | 11.5 | — | µs | |
| AD133* | TAMP | Amplifier Settling Time | — | — | 5 | µs | |
| AD134 | TGO | Q4 to A/D Clock Start | — | TOSC/2 | — | — | |
| | | | — | TOSC/2 + TCY | — | — | If the A/D clock source is selected as RC, a time of TCY is added before the A/D clock starts. This allows the SLEEP instruction to be executed. |

* These parameters are characterized but not tested.

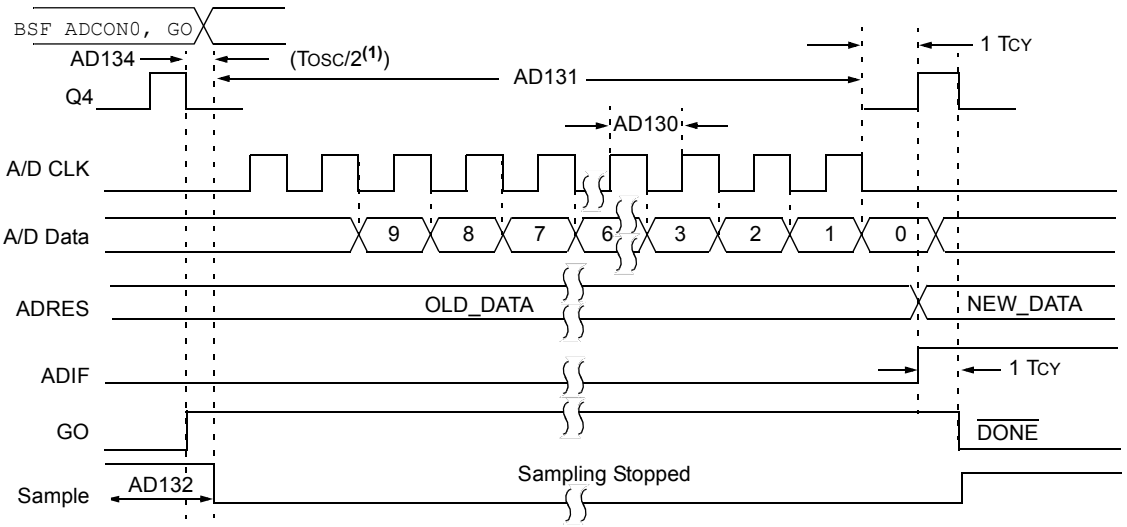
† Data in "Typ" column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** ADRESH and ADRESL registers may be read on the following TCY cycle.

**2:** See **Section 9.3 "A/D Acquisition Requirements"** for minimum conditions.

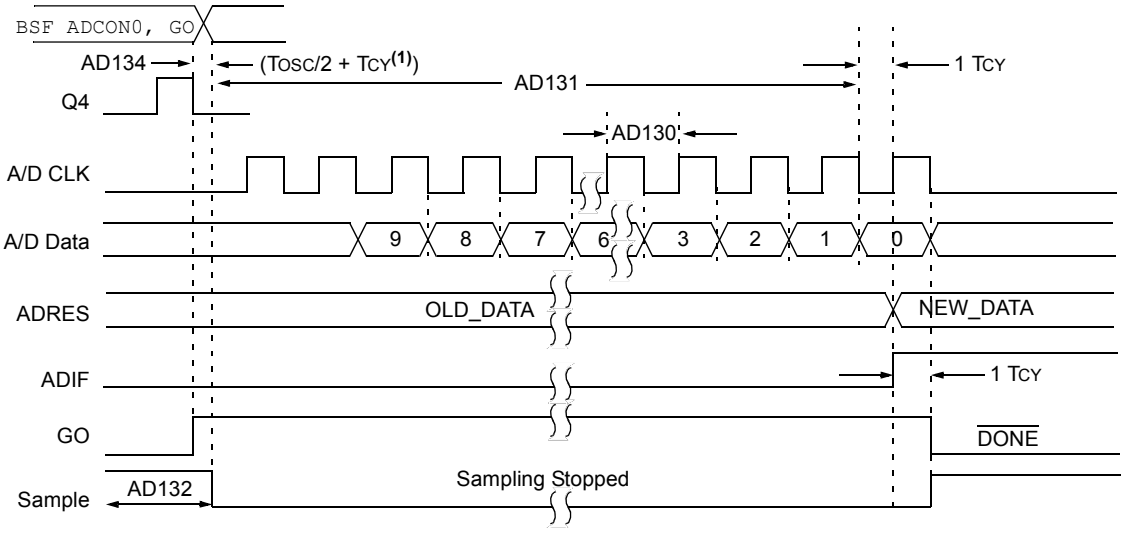

© 2006-2012 Microchip Technology Inc.

**FIGURE 17-10: PIC16F882/883/884/886/887 A/D CONVERSION TIMING (NORMAL MODE)**



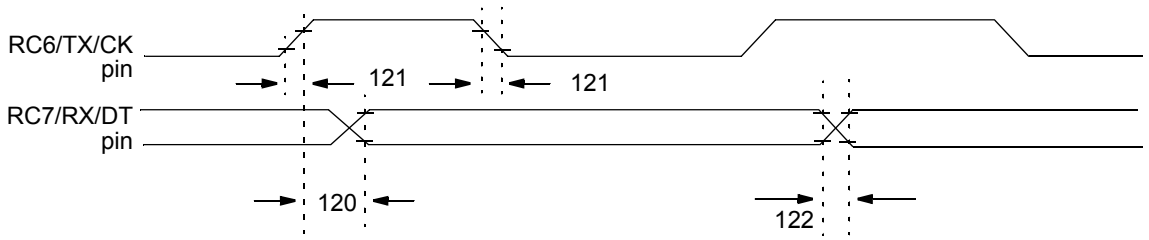

**Note 1:** If the A/D clock source is selected as RC, a time of TCY is added before the A/D clock starts. This allows the SLEEP instruction to be executed.

**FIGURE 17-11: PIC16F882/883/884/886/887 A/D CONVERSION TIMING (SLEEP MODE)**




**Note 1:** If the A/D clock source is selected as RC, a time of TCY is added before the A/D clock starts. This allows the SLEEP instruction to be executed.

**FIGURE 17-12: EUSART SYNCHRONOUS TRANSMISSION (MASTER/SLAVE) TIMING**

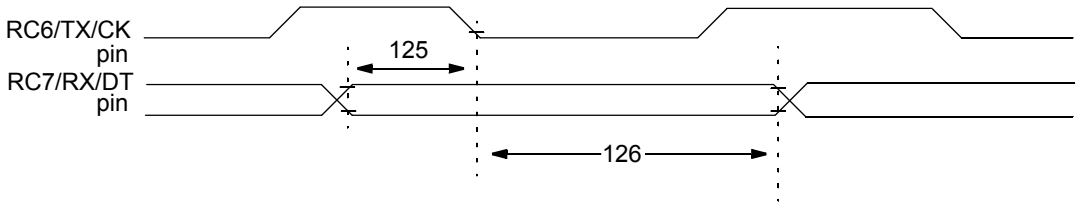



**Note:** Refer to Figure 17-3 for load conditions.

**TABLE 17-12: EUSART SYNCHRONOUS TRANSMISSION REQUIREMENTS**

| **Standard Operating Conditions (unless otherwise stated)**<br>Operating Temperature -40°C ≤ TA ≤ +125°C | | | | | | |
|---|---|---|---|---|---|---|
| **Param. No.** | **Symbol** | **Characteristic** | **Min.** | **Max.** | **Units** | **Conditions** |
| 120 | TCKH2DTV | SYNC XMIT (Master & Slave)<br>Clock high to data-out valid | — | 40 | ns | |
| 121 | TCKRF | Clock out rise time and fall time (Master mode) | — | 20 | ns | |
| 122 | TDTRF | Data-out rise time and fall time | — | 20 | ns | |

**FIGURE 17-13: EUSART SYNCHRONOUS RECEIVE (MASTER/SLAVE) TIMING**



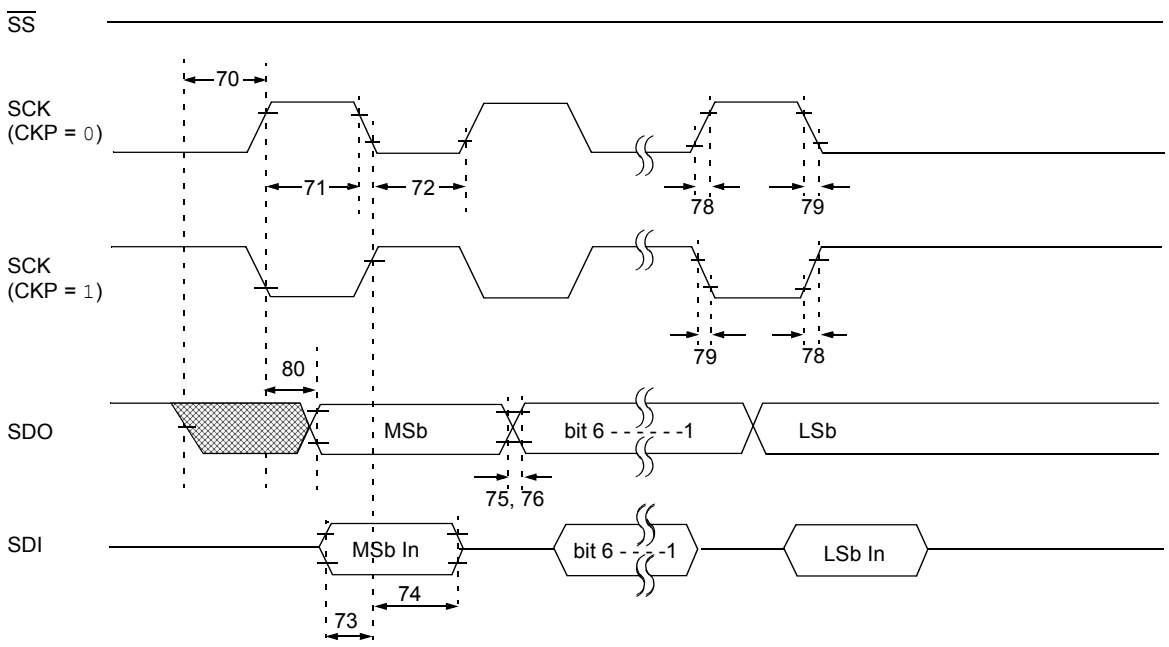

**Note:** Refer to Figure 17-3 for load conditions.

**TABLE 17-13: EUSART SYNCHRONOUS RECEIVE REQUIREMENTS**

| **Standard Operating Conditions (unless otherwise stated)**<br>Operating Temperature -40°C ≤ TA ≤ +125°C | | | | | | |
|---|---|---|---|---|---|---|
| **Param. No.** | **Symbol** | **Characteristic** | **Min.** | **Max.** | **Units** | **Conditions** |
| 125 | TDTV2CKL | SYNC RCV (Master & Slave)<br>Data-hold before CK ↓ (DT hold time) | 10 | — | ns | |
| 126 | TCKL2DTL | Data-hold after CK ↓ (DT hold time) | 15 | — | ns | |

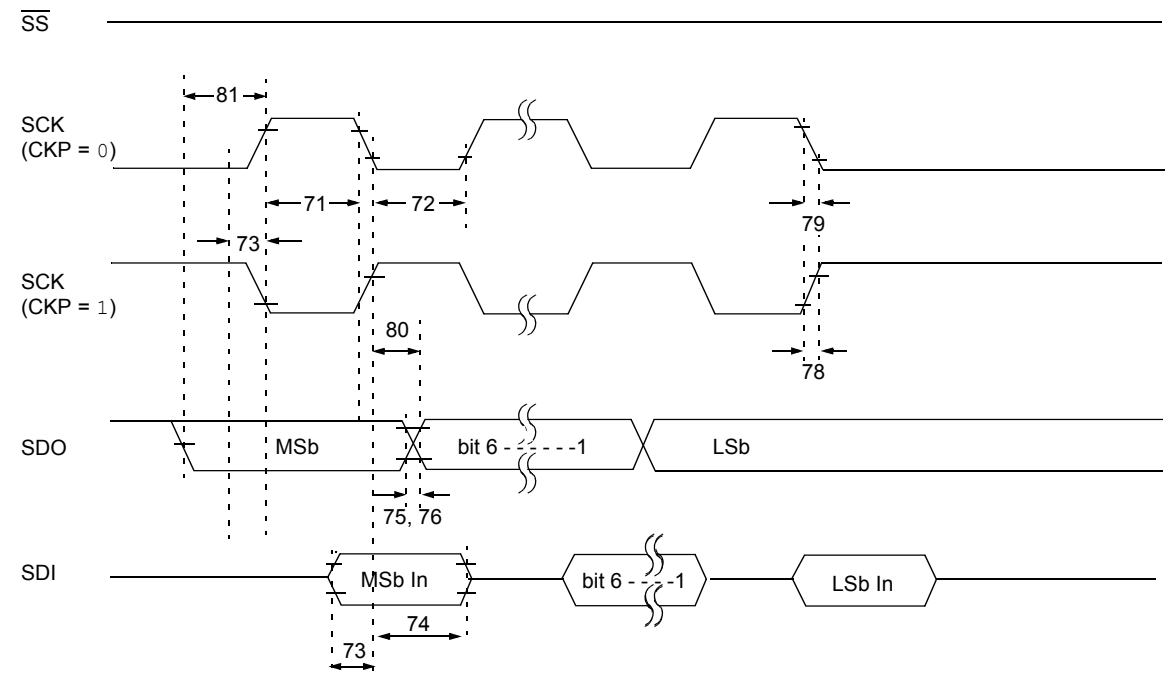

© 2006-2012 Microchip Technology Inc.

**FIGURE 17-14: SPI MASTER MODE TIMING (CKE = 0, SMP = 0)**



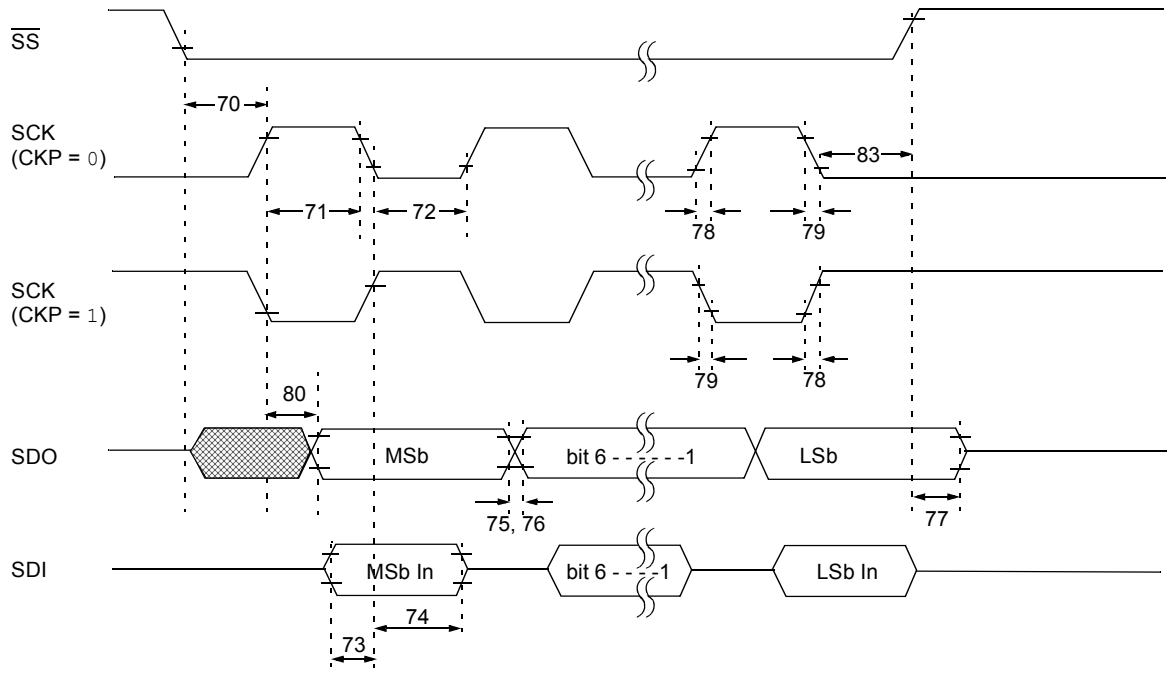

**Note:** Refer to Figure 17-3 for load conditions.

**FIGURE 17-15: SPI MASTER MODE TIMING (CKE = 1, SMP = 1)**



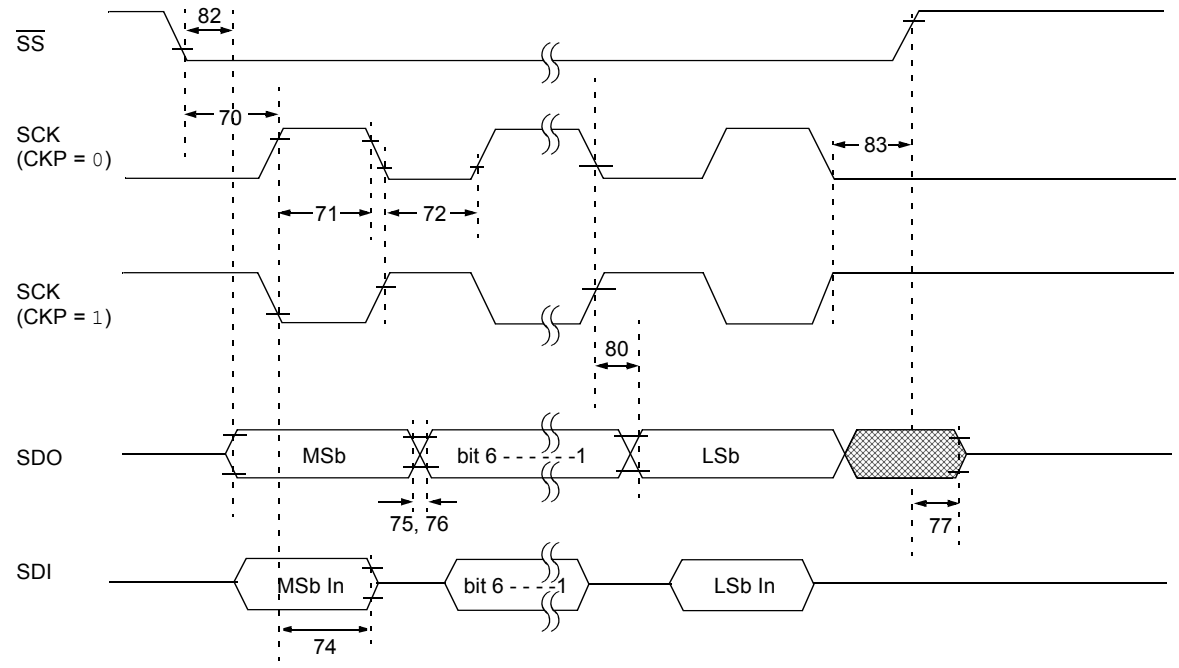

**Note:** Refer to Figure 17-3 for load conditions.

**FIGURE 17-16: SPI SLAVE MODE TIMING (CKE = 0)**




**Note:** Refer to Figure 17-3 for load conditions.

**FIGURE 17-17: SPI SLAVE MODE TIMING (CKE = 1)**




**Note:** Refer to Figure 17-3 for load conditions.


© 2006-2012 Microchip Technology Inc.

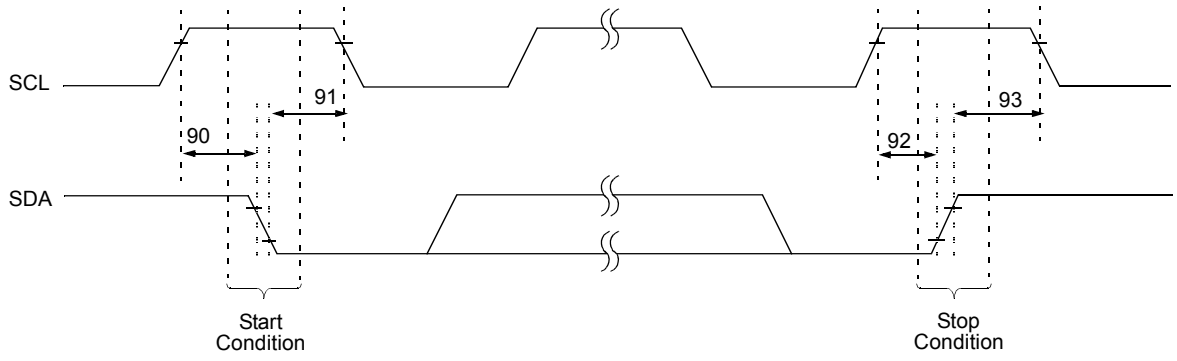
**TABLE 17-14: SPI MODE REQUIREMENTS**

| Param No. | Symbol | Characteristic | | Min. | Typ† | Max. | Units | Conditions |
|---|---|---|---|---|---|---|---|---|
| 70* | TssL2scH, TssL2scL | $\overline{SS}$↓ to SCK↓ or SCK↑ input | | Tcy | — | — | ns | |
| 71* | TscH | SCK input high time (Slave mode) | | Tcy + 20 | — | — | ns | |
| 72* | TscL | SCK input low time (Slave mode) | | Tcy + 20 | — | — | ns | |
| 73* | TdiV2scH, TdiV2scL | Setup time of SDI data input to SCK edge | | 100 | — | — | ns | |
| 74* | TscH2diL, TscL2diL | Hold time of SDI data input to SCK edge | | 100 | — | — | ns | |
| 75* | TdoR | SDO data output rise time | 3.0-5.5V | — | 10 | 25 | ns | |
| | | | 2.0-5.5V | — | 25 | 50 | ns | |
| 76* | TdoF | SDO data output fall time | | — | 10 | 25 | ns | |
| 77* | TssH2doZ | $\overline{SS}$↑ to SDO output high-impedance | | 10 | — | 50 | ns | |
| 78* | TscR | SCK output rise time (Master mode) | 3.0-5.5V | — | 10 | 25 | ns | |
| | | | 2.0-5.5V | — | 25 | 50 | ns | |
| 79* | TscF | SCK output fall time (Master mode) | | — | 10 | 25 | ns | |
| 80* | TscH2doV, TscL2doV | SDO data output valid after SCK edge | 3.0-5.5V | — | — | 50 | ns | |
| | | | 2.0-5.5V | — | — | 145 | ns | |
| 81* | TdoV2scH, TdoV2scL | SDO data output setup to SCK edge | | Tcy | — | — | ns | |
| 82* | TssL2doV | SDO data output valid after $\overline{SS}$↓ edge | | — | — | 50 | ns | |
| 83* | TscH2ssH, TscL2ssH | $\overline{SS}$ ↑ after SCK edge | | 1.5Tcy + 40 | — | — | ns | |

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**FIGURE 17-18: I²C™ BUS START/STOP BITS TIMING**



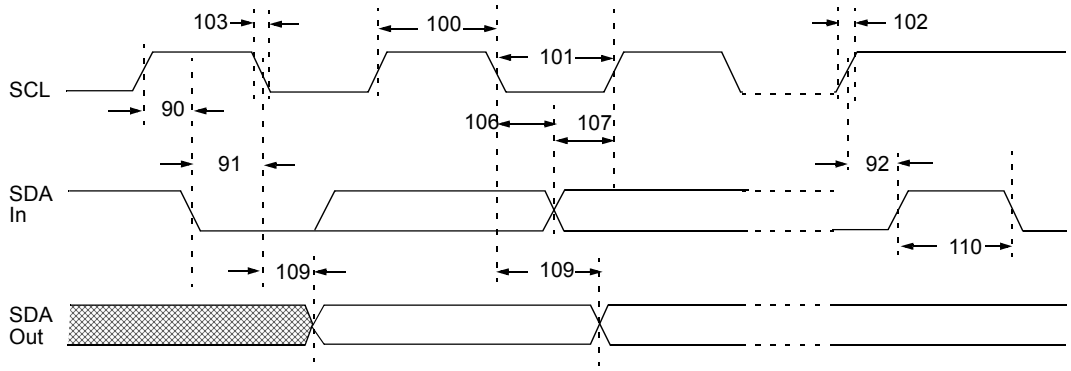

**Note**: Refer to Figure 17-3 for load conditions.

**TABLE 17-15: I²C™ BUS START/STOP BITS REQUIREMENTS**

| Param No. | Symbol | Characteristic | | Min. | Typ. | Max. | Units | Conditions |
|---|---|---|---|---|---|---|---|---|
| 90* | TSU:STA | Start condition Setup time | 100 kHz mode | 4700 | — | — | ns | Only relevant for Repeated Start condition |
| | | | 400 kHz mode | 600 | — | — | | |
| 91* | THD:STA | Start condition Hold time | 100 kHz mode | 4000 | — | — | ns | After this period, the first clock pulse is generated |
| | | | 400 kHz mode | 600 | — | — | | |
| 92* | TSU:STO | Stop condition Setup time | 100 kHz mode | 4700 | — | — | ns | |
| | | | 400 kHz mode | 600 | — | — | | |
| 93 | THD:STO | Stop condition Hold time | 100 kHz mode | 4000 | — | — | ns | |
| | | | 400 kHz mode | 600 | — | — | | |

* These parameters are characterized but not tested.

**FIGURE 17-19: I²C™ BUS DATA TIMING**




**Note:** Refer to Figure 17-3 for load conditions.


© 2006-2012 Microchip Technology Inc.

**TABLE 17-16: I²C™ BUS DATA REQUIREMENTS**

| Param. No. | Symbol | Characteristic | | Min. | Max. | Units | Conditions |
|---|---|---|---|---|---|---|---|
| 100* | THIGH | Clock high time | 100 kHz mode | 4.0 | — | µs | Device must operate at a minimum of 1.5 MHz |
| | | | 400 kHz mode | 0.6 | — | µs | Device must operate at a minimum of 10 MHz |
| | | | SSP Module | 1.5TCY | — | | |
| 101* | TLOW | Clock low time | 100 kHz mode | 4.7 | — | µs | Device must operate at a minimum of 1.5 MHz |
| | | | 400 kHz mode | 1.3 | — | µs | Device must operate at a minimum of 10 MHz |
| | | | SSP Module | 1.5TCY | — | | |
| 102* | TR | SDA and SCL rise time | 100 kHz mode | — | 1000 | ns | |
| | | | 400 kHz mode | 20 + 0.1CB | 300 | ns | CB is specified to be from 10-400 pF |
| 103* | TF | SDA and SCL fall time | 100 kHz mode | — | 300 | ns | |
| | | | 400 kHz mode | 20 + 0.1CB | 300 | ns | CB is specified to be from 10-400 pF |
| 90* | TSU:STA | Start condition setup time | 100 kHz mode | 4.7 | — | µs | Only relevant for Repeated Start condition |
| | | | 400 kHz mode | 0.6 | — | µs | |
| 91* | THD:STA | Start condition hold time | 100 kHz mode | 4.0 | — | µs | After this period the first clock pulse is generated |
| | | | 400 kHz mode | 0.6 | — | µs | |
| 106* | THD:DAT | Data input hold time | 100 kHz mode | 0 | — | ns | |
| | | | 400 kHz mode | 0 | 0.9 | µs | |
| 107* | TSU:DAT | Data input setup time | 100 kHz mode | 250 | — | ns | **(Note 2)** |
| | | | 400 kHz mode | 100 | — | ns | |
| 92* | TSU:STO | Stop condition setup time | 100 kHz mode | 4.7 | — | µs | |
| | | | 400 kHz mode | 0.6 | — | µs | |
| 109* | TAA | Output valid from clock | 100 kHz mode | — | 3500 | ns | **(Note 1)** |
| | | | 400 kHz mode | — | — | ns | |
| 110* | TBUF | Bus free time | 100 kHz mode | 4.7 | — | µs | Time the bus must be free before a new transmission can start |
| | | | 400 kHz mode | 1.3 | — | µs | |
| | CB | Bus capacitive loading | | — | 400 | pF | |

* These parameters are characterized but not tested.

**Note 1:** As a transmitter, the device must provide this internal minimum delay time to bridge the undefined region (min. 300 ns) of the falling edge of SCL to avoid unintended generation of Start or Stop conditions.

**2:** A Fast mode (400 kHz) I²C bus device can be used in a Standard mode (100 kHz) I²C bus system, but the requirement TSU:DAT ≥ 250 ns must then be met. This will automatically be the case if the device does not stretch the low period of the SCL signal. If such a device does stretch the low period of the SCL signal, it must output the next data bit to the SDA line TR max. + TSU:DAT = 1000 + 250 = 1250 ns (according to the Standard mode I²C bus specification), before the SCL line is released.

**NOTES:**


© 2006-2012 Microchip Technology Inc.

## 18.0 DC AND AC CHARACTERISTICS GRAPHS AND TABLES

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 18-1: TYPICAL IDD vs. FOSC OVER VDD (EC MODE)**

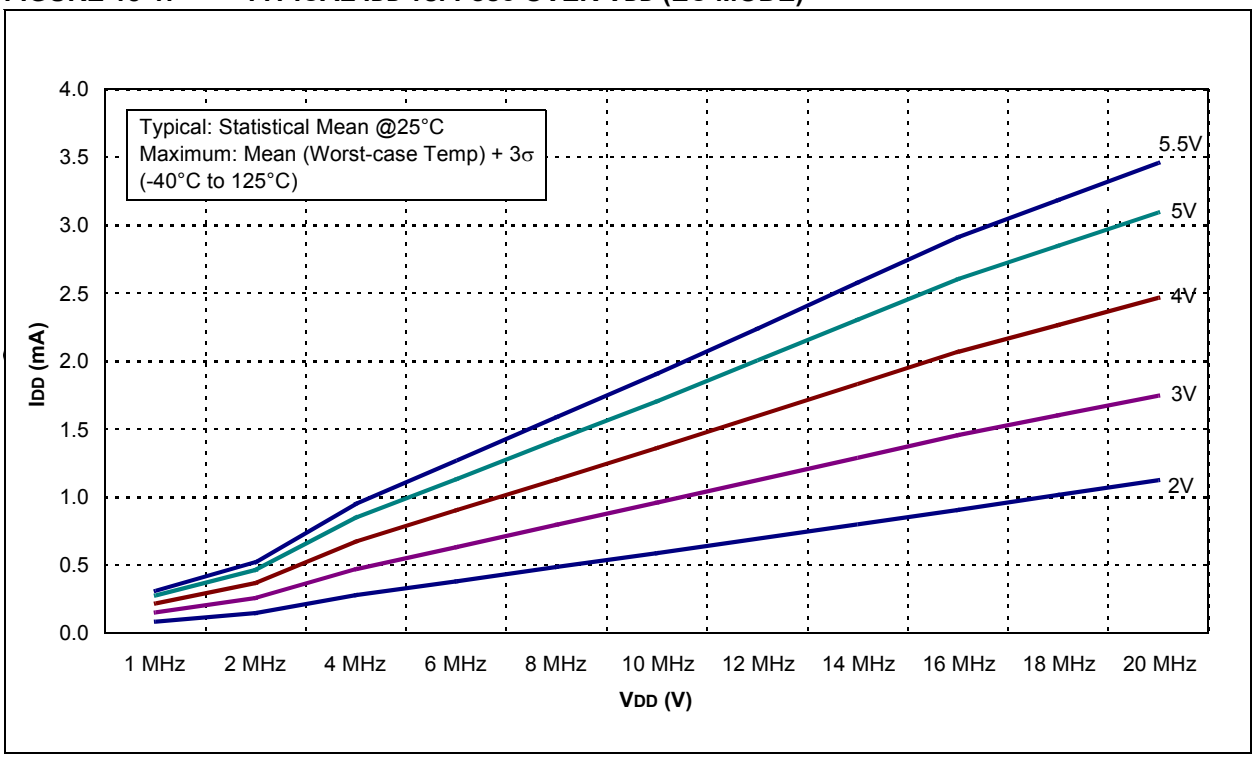
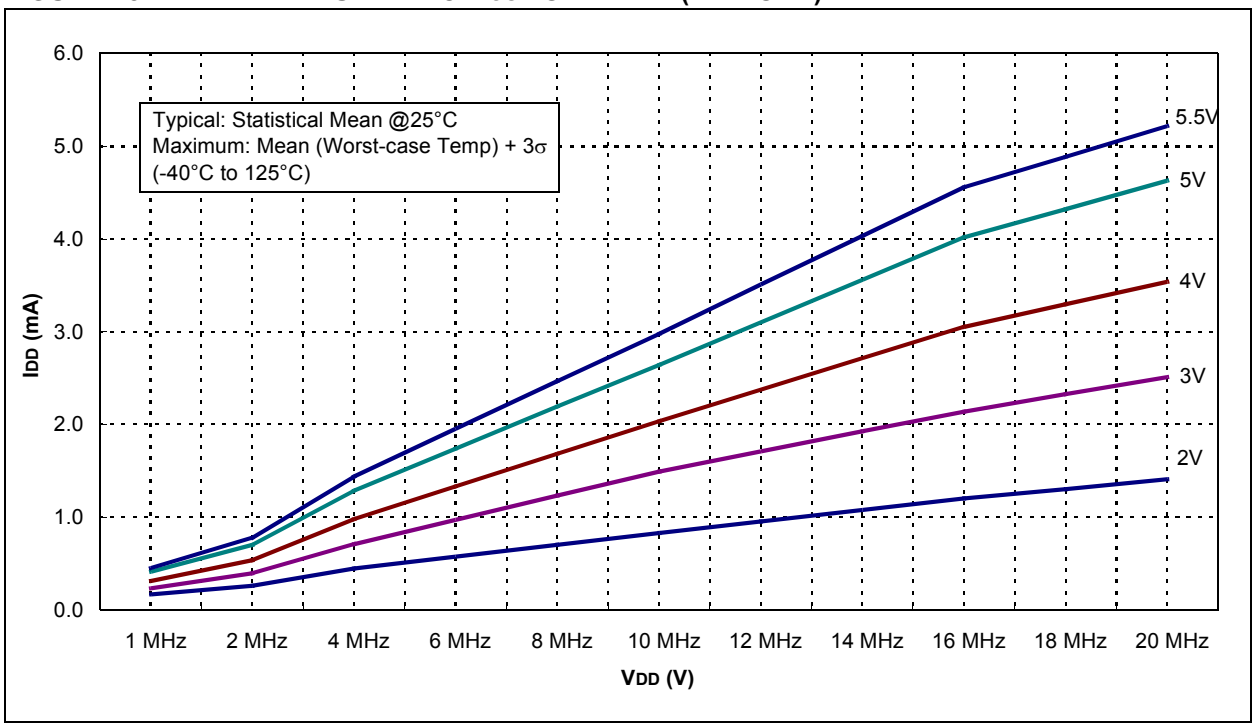



**FIGURE 18-2: MAXIMUM IDD vs. FOSC OVER VDD (EC MODE)**





© 2006-2012 Microchip Technology Inc.

**FIGURE 18-3: TYPICAL IDD vs. FOSC OVER VDD (HS MODE)**

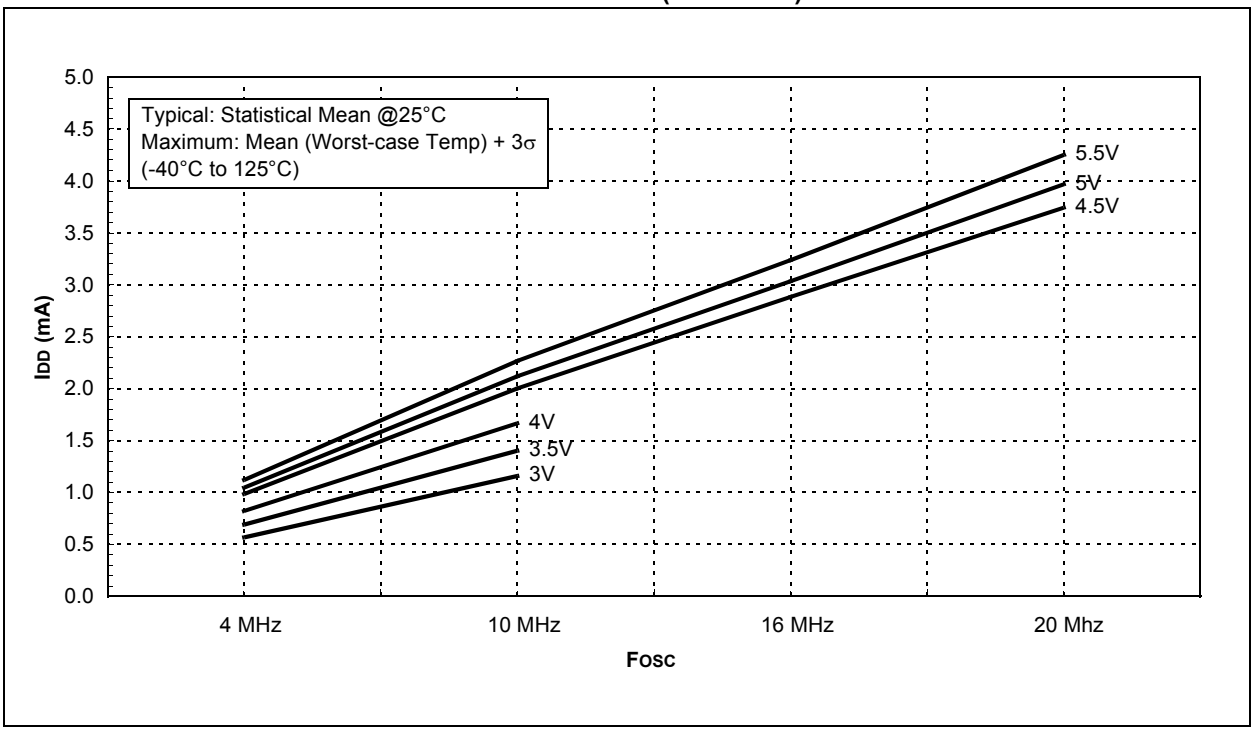



**FIGURE 18-4: MAXIMUM IDD vs. FOSC OVER VDD (HS MODE)**

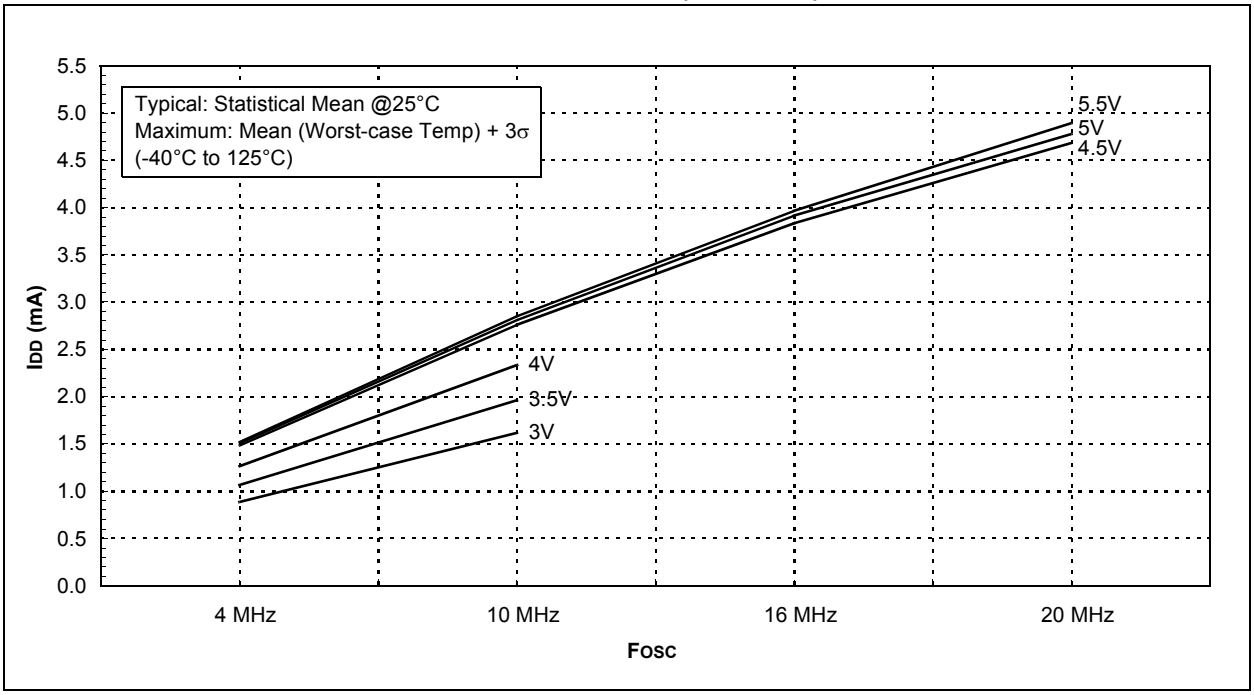


**FIGURE 18-5: TYPICAL IDD vs. VDD OVER FOSC (XT MODE)**

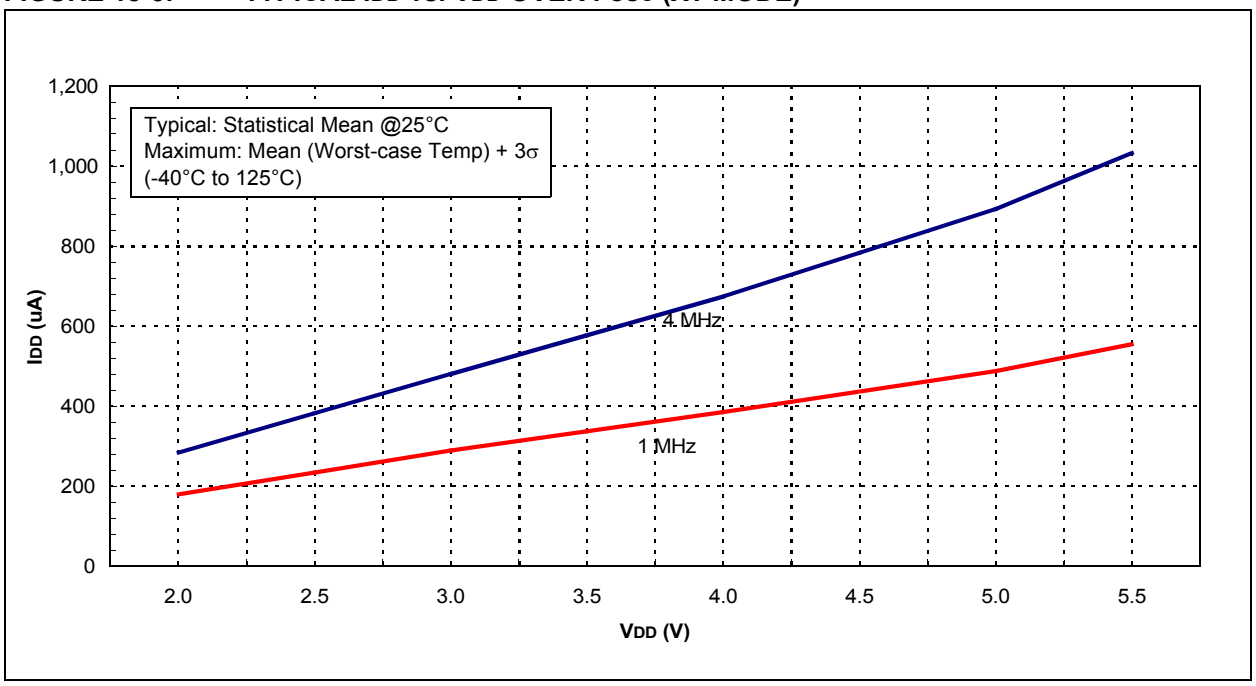
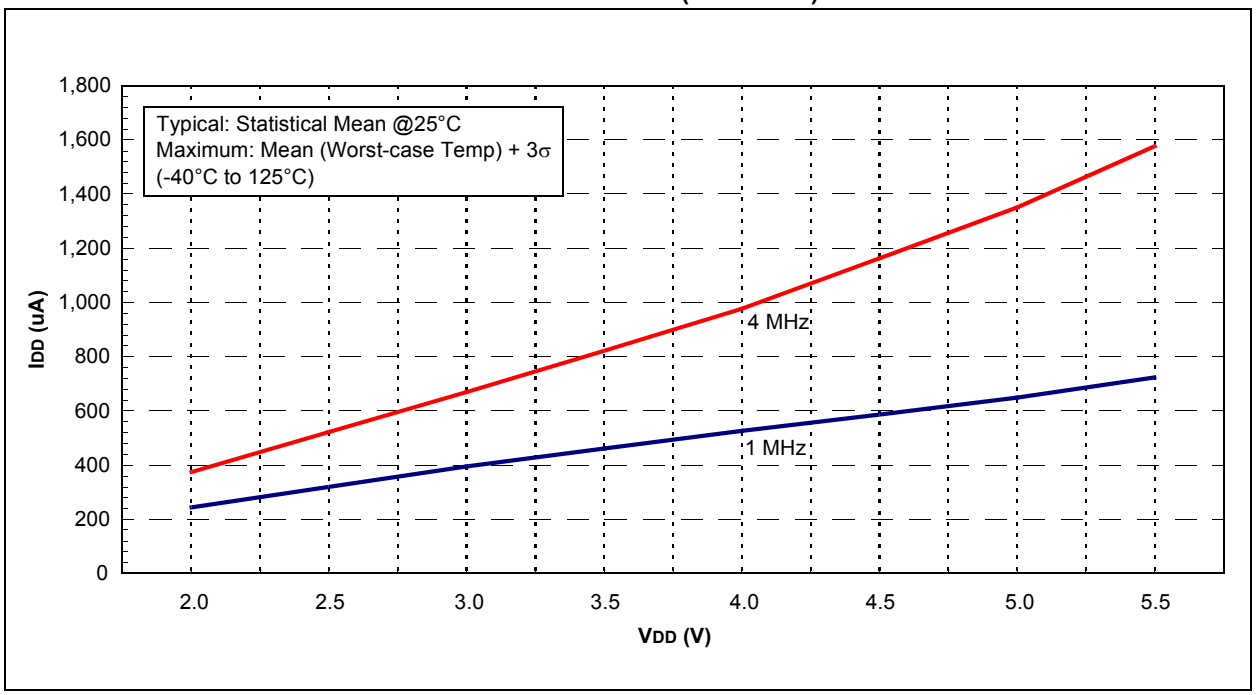



**FIGURE 18-6: MAXIMUM IDD vs. VDD OVER FOSC (XT MODE)**





© 2006-2012 Microchip Technology Inc.

**FIGURE 18-7: TYPICAL IDD vs. VDD OVER FOSC (EXTRC MODE)**

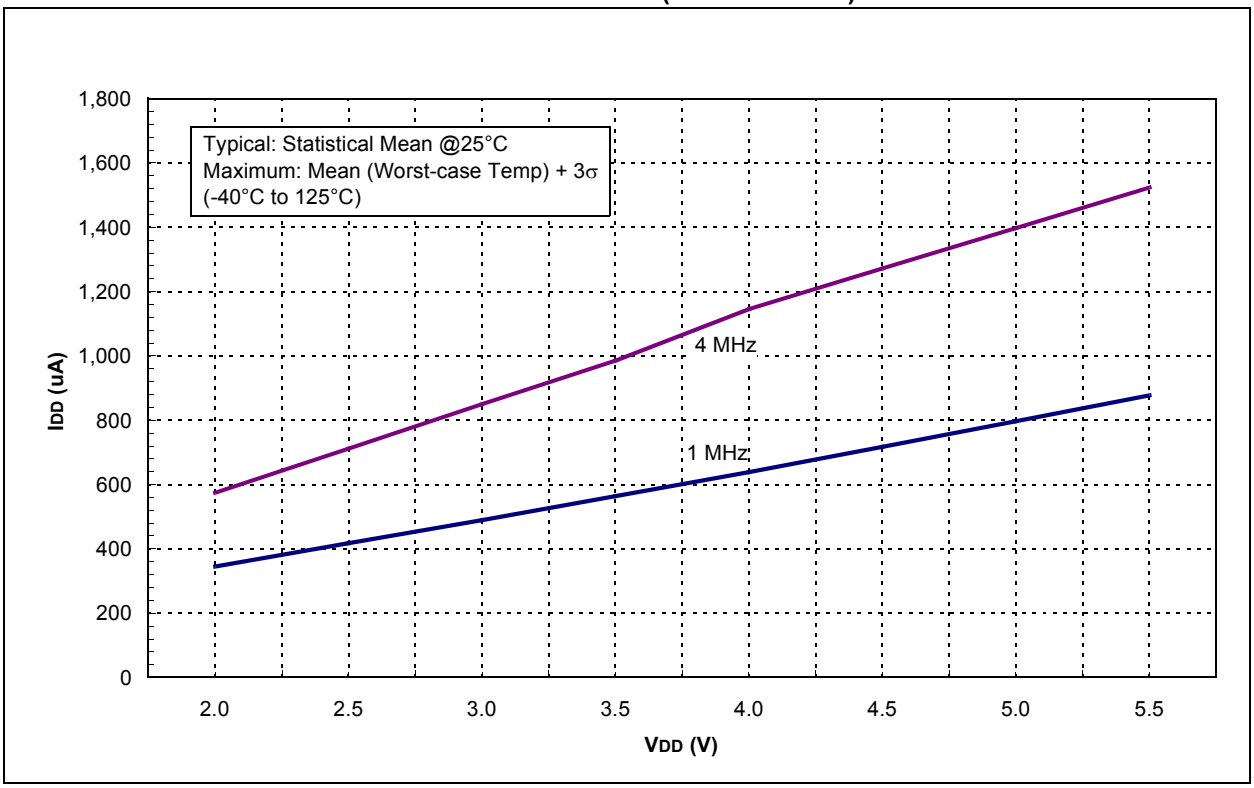



**FIGURE 18-8: MAXIMUM IDD vs. VDD (EXTRC MODE)**

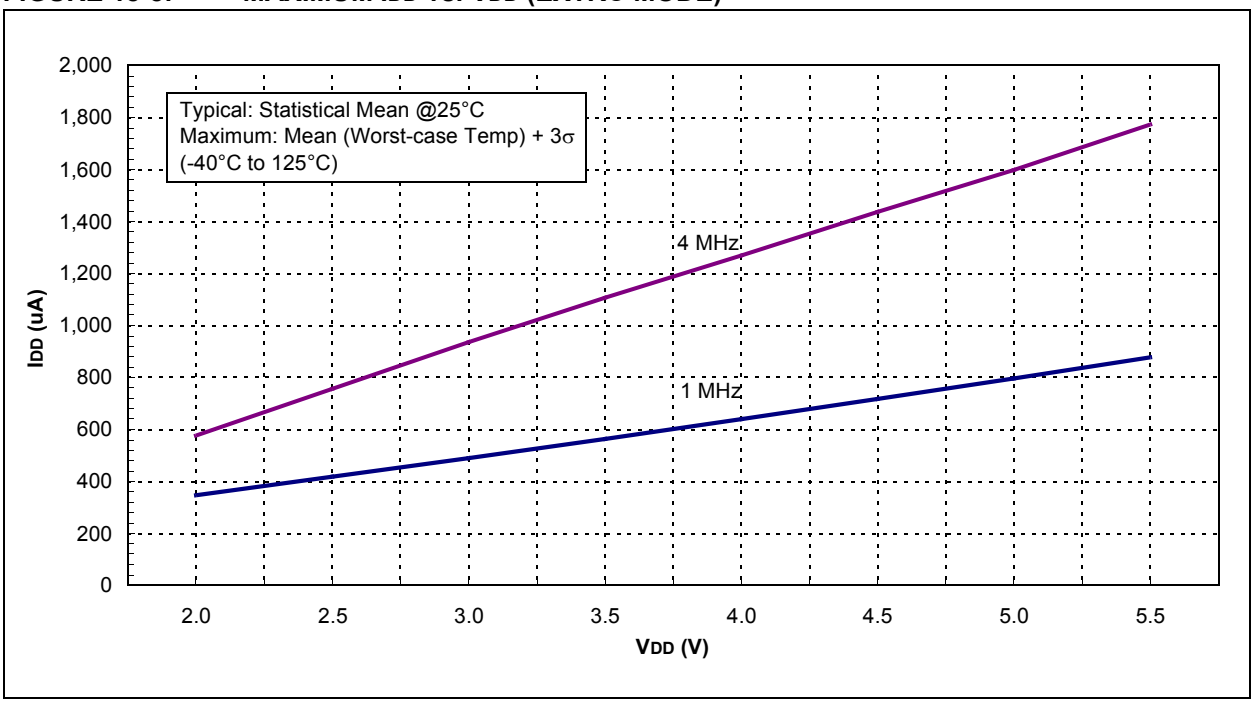


**FIGURE 18-9: IDD vs. VDD OVER FOSC (LFINTOSC MODE, 31 kHz)**

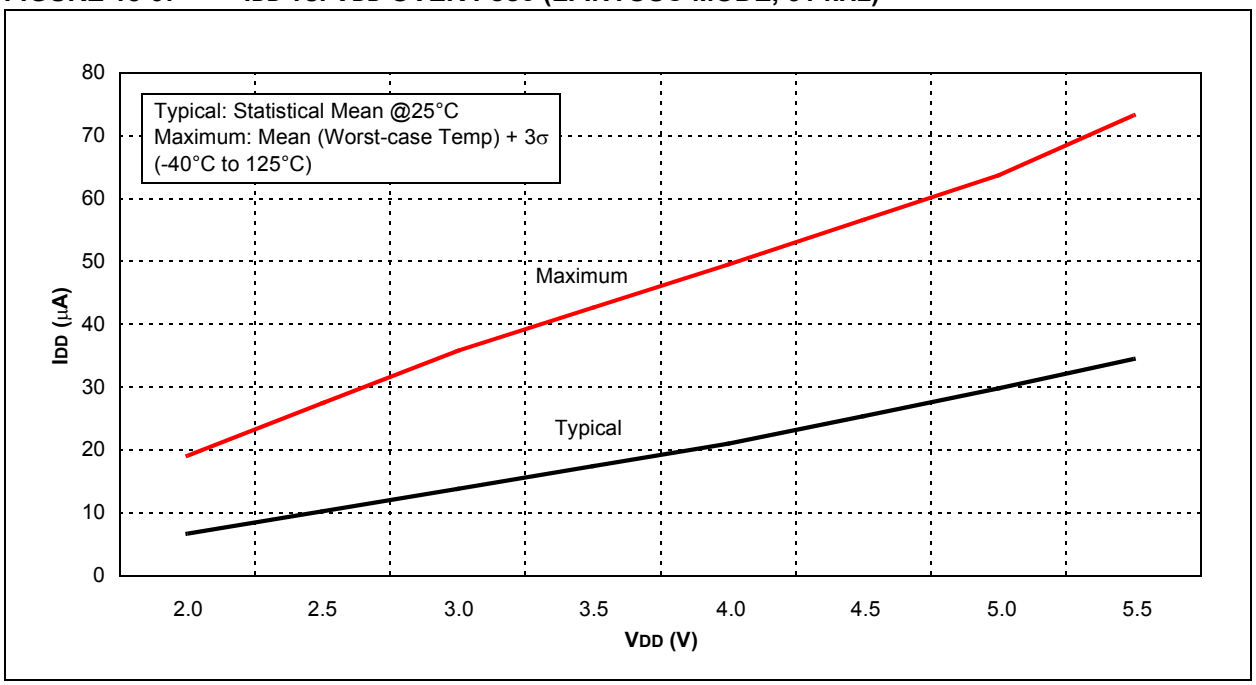
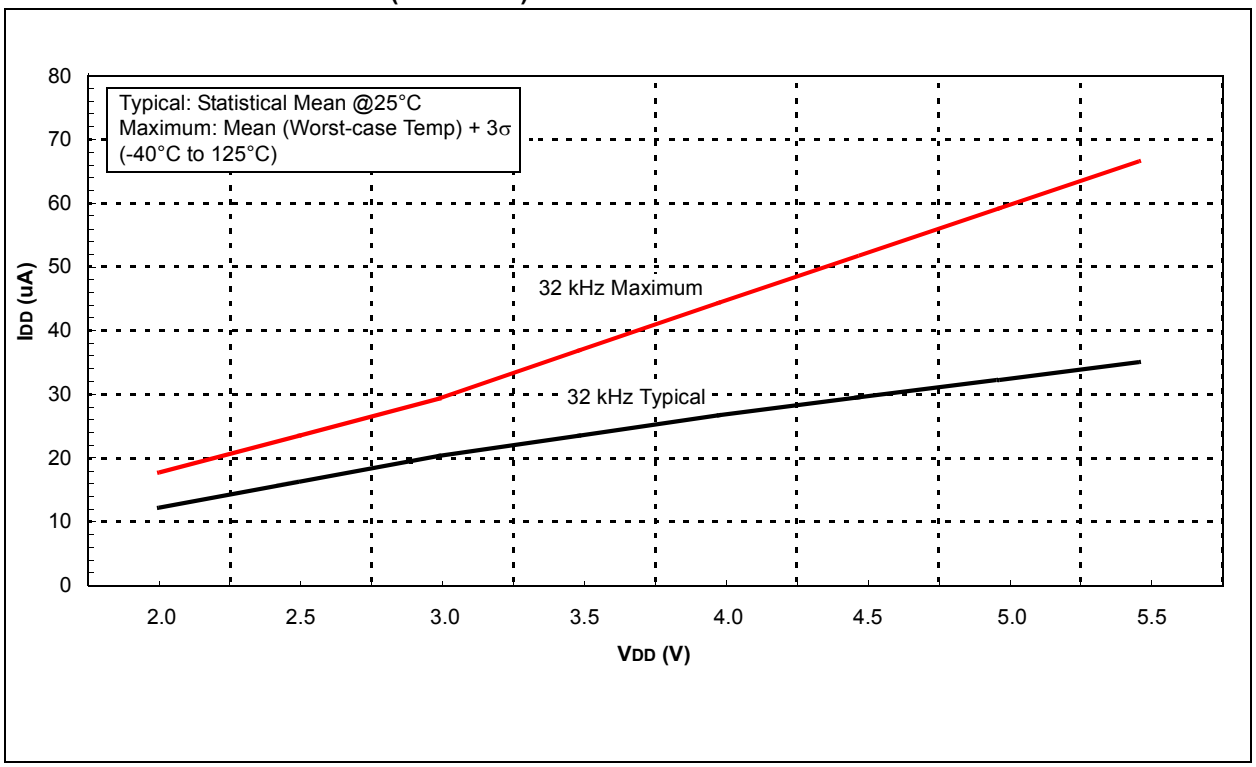



**FIGURE 18-10: IDD vs. VDD (LP MODE)**





© 2006-2012 Microchip Technology Inc.

**FIGURE 18-11: TYPICAL IDD vs. FOSC OVER VDD (HFINTOSC MODE)**

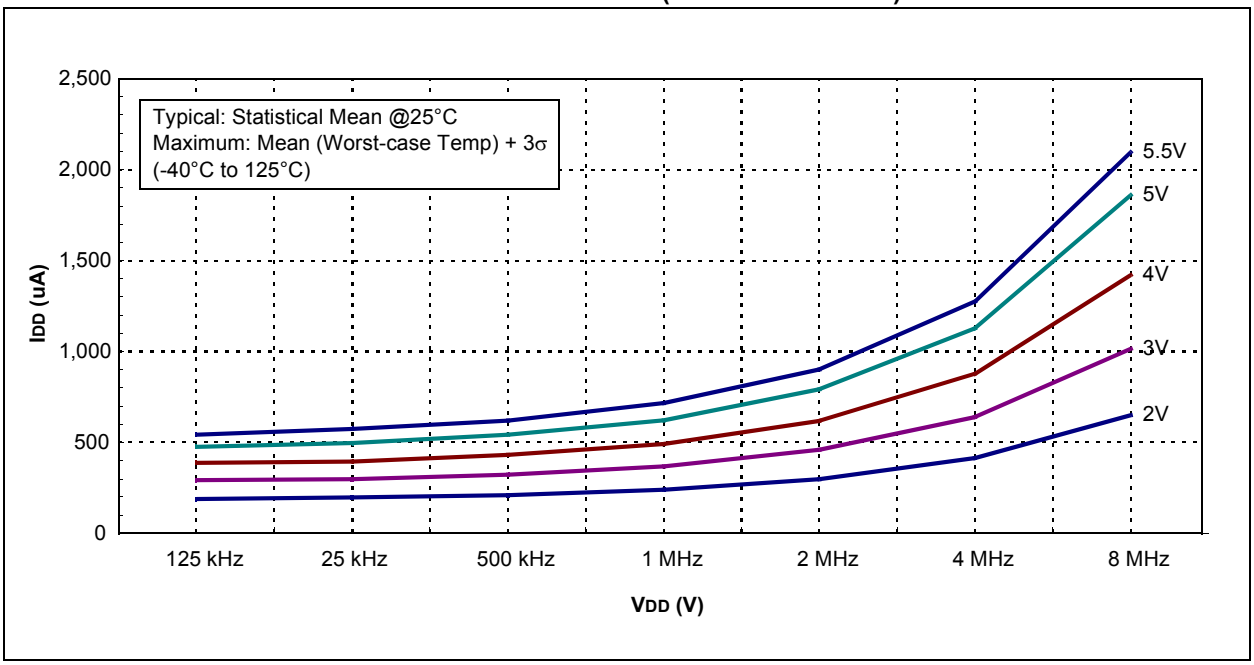



**FIGURE 18-12: MAXIMUM IDD vs. FOSC OVER VDD (HFINTOSC MODE)**

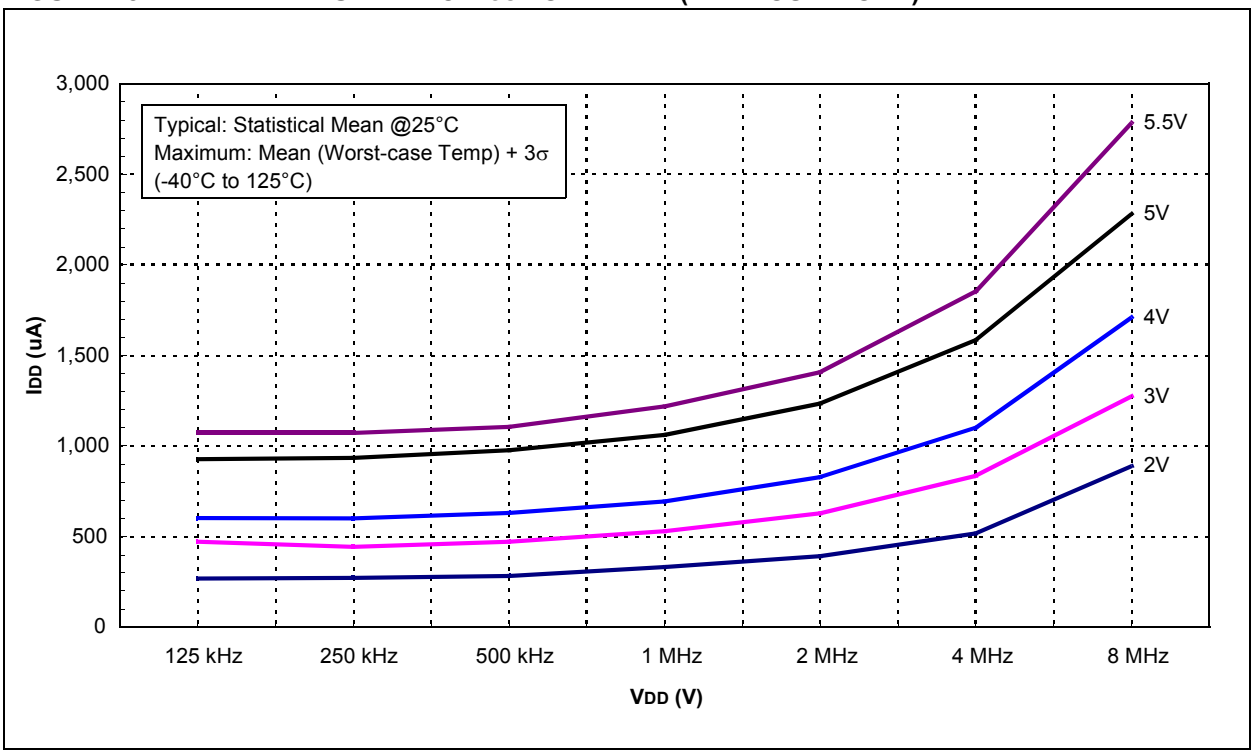


**FIGURE 18-13: TYPICAL IPD vs. VDD (SLEEP MODE, ALL PERIPHERALS DISABLED)**

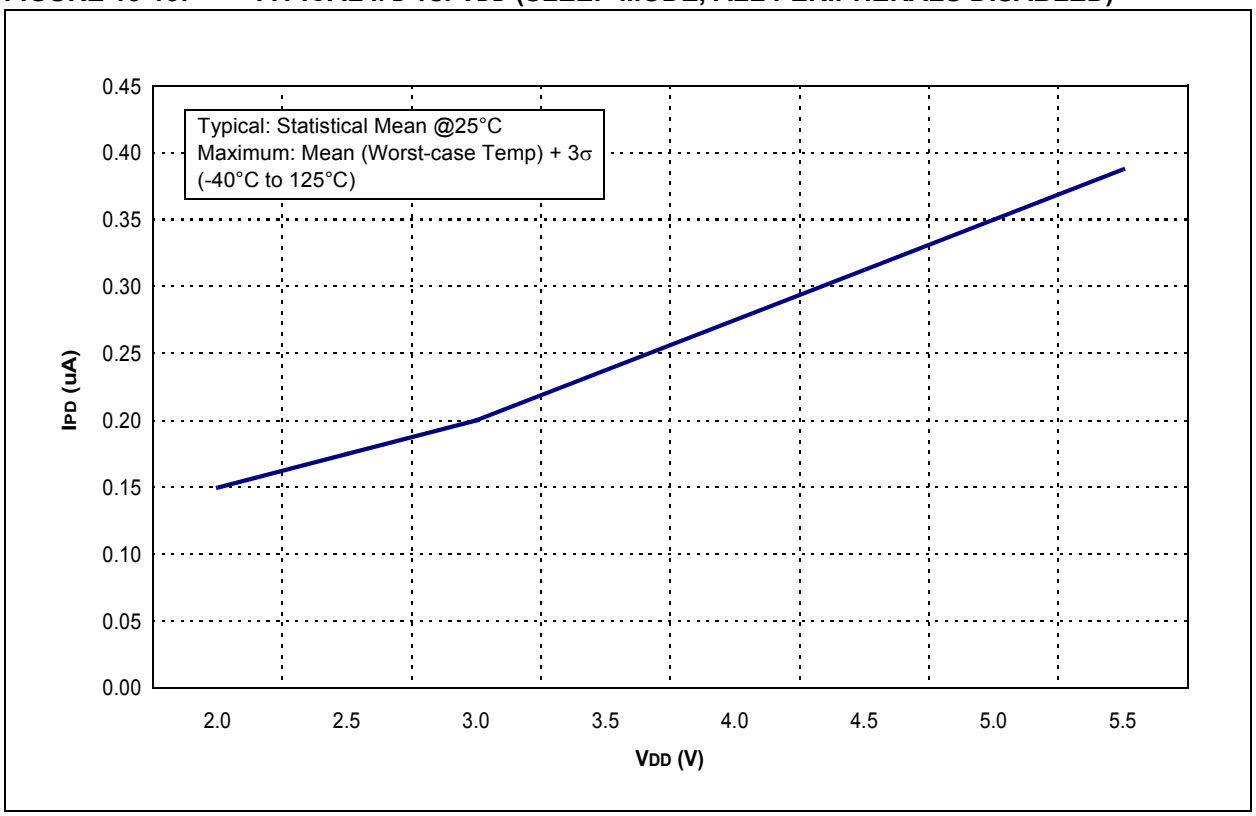
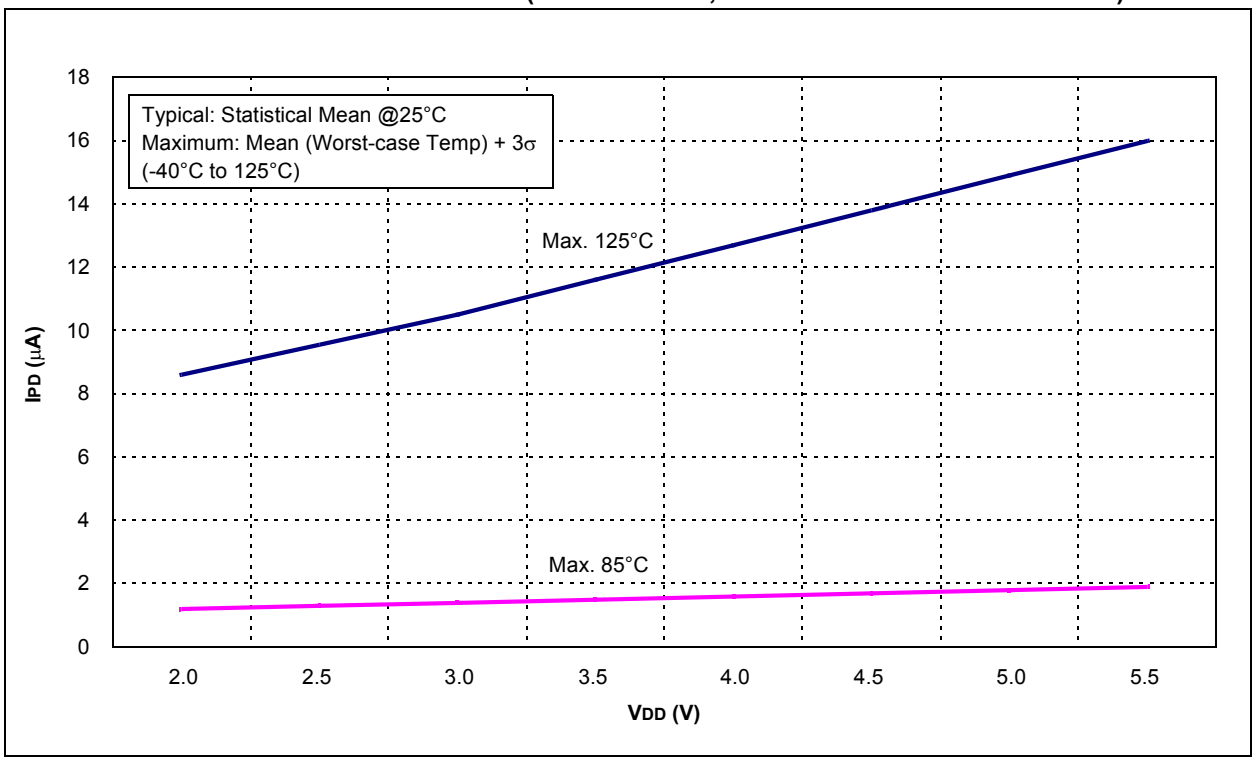



**FIGURE 18-14: MAXIMUM IPD vs. VDD (SLEEP MODE, ALL PERIPHERALS DISABLED)**





© 2006-2012 Microchip Technology Inc.

**FIGURE 18-15: COMPARATOR IPD vs. VDD (BOTH COMPARATORS ENABLED)**

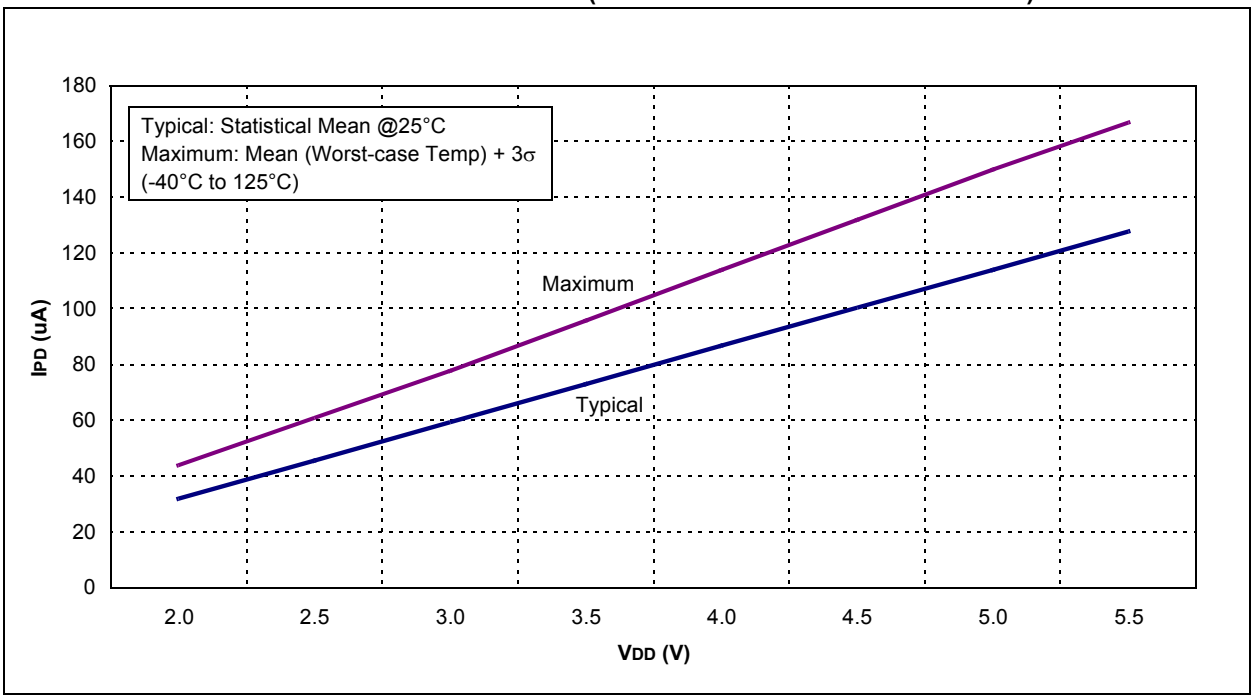



**FIGURE 18-16: BOR IPD vs. VDD OVER TEMPERATURE**

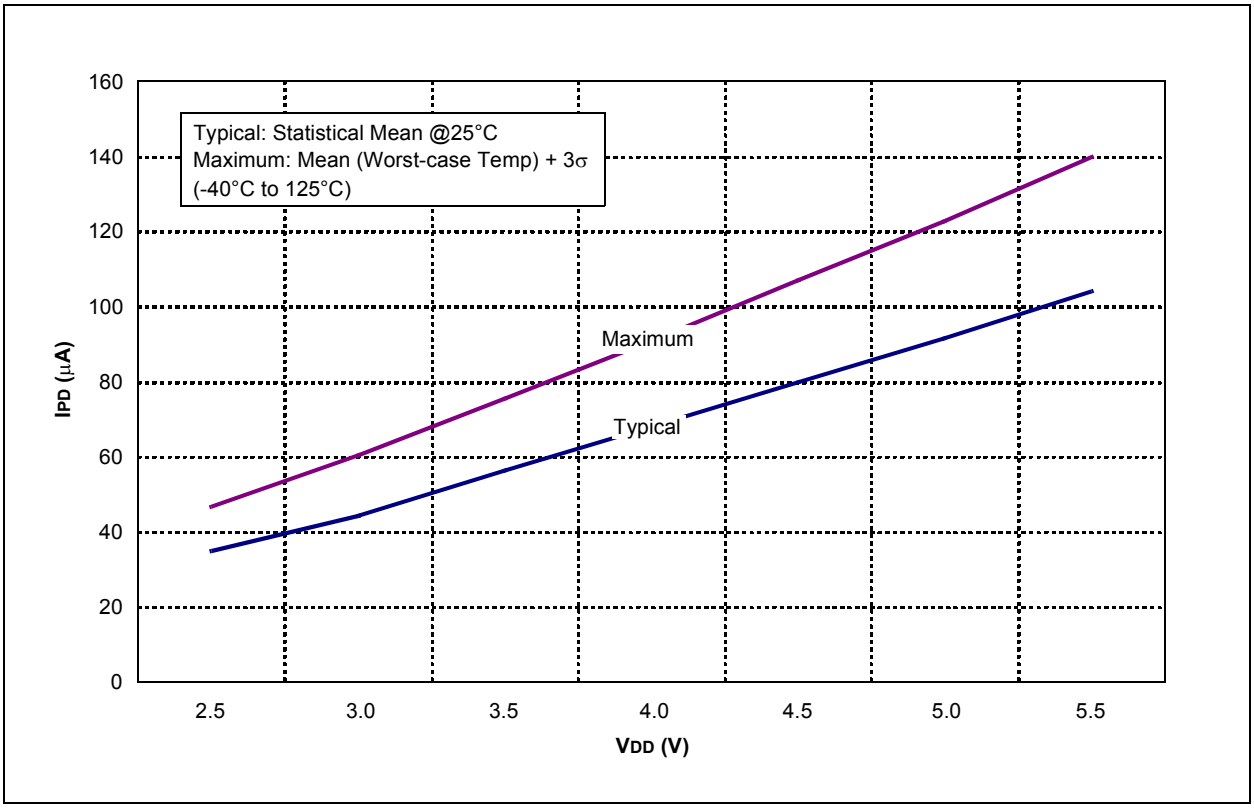


**FIGURE 18-17: TYPICAL WDT IPD vs. VDD (25°C)**

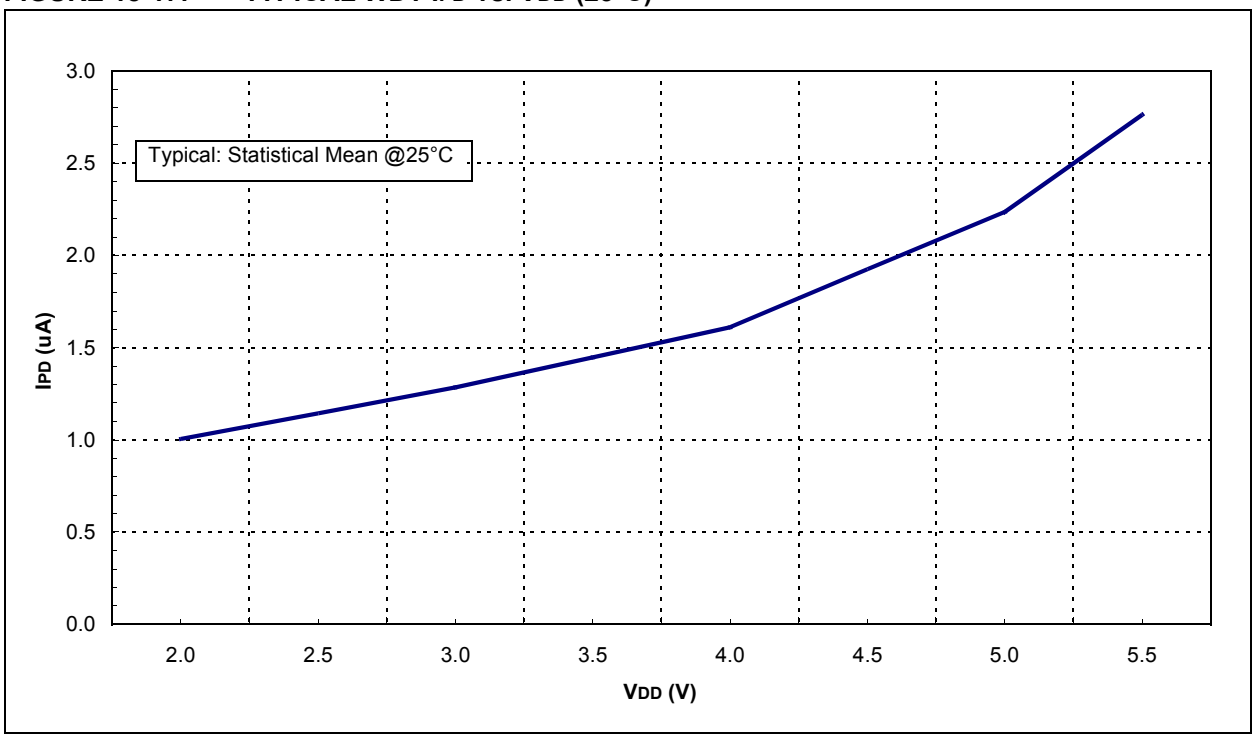
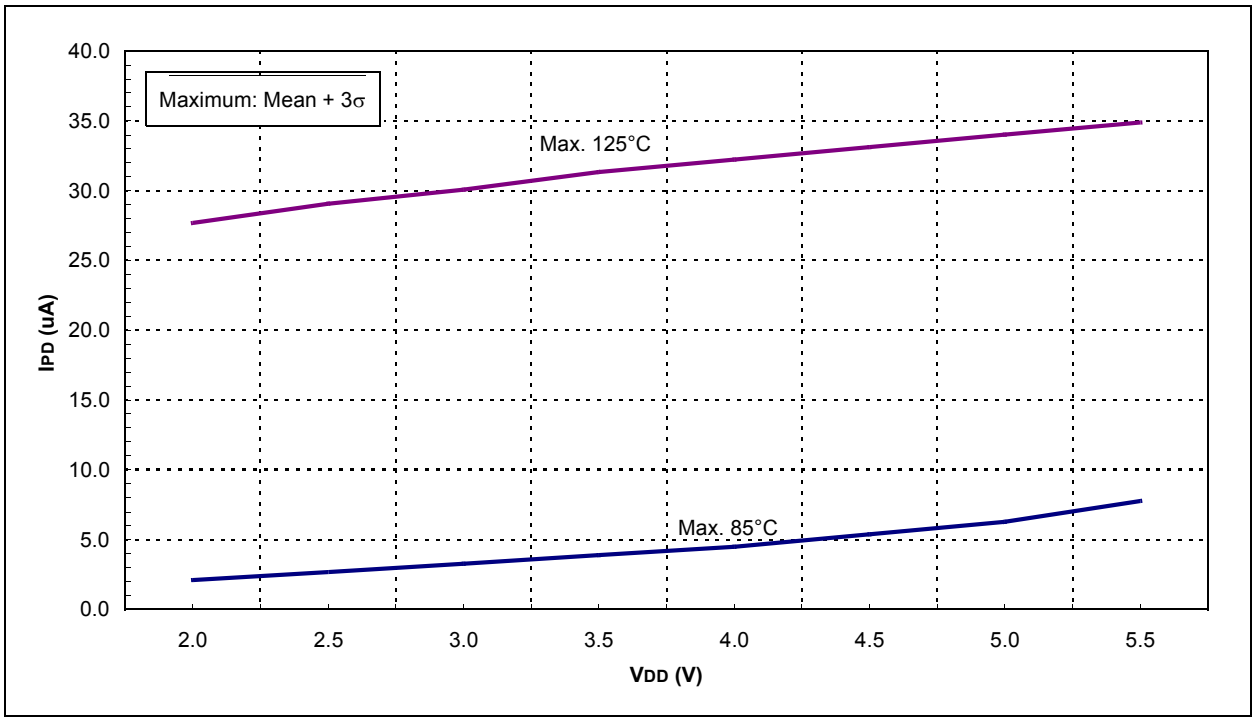



**FIGURE 18-18: MAXIMUM WDT IPD vs. VDD OVER TEMPERATURE**





© 2006-2012 Microchip Technology Inc.

**FIGURE 18-19: WDT PERIOD vs. VDD OVER TEMPERATURE**

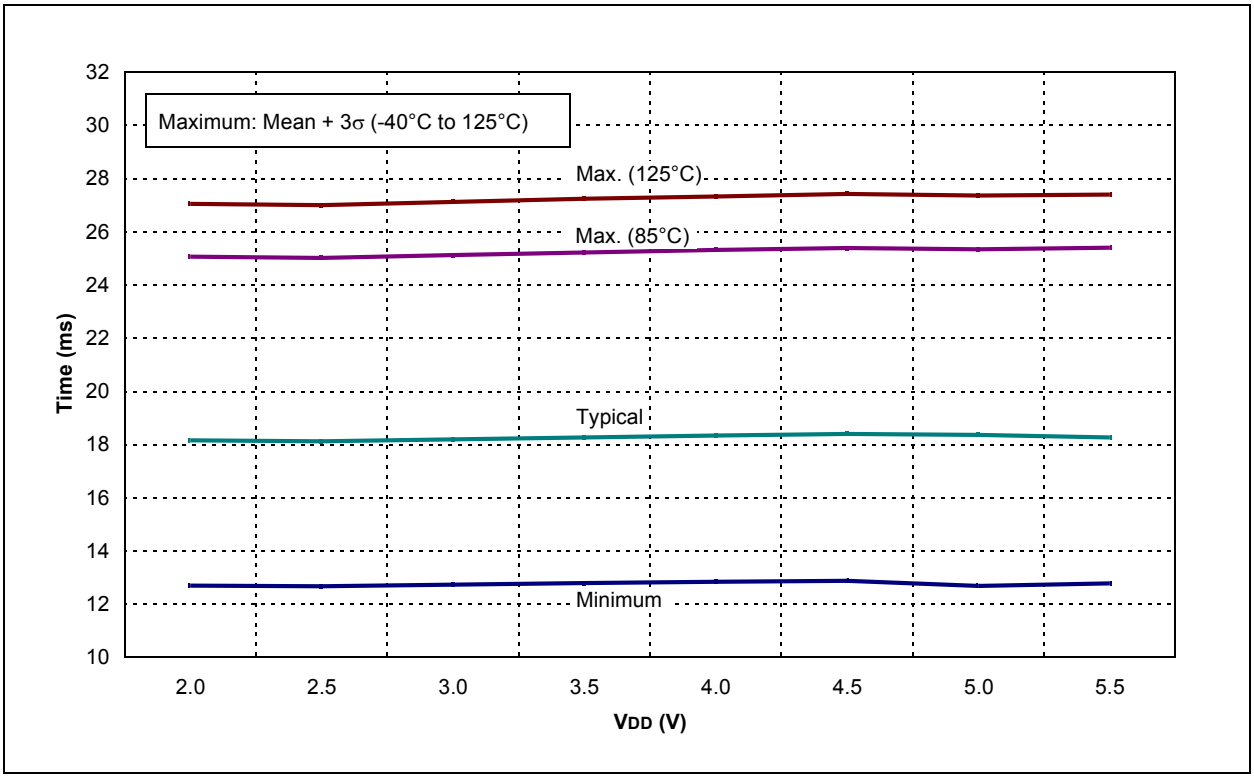



**FIGURE 18-20: WDT PERIOD vs. TEMPERATURE (VDD = 5.0V)**

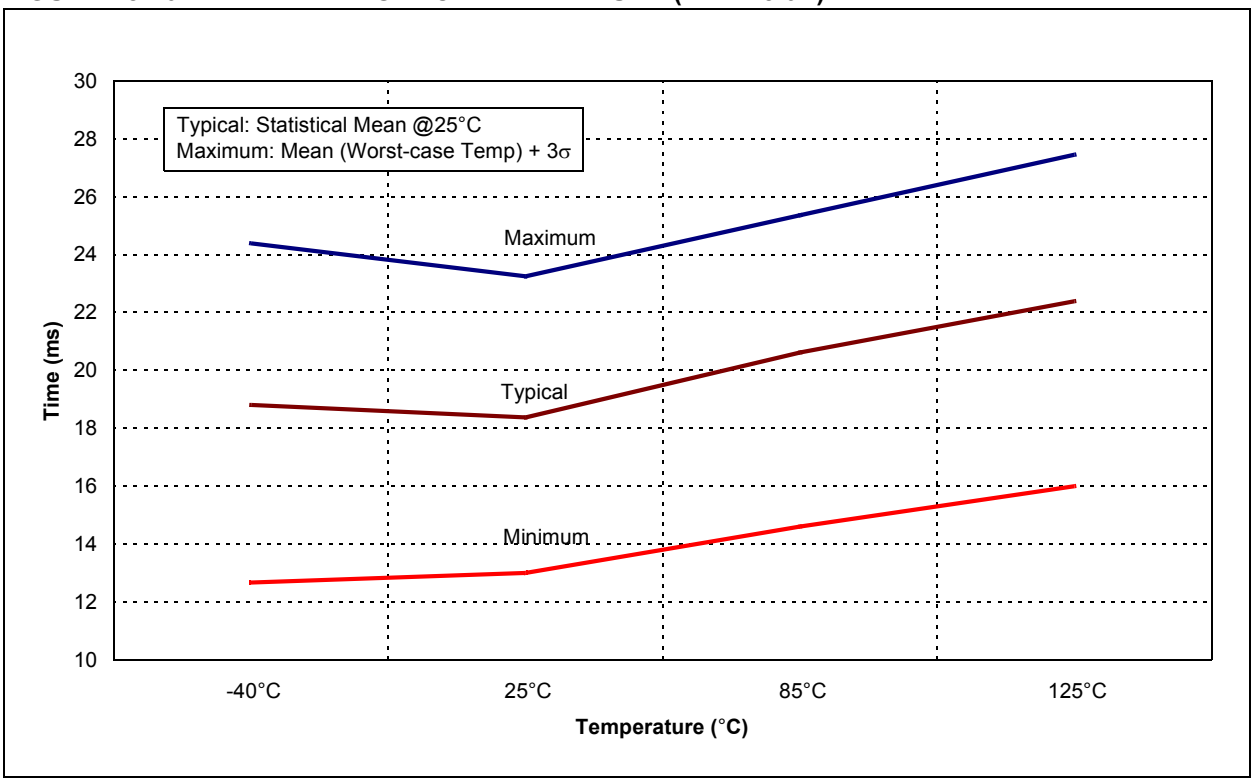


**FIGURE 18-21: CVREF IPD vs. VDD OVER TEMPERATURE (HIGH RANGE)**

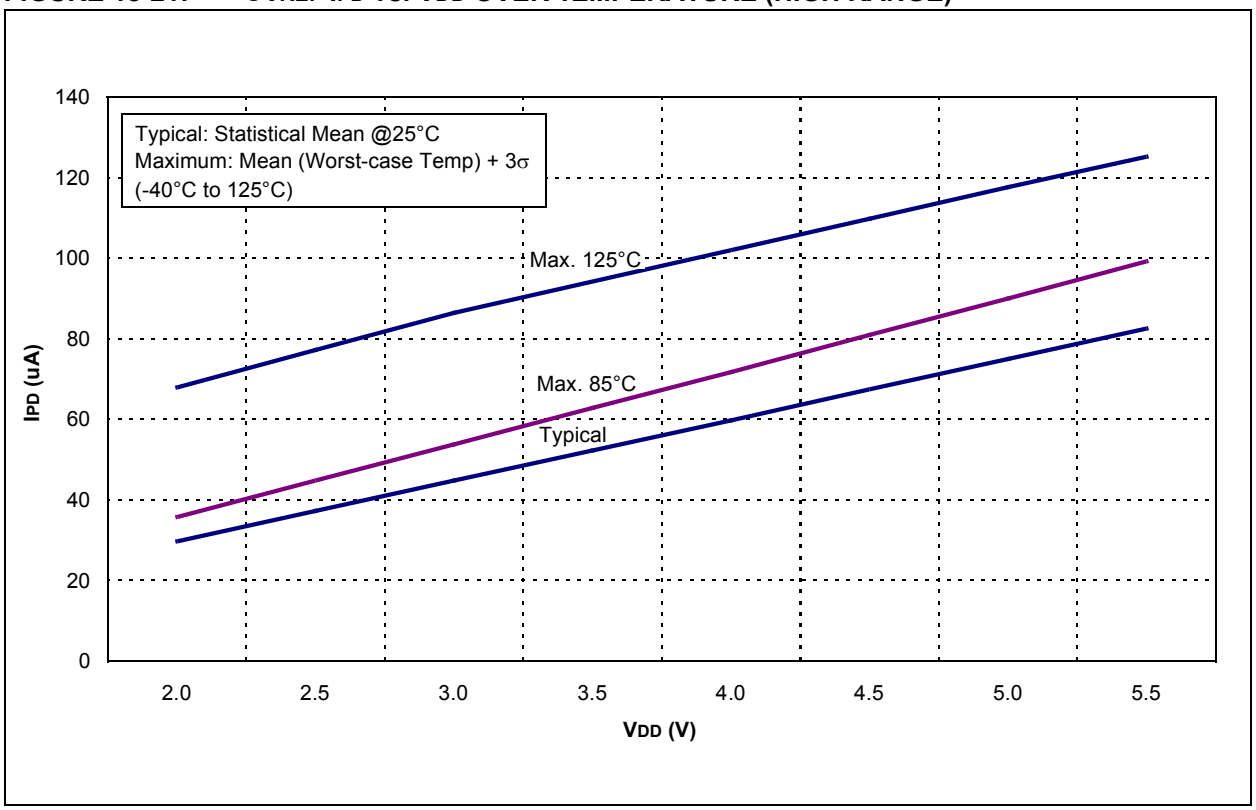
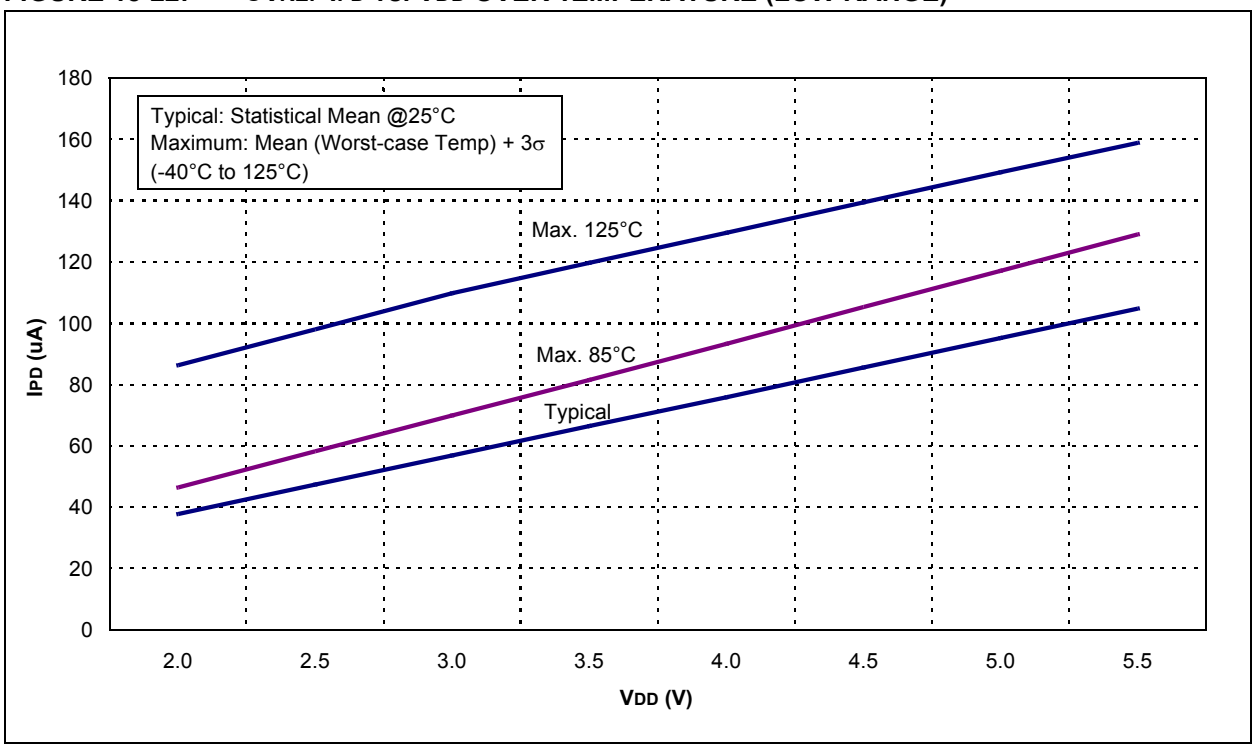



**FIGURE 18-22: CVREF IPD vs. VDD OVER TEMPERATURE (LOW RANGE)**





© 2006-2012 Microchip Technology Inc.

**FIGURE 18-23: TYPICAL VP6 REFERENCE IPD vs. VDD (25°C)**

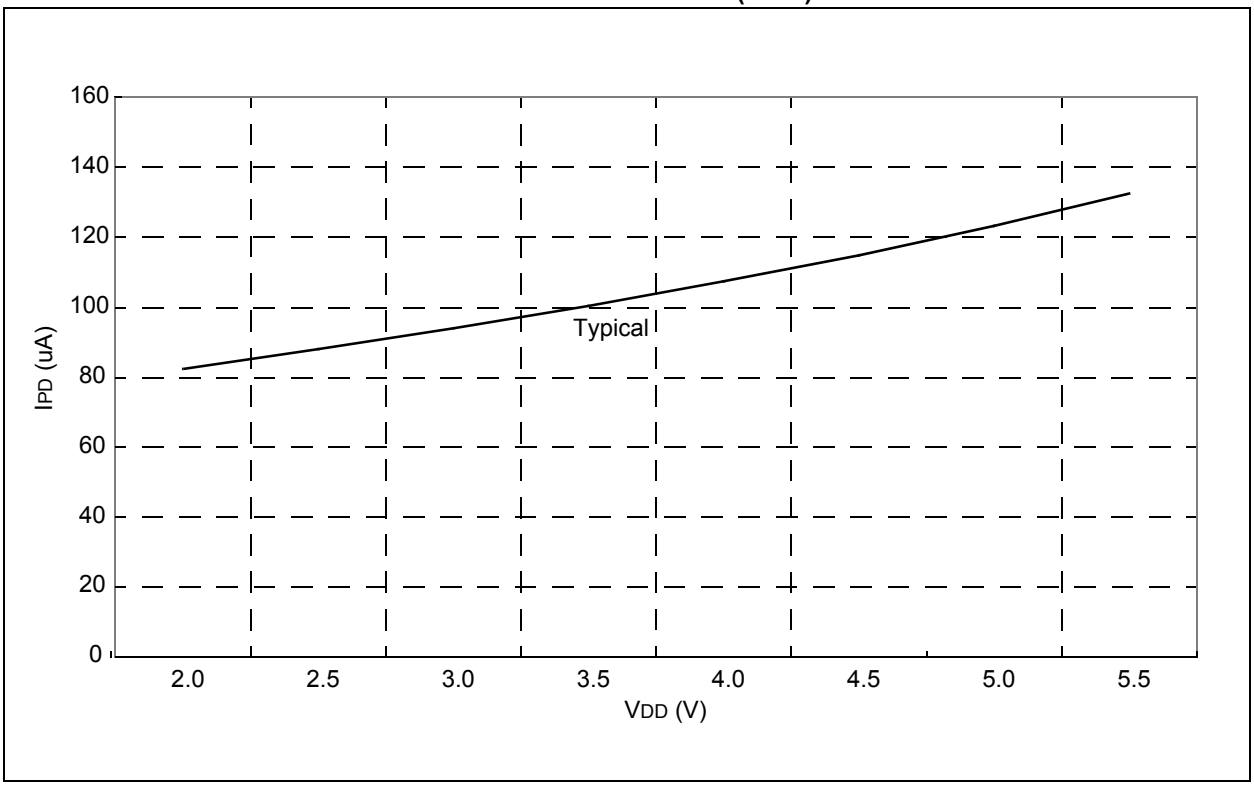



**FIGURE 18-24: MAXIMUM VP6 REFERENCE IPD vs. VDD OVER TEMPERATURE**

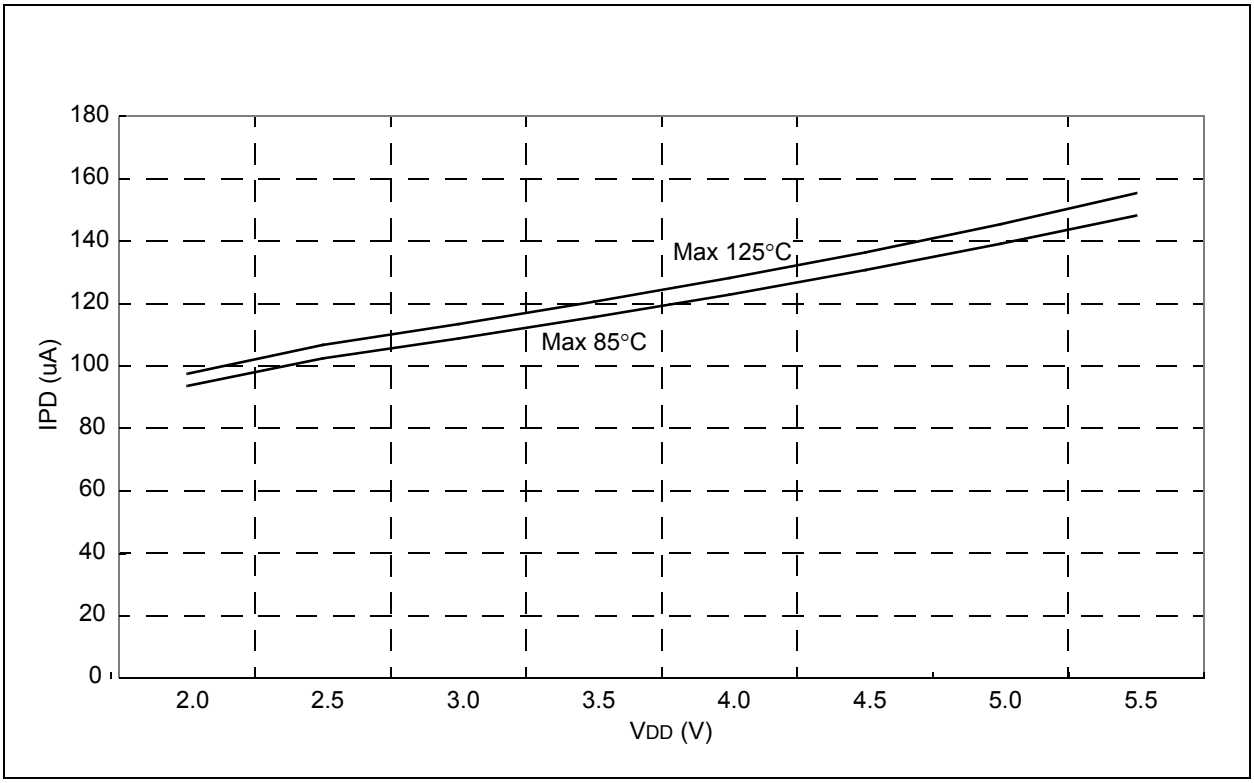


**FIGURE 18-25: T1OSC IPD vs. VDD OVER TEMPERATURE (32 kHz)**

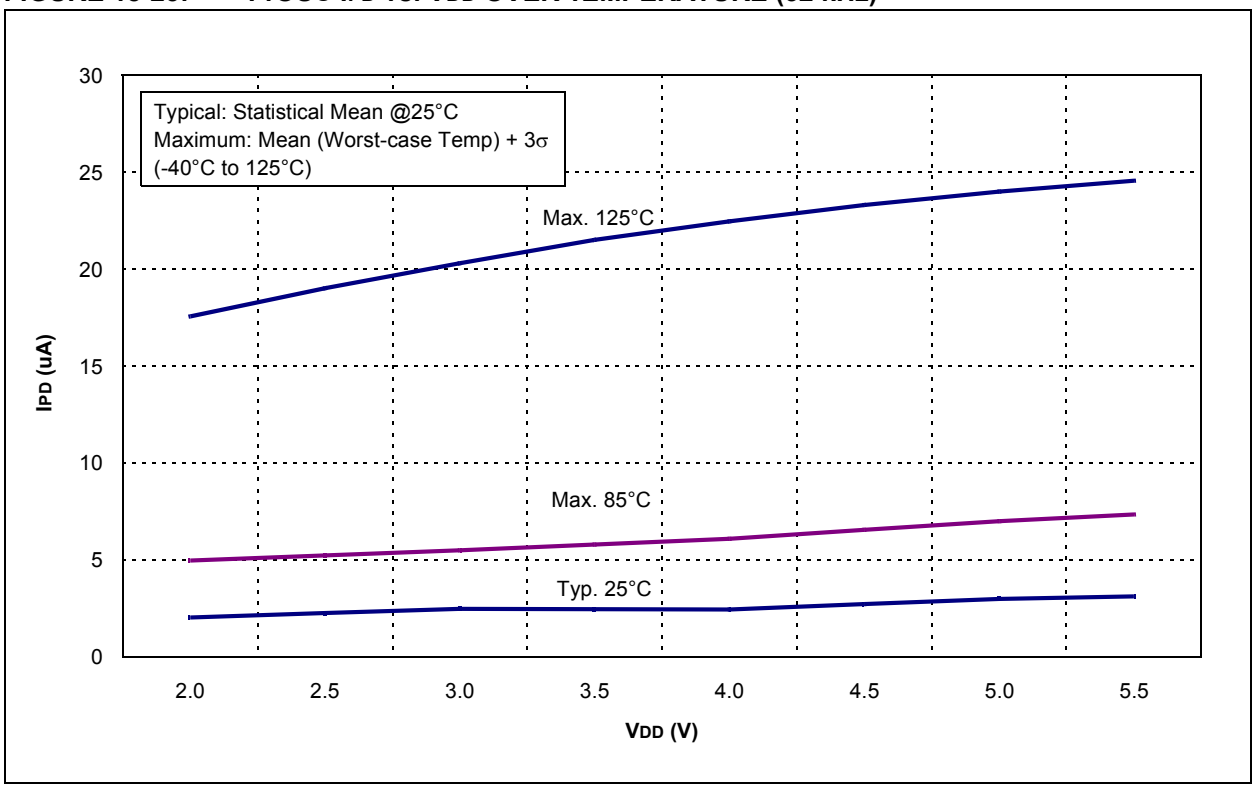
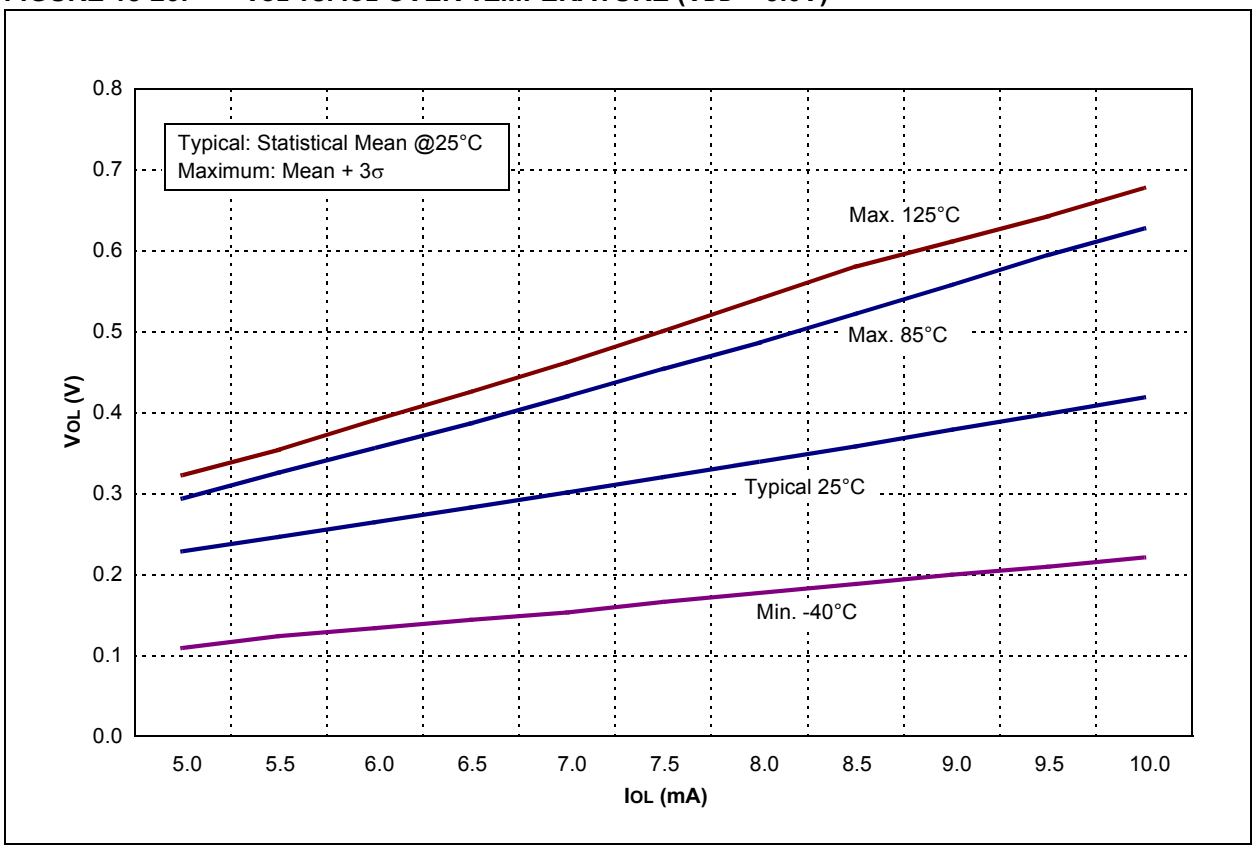



**FIGURE 18-26: VOL vs. IOL OVER TEMPERATURE (VDD = 3.0V)**





© 2006-2012 Microchip Technology Inc.

**FIGURE 18-27: VOL vs. IOL OVER TEMPERATURE (VDD = 5.0V)**

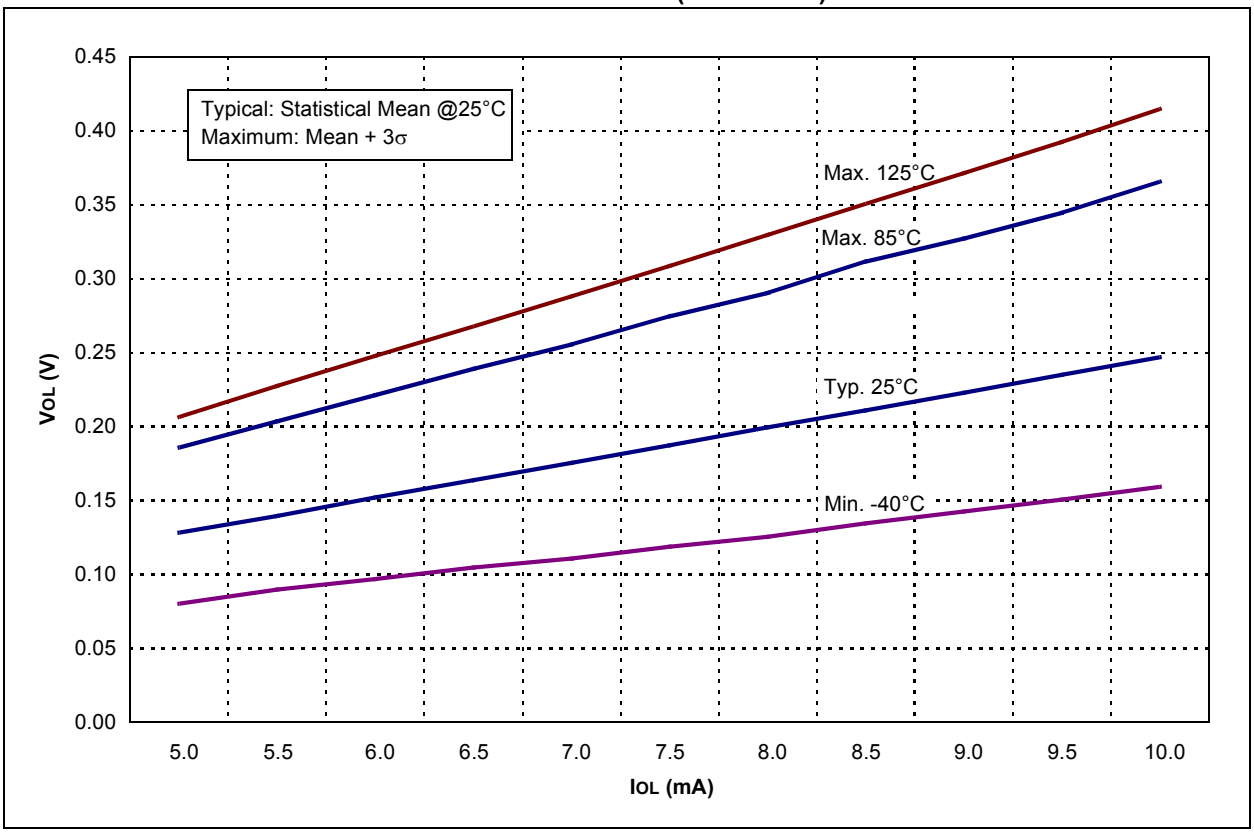



**FIGURE 18-28: VOH vs. IOH OVER TEMPERATURE (VDD = 3.0V)**

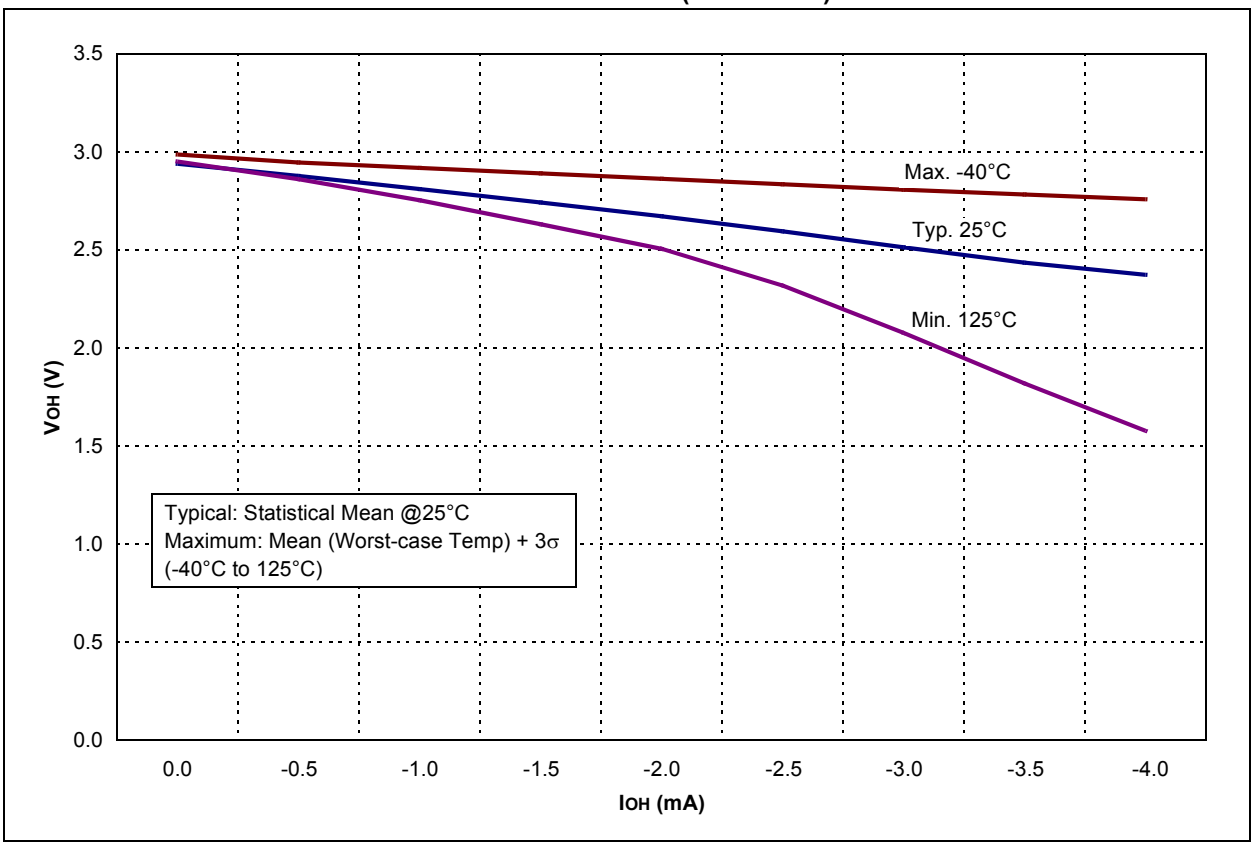


**FIGURE 18-29: VOH vs. IOH OVER TEMPERATURE (VDD = 5.0V)**

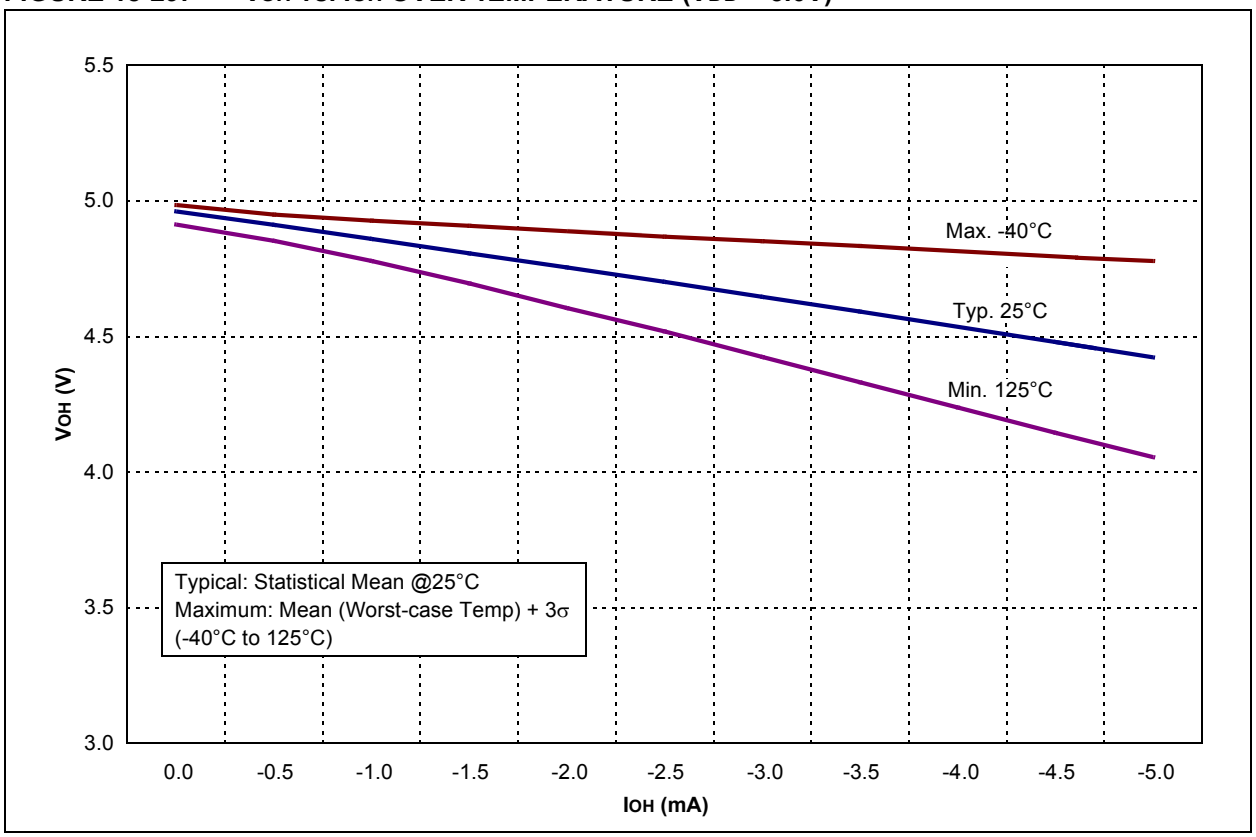
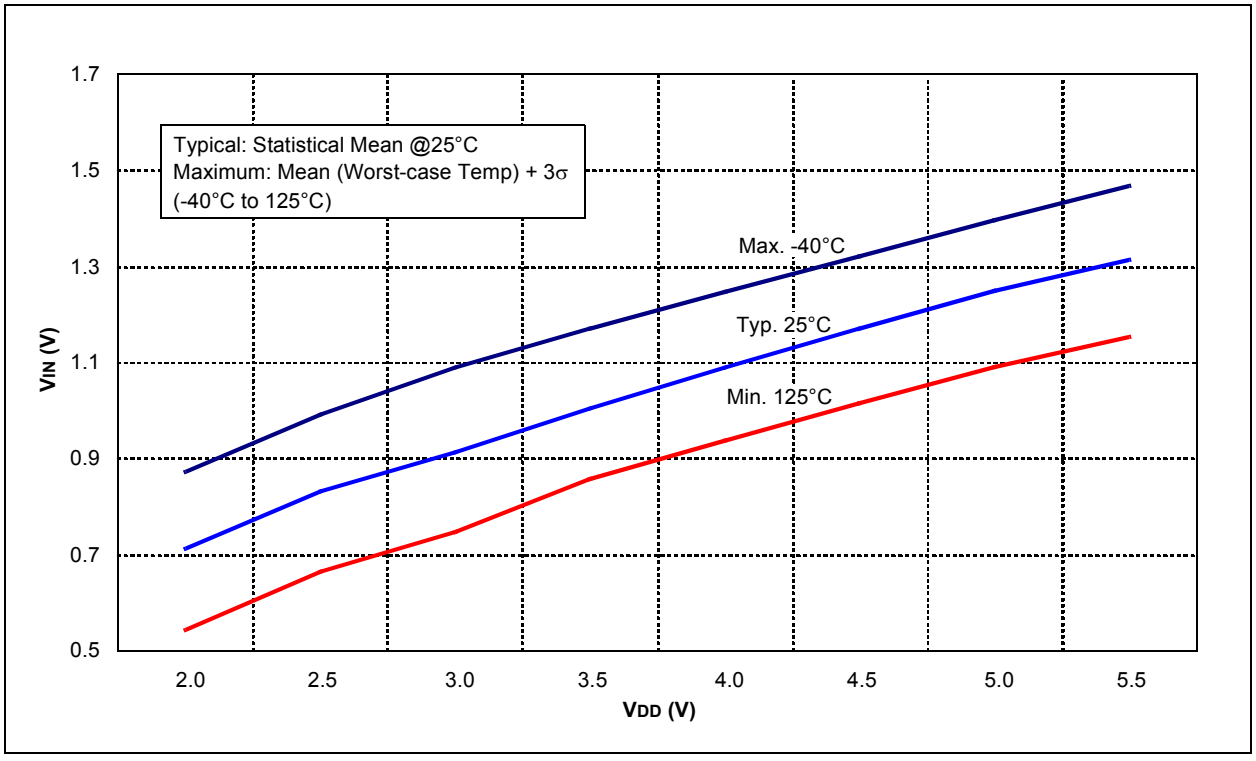



**FIGURE 18-30: TTL INPUT THRESHOLD VIN vs. VDD OVER TEMPERATURE**





© 2006-2012 Microchip Technology Inc.

**FIGURE 18-31: SCHMITT TRIGGER INPUT THRESHOLD VIN vs. VDD OVER TEMPERATURE**

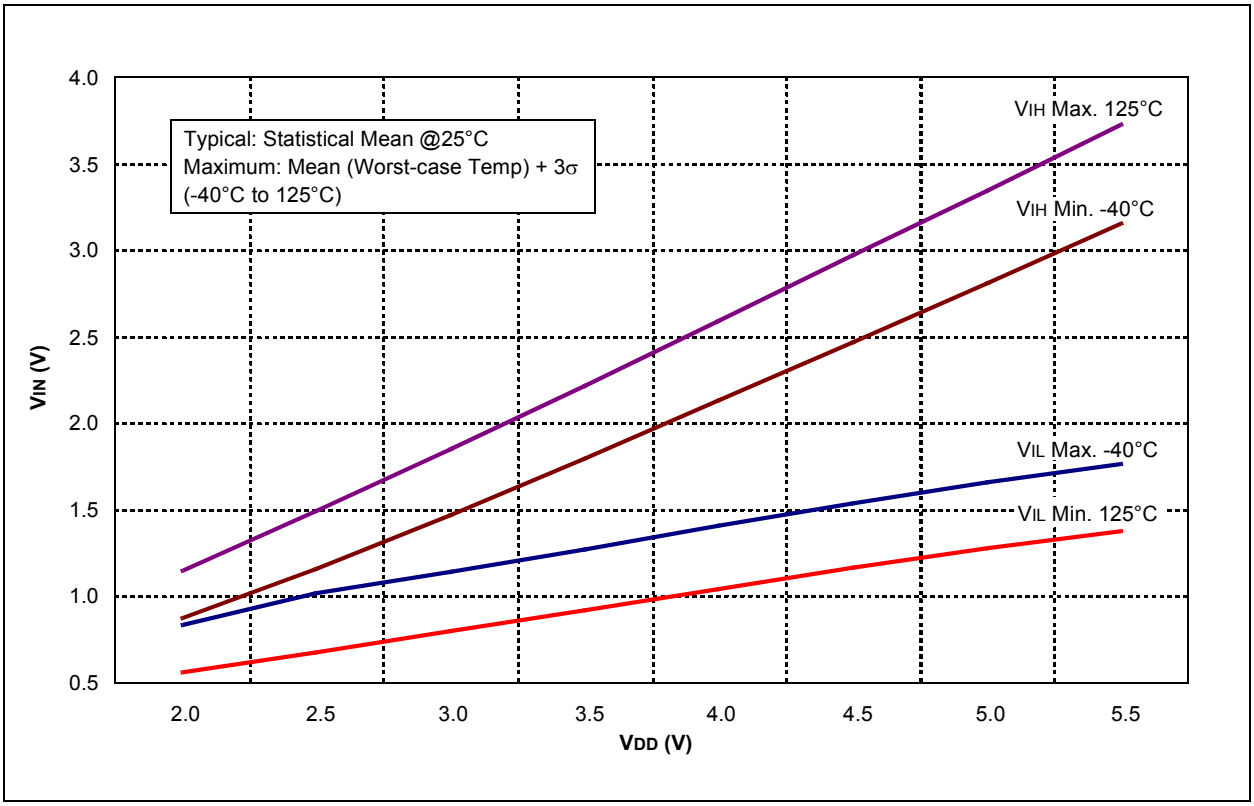



**FIGURE 18-32: COMPARATOR RESPONSE TIME (RISING EDGE)**

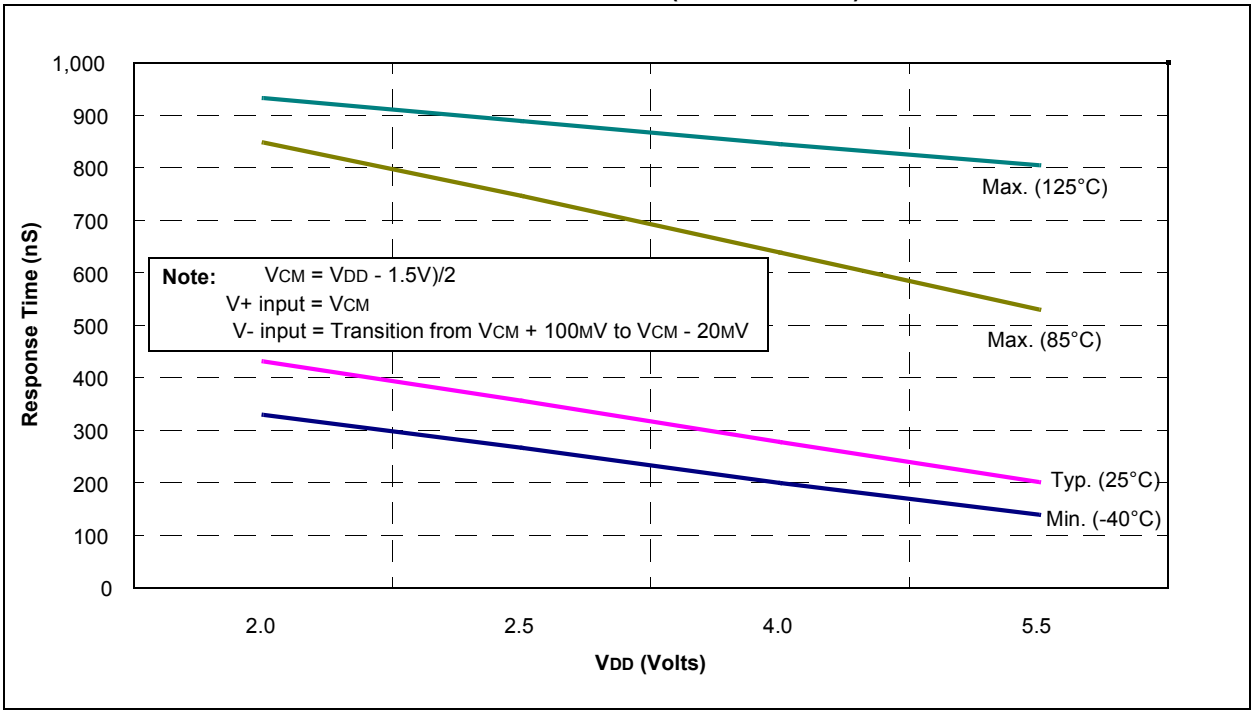


**FIGURE 18-33: COMPARATOR RESPONSE TIME (FALLING EDGE)**

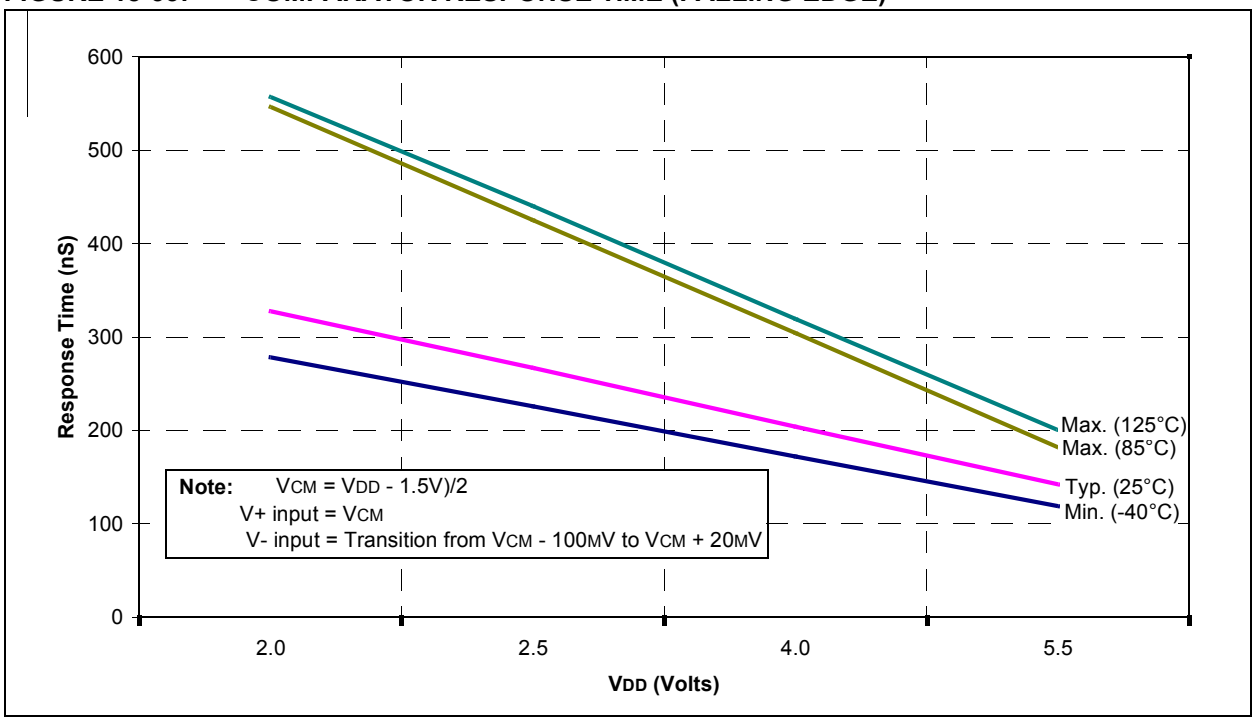
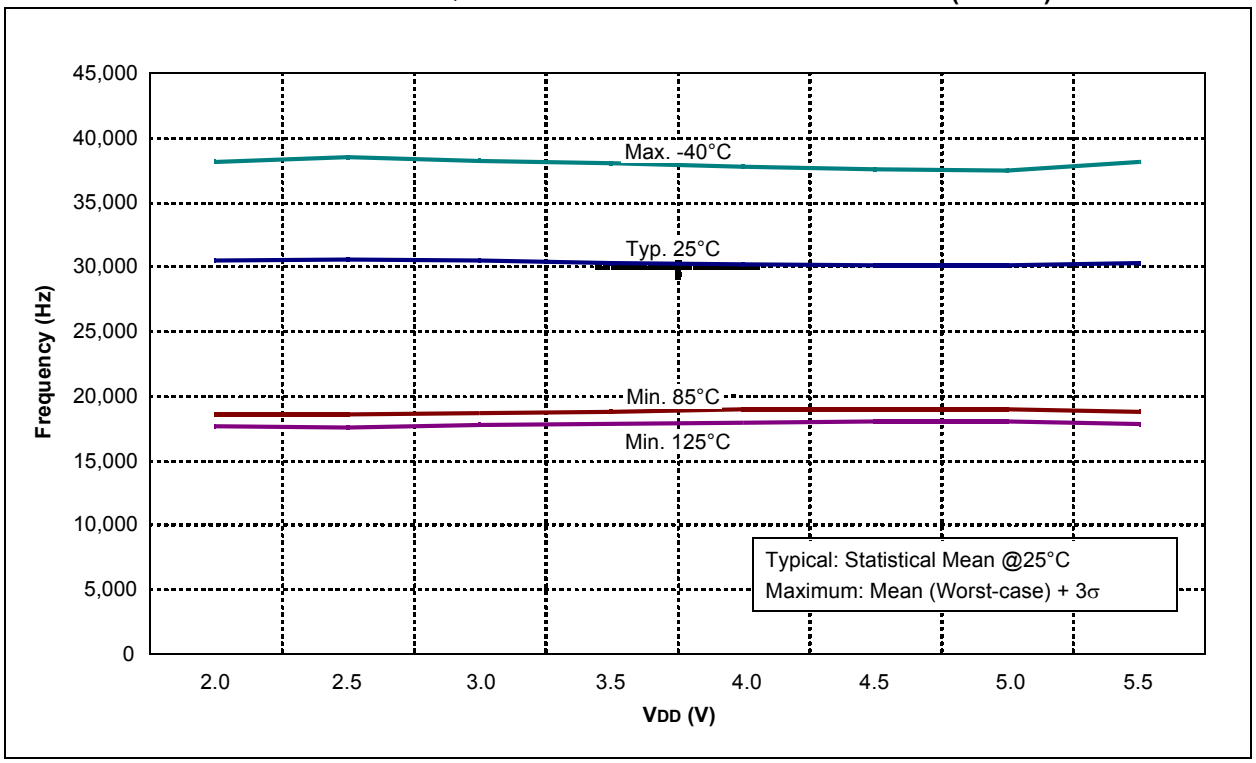



**FIGURE 18-34: LFINTOSC FREQUENCY vs. VDD OVER TEMPERATURE (31 kHz)**





© 2006-2012 Microchip Technology Inc.

**FIGURE 18-35: ADC CLOCK PERIOD vs. VDD OVER TEMPERATURE**

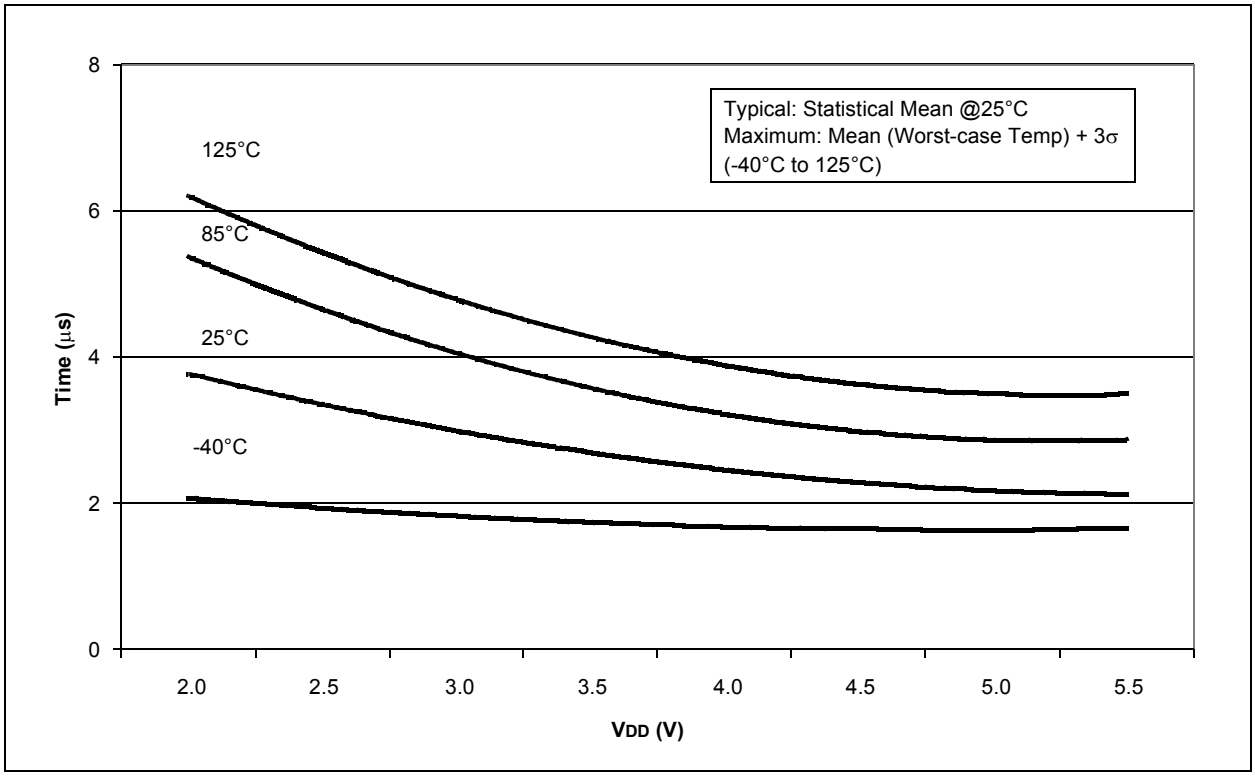



**FIGURE 18-36: TYPICAL HFINTOSC START-UP TIMES vs. VDD OVER TEMPERATURE**

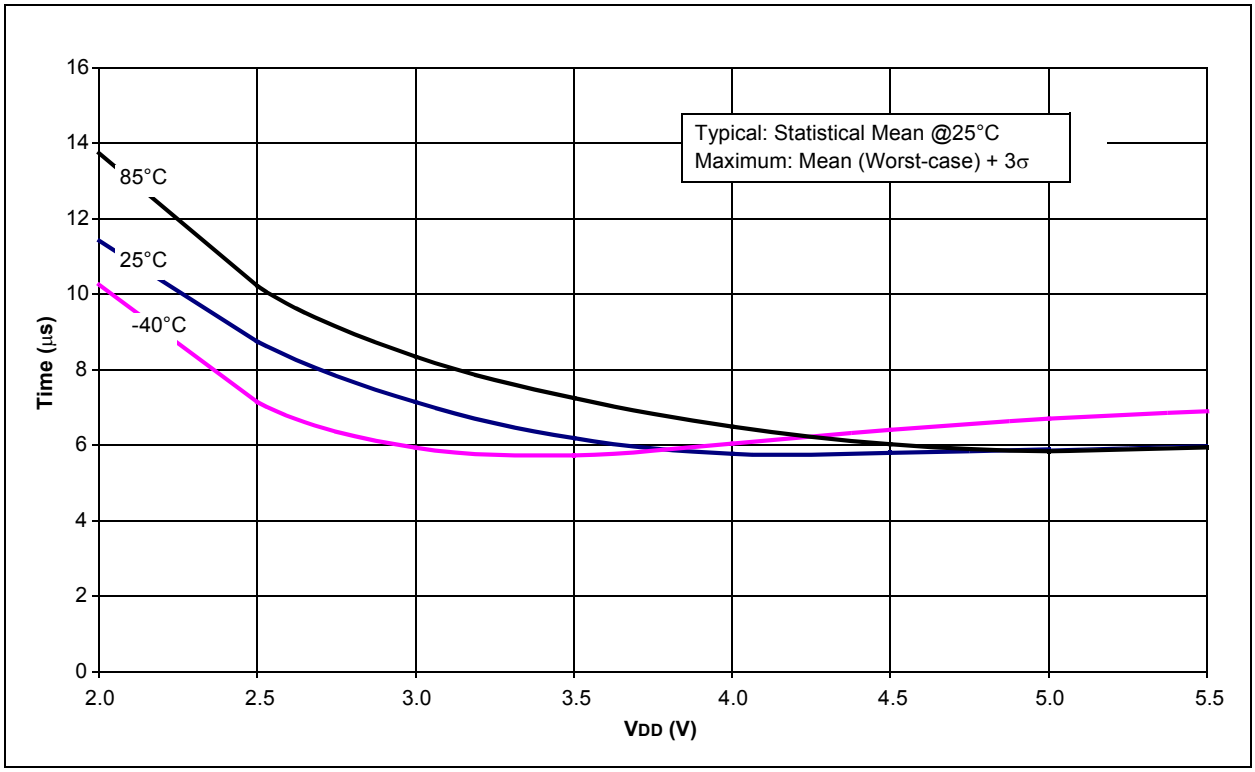


**FIGURE 18-37: MAXIMUM HFINTOSC START-UP TIMES vs. VDD OVER TEMPERATURE**

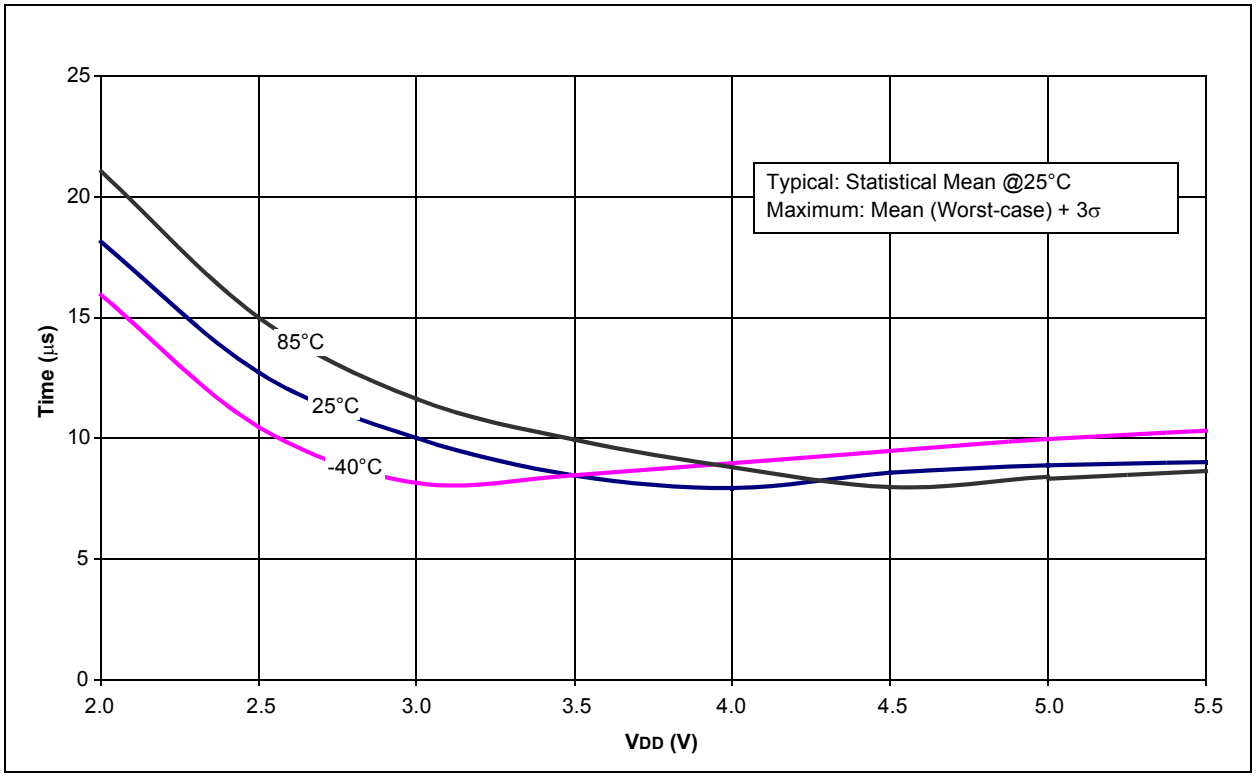
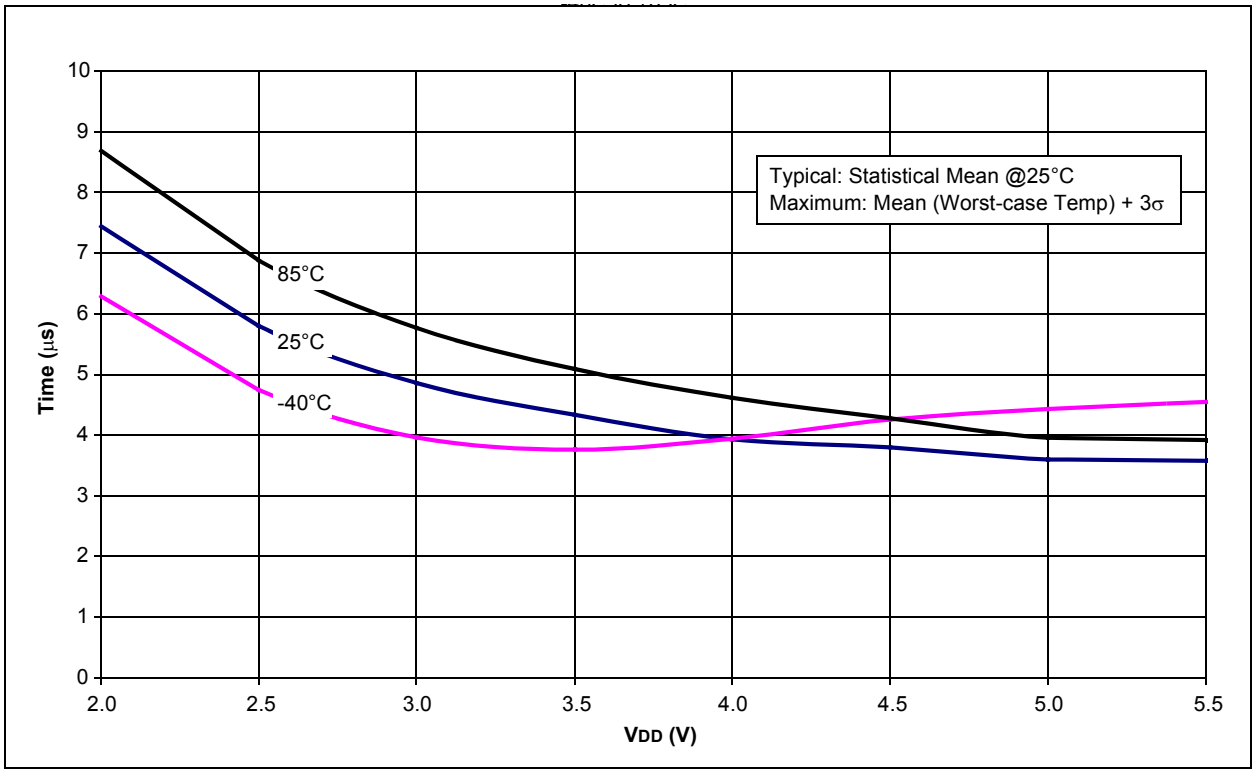



**FIGURE 18-38: MINIMUM HFINTOSC START-UP TIMES vs. VDD OVER TEMPERATURE**





© 2006-2012 Microchip Technology Inc.

**FIGURE 18-39: TYPICAL HFINTOSC FREQUENCY CHANGE vs. VDD (25°C)**

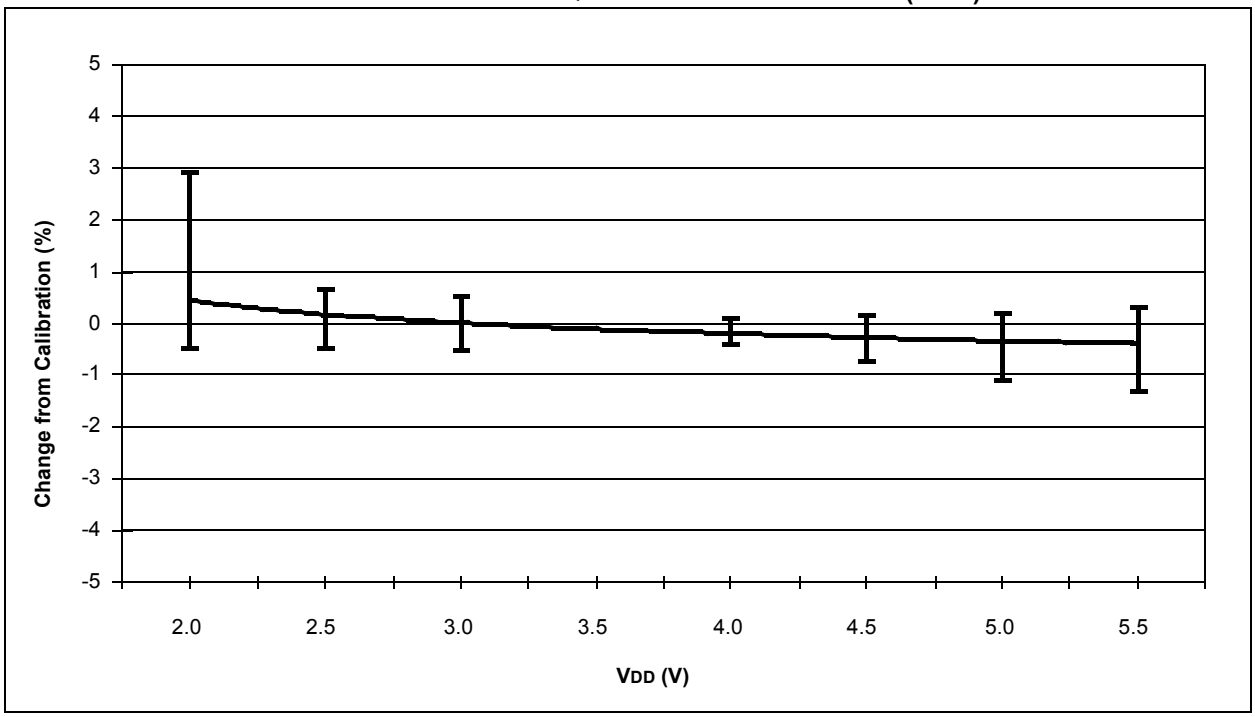



**FIGURE 18-40: TYPICAL HFINTOSC FREQUENCY CHANGE OVER DEVICE VDD (85°C)**

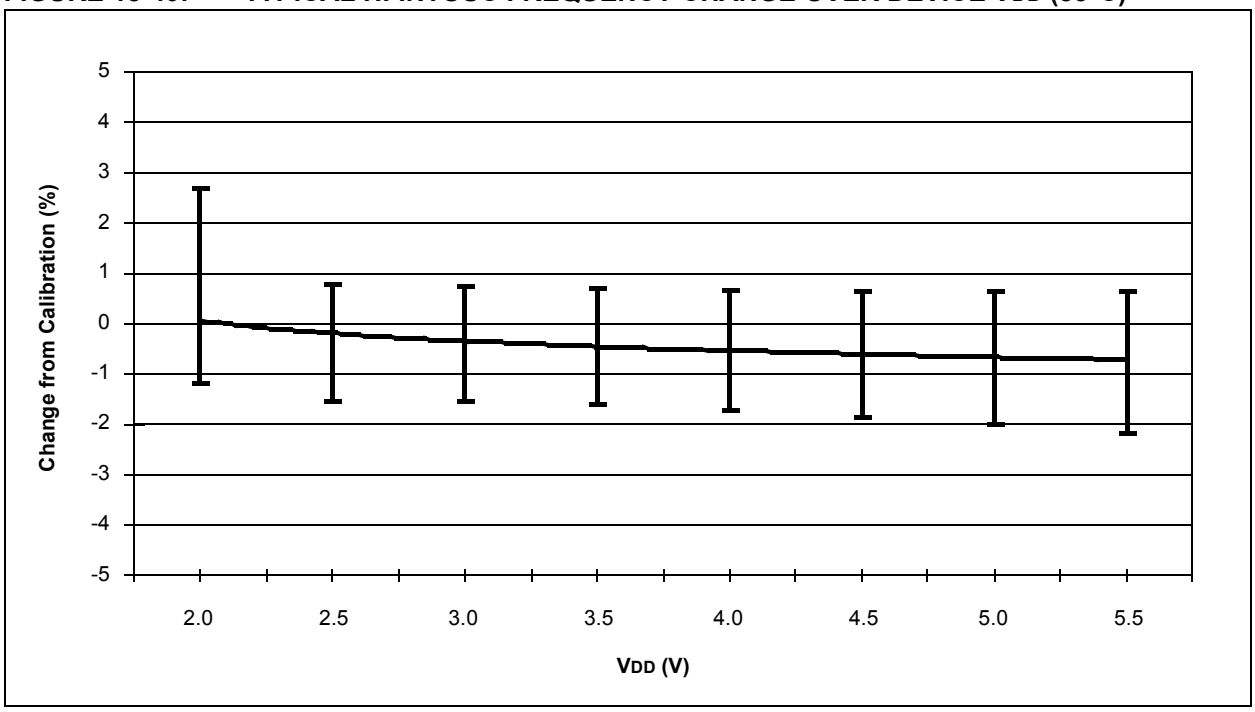


**FIGURE 18-41: TYPICAL HFINTOSC FREQUENCY CHANGE vs. VDD (125°C)**

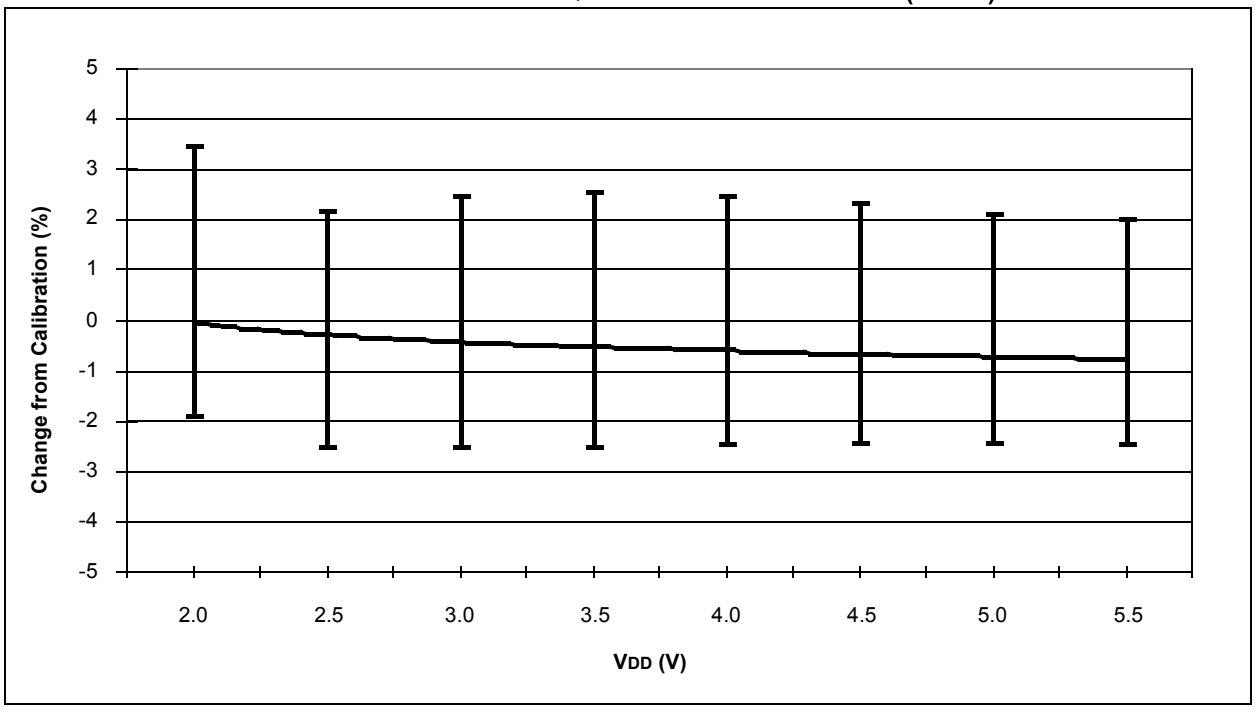
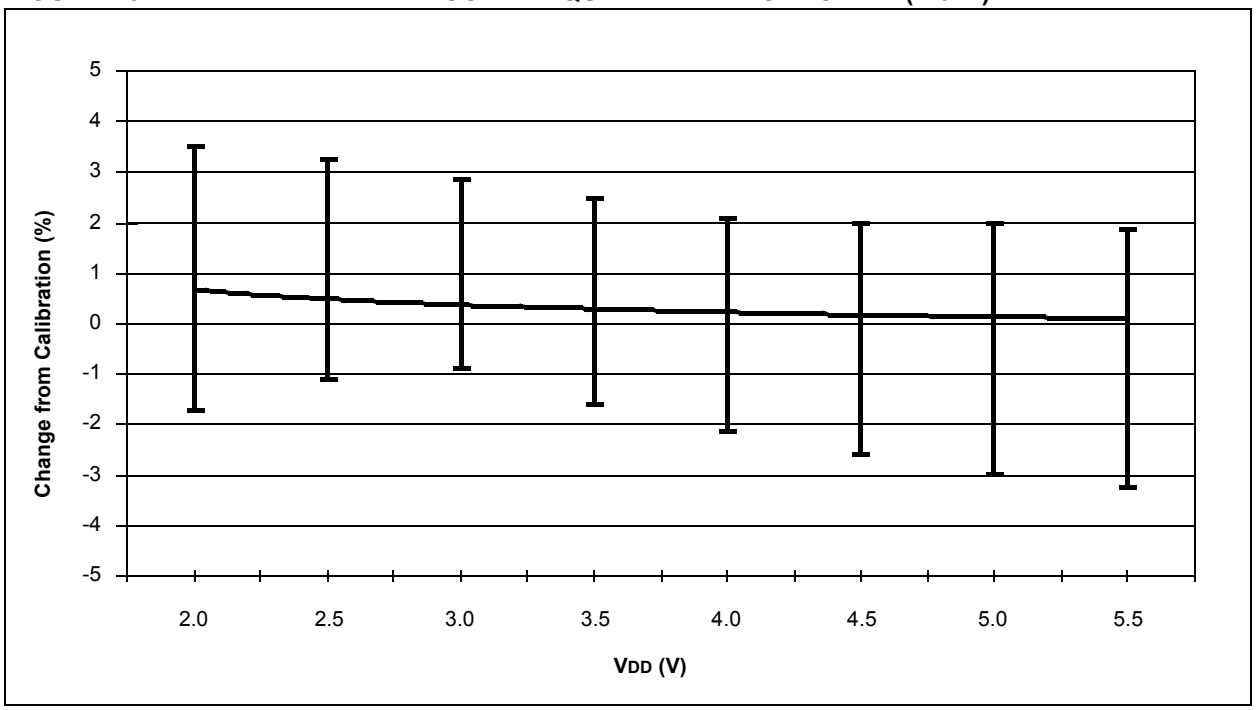



**FIGURE 18-42: TYPICAL HFINTOSC FREQUENCY CHANGE vs. VDD (-40°C)**





© 2006-2012 Microchip Technology Inc.

**FIGURE 18-43: TYPICAL VP6 REFERENCE VOLTAGE vs. VDD (25°C)**

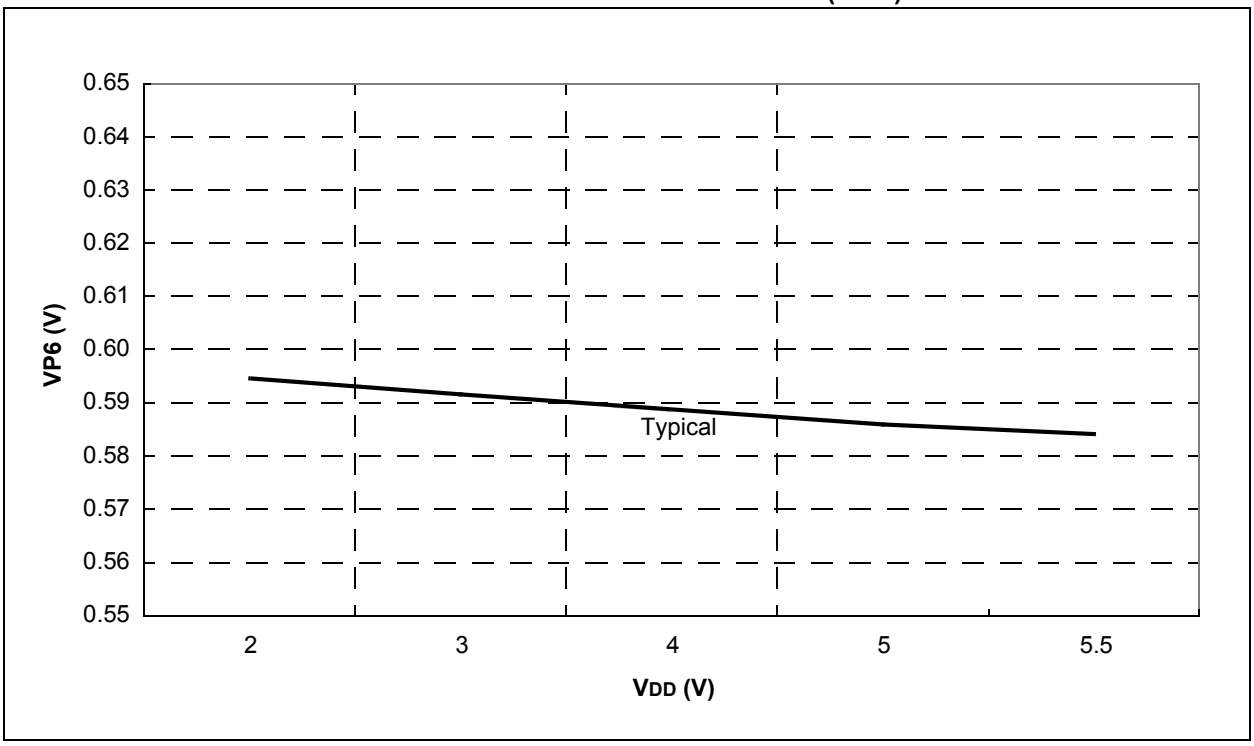



**FIGURE 18-44: VP6 DRIFT OVER TEMPERATURE NORMALIZED AT 25°C (VDD 5V)**

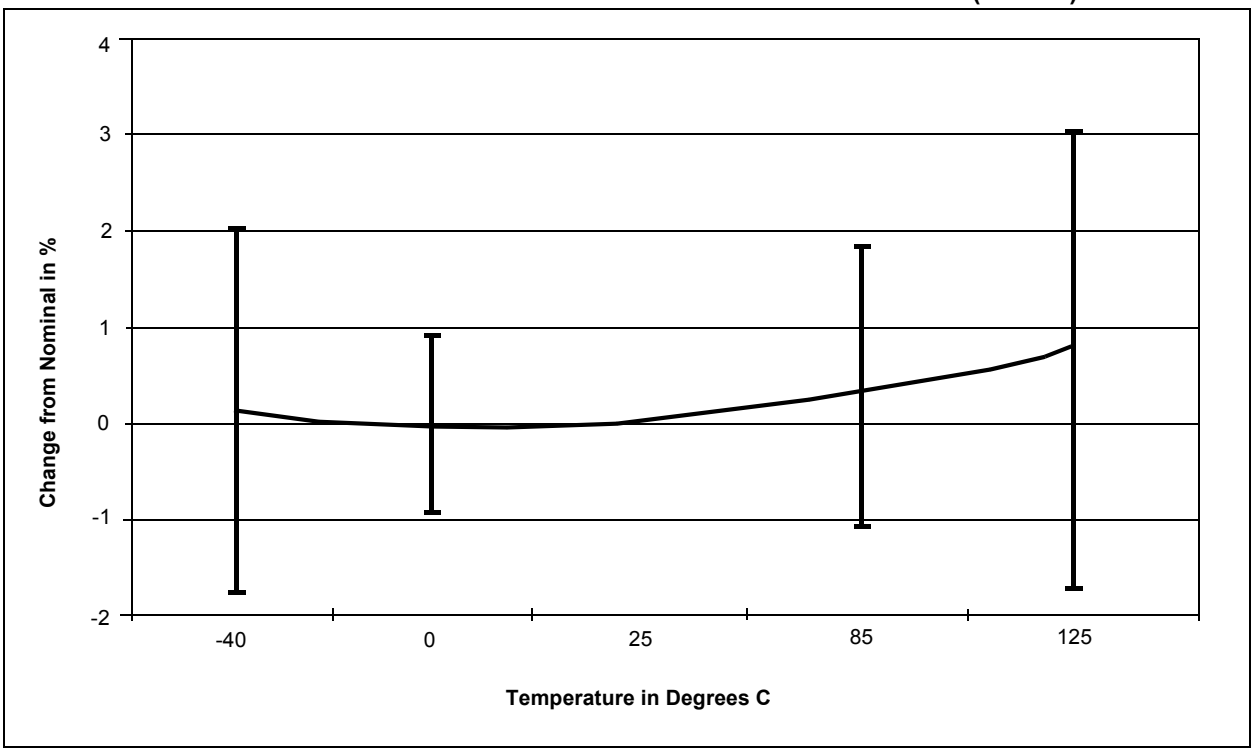


**FIGURE 18-45: VP6 DRIFT OVER TEMPERATURE NORMALIZED AT 25°C ($V_{DD}$ 3V)**

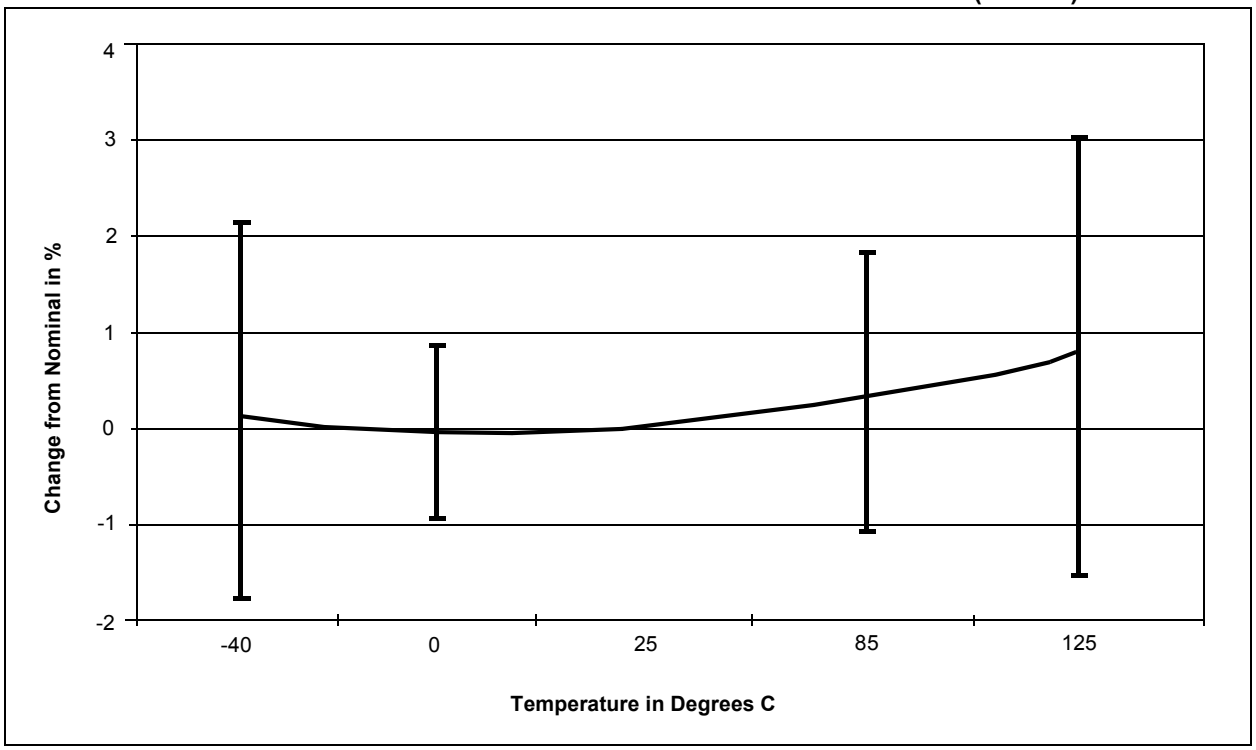
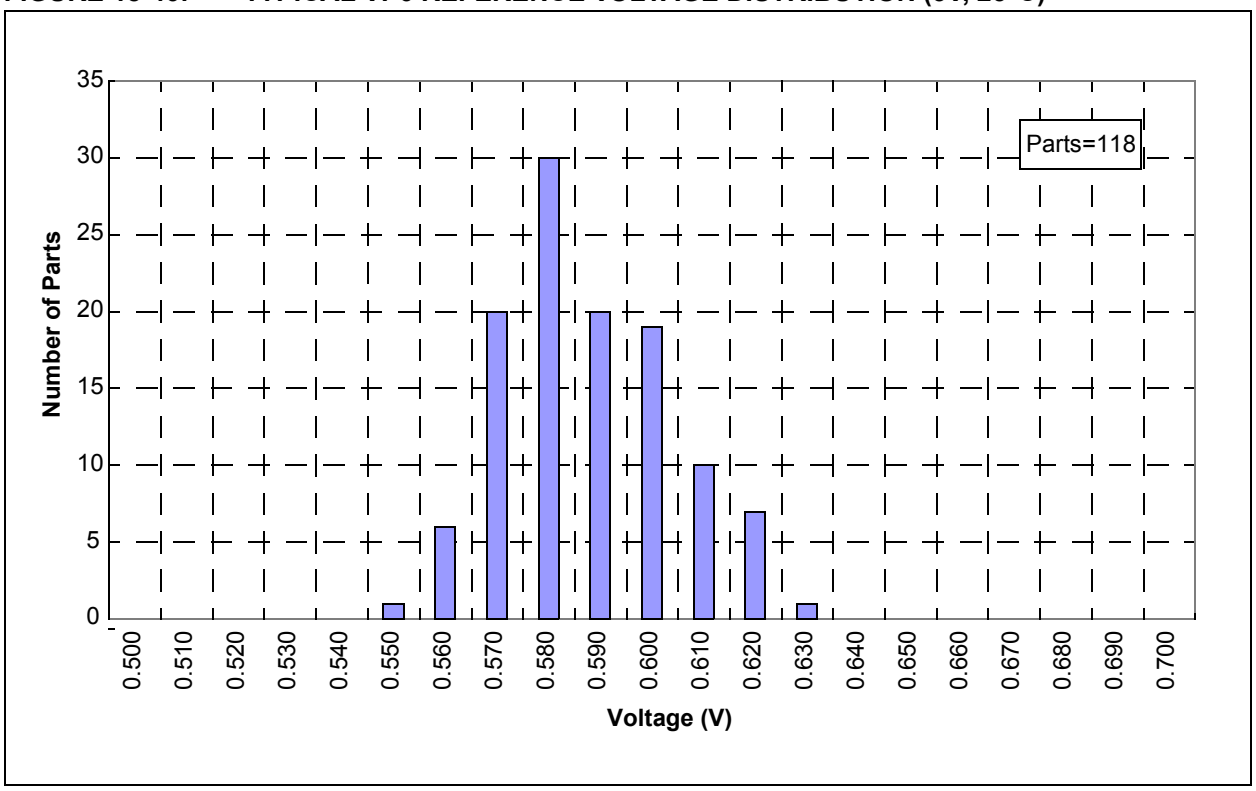



**FIGURE 18-46: TYPICAL VP6 REFERENCE VOLTAGE DISTRIBUTION (3V, 25°C)**





© 2006-2012 Microchip Technology Inc.

**FIGURE 18-47: TYPICAL VP6 REFERENCE VOLTAGE DISTRIBUTION (3V, 85°C)**

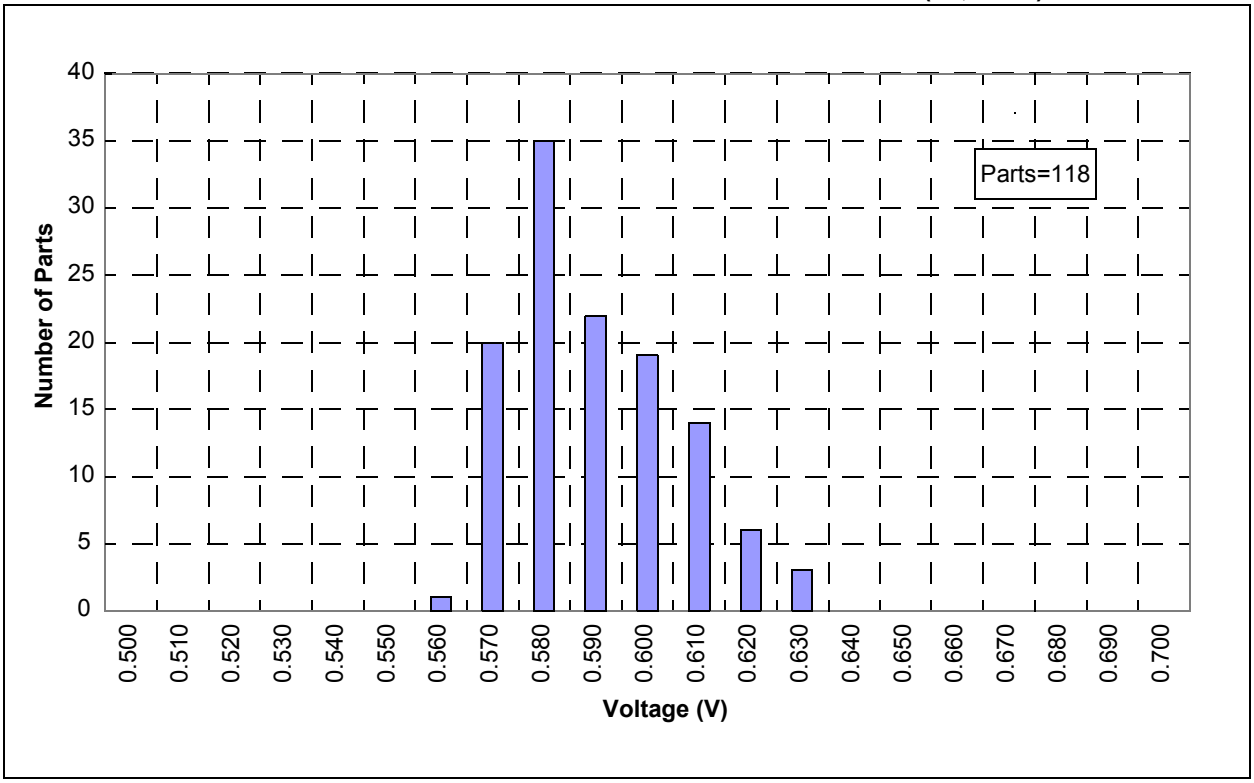



**FIGURE 18-48: TYPICAL VP6 REFERENCE VOLTAGE DISTRIBUTION (3V, 125°C)**

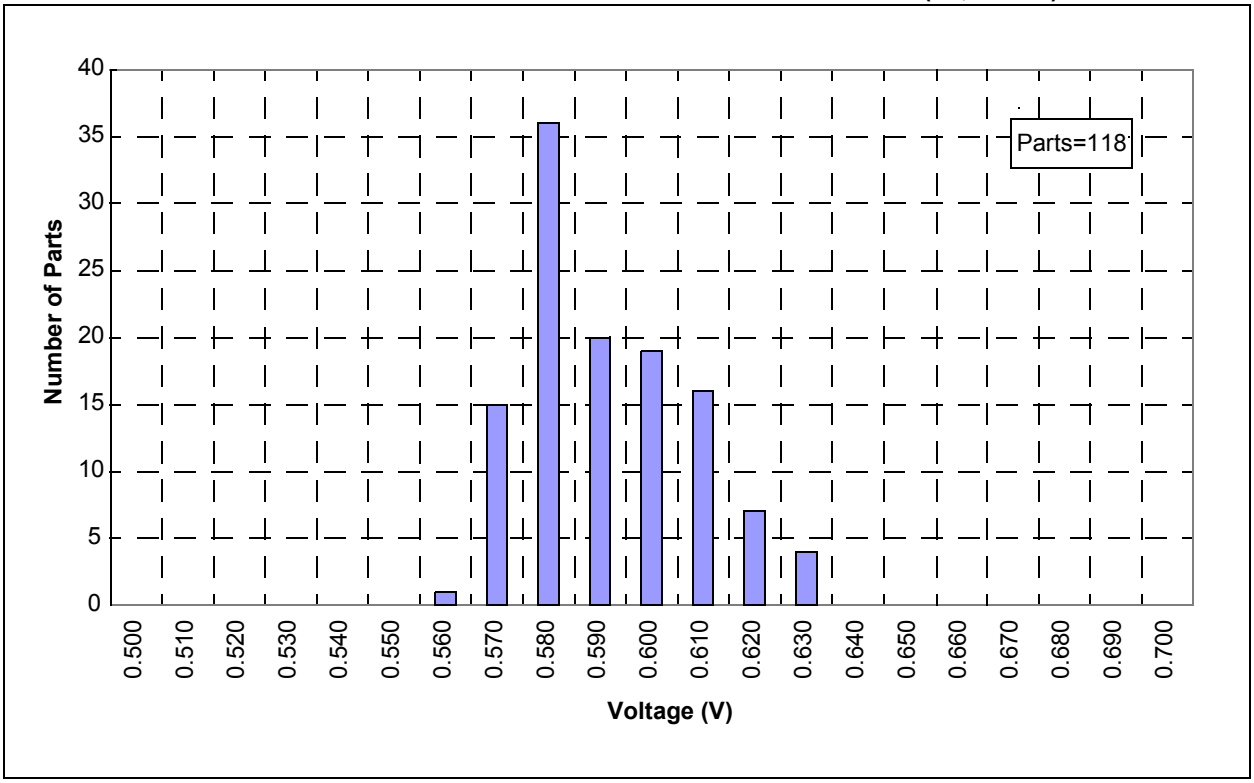


**FIGURE 18-49: TYPICAL VP6 REFERENCE VOLTAGE DISTRIBUTION (3V, -40°C)**

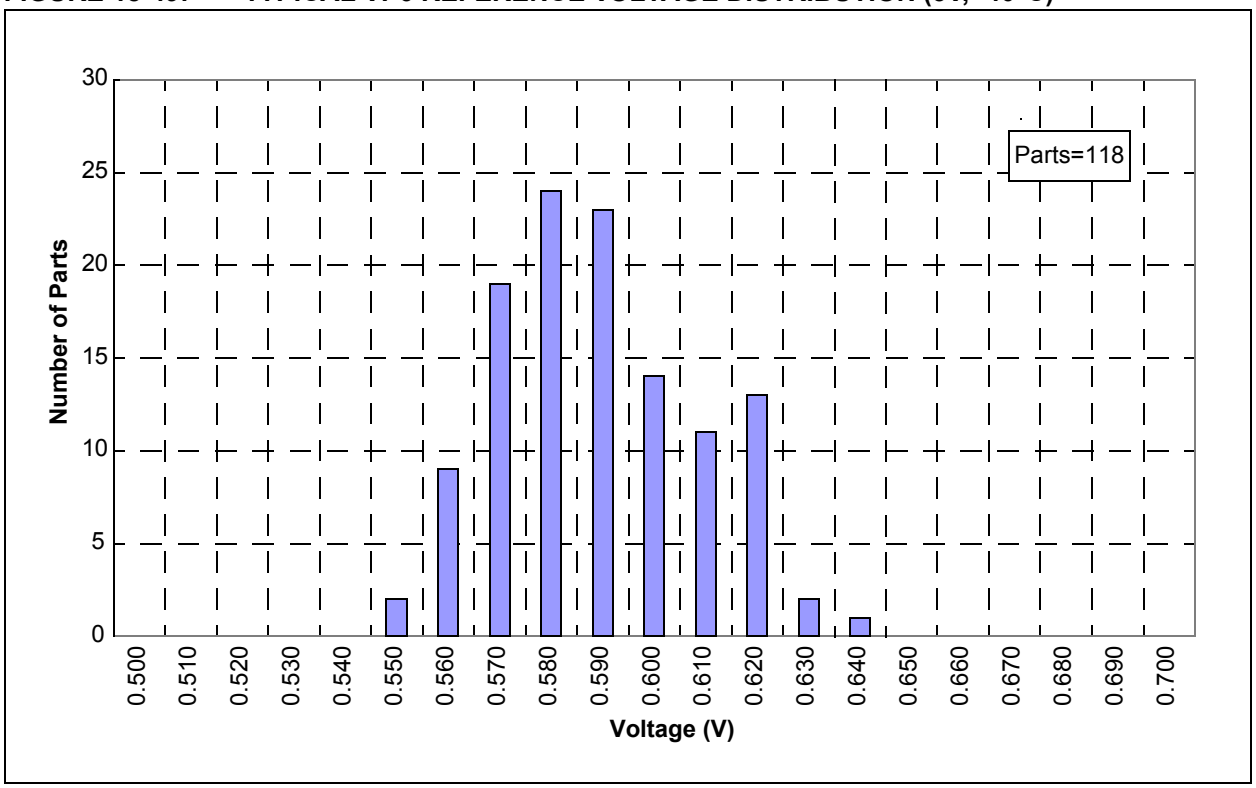
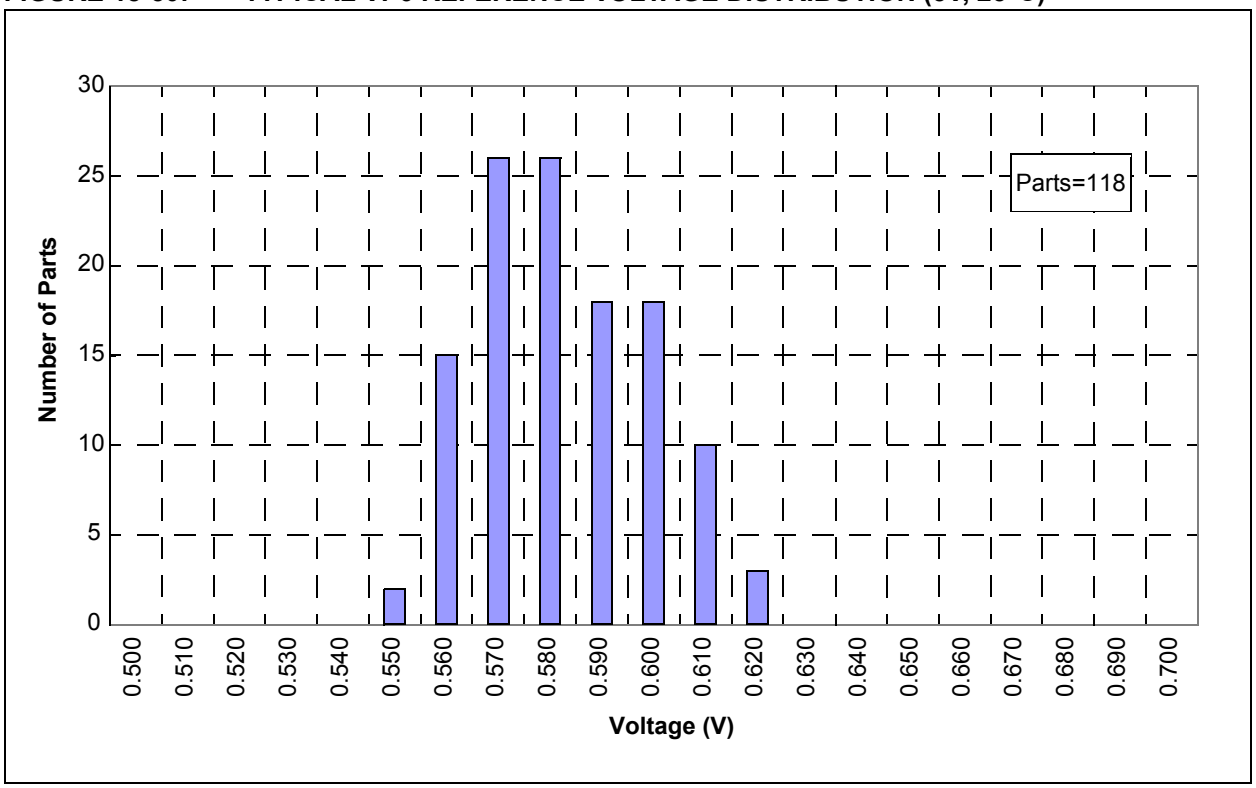



**FIGURE 18-50: TYPICAL VP6 REFERENCE VOLTAGE DISTRIBUTION (5V, 25°C)**





© 2006-2012 Microchip Technology Inc.

**FIGURE 18-51: TYPICAL VP6 REFERENCE VOLTAGE DISTRIBUTION (5V, 85°C)**

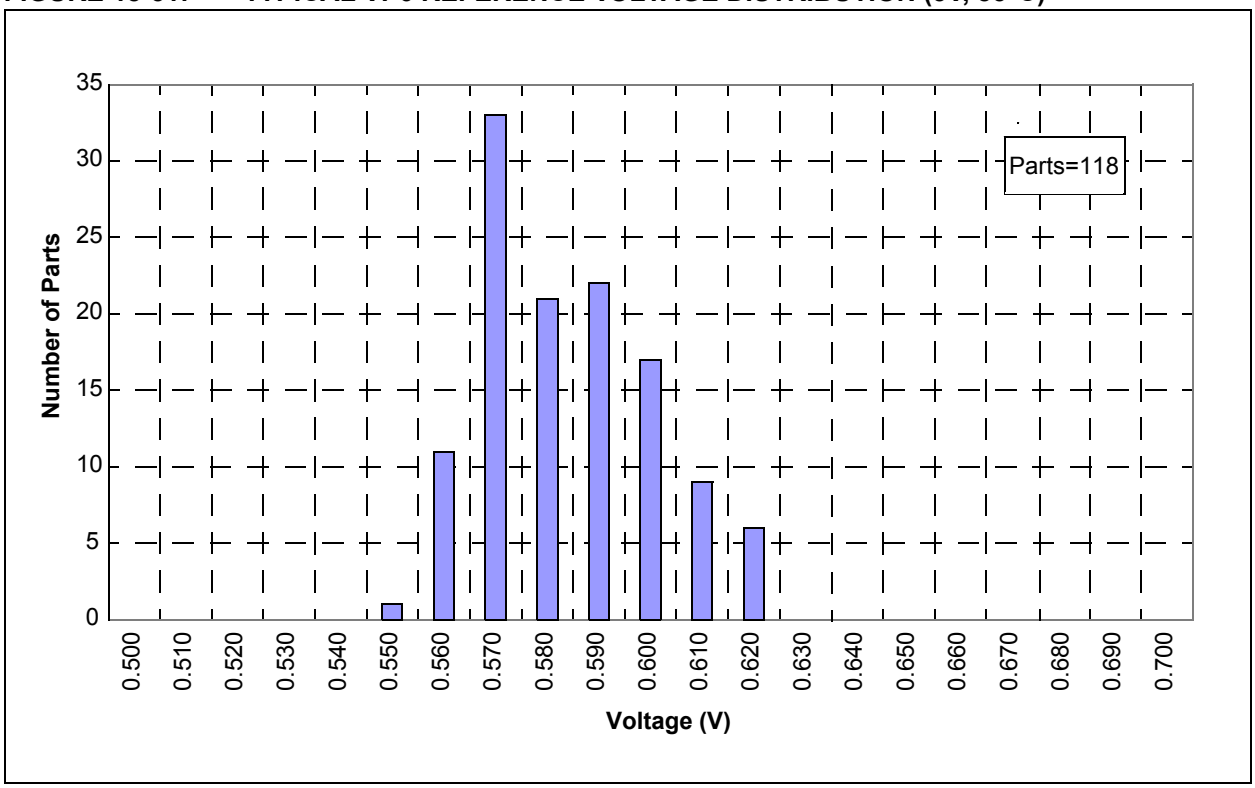



**FIGURE 18-52: TYPICAL VP6 REFERENCE VOLTAGE DISTRIBUTION (5V, 125°C)**

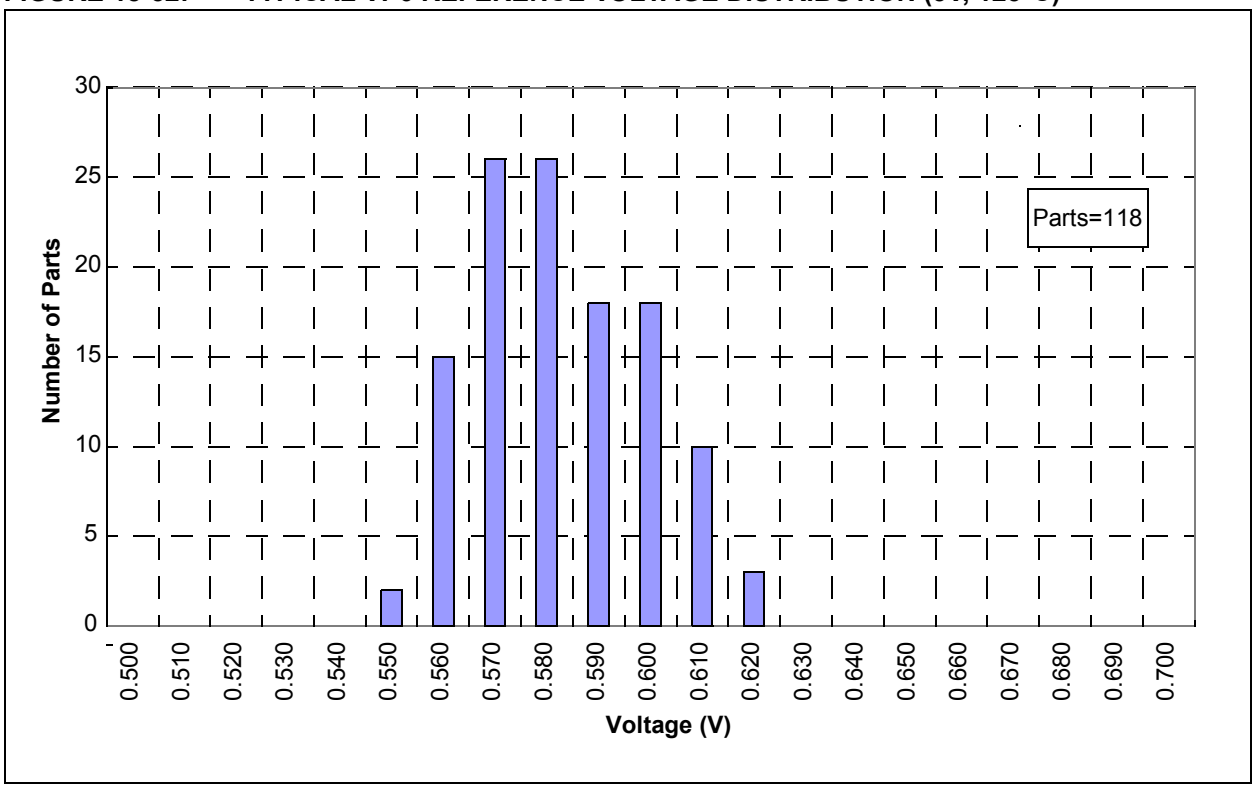
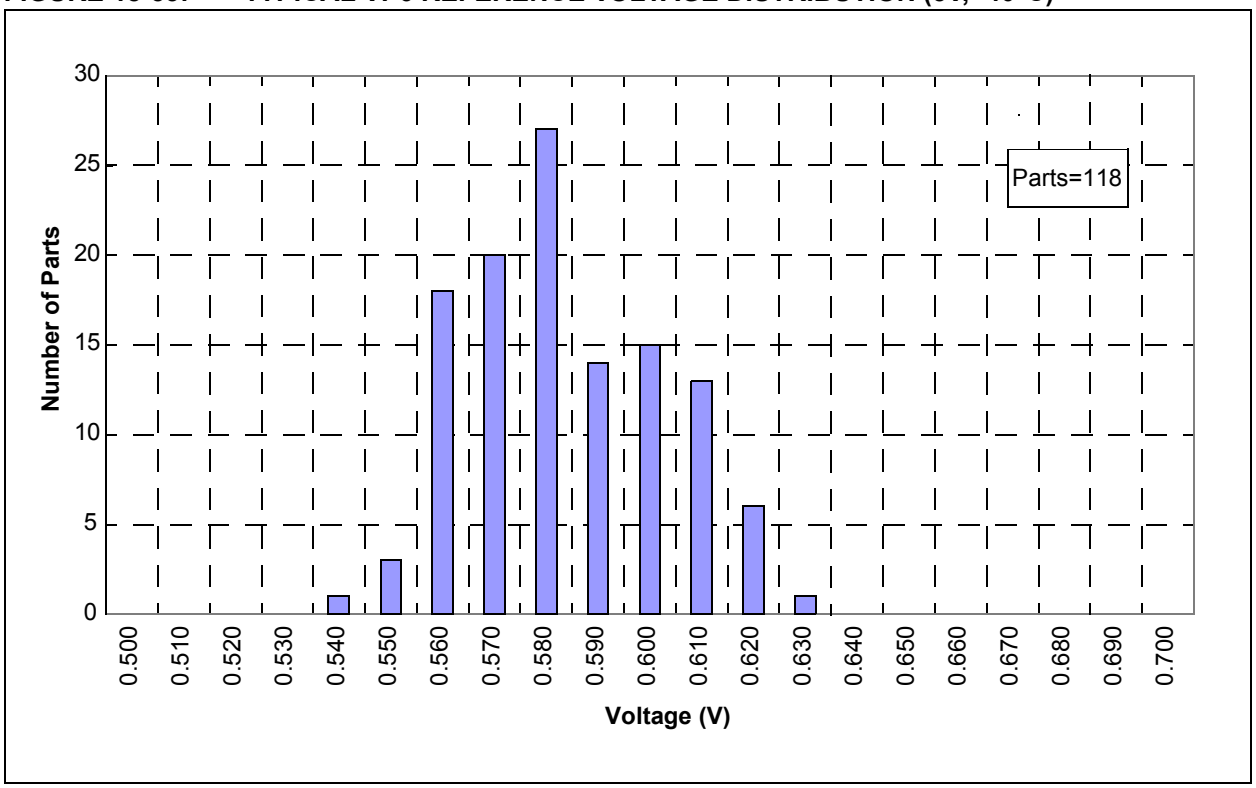


**FIGURE 18-53: TYPICAL VP6 REFERENCE VOLTAGE DISTRIBUTION (5V, -40°C)**



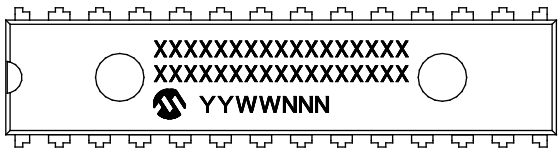


© 2006-2012 Microchip Technology Inc.

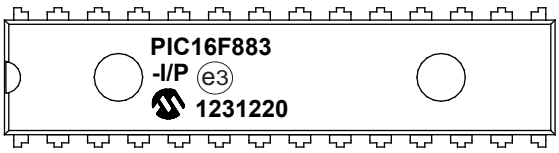
## 19.0 PACKAGING INFORMATION

### 19.1 Package Marking Information

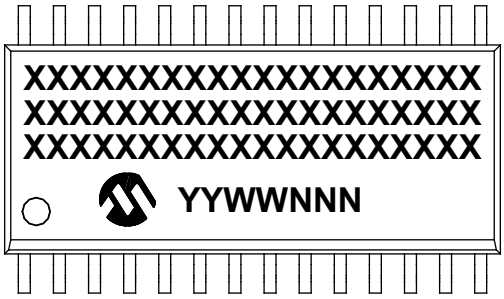
28-Lead SPDIP (.300")



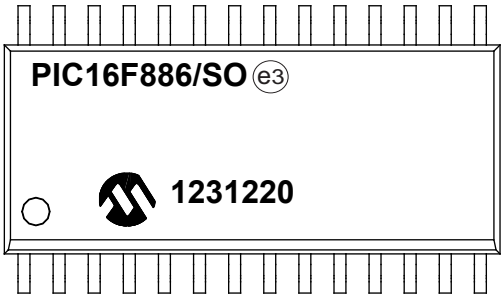

Example



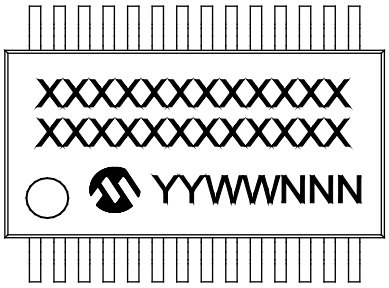

28-Lead SOIC (7.50 mm)



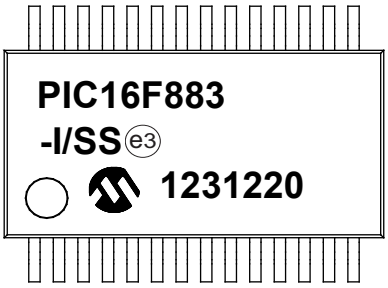

Example




28-Lead SSOP (5.30 mm)




Example



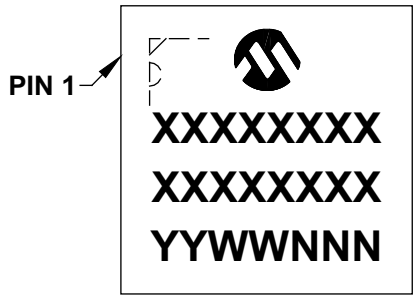

| | | |
|---|---|---|
| **Legend:** | XX...X | Customer-specific information |
| | Y | Year code (last digit of calendar year) |
| | YY | Year code (last 2 digits of calendar year) |
| | WW | Week code (week of January 1 is week '01') |
| | NNN | Alphanumeric traceability code |
| | (e3) | Pb-free JEDEC 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.

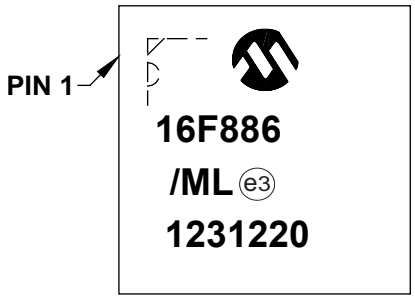
* Standard PICmicro® device marking consists of Microchip part number, year code, week code and traceability code. For PICmicro device marking beyond this, certain price adders apply. Please check with your Microchip Sales Office. For QTP devices, any special marking adders are included in QTP price.

### 19.1 Package Marking Information (Continued)

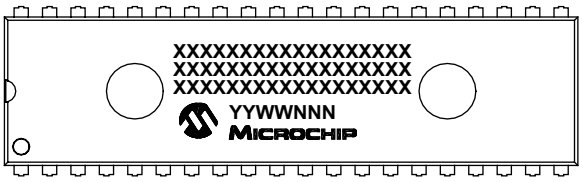
28-Lead QFN (6x6 mm)



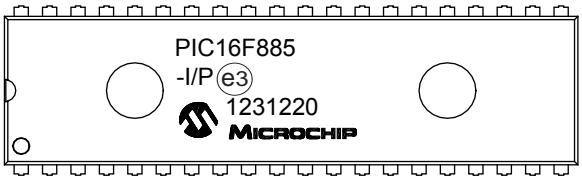

Example



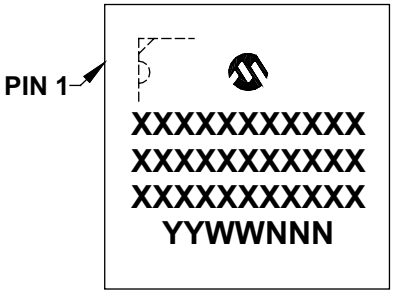

40-Lead PDIP (600 mil)



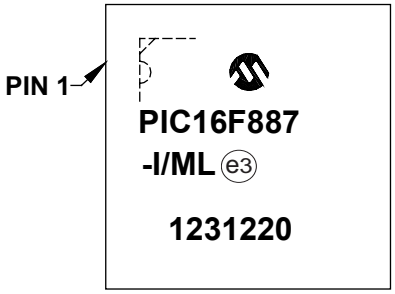

Example




44-Lead QFN (8x8x0.9 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. |

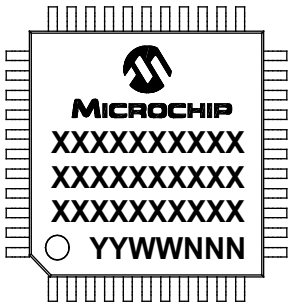
**Note**: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line, thus limiting the number of available characters for customer-specific information.

***** Standard PICmicro® device marking consists of Microchip part number, year code, week code and traceability code. For PICmicro device marking beyond this, certain price adders apply. Please check with your Microchip Sales Office. For QTP devices, any special marking adders are included in QTP price.

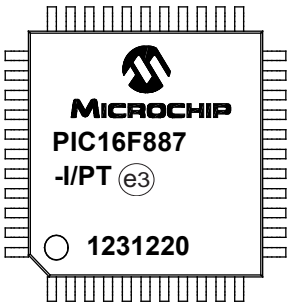

© 2006-2012 Microchip Technology Inc.

## 19.1 Package Marking Information (Continued)

44-Lead TQFP (10x10x1 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. | |

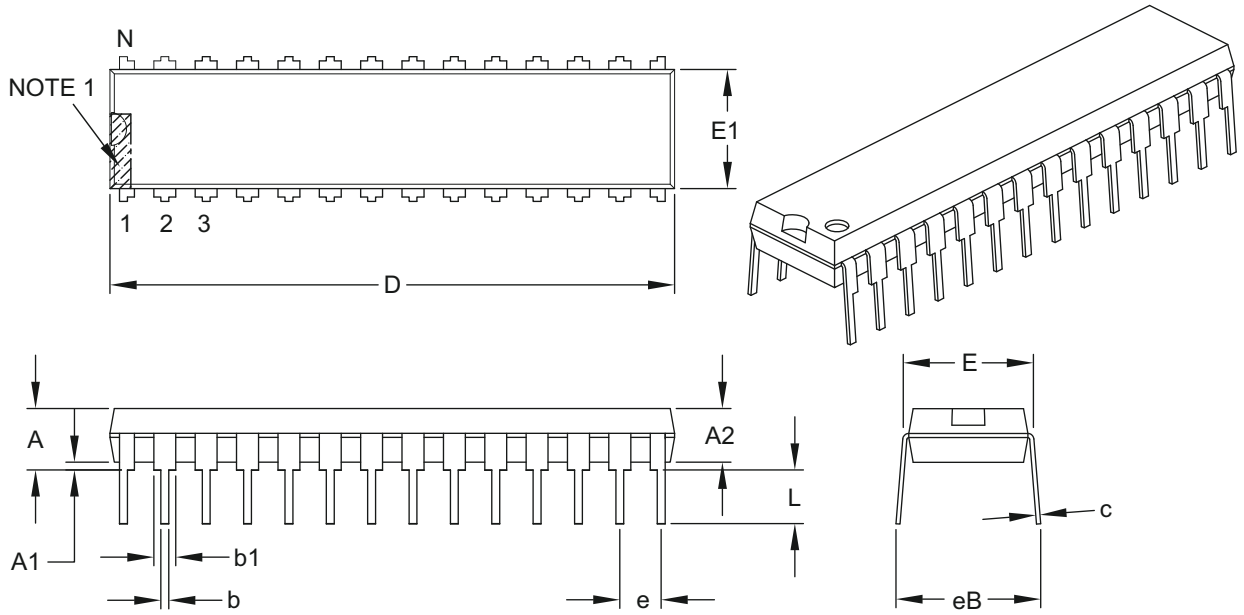
***** Standard PICmicro® device marking consists of Microchip part number, year code, week code and traceability code. For PICmicro device marking beyond this, certain price adders apply. Please check with your Microchip Sales Office. For QTP devices, any special marking adders are included in QTP price.

## 19.2 Package Details

The following sections give the technical details of the packages.

### 28-Lead Skinny Plastic Dual In-Line (SP) – 300 mil Body [SPDIP]

**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 | | 28 | |
| Pitch | e | | .100 BSC | |
| Top to Seating Plane | A | – | – | .200 |
| Molded Package Thickness | A2 | .120 | .135 | .150 |
| Base to Seating Plane | A1 | .015 | – | – |
| Shoulder to Shoulder Width | E | .290 | .310 | .335 |
| Molded Package Width | E1 | .240 | .285 | .295 |
| Overall Length | D | 1.345 | 1.365 | 1.400 |
| Tip to Seating Plane | L | .110 | .130 | .150 |
| Lead Thickness | c | .008 | .010 | .015 |
| Upper Lead Width | b1 | .040 | .050 | .070 |
| Lower Lead Width | b | .014 | .018 | .022 |
| Overall Row Spacing § | eB | – | – | .430 |

**Notes:**

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. § Significant Characteristic.
3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.
4. Dimensioning and tolerancing per ASME Y14.5M.

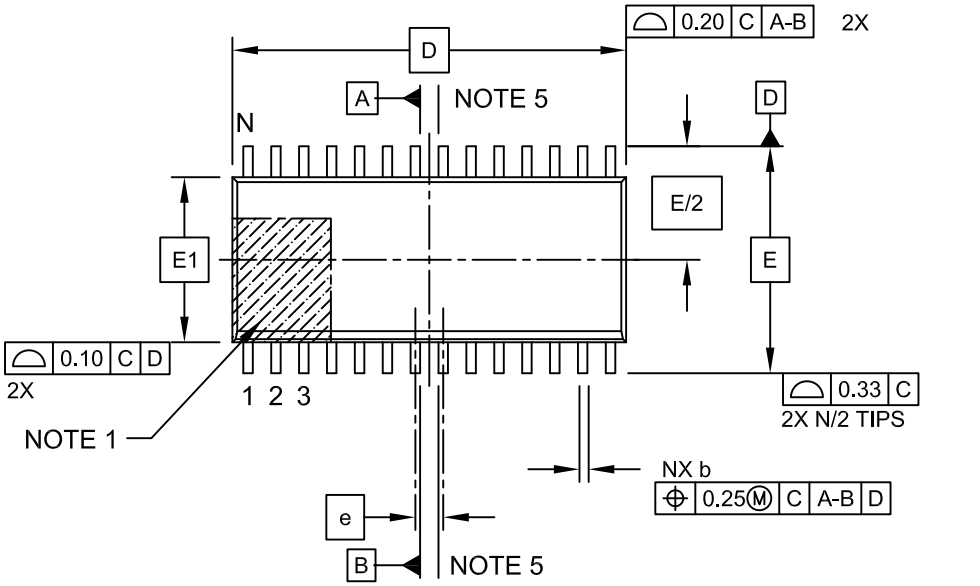
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing C04-070B

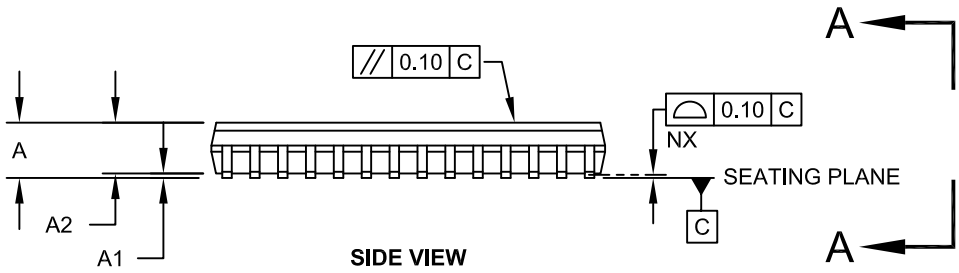

© 2006-2012 Microchip Technology Inc.

### 28-Lead Plastic Small Outline (SO) - Wide, 7.50 mm Body [SOIC]

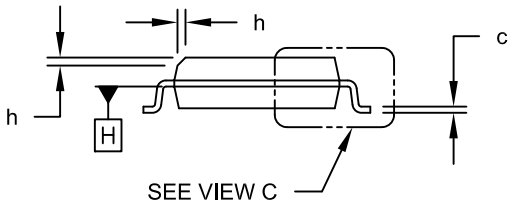
**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging




**TOP VIEW**




**SIDE VIEW**

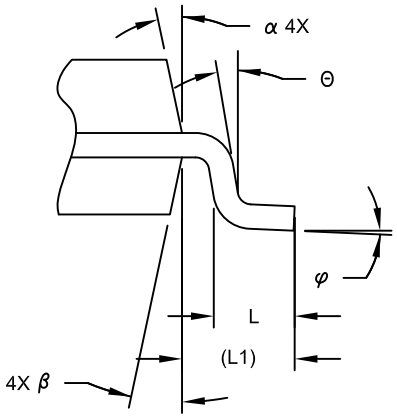



**VIEW A-A**

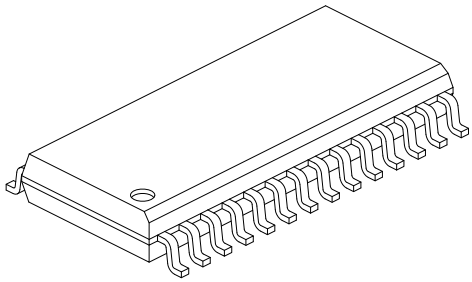
Microchip Technology Drawing C04-052C Sheet 1 of 2

### 28-Lead Plastic Small Outline (SO) - Wide, 7.50 mm Body [SOIC]

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging




VIEW C



| Units | | MILLIMETERS | | |
|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX |
| Number of Pins | N | | 28 | |
| Pitch | e | | 1.27 BSC | |
| Overall Height | A | - | - | 2.65 |
| Molded Package Thickness | A2 | 2.05 | - | - |
| Standoff § | A1 | 0.10 | - | 0.30 |
| Overall Width | E | | 10.30 BSC | |
| Molded Package Width | E1 | | 7.50 BSC | |
| Overall Length | D | | 17.90 BSC | |
| Chamfer (Optional) | h | 0.25 | - | 0.75 |
| Foot Length | L | 0.40 | - | 1.27 |
| Footprint | L1 | | 1.40 REF | |
| Lead Angle | Θ | 0° | - | - |
| Foot Angle | φ | 0° | - | 8° |
| Lead Thickness | c | 0.18 | - | 0.33 |
| Lead Width | b | 0.31 | - | 0.51 |
| Mold Draft Angle Top | α | 5° | - | 15° |
| Mold Draft Angle Bottom | β | 5° | - | 15° |

**Notes:**

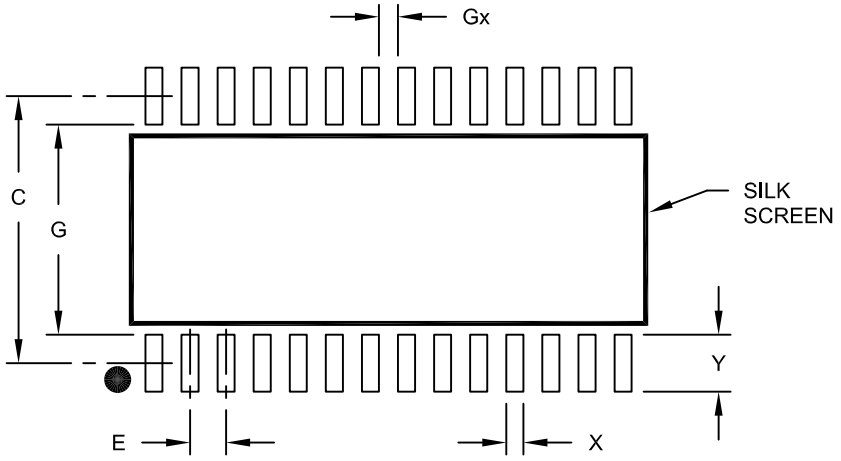
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. § Significant Characteristic
3. Dimension D does not include mold flash, protrusions or gate burrs, which shall not exceed 0.15 mm per end. Dimension E1 does not include interlead flash or protrusion, which shall not exceed 0.25 mm per side.
4. Dimensioning and tolerancing per ASME Y14.5M
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.
   REF: Reference Dimension, usually without tolerance, for information purposes only.
5. Datums A & B to be determined at Datum H.

Microchip Technology Drawing C04-052C Sheet 2 of 2


© 2006-2012 Microchip Technology Inc.

28-Lead Plastic Small Outline (SO) - Wide, 7.50 mm Body [SOIC]

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging




RECOMMENDED LAND PATTERN

| Units | | MILLIMETERS | | |
|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX |
| Contact Pitch | E | | 1.27 BSC | |
| Contact Pad Spacing | C | | 9.40 | |
| Contact Pad Width (X28) | X | | | 0.60 |
| Contact Pad Length (X28) | Y | | | 2.00 |
| Distance Between Pads | Gx | 0.67 | | |
| Distance Between Pads | G | 7.40 | | |

Notes:

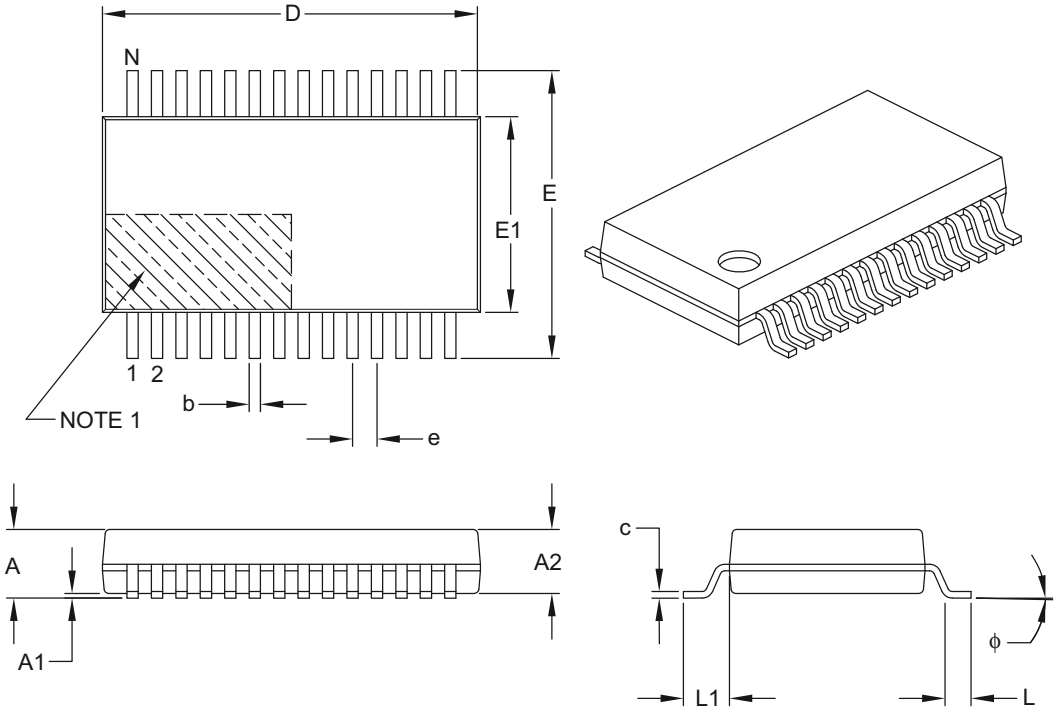
1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2052A

### 28-Lead Plastic Shrink Small Outline (SS) – 5.30 mm Body [SSOP]

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging




| Units | | MILLIMETERS | | |
|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX |
| Number of Pins | N | | 28 | |
| Pitch | e | | 0.65 BSC | |
| Overall Height | A | – | – | 2.00 |
| Molded Package Thickness | A2 | 1.65 | 1.75 | 1.85 |
| Standoff | A1 | 0.05 | – | – |
| Overall Width | E | 7.40 | 7.80 | 8.20 |
| Molded Package Width | E1 | 5.00 | 5.30 | 5.60 |
| Overall Length | D | 9.90 | 10.20 | 10.50 |
| Foot Length | L | 0.55 | 0.75 | 0.95 |
| Footprint | L1 | | 1.25 REF | |
| Lead Thickness | c | 0.09 | – | 0.25 |
| Foot Angle | ϕ | 0° | 4° | 8° |
| Lead Width | b | 0.22 | – | 0.38 |

**Notes:**

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.20 mm per side.
3. Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

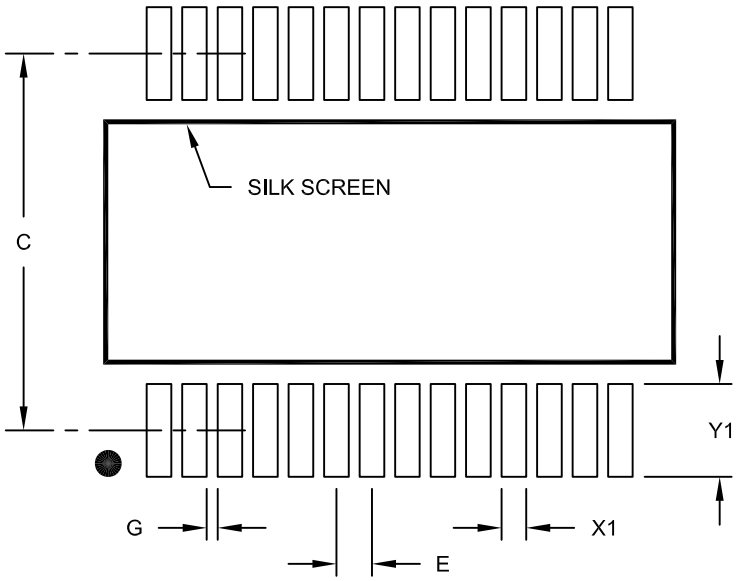
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-073B


© 2006-2012 Microchip Technology Inc.

28-Lead Plastic Shrink Small Outline (SS) - 5.30 mm Body [SSOP]

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging




RECOMMENDED LAND PATTERN

| Units | | MILLIMETERS | | |
|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX |
| Contact Pitch | E | 0.65 BSC | | |
| Contact Pad Spacing | C | | 7.20 | |
| Contact Pad Width (X28) | X1 | | | 0.45 |
| Contact Pad Length (X28) | Y1 | | | 1.75 |
| Distance Between Pads | G | 0.20 | | |

Notes:

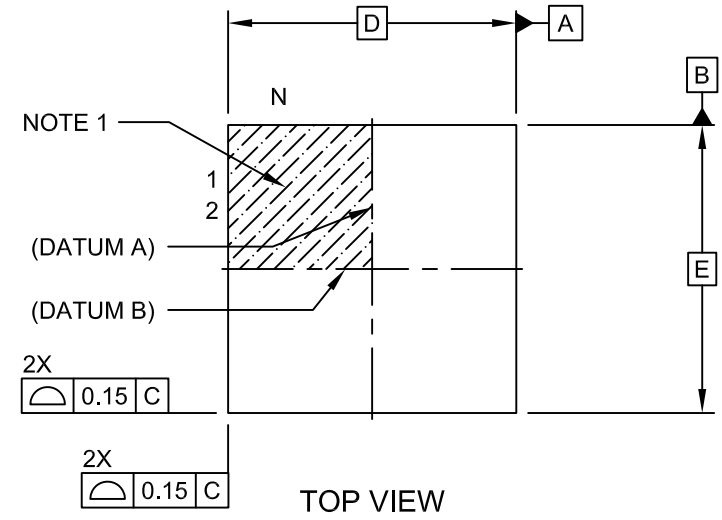
1. Dimensioning and tolerancing per ASME Y14.5M

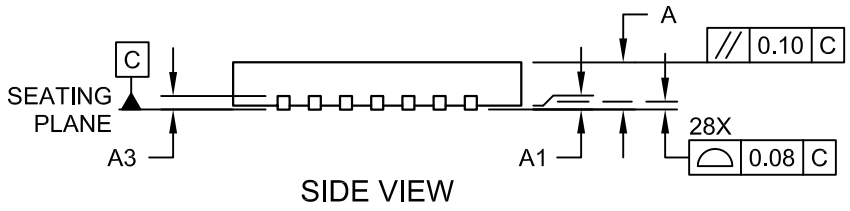
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

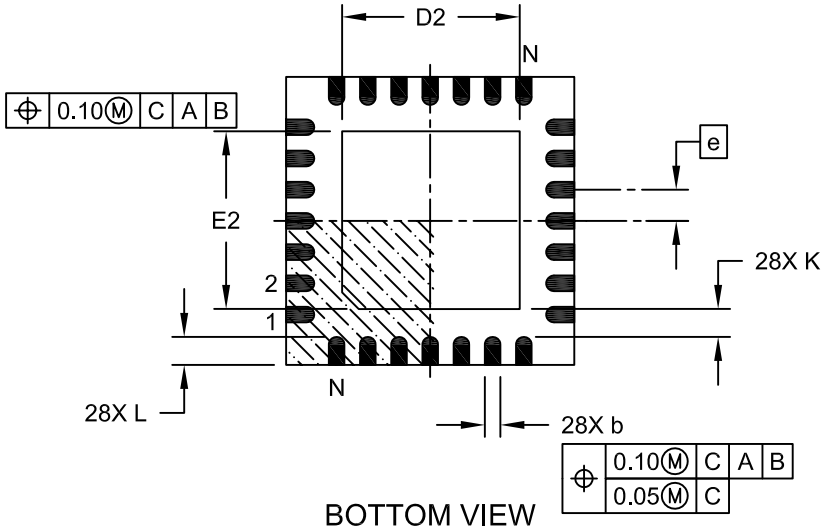
Microchip Technology Drawing No. C04-2073A

**28-Lead Plastic Quad Flat, No Lead Package (ML) - 6x6 mm Body [QFN]**
**With 0.55 mm Terminal Length**

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



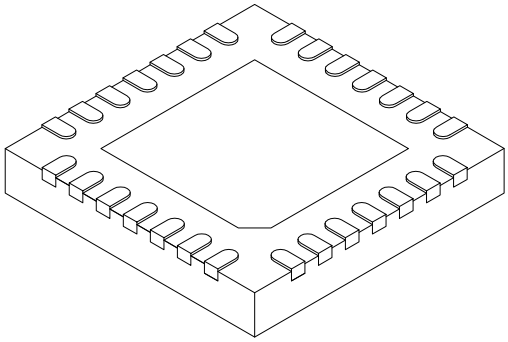




Microchip Technology Drawing C04-105C Sheet 1 of 2


© 2006-2012 Microchip Technology Inc.

## 28-Lead Plastic Quad Flat, No Lead Package (ML) - 6x6 mm Body [QFN] With 0.55 mm Terminal Length

**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 | | 28 | |
| Pitch | e | | 0.65 BSC | |
| Overall Height | A | 0.80 | 0.90 | 1.00 |
| Standoff | A1 | 0.00 | 0.02 | 0.05 |
| Terminal Thickness | A3 | | 0.20 REF | |
| Overall Width | E | | 6.00 BSC | |
| Exposed Pad Width | E2 | 3.65 | 3.70 | 4.20 |
| Overall Length | D | | 6.00 BSC | |
| Exposed Pad Length | D2 | 3.65 | 3.70 | 4.20 |
| Terminal Width | b | 0.23 | 0.30 | 0.35 |
| Terminal Length | L | 0.50 | 0.55 | 0.70 |
| Terminal-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.

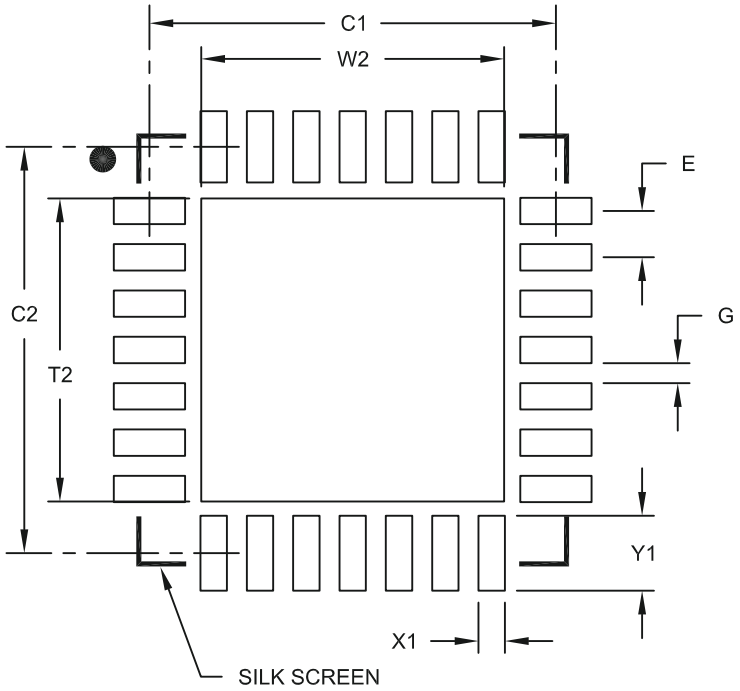
   REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-105C Sheet 2 of 2

© 2006-2012 Microchip Technology Inc.

### 28-Lead Plastic Quad Flat, No Lead Package (ML) – 6x6 mm Body [QFN] with 0.55 mm Contact Length

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging




RECOMMENDED LAND PATTERN

| Units | | MILLIMETERS | | |
|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX |
| Contact Pitch | E | | 0.65 BSC | |
| Optional Center Pad Width | W2 | | | 4.25 |
| Optional Center Pad Length | T2 | | | 4.25 |
| Contact Pad Spacing | C1 | | 5.70 | |
| Contact Pad Spacing | C2 | | 5.70 | |
| Contact Pad Width (X28) | X1 | | | 0.37 |
| Contact Pad Length (X28) | Y1 | | | 1.00 |
| Distance Between Pads | G | 0.20 | | |

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

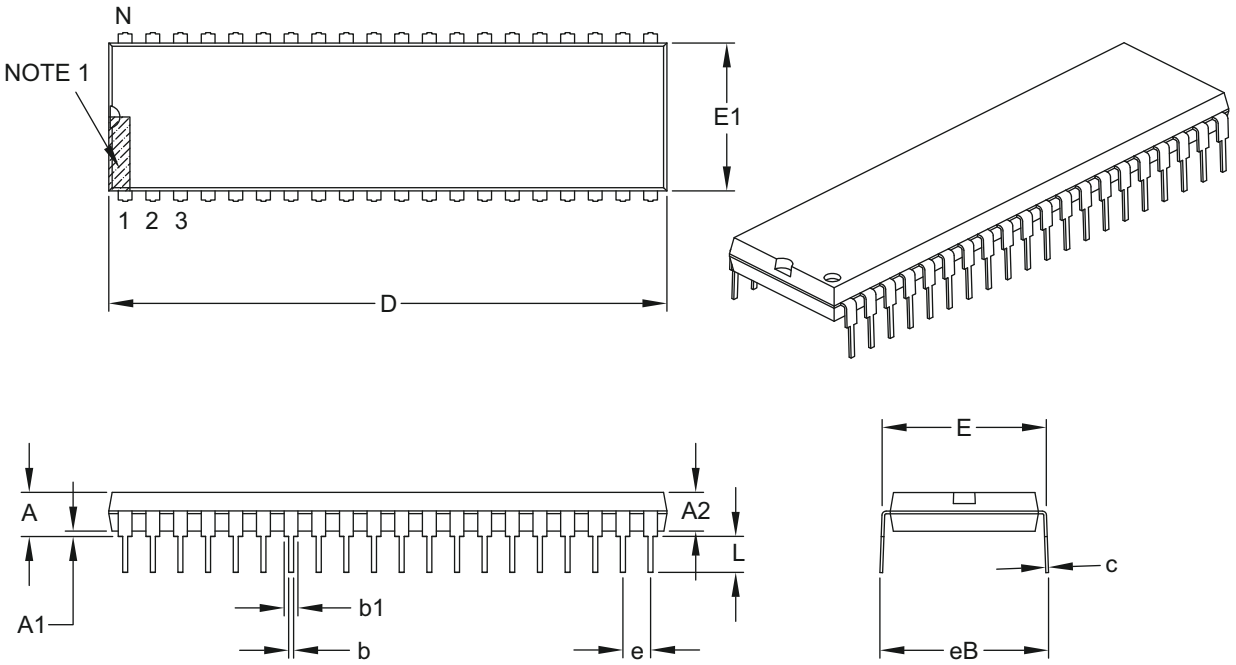
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2105A


© 2006-2012 Microchip Technology Inc.

## 40-Lead Plastic Dual In-Line (P) – 600 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 | | 40 | |
| Pitch | e | | .100 BSC | |
| Top to Seating Plane | A | – | – | .250 |
| Molded Package Thickness | A2 | .125 | – | .195 |
| Base to Seating Plane | A1 | .015 | – | – |
| Shoulder to Shoulder Width | E | .590 | – | .625 |
| Molded Package Width | E1 | .485 | – | .580 |
| Overall Length | D | 1.980 | – | 2.095 |
| Tip to Seating Plane | L | .115 | – | .200 |
| Lead Thickness | c | .008 | – | .015 |
| Upper Lead Width | b1 | .030 | – | .070 |
| Lower Lead Width | b | .014 | – | .023 |
| Overall Row Spacing § | eB | – | – | .700 |

**Notes:**

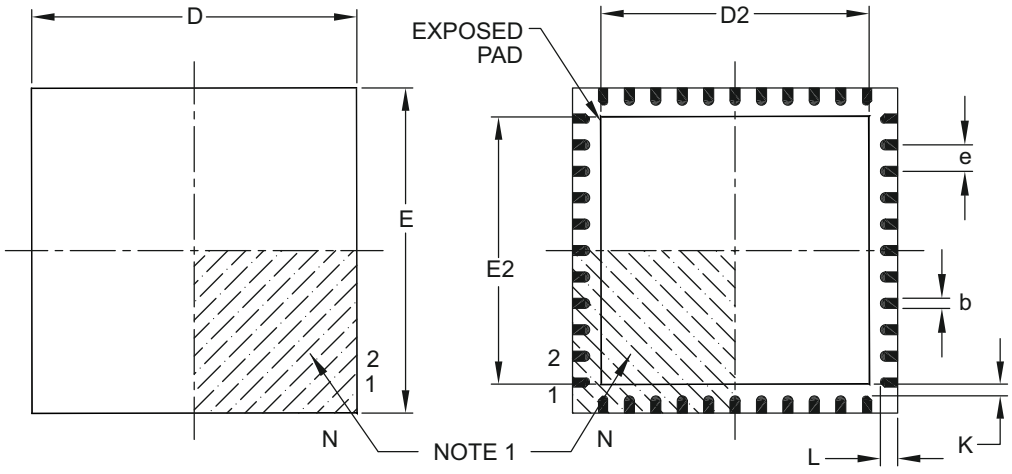
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. § Significant Characteristic.
3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.
4. Dimensioning and tolerancing per ASME Y14.5M.

   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

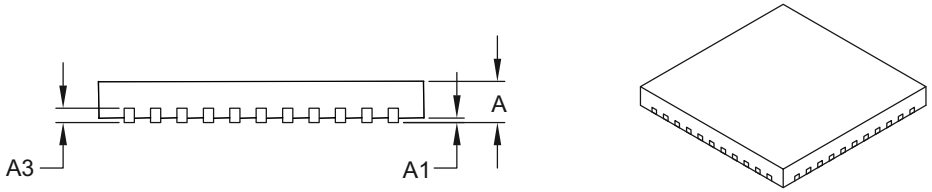
Microchip Technology Drawing C04-016B

### 44-Lead Plastic Quad Flat, No Lead Package (ML) – 8x8 mm Body [QFN]

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging




**TOP VIEW** **BOTTOM VIEW**




| Units | | MILLIMETERS | | |
|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX |
| Number of Pins | N | | 44 | |
| 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 | | 8.00 BSC | |
| Exposed Pad Width | E2 | 6.30 | 6.45 | 6.80 |
| Overall Length | D | | 8.00 BSC | |
| Exposed Pad Length | D2 | 6.30 | 6.45 | 6.80 |
| Contact Width | b | 0.25 | 0.30 | 0.38 |
| 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.

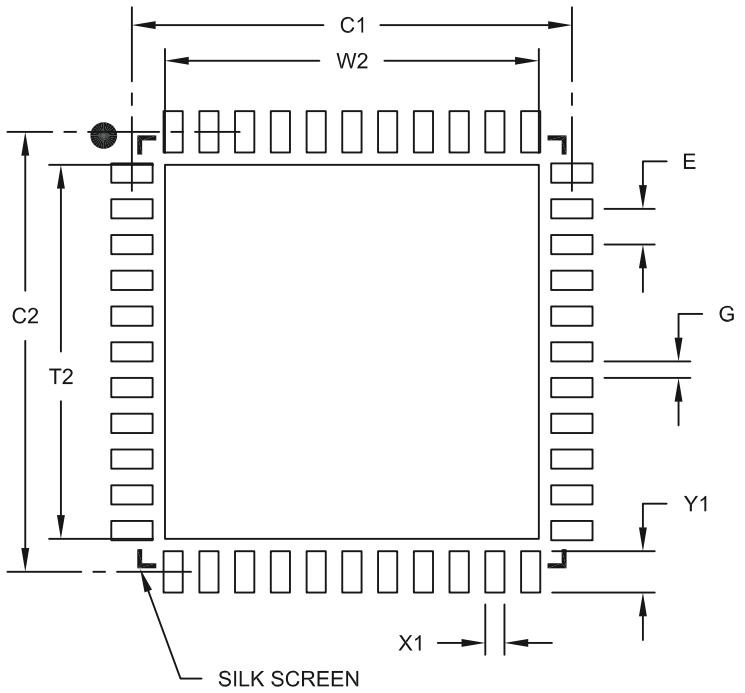
BSC: Basic Dimension. Theoretically exact value shown without tolerances.
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-103B


© 2006-2012 Microchip Technology Inc.

### 44-Lead Plastic Quad Flat, No Lead Package (ML) – 8x8 mm 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 | | | 6.80 |
| Optional Center Pad Length | T2 | | | 6.80 |
| Contact Pad Spacing | C1 | | 8.00 | |
| Contact Pad Spacing | C2 | | 8.00 | |
| Contact Pad Width (X44) | X1 | | | 0.35 |
| Contact Pad Length (X44) | Y1 | | | 0.80 |
| Distance Between Pads | G | 0.25 | | |

Notes:

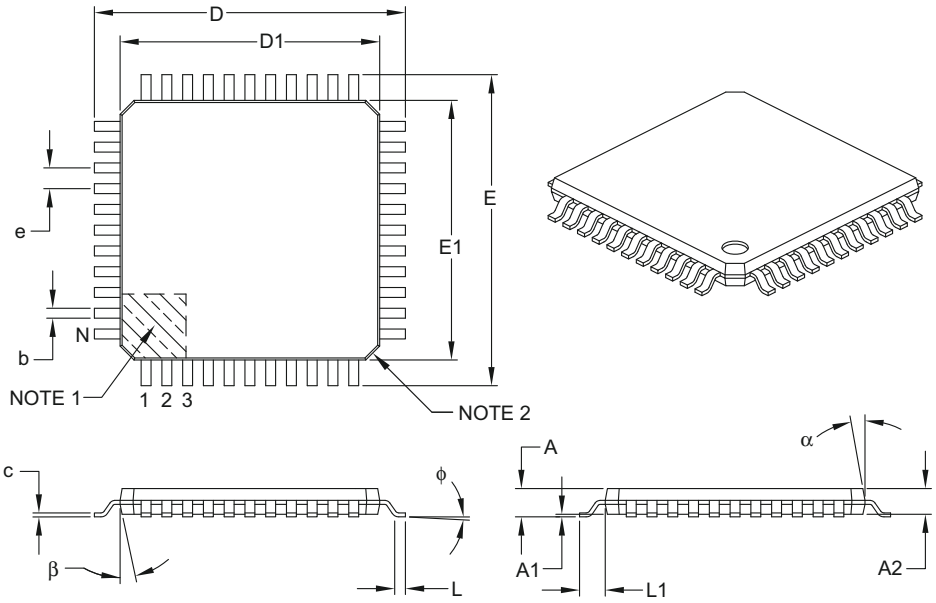
1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2103A

### 44-Lead Plastic Thin Quad Flatpack (PT) – 10x10x1 mm Body, 2.00 mm [TQFP]

**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 Leads | N | | 44 | |
| Lead Pitch | e | | 0.80 BSC | |
| Overall Height | A | – | – | 1.20 |
| Molded Package Thickness | A2 | 0.95 | 1.00 | 1.05 |
| Standoff | A1 | 0.05 | – | 0.15 |
| Foot Length | L | 0.45 | 0.60 | 0.75 |
| Footprint | L1 | | 1.00 REF | |
| Foot Angle | φ | 0° | 3.5° | 7° |
| Overall Width | E | | 12.00 BSC | |
| Overall Length | D | | 12.00 BSC | |
| Molded Package Width | E1 | | 10.00 BSC | |
| Molded Package Length | D1 | | 10.00 BSC | |
| Lead Thickness | c | 0.09 | – | 0.20 |
| Lead Width | b | 0.30 | 0.37 | 0.45 |
| Mold Draft Angle Top | α | 11° | 12° | 13° |
| Mold Draft Angle Bottom | β | 11° | 12° | 13° |

**Notes:**

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Chamfers at corners are optional; size may vary.
3. Dimensions D1 and E1 do not include mold flash or protrusions. Mold flash or protrusions 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.

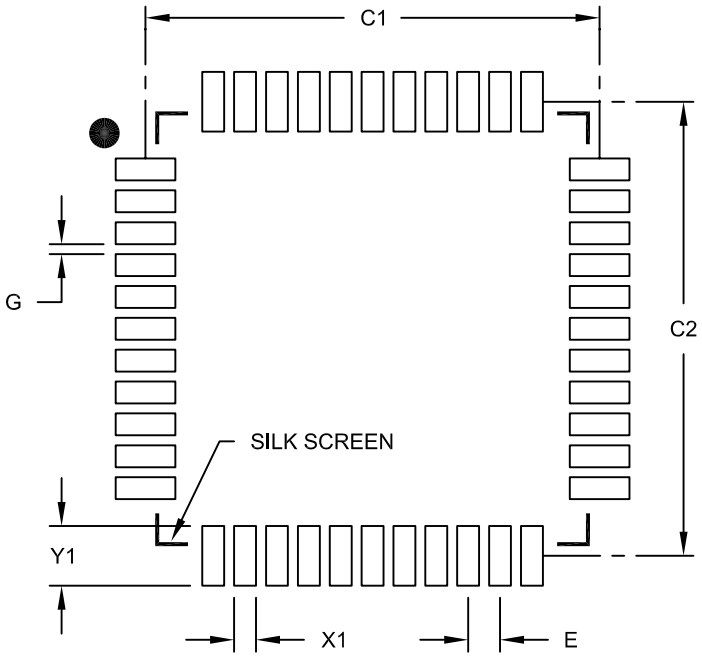
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-076B


© 2006-2012 Microchip Technology Inc.

44-Lead Plastic Thin Quad Flatpack (PT) 10X10X1 mm Body, 2.00 mm Footprint [TQFP]

**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 | | 0.80 BSC | |
| Contact Pad Spacing | C1 | | 11.40 | |
| Contact Pad Spacing | C2 | | 11.40 | |
| Contact Pad Width (X44) | X1 | | | 0.55 |
| Contact Pad Length (X44) | Y1 | | | 1.50 |
| Distance Between Pads | G | 0.25 | | |

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2076B

**NOTES:**


© 2006-2012 Microchip Technology Inc.

## APPENDIX A: DATA SHEET REVISION HISTORY

### Revision A (5/2006)

Initial release of this data sheet.

### Revision B (7/2006)

Pin Diagrams (44-Pin QFN drawing); Revised Table 2-1, Addr. 1DH (CCP2CON); Section 3.0, 3.1; Section 3.4.4.6; Table 3; Table 3-1 (ANSEL); Table 3-3 (CCP2CON); Register 3-1; Register 3.2; Register 3-3; Register 3-4; Register 3-9; Register 3-10; Register 3-11; Register 3-12; Register 3-14; Table 3-5 (ANSEL); Figure 3-5; Figure 3-11; Figure 8-2; Figure 8-3; Figure 9-1; Register 9-1; Section 9.1.4; Example 10-4; Figure 11-5; Table 11-5 (P1M); Section 11.5.2; Section 11.5.7, Number 4; Table 11-7 (CCP2CON); Section 12.3.1 (Para. 3); Figure 12-6 (Title); Sections 14.2, 14.3 and 14.4 DC Characteristics (Max); Table 14-4 (OSCCON); Section 14.3 (TMR0); Section 14.3.2 (TMR0).

### Revision C

Section 19.0 Packaging Information: Replaced package drawings and added note.
Added PIC16F882 part number.
Replaced PICmicro with PIC.

### Revision D

Replaced Package Drawings (Rev. AM); Replaced Development Support Section; Revised Product ID Section.

### Revision E (01/2008)

Added Char Data; Removed Preliminary status; Revised Device Table (PIC16F882, I/O); Revised the following: Pin Diagram 44 TQFP, pin 30; Table 5, I/O RA7; Table 1-1, RA1 and RA4; Section 2.2.1; Register 2-3, INTCON; Example 3-1; Section 3.2.2; Example 3-2; Figure 6-1; Section 6.2.2; Section 6.6; Section 8.10.3; Table 9-1; Equation 11-1; Added Figure 11-14 and renumbered remaining Figures; Register 11-3; Register 13-3; Section 14.0; Section 14.1; Section 14.9; Section 14.10; Section 17.0; Updated Package Drawings.

### Revision F (04/2009)

Revised Product ID: Removed ‘F’ (std. voltage range) from part numbers; Revised Figure 6-1: Timer1 Block Diagram; Revised Figure 8-3, Comparator C2 Block Diagram; Added note to Section 8.10.3; Revised Section 8.10.7.

### Revision G (10/2012)

Updated data sheet to new format; Updated Register 13-1 and Register 13-2; Updated the Packaging Information section; Updated the Product Identification System section; Other minor corrections.

# APPENDIX B: MIGRATING FROM OTHER PIC® DEVICES

This discusses some of the issues in migrating from other PIC devices to the PIC16F88X Family of devices.

## B.1 PIC16F87X to PIC16F88X

**TABLE B-1: FEATURE COMPARISON**

| Feature | PIC16F87X | PIC16F88X |
|---|---|---|
| Max Operating Speed | 20 MHz | 20 MHz |
| Max Program Memory (Words) | 8192 | 8192 |
| SRAM (bytes) | 368 | 368 |
| A/D Resolution | 10-bit | 10-bit |
| Data EEPROM (Bytes) | 256 | 256 |
| Timers (8/16-bit) | 2/1 | 2/1 |
| Oscillator Modes | 4 | 8 |
| Brown-out Reset | Y | Y (2.1V/4V) |
| Software Control Option of WDT/BOR | N | Y |
| Internal Pull-ups | RB<7:4> | RB<7:0>, $\overline{MCLR}$ |
| Interrupt-on-change | RB<7:4> | RB<7:0> |
| Comparator | 2 | 2 |
| References | $CV_{REF}$ | $CV_{REF}$ and VP6 |
| ECCP/CCP | 0/2 | 1/1 |
| Ultra Low-Power Wake-Up | N | Y |
| Extended WDT | N | Y |
| INTOSC Frequencies | N | 32 kHz-8 MHz |
| Clock Switching | N | Y |
| MSSP | Standard | w/Slave Address Mask |
| USART | AUSART | EUSART |
| ADC Channels | 8 | 14 |

**Note:** This device has been designed to perform to the parameters of its data sheet. It has been tested to an electrical specification designed to determine its conformance with these parameters. Due to process differences in the manufacture of this device, this device may have different performance characteristics than its earlier version. These differences may cause this device to perform differently in your application than the earlier version of this device.


© 2006-2012 Microchip Technology Inc.

# INDEX


## A

A/D
- Specifications ... 267, 268

Absolute Maximum Ratings ... 249
AC Characteristics
- Industrial and Extended ... 259
- Load Conditions ... 258

ACKSTAT ... 202
ACKSTAT Status Flag ... 202
ADC ... 103
- Acquisition Requirements ... 111
- Associated Registers ... 113
- Block Diagram ... 103
- Calculating Acquisition Time ... 111
- Channel Selection ... 104
- Configuration ... 104
- Configuring Interrupt ... 107
- Conversion Clock ... 104
- Conversion Procedure ... 107
- Internal Sampling Switch (RSS) Impedance ... 111
- Interrupts ... 105
- Operation ... 106
- Operation During Sleep ... 106
- Port Configuration ... 104
- Reference Voltage (VREF) ... 104
- Result Formatting ... 106
- Source Impedance ... 111
- Special Event Trigger ... 106
- Starting an A/D Conversion ... 106

ADCON0 Register ... 108
ADCON1 Register ... 109
ADRESH Register (ADFM = 0) ... 110
ADRESH Register (ADFM = 1) ... 110
ADRESL Register (ADFM = 0) ... 110
ADRESL Register (ADFM = 1) ... 110
Analog Input Connection Considerations ... 95
Analog-to-Digital Converter. *See* ADC
ANSEL Register ... 42
ANSELH Register ... 50
Assembler
- MPASM Assembler ... 246

## B

Baud Rate Generator ... 199
BAUDCTL Register ... 166
BF ... 202
BF Status Flag ... 202
Block Diagrams
- (CCP) Capture Mode Operation ... 130
- ADC ... 103
- ADC Transfer Function ... 112
- Analog Input Model ... 95, 112
- Auto-Shutdown ... 145
- Baud Rate Generator ... 199
- CCP PWM ... 132
- Clock Source ... 65
- Comparator C1 ... 90
- Comparator C1 and ADC Voltage Reference ... 100
- Comparator C2 ... 90
- Compare ... 131
- Crystal Operation ... 68
- EUSART Receive ... 156
- EUSART Transmit ... 155
- External RC Mode ... 69
- Fail-Safe Clock Monitor (FSCM) ... 75
- In-Circuit Serial Programming Connections ... 232
- Interrupt Logic ... 225
- MSSP ($I^2C$ Master Mode) ... 197
- MSSP ($I^2C$ Mode) ... 193
- MSSP (SPI Mode) ... 187
- On-Chip Reset Circuit ... 216
- PIC16F883/886 ... 16
- PIC16F884/887 ... 17
- PWM (Enhanced) ... 136
- RA0 Pins ... 44
- RA1 Pin ... 45
- RA2 Pin ... 45
- RA3 Pin ... 46
- RA4 Pin ... 46
- RA5 Pin ... 47
- RA6 Pin ... 47
- RA7 Pin ... 48
- RB0, RB1, RB2, RB3 Pins ... 52
- RB4, RB5, RB6, RB7 Pins ... 54
- RC0 Pin ... 56
- RC1 Pin ... 56
- RC2 Pin ... 56
- RC3 Pin ... 57
- RC4 Pin ... 57
- RC5 Pin ... 57
- RC6 Pin ... 58
- RC7 Pin ... 58
- RD0, RD1, RD2, RD3, RD4 Pins ... 60
- RD5, RD6, RD7 Pins ... 60
- RE3 Pin ... 62
- Resonator Operation ... 68
- Timer1 ... 81
- Timer2 ... 87
- TMR0/WDT Prescaler ... 77
- Watchdog Timer (WDT) ... 228

Break Character (12-bit) Transmit and Receive ... 173
BRG ... 199
Brown-out Reset (BOR) ... 218
- Associated ... 219
- Specifications ... 263
- Timing and Characteristics ... 262

Bus Collision During a Repeated Start Condition ... 210
Bus Collision During a Start Condition ... 208
Bus Collision During a Stop Condition ... 211

## C

C Compilers
- MPLAB C18 ... 246

Capture Module. *See* Enhanced Capture/Compare/PWM (ECCP)
Capture/Compare/PWM (CCP)
- Associated Registers w/ Capture, Compare and Timer1 ... 153
- Associated Registers w/ PWM and Timer2 ... 153
- Capture Mode ... 130
- CCP Pin Configuration ... 130
- Compare Mode ... 131
  - CCP Pin Configuration ... 131
  - Software Interrupt Mode ... 130, 131
  - Special Event Trigger ... 131
  - Timer1 Mode Selection ... 130, 131
- Prescaler ... 130

PWM Mode ..... 132
Duty Cycle ..... 133
Effects of Reset ..... 135
Example PWM Frequencies and Resolutions, 20 MHZ ..... 134
Example PWM Frequencies and Resolutions, 8 MHz ..... 134
Operation in Sleep Mode ..... 135
Setup for Operation ..... 135
System Clock Frequency Changes ..... 135
PWM Period ..... 132
Setup for PWM Operation ..... 135
Timer Resources ..... 129
CCP1CON (Enhanced) Register ..... 128
CCP2CON Register ..... 129
Clock Accuracy with Asynchronous Operation ..... 164
Clock Sources
External Modes ..... 67
EC ..... 67
HS ..... 68
LP ..... 68
OST ..... 67
RC ..... 69
XT ..... 68
Internal Modes ..... 69
Frequency Selection ..... 71
HFINTOSC ..... 69
HFINTOSC/LFINTOSC Switch Timing ..... 71
INTOSC ..... 69
INTOSCIO ..... 69
LFINTOSC ..... 71
Clock Switching ..... 73
CM1CON0 Register ..... 93
CM2CON0 Register ..... 94
CM2CON1 Register ..... 96
Code Examples
A/D Conversion ..... 107
Assigning Prescaler to Timer0 ..... 78
Assigning Prescaler to WDT ..... 78
Changing Between Capture Prescalers ..... 130
Indirect Addressing ..... 39
Initializing PORTA ..... 41
Initializing PORTB ..... 49
Initializing PORTC ..... 55
Initializing PORTD ..... 59
Initializing PORTE ..... 61
Loading the SSPBUF Register ..... 188
Saving STATUS and W Registers in RAM ..... 227
Ultra Low-Power Wake-up Initialization ..... 43
Write Verify ..... 124
Writing to Flash Program Memory ..... 123
Code Protection ..... 231
Comparator
C2OUT as T1 Gate ..... 82, 96
Effects of a Reset ..... 93
Operation ..... 89
Operation During Sleep ..... 93
Response Time ..... 91
Specifications ..... 266
Synchronizing COUT w/Timer1 ..... 96
Comparator Module ..... 89
Associated Registers ..... 102
C1 Output State Versus Input Conditions ..... 91
Comparator Voltage Reference (CVREF) ..... 99
Effects of a Reset ..... 93
Comparator Voltage Reference (CVREF)
Response Time ..... 91
Comparator Voltage Reference (CVREF)
Specifications ..... 266
Compare Module. *See* Enhanced Capture/Compare/PWM (ECCP)
CONFIG1 Register ..... 214
CONFIG2 Register ..... 215
Configuration Bits ..... 214
CPU Features ..... 213
Customer Change Notification Service ..... 333
Customer Notification Service ..... 333
Customer Support ..... 333


## D


Data EEPROM Memory ..... 115
Associated Registers ..... 125
Code Protection ..... 124
Reading ..... 118
Writing ..... 118
Data Memory ..... 24
DC Characteristics
Extended ..... 254
Industrial ..... 253
Industrial and Extended ..... 251, 252, 255
Development Support ..... 245
Device Overview ..... 15


## E


ECCP. *See* Enhanced Capture/Compare/PWM
ECCPAS Register ..... 146
EEADR Register ..... 116
EEADR Registers ..... 115
EEADRH Registers ..... 115
EECON1 Register ..... 115, 117
EECON2 Register ..... 115
EEDAT Register ..... 116
EEDATH Register ..... 116
EEPROM Data Memory
Avoiding Spurious Write ..... 124
Write Verify ..... 124
Effects of Reset
PWM mode ..... 135
Electrical Specifications ..... 249
Enhanced Capture/Compare/PWM ..... 127
Enhanced Capture/Compare/PWM (ECCP)
Enhanced PWM Mode ..... 136
Auto-Restart ..... 147
Auto-shutdown ..... 145
Direction Change in Full-Bridge Output Mode ..... 142
Full-Bridge Application ..... 140
Full-Bridge Mode ..... 140
Half-Bridge Application ..... 139
Half-Bridge Application Examples ..... 148
Half-Bridge Mode ..... 139
Output Relationships (Active-High and Active-Low) ..... 137
Output Relationships Diagram ..... 138
Programmable Dead Band Delay ..... 148
Shoot-through Current ..... 148
Start-up Considerations ..... 144
Specifications ..... 265
Timer Resources ..... 127
Enhanced Universal Synchronous Asynchronous Receiver Transmitter (EUSART) ..... 155
Errata ..... 13



© 2006-2012 Microchip Technology Inc.

EUSART ..... 155
Associated Registers
Baud Rate Generator ..... 167
Asynchronous Mode ..... 157
12-bit Break Transmit and Receive ..... 173
Associated Registers
Receive ..... 163
Transmit ..... 159
Auto-Wake-up on Break ..... 172
Baud Rate Generator (BRG) ..... 167
Clock Accuracy ..... 164
Receiver ..... 160
Setting up 9-bit Mode with Address Detect ..... 162
Transmitter ..... 157
Baud Rate Generator (BRG)
Auto Baud Rate Detect ..... 171
Baud Rate Error, Calculating ..... 167
Baud Rates, Asynchronous Modes ..... 168
Formulas ..... 167
High Baud Rate Select (BRGH Bit) ..... 167
Synchronous Master Mode ..... 175, 179
Associated Registers
Receive ..... 178
Transmit ..... 176
Reception ..... 177
Requirements, Synchronous Receive ..... 270
Requirements, Synchronous Transmission ..... 270
Timing Diagram, Synchronous Receive ..... 270
Timing Diagram, Synchronous Transmission ... 270
Transmission ..... 175
Synchronous Slave Mode
Associated Registers
Receive ..... 180
Transmit ..... 179
Reception ..... 180
Transmission ..... 179

## F

Fail-Safe Clock Monitor ..... 75
Fail-Safe Condition Clearing ..... 75
Fail-Safe Detection ..... 75
Fail-Safe Operation ..... 75
Reset or Wake-up from Sleep ..... 75
Firmware Instructions ..... 235
Flash Program Memory ..... 115
Writing ..... 121
Fuses. *See* Configuration Bits

## G

General Call Address Support ..... 196
General Purpose Register File ..... 24

## I

$I^2C$ (MSSP Module)
ACK Pulse ..... 193, 194
Addressing ..... 194
Read/Write Bit Information (R/$\overline{W}$ Bit) ..... 194
Reception ..... 194
Serial Clock (RC3/SCK/SCL) ..... 194
Slave Mode ..... 193
Transmission ..... 194
$I^2C$ Master Mode Reception ..... 202
$I^2C$ Master Mode Repeated Start Condition Timing ..... 201
$I^2C$ Module
Acknowledge Sequence Timing ..... 205
Baud Rate Generator ..... 199
BRG Block Diagram ..... 199
BRG Reset Due to SDA Arbitration During Start Condition ..... 209
BRG Timing ..... 199
Bus Collision
Acknowledge ..... 207
Repeated Start Condition ..... 210
Repeated Start Condition Timing (Case1) ..... 210
Repeated Start Condition Timing (Case2) ..... 210
Start Condition ..... 208
Start Condition Timing ..... 208, 209
Stop Condition ..... 211
Stop Condition Timing (Case 1) ..... 211
Stop Condition Timing (Case 2) ..... 211
Bus Collision timing ..... 207
Clock Arbitration ..... 206
Clock Arbitration Timing (Master Transmit) ..... 206
Effect of a Reset ..... 206
General Call Address Support ..... 196
Master Mode ..... 197
Master Mode 7-bit Reception Timing ..... 204
Master Mode Operation ..... 198
Master Mode Start Condition Timing ..... 200
Master Mode Support ..... 197
Master Mode Transmission ..... 202
Master Mode Transmit Sequence ..... 198
Multi-Master Mode ..... 207
Repeat Start Condition Timing Waveform ..... 201
Sleep Operation ..... 206
Stop Condition Receive or Transmit Timing ..... 206
Stop Condition Timing ..... 205
Waveforms for 7-bit Reception ..... 195
Waveforms for 7-bit Transmission ..... 195
ID Locations ..... 231
In-Circuit Debugger ..... 233
In-Circuit Serial Programming (ICSP) ..... 231
Indirect Addressing, INDF and FSR registers ..... 39
Instruction Format ..... 235
Instruction Set ..... 235
ADDLW ..... 237
ADDWF ..... 237
ANDLW ..... 237
ANDWF ..... 237
MOVF ..... 240
BCF ..... 237
BSF ..... 237
BTFSC ..... 237
BTFSS ..... 238
CALL ..... 238
CLRF ..... 238
CLRW ..... 238
CLRWDT ..... 238
COMF ..... 238
DECF ..... 238
DECFSZ ..... 239
GOTO ..... 239
INCF ..... 239
INCFSZ ..... 239
IORLW ..... 239
IORWF ..... 239
MOVLW ..... 240
MOVWF ..... 240
NOP ..... 240
RETFIE ..... 241
RETLW ..... 241

RETURN ..... 241
RLF ..... 242
RRF ..... 242
SLEEP ..... 242
SUBLW ..... 242
SUBWF ..... 242
SWAPF ..... 243
XORLW ..... 243
XORWF ..... 243
Summary Table ..... 236
INTCON Register ..... 33
Inter-Integrated Circuit. *See* I²C
Internal Oscillator Block ..... 260
INTOSC
Specifications ..... 261
Internal Sampling Switch (RSS) Impedance ..... 111
Internet Address ..... 333
Interrupts ..... 224
ADC ..... 107
Associated Registers ..... 226
Context Saving ..... 227
Interrupt-on-Change ..... 49
PORTB Interrupt-on-Change ..... 225
RB0/INT ..... 224
Timer0 ..... 225
TMR1 ..... 83
INTOSC
Specifications ..... 260
INTOSC Specifications ..... 260, 261
IOCB Register ..... 51

**L**

Load Conditions ..... 258

**M**

Master Mode ..... 197
Master Mode Support ..... 197
Master Synchronous Serial Port. *See* MSSP
MCLR ..... 217
Internal ..... 217
Memory Organization ..... 23
Data ..... 24
Program ..... 23
Microchip Internet Web Site ..... 333
Migrating from other PICmicro Devices ..... 324
MPLAB ASM30 Assembler, Linker, Librarian ..... 246
MPLAB Integrated Development Environment Software ..... 245
MPLAB PM3 Device Programmer ..... 248
MPLAB REAL ICE In-Circuit Emulator System ..... 247
MPLINK Object Linker/MPLIB Object Librarian ..... 246
MSSP ..... 183
Block Diagram (SPI Mode) ..... 187
I²C Mode. *See* I²C
SPI Mode ..... 187
SPI Mode. *See* SPI
MSSP Module
Control Registers ..... 183
I²C Operation ..... 193
SPI Master Mode ..... 189
SPI Slave Mode ..... 190
Multi-Master Communication, Bus Collision and Bus Arbitration ..... 207
Multi-Master Mode ..... 207

**O**

OPCODE Field Descriptions ..... 235
OPTION Register ..... 32
OPTION_REG Register ..... 79
OSCCON Register ..... 66
Oscillator
Associated Registers ..... 76, 85, 229
Oscillator Module ..... 65
EC ..... 65
HFINTOSC ..... 65
HS ..... 65
INTOSC ..... 65
INTOSCIO ..... 65
LFINTOSC ..... 65
LP ..... 65
RC ..... 65
RCIO ..... 65
XT ..... 65
Oscillator Parameters ..... 260
Oscillator Specifications ..... 259
Oscillator Start-up Timer (OST)
Specifications ..... 263
Oscillator Switching
Fail-Safe Clock Monitor ..... 75
Two-Speed Clock Start-up ..... 73
OSCTUNE Register ..... 70

**P**

P1A/P1B/P1C/P1D. *See* Enhanced Capture/Compare/PWM (ECCP) ..... 136
Packaging ..... 305
Marking ..... 305, 306, 307
PDIP Details ..... 308
PCL and PCLATH ..... 39
Stack ..... 39
PCON Register ..... 38, 219
PIE1 Register ..... 34
PIE2 Register ..... 35
Pin Diagram
PIC16F883/886, 28-pin (PDIP, SOIC, SSOP) ..... 3
PIC16F883/886, 28-pin (QFN) ..... 5
PIC16F884/887, 40-Pin (PDIP) ..... 7
PIC16F884/887, 44-pin (QFN) ..... 9
PIC16F884/887, 44-pin (TQFP) ..... 11
Pinout Descriptions
PIC16F883/886 ..... 18
PIC16F884/887 ..... 20
PIR1 Register ..... 36
PIR2 Register ..... 37
PORTA ..... 41
Additional Pin Functions ..... 42
ANSEL Register ..... 42
Ultra Low-Power Wake-up ..... 42, 43
Associated Registers ..... 48
Pin Descriptions and Diagrams ..... 44
RA0 ..... 44
RA1 ..... 45
RA2 ..... 45
RA3 ..... 46
RA4 ..... 46
RA5 ..... 47
RA6 ..... 47
RA7 ..... 48
Specifications ..... 261
PORTA Register ..... 41
PORTB ..... 49
Additional Pin Functions ..... 49
ANSELH Register ..... 49



© 2006-2012 Microchip Technology Inc.

Weak Pull-up ..... 49
Associated Registers ..... 54
Interrupt-on-Change ..... 49
P1B/P1C/P1D.*See* Enhanced Capture/Compare/PWM+ (ECCP+) ..... 49
Pin Descriptions and Diagrams ..... 52
RB0 ..... 52
RB1 ..... 52
RB2 ..... 52
RB3 ..... 52
RB4 ..... 53
RB5 ..... 53
RB6 ..... 53
RB7 ..... 53
PORTB Register ..... 50
PORTC ..... 55
Associated Registers ..... 58
P1A.*See* Enhanced Capture/Compare/PWM+ (ECCP+) ..... 55
RC0 ..... 56
RC1 ..... 56
RC2 ..... 56
RC3 ..... 57
RC3 Pin ..... 194
RC4 ..... 57
RC5 ..... 57
RC6 ..... 58
RC7 ..... 58
Specifications ..... 261
PORTC Register ..... 55
PORTD ..... 59
Associated Registers ..... 60
P1B/P1C/P1D.*See* Enhanced Capture/Compare/PWM+ (ECCP+) ..... 59
RD0, RD1, RD2, RD3, RD4 ..... 60
RD5 ..... 60
RD6 ..... 60
RD7 ..... 60
PORTD Register ..... 59
PORTE ..... 61
Associated Registers ..... 63
RE0 ..... 62
RE1 ..... 62
RE2 ..... 62
RE3 ..... 62
PORTE Register ..... 61
Power-Down Mode (Sleep) ..... 230
Power-on Reset (POR) ..... 217
Power-up Timer (PWRT) ..... 217
Specifications ..... 263
Precision Internal Oscillator Parameters ..... 261
Prescaler
Shared WDT/Timer0 ..... 78
Switching Prescaler Assignment ..... 78
Program Memory ..... 23
Map and Stack ..... 23
Map and Stack (PIC16F883/884) ..... 23
Map and Stack (PIC16F886/887) ..... 23
Programming, Device Instructions ..... 235
PSTRCON Register ..... 150
Pulse Steering ..... 150
PWM (ECCP Module)
Pulse Steering ..... 150
Steering Synchronization ..... 152
PWM Mode. *See* Enhanced Capture/Compare/PWM ..... 136
PWM1CON Register ..... 149


## R


RCREG ..... 162
RCSTA Register ..... 165
Reader Response ..... 334
Read-Modify-Write Operations ..... 235
Register
RCREG Register ..... 171
Registers
ADCON0 (ADC Control 0) ..... 108
ADCON1 (ADC Control 1) ..... 109
ADRESH (ADC Result High) with ADFM = 0) ..... 110
ADRESH (ADC Result High) with ADFM = 1) ..... 110
ADRESL (ADC Result Low) with ADFM = 0) ..... 110
ADRESL (ADC Result Low) with ADFM = 1) ..... 110
ANSEL (Analog Select) ..... 42
ANSELH (Analog Select High) ..... 50
BAUDCTL (Baud Rate Control) ..... 166
CCP1CON (Enhanced CCP1 Control) ..... 128
CCP2CON (CCP2 Control) ..... 129
CM1CON0 (C1 Control) ..... 93
CM2CON0 (C2 Control) ..... 94
CM2CON1 (C2 Control) ..... 96
CONFIG1 (Configuration Word Register 1) ..... 214
CONFIG2 (Configuration Word Register 2) ..... 215
ECCPAS (Enhanced CCP Auto-shutdown Control) . 146
EEADR (EEPROM Address) ..... 116
EECON1 (EEPROM Control 1) ..... 117
EEDAT (EEPROM Data) ..... 116
EEDATH (EEPROM Data) ..... 116
INTCON (Interrupt Control) ..... 33
IOCB (Interrupt-on-Change PORTB) ..... 51
OPTION_REG (OPTION) ..... 32, 79
OSCCON (Oscillator Control) ..... 66
OSCTUNE (Oscillator Tuning) ..... 70
PCON (Power Control Register) ..... 38
PCON (Power Control) ..... 219
PIE1 (Peripheral Interrupt Enable 1) ..... 34
PIE2 (Peripheral Interrupt Enable 2) ..... 35
PIR1 (Peripheral Interrupt Register 1) ..... 36
PIR2 (Peripheral Interrupt Request 2) ..... 37
PORTA ..... 41
PORTB ..... 50
PORTC ..... 55
PORTD ..... 59
PORTE ..... 61
PSTRCON (Pulse Steering Control) ..... 150
PWM1CON (Enhanced PWM Control) ..... 149
RCSTA (Receive Status and Control) ..... 165
Reset Values ..... 221
Reset Values (special registers) ..... 223
Special Function Register Map
PIC16F883/884 ..... 25, 26
PIC16F886/887 ..... 27
Special Function Registers ..... 24
Special Register Summary
Bank 0 ..... 28
Bank 1 ..... 29
Bank 2 ..... 30
Bank 3 ..... 30
SRCON (SR Latch Control) ..... 98
SSPCON (MSSP Control 1) ..... 185
SSPCON2 (SSP Control 2) ..... 186
SSPMSK (SSP Mask) ..... 212
SSPSTAT (SSP Status) ..... 184

STATUS ..... 31
T1CON ..... 84
T2CON ..... 88
TRISA (Tri-State PORTA) ..... 41
TRISB (Tri-State PORTB) ..... 50
TRISC (Tri-State PORTC) ..... 55
TRISD (Tri-State PORTD) ..... 59
TRISE (Tri-State PORTE) ..... 61
TXSTA (Transmit Status and Control) ..... 164
VRCON (Voltage Reference Control) ..... 102
WDTCON (Watchdog Timer Control) ..... 229
WPUB (Weak Pull-up PORTB) ..... 51
Reset ..... 216
Revision History ..... 323

**S**

SCK ..... 187
SDI ..... 187
SDO ..... 187
Serial Clock, SCK ..... 187
Serial Data In, SDI ..... 187
Serial Data Out, SDO ..... 187
Serial Peripheral Interface. *See* SPI
Shoot-through Current ..... 148
Slave Mode General Call Address Sequence ..... 196
Slave Select Synchronization ..... 190
Slave Select, $\overline{SS}$ ..... 187
Sleep ..... 230
Wake-up ..... 230
Wake-up Using Interrupts ..... 230
Software Simulator (MPLAB SIM) ..... 247
SPBRG ..... 167
SPBRGH ..... 167
Special Event Trigger ..... 106
Special Function Registers ..... 24
SPI
Master Mode ..... 189
Serial Clock ..... 187
Serial Data In ..... 187
Serial Data Out ..... 187
Slave Select ..... 187
SPI clock ..... 189
SPI Mode ..... 187
SPI Bus Modes ..... 192
SPI Mode
Associated Registers with SPI Operation ..... 192
Bus Mode Compatibility ..... 192
Effects of a Reset ..... 192
Enabling SPI I/O ..... 188
Operation ..... 187
Sleep Operation ..... 192
SPI Module
Slave Mode ..... 190
Slave Select Synchronization ..... 190
Slave Synchronization Timing ..... 190
Slave Timing with CKE = 0 ..... 191
Slave Timing with CKE = 1 ..... 191
$\overline{SR}$CON Register ..... 98
$\overline{SS}$ ..... 187
SSP
SSPBUF ..... 189
SSPSR ..... 189
SSPCON Register ..... 185
SSPCON2 Register ..... 186
SSPMSK Register ..... 212
SSPOV ..... 202
SSPOV Status Flag ..... 202
SSPSTAT Register ..... 184
R/$\overline{W}$ Bit ..... 194
STATUS Register ..... 31

**T**

T1CON Register ..... 84
T2CON Register ..... 88
Thermal Considerations ..... 257
Time-out Sequence ..... 219
Timer0 ..... 77
Associated Registers ..... 79
External Clock ..... 78
Interrupt ..... 79
Operation ..... 77, 81
Specifications ..... 264
T0CKI ..... 78
Timer1 ..... 81
Associated Registers ..... 85
Asynchronous Counter Mode ..... 82
Reading and Writing ..... 82
Interrupt ..... 83
Modes of Operation ..... 81
Operation During Sleep ..... 83
Oscillator ..... 82
Prescaler ..... 82
Specifications ..... 264
Timer1 Gate
Inverting Gate ..... 82
Selecting Source ..... 82, 96
SR Latch ..... 97
Synchronizing COUT w/Timer1 ..... 96
TMR1H Register ..... 81
TMR1L Register ..... 81
Timer2
Associated Registers ..... 88
Timers
Timer1
T1CON ..... 84
Timer2
T2CON ..... 88
Timing Diagrams
A/D Conversion ..... 269
A/D Conversion (Sleep Mode) ..... 269
Acknowledge Sequence Timing ..... 205
Asynchronous Reception ..... 162
Asynchronous Transmission ..... 158
Asynchronous Transmission (Back to Back) ..... 159
Auto Wake-up Bit (WUE) During Normal Operation ..... 172
Auto Wake-up Bit (WUE) During Sleep ..... 173
Automatic Baud Rate Calibration ..... 171
Baud Rate Generator with Clock Arbitration ..... 199
BRG Reset Due to SDA Arbitration ..... 209
Brown-out Reset (BOR) ..... 262
Brown-out Reset Situations ..... 218
Bus Collision
Start Condition Timing ..... 208
Bus Collision During a Repeated Start Condition (Case 1) ..... 210
Bus Collision During a Repeated Start Condition (Case2) ..... 210
Bus Collision During a Start Condition (SCL = 0) ..... 209
Bus Collision During a Stop Condition ..... 211
Bus Collision for Transmit and Acknowledge ..... 207
CLKOUT and I/O ..... 261
Clock Timing ..... 259



© 2006-2012 Microchip Technology Inc.

Comparator Output ..... 89
Enhanced Capture/Compare/PWM (ECCP) ..... 265
EUSART Synchronous Receive (Master/Slave) ..... 270
EUSART Synchronous Transmission (Master/Slave) ..... 270
Fail-Safe Clock Monitor (FSCM) ..... 76
Full-Bridge PWM Output ..... 141
Half-Bridge PWM Output ..... 139, 148
I2C Bus Data ..... 274
I2C Bus Start/Stop Bits ..... 273
I2C Master Mode First Start Bit Timing ..... 200
I2C Master Mode Reception Timing ..... 204
I2C Master Mode Transmission Timing ..... 203
I2C Module
Bus Collision
Transmit Timing ..... 207
INT Pin Interrupt ..... 226
Internal Oscillator Switch Timing ..... 72
Master Mode Transmit Clock Arbitration ..... 206
PWM Auto-shutdown
Auto-restart Enabled ..... 147
Firmware Restart ..... 147
PWM Direction Change ..... 142
PWM Direction Change at Near 100% Duty Cycle ..... 143
PWM Output (Active-High) ..... 137
PWM Output (Active-Low) ..... 138
Repeat Start Condition ..... 201
Reset, WDT, OST and Power-up Timer ..... 262
Send Break Character Sequence ..... 174
Slave Synchronization ..... 190
SPI Master Mode (CKE = 1, SMP = 1) ..... 271
SPI Mode Timing (Master Mode) SPI Mode Master Mode Timing Diagram ..... 189
SPI Mode Timing (Slave Mode with CKE = 0) ..... 191
SPI Mode Timing (Slave Mode with CKE = 1) ..... 191
SPI Slave Mode (CKE = 0) ..... 272
SPI Slave Mode (CKE = 1) ..... 272
Stop Condition Receive or Transmit ..... 206
Synchronous Reception (Master Mode, SREN) ..... 178
Synchronous Transmission ..... 176
Synchronous Transmission (Through TXEN) ..... 176
Time-out Sequence
Case 1 ..... 220
Case 2 ..... 220
Case 3 ..... 220
Timer0 and Timer1 External Clock ..... 264
Timer1 Incrementing Edge ..... 83
Two Speed Start-up ..... 74
Wake-up from Interrupt ..... 231
Timing Parameter Symbology ..... 258
Timing Requirements
I2C Bus Data ..... 275
I2C Bus Start/Stop Bits ..... 274
SPI Mode ..... 273
TRISA ..... 41
TRISA Register ..... 41
TRISB ..... 49
TRISB Register ..... 50
TRISC ..... 55
TRISC Register ..... 55
TRISD ..... 59
TRISD Register ..... 59
TRISE ..... 61
TRISE Register ..... 61
Two-Speed Clock Start-up Mode ..... 73
TXREG ..... 157
TXSTA Register ..... 164
BRGH Bit ..... 167

**U**

Ultra Low-Power Wake-up ..... 18, 20, 42, 43

**V**

Voltage Reference (VR)
Specifications ..... 266
Voltage Reference. *See* Comparator Voltage Reference (CVREF)
Voltage References
Associated Registers ..... 102
VP6 Stabilization ..... 99
VREF. *SEE* ADC Reference Voltage

**W**

Wake-up on Break ..... 172
Wake-up Using Interrupts ..... 230
Watchdog Timer (WDT) ..... 228
Associated Registers ..... 229
Clock Source ..... 228
Modes ..... 228
Period ..... 228
Specifications ..... 263
Waveform for Slave Mode General Call Address Sequence ..... 196
WCOL ..... 200, 202, 205
WCOL Status Flag ..... 200, 202, 205
WDTCON Register ..... 229
WPUB Register ..... 51
WWW Address ..... 333
WWW, On-Line Support ..... 13

**NOTES:**


© 2006-2012 Microchip Technology Inc.

## THE MICROCHIP WEB SITE

Microchip provides online support via our WWW site at www.microchip.com. This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

## CUSTOMER CHANGE NOTIFICATION SERVICE

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip web site at www.microchip.com. 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
- Development Systems Information Line

Customers should contact their distributor, representative or field application engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

**Technical support is available through the web site at: http://microchip.com/support**

## READER RESPONSE

It is our intention to provide you with the best documentation possible to ensure successful use of your Microchip product. If you wish to provide your comments on organization, clarity, subject matter, and ways in which our documentation can better serve you, please FAX your comments to the Technical Publications Manager at (480) 792-4150.

Please list the following information, and use this outline to provide us with your comments about this document.

TO: Technical Publications Manager  Total Pages Sent _____

RE: Reader Response

From: Name _____

Company _____

Address _____

City / State / ZIP / Country _____

Telephone: (_____) _____ - _____  FAX: (_____) _____ - _____

Application (optional):

Would you like a reply? ___ Y ___ N

Device: PIC16F882/883/884/886/887  Literature Number: DS41291G

Questions:

1. What are the best features of this document?

2. How does this document meet your hardware and software development needs?

3. Do you find the organization of this document easy to follow? If not, why?

4. What additions to the document do you think would enhance the structure and subject?

5. What deletions from the document could be made without affecting the overall usefulness?

6. Is there any incorrect or misleading information (what and where)?

7. How would you improve this document?


© 2006-2012 Microchip Technology Inc.

## 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:** | PIC16F883, PIC16F883T(1), PIC16F884, PIC16F884T(1), PIC16F886, PIC16F886T(1), PIC16F887, PIC16F887T(1), VDD range 2.0V to 5.5V | |
| **Tape and Reel Option:** | Blank<br>T | = Standard packaging (tube or tray)<br>= Tape and Reel(1) |
| **Temperature Range:** | I<br>E | = -40°C to +85°C (Industrial)<br>= -40°C to +125°C (Extended) |
| **Package:(2)** | ML<br>P<br>PT<br>SO<br>SP<br>SS | = Quad Flat No Leads (QFN)<br>= Plastic DIP<br>= Plastic Thin-Quad Flatpack (TQFP)<br>= Plastic Small Outline (SOIC) (7.50 mm)<br>= Skinny Plastic DIP<br>= Plastic Shrink Small Outline |
| **Pattern:** | QTP, SQTP, Code or Special Requirements (blank otherwise) | |

**Examples:**

a) PIC16F883-E/P 301 = Extended Temp., PDIP package, 20 MHz, QTP pattern #301

b) PIC16F883-I/SO = Industrial Temp., SOIC package, 20 MHz

**Note 1:** Tape and Reel identifier only appears in the catalog part number description. This identifier is used for ordering purposes and is not printed on the device package. Check with your Microchip Sales Office for package availability with the Tape and Reel option.

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

**NOTES:**


© 2006-2012 Microchip Technology Inc.

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

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

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

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE**.** Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights.

**Trademarks**

The Microchip name and logo, the Microchip logo, dsPIC, FlashFlex, KEELOQ, KEELOQ logo, MPLAB, PIC, PICmicro, PICSTART, PIC$^{32}$ logo, rfPIC, SST, SST Logo, SuperFlash and UNI/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

FilterLab, Hampshire, HI-TECH C, Linear Active Thermistor, MTP, SEEVAL and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Silicon Storage Technology is a registered trademark of Microchip Technology Inc. in other countries.

Analog-for-the-Digital Age, Application Maestro, BodyCom, chipKIT, chipKIT logo, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, dsSPEAK, ECAN, ECONOMONITOR, FanSense, HI-TIDE, In-Circuit Serial Programming, ICSP, Mindi, MiWi, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, mTouch, Omniscient Code Generation, PICC, PICC-18, PICDEM, PICDEM.net, PICkit, PICtail, REAL ICE, rfLAB, Select Mode, SQI, Serial Quad I/O, Total Endurance, TSHARC, UniWinDriver, WiperLock, ZENA and Z-Scale are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

GestIC and ULPP are registered trademarks of Microchip Technology Germany II GmbH & Co. & KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

© 2006-2012, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

Printed on recycled paper.

ISBN: 9781620766743

**QUALITY MANAGEMENT SYSTEM**
**CERTIFIED BY DNV**
**ISO/TS 16949**

*Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.*

# Worldwide Sales and Service

## AMERICAS

**Corporate Office**
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://www.microchip.com/support
Web Address:
www.microchip.com

**Atlanta**
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

**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**
Farmington Hills, MI
Tel: 248-538-2250
Fax: 248-538-2260

**Indianapolis**
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453

**Los Angeles**
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

**Santa Clara**
Santa Clara, CA
Tel: 408-961-6444
Fax: 408-961-6445

**Toronto**
Mississauga, Ontario,
Canada
Tel: 905-673-0699
Fax: 905-673-6509

## ASIA/PACIFIC

**Asia Pacific Office**
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon
Hong Kong
Tel: 852-2401-1200
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 - Hangzhou**
Tel: 86-571-2819-3187
Fax: 86-571-2819-3189

**China - Hong Kong SAR**
Tel: 852-2401-1200
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: 86-755-8203-2660
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

**China - Xiamen**
Tel: 86-592-2388138
Fax: 86-592-2388130

**China - Zhuhai**
Tel: 86-756-3210040
Fax: 86-756-3210049

## ASIA/PACIFIC

**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: 91-20-2566-1512
Fax: 91-20-2566-1513

**Japan - Osaka**
Tel: 81-66-152-7160
Fax: 81-66-152-9310

**Japan - Yokohama**
Tel: 81-45-471- 6166
Fax: 81-45-471-6122

**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
Fax: 886-7-330-9305

**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 - Munich**
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

**Italy - Milan**
Tel: 39-0331-742611
Fax: 39-0331-466781

**Netherlands - Drunen**
Tel: 31-416-690399
Fax: 31-416-690340

**Spain - Madrid**
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

**UK - Wokingham**
Tel: 44-118-921-5869
Fax: 44-118-921-5820

10/26/12


© 2006-2012 Microchip Technology Inc.