



PIC16(L)F18426/46

14/20-Pin Full-Featured, Low Pin Count Microcontrollers with XLP

Description

PIC16(L)F184XX microcontrollers feature Intelligent Analog, Core Independent Peripherals (CIPs) and communication peripherals combined with eXtreme Low-Power (XLP) for a wide range of general purpose and low-power applications. Features such as a 12-bit Analog-to-Digital Converter with Computation (ADC²), Memory Access Partitioning (MAP), the Device Information Area (DIA), Power-Saving operating modes, and Peripheral Pin Select (PPS), offer flexible solutions for a wide variety of custom applications.

Core Features

- C Compiler Optimized RISC Architecture
- Operating Speed:
 - DC – 32 MHz clock input
 - 125 ns minimum instruction cycle
- Interrupt Capability
- 16-Level Deep Hardware Stack
- Up to Four 8-Bit Timers
- Up to Four 16-Bit Timers
- Low-Current Power-on Reset (POR)
- Configurable Power-up Timer (PWRT)
- Brown-out Reset (BOR)
- Low-Power BOR (LPBOR) Option
- Windowed Watchdog Timer (WWDT):
 - Variable prescaler selection
 - Variable window size selection
 - All sources configurable in hardware or software
- Programmable Code Protection

Memory

- Up to 28 Kbytes Program Flash Memory
- Up to 2 KB Data SRAM Memory
- 256B Data EEPROM
- Direct, Indirect and Relative Addressing Modes
- Memory Access Partition (MAP):
 - Write-protect
 - Customizable partition
- Device Information Area (DIA)
- Device Characteristics Information (DCI)

Operating Characteristics

- Operating Voltage Range:
 - 1.8V to 3.6V (PIC16LF184XX)
 - 2.3V to 5.5V (PIC16F184XX)
- Temperature Range:
 - Industrial: -40°C to 85°C
 - Extended: -40°C to 125°C

eXtreme Low-Power (XLP) Features

- Doze: CPU and Peripherals Running at Different Cycle Rates (Typically CPU is Lower)
- Idle: CPU Halted While Peripherals Operate
- Sleep: Lowest Power Consumption
- Peripheral Module Disable (PMD):
 - Ability to selectively disable hardware module to minimize active power consumption of unused peripherals
- Extreme Low-Power Mode (XLP)
 - Sleep: 500 nA typical @ 1.8V
 - Sleep and Watchdog Timer: 900 nA typical @ 1.8V

Power-Saving Operation Modes

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

Digital Peripherals

- Configurable Logic Cell (CLC):
 - Four CLCs
 - Integrated combinational and sequential logic
- Complementary Waveform Generator (CWG):
 - Two CWGs
 - Rising and falling edge dead-band control
 - Full-bridge, half-bridge, 1-channel drive
 - Multiple signal sources
- Capture/Compare/PWM (CCP) Modules:
 - Four CCPs
 - 16-bit resolution for Capture/Compare modes
 - 10-bit resolution for PWM mode
- Pulse-Width Modulators (PWM):
 - Two 10-bit PWMs
- Numerically Controlled Oscillator (NCO):
 - Precision linear frequency generator (@50% duty cycle) with 0.0001% step size of source input clock
 - Input Clock: $0 \text{ Hz} < f_{\text{NCO}} < 32 \text{ MHz}$
 - Resolution: $f_{\text{NCO}}/2^{20}$

- Serial Communications:
 - EUSART
 - One EUSART(s)
 - RS-232, RS-485, LIN compatible
 - Auto-Baud Detect, Auto-wake-up on Start.
 - Master Synchronous Serial Port (MSSP)
 - Two MSSP(s)
 - SPI
 - I²C, SMBus and PMBus™ compatible
- Data Signal Modulator (DSM):
 - Modulates a carrier signal with digital data to create custom carrier synchronized output waveforms
- Up to 18 I/O Pins:
 - Individually programmable pull-ups
 - Slew rate control
 - Interrupt-on-change with edge-select
 - Input level selection control (ST or TTL)
 - Digital open-drain enable
- Peripheral Pin Select (PPS):
 - I/O pin remapping of digital peripherals
- Timer Modules:
 - Timer0:
 - 8/16-bit timer/counter
 - Synchronous or asynchronous operation
 - Programmable prescaler/postscaler
 - Time base for capture/compare function
 - Timer1/3/5 with gate control:
 - 16-bit timer/counter
 - Programmable internal or external clock sources
 - Multiple gate sources
 - Multiple gate modes
 - Time base for capture/compare function
 - Timer2/4/6 with Hardware Limit Timer:
 - 8-bit timers
 - Programmable prescaler/postscaler
 - Time base for PWM function
 - Hardware Limit (HLT) and one-shot extensions
 - Selectable clock sources
 - Signal Measurement Timer (SMT):
 - One SMT(s)
 - 24-bit timer/counter with programmable prescaler

Analog Peripherals

- 12-bit Analog-to-Digital Converter with Computation (ADC²):
 - up to 140 ksps
 - up to 17 external channels
 - Conversion available during Sleep
 - Automated post-processing
 - Automated math functions on input signals:

- Averaging, filter calculations, oversampling and threshold comparison
 - Integrated charge pump for low-voltage operation
 - CVD support
- Zero-Cross Detect (ZCD):
 - AC high voltage zero-crossing detection
 - Synchronized switching control and timing
- Temperature Sensor Circuit
- Comparator:
 - Two Comparators
 - Fixed Voltage Reference at (non)inverting input(s)
 - Comparator outputs externally accessible
- Digital-to-Analog Converter (DAC):
 - 5-bit resolution, rail-to-rail
 - Positive Reference Selection
 - Unbuffered I/O pin output
 - Internal connections to ADC² and comparators
- Fixed Voltage Reference with 1.024V, 2.048V and 4.096V Output Levels

Flexible Oscillator Structure

- High-Precision Internal Oscillator:
 - Software-selectable frequency range up to 32 MHz
 - $\pm 2\%$ at calibration (nominal)
- 4x PLL for use with External Sources:
 - up to 32 MHz (4-8 MHz input)
- 2x PLL for use with the HFINTOSC:
 - up to 32 MHz
- Low-Power Internal 31 kHz Oscillator (LFINTOSC)
- External 32.768 kHz Crystal Oscillator (SOCS)
- External Oscillator Block with:
 - Three crystal/resonator modes up to 20 MHz
 - Three external clock modes up to 32 MHz
- Fail-Safe Clock Monitor:
 - Detects clock source failure
- Oscillator Start-up Timer (OST):
 - Ensures stability of crystal oscillator sources

PIC16(L)F184XX Family Types

Table 1. Devices Included In This Data Sheet

| Device | Program Flash Memory (KW) | Program Flash Memory (Kbytes) | EEPROM (B) | RAM (B) | I/O's ⁽¹⁾ | 12-bit ADC ² (ch) | 5-bit DAC | Comparators | CWG | Clock Ref | Timers (8/16-bit) | CCP | PWM | NCO | EUSART | MSSP (I ² C/SPI) | CLC | DSM | PPS | XLP | PMD | WWDT | MAP | DIA | ICD ⁽²⁾ |
|----------------|---------------------------|-------------------------------|------------|---------|----------------------|------------------------------|-----------|-------------|-----|-----------|-------------------|-----|-----|-----|--------|-----------------------------|-----|-----|-----|-----|-----|------|-----|-----|--------------------|
| PIC16(L)F18426 | 16 | 28 | 256 | 2048 | 12 | 11 | 1 | 2 | 2 | 1 | 4/4 | 4 | 2 | 1 | 1 | 2 | 4 | 1 | Y | Y | Y | Y | Y | Y | I |
| PIC16(L)F18446 | 16 | 28 | 256 | 2048 | 18 | 17 | 1 | 2 | 2 | 1 | 4/4 | 4 | 2 | 1 | 1 | 2 | 4 | 1 | Y | Y | Y | Y | Y | Y | I |

Note:

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

Table 2. Devices Not Included In This Data Sheet

| Device | Program Flash Memory (Words) | Program Flash Memory (Kbytes) | EEPROM (B) | RAM (B) | I/O's ⁽¹⁾ | 12-bit ADC ² (ch) | 5-bit DAC | Comparators | CWG | Clock Ref | Timers (8/16-bit) | CCP | PWM | NCO | EUSART | MSSP (I ² C/SPI) | CLC | DSM | PPS | XLP | PMD | WWDT | MAP | DIA | ICD ⁽²⁾ |
|----------------|------------------------------|-------------------------------|------------|---------|----------------------|------------------------------|-----------|-------------|-----|-----------|-------------------|-----|-----|-----|--------|-----------------------------|-----|-----|-----|-----|-----|------|-----|-----|--------------------|
| PIC16(L)F18424 | 4 | 7 | 256 | 512 | 12 | 11 | 1 | 2 | 2 | 1 | 4/4 | 4 | 2 | 1 | 1 | 1 | 4 | 1 | Y | Y | Y | Y | Y | Y | I |
| PIC16(L)F18425 | 8 | 14 | 256 | 1024 | 12 | 11 | 1 | 2 | 2 | 1 | 4/4 | 4 | 2 | 1 | 1 | 2 | 4 | 1 | Y | Y | Y | Y | Y | Y | I |
| PIC16(L)F18444 | 4 | 7 | 256 | 512 | 18 | 17 | 1 | 2 | 2 | 1 | 4/4 | 4 | 2 | 1 | 1 | 1 | 4 | 1 | Y | Y | Y | Y | Y | Y | I |
| PIC16(L)F18445 | 8 | 14 | 256 | 1024 | 18 | 17 | 1 | 2 | 2 | 1 | 4/4 | 4 | 2 | 1 | 1 | 2 | 4 | 1 | Y | Y | Y | Y | Y | Y | I |
| PIC16(L)F18455 | 8 | 14 | 256 | 1024 | 26 | 24 | 1 | 2 | 3 | 1 | 4/4 | 5 | 2 | 1 | 2 | 2 | 4 | 1 | Y | Y | Y | Y | Y | Y | I |
| PIC16(L)F18456 | 16 | 28 | 256 | 2048 | 26 | 24 | 1 | 2 | 3 | 1 | 4/4 | 5 | 2 | 1 | 2 | 2 | 4 | 1 | Y | Y | Y | Y | Y | Y | I |

Data Sheet Index:

1. [DS40002000A, PIC16\(L\)F18424/44 Data Sheet, 14/20-Pin Full-Featured, Low Pin Count Microcontrollers with XLP](#)
2. [DS40002002A, PIC16\(L\)F18425/45 Data Sheet, 14/20-Pin Full-Featured, Low Pin Count Microcontrollers with XLP](#)
3. [DS40002038B, PIC16\(L\)F18455/56 Data Sheet, 28-Pin Full-Featured, Low Pin Count Microcontrollers with XLP](#)

Packages

| Packages | PDIP | SOIC | SSOP | TSSOP | UQFN (4x4) |
|----------------|------|------|------|-------|------------|
| PIC16(L)F18426 | • | • | | • | • |
| PIC16(L)F18446 | • | • | • | | • |

Note: Pin details are subject to change.



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

Pin Diagrams

14/16-Pin Diagrams

Figure 1. 14-Pin PDIP, SOIC, TSSOP

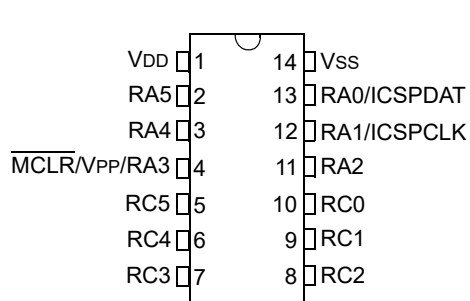
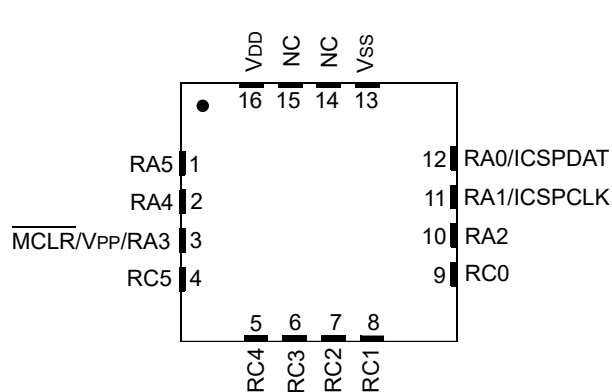


Figure 2. 16-Pin UQFN (4x4)



Note: It is recommended that the exposed bottom pad be connected to V_{SS} .

See [Table 3](#) for more information.

20-Pin Diagrams

Figure 3. 20-Pin PDIP, SOIC, SSOP

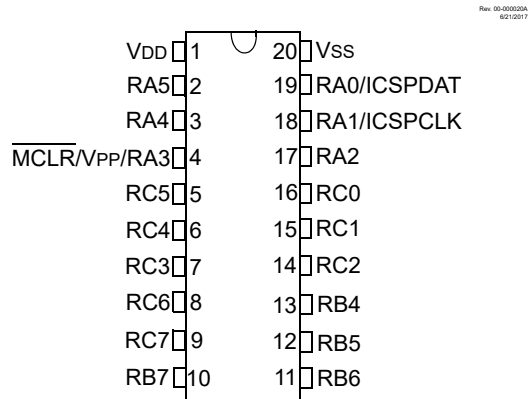
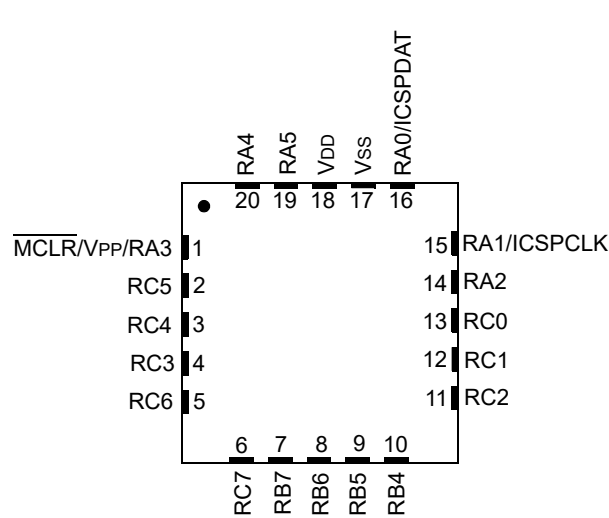


Figure 4. 20-Pin UQFN (4x4)



Note: It is recommended that the exposed bottom pad be connected to V_{SS}.
See [Table 4](#) for more information.

Pin Allocation Tables

Table 3. 14/16-Pin Allocation Table

| I/O | 14-pin PDIP/SOIC/TSSOP | 16-pin UQFN | ADC | Reference | Comparator | NCO | DAC | DSM | Timers | CCP | PWM | CWG | MSSP | ZCD | EUSART | CLC | CLKR | Interrupts | Pull-up | Basic | |
|--------------------|------------------------|-------------|------------------------------|-----------|------------------|---------|-----------|-----------------------|---|-----------------------|---------|--|--|------|--|-----------------------|------|------------|---------|-------------------------|-----|
| RA0 | 13 | 12 | ANA0 | — | C1IN0+ | — | DAC1OUT1 | MDSRC ⁽¹⁾ | — | — | — | — | SSZ ⁽¹⁾ | — | — | — | — | IOCA0 | Y | ICDDAT ICSPDAT | |
| RA1 | 12 | 11 | ANA1 | ADCVREF+ | C1IN0 C2IN0- | — | DAC1VREF+ | — | — | — | — | — | — | — | — | — | — | IOCA1 | Y | ICDCLK ICSPCLK | |
| RA2 | 11 | 10 | ANA2 | ADCVREF- | — | — | DAC1VREF- | — | T0CKI ⁽¹⁾ | CCP3IN ⁽¹⁾ | — | CWG1IN ⁽¹⁾ CWG2IN ⁽¹⁾ | — | ZCD1 | — | — | — | IOCA2 | Y | INT ⁽¹⁾ | |
| RA3 | 4 | 3 | — | — | — | — | — | — | T6IN ⁽¹⁾ | — | — | — | — | — | — | — | — | IOCA3 | Y | MCLR Vpp | |
| RA4 | 3 | 2 | ANA4 | — | — | — | — | — | T1G ⁽¹⁾ SMT1WIN ⁽¹⁾ | — | — | — | — | — | — | — | — | IOCA4 | Y | CLKOUT SOSCO OSC2 | |
| RA5 | 2 | 1 | ANA5 | — | — | — | — | — | T1CKI ⁽¹⁾ T2IN ⁽¹⁾ SMT1SIG ⁽¹⁾ | — | — | — | — | — | — | CLCIN3 ⁽¹⁾ | — | IOCA5 | Y | CLKIN SOSCI OSC1 | |
| RC0 | 10 | 9 | ANC0 | — | C2IN0+ | — | — | — | T5CKI ⁽¹⁾ | — | — | — | SCK1 ⁽¹⁾ SCL1 ^(1,3,4) | — | — | — | — | IOCC0 | Y | — | |
| RC1 | 9 | 8 | ANC1 | — | C1IN1- C2IN1- | — | — | — | T4IN ⁽¹⁾ | CCP4IN ⁽¹⁾ | — | — | SDI1 ⁽¹⁾ SDA1 ^(1,3,4) | — | — | CLCIN2 ⁽¹⁾ | — | IOCC1 | Y | — | |
| RC2 | 8 | 7 | ANC2 ADACT ⁽¹⁾ | — | C1IN2- C2IN2- | — | — | MDCARL ⁽¹⁾ | — | — | — | — | — | — | — | — | — | IOCC2 | Y | — | |
| RC3 | 7 | 6 | ANC3 | — | C1IN3- C2IN3- | — | — | — | T5G ⁽¹⁾ | CCP2IN ⁽¹⁾ | — | — | SS1 ⁽¹⁾ | — | — | CLCIN0 ⁽¹⁾ | — | IOCC3 | Y | — | |
| RC4 | 6 | 5 | ANC4 | — | — | — | — | — | T3G ⁽¹⁾ | — | — | — | SCK2 ^(1,5) SCL2 ^(1,3,4,5) | — | CK1 ^(1,3) | CLCIN1 ⁽¹⁾ | — | IOCC4 | Y | — | |
| RC5 | 5 | 4 | ANC5 | — | — | — | — | MDCARH ⁽¹⁾ | T3CKI ⁽¹⁾ | CCP1IN ⁽¹⁾ | — | — | SDI2 ^(1,5) SDA2 ^(1,3,4,5) | — | RX1 ⁽¹⁾ DT1 ^(1,3) | — | — | IOCC5 | Y | — | |
| VDD | 1 | 16 | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | VDD | |
| VSS | 14 | 13 | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | VSS |
| OUT ⁽²⁾ | — | — | ADCGRDA | — | C1OUT | NCO1OUT | — | DSM1OUT | TMR0OUT | CCP1OUT | PWM6OUT | CWG1A CWG2A | SDO1 SDO2 | — | DT1 ⁽³⁾ | CLC1OUT | CLKR | — | — | — | |
| | — | — | ADCGRDB | — | C2OUT | — | — | — | — | CCP2OUT | PWM7OUT | CWG1B CWG2B | SCK1 SCK2 | — | CK1 ⁽³⁾ | CLC2OUT | — | — | — | — | |
| | — | — | — | — | — | — | — | — | — | CCP3OUT | — | CWG1C CWG2C | SCL1 ⁽³⁾ SCL2 ⁽³⁾ | — | TX1 | CLC3OUT | — | — | — | — | |
| | — | — | — | — | — | — | — | — | — | CCP4OUT | — | CWG1D CWG2D | SDA1 ⁽³⁾ SDA2 ⁽³⁾ | — | — | CLC4OUT | — | — | — | — | |

Table 4. 20-Pin Allocation Table

| I/O | 20-pin PDIP/SOIC/TSSOP | 20-pin UQFN | ADC | Reference | Comparator | NCO | DAC | DSM | Timers | CCP | PWM | CWG | MSSP | ZCD | EUSART | CLC | CLKR | Interrupts | Pull-up | Basic |
|-----|------------------------|-------------|------|-----------|------------------|-----|-----------|----------------------|----------------------|-----|-----|--|--------------------|------|--------|-----------------------|------|------------|---------|--------------------|
| RA0 | 19 | 16 | ANA0 | — | C1IN0+ | — | DAC1OUT1 | — | — | — | — | — | — | — | — | — | — | IOCA0 | Y | ICDDAT/ ICSPDAT |
| RA1 | 18 | 15 | ANA1 | ADCVREF+ | C1IN0- C2IN0- | — | DAC1VREF+ | MDSRC ⁽¹⁾ | — | — | — | — | SSZ ⁽¹⁾ | — | — | — | — | IOCA1 | Y | ICDCLK/ ICSPCLK |
| RA2 | 17 | 14 | ANA2 | ADCVREF- | — | — | DAC1VREF- | — | T0CKI ⁽¹⁾ | — | — | CWG1IN ⁽¹⁾ CWG2IN ⁽¹⁾ | — | ZCD1 | — | CLCIN0 ⁽¹⁾ | — | IOCA2 | Y | INT ⁽¹⁾ |
| RA3 | 4 | 1 | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | IOCA3 | Y | MCLR Vpp |

.....continued

| IO | 20-pin PDIP/SOIC/TSSOP | 20-pin UDFN | ADC | Reference | Comparator | NCO | DAC | DSM | Timers | CCP | PWM | CWG | MSSP | ZCD | EUSART | CLC | CLKR | Interrupts | Pull-up | Basic |
|--------|------------------------|-------------|------------------|-----------|------------------|---------|-----|-----------|----------------------------------|-----------|---------|----------------|----------------------------|-----|--------------------|-----------|------|------------|---------|-------------------------|
| RA4 | 3 | 20 | ANA4 | — | — | — | — | — | T1G(1) SMT1WIN(1) | CCP4IN(1) | — | — | — | — | — | — | — | IOCA4 | Y | CLKOUT SOSCO OSC2 |
| RA5 | 2 | 19 | ANA5 | — | — | — | — | — | T1CK(1) T2IN(1) SMT1SIG(1) | — | — | — | — | — | — | — | — | IOCA5 | Y | CLKIN SOSCI OSC1 |
| RB4 | 13 | 10 | ANB4 | — | — | — | — | — | T5G(1) | — | — | — | SDI1(1) SDA1(1,3,4) | — | — | CLCIN2(1) | — | IOCB4 | Y | — |
| RB5 | 12 | 9 | ANB5 | — | — | — | — | — | — | CCP3IN(1) | — | — | SDI2(1,5) SDA2(1,3,4,5) | — | RX1(1) DT1(1,3) | CLCIN3(1) | — | IOCB5 | Y | — |
| RB6 | 11 | 8 | ANB6 | — | — | — | — | — | — | — | — | — | SCK1(1) SCL1(1,3,4) | — | — | — | — | IOCB6 | Y | — |
| RB7 | 10 | 7 | ANB7 | — | — | — | — | — | T6IN(1) | — | — | — | SCK2(1,5) SCL2(1,3,4,5) | — | CK1(1,3) | — | — | IOCB7 | Y | — |
| RC0 | 16 | 13 | ANC0 | — | C2IN0+ | — | — | — | T3CK(1) T3G(1) | — | — | — | — | — | — | — | — | IOCC0 | Y | — |
| RC1 | 15 | 12 | ANC1 | — | C1IN1- C2IN1- | — | — | — | — | — | — | — | — | — | — | — | — | IOCC1 | Y | — |
| RC2 | 14 | 11 | ANC2 ADACT(1) | — | C1IN2- C2IN2- | — | — | MDCARL(1) | T5CK(1) | — | — | — | — | — | — | — | — | IOCC2 | Y | — |
| RC3 | 7 | 4 | ANC3 | — | C1IN3- C2IN3- | — | — | — | — | CCP2IN(1) | — | — | — | — | — | CLCIN1(1) | — | IOCC3 | Y | — |
| RC4 | 6 | 3 | ANC4 | — | — | — | — | — | — | — | — | — | — | — | — | — | — | IOCC4 | Y | — |
| RC5 | 5 | 2 | ANC5 | — | — | — | — | MDCARH(1) | T4IN(1) | CCP1IN(1) | — | — | — | — | — | — | — | IOCC5 | Y | — |
| RC6 | 8 | 5 | ANC6 | — | — | — | — | — | — | — | — | — | SS1(1) | — | — | — | — | IOCC6 | Y | — |
| RC7 | 9 | 6 | ANC7 | — | — | — | — | — | — | — | — | — | — | — | — | — | — | IOCC7 | Y | — |
| VDD | 1 | 18 | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | VDD |
| VSS | 20 | 17 | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | VSS |
| OUT(2) | — | — | ADCGRDA | — | C1OUT | NCO1OUT | — | DSM1OUT | TMR0OUT | CCP1OUT | PWM6OUT | CWG1A CWG2A | SDO1 SDO2 | — | DT1(3) | CLC1OUT | CLKR | — | — | — |
| | — | — | ADCGRDB | — | C2OUT | — | — | — | — | CCP2OUT | PWM7OUT | CWG1B CWG2B | SCK1 SCK2 | — | CK1(3) | CLC2OUT | — | — | — | — |
| | — | — | — | — | — | — | — | — | — | CCP3OUT | — | CWG1C CWG2C | SCL1(3) SCL2(3) | — | TX1 | CLC3OUT | — | — | — | — |
| | — | — | — | — | — | — | — | — | — | CCP4OUT | — | CWG1D CWG2D | SDA1(3) SDA2(3) | — | — | CLC4OUT | — | — | — | — |

Note:

1. Default peripheral input. Input can be moved to any other pin with the PPS input selections registers.
2. All pin outputs default to PORT latch data. Any pin can be selected as a digital peripherals output with the PPS output selection registers.
3. These peripheral functions are bidirectional. The output pin selections must be the same as the input pin selections.
4. These pins are configured for I²C logic levels; clock and data signals may be assigned to any of these pins. Assignments to the other pins (e.g., RA5) will operate, but logic levels will be standard TTL/ST as selected by the INLVL register.

Table of Contents

| | |
|--|-----|
| Description..... | 1 |
| Core Features..... | 1 |
| Memory..... | 1 |
| Operating Characteristics..... | 2 |
| eXtreme Low-Power (XLP) Features..... | 2 |
| Power-Saving Operation Modes..... | 2 |
| Digital Peripherals..... | 2 |
| Analog Peripherals..... | 3 |
| Flexible Oscillator Structure..... | 4 |
| PIC16(L)F184XX Family Types..... | 5 |
| Packages..... | 6 |
| Pin Diagrams..... | 6 |
| Pin Allocation Tables..... | 8 |
| 1. Device Overview..... | 13 |
| 2. Guidelines for Getting Started with PIC16(L)F18426/46 Microcontrollers..... | 19 |
| 3. Enhanced Mid-Range CPU..... | 24 |
| 4. Device Configuration..... | 26 |
| 5. Memory Organization..... | 39 |
| 6. NVM - Nonvolatile Memory Control | 76 |
| 7. Interrupts..... | 98 |
| 8. OSC - Oscillator Module..... | 122 |
| 9. REFCLK - Reference Clock Output Module..... | 143 |
| 10. Resets..... | 148 |
| 11. WWDT - Windowed Watchdog Timer | 161 |
| 12. Power-Saving Operation Modes..... | 172 |
| 13. PMD - Peripheral Module Disable..... | 181 |
| 14. I/O Ports..... | 191 |
| 15. IOC - Interrupt-On-Change..... | 221 |
| 16. PPS - Peripheral Pin Select Module | 233 |

| | |
|--|-----|
| 17. CLC - Configurable Logic Cell..... | 244 |
| 18. TMR0 - Timer0 Module..... | 267 |
| 19. TMR1 - Timer1 Module with Gate Control..... | 275 |
| 20. TMR2 - Timer2 Module..... | 293 |
| 21. SMT - Signal Measurement Timer..... | 315 |
| 22. Capture/Compare/PWM Module..... | 338 |
| 23. CCP/PWM Timer Resource Selection..... | 351 |
| 24. PWM - Pulse-Width Modulation..... | 355 |
| 25. CWG - Complementary Waveform Generator..... | 363 |
| 26. NCO - Numerically Controlled Oscillator..... | 390 |
| 27. DSM - Data Signal Modulator Module..... | 400 |
| 28. EUSART - Enhanced Universal Synchronous Asynchronous Receiver Transmitter..... | 411 |
| 29. MSSP - Master Synchronous Serial Port Module..... | 441 |
| 30. FVR - Fixed Voltage Reference..... | 500 |
| 31. Temperature Indicator Module..... | 504 |
| 32. ADC ² - Analog-to-Digital Converter..... | 507 |
| 33. DAC - 5-Bit Digital-to-Analog Converter..... | 552 |
| 34. CMP - Comparator Module..... | 558 |
| 35. ZCD - Zero-Cross Detection Module..... | 570 |
| 36. Register Summary..... | 577 |
| 37. Instruction Set Summary..... | 602 |
| 38. ICSP [™] - In-Circuit Serial Programming [™] | 622 |
| 39. Development Support..... | 625 |
| 40. Electrical Specifications..... | 629 |
| 41. DC and AC Characteristics Graphs and Tables..... | 661 |
| 42. Packaging Information..... | 682 |
| 43. Revision History..... | 704 |
| The Microchip Website..... | 705 |
| Product Change Notification Service..... | 705 |
| Customer Support..... | 705 |

| | |
|--|-----|
| Product Identification System..... | 706 |
| Microchip Devices Code Protection Feature..... | 706 |
| Legal Notice..... | 707 |
| Trademarks..... | 707 |
| Quality Management System..... | 707 |
| Worldwide Sales and Service..... | 708 |

1. Device Overview

This document contains device-specific information for the following devices:

| | |
|---|--|
| <ul style="list-style-type: none">• PIC16F18426 | <ul style="list-style-type: none">• PIC16LF18426 |
| <ul style="list-style-type: none">• PIC16F18446 | <ul style="list-style-type: none">• PIC16LF18446 |

1.1 New Core Features

1.1.1 XLP Technology

All of the devices in the PIC16(L)F184XX family incorporate a range of features that can significantly reduce power consumption during operation. Key items include:

- **Alternate Run Modes:** By clocking the controller from the secondary oscillator or the internal oscillator block, power consumption during code execution can be reduced by as much as 90%.
- **Multiple Idle Modes:** The controller can also run with its CPU core disabled but the peripherals still active. In these states, power consumption can be reduced even further, to as little as 4% of normal operation requirements.
- **On-the-Fly Mode Switching:** The power-managed modes are invoked by user code during operation, allowing the user to incorporate power-saving ideas into their application's software design.
- **Peripheral Module Disable:** Modules that are not being used in the code can be selectively disabled using the PMD module. This further reduces the power consumption.

1.1.2 Multiple Oscillator Options and Features

All of the devices in the PIC16(L)F184XX family offer several different oscillator options. The PIC16(L)F184XX family can be clocked from several different sources:

- HFINTOSC
 - 1-32 MHz precision digitally controlled internal oscillator
- LFINTOSC
 - 31 kHz internal oscillator
- EXTOSC
 - External clock (EC)
 - Low-power oscillator (LP)
 - Medium-power oscillator (XT)
 - High-power oscillator (HS)
- SOSC
 - Secondary oscillator circuit optimized for 32 kHz clock crystals
- A Phase Lock Loop (PLL) frequency multiplier (2x/4x) is available to the External Oscillator modes enabling clock speeds of up to 32 MHz
- Fail-Safe Clock Monitor: This option constantly monitors the main clock source against a reference signal provided by the LFINTOSC. If a clock failure occurs, the controller is switched to the internal oscillator block, allowing for continued operation or a safe application shutdown.

1.2 Other Special Features

- 12-bit A/D Converter with Computation: This module incorporates programmable acquisition time, allowing for a channel to be selected and a conversion to be initiated without waiting for a sampling period and thus, reduce code overhead. It has a new module called ADC² with computation features, which provides a digital filter and threshold interrupt functions.

- **Memory Endurance:** The Flash cells for both program memory and data EEPROM are rated to last for many thousands of erase/write cycles – up to 10K for program memory and 100K for EEPROM. Data retention without refresh is conservatively estimated to be greater than 40 years.
- **Self-programmability:** These devices can write to their own program memory spaces under internal software control. By using a boot loader routine located in the protected Boot Block at the top of program memory, it becomes possible to create an application that can update itself in the field.
- **Enhanced Peripheral Pin Select:** The Peripheral Pin Select (PPS) module connects peripheral inputs and outputs to the device I/O pins. Only digital signals are included in the selections. All analog inputs and outputs remain fixed to their assigned pins.
- **Windowed Watchdog Timer (WWDT):**
 - Timer monitoring of overflow and underflow events
 - Variable prescaler selection
 - Variable window size selection
 - All sources configurable in hardware or software

1.3 Details on Individual Family Members

The devices of the PIC16(L)F184XX family described in the current datasheet are available in 14/20-pin packages. The block diagram for this device is shown in [Figure 1-1](#).

The devices have the following differences:

1. Program Flash Memory
2. Data Memory SRAM
3. Data Memory EEPROM
4. A/D channels
5. I/O ports
6. Enhanced USART
7. Input Voltage Range/Power Consumption

All other features for devices in this family are identical. These are summarized in the following Device Features table.

The pinouts for all devices are listed in the pin summary tables.

Table 1-1. Device Features

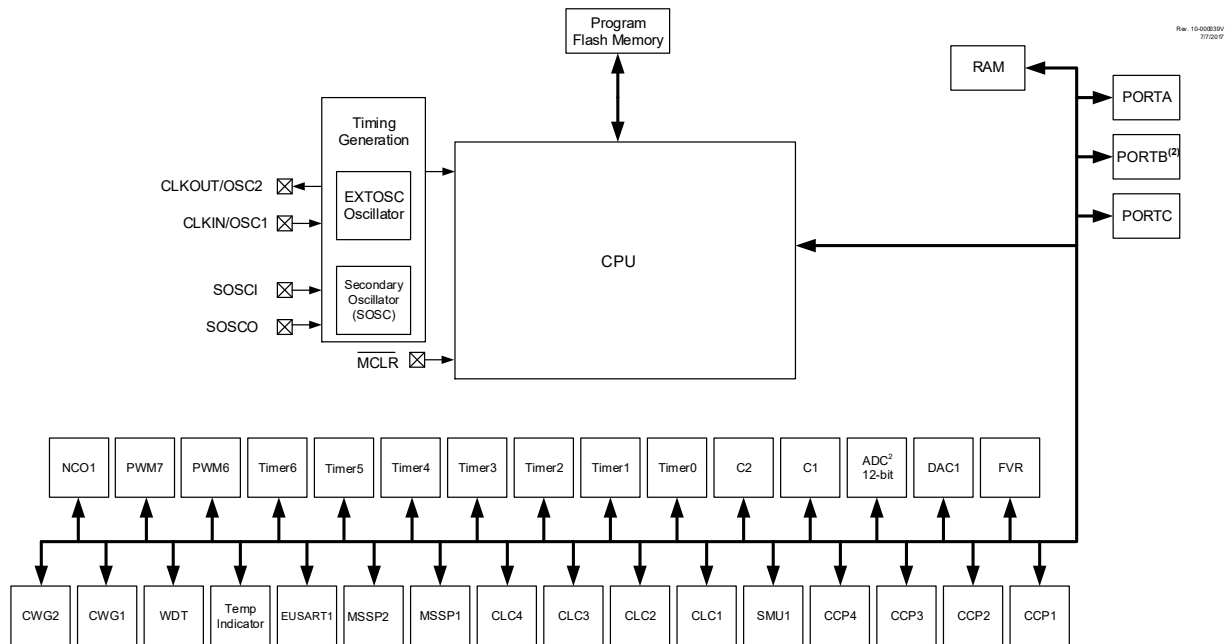
| Features | PIC16(L)F18426 | PIC16(L)F18446 |
|------------------------------------|--|---|
| Program Memory (KBytes) | 28 | 28 |
| Program Memory (Instructions) | 16384 | 16384 |
| Data Memory (Bytes) | 2048 | 2048 |
| Data EEPROM Memory (Bytes) | 256 | 256 |
| Packages | 14 - PDIP 14 - SOIC (3.9 mm) 14 - TSSOP 16 - uQFN (4x4) | 20 - PDIP 20 - SOIC (7.5 mm) 20 - SSOP 20 - uQFN (4x4) |
| I/O Ports | A, C | A, B, C |
| Capture/Compare/PWM Modules (CCP) | 4 | 4 |
| Configurable Logic Cell (CLC) | 4 | 4 |
| 10-Bit Pulse-Width Modulator (PWM) | 2 | 2 |

PIC16(L)F18426/46

Device Overview

|continued | | |
|--|--|--|
| Features | PIC16(L)F18426 | PIC16(L)F18446 |
| 12-Bit Analog-to-Digital Module (ADC ²) with Computation Accelerator | 11 channels | 17 channels |
| 5-Bit Digital-to-Analog Module (DAC) | 1 | 1 |
| Comparators | 2 | 2 |
| Numerical Contolled Oscillator (NCO) | 1 | 1 |
| Interrupt Sources | 40 | 40 |
| Timers (16-/8-bit) | 4 | 4 |
| Serial Communications | 2 MSSP 1 EUSART | 2 MSSP 1 EUSART |
| Complementary Waveform Generator (CWG) | 2 | 2 |
| Zero-Cross Detect (ZCD) | 1 | 1 |
| Data Signal Modulator (DSM) | 1 | 1 |
| Reference Clock Output Module | 1 | 1 |
| Peripheral Pin Select (PPS) | YES | YES |
| Peripheral Module Disable (PMD) | YES | YES |
| Programmable Brown-out Reset (BOR) | YES | YES |
| Resets (and Delays) | POR, BOR, <small>RESET</small> Instruction, Stack Overflow, Stack Underflow (PWRT, OST), MCLR, WDT | POR, BOR, <small>RESET</small> Instruction, Stack Overflow, Stack Underflow (PWRT, OST), MCLR, WDT |
| Instruction Set | 50 instructions 16-levels hardware stack | 50 instructions 16-levels hardware stack |
| Operating Frequency | DC – 32 MHz | DC – 32 MHz |

Figure 1-1. PIC16(L)F18426/46 Device Block Diagram



Note:

1. See applicable chapters for more information on peripherals.
2. PORTB available only on 20-pin or higher pin-count devices.

1.4 Register and Bit Naming Conventions

1.4.1 Register Names

When there are multiple instances of the same peripheral in a device, the Peripheral Control registers will be depicted as the concatenation of a peripheral identifier, peripheral instance, and control identifier. The control registers section will show just one instance of all the register names with an 'x' in the place of the peripheral instance number. This naming convention may also be applied to peripherals when there is only one instance of that peripheral in the device to maintain compatibility with other devices in the family that contain more than one.

1.4.2 Bit Names

There are two variants for bit names:

- Short name: Bit function abbreviation
- Long name: Peripheral abbreviation + short name

1.4.2.1 Short Bit Names

Short bit names are an abbreviation for the bit function. For example, some peripherals are enabled with the EN bit. The bit names shown in the registers are the short name variant.

Short bit names are useful when accessing bits in C programs. The general format for accessing bits by the short name is RegisterNamebits.ShortName. For example, the enable bit, EN, in the CM1CON0 register can be set in C programs with the instruction `CM1CON0bits.EN = 1`.

Short names are generally not useful in assembly programs because the same name may be used by different peripherals in different bit positions. When this occurs, during the include file generation, all instances of that short bit name are appended with an underscore plus the name of the register in which the bit resides to avoid naming contentions.

1.4.2.2 Long Bit Names

Long bit names are constructed by adding a peripheral abbreviation prefix to the short name. The prefix is unique to the peripheral, thereby making every long bit name unique. The long bit name for the COG1 enable bit is the COG1 prefix, G1, appended with the enable bit short name, EN, resulting in the unique bit name G1EN.



Important: The COG1 peripheral is used as an example. Not all devices have the COG peripheral.

Long bit names are useful in both C and assembly programs. For example, in C the COG1CON0 enable bit can be set with the `G1EN = 1` instruction. In assembly, this bit can be set with the `BSF COG1CON0, G1EN` instruction.

1.4.2.3 Bit Fields

Bit fields are two or more adjacent bits in the same register. Bit fields adhere only to the short bit naming convention. For example, the three Least Significant bits of the COG1CON0 register contain the Mode Control bits. The short name for this field is MD. There is no long bit name variant. Bit field access is only possible in C programs. The following example demonstrates a C program instruction for setting the COG1 to the Push-Pull mode:

```
COG1CON0bits.MD = 0x5;
```

Individual bits in a bit field can also be accessed with long and short bit names. Each bit is the field name appended with the number of the bit position within the field. For example, the Most Significant mode bit has the short bit name MD2 and the long bit name is G1MD2. The following two examples demonstrate assembly program sequences for setting the COG1 to Push-Pull mode:

Example 1:

```
MOVLW  ~ (1<<G1MD1)
ANDWF  COG1CON0, F
MOVLW  1<<G1MD2 | 1<<G1MD0
IORWF  COG1CON0, F
```

Example 2:

```
BSF    COG1CON0, G1MD2
BCF    COG1CON0, G1MD1
BSF    COG1CON0, G1MD0
```

1.4.3 Register and Bit Naming Exceptions

1.4.3.1 Status, Interrupt, and Mirror Bits

Status, interrupt enables, Interrupt flags, and Mirror bits are contained in registers that span more than one peripheral. In these cases, the bit name shown is unique so there is no prefix or short name variant.

1.4.3.2 Legacy Peripherals

There are some peripherals that do not strictly adhere to these naming conventions. Peripherals that have existed for many years and are present in almost every device are the exceptions. These exceptions were necessary to limit the adverse impact of the new conventions on legacy code. Peripherals that do adhere to the new convention will include a table in the registers section indicating the long name prefix for each peripheral instance. Peripherals that fall into the exception category will not have this table. These peripherals include, but are not limited to the following:

- EUSART
- MSSP

1.5 Register Legend

The table below describes the conventions for bit types and bit Reset values used in the current data sheet.

Table 1-2. Register Legend

| Value | Description |
|-------|--|
| RO | Read-only bit |
| W | Writable bit |
| U | Unimplemented bit, read as '0' |
| P | Programmable bit |
| '1' | Bit is set |
| '0' | Bit is cleared |
| x | Bit is unknown |
| u | Bit is unchanged |
| -n/n | Value at POR and BOR/Value at all other Resets |
| q | Reset Value is determined by hardware |
| f | Reset Value is determined by fuse setting |
| g | Reset Value at POR for PPS re-mappable signals |

2. Guidelines for Getting Started with PIC16(L)F18426/46 Microcontrollers

2.1 Basic Connection Requirements

Getting started with the PIC16(L)F18426/46 family of 8-bit microcontrollers requires attention to a minimal set of device pin connections before proceeding with development.

The following pins must always be connected:

- All V_{DD} and V_{SS} pins (see [2.2 Power Supply Pins](#))
- \overline{MCLR} pin (see [2.3 Master Clear \(MCLR\) Pin](#))

These pins must also be connected if they are being used in the end application:

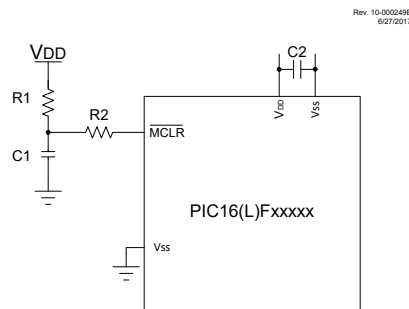
- PGC/PGD pins used for In-Circuit Serial Programming™ (ICSP™) and debugging purposes (see [2.4 In-Circuit Serial Programming \(ICSP\) Pins](#))
- OSCI and OSCO pins when an external oscillator source is used (see [2.5 External Oscillator Pins](#))

Additionally, the following may be required:

- V_{REF+}/V_{REF-} pins are used when external voltage reference for analog modules is implemented

The minimum mandatory connections are shown in the figure below.

Figure 2-1. Recommended Minimum Connections



Key (all values are recommendations):

- C1: 10 nF, 16V ceramic
- C2: 0.1 μ F, 16V ceramic
- R1: 10 k Ω
- R2: 100 Ω to 470 Ω

2.2 Power Supply Pins

2.2.1 Decoupling Capacitors

The use of decoupling capacitors on every pair of power supply pins (V_{DD} and V_{SS}) is required.

Consider the following criteria when using decoupling capacitors:

- Value and type of capacitor: A 0.1 μ F (100 nF), 10-25V capacitor is recommended. The capacitor should be a low-ESR device, with a resonance frequency in the range of 200 MHz and higher. Ceramic capacitors are recommended.
- Placement on the printed circuit board: The decoupling capacitors should be placed as close to the pins as possible. It is recommended to place the capacitors on the same side of the board as the device. If space is constricted, the capacitor can be placed on another layer on the PCB using a via; however, ensure that the trace length from the pin to the capacitor is no greater than 0.25 inch (6 mm).

- Handling high-frequency noise: If the board is experiencing high-frequency noise (upward of tens of MHz), add a second ceramic type capacitor in parallel to the above described decoupling capacitor. The value of the second capacitor can be in the range of 0.01 μF to 0.001 μF . Place this second capacitor next to each primary decoupling capacitor. In high-speed circuit designs, consider implementing a decade pair of capacitances as close to the power and ground pins as possible (e.g., 0.1 μF in parallel with 0.001 μF).
- Maximizing performance: On the board layout from the power supply circuit, run the power and return traces to the decoupling capacitors first, and then to the device pins. This ensures that the decoupling capacitors are first in the power chain. Equally important is to keep the trace length between the capacitor and the power pins to a minimum, thereby reducing PCB trace inductance.

2.2.2 Tank Capacitors

On boards with power traces running longer than six inches in length, it is suggested to use a tank capacitor for integrated circuits, including microcontrollers, to supply a local power source. The value of the tank capacitor should be determined based on the trace resistance that connects the power supply source to the device, and the maximum current drawn by the device in the application. In other words, select the tank capacitor that meets the acceptable voltage sag at the device. Typical values range from 4.7 μF to 47 μF .

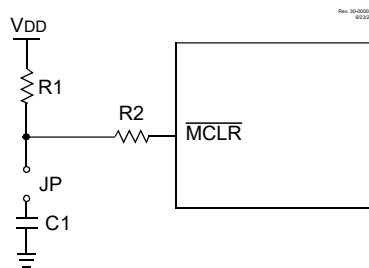
2.3 Master Clear ($\overline{\text{MCLR}}$) Pin

The $\overline{\text{MCLR}}$ pin provides two specific device functions: Device Reset, and Device Programming and Debugging. If programming and debugging are not required in the end application, a direct connection to V_{DD} may be all that is required. The addition of other components, to help increase the application's resistance to spurious Resets from voltage sags, may be beneficial. A typical configuration is shown in [Figure 2-1](#). Other circuit designs may be implemented, depending on the application's requirements.

During programming and debugging, the resistance and capacitance that can be added to the pin must be considered. Device programmers and debuggers drive the $\overline{\text{MCLR}}$ pin. Consequently, specific voltage levels (V_{IH} and V_{IL}) and fast signal transitions must not be adversely affected. Therefore, specific values of R1 and C1 will need to be adjusted based on the application and PCB requirements. For example, it is recommended that the capacitor, C1, be isolated from the $\overline{\text{MCLR}}$ pin during programming and debugging operations by using a jumper ([Figure 2-2](#)). The jumper is replaced for normal run-time operations.

Any components associated with the $\overline{\text{MCLR}}$ pin should be placed within 0.25 inch (6 mm) of the pin.

Figure 2-2. Example of $\overline{\text{MCLR}}$ Pin Connections



Note:

1. $R1 \leq 10 \text{ k}\Omega$ is recommended. A suggested starting value is 10 k Ω . Ensure that the $\overline{\text{MCLR}}$ pin V_{IH} and V_{IL} specifications are met.
2. $R2 \leq 470\Omega$ will limit any current flowing into $\overline{\text{MCLR}}$ from the extended capacitor, C1, in the event of $\overline{\text{MCLR}}$ pin breakdown, due to Electrostatic Discharge (ESD) or Electrical Overstress (EOS). Ensure that the $\overline{\text{MCLR}}$ pin V_{IH} and V_{IL} specifications are met.

2.4 In-Circuit Serial Programming™ (ICSP™) Pins

The ICSPCLK and ICSPDAT pins are used for ICSP and debugging purposes. It is recommended to keep the trace length between the ICSP connector and the ICSP pins on the device as short as possible. If the ICSP connector is expected to experience an ESD event, a series resistor is recommended, with the value in the range of a few tens of Ohms, not to exceed 100 Ω .

Pull-up resistors, series diodes and capacitors on the ICSPCLK and ICSPDAT pins are not recommended as they can interfere with the programmer/debugger communications to the device. If such discrete components are an application requirement, they should be removed from the circuit during programming and debugging. Alternatively, refer to the AC/DC characteristics and timing requirements information in the respective device Flash programming specification for information on capacitive loading limits, and pin input voltage high (V_{IH}) and input low (V_{IL}) requirements.

For device emulation, ensure that the “Communication Channel Select” (i.e., ICSPCLK/ICSPDAT pins), programmed into the device, matches the physical connections for the ICSP to the Microchip debugger/emulator tool.

For more information on available Microchip development tools connection requirements, refer to the “**Development Support**” section.

Related Links

[39. Development Support](#)

2.5 External Oscillator Pins

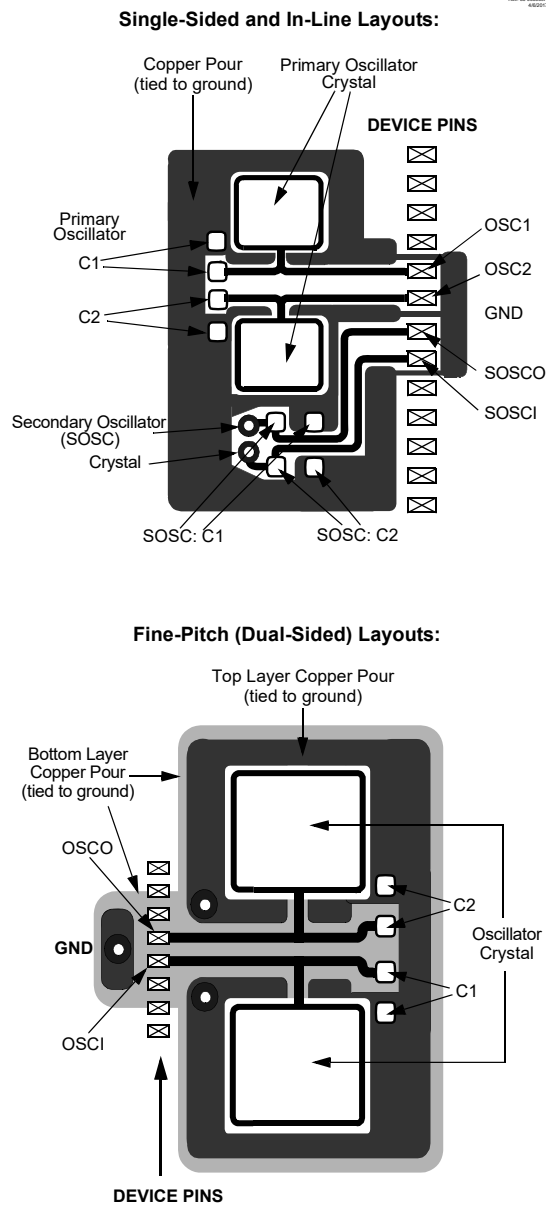
Many microcontrollers have options for at least two oscillators: a high-frequency primary oscillator and a low-frequency secondary oscillator.

The oscillator circuit should be placed on the same side of the board as the device. Place the oscillator circuit close to the respective oscillator pins with no more than 0.5 inch (12 mm) between the circuit components and the pins. The load capacitors should be placed next to the oscillator itself, on the same side of the board.

Use a grounded copper pour around the oscillator circuit to isolate it from surrounding circuits. The grounded copper pour should be routed directly to the MCU ground. Do not run any signal traces or power traces inside the ground pour. Also, if using a two-sided board, avoid any traces on the other side of the board where the crystal is placed.

Layout suggestions are shown in the following figure. In-line packages may be handled with a single-sided layout that completely encompasses the oscillator pins. With fine-pitch packages, it is not always possible to completely surround the pins and components. A suitable solution is to tie the broken guard sections to a mirrored ground layer. In all cases, the guard trace(s) must be returned to ground.

Figure 2-3. Suggested Placement of the Oscillator Circuit



In planning the application's routing and I/O assignments, ensure that adjacent port pins, and other signals in close proximity to the oscillator, are benign (i.e., free of high frequencies, short rise and fall times, and other similar noise).

For additional information and design guidance on oscillator circuits, refer to these Microchip Application Notes, available at the corporate website (www.microchip.com):

- AN826, "Crystal Oscillator Basics and Crystal Selection for rPIC™ and PICmicro® Devices"
- AN849, "Basic PICmicro® Oscillator Design"
- AN943, "Practical PICmicro® Oscillator Analysis and Design"
- AN949, "Making Your Oscillator Work"

Related Links

[8. OSC - Oscillator Module](#)

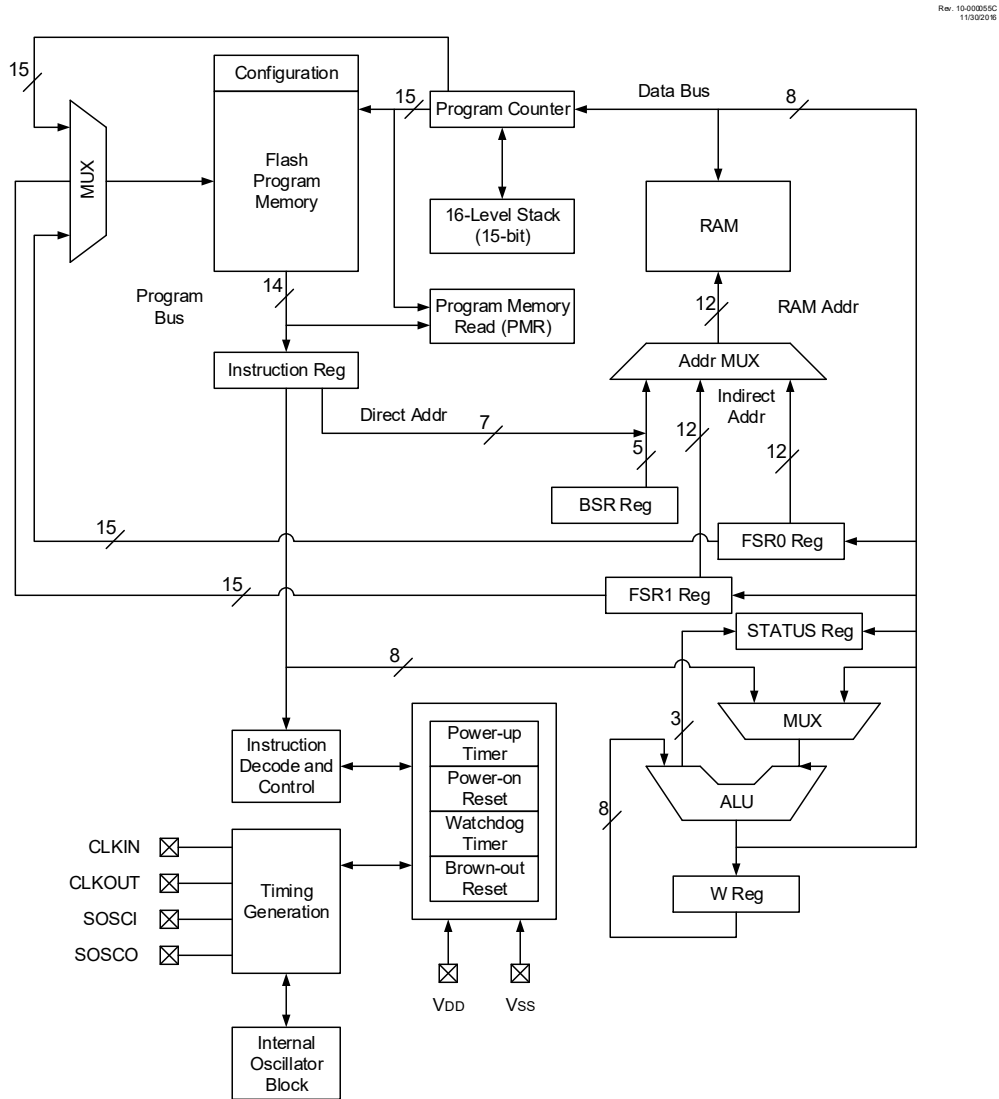
2.6 Unused I/Os

Unused I/O pins should be configured as outputs and driven to a logic low state. Alternatively, connect a 1 k Ω to 10 k Ω resistor to V_{SS} on unused pins to drive the output to logic low.

3. Enhanced Mid-Range CPU

This family of devices contains an enhanced mid-range 8-bit CPU core. The CPU has 50 instructions. Interrupt capability includes automatic context saving. The hardware stack is 16-levels deep and has Overflow and Underflow Reset capability. Direct, Indirect, and Relative Addressing modes are available. Two File Select Registers (FSRs) provide the ability to read program and data memory.

Figure 3-1. Core Data Path Diagram



3.1 Automatic Interrupt Context Saving

During interrupts, certain registers are automatically saved in shadow registers and restored when returning from the interrupt. This saves stack space and user code.

Related Links

[7.5 Automatic Context Saving](#)

3.2 16-Level Stack with Overflow and Underflow

These devices have a hardware stack memory 15 bits wide and 16 words deep. A Stack Overflow or Underflow will set the appropriate bit (STKOVF or STKUNF) in the PCON0 register, and if enabled, will cause a software Reset.

Related Links

[5.5 Stack](#)
[10.14.2 PCON0](#)

3.3 File Select Registers

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

Related Links

[5.6 Indirect Addressing](#)

3.4 Instruction Set

There are 50 instructions for the enhanced mid-range CPU to support the features of the CPU.

Related Links

[37. Instruction Set Summary](#)

4. Device Configuration

Device configuration consists of the Configuration Words, user ID, device ID, Device Information Area (DIA), and the Device Configuration Information (DCI) regions.

Related Links

[5.12 Device Information Area](#)

[5.11 Device Configuration Information](#)

4.1 Configuration Words

There are five Configuration Words that allow the user to select the device oscillator, reset, and memory protection options. These are implemented at 8007h through 800Bh.

Note: The DEBUG bit in [Configuration Word 2](#) is managed automatically by device development tools including debuggers and programmers. For normal device operation, this bit should be maintained as a '1'.

4.2 Code Protection

Code protection allows the device to be protected from unauthorized access. Internal access to the program memory is unaffected by any code protection setting. A single code protect bit controls the access for both program memory and data EEPROM memory.

The entire program memory and Data EEPROM space is protected from external reads and writes by the \overline{CP} bit. When $\overline{CP} = 0$, external reads and writes of program memory are inhibited and a read will return all '0's. The CPU can continue to read program memory, regardless of the protection bit settings. Self-writing the program memory is dependent upon the write protection setting.

4.3 Write Protection

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

The [WRT](#) bits define the size of the program memory block that is protected.

4.4 User ID

Four words in the memory space (8000h-8003h) are designated as ID locations where the user can store checksum or other code identification numbers. These locations are readable and writable during normal execution. See the *"NVMREG Access to Device Information Area, Device Configuration Area, User ID, Device ID, EEPROM, and Configuration Words"* section for more information on accessing these memory locations. For more information on checksum calculation, see the *"PIC16(L)F184XX Memory Programming Specification"*, (DS40001970).

Related Links

[6.4.7 NVMREG Access to Device Information Area, Device Configuration Area, User ID, Device ID, EEPROM, and Configuration Words](#)

4.5 Device ID and Revision ID

The 14-bit Device ID word is located at 0x8006 and the 14-bit revision ID is located at 0x8005. These locations are read-only and cannot be erased or modified.

Development tools, such as device programmers and debuggers, may be used to read the Device ID, Revision ID and Configuration Words. Refer to the **"Nonvolatile Memory (NVM) Control"** section for more information on accessing these locations.

Related Links

[6. NVM - Nonvolatile Memory Control](#)

4.6 Register Summary - Configuration Words

| Offset | Name | Bit Pos. | | | | | | | | |
|--------|---------|----------|-----------|-------------|---------|---------|--------------|--------------|-------|------|
| 0x8007 | CONFIG1 | 7:0 | | RSTOSC[2:0] | | | | FEXTOSC[2:0] | | |
| | | 13:8 | | FCMEN | | CSWEN | | CLKOUTEN | | |
| 0x8008 | CONFIG2 | 7:0 | BOREN | | LPBOREN | | PWRTS[1:0] | | MCLRE | |
| | | 13:8 | | DEBUG | STVREN | PPS1WAY | ZCDDIS | BORV | | |
| 0x8009 | CONFIG3 | 7:0 | WDTE[1:0] | | | | WDTCCPS[4:0] | | | |
| | | 13:8 | | WDTCCS[2:0] | | | | WDTCWS[2:0] | | |
| 0x800A | CONFIG4 | 7:0 | WRTAPP | | | SAFEN | BBEN | BBSIZE[2:0] | | |
| | | 13:8 | | | LVP | | WRSAF | WRTD | WRTC | WRTB |
| 0x800B | CONFIG5 | 7:0 | | | | | | | | CP |
| | | 13:8 | | | | | | | | |

4.7 Register Definitions: Configuration Words

4.7.1 CONFIG1

Name: CONFIG1

Address: 0x8007

Configuration Word 1

Oscillators

| | | | | | | | | |
|--------|----|-------------|-------|-----|-------|--------------|-----|----------|
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| | | | FCMEN | | CSWEN | | | CLKOUTEN |
| Access | | | R/P | U | R/P | U | U | R/P |
| Reset | | | 1 | 1 | 1 | 1 | 1 | 1 |
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | RSTOSC[2:0] | | | | FEXTOSC[2:0] | | |
| Access | U | R/P | R/P | R/P | U | R/P | R/P | R/P |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Bit 13 – FCMEN Fail-Safe Clock Monitor Enable bit

| Value | Description |
|-------|---------------------|
| 1 | FSCM timer enabled |
| 0 | FSCM timer disabled |

Bit 11 – CSWEN Clock Switch Enable bit

| Value | Description |
|-------|---|
| 1 | Writing to NOSC and NDIV is allowed |
| 0 | The NOSC and NDIV bits cannot be changed by user software |

Bit 8 – CLKOUTEN Clock Out Enable bit

| Value | Condition | Description |
|-------|---|---|
| 1 | If FEXTOSC = EC (high, mid or low) or Not Enabled | CLKOUT function is disabled; I/O or oscillator function on OSC2 |
| 0 | If FEXTOSC = EC (high, mid or low) or Not Enabled | CLKOUT function is enabled; F _{OSC} /4 clock appears at OSC2 |
| | Otherwise | This bit is ignored. |

Bits 6:4 – RSTOSC[2:0] Power-up Default Value for COSC bits

This value is the Reset default value for COSC and selects the oscillator first used by user software. Refer to COSC operation.

| Value | Description |
|-------|---|
| 111 | EXTOSC operating per FEXTOSC bits |
| 110 | HFINTOSC (1 MHz), with OSCFRQ = '010' (4 MHz) and CDIV = '0010' (4:1) |
| 101 | LFINTOSC |
| 100 | SOSC |
| 011 | Reserved |
| 010 | EXTOSC with 4x PLL, with EXTOSC operating per FEXTOSC bits |
| 001 | HFINTOSC with 2x PLL (32 MHz), with OSCFRQ = '101' (16 MHz) and CDIV = '0000' (1:1) |
| 000 | HFINTOSC with OSCFRQ = 32 MHz and CDIV = 1:1 |

Bits 2:0 – FEXTOSC[2:0] FEXTOSC External Oscillator Mode Selection bits

| Value | Description |
|-------|---|
| 111 | ECH (External Clock) above 8 MHz |
| 110 | ECM (External Clock) for 500 kHz to 8 MHz |
| 101 | ECL (External Clock) below 500 kHz |
| 100 | Oscillator not enabled |
| 011 | Reserved (do not use) |

PIC16(L)F18426/46

Device Configuration

| Value | Description |
|-------|--|
| 010 | HS (Crystal oscillator) above 4 MHz |
| 001 | XT (Crystal oscillator) above 100 kHz, below 4 MHz |
| 000 | LP (Crystal oscillator) optimized for 32.768 kHz |

Related Links

[8.6.7 OSCFRQ](#)

[8.6.2 OSCCON2](#)

4.7.2 CONFIG2

Name: CONFIG2

Address: 0x8008

Configuration Word 2

Supervisor

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|--------|------------|-----|---------|--------|---------|------------|------|-------|
| | | | DEBUG | STVREN | PPS1WAY | ZCD | BORV | |
| Access | | | R/P | R/P | R/P | R/P | R/P | U |
| Reset | | | 1 | 1 | 1 | 1 | 1 | 1 |
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | BOREN[1:0] | | LPBOREN | | | PWRTS[1:0] | | MCLRE |
| Access | R/P | R/P | R/P | U | U | R/P | R/P | R/P |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Bit 13 – DEBUG Debugger Enable bit⁽¹⁾

| Value | Description |
|-------|------------------------------|
| 1 | Background debugger disabled |
| 0 | Background debugger enabled |

Bit 12 – STVREN Stack Overflow/Underflow Reset Enable bit

| Value | Description |
|-------|--|
| 1 | Stack Overflow or Underflow will cause a Reset |
| 0 | Stack Overflow or Underflow will not cause a Reset |

Bit 11 – PPS1WAY PPSLOCKED bit One-Way Set Enable bit

| Value | Description |
|-------|---|
| 1 | The PPSLOCKED bit can be cleared and set only once; PPS registers remain locked after one clear/set cycle |
| 0 | The PPSLOCKED bit can be set and cleared repeatedly (subject to the unlock sequence) |

Bit 10 – ZCD ZCD Control bit

| Value | Description |
|-------|--|
| 1 | ZCD disabled. ZCD can be enabled by setting the ZCDSEN bit of the ZCDCON register. |
| 0 | ZCD always enabled, ZCDSEN bit is ignored |

Bit 9 – BORV Brown-out Reset Voltage Selection bit⁽²⁾

| Value | Description |
|-------|--|
| 1 | Brown-out Reset voltage (V_{BOR}) set to lower trip point level |
| 0 | Brown-out Reset voltage (V_{BOR}) set to higher trip point level |

Bits 7:6 – BOREN[1:0] Brown-out Reset Enable bits

When enabled, Brown-out Reset Voltage (V_{BOR}) is set by BORV bit

| Value | Description |
|-------|---|
| 11 | Brown-out Reset enabled, SBOREN bit is ignored |
| 10 | Brown-out Reset enabled while running, disabled in Sleep; SBOREN is ignored |
| 01 | Brown-out Reset enabled according to SBOREN |
| 00 | Brown-out Reset disabled |

Bit 5 – LPBOREN Low-Power BOR Enable bit

| Value | Description |
|-------|---------------------------------------|
| 1 | Low-Power Brown-out Reset is disabled |
| 0 | Low-Power Brown-out Reset is enabled |

Bits 2:1 – PWRTS[1:0] Power-up Timer Selection bits

| Value | Description |
|-------|-------------------|
| 11 | PWRT disabled |
| 10 | PWRT set at 64 ms |
| 01 | PWRT set at 16 ms |
| 00 | PWRT set at 1 ms |

Bit 0 – MCLRE Master Clear ($\overline{\text{MCLR}}$) Enable bit

| Value | Condition | Description |
|-------|------------|---|
| | If LVP = 1 | RE3 pin function is MCLR (it will reset the device when driven low) |
| 1 | If LVP = 0 | MCLR pin is MCLR (it will reset the device when driven low) |
| 0 | If LVP = 0 | MCLR pin function is port defined function |

Note:

1. The $\overline{\text{DEBUG}}$ bit in the [Configuration Word 2](#) is managed automatically by device development tools including debuggers and programmers. For normal device operation, this bit should be maintained as a '1'.
2. See V_{BOR} parameter in the “*Electrical Specifications*” chapter for specific trip point voltages.

Related Links

[40.4.5 Reset, WDT, Oscillator Start-up Timer, Power-up Timer, Brown-Out Reset and Low-Power Brown-Out Reset Specifications](#)

4.7.3 CONFIG3

Name: CONFIG3

Address: 0x8009

Configuration Word 3

Windowed Watchdog Timer

| | | | | | | | | |
|--------|-------------|-----|-----|-------------|-----|-----|-----|-----|
| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| | WDTCSS[2:0] | | | WDTWWS[2:0] | | | | |
| Access | | | R/P | R/P | R/P | R/P | R/P | R/P |
| Reset | | | 1 | 1 | 1 | 1 | 1 | 1 |
| | | | | | | | | |
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | WDTE[1:0] | | | WDTWPS[4:0] | | | | |
| Access | U | R/P | R/P | R/P | R/P | R/P | R/P | R/P |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Bits 13:11 – WDTCSS[2:0] WDT Input Clock Selector bits

| Value | Description |
|------------|---|
| 111 | Software Control |
| 110 to 011 | Reserved |
| 010 | 32 kHz SOSC |
| 001 | WDT reference clock is the 31.25 kHz HFINTOSC (MFINTOSC) output |
| 000 | WDT reference clock is the 31.0 kHz LFINTOSC |

Bits 10:8 – WDTWWS[2:0] WDT Window Select bits

| WDTWWS | WDTCON1 [WINDOW] at POR | | | Software control of WINDOW? | Keyed access required? |
|--------|-------------------------|------------------------------|--------------------------------|-----------------------------|------------------------|
| | Value | Window delay Percent of time | Window opening Percent of time | | |
| 111 | 111 | n/a | 100 | Yes | No |
| 110 | 110 | n/a | 100 | No | Yes |
| 101 | 101 | 25 | 75 | | |
| 100 | 100 | 37.5 | 62.5 | | |
| 011 | 011 | 50 | 50 | | |
| 010 | 010 | 62.5 | 37.5 | | |
| 001 | 001 | 75 | 25 | | |
| 000 | 000 | 87.5 | 12.5 | | |

Bits 6:5 – WDTE[1:0] WDT Operating Mode bits

| Value | Description |
|-------|---|
| 11 | WDT enabled regardless of Sleep; SEN is ignored |
| 10 | WDT enabled while Sleep = 0, suspended when Sleep = 1; SEN bit is ignored |
| 01 | WDT enabled/disabled by SEN bit |
| 00 | WDT disabled, SEN bit is ignored |

PIC16(L)F18426/46

Device Configuration

Bits 4:0 – WDTCP5[4:0] WDT Period Select bits

| WDTCP5 | WDTCON0[WDTPS] at POR | | | Software Control of WDTPS? | |
|--------|-----------------------|---------------|---|----------------------------|-----|
| | Value | Divider Ratio | Typical Time Out ($F_{IN} = 31 \text{ kHz}$) | | |
| 11111 | 01011 | 1:65536 | 2^{16} | 2s | Yes |
| 11110 | 11110 | 1:32 | 2^5 | 1 ms | No |
| ... | ... | | | | |
| 10011 | 10011 | | | | |
| 10010 | 10010 | 1:8388608 | 2^{23} | 256s | No |
| 10001 | 10001 | 1:4194304 | 2^{22} | 128s | |
| 10000 | 10000 | 1:2097152 | 2^{21} | 64s | |
| 01111 | 01111 | 1:1048576 | 2^{20} | 32s | |
| 01110 | 01110 | 1:524288 | 2^{19} | 16s | |
| 01101 | 01101 | 1:262144 | 2^{18} | 8s | |
| 01100 | 01100 | 1:131072 | 2^{17} | 4s | |
| 01011 | 01011 | 1:65536 | 2^{16} | 2s | |
| 01010 | 01010 | 1:32768 | 2^{15} | 1s | |
| 01001 | 01001 | 1:16384 | 2^{14} | 512 ms | |
| 01000 | 01000 | 1:8192 | 2^{13} | 256 ms | |
| 00111 | 00111 | 1:4096 | 2^{12} | 128 ms | |
| 00110 | 00110 | 1:2048 | 2^{11} | 64 ms | |
| 00101 | 00101 | 1:1024 | 2^{10} | 32 ms | |
| 00100 | 00100 | 1:512 | 2^9 | 16 ms | |
| 00011 | 00011 | 1:256 | 2^8 | 8 ms | |
| 00010 | 00010 | 1:128 | 2^7 | 4 ms | |
| 00001 | 00001 | 1:64 | 2^6 | 2 ms | |
| 00000 | 00000 | 1:32 | 2^5 | 1 ms | |

4.7.4 CONFIG4

Name: CONFIG4

Address: 0x800A

Configuration Word 4

Memory Write Protection

| | | | | | | | | |
|--------|--------|----|-----|-------|--------|-------------|------|------|
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| | | | LVP | | WRTSAF | WRTD | WRTC | WRTB |
| Access | | | R/P | U | R/P | R/P | R/P | R/P |
| Reset | | | 1 | 1 | 1 | 1 | 1 | 1 |
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | WRTAPP | | | SAFEN | BBEN | BBSIZE[2:0] | | |
| Access | R/P | U | U | R/P | R/P | R/P | R/P | R/P |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Bit 13 – LVP Low-Voltage Programming Enable bit

The LVP bit cannot be written (to zero) while operating from the LVP programming interface. The purpose of this rule is to prevent the user from dropping out of LVP mode while programming from LVP mode, or accidentally eliminating LVP mode from the Configuration state.

The preconditioned (erased) state for this bit is critical.

| Value | Description |
|-------|---|
| 1 | Low-voltage programming enabled. MCLR/V _{PP} pin function is MCLR. MCLRE Configuration bit is ignored. |
| 0 | HV on MCLR/V _{PP} must be used for programming |

Bit 11 – WRTSAF Storage Area Flash Write Protection bit⁽¹⁾

| Value | Description |
|-------|-------------------------|
| 1 | SAF NOT write-protected |
| 0 | SAF write-protected |

Bit 10 – WRTD Data EEPROM Write Protection bit⁽¹⁾

| Value | Description |
|-------|---------------------------------|
| 1 | Data EEPROM NOT write-protected |
| 0 | Data EEPROM write-protected |

Bit 9 – WRTC Configuration Register Write Protection bit⁽¹⁾

| Value | Description |
|-------|---|
| 1 | Configuration Registers NOT write-protected |
| 0 | Configuration Registers write-protected |

Bit 8 – WRTB Boot Block Write Protection bit⁽¹⁾

| Value | Description |
|-------|--------------------------------|
| 1 | Boot Block NOT write-protected |
| 0 | Boot Block write-protected |

Bit 7 – WRTAPP Application Block Write Protection bit⁽¹⁾

| Value | Description |
|-------|---------------------------------------|
| 1 | Application Block NOT write-protected |
| 0 | Application Block write-protected |

Bit 4 – SAFEN SAF Enable bit⁽¹⁾

| Value | Description |
|-------|--------------|
| 1 | SAF disabled |

| Value | Description |
|-------|-------------|
| 0 | SAF enabled |

Bit 3 – $\overline{\text{BBEN}}$ Boot Block Enable bit⁽¹⁾

| Value | Description |
|-------|---------------------|
| 1 | Boot Block disabled |
| 0 | Boot Block enabled |

Bits 2:0 – $\text{BBSIZE}[2:0]$ Boot Block Size Selection bits

BBSIZE is used only when $\overline{\text{BBEN}} = 0$

BBSIZE bits can only be written while $\overline{\text{BBEN}} = 1$; after $\overline{\text{BBEN}} = 0$, BBSIZ is write-protected.

Table 4-1. Boot Block Size Bits

| $\overline{\text{BBEN}}$ | BBSIZE | Actual Boot Block Size User Program Memory Size (words) | Last Boot Block Memory Access |
|--------------------------|-----------------|---|-------------------------------|
| | | PIC16(L)F18426/46 | |
| 1 | xxx | 0 | — |
| 0 | 111 | 512 | 01FFh |
| 0 | 110 | 1024 | 03FFh |
| 0 | 101 | 2048 | 07FFh |
| 0 | 100 | 4096 | 0FFFh |
| 0 | 011-000 | 8192 | 1FFFh |

Note: The maximum boot block size is half the user program memory size. All selections higher than the maximum are set to half size. For example, all $\text{BBSIZE} = 000 - 100$ produce a boot block size of 4 kW on a 8 kW device.

Note:

1. Bits are implemented as sticky bits. Once protection is enabled, it can only be reset through a Bulk Erase.

4.7.5 CONFIG5

Name: CONFIG5

Address: 0x800B

Configuration Word 5

Code Protection

| | | | | | | | | |
|--------|----|----|----|----|----|----|---|-----|
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Access | | | U | U | U | U | U | U |
| Reset | | | 1 | 1 | 1 | 1 | 1 | 1 |
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Access | U | U | U | U | U | U | U | R/P |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Bit 0 – \overline{CP} Program Flash Memory Code Protection bit

| Value | Description |
|-------|---|
| 1 | Program Flash Memory code protection disabled |
| 0 | Program Flash Memory code protection enabled |

4.8 Register Summary - Device and Revision

| Offset | Name | Bit Pos. | | | | | | | |
|--------|-------------|----------|-------------|---|-------------|-------------|--|--|--|
| 0x8005 | REVISION ID | 7:0 | MJRREV[1:0] | | MNRREV[5:0] | | | | |
| | | 13:8 | | 1 | 0 | MJRREV[5:2] | | | |
| 0x8006 | DEVICE ID | 7:0 | DEV[7:0] | | | | | | |
| | | 13:8 | 1 | 1 | DEV[11:8] | | | | |

4.9 Register Definitions: Device and Revision

4.9.1 REVISION ID

Name: REVISION ID
Address: 0x8005

Revision ID Register

| | | | | | | | | | |
|--------|-----|-------------|----|----|----|-------------|----|---|---|
| | Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| | | | | 1 | 0 | MJRREV[5:2] | | | |
| Access | | | | R | R | R | R | R | R |
| Reset | | | | 1 | 0 | | | | |
| | | | | | | | | | |
| | Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | MJRREV[1:0] | | | | MNRREV[5:0] | | | |
| Access | | R | R | R | R | R | R | R | R |
| Reset | | | | | | | | | |

Bit 13 – 1 Read as ‘1’
 These bits are fixed with value ‘1’ for all devices in this family.

Bit 12 – 0 Read as ‘0’
 These bits are fixed with value ‘0’ for all devices in this family.

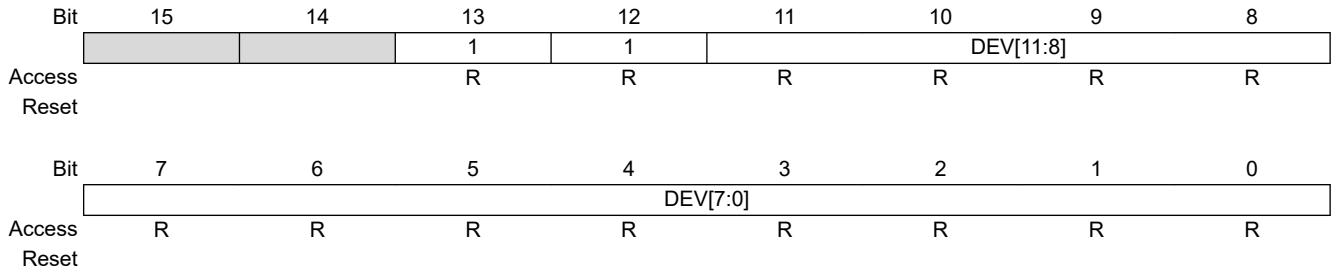
Bits 11:6 – MJRREV[5:0] Major Revision ID bits
 These bits are used to identify a major revision.

Bits 5:0 – MNRREV[5:0] Minor Revision ID bits
 These bits are used to identify a minor revision.

4.9.2 DEVICE ID

Name: DEVICE ID
Address: 0x8006

Device ID Register



Bit 13 – 1

These bit must be '1' to be distinguishable from the previous Device ID scheme

Bit 12 – 1

These bit must be '1' to be distinguishable from the previous Device ID scheme

Bits 11:0 – DEV[11:0]

Device ID bits

| Device | Device ID |
|--------------|-----------|
| PIC16F18426 | 30D2h |
| PIC16LF18426 | 30D3h |
| PIC16F18446 | 30D4h |
| PIC16LF18446 | 30D5h |

5. Memory Organization

These devices contain the following types of memory:

- Program Memory
 - Configuration Words
 - Device ID
 - User ID
 - Program Flash Memory
 - Device Information Area (DIA)
 - Device Configuration Information (DCI)
 - Revision ID
- Data Memory
 - Core Registers
 - Special Function Registers
 - General Purpose RAM
 - Common RAM
- Data EEPROM

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

- PCL and PCLATH
- Stack
- Indirect Addressing
- NVMREG access

5.1 Program Memory Organization

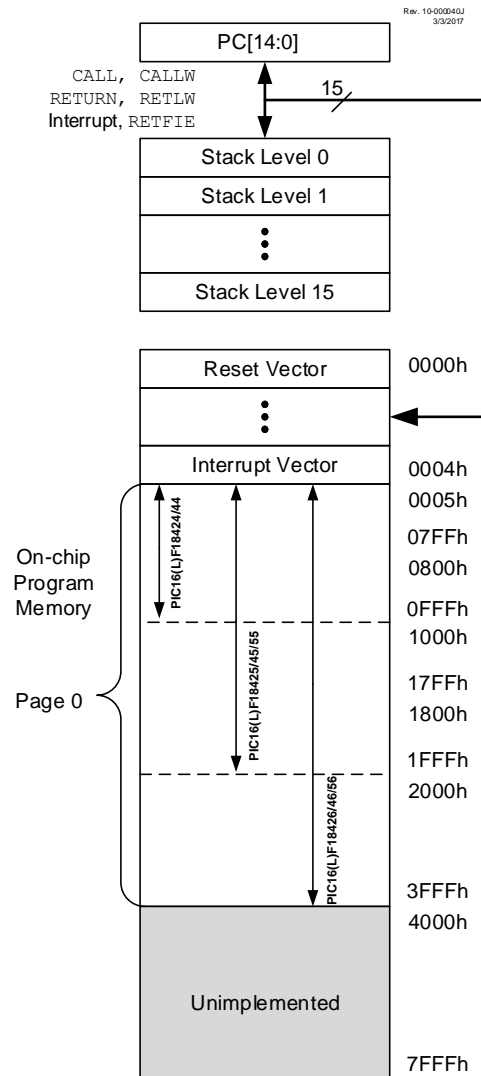
The enhanced mid-range core has a 15-bit Program Counter capable of addressing 32K x 14 program memory space. The table below shows the memory sizes implemented. Accessing a location above these boundaries will cause a wrap-around within the implemented memory space.

The Reset vector is at 0000h and the interrupt vector is at 0004h. Refer to the “*Interrupt Controller*” chapter for more details.

Table 5-1. Device Sizes And Addresses

| Device | Program Memory Size (Words) | Last Program Memory Address |
|----------------|-----------------------------|-----------------------------|
| PIC16(L)F18426 | 16384 | 0x3FFF |
| PIC16(L)F18446 | 16384 | 0x3FFF |

Figure 5-1. Program Memory and Stack



Related Links

- [4.7.5 CONFIG5](#)
- [5.2.5 Memory Violation](#)

5.1.1 Reading Program Memory as Data

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

Related Links

- [6.4 NVMREG Access](#)

5.1.1.1 RETLW Instruction

The RETLW instruction can be used to provide access to tables of constants. The recommended way to create such a table is shown in the following example.

Example 5-1. Accessing table of constants using RETLW instruction

```
constants
    BRW                ;Add Index in W to
                      ;program counter to
                      ;select data
    RETLW DATA0       ;Index0 data
    RETLW DATA1       ;Index1 data
    RETLW DATA2
    RETLW DATA3
my_function
;... LOTS OF CODE...
    MOVLW             DATA_INDEX
    call constants
;... THE CONSTANT IS IN W
```

The BRW instruction makes this type of table very simple to implement.

5.1.1.2 Indirect Read with FSR

The program memory can be accessed as data by setting bit 7 of an FSRxH register and reading the matching INDFx register. The MOVLW instruction will place the lower eight bits of the addressed word in the W register. Writes to the program memory cannot be performed via the INDFx registers. Instructions that read the program memory via the FSR require one extra instruction cycle to complete. The following example demonstrates reading the program memory via an FSR.

The HIGH directive will set bit 7 if a label points to a location in the program memory. This applies to the assembly code shown below.

Example 5-2. Read of Program memory using FSR register

```
constants
    RETLW DATA0       ;Index0 data
    RETLW DATA1       ;Index1 data
    RETLW DATA2
    RETLW DATA3
my_function
;... LOTS OF CODE...
    MOVLW             LOW constants
    MOVWF             FSR1L
    MOVLW             HIGH constants
    MOVWF             FSR1H
    MOVIW             2[FSR1] ;DATA2 IS IN W
```

5.2 Memory Access Partition (MAP)

User Flash is partitioned into:

- Application Block
- Boot Block
- Storage Area Flash (SAF) Block

The user can allocate the memory usage by setting the $\overline{\text{BBEN}}$ bit, selecting the size of the partition defined by BBSIZE bits and enabling the Storage Area Flash by the $\overline{\text{SAFEN}}$ bit of the [Configuration Word 4](#). Refer to the following links for the different user Flash memory partitions.

Related Links

[4.7.4 CONFIG4](#)

5.2.1 Application Block

Default settings of the Configuration bits ($\overline{\text{BBEN}} = 1$ and $\overline{\text{SAFEN}} = 1$) assign all memory in the user Flash area to the application block.

5.2.2 Boot Block

If $\overline{\text{BBEN}} = 1$, the boot block is enabled and a specific address range is allotted as the boot block based on the value of the BBSIZE bits and the sizes provided in [Configuration Word 4](#).

Related Links

[4.7.4 CONFIG4](#)

5.2.3 Storage Area Flash

Storage Area Flash (SAF) is enabled by clearing the $\overline{\text{SAFEN}}$ bit of the [Configuration Word 4](#). If enabled, the SAF block is placed at the end of memory and spans 128 words. If the Storage Area Flash (SAF) is enabled, the SAF area is not available for program execution.

Related Links

[4.7.4 CONFIG4](#)

5.2.4 Memory Write Protection

All the memory blocks have corresponding write protection fuses $\overline{\text{WRTAPP}}$, $\overline{\text{WRTB}}$ and $\overline{\text{WRTC}}$ bits in the [Configuration Word 4](#). If write-protected locations are written from NVMCON registers, the memory is not changed and the WRERR bit defined in NVMCON1 register is set as explained in the “*WRERR Bit*” section.

Related Links

[4.7.4 CONFIG4](#)

[6.6.3 NVMCON1](#)

[6.4.9 WRERR Bit](#)

5.2.5 Memory Violation

A Memory Execution Violation Reset occurs while executing an instruction that has been fetched from outside a valid execution area, clearing the MEMV bit. Refer to the “*Memory Execution Violation*” section for the available valid program execution areas and the PCON1 register definition for $\overline{\text{MEMV}}$ bit conditions.

Table 5-2. Memory Access Partition

| REG | Address | Partition | | | |
|--------|---|---|---|---|---|
| | | $\overline{\text{BBEN}} = 1$ $\overline{\text{SAFEN}} = 1$ | $\overline{\text{BBEN}} = 1$ $\overline{\text{SAFEN}} = 0$ | $\overline{\text{BBEN}} = 0$ $\overline{\text{SAFEN}} = 1$ | $\overline{\text{BBEN}} = 0$ $\overline{\text{SAFEN}} = 0$ |
| PFM | 00 0000h ... Last Block Memory Address | APPLICATION BLOCK ⁽⁴⁾ | APPLICATION BLOCK ⁽⁴⁾ | BOOT BLOCK ⁽⁴⁾ | BOOT BLOCK ⁽⁴⁾ |
| | Last Boot Block Memory Address + 1 ⁽¹⁾ ... Last Program Memory Address - 80h | | | APPLICATION BLOCK ⁽⁴⁾ | APPLICATION BLOCK ⁽⁴⁾ |
| | Last Program Memory Address - 7Fh ⁽²⁾ ... Last Program Memory Address | | SAF ⁽⁴⁾ | SAF ⁽⁴⁾ | |
| CONFIG | Config Memory Address ⁽³⁾ | CONFIG | | | |

Note:

1. Last Boot Block Memory Address is based on BBSIZE given in “*Configuration Word 4*”.
2. Last Program Memory Address is the Flash size given in the “*Program Memory Organization*”.
3. Config Memory Address are the address locations of the Configuration Words given in the “*NVMREG Access to Device Information Area, Device Configuration Area, User ID, Device ID, EEPROM, and Configuration Words*” section.
4. Each memory block has a corresponding write protection fuse defined by the $\overline{\text{WRTAPP}}$, $\overline{\text{WRTB}}$ and $\overline{\text{WRTC}}$ bits in the “*Configuration Word 4*”.

Related Links

[10.11 Memory Execution Violation](#)

[10.14.3 PCON1](#)

[4.7.4 CONFIG4](#)

[5.1 Program Memory Organization](#)

[6.4.7 NVMREG Access to Device Information Area, Device Configuration Area, User ID, Device ID, EEPROM, and Configuration Words](#)

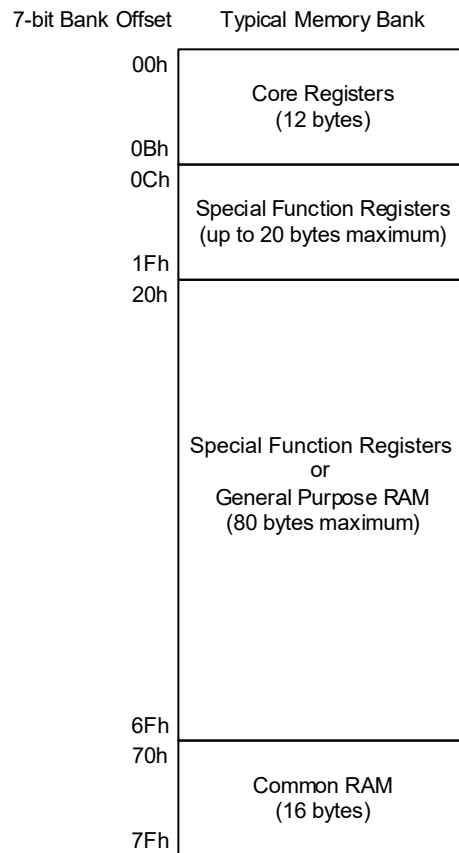
5.3 Data Memory Organization

The data memory is partitioned into up to 64 memory banks with 128 bytes in each bank. Each bank consists of:

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

Figure 5-2. Banked Memory Partition

Rev. 10-000,041C
11/8/2017



5.3.1 Bank Selection

The active bank is selected by writing the bank number into the Bank Select Register (BSR). All data memory can be accessed either directly (via instructions that use the file registers) or indirectly via the two File Select Registers (FSR). Data memory uses a 13-bit address. The upper six bits of the address define the Bank address and the lower seven bits select the registers/RAM in that bank.

Related Links

[5.6 Indirect Addressing](#)

[5.8.7 BSR](#)

5.3.2 Core Registers

The core registers contain the registers that directly affect the basic operation. The core registers occupy the first 12 addresses of every data memory bank (addresses x00h/x80h through x0Bh/x8Bh). These registers are listed below.

Table 5-3. Core Registers

| Addresses in BANKx | Core Registers |
|--------------------|----------------|
| x00h or x80h | INDF0 |
| x01h or x81h | INDF1 |

|continued | |
|--------------------|----------------|
| Addresses in BANKx | Core Registers |
| x02h or x82h | PCL |
| x03h or x83h | STATUS |
| x04h or x84h | FSR0L |
| x05h or x85h | FSR0H |
| x06h or x86h | FSR1L |
| x07h or x87h | FSR1H |
| x08h or x88h | BSR |
| x09h or x89h | WREG |
| x0Ah or x8Ah | PCLATH |
| x0Bh or x8Bh | INTCON |

5.3.2.1 STATUS Register

The STATUS register contains:

- the arithmetic status of the ALU
- the Reset status

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

For example, `CLRF STATUS` will clear bits [4:3] and [1:0], and set the Z bit. This leaves the STATUS register as '000u u1uu' (where u = unchanged).

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



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

Related Links

[5.8.4 STATUS](#)

5.3.3 Special Function Register

The Special Function Registers (SFR) are registers used by the application to control the desired operation of peripheral functions in the device. The SFRs occupy the first 20 bytes of the data banks 0-59 and the first 100 bytes of the data banks 60-63, after the core registers.

The SFRs associated with the operation of the peripherals are described in the appropriate peripheral chapter of this data sheet.

5.3.4 General Purpose RAM

There are up to 80 bytes of GPR in each data memory bank. The general purpose RAM can be accessed in a non-banked method via the FSRs. This can simplify access to large memory structures.

Refer to section [5.6.2 Linear Data Memory](#) for details about details about linear memory accessing.

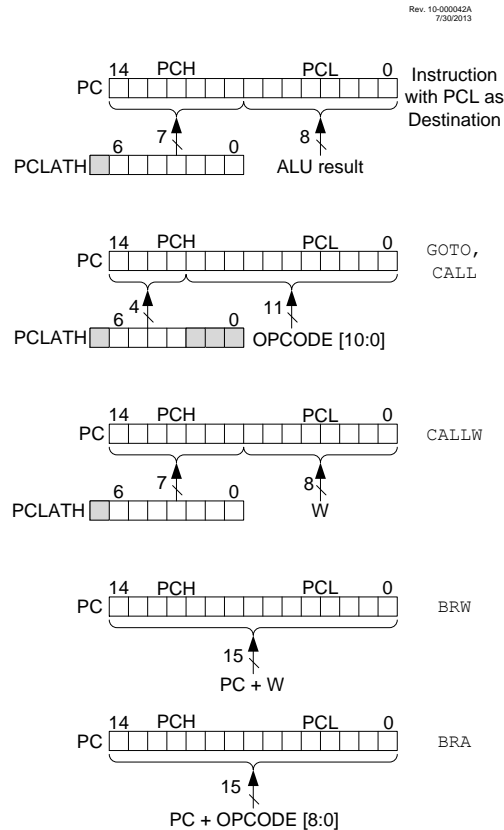
5.3.5 Common RAM

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

5.4 PCL and PCLATH

The Program Counter (PC) is 15 bits wide. The low byte comes from the PCL register, which is a readable and writable register. The high byte (PC[14:8]) is not directly readable or writable and comes from PCLATH. On any Reset the PC is cleared. The [Figure 5-3](#) shows the five situations for the loading of the PC.

Figure 5-3. Loading of PC in Different Situations



5.4.1 Modifying PCL

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

5.4.2 Computed GOTO

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

5.4.3 Computed Function Calls

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

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

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

5.4.4 Branching

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

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

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

5.5 Stack

All devices have a 16-level by 15-bit wide hardware stack. The stack space is not part of either program or data space. The PC is PUSHed onto the stack when `CALL` or `CALLW` instructions are executed or an interrupt causes a branch. The stack is POPed in the event of a `RETURN`, `RETLW` or a `RETFIE` instruction execution. `PCLATH` is not affected by a `PUSH` or `POP` operation.

The stack operates as a circular buffer if the `STVREN` Configuration bit is programmed to '0'. This means that after the stack has been PUSHed sixteen times, the seventeenth PUSH overwrites the value that was stored from the first PUSH. The eighteenth PUSH overwrites the second PUSH (and so on). The `STKOVF` and `STKUNF` flag bits will be set on an Overflow/Underflow, regardless of whether the Reset is enabled.

If the `STVREN` bit in [Configuration Word 2](#) is programmed to '1', the device will be Reset if the stack is PUSHed beyond the sixteenth level or POPed beyond the first level, setting the appropriate bits (`STKOVF` or `STKUNF`, respectively) in the `PCON` register.



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

5.5.1 Accessing the Stack

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



Important: Care should be taken when modifying the `STKPTR` while interrupts are enabled.

During normal program operation, `CALL`, `CALLW` and interrupts will increment `STKPTR` while `RETLW`, `RETURN`, and `RETFIE` will decrement `STKPTR`. `STKPTR` can be monitored to obtain to value of stack memory left at any given time. The `STKPTR` always points at the currently used place on the stack. Therefore, a `CALL` or `CALLW` will increment the `STKPTR` and then write the PC, and a return will unload the PC value from the stack and then decrement the `STKPTR`.

Reference the following figures for examples of accessing the stack.

Figure 5-4. Accessing the Stack Example 1

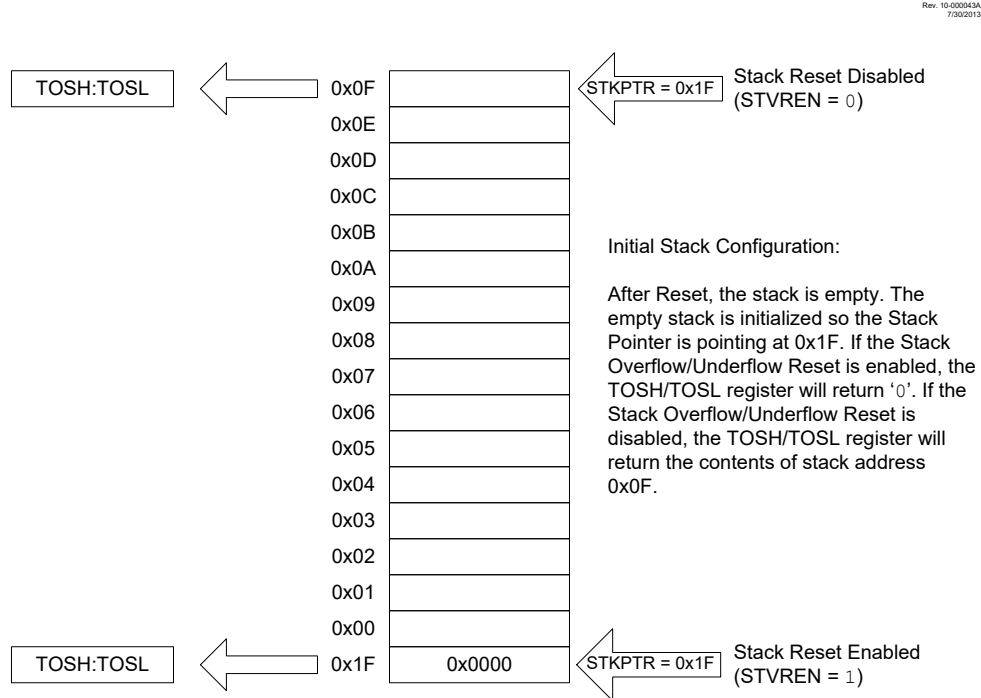


Figure 5-5. Accessing the Stack Example 2

Rev: 10-000043B
7/30/2013

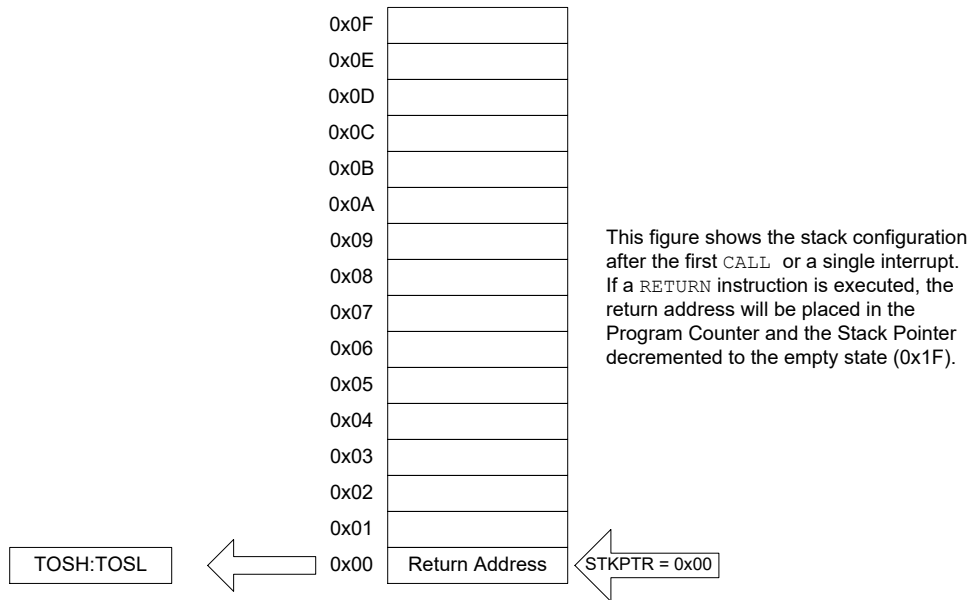


Figure 5-6. Accessing the Stack Example 3

Rev: 10-000043C
7/30/2013

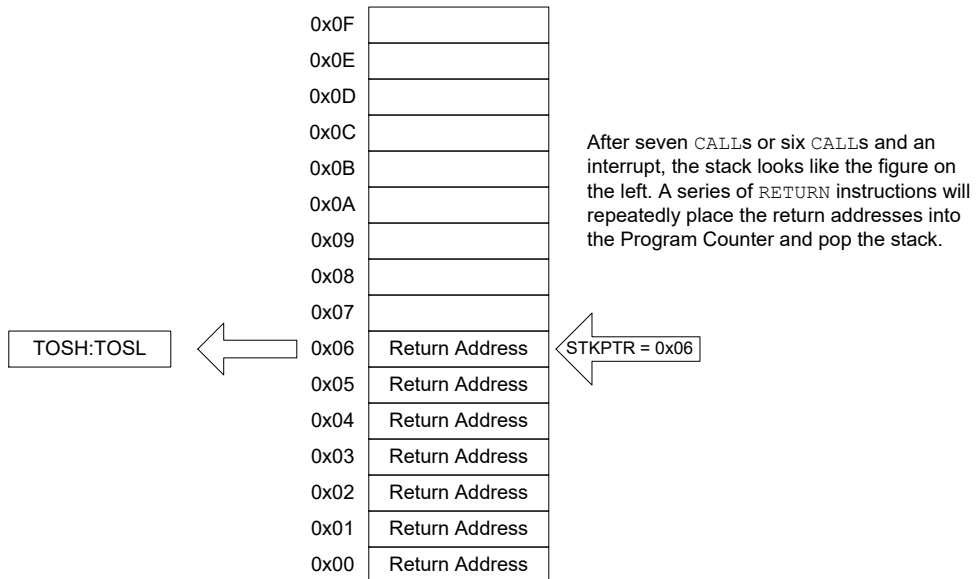
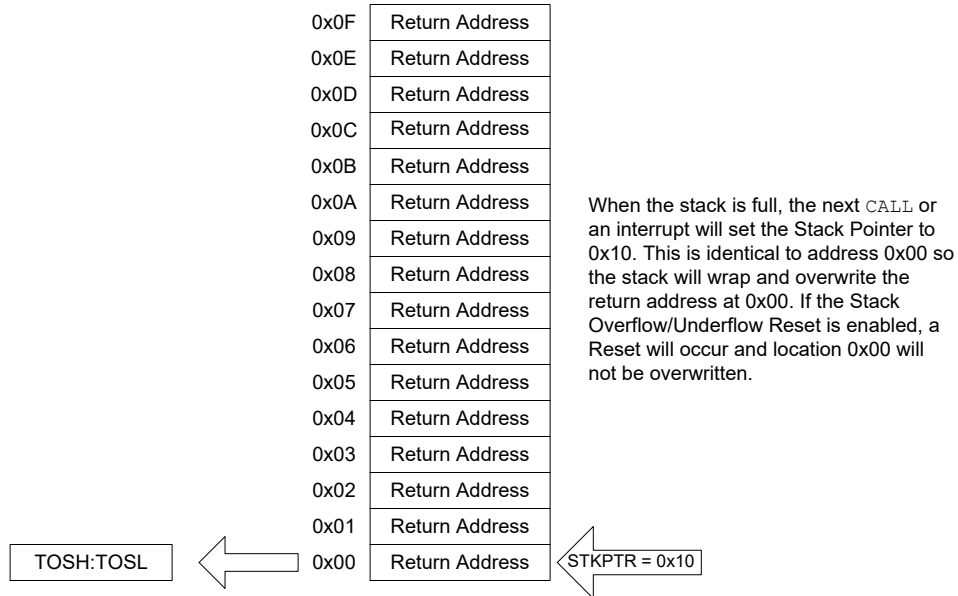


Figure 5-7. Accessing the Stack Example 4

Rev. 10-000043D
7/30/2013



Related Links

[5.8.11 TOS](#)

5.6 Indirect Addressing

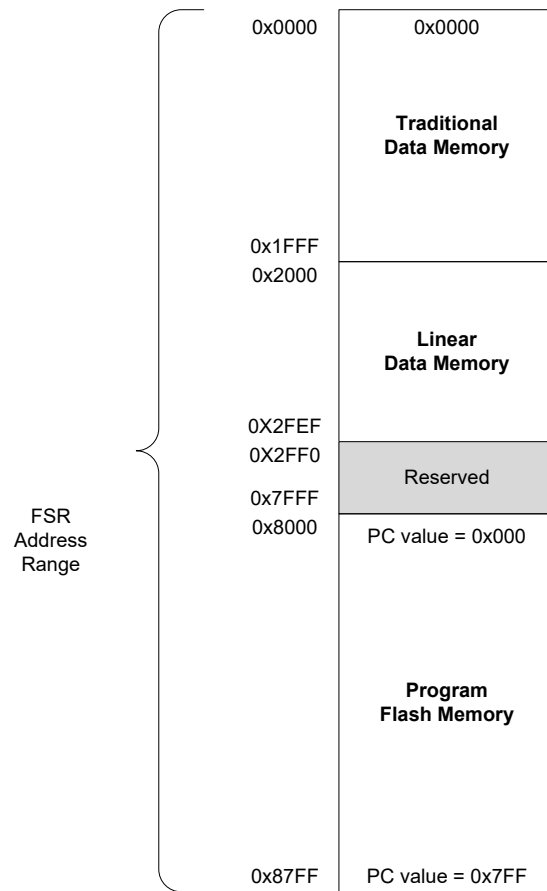
The INDFn registers are not physical registers. Any instruction that accesses an INDFn register actually accesses the register at the address specified by the File Select Registers (FSR). If the FSRn address specifies one of the two INDFn registers, the read will return '0' and the write will not occur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

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

- Traditional/Banked Data Memory
- Linear Data Memory
- Program Flash Memory
- Data EEPROM

Figure 5-8. Indirect Addressing PIC16(L)F18426/46

Rev. 10-000044F
1/13/2017



Related Links

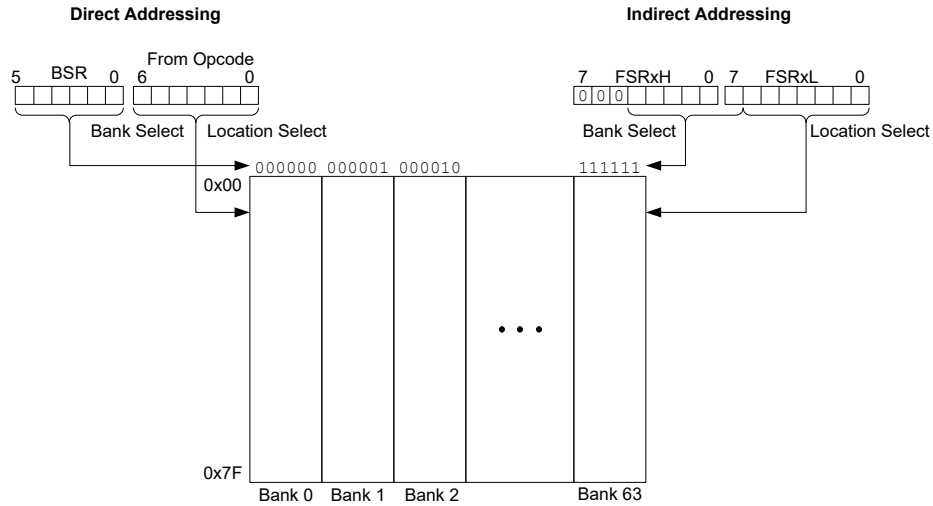
[5.8.5 FSR0](#)

5.6.1 Traditional/Banked Data Memory

The traditional or banked data memory is a region from FSR address 0x000 to FSR address 0x1FFF. The addresses correspond to the absolute addresses of all SFR, GPR and common registers.

Figure 5-9. Traditional/Banked Data Memory Map

Rev. 10-000056B
12/14/2016

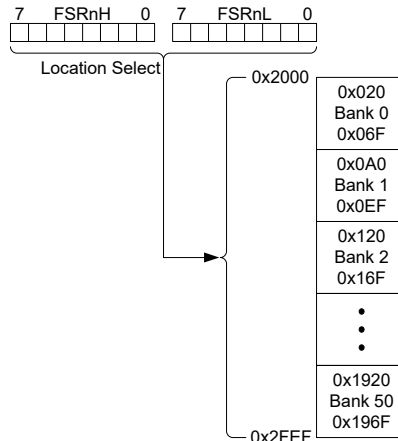


5.6.2 Linear Data Memory

The linear data memory is the region from FSR address 0x2000 to FSR address 0x2FEF. This region is a virtual region that points back to the 80-byte blocks of GPR memory in all the banks. Refer to the [Figure 5-10](#) for the Linear Data Memory Map.

Figure 5-10. Linear Data Memory Map

Rev. 10-000057B
8/24/2016



Important: The address range 0x2000 to 0x2FEF represents the complete addressable Linear Data Memory for PIC devices (up to Bank 50). The actual implemented Linear Data Memory will differ from one device to the other in a family. Refer to [General Purpose RAM Distribution](#) for the memory limits of PIC16(L)F18426/46 devices.

Table 5-4. General Purpose RAM Distribution

| Bank # | {bank:offset} | Linear address | PIC16(L)F18426 | PIC16(L)F18446 |
|--------|---------------|----------------|----------------|----------------|
| 0 | 0x020-0x06F | 0x2000-0x204F | 80 | 80 |
| 1 | 0x0A0-0x0EF | 0x2050-0x209F | 80 | 80 |
| 2 | 0x120-0x16F | 0x20A0-0x20EF | 80 | 80 |
| 3 | 0x1A0-0x1EF | 0x20F0-0x213F | 80 | 80 |
| 4 | 0x220-0x26F | 0x2140-0x218F | 80 | 80 |
| 5 | 0x2A0-0x2EF | 0x2190-0x21DF | 80 | 80 |
| 6 | 0x320-0x36F | 0x21E0-0x222F | 80 | 80 |
| 7 | 0x3A0-0x3EF | 0x2230-0x227F | 80 | 80 |
| 8 | 0x420-0x46F | 0x2280-0x22CF | 80 | 80 |
| 9 | 0x4A0-0x4EF | 0x22D0-0x231F | 80 | 80 |
| 10 | 0x520-0x56F | 0x2320-0x236F | 80 | 80 |
| 11 | 0x5A0-0x5EF | 0x2370-0x23BF | 80 | 80 |
| 12 | 0x620-0x64F | 0x23C0-0x23EF | 48 | 80 |
| | 0x620-0x66F | 0x23C0-0x240F | | |
| 13 | 0x6A0-0x6EF | 0x2410-0x245F | | 80 |
| 14 | 0x720-0x76F | 0x2460-0x24AF | | 80 |
| 15 | 0x7A0-0x7EF | 0x24B0-0x24FF | | 80 |
| 16 | 0x820-0x86F | 0x2500-0x254F | | 80 |
| 17 | 0x8A0-0x8EF | 0x2550-0x259F | | 80 |
| 18 | 0x920-0x96F | 0x25A0-0x25EF | | 80 |
| 19 | 0x9A0-0x9EF | 0x25F0-0x263F | | 80 |
| 20 | 0xA20-0xA6F | 0x2640-0x268F | | 80 |
| 21 | 0xAA0-0xAEF | 0x2690-0x26DF | | 80 |
| 22 | 0xB20-0xB6F | 0x26E0-0x272F | | 80 |
| 23 | 0xBA0-0xBEF | 0x2730-0x277F | | 80 |
| 24 | 0xC20-0xC6F | 0x2780-0x27CF | | 80 |
| 25 | 0xCA0-0xCBF | 0x27D0-0x27EF | | 32 |

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

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

5.6.3 Data EEPROM Memory

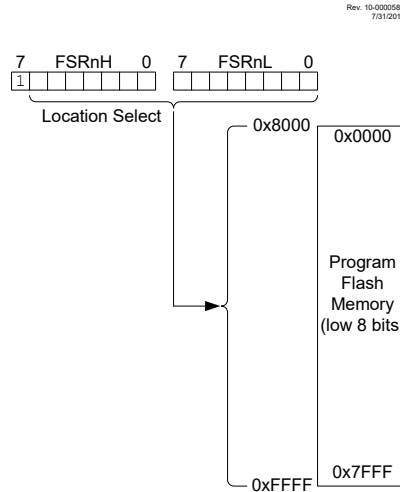
The EEPROM memory can be read or written through the NVMCON register interface (6.2 Data EEPROM). However, to make access to the EEPROM easier, read-only access to the EEPROM contents are also available through indirect addressing via an FSR. When the MSP of the FSR (ex: FSRxH) is set to 0x70, the lower 8-bit address value (in FSRxL) determines the EEPROM location that may be read via the INDF register). In other words, the EEPROM address range 0x00-0xFF is mapped into the FSR address space between 0x7000 and 0x70FF.

Writing to the EEPROM cannot be accomplished via the FSR/INDF interface. Reads from the EEPROM through the FSR/INDF interface will require one additional instruction cycle to complete.

5.6.4 Program Flash Memory

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

Figure 5-11. Program Flash Memory Map



5.7 Register Summary - Memory and Status

| Offset | Name | Bit Pos. | | | | | | | | | |
|--------|----------|----------|-----|------|--|--|----|----|---|-------------|-----------|
| 0x00 | INDF0 | 7:0 | | | | | | | | INDF[7:0] | |
| 0x01 | INDF1 | 7:0 | | | | | | | | INDF[7:0] | |
| 0x02 | PCL | 7:0 | | | | | | | | PCL[7:0] | |
| 0x03 | STATUS | 7:0 | | | | | TO | PD | Z | DC | C |
| 0x04 | FSR0 | 7:0 | | | | | | | | FSRL[7:0] | |
| 0x05 | | 15:8 | | | | | | | | FSRH[7:0] | |
| 0x06 | FSR1 | 7:0 | | | | | | | | FSRL[7:0] | |
| 0x07 | | 15:8 | | | | | | | | FSRH[7:0] | |
| 0x08 | BSR | 7:0 | | | | | | | | BSR[5:0] | |
| 0x09 | WREG | 7:0 | | | | | | | | WREG[7:0] | |
| 0x0A | PCLATH | 7:0 | | | | | | | | PCLATH[6:0] | |
| 0x0B | INTCON | 7:0 | GIE | PEIE | | | | | | INTEDG | |
| 0x0C | Reserved | | | | | | | | | | |
| 0x0FEC | | | | | | | | | | | |
| 0x0FED | STKPTR | 7:0 | | | | | | | | STKPTR[4:0] | |
| 0x0FEE | TOS | 7:0 | | | | | | | | TOSL[7:0] | |
| 0x0FEF | | 15:8 | | | | | | | | | TOSH[7:0] |

5.8 Register Definitions: Memory and Status

5.8.1 INDF0

Name: INDF0
Address: 0x00 + n*0x80 [n=0..63]

Indirect Data Register. This is a virtual register. The GPR/SFR register addressed by the FSR0 register is the target for all operations involving the INDF0 register.

| | | | | | | | | |
|--------|------------|-----|-----|-----|-----|-----|-----|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | INDF0[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bits 7:0 – INDF0[7:0]

Indirect data pointed to by the FSR0 register

Related Links

[5.3.2 Core Registers](#)

5.8.2 INDF1

Name: INDF1
Address: 0x01 + n*0x80 [n=0..63]

Indirect Data Register. This is a virtual register. The GPR/SFR register addressed by the FSR1 register is the target for all operations involving the INDF1 register.

| | | | | | | | | |
|--------|------------|-----|-----|-----|-----|-----|-----|-----|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | INDF1[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bits 7:0 – INDF1[7:0]
 Indirect data pointed to by the FSR1 register

Related Links
[5.3.2 Core Registers](#)

5.8.3 PCL

Name: PCL
Address: 0x02 + n*0x80 [n=0..63]

Low byte of the Program Counter

| | | | | | | | | |
|--------|----------|-----|-----|-----|-----|-----|-----|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | PCL[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bits 7:0 – PCL[7:0]

Provides direct read and write access to the Program Counter

5.8.4 STATUS

Name: STATUS
Address: 0x03 + n*0x80 [n=0..63]

Status Register

| | | | | | | | | |
|--------|---|---|---|----|----|-----|-----|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | | | TO | PD | Z | DC | C |
| Access | | | | RO | RO | R/W | R/W | R/W |
| Reset | | | | 1 | 1 | 0 | 0 | 0 |

Bit 4 – TO Time-Out bit

Reset States: POR/BOR = 1
All Other Resets = q

| Value | Description |
|-------|--|
| 1 | Set at power-up or by execution of CLRWDT or SLEEP instruction |
| 0 | A WDT time-out occurred |

Bit 3 – PD Power-Down bit

Reset States: POR/BOR = 1
All Other Resets = q

| Value | Description |
|-------|---|
| 1 | Set at power-up or by execution of CLRWDT instruction |
| 0 | Cleared by execution of the SLEEP instruction |

Bit 2 – Z Zero bit

Reset States: POR/BOR = 0
All Other Resets = u

| Value | Description |
|-------|--|
| 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/Borrow bit⁽¹⁾

ADDWF, ADDLW, SUBLW, SUBWF instructions
Reset States: POR/BOR = 0
All Other Resets = u

| Value | Description |
|-------|---|
| 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/Borrow bit⁽¹⁾

ADDWF, ADDLW, SUBLW, SUBWF instructions
Reset States: POR/BOR = 0
All Other Resets = u

| Value | Description |
|-------|---|
| 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:

- For Borrow, the polarity is reversed. A subtraction is executed by adding the two's complement of the second operand. For Rotate (RRCF, RLCF) instructions, this bit is loaded with either the high or low-order bit of the Source register.

Related Links

[5.3.2 Core Registers](#)

5.8.5 FSR0

Name: FSR0

Address: 0x04 + n*0x80 [n=0..63]

Indirect Address Register

The FSR0 value is the address of the data to which the INDF0 register points.

| | | | | | | | | |
|--------|------------|-----|-----|-----|-----|-----|-----|-----|
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| | FSR0[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | FSR0[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bits 15:0 – FSR0[15:0]

Address of INDF0 data

Note: The individual bytes in this multi-byte register can be accessed with the following register names:

- FSR0H: Accesses the high byte FSR0[15:8]
- FSR0L: Accesses the low byte FSR0[7:0]

Related Links

[5.3.2 Core Registers](#)

5.8.6 FSR1

Name: FSR1
Address: 0x06 + n*0x80 [n=0..63]

Indirect Address Register

The FSR1 value is the address of the data to which the INDF1 register points.

| | | | | | | | | | |
|--------|-----|------------|-----|-----|-----|-----|-----|-----|-----|
| | Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| | | FSR1[15:8] | | | | | | | |
| Access | | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | | | | | | | | |
| | Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | FSR1[7:0] | | | | | | | |
| Access | | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bits 15:0 – FSR1[15:0]
Address of INDF1 data

Note: The individual bytes in this multi-byte register can be accessed with the following register names:

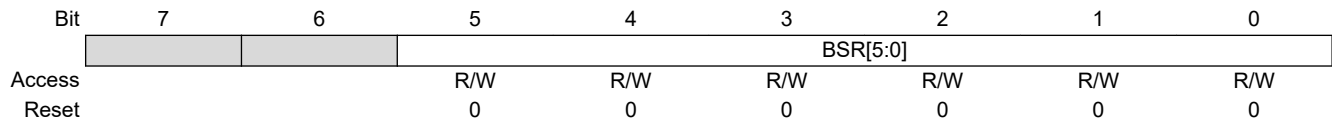
- FSR1H: Accesses the high byte FSR1[15:8]
- FSR1L: Accesses the low byte FSR1[7:0]

5.8.7 BSR

Name: BSR
Address: 0x08 + n*0x80 [n=0..63]

Bank Select Register

The BSR indicates the data memory bank by writing the bank number into the register. All data memory can be accessed directly via instructions, or indirectly via FSRs.



Bits 5:0 – BSR[5:0]

Six Most Significant bits of the data memory address

Related Links

[5.3.2 Core Registers](#)

5.8.8 WREG

Name: WREG
Address: $0x09 + n \cdot 0x80$ [n=0..63]

Working Data Register

| | | | | | | | | |
|--------|-----------|-----|-----|-----|-----|-----|-----|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | WREG[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bits 7:0 – WREG[7:0]

Related Links

[5.3.2 Core Registers](#)

5.8.9 PCLATH

Name: PCLATH
Address: 0x0A + n*0x80 [n=0..63]

Program Counter Latches.

Write Buffer for the upper 7 bits of the Program Counter

| | | | | | | | | |
|--------|-------------|-----|-----|-----|-----|-----|-----|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | PCLATH[6:0] | | | | | | | |
| Access | | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bits 6:0 – PCLATH[6:0] High PC Latch Register
Holding register for Program Counter bits [6:0]

Related Links

[5.3.2 Core Registers](#)

5.8.10 INTCON

Name: INTCON
Address: 0x0B + n*0x80 [n=0..63]

Interrupt Control Register

| | | | | | | | | |
|--------|-----|------|---|---|---|---|---|--------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | GIE | PEIE | | | | | | INTEDG |
| Access | R/W | R/W | | | | | | R/W |
| Reset | 0 | 0 | | | | | | 1 |

Bit 7 – GIE Global Interrupt Enable bit

| Value | Description |
|-------|-------------------------------|
| 1 | Enables all active interrupts |
| 0 | Disables all interrupts |

Bit 6 – PEIE Peripheral Interrupt Enable bit

| Value | Description |
|-------|--|
| 1 | Enables all active peripheral interrupts |
| 0 | Disables all peripheral interrupts |

Bit 0 – INTEDG External Interrupt Edge Select bit

| Value | Description |
|-------|--------------------------------------|
| 1 | Interrupt on rising edge of INT pin |
| 0 | Interrupt on falling edge of INT pin |



Important: Interrupt flag bits are set when an interrupt condition occurs, regardless of the state of its corresponding enable bit or the global enable bit. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt. This feature allows for software polling.

Related Links

[5.3.2 Core Registers](#)

5.8.11 TOS

Name: TOS
Address: 0xFEE

Top-of-Stack Registers

Contents of the stack pointed to by the STKPTR register. To access the stack, adjust the value of STKPTR, which will position TOSH:TOSL, then read/write to TOSH:TOSL.

| | | | | | | | | |
|--------|-----------|-----|-----|-----|-----|-----|-----|-----|
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| | TOS[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TOS[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bits 15:0 – TOS[15:0] TOS Register Value

Note: The individual bytes in this multi-byte register can be accessed with the following register names:

- TOSH: Accesses the high byte TOS[15:8]
- TOSL: Accesses the low byte TOS[7:0]

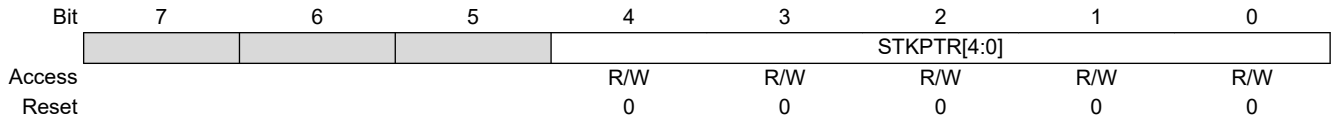
Related Links

[5.3.2 Core Registers](#)

5.8.12 STKPTR

Name: STKPTR
Address: 0xFED

Stack Pointer Register



Bits 4:0 – STKPTR[4:0] Stack Pointer Location bits

5.9 Register Summary: Shadow Registers

| Address | Name | Bit Pos. | | | | | | | | |
|-------------|-------------|----------|----------------|--|--|-------------|-------------|---|----|---|
| 0x00 ... | Reserved | | | | | | | | | |
| 0x1FE3 | | | | | | | | | | |
| 0x1FE4 | STATUS_SHAD | 7:0 | | | | T \bar{O} | P \bar{D} | Z | DC | C |
| 0x1FE5 | WREG_SHAD | 7:0 | WREG[7:0] | | | | | | | |
| 0x1FE6 | BSR_SHAD | 7:0 | BSR[5:0] | | | | | | | |
| 0x1FE7 | PCLATH_SHAD | 7:0 | PCLATH[6:0] | | | | | | | |
| 0x1FE8 | FSR0_SHAD | 7:0 | FSR0_SHAD[7:0] | | | | | | | |
| | | 15:8 | | | | | | | | |
| 0x1FEA | FSR1_SHAD | 7:0 | FSR1_SHAD[7:0] | | | | | | | |
| | | 15:8 | | | | | | | | |

5.10 Register Definitions: Shadow Registers

5.10.1 STATUS_SHAD

Name: STATUS_SHAD
Address: 0x1FE4

Shadow of Status Register

| | | | | | | | | |
|--------|---|---|---|----|----|-----|-----|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | | | TO | PD | Z | DC | C |
| Access | | | | RO | RO | R/W | R/W | R/W |
| Reset | | | | x | x | x | x | x |

Bit 4 – TO Time-Out bit

Reset States: POR/BOR = x
All Other Resets = u

| Value | Description |
|-------|--|
| 1 | Set at power-up or by execution of CLRWDT or SLEEP instruction |
| 0 | A WDT Time-out occurred |

Bit 3 – PD Power-Down bit

Reset States: POR/BOR = x
All Other Resets = u

| Value | Description |
|-------|---|
| 1 | Set at power-up or by execution of CLRWDT instruction |
| 0 | Cleared by execution of the SLEEP instruction |

Bit 2 – Z Zero bit

Reset States: POR/BOR = x
All Other Resets = u

| Value | Description |
|-------|--|
| 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/Borrow bit⁽¹⁾

ADDWF, ADDLW, SUBLW, SUBWF instructions
Reset States: POR/BOR = x
All Other Resets = u

| Value | Description |
|-------|---|
| 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/Borrow bit⁽¹⁾

ADDWF, ADDLW, SUBLW, SUBWF instructions
Reset States: POR/BOR = x
All Other Resets = u

| Value | Description |
|-------|---|
| 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:

- For Borrow, the polarity is reversed. A subtraction is executed by adding the two's complement of the second operand. For Rotate (RRCF, RLCF) instructions, this bit is loaded with either the high or low-order bit of the Source register.

5.10.2 WREG_SHAD

Name: WREG_SHAD
Address: 0x1FE5

Shadow of Working Data Register

| | | | | | | | | |
|--------|-----------|-----|-----|-----|-----|-----|-----|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | WREG[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | x | x | x | x | x | x | x | x |

Bits 7:0 – WREG[7:0]

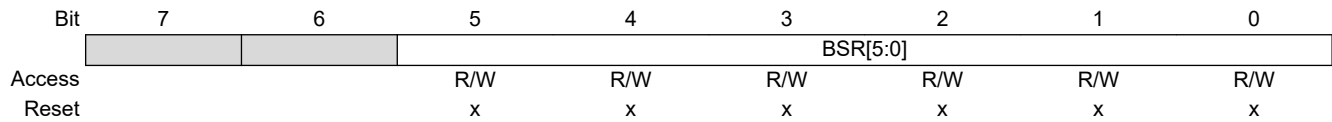
Reset States: POR/BOR = xxxxxxxx
 All Other Resets = uuuuuuuu

5.10.3 BSR_SHAD

Name: BSR_SHAD
Address: 0x1FE6

Shadow of Bank Select Register

The BSR indicates the data memory bank by writing the bank number into the register. All data memory can be accessed directly via instructions, or indirectly via FSRs.



Bits 5:0 – BSR[5:0]

Six Most Significant bits of the data memory address

Reset States: POR/BOR = xxxxxx

All Other Resets = uuuuuu

5.10.4 PCLATH_SHAD

Name: PCLATH_SHAD

Address: 0x1FE7

Shadow of Program Counter Latches.

Write Buffer for the upper 7 bits of the Program Counter

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|-------------|-----|-----|-----|-----|-----|-----|-----|
| | PCLATH[6:0] | | | | | | | |
| Access | | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | x | x | x | x | x | x | x |

Bits 6:0 – PCLATH[6:0] High PC Latch Register

Holding register for Program Counter bits [6:0]

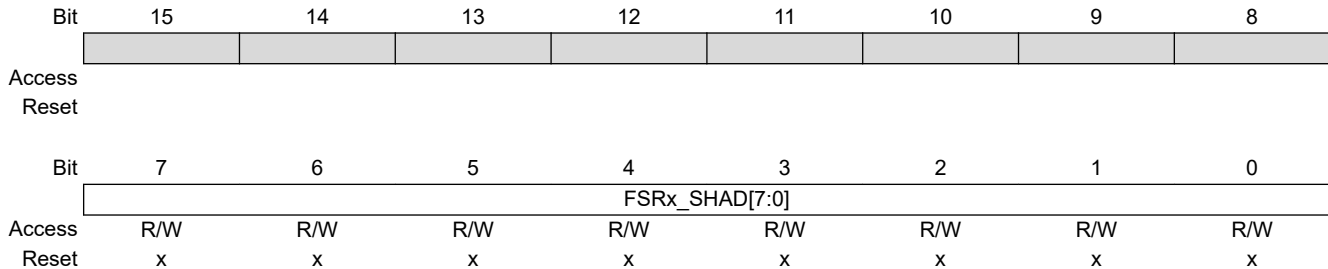
Reset States: POR/BOR = xxxxxxx

All Other Resets = uuuuuuu

5.10.5 FSR_SHAD

Name: FSRx_SHAD
Address: 0x1FE8,0x1FEA

Shadow of Indirect Address Register. The FSRx_SHAD value is the address of the data to which the INDFx register points.



Bits 7:0 – FSRx_SHAD[7:0]

Address of INDFx data

Reset States: POR/BOR = xxxxxxxxxxxxxxxxx

All Other Resets = uuuuuuuuuuuuuuuuu

Note: The individual bytes in this multi-byte register can be accessed with the following register names:

- FSRx_SHADH: Accesses the high byte FSRx_SHAD[15:8]
- FSRx_SHADL: Accesses the low byte FSRx_SHAD[7:0]

5.11 Device Configuration Information

The Device Configuration Information (DCI) is a dedicated region in the program Flash memory mapped from 8200h to 821Fh. The data stored in the DCI memory is hard-coded into the device during manufacturing. Refer to the table below for the complete DCI table address and description. The DCI holds information about the device, which is useful for programming and bootloader applications. These locations are read-only and cannot be erased or modified.

Table 5-5. Device Configuration Information for PIC16(L)F18426/46 Devices

| ADDRESS | Name | DESCRIPTION | PIC16(L)F18426/46 | UNITS |
|---------|-------|-------------------------|-------------------|---------|
| 8200h | ERSIZ | Erase Row Size | 32 | Words |
| 8201h | WLSIZ | Number of Write Latches | 32 | Latches |
| 8202h | URSIZ | Number of User Rows | 512 | Rows |
| 8203h | EESIZ | EE Data Memory Size | 256 | Bytes |
| 8204h | PCNT | Pin Count | 14, 16, 20 | Pins |

5.12 Device Information Area

The Device Information Area (DIA) is a dedicated region in the program memory space; it is a new feature in the PIC16(L)F184XX family of devices. The DIA contains the calibration data for the internal temperature indicator module, stores the Microchip Unique Identifier words, and the Fixed Voltage Reference voltage readings measured in mV. The complete DIA table is shown below, followed by a description of each region and its functionality. The data is mapped from 8100h to 811Fh in the PIC16(L)F184XX family. These locations are read-only and cannot be erased or modified. The data is programmed into the device during manufacturing.

Table 5-6. Device Information Area

| Address Range | Name of Region | Standard Device Information |
|-------------------------------------|-----------------------|--|
| 8100h-8108h | MUI0 | Microchip Unique Identifier (9 Words) |
| | MUI1 | |
| | MUI2 | |
| | MUI3 | |
| | MUI4 | |
| | MUI5 | |
| | MUI6 | |
| | MUI7 | |
| | MUI8 | |
| 8109h | MUI9 | 1 Word Reserved |
| 810Ah-8111h | EUI0 | Unassigned (8 Words) |
| | EUI1 | |
| | EUI2 | |
| | EUI3 | |
| | EUI4 | |
| | EUI5 | |
| | EUI6 | |
| | EUI7 | |
| 8112h | TSLR1 | Unassigned (1 Word) |
| 8113h | TSLR2 | Temperature indicator ADC reading at 90°C (low range setting) |
| 8114h | TSLR3 | Unassigned (1 Word) |
| 8115h | TSHR1 | Unassigned (1 Word) |
| 8116h | TSHR2 | Temperature indicator ADC reading at 90°C (high range setting) |
| 8117h | TSHR3 | Unassigned (1 Word) |
| 8118h | FVRA1X | ADC FVR1 Output voltage for 1x setting (in mV) |
| 8119h | FVRA2X | ADC FVR1 Output Voltage for 2x setting (in mV) |
| 811Ah | FVRA4X ⁽¹⁾ | ADC FVR1 Output Voltage for 4x setting (in mV) |
| 811Bh | FVRC1X | Comparator FVR2 Output Voltage for 1x setting (in mV) |
| 811Ch | FVRC2X | Comparator FVR2 Output Voltage for 2x setting (in mV) |
| 811Dh | FVRC4X ⁽¹⁾ | Comparator FVR2 Output Voltage for 4x setting (in mV) |
| 811Eh-811Fh | | Unassigned (2 Words) |
| Note: | | |
| 1. Value not present on LF devices. | | |

5.12.1 Microchip Unique Identifier (MUI)

The PIC16(L)F184XX devices are individually encoded during final manufacturing with a Microchip Unique Identifier (MUI). The MUI cannot be erased by a Bulk Erase command or any other user-accessible means. This feature allows for manufacturing traceability of Microchip Technology devices in applications where this is required. It may also be used by the application manufacturer for a number of functions that require unverified unique identification, such as:

- Tracking the device
- Unique serial number

The MUI consists of nine program words and one reserved program word. When taken together, these fields form a unique identifier. The MUI is stored in read-only locations, located between 8100h to 8109h in the DIA space. The above table lists the addresses of the identifier words.



Important: For applications that require verified unique identification, contact your Microchip Technology sales office to create a serialized quick turn programming option.

5.12.2 External Unique Identifier (EUI)

The EUI data is stored at locations 810Ah to 8111h in the program memory region. This region is an optional space for placing application specific information. The data is coded per customer requirements during manufacturing. The EUI cannot be erased by a Bulk Erase command.



Important: Data is stored in this address range on receiving a request from the customer. The customer may contact the local sales representative or Field Applications Engineer, and provide them the unique identifier information that is required to be stored in this region.

5.12.3 Analog-to-Digital Conversion Data of the Temperature Sensor

The purpose of the temperature indicator module is to provide a temperature-dependent voltage that can be measured by an analog module. The “*Temperature Indicator Module*” chapter explains the operation of the Temperature Indicator module and defines terms such as the low range and high range settings of the sensor. The DIA table contains the internal ADC measurement values of the temperature sensor for low and high range at fixed points of reference. The values are measured during test and are unique to each device. The right-justified ADC readings are stored in the DIA memory region. The calibration data can be used to plot the approximate sensor output voltage, V_{TSENSE} vs. Temperature curve.

- TSLR: Address 8112h to 8114h store the measurements for the low range setting of the temperature sensor at $V_{DD} = 3V$.
- TSHR: Address 8115h to 8117h store the measurements for the high range setting of the temperature sensor at $V_{DD} = 3V$.

The stored measurements are made by the device ADC using the internal $V_{REF} = 2.048V$.

5.12.4 Fixed Voltage Reference Data

The Fixed Voltage Reference, or FVR, is a stable voltage reference, independent of V_{DD} , with 1.024V, 2.048V or 4.096V selectable output levels. The output of the FVR can be configured to supply a reference voltage to the following:

- ADC input channel
- ADC positive reference
- Comparator positive input
- Digital-to-Analog Converter

For more information on the FVR, refer to the “*Fixed Voltage Reference (FVR)*” chapter.

The DIA stores measured FVR voltages for this device in mV for the different buffer settings of 1x, 2x or 4x at program memory locations 8118h to 811Dh.

- FVRA1X stores the value of ADC FVR1 Output Voltage for 1x setting (in mV)
- FVRA2X stores the value of ADC FVR1 Output Voltage for 2x setting (in mV)
- FVRA4X stores the value of ADC FVR1 Output Voltage for 4x setting (in mV)
- FVRC1X stores the value of Comparator FVR2 Output Voltage for 1x setting (in mV)
- FVRC2X stores the value of Comparator FVR2 Output Voltage for 2x setting (in mV)
- FVRC4X stores the value of Comparator FVR2 Output Voltage for 4x setting (in mV)

Related Links

[30. FVR - Fixed Voltage Reference](#)

6. NVM - Nonvolatile Memory Control

Nonvolatile Memory (NVM) is separated into two types: Program Flash Memory (PFM) and Data EEPROM Memory. NVM is accessible by using both the FSR and INDF registers, or through the NVMREG register interface.

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

NVM can be protected in two ways, by either code protection or write protection.

Code protection (\overline{CP} bit in the CONFIG5) disables access, reading and writing to both PFM and Data EEPROM Memory via external device programmers. Code protection does not affect the self-write and erase functionality. Code protection can only be reset by a device programmer performing a Bulk Erase to the device, clearing all nonvolatile memory, Configuration bits and User IDs.

Write protection prohibits self-write and erase to a portion or all of the NVM, as defined by the \overline{WRSAF} , \overline{WRD} , \overline{WRTC} , \overline{WRTB} , and \overline{WRTAPP} bits of CONFIG4. Write protection does not affect a device programmer's ability to read, write, or erase the device.

Related Links

- [4.7.4 CONFIG4](#)
- [4.7.5 CONFIG5](#)

6.1 Program Flash Memory

Program Flash memory consists of an array of 14-bit words as user memory, with additional words for user ID information, Configuration words, and interrupt vectors. Program memory provides storage locations for:

- User program instructions
- User defined data

Program memory data can be read and/or written to through:

- CPU instruction fetch (read-only)
- FSR/INDF indirect access (read-only)
- NVMREG access
- In-Circuit Serial Programming™ (ICSP™)

Read operations return a single word of memory. When write and erase operations are done on a row basis, the row size is defined. Program memory will erase to a logic '1' and program to a logic '0'.

It is important to understand the program memory structure for erase and programming operations. Program memory is arranged in rows. A row consists of 32 14-bit program memory words. A row is the minimum size that can be erased by user software.

All or a portion of a row can be programmed. Data to be written into the program memory row is written to 14-bit wide data write latches. These latches are not directly accessible, but may be loaded via sequential writes to the NVMDATH:NVMDATL register pair.



Important: To modify only a portion of a previously programmed row, the contents of the entire row must be read. Then, the new data and retained data can be written into the write latches to reprogram the row of program memory. However, any unprogrammed locations can be written without first erasing the row. In this case, it is not necessary to save and rewrite the other previously programmed locations.

Related Links

- [6.3 FSR and INDF Access](#)
- [6.4 NVMREG Access](#)

6.1.1 Program Memory Voltages

The program memory is readable and writable during normal operation over the full V_{DD} range.

6.1.1.1 Programming Externally

The program memory cell and control logic support write and Bulk Erase operations down to the minimum device operating voltage. Special BOR operation is enabled during Bulk Erase.

Related Links

[10.2.4 BOR is Always OFF](#)

6.1.1.2 Self-Programming

The program memory cell and control logic will support write and row erase operations across the entire V_{DD} range. Bulk Erase is not available when self-programming.

6.2 Data EEPROM

Data EEPROM consists of 256 bytes of user data memory. The EEPROM provides storage locations for 8-bit user defined data.

EEPROM can be read and/or written through:

- NVMREG access
- External device programmer

Additionally, read-only access to the EEPROM contents are also available through indirect addressing via FSR/INDF registers.

Unlike Program Flash Memory, which must be written to by row, EEPROM can be written to byte by byte.

Related Links

[6.3 FSR and INDF Access](#)

[6.4 NVMREG Access](#)

6.3 FSR and INDF Access

The FSR and INDF registers allow indirect access to the program memory or EEPROM.

Related Links

[5.8.5 FSR0](#)

6.3.1 FSR Read

With the intended address loaded into an FSR register a `MOVIW` instruction or read of INDF will read data from the program memory or EEPROM.

Reading from NVM requires one instruction cycle. The CPU operation is suspended during the read, and resumes immediately after. Read operations return a single byte of memory.

6.3.2 FSR Write

Writing/erasing the NVM through the FSR registers (ex. `MOVWI` instruction) is not supported in the PIC16(L)F184XX devices.

6.4 NVMREG Access

The NVMREG interface allows read/write access to all the locations accessible by FSRs, and also read/write access to the User ID locations and EEPROM, and read-only access to the device identification, revision, and configuration data.

Writing or erasing of NVM via the NVMREG interface is prevented when the device is write-protected.

6.4.1 NVMREG Read Operation

To read a NVM location using the NVMREG interface, the user must:

1. Clear the NVMREGS bit of the NVMCON1 register if the user intends to access program memory locations, or set NVMREGS if the user intends to access User ID, EEPROM, or configuration locations.
2. Write the desired address into the NVMADRH:NVMADRL register pair.
3. Set the RD bit of the NVMCON1 register to initiate the read.

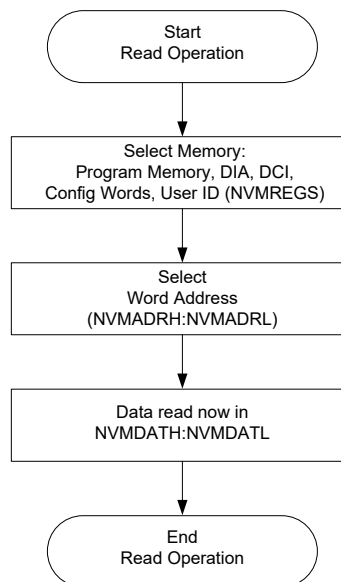
Once the read control bit is set, the CPU operation is suspended during the read, and resumes immediately after. The data is available in the very next cycle, in the NVMDATH:NVMDATL register pair; therefore, it can be read as two bytes in the following instructions.

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

Upon completion, the RD bit is cleared by hardware.

Figure 6-1. Flash Program Memory Read Flowchart

Rev. 10-000346E
5/17/2017



Example 6-1. Program Memory read

```

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

      BANKSEL      NVMADRL      ; Select Bank for NVMCON registers
      MOVLW        PROG_ADDR_LO ;
      MOVWF        NVMADRL      ; Store LSB of address
      MOVLW        PROG_ADDR_HI ;
  
```

```

MOVWF    NVMADRH        ; Store MSB of address

BCF      NVMCON1,NVMREGS ; Do not select Configuration Space
BSF      NVMCON1,RD     ; Initiate read

MOVWF    NVMADRH        ; Store MSB of address

MOVWF    NVMADRH        ; Store MSB of address

MOV      NVMADRH,W      ; Get LSB of word
MOVWF    PROG_DATA_LO   ; Store in user location
MOV      NVMADRH,W      ; Get MSB of word
MOVWF    PROG_DATA_HI   ; Store in user location

```

Related Links

[6.6.1 NVMADR](#)

[6.6.2 NVMDAT](#)

[6.6.3 NVMCON1](#)

6.4.2 NVM Unlock Sequence

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

- Program Flash Memory Row Erase
- Load of Program Flash Memory write latches
- Write of Program Flash Memory write latches to program memory
- Write of Program Flash Memory write latches to User IDs
- Write to EEPROM

The unlock sequence consists of the following steps and must be completed in order:

- Write 0x55 to NVMCON2
- Write 0xAA to NVMCON2
- Set the WR bit of NVMCON1

Once the WR bit is set, the processor will stall internal operations until the operation is complete and then resume with the next instruction.

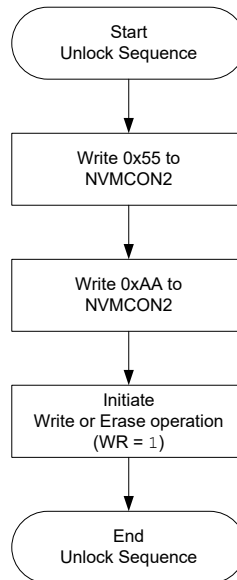


Important: The two NOP instructions after setting the WR bit, that were required in previous devices, are not required for PIC16(L)F184XX devices. See [Figure 6-2](#) for an example of the Unlock sequence.

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

Figure 6-2. NVM Unlock Sequence Flowchart

Rev. 10-00047B
8/24/2015



Example 6-2. NVM Unlock Sequence

```

BCF      INTCON, GIE      ; Recommended so sequence is not interrupted
BANKSEL  NVMCON1         ;
BSF      NVMCON1, WREN   ; Enable write/erase
MOVLW   55h              ; Load 55h
MOVWF   NVMCON2         ; Step 1: Load 55h into NVMCON2
MOVLW   AAh              ; Step 2: Load W with AAh
MOVWF   NVMCON2         ; Step 3: Load AAh into NVMCON2
BSF      NVMCON1, WR     ; Step 4: Set WR bit to begin write/erase
BSF      INTCON, GIE     ; Re-enable interrupts
  
```

Note:

1. Sequence begins when NVMCON2 is written; steps 1-4 must occur in the cycle-accurate order shown.
2. Opcodes shown are illustrative; any instruction that has the indicated effect may be used.

6.4.3 NVMREG Write to EEPROM

Writing to the EEPROM is accomplished by the following steps:

1. Set the NVMREGS and WREN bits of the NVMCON1 register.
2. Write the desired address (address 0x7000) into the NVMADRH:NVMADRL register pair.
3. Perform the unlock sequence as described in the [6.4.2 NVM Unlock Sequence](#) section.

A single EEPROM byte is written through NVMDATA register. The operation includes an implicit erase cycle for that byte (it is not necessary to set the FREE bit in NVMCON1 register), and will require multiple instruction cycles to finish. CPU execution continues in parallel and, when complete, WR bit in NVMCON1 register is cleared by hardware, NVMIF flag in PIR7 register is set, and an interrupt will occur if NVMIE in PIE7 is also set. Software must poll the WR bit to determine when writing is complete, or wait for the interrupt to occur. WREN will remain unchanged. Once the EEPROM write operation begins, clearing the WR bit will have no effect; the operation will run to completion.

Related Links

[6.4.2 NVM Unlock Sequence](#)

[6.4.4 NVMREG Erase of Program Memory](#)

6.4.4 NVMREG Erase of Program Memory

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

1. Clear the NVMREGS bit of the NVMCON1 register to erase program memory locations, or set the NVMREGS bit to erase User ID locations.
2. Write the desired address into the NVMADRH:NVMADRL register pair.
3. Set the FREE and WREN bits of the NVMCON1 register.
4. Perform the unlock sequence as described in the “*NVM Unlock Sequence*” section.

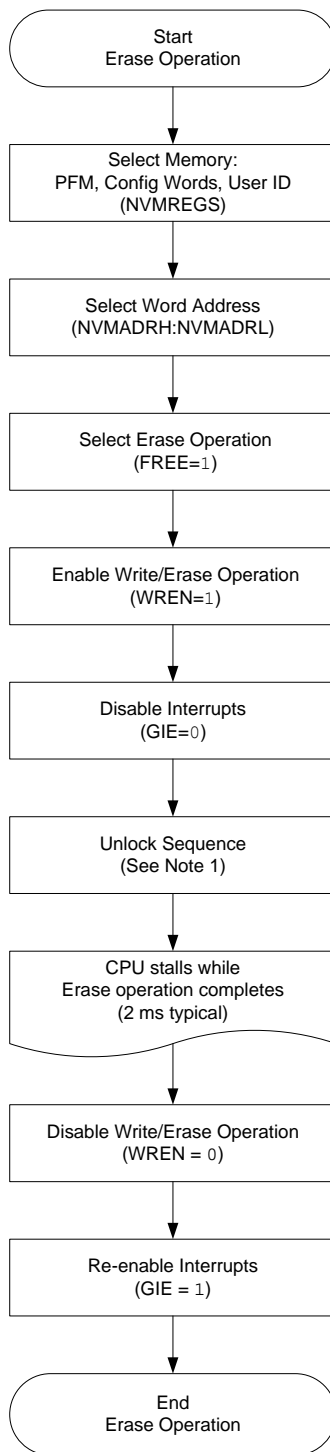
If the program memory address is write-protected, the WR bit will be cleared and the erase operation will not take place.

While erasing program memory, CPU operation is suspended, and resumes when the operation is complete. Upon completion, the NVMIF is set, and an interrupt will occur if the NVMIE bit is also set.

Write latch data is not affected by erase operations, and WREN will remain unchanged.

Figure 6-3. NVM Erase Flowchart

Rev. 10-000048B
8/24/2015



Note: See previous figure.

PIC16(L)F18426/46

NVM - Nonvolatile Memory Control

Example 6-3. Erasing One Row of Program Flash Memory

```

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

BANKSEL    NVMADRL
MOVWF     ADDRL,W
MOVWF     NVMADRL           ; Load lower 8 bits of erase address boundary
MOVWF     ADDRH,W
MOVWF     NVMADRH          ; Load upper 6 bits of erase address boundary
BCF       NVMCON1,NVMREGS ; Choose PFM memory area
BSF       NVMCON1,FREE     ; Specify an erase operation
BSF       NVMCON1,WREN     ; Enable writes
BCF       INTCON,GIE       ; Disable interrupts during unlock sequence

; -----REQUIRED UNLOCK SEQUENCE:-----

MOVLW     0x55             ; Load 0x55 to get ready for unlock sequence
MOVWF     NVMCON2          ; First step is to load 0x55 into NVMCON2
MOVLW     0xAA             ; Second step is to load 0xAA into W
MOVWF     NVMCON2          ; Third step is to load 0xAA into NVMCON2
BSF       NVMCON1,WR       ; Final step is to set WR bit

; -----

BSF       INTCON,GIE       ; Re-enable interrupts, erase is complete
BCF       NVMCON1,WREN     ; Disable writes

```

Table 6-1. NVM Organization and Access Information

| Master Values | | | | NVMREG Access | | | FSR Access | | | |
|-----------------|----------------------|-----------------------|-----------------------|-----------------------|-----------------------|--------------------|-------------|------------------------|--------|-----------|
| Memory Function | Memory Type | Program Counter (PC) | ICSP™ Address | NVMREGS bit (NVMCON1) | NVMADR[14:0] | Allowed Operations | FSR Address | FSR Programming Access | | |
| RESET VECTOR | Program Flash Memory | 0x0000 | 0x0000 | 0 | 0x0000 | READ/WRITE | 0x8000 | READ ONLY | | |
| USER MEMORY | | 0x0001 | 0x0001 | 0 | 0x0001 | | 0x8001 | | | |
| | | 0x0003 | 0x0003 | | 0x8003 | | | | | |
| INT VECTOR | | 0x0004 | 0x0004 | 0 | 0x0004 | | 0x8004 | | | |
| USER MEMORY | | 0x0005 | 0x0005 | 0 | 0x0005 | | 0x8005 | | | |
| | | 0x7FFF ⁽¹⁾ | 0x7FFF ⁽¹⁾ | | 0x7FFF ⁽¹⁾ | | 0xFFFF | | | |
| USER ID | Program Flash Memory | NO PC ACCESS | 0x8000 | 1 | 0x0000 | READ/WRITE | NO ACCESS | | | |
| | | | 0x8003 | | 0x0003 | | | | | |
| Reserved | — | | — | — | 0x0004 | — | | | | |
| REV ID | HC | | 0x8005 | 1 | 0x0005 | READ | | | | |
| DEVICE ID | | | 0x8006 | 1 | 0x0006 | | | | | |
| CONFIG1 | FUSE | | 0x8007 | 1 | 0x0007 | READ/WRITE | | | | |
| CONFIG2 | | | 0x8008 | 1 | 0x0008 | | | | | |
| CONFIG3 | | | 0x8009 | 1 | 0x0009 | | | | | |
| CONFIG4 | | | 0x800A | 1 | 0x000A | | | | | |
| CONFIG5 | | | 0x800B | 1 | 0x000B | | | | | |
| DIA and DCI | PFM | | 0x8100 | 1 | 0x0100 | READ | | | | |
| | | | 0x82FF | 1 | 0x02FF | | | | | |
| USER MEMORY | EEPROM | | 0xF000 | 1 | 0x7000 | READ/WRITE | | | 0x7000 | READ ONLY |
| | | | 0xF0FF | | 0x70FF | | | | | |

Note:

1. The maximum Program Flash Memory address for the PIC16(L)F184XX family is 0x7FFF.

Related Links

[5. Memory Organization](#)

6.4.5 NVMREG Write to Program Memory

Program memory is programmed using the following steps:

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

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

Program memory can be written one or more words at a time. The maximum number of words written at one time is equal to the number of write latches. See [Figure 6-4](#) for more details.

The write latches are aligned to the Flash row address boundary defined by the upper ten bits of NVMADRH:NVMADRL, (NVMADRH[6:0]:NVMADRL[7:5]) with the lower five bits of NVMADRL, (NVMADRL[4:0]) determining the write latch being loaded. Write operations do not cross these boundaries. At the completion of a program memory write operation, the data in the write latches is reset to contain 0x7FFF.

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



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

1. Set the WREN bit of the NVMCON1 register.
2. Clear the NVMREGS bit of the NVMCON1 register.
3. Set the LWLO bit of the NVMCON1 register. When the LWLO bit of the NVMCON1 register is '1', the write sequence will only load the write latches and will not initiate the write to Flash program memory.
4. Load the NVMADRH:NVMADRL register pair with the address of the location to be written.
5. Load the NVMDATH:NVMDATL register pair with the program memory data to be written.
6. Execute the unlock sequence. The write latch is now loaded.
7. Increment the NVMADRH:NVMADRL register pair to point to the next location.
8. Repeat steps 5 through 7 until all but the last write latch has been loaded.
9. Clear the LWLO bit of the NVMCON1 register. When the LWLO bit of the NVMCON1 register is '0', the write sequence will initiate the write to Flash program memory.
10. Load the NVMDATH:NVMDATL register pair with the program memory data to be written.
11. Execute the unlock sequence. The entire program memory latch content is now written to Flash program memory.



Important: The program memory write latches are reset to the blank state (0x7FFF) at the completion of every write or erase operation. As a result, it is not necessary to load all the program memory write latches. Unloaded latches will remain in the blank state.

PIC16(L)F18426/46

NVM - Nonvolatile Memory Control

An example of the complete write sequence is shown in [Writing to Program Flash Memory](#). The initial address is loaded into the NVMADRH:NVMADRL register pair; the data is loaded using indirect addressing.

Figure 6-4. NVMREG Writes to Program Flash Memory With 32 Write Latches

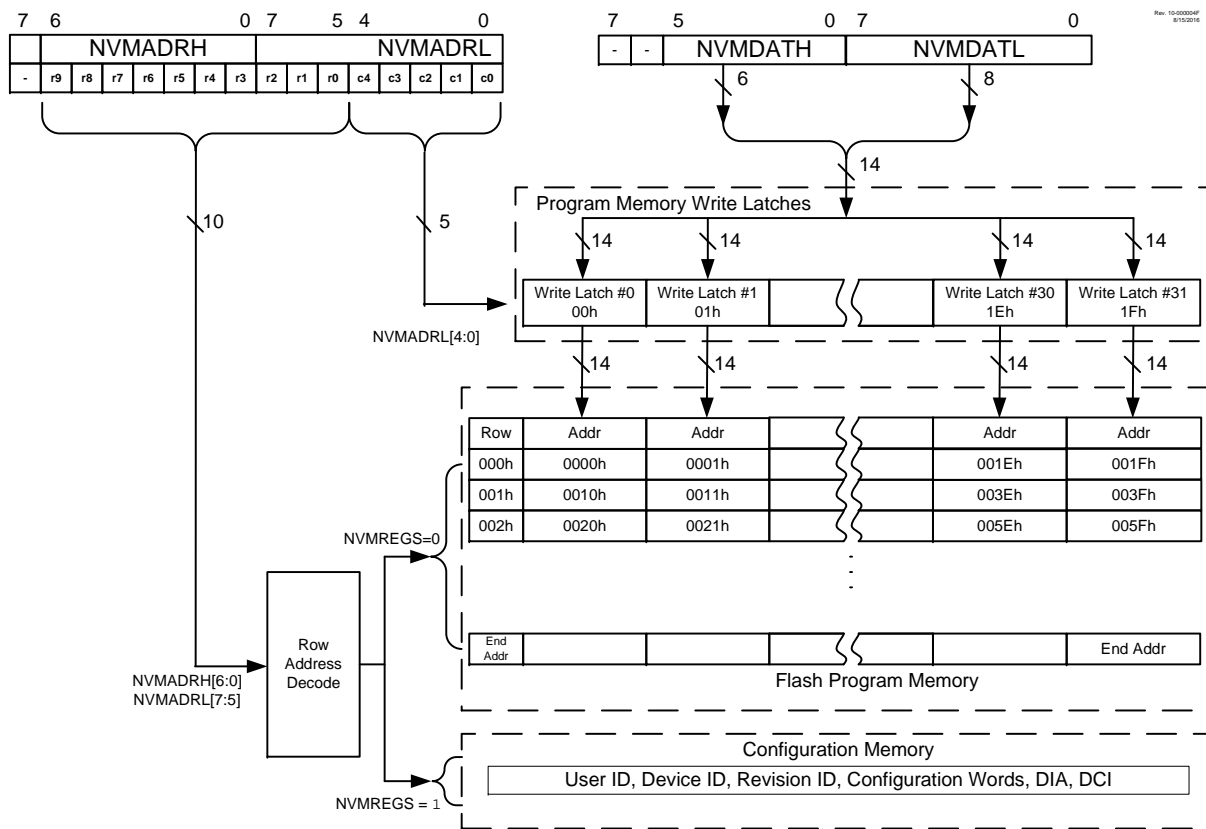
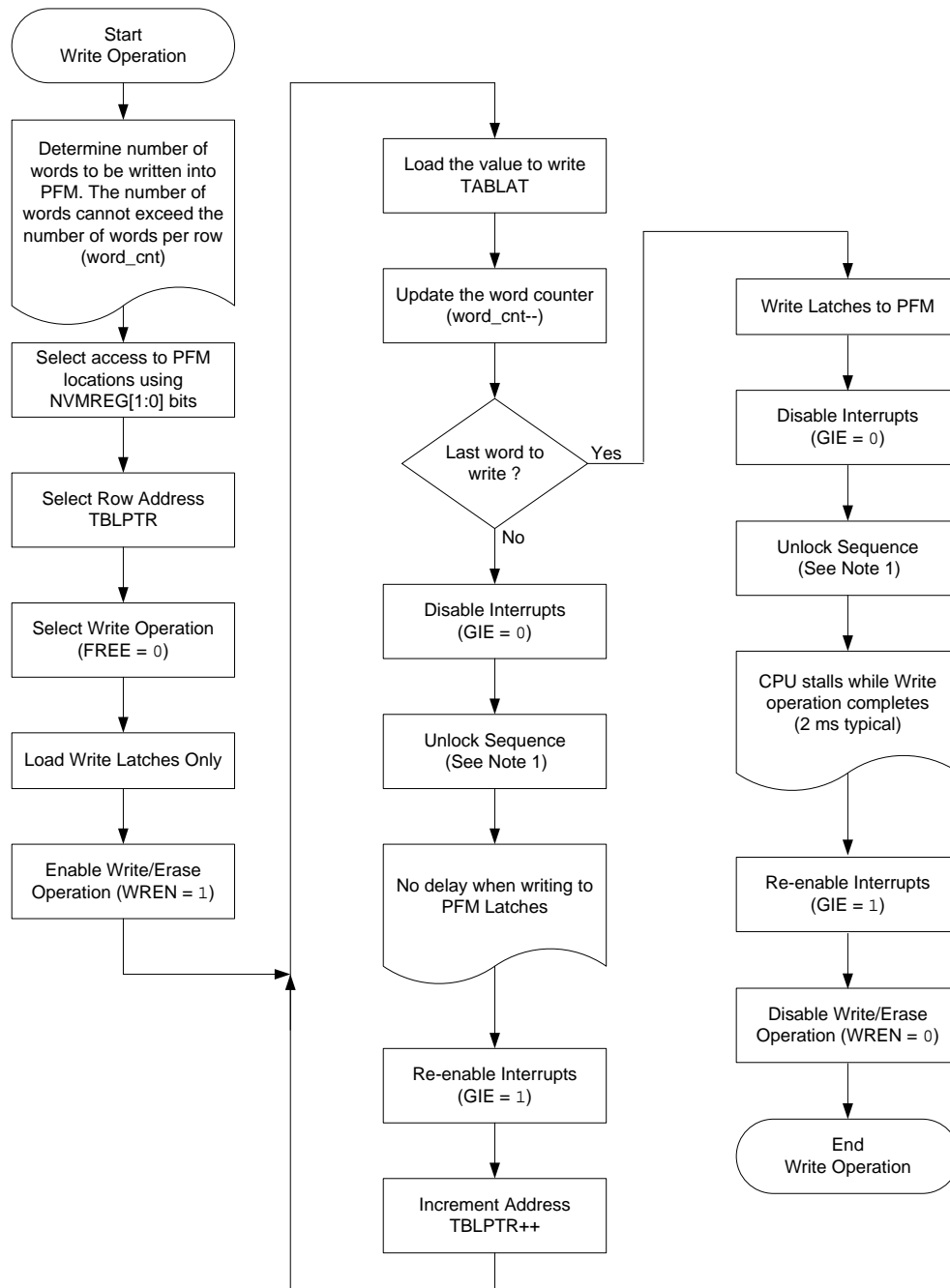


Figure 6-5. Program Flash Memory Flowchart

Rev. 10-000049C
8/24/2015



Note:

1. See [Figure 6-2](#) for an example of the Unlock sequence.

Example 6-4. Writing to Program Flash Memory

```

; This write routine assumes the following:
; 1. 64 bytes of data are loaded, starting at the address in DATA_ADDR
; 2. Each word of data to be written is made up of two adjacent bytes in
    
```

```

DATA_ADDR,
; stored in little endian format
; 3. A valid starting address (the least significant bits = 00000) is
loaded in ADDRH:ADDRL
; 4. ADDRH and ADDRL are located in common RAM (locations 0x70 - 0x7F)
; 5. NVM interrupts are not taken into account

BANKSEL    NVMADRH
MOVWF     ADDRH,W
MOVWF     NVMADRH           ; Load initial address
MOVWF     ADDRL,W
MOVWF     NVMADRL
MOVLW    LOW DATA_ADDR   ; Load initial data address
MOVWF     FSR0L
MOVLW    HIGH DATA_ADDR
MOVWF     FSR0H
BCF      NVMCON1,NVMREGS  ; Set Program Flash Memory as write location
BSF      NVMCON1,WREN     ; Enable writes
BSF      NVMCON1,LWLO    ; Load only write latches

LOOP

    MOVIW    FSR0++
    MOVWF    NVMDATL       ; Load first data byte
    MOVIW    FSR0++
    MOVWF    NVMDATH       ; Load second data byte

    MOVF     NVMADRL,W
    XORLW   0x1F           ; Check if lower bits of address are 00000
    ANDLW   0x1F           ; and if on last of 32 addresses
    BTFSC   STATUS,Z       ; Last of 32 words?
    GOTO    START_WRITE    ; If so, go write latches into memory

    CALL    UNLOCK_SEQ     ; If not, go load latch
    INCF   NVMADRL,F       ; Increment address
    GOTO    LOOP

START_WRITE

    BCF     NVMCON1,LWLO   ; Latch writes complete, now write memory
    CALL    UNLOCK_SEQ     ; Perform required unlock sequence
    BCF     NVMCON1,WREN   ; Disable writes

UNLOCK_SEQ

    MOVLW   55h
    BCF     INTCON,GIE     ; Disable interrupts
    MOVWF   NVMCON2        ; Begin unlock sequence
    MOVLW   AAh
    MOVWF   NVMCON2
    BSF     NVMCON1,WR
    BSF     INTCON,GIE     ; Unlock sequence complete, re-enable
interrupts
return

```

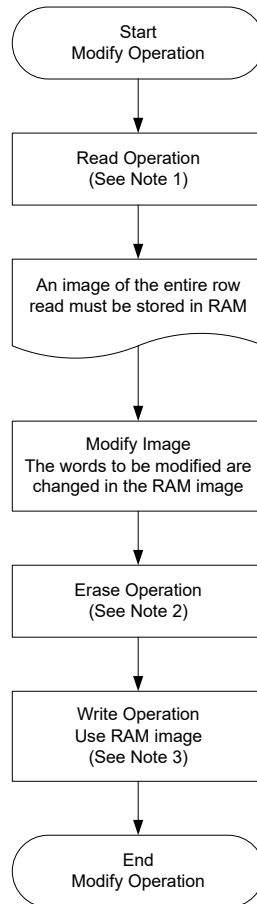
6.4.6 Modifying Flash Program Memory

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

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

Figure 6-6. Flash Program Memory Modify Flowchart

Rev. 10-000050B
8/21/2015



Note:

1. See [Flash Program Memory Read Flowchart](#).
2. See [NVM Erase Flowchart](#).
3. See [Program Flash Memory Flowchart](#).

6.4.7 NVMREG Access to Device Information Area, Device Configuration Area, User ID, Device ID, EEPROM, and Configuration Words

NVMREGS can be used to access the following memory regions:

- Device Information Area (DIA)
- Device Configuration Information (DCI)
- User ID region
- Device ID and Revision ID
- Configuration Words
- EEPROM

The value of NVMREGS is set to '1' in the NVMCON1 register to access these regions. The memory regions listed above would be pointed to by PC[15] = 1, but not all addresses reference valid data. Different access may exist for reads and writes. Refer to the table below. When read access is initiated on an address outside the parameters listed in the following table, the NVMDATH: NVMDATL register pair is cleared, reading back '0's.

Table 6-2. NVMREG Access to Device Information Area, Device Configuration Area, User ID, Device ID, EEPROM, and Configuration Words (NVMREGS = 1)

| Address | Function | Read Access | Write Access |
|-----------------|-------------------------|-------------|--------------|
| 0x8000 - 0x8003 | User IDs | Yes | Yes |
| 0x8005 - 0x8006 | Device ID/Revision ID | Yes | No |
| 0x8007 - 0x800B | Configuration Words 1-5 | Yes | Yes |
| 0x8100 - 0x82FF | DIA and DCI | Yes | No |
| 0xF000 - 0xF0FF | EEPROM | Yes | Yes |

Example 6-5. Device ID Access

```

; This write routine assumes the following:
; 1. A full row of data are loaded, starting at the address in DATA_ADDR
; 2. Each word of data to be written is made up of two adjacent bytes in
DATA_ADDR,
; stored in little endian format
; 3. A valid starting address (the least significant bits = 00000) is
loaded in ADDRH:ADDRL
; 4. ADDRH and ADDRL are located in common RAM (locations 0x70 - 0x7F)
; 5. NVM interrupts are not taken into account

    BANKSEL    NVMADRH
    MOVF       ADDRH,W
    MOVWF     NVMADRH           ; Load initial address
    MOVF       ADDRL,W
    MOVWF     NVMADRL
    MOVLW     LOW DATA_ADDR   ; Load initial data address
    MOVWF     FSR0L
    MOVLW     HIGH DATA_ADDR
    MOVWF     FSR0H
    BCF       NVMCON1,NVMREGS  ; Set PFM as write location
    BSF       NVMCON1,WREN     ; Enable writes
    BSF       NVMCON1,LWLO     ; Load only write latches

LOOP

    MOVIW    FSR0++
    MOVWF    NVMDATL           ; Load first data byte
    MOVIW    FSR0++
    MOVWF    NVMDATH           ; Load second data byte
    CALL     UNLOCK_SEQ        ; If not, go load latch
    INCF     NVMADRL,F         ; Increment address
    MOVF     NVMADRL,W
    XORLW   0x1F
    ANDLW   0x1F
    BTFSC   STATUS,Z          ; Check if lower bits of address are 00000
                                ; and if on last of 32 addresses
    GOTO     START_WRITE      ; Last of 32 words?
                                ; If so, go write latches into memory

    GOTO     LOOP

START_WRITE

    BCF       NVMCON1,LWLO     ; Latch writes complete, now write memory
    CALL     UNLOCK_SEQ        ; Perform required unlock sequence
    BCF       NVMCON1,LWLO     ; Disable writes

UNLOCK_SEQ

    MOVLW    0x55
    BCF      INTCON,GIE        ; Disable interrupts
    MOVWF    NVMCON2           ; Begin unlock sequence
    MOVLW    0xAA
    MOVWF    NVMCON2
    BSF      NVMCON1,WR
    BSF      INTCON,GIE        ; Unlock sequence complete, re-enable

```

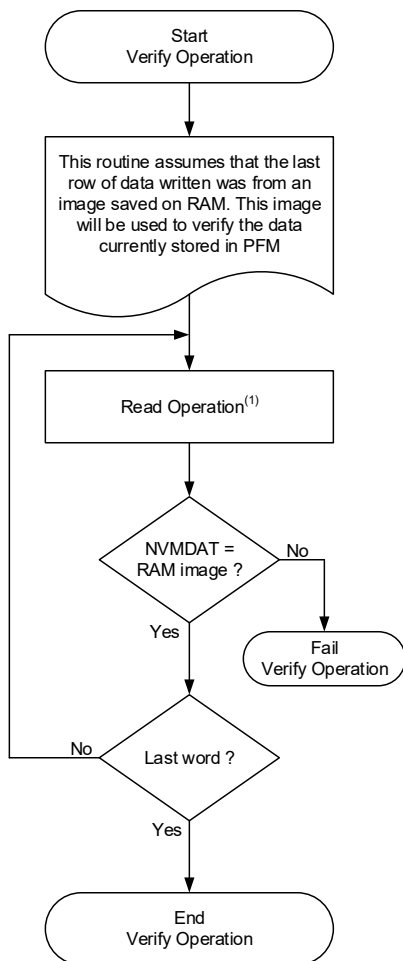
```
interrupts
RETURN
```

6.4.8 Write Verify

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

Figure 6-7. Flash Program Memory Verify Flowchart

Rev. 10-00051B
12/4/2015



Note:

1. See [Flash Program Memory Read Flowchart](#).

6.4.9 WRERR Bit

The WRERR bit in the NVMCON1 register can be used to determine if a write error occurred. WRERR will be set if one of the following conditions occurs:

- If WR is set while the NVMADRH:NMVADRL points to a write-protected address
- A Reset occurs while a self-write operation was in progress
- An unlock sequence was interrupted

The WRERR bit is normally set by hardware, but can be set by the user for test purposes. Once set, WRERR must be cleared in software.

Table 6-3. Actions for PFM When WR = 1

| Free | LWLO | Actions for PFM when WR = 1 | Comments |
|------|------|---|---|
| 1 | x | Erase the 32-word row of NVMADRH:NVMADRL location. | <ul style="list-style-type: none"> • If WP is enabled, WR is cleared and WRERR is set • All 32 words are erased • NVMDATH:NVMDATL is ignored |
| 0 | 1 | Copy NVMDATH:NVMDATL to the write latch corresponding to NVMADR LSBs. | <ul style="list-style-type: none"> • Write protection is ignored • No memory access occurs |
| 0 | 0 | Write the write-latch data to PFM row. | <ul style="list-style-type: none"> • If WP is enabled, WR is cleared and WRERR is set • Write latches are reset to 0x03FF • NVMDATH:NVMDATL is ignored |

Related Links

[6.4.4 NVMREG Erase of Program Memory](#)

6.5 Register Summary: NVM Control

| Address | Name | Bit Pos. | | | | | | | | |
|-------------|----------|----------|--------------|---------|------|------|-------|------|----|----|
| 0x00 ... | Reserved | | | | | | | | | |
| 0x0819 | | | | | | | | | | |
| 0x081A | NVMADR | 7:0 | NVMADR[7:0] | | | | | | | |
| | | 15:8 | NVMADR[14:8] | | | | | | | |
| 0x081C | NVMDAT | 7:0 | NVMDAT[7:0] | | | | | | | |
| | | 15:8 | NVMDAT[13:8] | | | | | | | |
| 0x081E | NVMCON1 | 7:0 | | NVMREGS | LWLO | FREE | WRERR | WREN | WR | RD |
| 0x081F | NVMCON2 | 7:0 | NVMCON2[7:0] | | | | | | | |

6.6 Register Definitions: Nonvolatile Memory

6.6.1 NVMADR

Name: NVMADR
Address: 0x81A

Nonvolatile Memory Address Register

| | | | | | | | | |
|--------|--------------|-----|-----|-----|-----|-----|-----|-----|
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| | NVMADR[14:8] | | | | | | | |
| Access | | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | | | | | | | |
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | NVMADR[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bits 14:0 – NVMADR[14:0] NVM Address Bits

Note: The individual bytes in this multi-byte register can be accessed with the following register names:

- NVMADRH: Accesses the high byte NVMADR[15:8]
- NVMADRL: Accesses the low byte NVMADR[7:0]

Note: Bit [15] is undefined while WR = 1

6.6.2 NVMDAT

Name: NVMDAT
Address: 0x81C

Nonvolatile Memory Data Register

| | | | | | | | | |
|--------|--------------|-----|-----|-----|-----|-----|-----|-----|
| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| | NVMDAT[13:8] | | | | | | | |
| Access | | | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | | x | x | x | x | x | x |
| | | | | | | | | |
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | NVMDAT[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | x | x | x | x | x | x | x | x |

Bits 13:0 – NVMDAT[13:0] NVM Data bits

Note: The individual bytes in this multi-byte register can be accessed with the following register names:

- NVMDATH: Accesses the high byte NVMDAT[13:8]
- NVMDATL: Accesses the low byte NVMDAT[7:0]

Reset States: POR/BOR = xxxxxxxxxxxxxxxx
 All Other Resets = uuuuuuuuuuuuuuuu

6.6.3 NVMCON1

Name: NVMCON1
Address: 0x81E

Nonvolatile Memory Control 1 Register

| | | | | | | | | |
|--------|---|---------|------|--------|--------|------|--------|--------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | NVMREGS | LWLO | FREE | WRERR | WREN | WR | RD |
| Access | | R/W | R/W | R/S/HC | R/W/HS | R/W | R/S/HC | R/S/HC |
| Reset | | 0 | 0 | 0 | x | 0 | 0 | 0 |

Bit 6 – NVMREGS NVM Region Selection bit

| Value | Description |
|-------|---|
| 1 | Access EEPROM, DIA, DCI, Configuration, User ID and Device ID Registers |
| 0 | Access Program Flash Memory |

Bit 5 – LWLO Load Write Latches Only bit

| Value | Condition | Description |
|-------|---------------|---|
| 1 | When FREE = 0 | The next WR command updates the write latch for this word within the row; no memory operation is initiated. |
| 0 | When FREE = 0 | The next WR command writes data or erases |
| - | Otherwise: | This bit is ignored. |

Bit 4 – FREE Program Flash Memory Erase Enable bit

| Value | Description |
|-------|---|
| 1 | Performs an erase operation with the next WR command; the 32-word pseudo-row containing the indicated address is erased (to all 1s) to prepare for writing. |
| 0 | The next WR command writes without erasing. |

Bit 3 – WRERR

Write-Reset Error Flag bit^(1,2,3)

Reset States: POR/BOR = x

All Other Resets = q

| Value | Description |
|-------|---|
| 1 | A write operation was interrupted by a Reset, interrupted unlock sequence, or WR was written to one while NVMADR points to a write-protected address. |
| 0 | All write operations have completed normally. |

Bit 2 – WREN Program/Erase Enable bit

| Value | Description |
|-------|---|
| 1 | Allows program/erase cycles |
| 0 | Inhibits programming/erasing of program Flash |

Bit 1 – WR Write Control bit^(4,5,6)

| Value | Condition | Description |
|-------|---|--|
| 1 | When NVMREG:NVMADR points to a Program Flash Memory location: | Initiates the operation indicated by table in “WRERR Bit” section. |
| 0 | When NVMREG:NVMADR points to a Program Flash Memory location: | NVM program/erase operation is complete and inactive. |
| 1 | When NVMREG:NVMADR points to a EEPROM location: | Initiates an erase/program cycle at the corresponding EEPROM location. |
| 0 | When NVMREG:NVMADR points to a EEPROM location: | NVM program/erase operation is complete and inactive. |

Bit 0 – RD Read Control bit⁽⁷⁾

PIC16(L)F18426/46

NVM - Nonvolatile Memory Control

| Value | Description |
|-------|---|
| 1 | Initiates a read at address = NVMADR1, and loads data to NVMDAT. Read takes one instruction cycle and the bit is cleared when the operation is complete. The bit can only be set (not cleared) in software. |
| 0 | NVM read operation is complete and inactive |

Note:

1. Bit is undefined while WR = 1 (during the EEPROM write operation it may be '0' or '1').
2. Bit must be cleared by software; hardware will not clear this bit.
3. Bit may be written to '1' by the user in order to implement test sequences.
4. This bit can only be set by following the sequence described in the "NVM Unlock Sequence" section.
5. Operations are self-timed and the WR bit is cleared by hardware when complete.
6. Once a write operation is initiated, setting this bit to zero will have no effect.
7. Reading from EEPROM loads only NVMDATL.

Related Links

[6.4.2 NVM Unlock Sequence](#)

[6.4.9 WRERR Bit](#)

6.6.4 NVMCON2

Name: NVMCON2
Address: 0x81F

Nonvolatile Memory Control 2 Register

| | | | | | | | | |
|--------|--------------|----|----|----|----|----|----|----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | NVMCON2[7:0] | | | | | | | |
| Access | WO | WO | WO | WO | WO | WO | WO | WO |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bits 7:0 – NVMCON2[7:0] Flash Memory Unlock Pattern bits

Note: To unlock writes, a 0x55 must be written first followed by an 0xAA before setting the WR bit of the NVMCON1 register. The value written to this register is used to unlock the writes.

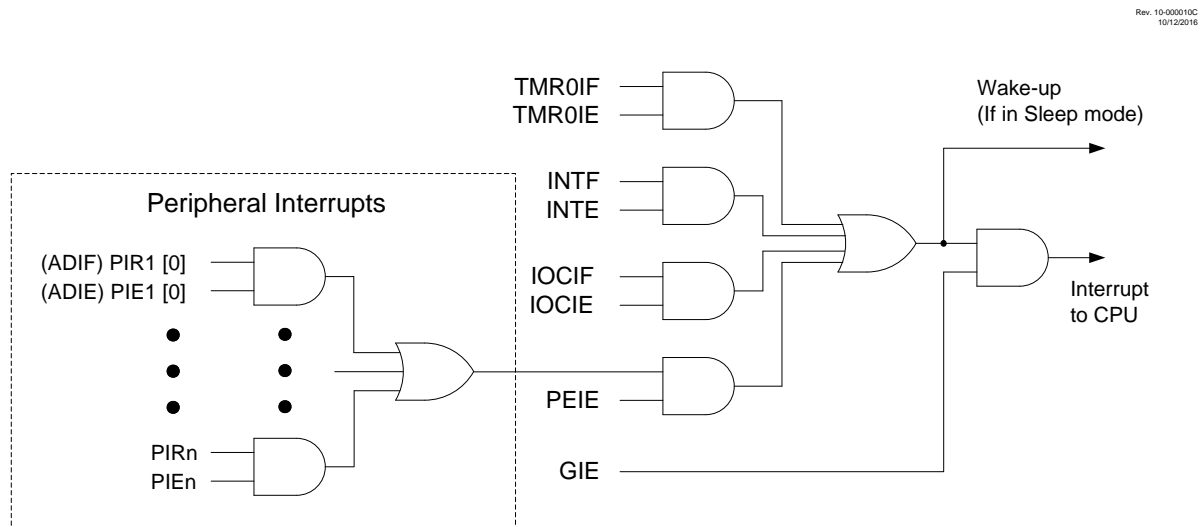
7. Interrupts

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

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

A block diagram of the interrupt logic is shown in [Figure 7-1](#).

Figure 7-1. Interrupt Logic



Rev. 10-000010C
10/12/2016

7.1 Operation

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

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

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

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

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

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

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

For additional information on a specific interrupts operation, refer to its peripheral chapter.



Important:

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

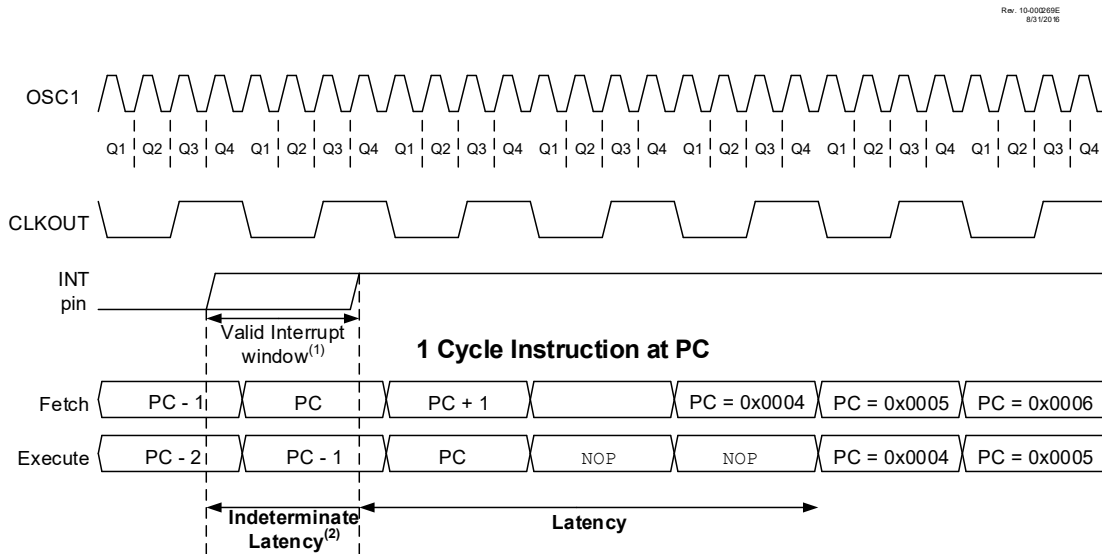
Related Links

[7.5 Automatic Context Saving](#)

7.2 Interrupt Latency

Interrupt latency is defined as the time from when the interrupt event occurs to the time code execution at the interrupt vector begins. The interrupt is sampled during Q1 of the instruction cycle. The actual interrupt latency then depends on the instruction that is executing at the time the interrupt is detected. See the following figures for more details.

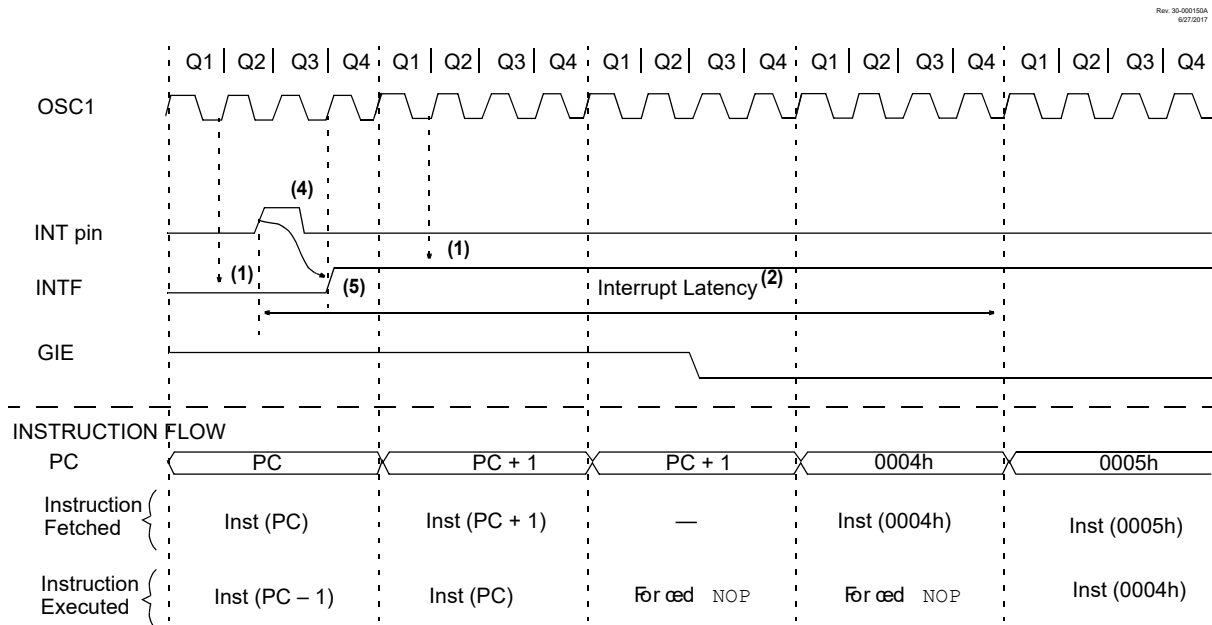
Figure 7-2. Interrupt Latency



Note:

1. An interrupt may occur at any time during the interrupt window.
2. Since an interrupt may occur any time during the interrupt window, the actual latency can vary.

Figure 7-3. INT Pin Interrupt Timing



Note:

1. INTF flag is sampled here (every Q1).
2. Asynchronous interrupt latency = 3-5 T_{CY} . Synchronous latency = 3-4 T_{CY} , where T_{CY} = instruction cycle time. Latency is the same whether Inst (PC) is a single cycle or a 2-cycle instruction.
3. For minimum width of INT pulse, refer to AC specifications in the “Electrical Specifications” section.
4. INTF may be set any time during the Q4-Q1 cycles.

7.3 Interrupts During Sleep

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

On waking from Sleep, if the GIE bit is also set, the processor will branch to the interrupt vector. Otherwise, the processor will continue executing instructions after the SLEEP instruction. The instruction directly after the SLEEP instruction will always be executed before branching to the ISR.

Related Links

[12. Power-Saving Operation Modes](#)

7.4 INT Pin

The INT pin can be used to generate an asynchronous edge-triggered interrupt. This interrupt is enabled by setting the INTE bit of the PIE0 register. The INTEDG bit of the INTCON register determines on which edge the interrupt will occur. When the INTEDG bit is set, the rising edge will cause the interrupt. When the INTEDG bit is clear, the falling edge will cause the interrupt. The INTF bit of the PIR0 register will be set when a valid edge appears on the INT pin. If the GIE and INTE bits are also set, the processor will redirect program execution to the interrupt vector.

7.5 Automatic Context Saving

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

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

Upon exiting the Interrupt Service Routine, these registers are automatically restored. Any modifications to these registers during the ISR will be lost. If modifications to any of these registers are desired, the corresponding shadow register should be modified and the value will be restored when exiting the ISR. The shadow registers are available in Bank 63 and are readable and writable. Depending on the user's application, other registers may also need to be saved.

Related Links

[5.10 Register Definitions: Shadow Registers](#)

7.6 Register Summary - Interrupt Control

| Address | Name | Bit Pos. | | | | | | | | |
|-------------|----------|----------|--------|--------|--------|--------|--------|-----------|-----------|---------|
| 0x00 ... | Reserved | | | | | | | | | |
| 0x070B | | | | | | | | | | |
| 0x070C | PIR0 | 7:0 | | | TMR0IF | IOCIF | | | | INTF |
| 0x070D | PIR1 | 7:0 | OSFIF | CSWIF | | | | | ADTIF | ADIF |
| 0x070E | PIR2 | 7:0 | | ZCDIF | | | | | C2IF | C1IF |
| 0x070F | PIR3 | 7:0 | | | RC1IF | TX1IF | BCL2IF | SSP2IF | BCL1IF | SSP1IF |
| 0x0710 | PIR4 | 7:0 | | | TMR6IF | TMR5IF | TMR4IF | TMR3IF | TMR2IF | TMR1IF |
| 0x0711 | PIR5 | 7:0 | CLC4IF | CLC3IF | CL24IF | CLC1IF | | TMR5GIF | TMR3GIF | TMR1GIF |
| 0x0712 | PIR6 | 7:0 | | | | | CCP4IF | CCP3IF | CCP2IF | CCP1IF |
| 0x0713 | PIR7 | 7:0 | | | NVMIF | NCO1IF | | | CWG2IF | CWG1IF |
| 0x0714 | PIR8 | 7:0 | | | | | | SMT1PWAIF | SMT1PRAIF | SMT1IF |
| 0x0715 | Reserved | | | | | | | | | |
| 0x0716 | PIE0 | 7:0 | | | TMR0IE | IOCIE | | | | INTE |
| 0x0717 | PIE1 | 7:0 | OSFIE | CSWIE | | | | | ADTIE | ADIE |
| 0x0718 | PIE2 | 7:0 | | ZCDIE | | | | | C2IE | C1IE |
| 0x0719 | PIE3 | 7:0 | | | RC1IE | TX1IE | BCL2IE | SSP2IE | BCL1IE | SSP1IE |
| 0x071A | PIE4 | 7:0 | | | TMR6IE | TMR5IE | TMR4IE | TMR3IE | TMR2IE | TMR1IE |
| 0x071B | PIE5 | 7:0 | CLC4IE | CLC3IE | CLC2IE | CLC1IE | | TMR5GIE | TMR3GIE | TMR1GIE |
| 0x071C | PIE6 | 7:0 | | | | | CCP4IE | CCP3IE | CCP2IE | CCP1IE |
| 0x071D | PIE7 | 7:0 | | | NVMIE | NCO1IE | | | CWG2IE | CWG1IE |
| 0x071E | PIE8 | 7:0 | | | | | | SMT1PWAIE | SMT1PRAIE | SMT1IE |

7.7 Register Definitions: Interrupt Control

7.7.1 INTCON

Name: INTCON

Address: 0x00B

Interrupt Control Register

| | | | | | | | | |
|--------|-----|------|---|---|---|---|---|--------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | GIE | PEIE | | | | | | INTEDG |
| Access | R/W | R/W | | | | | | R/W |
| Reset | 0 | 0 | | | | | | 1 |

Bit 7 – GIE Global Interrupt Enable bit

| Value | Description |
|-------|-------------------------------|
| 1 | Enables all active interrupts |
| 0 | Disables all interrupts |

Bit 6 – PEIE Peripheral Interrupt Enable bit

| Value | Description |
|-------|--|
| 1 | Enables all active peripheral interrupts |
| 0 | Disables all peripheral interrupts |

Bit 0 – INTEDG External Interrupt Edge Select bit

| Value | Description |
|-------|--------------------------------------|
| 1 | Interrupt on rising edge of INT pin |
| 0 | Interrupt on falling edge of INT pin |



Important: Interrupt flag bits are set when an interrupt condition occurs, regardless of the state of its corresponding enable bit or the global enable bit. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt. This feature allows for software polling.

7.7.2 PIE0

Name: PIE0
Address: 0x716

Peripheral Interrupt Enable Register 0

| | | | | | | | | |
|--------|---|---|--------|------|---|---|---|------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | | TMR0IE | IOIE | | | | INTE |
| Access | | | R/W | R/W | | | | R/W |
| Reset | | | 0 | 0 | | | | 0 |

Bit 5 – TMR0IE Timer0 Interrupt Enable bit

| Value | Description |
|-------|-------------|
| 1 | Enabled |
| 0 | Disabled |

Bit 4 – IOIE Interrupt-on-Change Enable bit

| Value | Description |
|-------|-------------|
| 1 | Enabled |
| 0 | Disabled |

Bit 0 – INTE External Interrupt Enable bit⁽¹⁾

| Value | Description |
|-------|-------------|
| 1 | Enabled |
| 0 | Disabled |

Note:

- The External Interrupt INT pin is selected by INTPPS.

Note: Bit PEIE of the INTCON register must be set to enable any peripheral interrupt controlled by PIE1-PIE8. Interrupt sources controlled by the PIE0 register do not require PEIE to be set in order to allow interrupt vectoring (when GIE is set).

Related Links

[16.9.2 xxxPPS](#)

7.7.3 PIE1

Name: PIE1
Address: 0x717

Peripheral Interrupt Enable Register 1

| | | | | | | | | |
|--------|-------|-------|---|---|---|---|-------|------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | OSFIE | CSWIE | | | | | ADTIE | ADIE |
| Access | R/W | R/W | | | | | R/W | R/W |
| Reset | 0 | 0 | | | | | 0 | 0 |

Bit 7 – OSFIE Oscillator Fail Interrupt Enable bit

| Value | Description |
|-------|-------------|
| 1 | Enabled |
| 0 | Disabled |

Bit 6 – CSWIE Clock-Switch Interrupt Enable bit

| Value | Description |
|-------|-------------|
| 1 | Enabled |
| 0 | Disabled |

Bit 1 – ADTIE ADC Threshold Interrupt Enable bit

| Value | Description |
|-------|-------------|
| 1 | Enabled |
| 0 | Disabled |

Bit 0 – ADIE ADC Interrupt Enable bit

| Value | Description |
|-------|-------------|
| 1 | Enabled |
| 0 | Disabled |

Note: Bit PEIE of the INTCON register must be set to enable any peripheral interrupt controlled by registers PIE1-PIE8.

7.7.4 PIE2

Name: PIE2
Address: 0x718

Peripheral Interrupt Enable Register 2

| | | | | | | | | |
|--------|---|-------|---|---|---|---|------|------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | ZCDIE | | | | | C2IE | C1IE |
| Access | | R/W | | | | | R/W | R/W |
| Reset | | 0 | | | | | 0 | 0 |

Bit 6 – ZCDIE Zero-Cross Detect Interrupt Enable bit

| Value | Description |
|-------|-------------|
| 1 | Enabled |
| 0 | Disabled |

Bits 0, 1 – CnIE Comparator ‘n’ Interrupt Enable bit

| Value | Description |
|-------|-------------|
| 1 | Enabled |
| 0 | Disabled |

Note: Bit PEIE of the INTCON register must be set to enable any peripheral interrupt controlled by registers PIE1-PIE8.

7.7.5 PIE3

Name: PIE3
Address: 0x719

Peripheral Interrupt Enable Register 3

| | | | | | | | | |
|--------|---|---|-------|-------|--------|--------|--------|--------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | | RC1IE | TX1IE | BCL2IE | SSP2IE | BCL1IE | SSP1IE |
| Access | | | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 |

Bit 5 – RCnIE EUSARTn Receive Interrupt Enable bit

| Value | Description |
|-------|-------------|
| 1 | Enabled |
| 0 | Disabled |

Bit 4 – TXnIE EUSARTn Transmit Interrupt Enable bit

| Value | Description |
|-------|-------------|
| 1 | Enabled |
| 0 | Disabled |

Bits 1, 3 – BCLnIE MSSPn Bus Collision Interrupt Enable bit

| Value | Description |
|-------|-------------|
| 1 | Enabled |
| 0 | Disabled |

Bits 0, 2 – SSPnIE Synchronous Serial Port ‘n’ Interrupt Enable bit

| Value | Description |
|-------|-------------|
| 1 | Enabled |
| 0 | Disabled |

Note: Bit PEIE of the INTCON register must be set to enable any peripheral interrupt controlled by registers PIE1-PIE8.

7.7.6 PIE4

Name: PIE4
Address: 0x71A

Peripheral Interrupt Enable Register 4

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|---|---|--------|--------|--------|--------|--------|--------|
| | | | TMR6IE | TMR5IE | TMR4IE | TMR3IE | TMR2IE | TMR1IE |
| Access | | | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 |

Bit 5 – TMR6IE TMR6 to PR6 Match Interrupt Enable bit

| Value | Description |
|-------|-------------|
| 1 | Enabled |
| 0 | Disabled |

Bit 4 – TMR5IE TMR5 Overflow Interrupt Enable bit

| Value | Description |
|-------|-------------|
| 1 | Enabled |
| 0 | Disabled |

Bit 3 – TMR4IE TMR4 to PR4 Match Interrupt Enable bit

| Value | Description |
|-------|-------------|
| 1 | Enabled |
| 0 | Disabled |

Bit 2 – TMR3IE TMR3 Overflow Interrupt Enable bit

| Value | Description |
|-------|-------------|
| 1 | Enabled |
| 0 | Disabled |

Bit 1 – TMR2IE TMR2 to PR2 Match Interrupt Enable bit

| Value | Description |
|-------|-------------|
| 1 | Enabled |
| 0 | Disabled |

Bit 0 – TMR1IE TMR1 Overflow Interrupt Enable bit

| Value | Description |
|-------|-------------|
| 1 | Enabled |
| 0 | Disabled |

Note: Bit PEIE of the INTCON register must be set to enable any peripheral interrupt controlled by registers PIE1-PIE8.

7.7.7 PIE5

Name: PIE5
Address: 0x71B

Peripheral Interrupt Enable Register 5

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|--------|--------|--------|--------|---|---------|---------|---------|
| | CLC4IE | CLC3IE | CLC2IE | CLC1IE | | TMR5GIE | TMR3GIE | TMR1GIE |
| Access | R/W | R/W | R/W | R/W | | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | | 0 | 0 | 0 |

Bit 7 – CLC4IE CLC4 Interrupt Enable bit

| Value | Description |
|-------|-------------|
| 1 | Enabled |
| 0 | Disabled |

Bit 6 – CLC3IE CLC3 Interrupt Enable bit

| Value | Description |
|-------|-------------|
| 1 | Enabled |
| 0 | Disabled |

Bit 5 – CLC2IE CLC2 Interrupt Enable bit

| Value | Description |
|-------|-------------|
| 1 | Enabled |
| 0 | Disabled |

Bit 4 – CLC1IE CLC1 Interrupt Enable bit

| Value | Description |
|-------|-------------|
| 1 | Enabled |
| 0 | Disabled |

Bit 2 – TMR5GIE TMR5 Gate Interrupt Enable bit

| Value | Description |
|-------|-------------|
| 1 | Enabled |
| 0 | Disabled |

Bit 1 – TMR3GIE TMR3 Gate Interrupt Enable bit

| Value | Description |
|-------|-------------|
| 1 | Enabled |
| 0 | Disabled |

Bit 0 – TMR1GIE TMR1 Gate Interrupt Enable bit

| Value | Description |
|-------|-------------|
| 1 | Enabled |
| 0 | Disabled |

Note: Bit PEIE of the INTCON register must be set to enable any peripheral interrupt controlled by registers PIE1-PIE8.

7.7.8 PIE6

Name: PIE6
Address: 0x71C

Peripheral Interrupt Enable Register 6

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|---|---|---|---|--------|--------|--------|--------|
| | | | | | CCP4IE | CCP3IE | CCP2IE | CCP1IE |
| Access | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | 0 | 0 | 0 | 0 |

Bit 3 – CCP4IE CCP4 Interrupt Enable bit

| Value | Description |
|-------|-------------|
| 1 | Enabled |
| 0 | Disabled |

Bit 2 – CCP3IE CCP3 Interrupt Enable bit

| Value | Description |
|-------|-------------|
| 1 | Enabled |
| 0 | Disabled |

Bit 1 – CCP2IE CCP2 Interrupt Enable bit

| Value | Description |
|-------|-------------|
| 1 | Enabled |
| 0 | Disabled |

Bit 0 – CCP1IE CCP1 Interrupt Enable bit

| Value | Description |
|-------|-------------|
| 1 | Enabled |
| 0 | Disabled |

Note: Bit PEIE of the INTCON register must be set to enable any peripheral interrupt controlled by registers PIE1-PIE8.

7.7.9 PIE7

Name: PIE7
Address: 0x71D

Peripheral Interrupt Enable Register 7

| | | | | | | | | |
|--------|---|---|-------|--------|---|---|--------|--------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | | NVMIE | NCO1IE | | | CWG2IE | CWG1IE |
| Access | | | R/W | R/W | | | R/W | R/W |
| Reset | | | 0 | 0 | | | 0 | 0 |

Bit 5 – NVMIE NVM Interrupt Enable bit

| Value | Description |
|-------|-------------|
| 1 | Enabled |
| 0 | Disabled |

Bit 4 – NCO1IE NCO Interrupt Enable bit

| Value | Description |
|-------|-------------|
| 1 | Enabled |
| 0 | Disabled |

Bit 1 – CWG2IE CWG2 Interrupt Enable bit

| Value | Description |
|-------|-------------|
| 1 | Enabled |
| 0 | Disabled |

Bit 0 – CWG1IE CWG1 Interrupt Enable bit

| Value | Description |
|-------|-------------|
| 1 | Enabled |
| 0 | Disabled |

Note: Bit PEIE of the INTCON register must be set to enable any peripheral interrupt controlled by registers PIE1-PIE8.

7.7.10 PIE8

Name: PIE8
Address: 0x71E

Peripheral Interrupt Enable Register 8

| | | | | | | | | |
|--------|---|---|---|---|---|-----------|-----------|--------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | | | | | SMT1PWAIE | SMT1PRAIE | SMT1IE |
| Access | | | | | | R/W | R/W | R/W |
| Reset | | | | | | 0 | 0 | 0 |

Bit 2 – SMT1PWAIE SMT1 Pulse-width Acquisition Interrupt Enable bit

| Value | Description |
|-------|-------------|
| 1 | Enabled |
| 0 | Disabled |

Bit 1 – SMT1PRAIE SMT1 Period Acquisition Interrupt Enable bit

| Value | Description |
|-------|-------------|
| 1 | Enabled |
| 0 | Disabled |

Bit 0 – SMT1IE SMT1 Counter Overflow Interrupt Enable bit

| Value | Description |
|-------|-------------|
| 1 | Enabled |
| 0 | Disabled |

Note: Bit PEIE of the INTCON register must be set to enable any peripheral interrupt controlled by registers PIE1-PIE8.

7.7.11 PIR0

Name: PIR0
Address: 0x70C

Peripheral Interrupt Request (Flag) Register 0

| | | | | | | | | |
|--------|---|---|--------|-------|---|---|---|--------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | | TMR0IF | IOCIF | | | | INTF |
| Access | | | R/W/HS | RO | | | | R/W/HS |
| Reset | | | 0 | 0 | | | | 0 |

Bit 5 – TMR0IF Timer0 Interrupt Flag bit

| Value | Description |
|-------|--|
| 1 | TMR0 register has overflowed (must be cleared by software) |
| 0 | TMR0 register has not overflowed |

Bit 4 – IOCIF Interrupt-on-Change Flag bit⁽²⁾

| Value | Description |
|-------|--|
| 1 | One or more of the IOCAF-IOCEF register bits are currently set, indicating an enabled edge was detected by the IOC module. |
| 0 | None of the IOCAF-IOCEF register bits are currently set |

Bit 0 – INTF External Interrupt Flag bit⁽¹⁾

| Value | Description |
|-------|-------------------------------------|
| 1 | External Interrupt has occurred |
| 0 | External Interrupt has not occurred |

Note:

1. The External Interrupt INT pin is selected by INTPPS.
2. The IOCIF bit is the logical OR of all the IOCAF-IOCEF flags. Therefore, to clear the IOCIF flag, application firmware must clear all of the lower level IOCAF-IOCEF register bits.

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 appropriate interrupt flag bits are clear prior to enabling an interrupt.

Related Links

[16.9.2 xxxPPS](#)

7.7.12 PIR1

Name: PIR1
Address: 0x70D

Peripheral Interrupt Request (Flag) Register 1

| | | | | | | | | |
|--------|--------|--------|---|---|---|---|--------|--------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | OSFIF | CSWIF | | | | | ADTIF | ADIF |
| Access | R/W/HS | R/W/HS | | | | | R/W/HS | R/W/HS |
| Reset | 0 | 0 | | | | | 0 | 0 |

Bit 7 – OSFIF Oscillator Fail Interrupt Flag bit

| Value | Description |
|-------|---|
| 1 | Oscillator fail-safe interrupt has occurred (must be cleared in software) |
| 0 | No oscillator fail-safe interrupt |

Bit 6 – CSWIF Clock-Switch Complete Interrupt Flag bit

| Value | Description |
|-------|--|
| 1 | The clock switch module indicates an interrupt condition and is ready to complete the clock switch operation (must be cleared in software) |
| 0 | The clock switch does not indicate an interrupt condition |

Bit 1 – ADTIF ADC Threshold Interrupt Flag bit

| Value | Description |
|-------|--|
| 1 | An A/D conversion or complex operation has completed (must be cleared in software) |
| 0 | An A/D conversion or complex operation is not complete |

Bit 0 – ADIF ADC Interrupt Flag bit

| Value | Description |
|-------|--|
| 1 | An A/D conversion or complex operation has completed (must be cleared in software) |
| 0 | An A/D conversion or complex operation is not complete |

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 appropriate interrupt flag bits are clear prior to enabling an interrupt.

7.7.13 PIR2

Name: PIR2
Address: 0x70E

Peripheral Interrupt Request (Flag) Register 2

| | | | | | | | | |
|--------|---|--------|---|---|---|---|--------|--------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | ZCDIF | | | | | C2IF | C1IF |
| Access | | R/W/HS | | | | | R/W/HS | R/W/HS |
| Reset | | 0 | | | | | 0 | 0 |

Bit 6 – ZCDIF Zero-Cross Detect Interrupt Flag bit

| Value | Description |
|-------|---|
| 1 | An enabled rising and/or falling ZCD1 event has been detected (must be cleared in software) |
| 0 | No ZCD1 event has occurred |

Bits 0, 1 – CnIF Comparator ‘n’ Interrupt Flag bit

| Value | Description |
|-------|--|
| 1 | Comparator Cn interrupt asserted (must be cleared in software) |
| 0 | Comparator Cn interrupt not asserted |

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 appropriate interrupt flag bits are clear prior to enabling an interrupt.

7.7.14 PIR3

Name: PIR3
Address: 0x70F

Peripheral Interrupt Request (Flag) Register 3

| | | | | | | | | |
|--------|---|---|-------|-------|--------|--------|--------|--------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Access | | | RC1IF | TX1IF | BCL2IF | SSP2IF | BCL1IF | SSP1IF |
| Reset | | | RO/HS | RO/HS | R/W/HS | R/W/HS | R/W/HS | R/W/HS |
| | | | 0 | 0 | 0 | 0 | 0 | 0 |

Bit 5 – RCnIF EUSARTn Receive Interrupt Flag bit⁽¹⁾

| Value | Description |
|-------|--|
| 1 | The EUSARTn receive buffer is not empty (contains at least one byte) |
| 0 | The EUSARTn receive buffer is empty |

Bit 4 – TXnIF EUSARTn Transmit Interrupt Flag bit⁽²⁾

| Value | Description |
|-------|---|
| 1 | The EUSARTn transmit buffer contains at least one unoccupied space |
| 0 | The EUSARTn transmit buffer is currently full. The application firmware should not write to TXnREG again, until more room becomes available in the transmit buffer. |

Bits 1, 3 – BCLnIF MSSPn Bus Collision Interrupt Flag bit

| Value | Description |
|-------|--|
| 1 | A bus collision was detected (must be cleared in software) |
| 0 | No bus collision was detected |

Bits 0, 2 – SSPnIF Synchronous Serial Port 'n' Interrupt Flag bit

| Value | Description |
|-------|--|
| 1 | The Transmission/Reception/Bus Condition is complete (must be cleared in software) |
| 0 | Waiting for the Transmission/Reception/Bus Condition in progress |

Note:

1. The RCnIF flag is a read-only bit. To clear the RCnIF flag, the firmware must read from RCnREG enough times to remove all bytes from the receive buffer.
2. The TXnIF flag is a read-only bit, indicating if there is room in the transmit buffer. To clear the TXnIF flag, the firmware must write enough data to TXnREG to completely fill all available bytes in the buffer. The TXnIF flag does not indicate transmit completion (use TRMT for this purpose instead).

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 appropriate interrupt flag bits are clear prior to enabling an interrupt.

7.7.15 PIR4

Name: PIR4
Address: 0x710

Peripheral Interrupt Request (Flag) Register 4

| | | | | | | | | |
|--------|---|---|--------|--------|--------|--------|--------|--------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | | TMR6IF | TMR5IF | TMR4IF | TMR3IF | TMR2IF | TMR1IF |
| Access | | | R/W/HS | R/W/HS | R/W/HS | R/W/HS | R/W/HS | R/W/HS |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 |

Bit 5 – TMR6IF TMR6 to PR6 Match Interrupt Flag bit

| Value | Description |
|-------|--|
| 1 | The TMR6 postscaler overflowed, or in 1:1 mode, a TMR6 to PR6 match occurred (must be cleared in software) |
| 0 | No TMR6 event has occurred |

Bit 4 – TMR5IF TMR5 Overflow Interrupt Flag bit

| Value | Description |
|-------|--|
| 1 | TMR5 register overflowed (must be cleared in software) |
| 0 | TMR5 register did not overflow |

Bit 3 – TMR4IF TMR4 to PR4 Match Interrupt Flag bit

| Value | Description |
|-------|--|
| 1 | The TMR4 postscaler overflowed, or in 1:1 mode, a TMR4 to PR4 match occurred (must be cleared in software) |
| 0 | No TMR4 event has occurred |

Bit 2 – TMR3IF TMR3 Overflow Interrupt Flag bit

| Value | Description |
|-------|--|
| 1 | TMR3 register overflowed (must be cleared in software) |
| 0 | TMR3 register did not overflow |

Bit 1 – TMR2IF TMR2 to PR2 Match Interrupt Flag bit

| Value | Description |
|-------|--|
| 1 | The TMR2 postscaler overflowed, or in 1:1 mode, a TMR2 to PR2 match occurred (must be cleared in software) |
| 0 | No TMR2 event has occurred |

Bit 0 – TMR1IF TMR1 Overflow Interrupt Flag bit

| Value | Description |
|-------|--|
| 1 | TMR1 register overflowed (must be cleared in software) |
| 0 | TMR1 register did not overflow |

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 appropriate interrupt flag bits are clear prior to enabling an interrupt.

7.7.16 PIR5

Name: PIR5
Address: 0x711

Peripheral Interrupt Request (Flag) Register 5

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|--------|--------|--------|--------|---|---------|---------|---------|
| | CLC4IF | CLC3IF | CL24IF | CLC1IF | | TMR5GIF | TMR3GIF | TMR1GIF |
| Access | R/W/HS | R/W/HS | R/W/HS | R/W/HS | | R/W/HS | R/W/HS | R/W/HS |
| Reset | 0 | 0 | 0 | 0 | | 0 | 0 | 0 |

Bit 7 – CLC4IF CLC4 Interrupt Flag bit

| Value | Description |
|-------|--|
| 1 | A CLC4OUT interrupt condition has occurred (must be cleared in software) |
| 0 | No CLC4 interrupt event has occurred |

Bit 6 – CLC3IF CLC3 Interrupt Flag bit

| Value | Description |
|-------|--|
| 1 | A CLC3OUT interrupt condition has occurred (must be cleared in software) |
| 0 | No CLC3 interrupt event has occurred |

Bit 5 – CL24IF CLC2 Interrupt Flag bit

| Value | Description |
|-------|--|
| 1 | A CLC2OUT interrupt condition has occurred (must be cleared in software) |
| 0 | No CLC2 interrupt event has occurred |

Bit 4 – CLC1IF CLC1 Interrupt Flag bit

| Value | Description |
|-------|--|
| 1 | A CLC1OUT interrupt condition has occurred (must be cleared in software) |
| 0 | No CLC1 interrupt event has occurred |

Bit 2 – TMR5GIF TMR5 Gate Interrupt Flag bit

| Value | Description |
|-------|---|
| 1 | The Timer5 Gate has gone inactive (the acquisition is complete) |
| 0 | The Timer5 Gate has not gone inactive |

Bit 1 – TMR3GIF TMR3 Gate Interrupt Flag bit

| Value | Description |
|-------|---|
| 1 | The Timer3 Gate has gone inactive (the acquisition is complete) |
| 0 | The Timer3 Gate has not gone inactive |

Bit 0 – TMR1GIF TMR1 Gate Interrupt Flag bit

| Value | Description |
|-------|---|
| 1 | The Timer1 Gate has gone inactive (the acquisition is complete) |
| 0 | The Timer1 Gate has not gone inactive |

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 appropriate interrupt flag bits are clear prior to enabling an interrupt.

7.7.17 PIR6

Name: PIR6
Address: 0x712

PIR6 Peripheral Interrupt Request (Flag) Register 6

| | | | | | | | | |
|--------|---|---|---|---|--------|--------|--------|--------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | | | | CCP4IF | CCP3IF | CCP2IF | CCP1IF |
| Access | | | | | R/W/HS | R/W/HS | R/W/HS | R/W/HS |
| Reset | | | | | 0 | 0 | 0 | 0 |

Bit 3 – CCP4IF CCP4 Interrupt Flag bit

| Value | Condition | Description |
|-------|--------------|---|
| 1 | Capture mode | Capture occurred (must be cleared in software) |
| 0 | Capture mode | Capture did not occur |
| 1 | Compare mode | Compare match occurred (must be cleared in software) |
| 0 | Compare mode | Compare match did not occur |
| 1 | PWM mode | Output trailing edge occurred (must be cleared in software) |
| 0 | PWM mode | Output trailing edge did not occur |

Bit 2 – CCP3IF CCP3 Interrupt Flag bit

| Value | Condition | Description |
|-------|--------------|---|
| 1 | Capture mode | Capture occurred (must be cleared in software) |
| 0 | Capture mode | Capture did not occur |
| 1 | Compare mode | Compare match occurred (must be cleared in software) |
| 0 | Compare mode | Compare match did not occur |
| 1 | PWM mode | Output trailing edge occurred (must be cleared in software) |
| 0 | PWM mode | Output trailing edge did not occur |

Bit 1 – CCP2IF CCP2 Interrupt Flag bit

| Value | Condition | Description |
|-------|--------------|---|
| 1 | Capture mode | Capture occurred (must be cleared in software) |
| 0 | Capture mode | Capture did not occur |
| 1 | Compare mode | Compare match occurred (must be cleared in software) |
| 0 | Compare mode | Compare match did not occur |
| 1 | PWM mode | Output trailing edge occurred (must be cleared in software) |
| 0 | PWM mode | Output trailing edge did not occur |

Bit 0 – CCP1IF CCP1 Interrupt Flag bit

| Value | Condition | Description |
|-------|--------------|---|
| 1 | Capture mode | Capture occurred (must be cleared in software) |
| 0 | Capture mode | Capture did not occur |
| 1 | Compare mode | Compare match occurred (must be cleared in software) |
| 0 | Compare mode | Compare match did not occur |
| 1 | PWM mode | Output trailing edge occurred (must be cleared in software) |
| 0 | PWM mode | Output trailing edge did not occur |

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 appropriate interrupt flag bits are clear prior to enabling an interrupt.

7.7.18 PIR7

Name: PIR7
Address: 0x713

Peripheral Interrupt Request (Flag) Register 7

| | | | | | | | | |
|--------|---|---|--------|--------|---|---|--------|--------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | | NVMIF | NCO1IF | | | CWG2IF | CWG1IF |
| Access | | | R/W/HS | R/W/HS | | | R/W/HS | R/W/HS |
| Reset | | | 0 | 0 | | | 0 | 0 |

Bit 5 – NVMIF NVM Interrupt Flag bit

| Value | Description |
|-------|---|
| 1 | The requested NVM operation has completed |
| 0 | NVM interrupt not asserted |

Bit 4 – NCO1IF Numerically Controlled Oscillator (NCO) Interrupt Flag bit

| Value | Description |
|-------|-------------------------------------|
| 1 | The NCO has rolled over |
| 0 | No NCO interrupt event has occurred |

Bit 1 – CWG2IF CWG2 Interrupt Flag bit

| Value | Description |
|-------|--|
| 1 | CWG2 has gone into shutdown |
| 0 | CWG2 is operating normally, or interrupt cleared |

Bit 0 – CWG1IF CWG1 Interrupt Flag bit

| Value | Description |
|-------|--|
| 1 | CWG1 has gone into shutdown |
| 0 | CWG1 is operating normally, or interrupt cleared |

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 appropriate interrupt flag bits are clear prior to enabling an interrupt.

7.7.19 PIR8

Name: PIR8
Address: 0x714

Peripheral Interrupt Request (Flag) Register 8

| | | | | | | | | |
|--------|---|---|---|---|---|-----------|-----------|--------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Access | | | | | | SMT1PWAIF | SMT1PRAIF | SMT1IF |
| Reset | | | | | | R/W/HS | R/W/HS | R/W/HS |
| | | | | | | 0 | 0 | 0 |

Bit 2 – SMT1PWAIF SMT1 Pulse-Width Acquisition Interrupt Flag bit

| Value | Description |
|-------|--|
| 1 | Interrupt has occurred (must be cleared by software) |
| 0 | Interrupt event has not occurred |

Bit 1 – SMT1PRAIF SMT1 Period Acquisition Interrupt Flag bit

| Value | Description |
|-------|--|
| 1 | Interrupt has occurred (must be cleared by software) |
| 0 | Interrupt event has not occurred |

Bit 0 – SMT1IF SMT1 Interrupt Flag bit

| Value | Description |
|-------|--|
| 1 | Interrupt has occurred (must be cleared by software) |
| 0 | Interrupt event has not occurred |

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 appropriate interrupt flag bits are clear prior to enabling an interrupt.

8. OSC - Oscillator Module

8.1 Overview

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

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

- Selectable system clock source between external or internal sources via software.
- Fail-Safe Clock Monitor (FSCM) designed to detect a failure of the external clock source (LP, XT, HS, ECH, ECM, ECL) and switch automatically to the internal oscillator.
- Oscillator Start-up Timer (OST) ensures stability of crystal oscillator sources.

The RSTOSC bits of Configuration Word 1 determine the type of oscillator that will be used when the device runs after Reset, including when it is first powered up.

The internal clock modes, LFINTOSC, HFINTOSC (set at 1 MHz), or HFINTOSC (set at 32 MHz) can be set through the RSTOSC bits.

If an external clock source is selected, the FEXTOSC bits of Configuration Word 1 must be used in conjunction with the RSTOSC bits to select the External Clock mode.

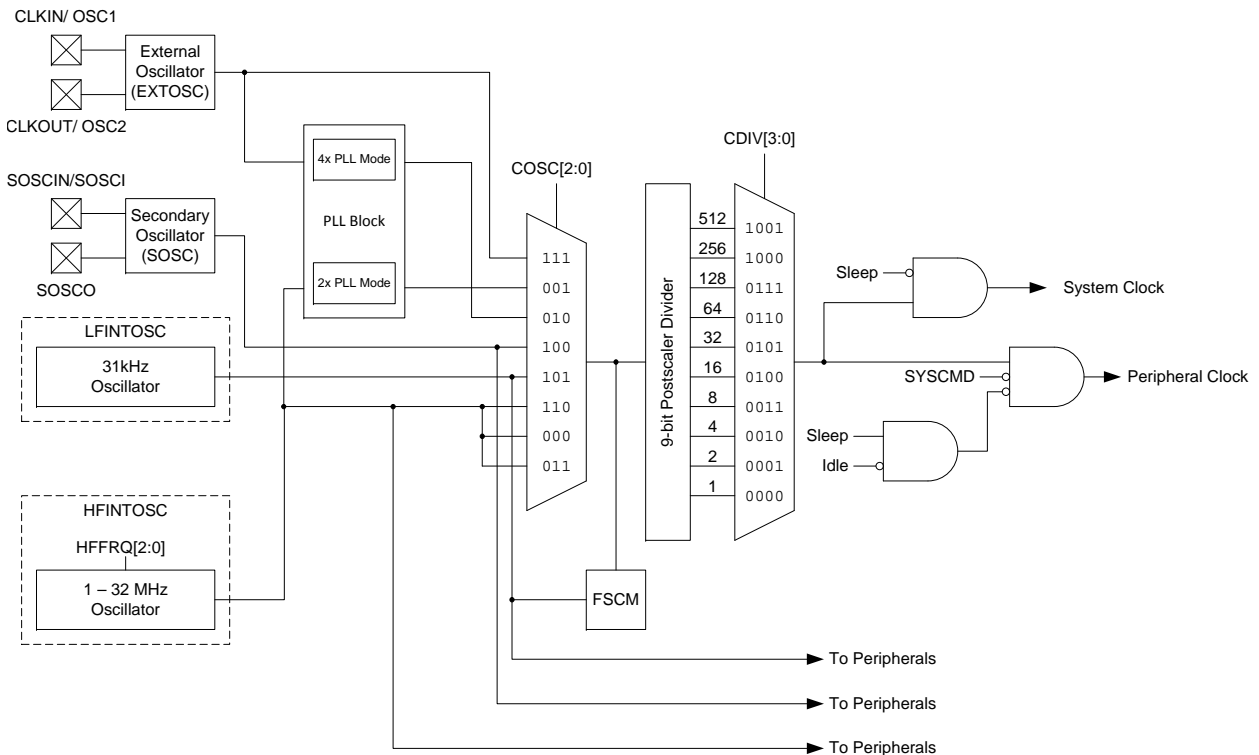
The external oscillator module can be configured in one of the following clock modes, by setting the FEXTOSC bits of [Configuration Word 1](#):

1. ECL – External Clock Low-Power mode (≤ 500 kHz)
2. ECM – External Clock Medium-Power mode (≤ 8 MHz)
3. ECH – External Clock High-Power mode (≤ 32 MHz)
4. LP – 32 kHz Low-Power Crystal mode
5. XT – Medium Gain Crystal or Ceramic Resonator Oscillator mode (between 100 kHz and 4 MHz)
6. HS – High Gain Crystal or Ceramic Resonator mode (above 4 MHz)

The ECH, ECM, and ECL Clock modes rely on an external logic level signal as the device clock source. The LP, XT, and HS Clock modes require an external crystal or resonator to be connected to the device. Each mode is optimized for a different frequency range. The internal oscillator block produces low and high-frequency clock sources, designated LFINTOSC and HFINTOSC. Multiple device clock frequencies may be derived from these clock sources.

Figure 8-1. Simplified PIC[®] MCU Clock Source Block Diagram

Rev. 10-00028M
5/17/2017



Related Links

[4.7.1 CONFIG1](#)

8.2 Clock Source Types

Clock sources can be classified as external or internal.

External clock sources rely on external circuitry for the clock source to function. Examples include: oscillator modules (ECH, ECM, ECL mode) and quartz crystal resonators or ceramic resonators (LP, XT and HS modes).

There is also a secondary oscillator block which is optimized for a 32.768 kHz external clock source, which can be used as an alternate clock source.

There are two internal oscillator blocks:

- HFINTOSC
- LFINTOSC

The HFINTOSC can produce clock frequencies from 1-32 MHz, and is responsible for generating the two MFINTOSC frequencies (500 kHz and 32 kHz) that can be used by some peripherals. The LFINTOSC generates a 31 kHz clock frequency.

There is a 4x PLL that can be used by the external oscillator. Additionally, there is a 2x PLL that can be used by the HFINTOSC at certain frequencies.

Related Links

[8.2.1.4 4x PLL](#)

[8.2.2.3 2x PLL](#)

8.2.1 External Clock Sources

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

- Program the RSTOSC bits in the [Configuration Word 1](#) to select an external clock source that will be used as the default system clock upon a device Reset.
- Write the **NOSC** and **NDIV** bits to switch the system clock source.

Related Links

[8.3 Clock Switching](#)

8.2.1.1 EC Mode

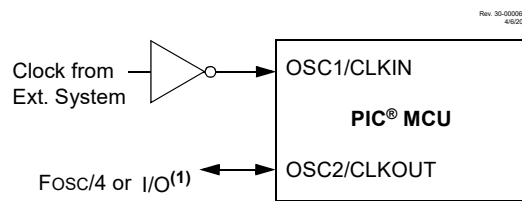
The External Clock (EC) mode allows an externally generated logic level signal to be the system clock source. When operating in this mode, an external clock source is connected to the CLKIN/OSC1 input. OSC2/CLKOUT is available for general purpose I/O or CLKOUT. The [Figure 8-2](#) shows the pin connections for EC mode.

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

- ECH – High power, ≤ 32 MHz
- ECM – Medium power, ≤ 8 MHz
- ECL – Low power, ≤ 0.1 MHz

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 8-2. External Clock (EC) Mode Operation



Note:

1. Output depends upon $\overline{\text{CLKOUTEN}}$ bit of the [Configuration Word 1 \(CONFIG1\)](#).

8.2.1.2 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 8-3](#)). The three modes select a low, medium or high gain setting of the internal inverter-amplifier to support various resonator types and speed.

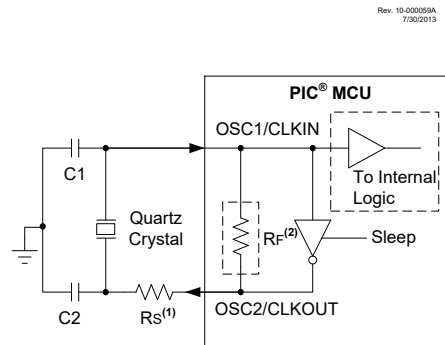
LP Oscillator mode selects the lowest gain setting of the internal inverter-amplifier. LP mode current consumption is the least of the three modes. This mode is designed to drive only 32.768 kHz tuning-fork type crystals (watch crystals). but can operate up to 100 kHz.

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 crystals and resonators with a frequency range up to 4 MHz.

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 operating frequencies up to 20 MHz.

[Figure 8-3](#) and [Figure 8-4](#) show typical circuits for quartz crystal and ceramic resonators, respectively.

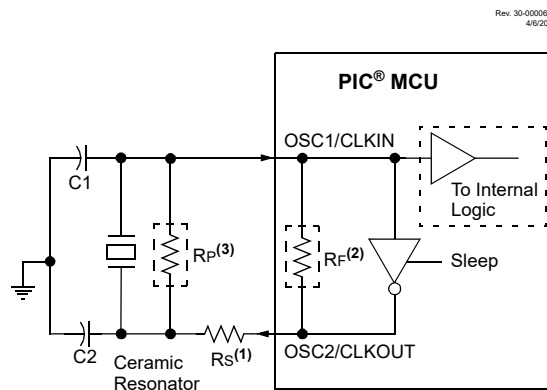
Figure 8-3. Quartz Crystal Operation (LP, XT or HS Mode)



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 V_{DD} and temperature range that is expected for the application.
3. For oscillator design assistance, reference the following Microchip Application Notes:
 - AN826, “Crystal Oscillator Basics and Crystal Selection for PIC® 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 8-4. Ceramic Resonator Operation (XT or HS Mode)



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 V_{DD} and temperature range that is expected for the application.
3. For oscillator design assistance, reference the following Microchip Application 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)

8.2.1.3 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), or a wake-up from Sleep. 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.

8.2.1.4 4x PLL

The oscillator module contains a 4x PLL that can be used with the external clock sources to provide a system clock source. The input frequency for the PLL must fall within specifications.

The PLL can be enabled for use by one of two methods:

1. Program the RSTOSC bits in the [Configuration Word 1](#) to '010' (enable EXTOSC with 4x PLL).
2. Write the NOSC bits to '010' (enable EXTOSC with 4x PLL).

Related Links

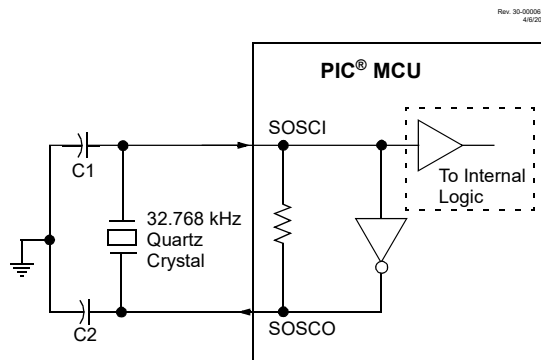
[8.6.1 OSCCON1](#)

[40.4.3 PLL Specifications](#)

8.2.1.5 Secondary Oscillator

The secondary oscillator is a separate oscillator block that can be used as an alternate system clock source. The secondary oscillator is optimized for 32.768 kHz, and can be used with an external crystal oscillator connected to the SOSCI and SOSCO device pins, or an external clock source connected to the SOSCIN pin. The secondary oscillator can be selected during run-time using clock switching.

Figure 8-5. Quartz Crystal Operation (Secondary Oscillator)



Related Links

[8.3 Clock Switching](#)

8.2.2 Internal Clock Sources

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

- Program the RSTOSC bits in [Configuration Word 1](#) to select the INTOSC clock source, which will be used as the default system clock upon a device Reset.
- Write the NOSC bits to switch the system clock source to the internal oscillator during run-time.

The function of the OSC2/CLKOUT pin is determined by the $\overline{\text{CLKOUTEN}}$ bit in [Configuration Word 1](#).

The internal oscillator block has two independent oscillators that can produce two internal system clock sources.

1. The **HFINTOSC** (High-Frequency Internal Oscillator) is factory-calibrated and operates up to 32 MHz.
2. The **LFINTOSC** (Low-Frequency Internal Oscillator) is factory-calibrated and operates at 31 kHz.

Related Links

[8.3 Clock Switching](#)

8.2.2.1 HFINTOSC

The High-Frequency Internal Oscillator (HFINTOSC) is a precision digitally-controlled internal clock source that produces a stable clock up to 32 MHz. The HFINTOSC can be enabled through one of the following methods:

- Programming the RSTOSC bits in [Configuration Word 1](#) to '110' ($F_{\text{OSC}} = 1 \text{ MHz}$) or '000' ($F_{\text{OSC}} = 32 \text{ MHz}$) to set the oscillator upon device Power-up or Reset.

- Write to the [NOSC](#) bits during run-time.

The HFINTOSC frequency can be selected by setting the [HFFRQ](#) bits.

The [NDIV](#) bits allow for division of the output of the selected clock source by a range between 1:1 and 1:512.

Related Links

[8.3 Clock Switching](#)

[8.6.1 OSCCON1](#)

[8.6.7 OSCFRQ](#)

8.2.2.2 MFINTOSC

The module provides two (500 kHz and 31.25 kHz) constant clock outputs. These clocks are digital divisors of the HFINTOSC clock. Dynamic divider logic is used to provide constant MFINTOSC clock rates for all settings of HFINTOSC.

The MFINTOSC cannot be used to drive the system but it is used to clock certain modules such as the Timers and WWDT.

8.2.2.3 2x PLL

The oscillator module contains a PLL that can be used with the HFINTOSC clock source to provide a system clock source. The input frequency to the PLL is limited to 8, 12, or 16 MHz, which will yield a system clock source of 16, 24, or 32 MHz, respectively.

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

1. Program the RSTOSC bits in the Configuration Word 1 to '001' to enable the HFINTOSC (32 MHz). This setting configures the [HFFRQ](#) bits to '101' (16 MHz) and activates the 2x PLL.
2. Write '001' the [NOSC](#) bits to enable the 2x PLL, and write the correct value into the [HFFRQ](#) to select the desired system clock frequency.

Related Links

[8.6.1 OSCCON1](#)

[8.6.7 OSCFRQ](#)

8.2.2.4 Internal Oscillator Frequency Adjustment

The internal oscillator is factory-calibrated. This internal oscillator can be adjusted in software by writing to the OSCTUNE register.

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), WWDT, Fail-Safe Clock Monitor (FSCM) and peripherals, are not affected by the change in frequency.

The default value of the OSCTUNE register is 00h. The value is a 6-bit two's complement number. A value of 1Fh will provide an adjustment to the maximum frequency. A value of 20h will provide an adjustment to the minimum frequency.

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

Related Links

[8.6.6 OSCTUNE](#)

8.2.2.5 LFINTOSC

The Low-Frequency Internal Oscillator (LFINTOSC) is a factory-calibrated 31 kHz internal clock source.

The LFINTOSC is the frequency for the Power-up Timer (PWRT), Windowed Watchdog Timer (WWDT) and Fail-Safe Clock Monitor (FSCM).

The LFINTOSC is enabled through one of the following methods:

- Programming the RSTOSC bits of Configuration Word 1 to enable LFINTOSC.
- Write to the [NOSC](#) bits during run-time.

Related Links

- [8.3 Clock Switching](#)
- [4.7.1 CONFIG1](#)
- [8.6.1 OSCCON1](#)

8.2.2.6 Oscillator Status and Manual Enable

The 'ready' status of each oscillator is displayed in the OSCSTAT register. The oscillators can also be manually enabled through the OSCEN register. Manual enabling makes it possible to verify the operation of the EXTOSC or SOSC crystal oscillators. This can be achieved by enabling the selected oscillator, then watching the corresponding 'ready' state of the oscillator in the OSCSTAT register.

Related Links

- [8.6.4 OSCSTAT](#)
- [8.6.5 OSCEN](#)

8.2.2.7 HFOR and MFOR Bits

The [HFOR](#) and [MFOR](#) bits indicate that the HFINTOSC and MFINTOSC is ready. These clocks are always valid for use at all times, but only accurate after they are ready.

When a new value is loaded into the OSCFRQ register, the HFOR and MFOR bits will clear, and set again when the oscillator is ready. During pending OSCFRQ changes the MFINTOSC clock will stall at a high or a low state, until the HFINTOSC resumes operation.

8.3 Clock Switching

The system clock source can be switched between external and internal clock sources via software using the New Oscillator Source ([NOSC](#)) and New Divider selection request ([NDIV](#)) bits. The following clock sources can be selected:

- External Oscillator (EXTOSC)
- High-Frequency Internal Oscillator (HFINTOSC)
- Low-Frequency Internal Oscillator (LFINTOSC)
- Secondary Oscillator (SOSC)
- EXTOSC with 4x PLL
- HFINTOSC with 2x PLL

8.3.1 New Oscillator Source (NOSC) and New Divider Selection Request (NDIV) Bits

The New Oscillator Source (NOSC) and New Divider Selection Request (NDIV) bits select the system clock source and frequency that are used for the CPU and peripherals.

When new values of NOSC and NDIV are written to OSCCON1, the current oscillator selection will continue to operate while waiting for the new clock source to indicate that it is stable and ready. In some cases, the newly requested source may already be in use, and is ready immediately. In the case of a divider-only change, the new and old sources are the same, so the old source will be ready immediately. The device may enter Sleep while waiting for the switch, as described in [8.3.3 Clock Switch and Sleep](#).

When the new oscillator is ready, the New Oscillator Ready (NOSCR) bit of OSCCON3 is set and also the Clock Switch Interrupt Flag (CSWIF) bit of PIR1 is set. If Clock Switch Interrupts are enabled (CSWIE = 1), an interrupt will be generated at that time. The Oscillator Ready (ORDY) bit of OSCCON3 can also be polled to determine when the oscillator is ready in lieu of an interrupt.

If the Clock Switch Hold (CSWHOLD) bit is clear, the oscillator switch will occur when the New Oscillator is READY bit (NOSCR) is set, and the interrupt (if enabled) will be serviced at the new oscillator setting.

If CSWHOLD is set, the oscillator switch is suspended, while execution continues using the current (old) clock source. When the NOSCR bit is set, software should:

- Set CSWHOLD = 0 so the switch can complete, or
- Copy COSC into NOSC to abandon the switch.

If Doze is in effect, the switch occurs on the next clock cycle, whether or not the CPU is operating during that cycle.

Changing the clock post-divider without changing the clock source (i.e., changing F_{OSC} from 1 MHz to 2 MHz) is handled in the same manner as a clock source change, as described previously. The clock source will already be active, so the switch is relatively quick. CSWHOLD must be clear (CSWHOLD = 0) for the switch to complete.

The current COSC and CDIV are indicated in the OSCCON2 register up to the moment when the switch actually occurs, at which time OSCCON2 is updated and ORDY is set. NOSCR is cleared by hardware to indicate that the switch is complete.

Related Links

- [8.3.3 Clock Switch and Sleep](#)
- [8.6.1 OSCCON1](#)
- [8.6.2 OSCCON2](#)
- [8.6.3 OSCCON3](#)

8.3.2 PLL Input Switch

Switching between the PLL and any non-PLL source is managed as described above. The input to the PLL is established when NOSC selects the PLL, and maintained by the COSC setting.

When NOSC and COSC select the PLL with different input sources, the system continues to run using the COSC setting, and the new source is enabled per NOSC. When the new oscillator is ready (and CSWHOLD = 0), system operation is suspended while the PLL input is switched and the PLL acquires lock. This provides a truly glitch-free clock switch operation.



Important: If the PLL fails to lock, the FSCM will trigger.

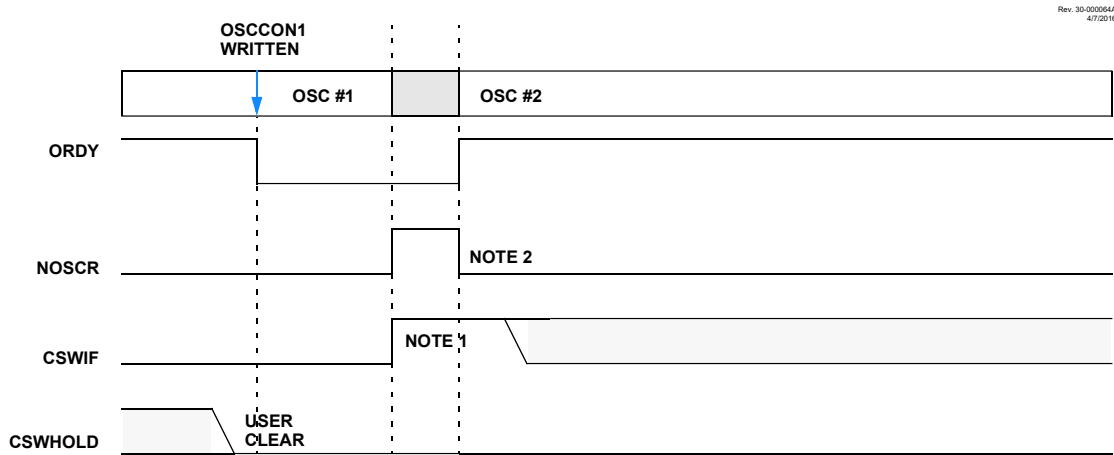
8.3.3 Clock Switch and Sleep

If OSCCON1 is written with a new value and the device is put to Sleep before the switch completes, the switch will not take place and the device will enter Sleep mode.

When the device wakes from Sleep and the CSWHOLD bit is clear, the device will wake with the 'new' clock active, and the Clock Switch Interrupt Flag bit (CSWIF) will be set.

When the device wakes from Sleep and the CSWHOLD bit is set, the device will wake with the 'old' clock active and the new clock will be requested again.

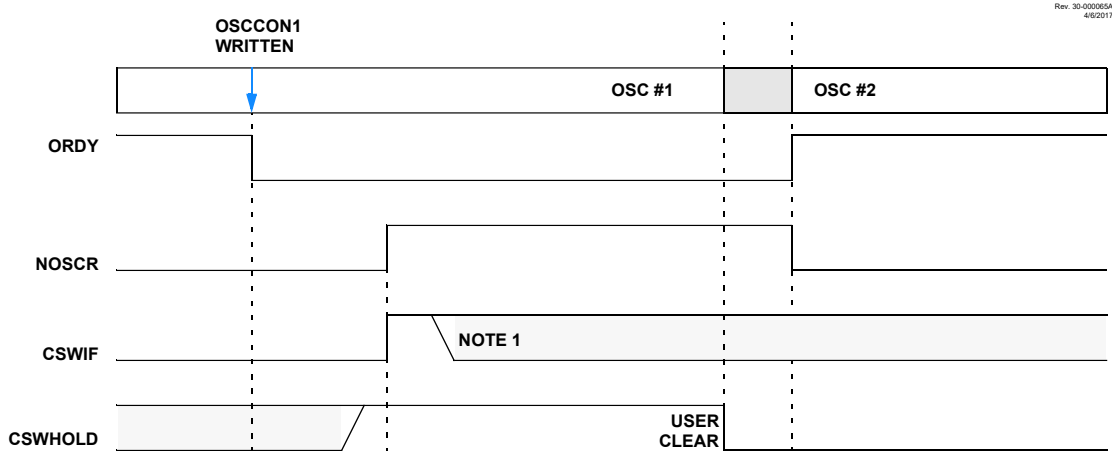
Figure 8-6. Clock Switch (CSWHOLD = 0)



Note:

1. CSWIF is asserted coincident with NOSCR; interrupt is serviced at OSC#2 speed.
2. The assertion of NOSCR is hidden from the user because it appears only for the duration of the switch.

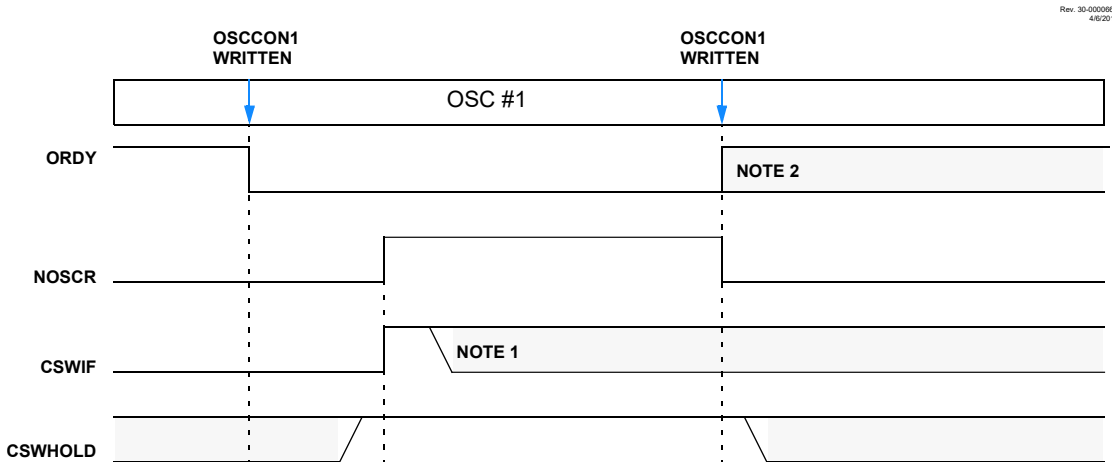
Figure 8-7. Clock Switch (CSWHOLD = 1)



Note:

1. CSWIF is asserted coincident with NOSCR, and may be cleared before or after clearing CSWHOLD = 0.

Figure 8-8. Clock Switch Abandoned



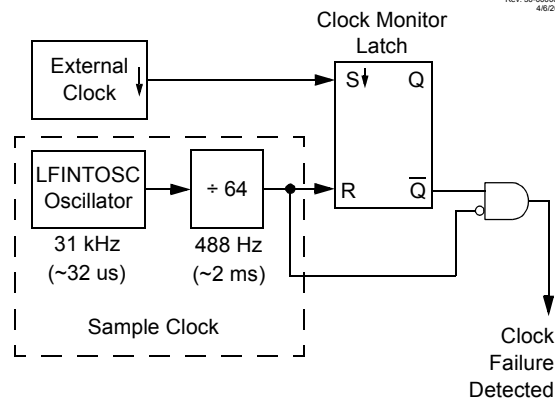
Note:

1. CSWIF may be cleared before or after rewriting OSCCON1; CSWIF is not automatically cleared.
2. ORDY = 0 if OSCCON1 does not match OSCCON2; a new switch will begin.

8.4 Fail-Safe Clock Monitor

The Fail-Safe Clock Monitor (FSCM) allows the device to continue operating should the external oscillator fail. The FSCM is enabled by setting the FCMEN bit in the [Configuration Word 1](#). The FSCM is applicable to all external Oscillator modes (LP, XT, HS, ECL/M/H and Secondary Oscillator).

Figure 8-9. FSCM Block Diagram



8.4.1 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 8-9. 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 external clock goes low.

8.4.2 Fail-Safe Operation

When the external clock fails, the FSCM switches the device clock to the HFINTOSC at 1 MHz clock frequency and sets the OSFIF bit flag of the PIR3 register. Setting this flag will generate an interrupt if the OSFIE bit of the PIE3 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, by writing to the NOSC and NDIV bits of the OSCCON1 register.

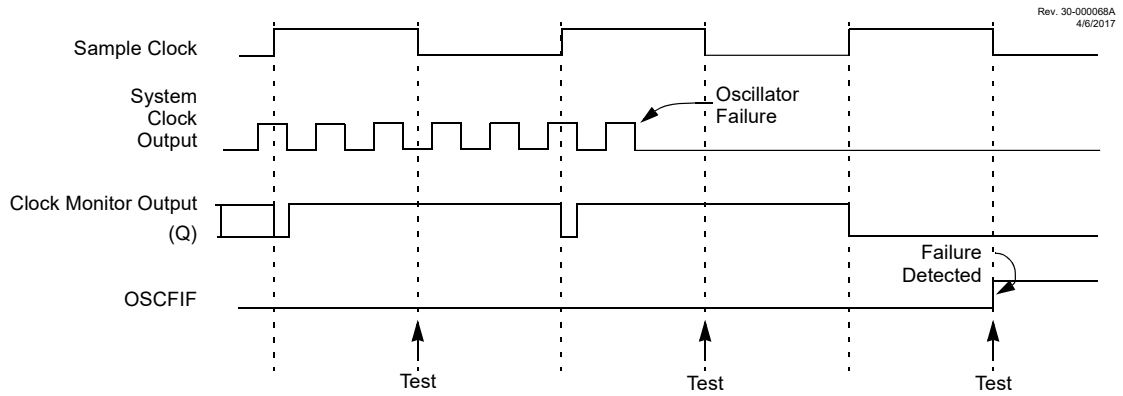
8.4.3 Fail-Safe Condition Clearing

The Fail-Safe condition is cleared after a Reset, executing a `SLEEP` instruction or changing the NOSC and NDIV bits of the OSCCON1 register. When switching to the external oscillator or external oscillator with PLL, the OST is restarted. While the OST is running, the device continues to operate from the INTOSC selected in OSCCON1. When the OST times out, the Fail-Safe condition is cleared after successfully switching to the external clock source. The OSFIF bit should be cleared prior to switching to the external clock source. If the Fail-Safe condition still exists, the OSFIF flag will again be set by hardware.

8.4.4 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 Clock modes so that the FSCM will be active as soon as the Reset or wake-up has completed. Therefore, the device will always be executing code while the OST is operating when using one of the EC modes.

Figure 8-10. 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.

8.5 Register Summary - OSC

| Address | Name | Bit Pos. | | | | | | | | |
|-------------|----------|----------|---------|-----------|------------|-------|-----------|------------|--|------|
| 0x00 ... | Reserved | | | | | | | | | |
| 0x088C | | | | | | | | | | |
| 0x088D | OSCCON1 | 7:0 | | NOSC[2:0] | | | NDIV[3:0] | | | |
| 0x088E | OSCCON2 | 7:0 | | COSC[2:0] | | | CDIV[3:0] | | | |
| 0x088F | OSCCON3 | 7:0 | CSWHOLD | SOSCPWR | | ORDY | NOSCR | | | |
| 0x0890 | OSCSTAT | 7:0 | EXTOR | HFOR | MFOR | LFOR | SOR | ADOR | | PLLR |
| 0x0891 | OSCEN | 7:0 | EXTOEN | HFOEN | MFOEN | LFOEN | SOSCEN | ADOEN | | |
| 0x0892 | OSCTUNE | 7:0 | | | HFTUN[5:0] | | | | | |
| 0x0893 | OSCFRQ | 7:0 | | | | | | HFFRQ[2:0] | | |

8.6 Register Definitions: Oscillator Control

8.6.1 OSCCON1

Name: OSCCON1

Address: 0x88D

Oscillator Control Register1

| | | | | | | | | |
|--------|---|-----------|-----|-----|-----------|-----|-----|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | NOSC[2:0] | | | NDIV[3:0] | | | |
| Access | | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | f | f | f | q | q | q | q |

Bits 6:4 – NOSC[2:0] New Oscillator Source Request bits^(1,2,3)

The setting requests a source oscillator and PLL combination per [Table 8-1](#).

Table 8-1. NOSC Bit Settings

| NOSC[2:0] | Clock Source |
|-----------|----------------------------------|
| 111 | EXTOSC ⁽⁵⁾ |
| 110 | HFINTOSC ⁽⁶⁾ |
| 101 | LFINTOSC |
| 100 | SOSC |
| 011 | Reserved |
| 010 | EXTOSC + 4x PLL ⁽⁵⁾ |
| 001 | HFINTOSC + 2x PLL ⁽⁶⁾ |
| 000 | Reserved |

Bits 3:0 – NDIV[3:0] New Divider Selection Request bits^(2,3,4)

The setting determines the new postscaler division ratio per [Table 8-2](#).

Table 8-2. NDIV Bit Settings

| NDIV[3:0] | Clock Divider |
|-----------|---------------|
| 1111-1010 | Reserved |
| 1001 | 512 |
| 1000 | 256 |
| 0111 | 128 |
| 0110 | 64 |
| 0101 | 32 |
| 0100 | 16 |
| 0011 | 8 |
| 0010 | 4 |
| 0001 | 2 |
| 0000 | 1 |

Note:

1. The default value (f) is determined by the CONFIG1[RSTOSC] Configuration bits.
2. If NOSC is written with a reserved value, the operation is ignored and NOSC is not written.
3. When CONFIG1[CSWEN] = 0, this register is read-only and cannot be changed from the POR value.
4. When NOSC = 110 (HFINTOSC 1 MHz), the NDIV bits will default to '0010' upon Reset; for all other NOSC settings the NDIV bits will default to '0000' upon Reset.
5. EXTOSC configured by CONFIG1[FEXTOSC].
6. HFINTOSC frequency is set with the FRQ bits of the OSCFRQ register.

Related Links

[4.7.1 CONFIG1](#)

[40.4.3 PLL Specifications](#)

8.6.2 OSCCON2

Name: OSCCON2

Address: 0x88E

Oscillator Control Register 2

| | | | | | | | | |
|--------|---|-----------|----|----|-----------|----|----|----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | COSC[2:0] | | | CDIV[3:0] | | | |
| Access | | RO | RO | RO | RO | RO | RO | RO |
| Reset | | n | n | n | n | n | n | n |

Bits 6:4 – COSC[2:0] Current Oscillator Source Select bits (read-only)^(1,2)

Indicates the current source oscillator and PLL combination, as shown in the following table.

Table 8-3. COSC Bit Settings

| COSC/NOSC | Clock Source |
|-----------|----------------------------------|
| 111 | EXTOSC ⁽³⁾ |
| 110 | HFINTOSC ⁽⁴⁾ |
| 101 | LFINTOSC |
| 100 | SOSC |
| 011 | Reserved |
| 010 | EXTOSC + 4x PLL ⁽³⁾ |
| 001 | HFINTOSC + 2x PLL ⁽⁴⁾ |
| 000 | Reserved |

Bits 3:0 – CDIV[3:0] Current Divider Select bits (read-only)^(1,2)

Indicates the current postscaler division ratio, as shown in the following table.

Table 8-4. CDIV Bit Settings

| CDIV/NDIV | Clock Divider |
|-----------|---------------|
| 1111-1010 | Reserved |
| 1001 | 512 |
| 1000 | 256 |
| 0111 | 128 |
| 0110 | 64 |
| 0101 | 32 |
| 0100 | 16 |
| 0011 | 8 |
| 0010 | 4 |
| 0001 | 2 |
| 0000 | 1 |

Note:

1. The POR value is the value present when user code execution begins.
2. The Reset value (n) is the same as the OSCCON1[NOSC/NDIV] bits.
3. EXTOSC configured by the CONFIG1[FEXTOSC] bits.
4. HFINTOSC frequency is configured with the FRQ bits of the OSCFRQ register.

Related Links

[4.7.1 CONFIG1](#)

[40.4.3 PLL Specifications](#)

8.6.3 OSCCON3

Name: OSCCON3

Address: 0x88F

Oscillator Control Register 3

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|---------|---------|---|------|-------|---|---|---|
| | CSWHOLD | SOSCPWR | | ORDY | NOSCR | | | |
| Access | R/W/HC | R/W | | RO | RO | | | |
| Reset | 0 | 0 | | 0 | 0 | | | |

Bit 7 – CSWHOLD Clock Switch Hold bit

| Value | Description |
|-------|---|
| 1 | Clock switch will hold (with interrupt) when the oscillator selected by NOSC is ready |
| 0 | Clock switch may proceed when the oscillator selected by NOSC is ready; when NOSCR becomes '1', the switch will occur |

Bit 6 – SOSCPWR Secondary Oscillator Power Mode Select bit

| Value | Description |
|-------|---|
| 1 | Secondary oscillator operating in High-Power mode |
| 0 | Secondary oscillator operating in Low-Power mode |

Bit 4 – ORDY Oscillator Ready bit (read-only)

| Value | Description |
|-------|--|
| 1 | OSCCON1 = OSCCON2; the current system clock is the clock specified by NOSC |
| 0 | A clock switch is in progress |

Bit 3 – NOSCR New Oscillator is Ready bit (read-only)⁽¹⁾

| Value | Description |
|-------|---|
| 1 | A clock switch is in progress and the oscillator selected by NOSC indicates a ready condition |
| 0 | A clock switch is not in progress, or the NOSC-selected oscillator is not yet ready |

Note:

1. If CSWHOLD = 0, the user may not see this bit set because the bit is set for less than one instruction cycle.

8.6.4 OSCSTAT

Name: OSCSTAT
Address: 0x890

Oscillator Status Register 1

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|-------|------|------|------|-----|------|---|------|
| | EXTOR | HFOR | MFOR | LFOR | SOR | ADOR | | PLLR |
| Access | RO | RO | RO | RO | RO | RO | | RO |
| Reset | q | q | q | q | q | q | | q |

Bit 7 – EXTOR EXTOSC (external) Oscillator Ready bit

| Value | Description |
|-------|---|
| 1 | The oscillator is ready to be used |
| 0 | The oscillator is not enabled, or is not yet ready to be used |

Bit 6 – HFOR HFINTOSC Oscillator Ready bit

| Value | Description |
|-------|---|
| 1 | The oscillator is ready to be used |
| 0 | The oscillator is not enabled, or is not yet ready to be used |

Bit 5 – MFOR MFINTOSC Oscillator Ready bit

| Value | Description |
|-------|---|
| 1 | The oscillator is ready to be used |
| 0 | The oscillator is not enabled, or is not yet ready to be used |

Bit 4 – LFOR LFINTOSC Oscillator Ready bit

| Value | Description |
|-------|---|
| 1 | The oscillator is ready to be used |
| 0 | The oscillator is not enabled, or is not yet ready to be used |

Bit 3 – SOR Secondary (Timer1) Oscillator Ready bit

| Value | Description |
|-------|---|
| 1 | The oscillator is ready to be used |
| 0 | The oscillator is not enabled, or is not yet ready to be used |

Bit 2 – ADOR ADC Oscillator Ready bit

| Value | Description |
|-------|---|
| 1 | The oscillator is ready to be used |
| 0 | The oscillator is not enabled, or is not yet ready to be used |

Bit 0 – PLLR PLL Ready bit

| Value | Description |
|-------|--|
| 1 | The PLL is ready to be used |
| 0 | The PLL is not enabled, the required input source is not ready, or the PLL is not locked |

8.6.5 OSCEN

Name: OSCEN
Address: 0x891

Oscillator Manual Enable Register

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|--------|-------|-------|-------|--------|-------|---|---|
| | EXTOEN | HFOEN | MFOEN | LFOEN | SOSCEN | ADOEN | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | | |

Bit 7 – EXTOEN External Oscillator Manual Request Enable bit

| Value | Description |
|-------|--|
| 1 | EXTOSC is explicitly enabled, operating as specified by CONFIG1[FEXTOSC] |
| 0 | EXTOSC is only enabled if requested by a peripheral |

Bit 6 – HFOEN HFINTOSC Oscillator Manual Request Enable bit

| Value | Description |
|-------|--|
| 1 | HFINTOSC is explicitly enabled, operating as specified by OSCFRQ |
| 0 | HFINTOSC is only enabled if requested by a peripheral |

Bit 5 – MFOEN MFINTOSC (500 kHz/31.25 kHz) Oscillator Manual Request Enable bit (Derived from HFINTOSC)

| Value | Description |
|-------|---|
| 1 | MFINTOSC is explicitly enabled |
| 0 | MFINTOSC is only enabled if requested by a peripheral |

Bit 4 – LFOEN LFINTOSC (31 kHz) Oscillator Manual Request Enable bit

| Value | Description |
|-------|---|
| 1 | LFINTOSC is explicitly enabled |
| 0 | LFINTOSC is only enabled if requested by a peripheral |

Bit 3 – SOSCEN Secondary Oscillator Manual Request Enable bit

| Value | Description |
|-------|---|
| 1 | Secondary Oscillator is explicitly enabled, operating as specified by SOSCPWR |
| 0 | Secondary Oscillator is only enabled if requested by a peripheral |

Bit 2 – ADOEN ADC Oscillator Manual Request Enable bit

| Value | Description |
|-------|---|
| 1 | ADC oscillator is explicitly enabled |
| 0 | ADC oscillator is only enabled if requested by a peripheral |

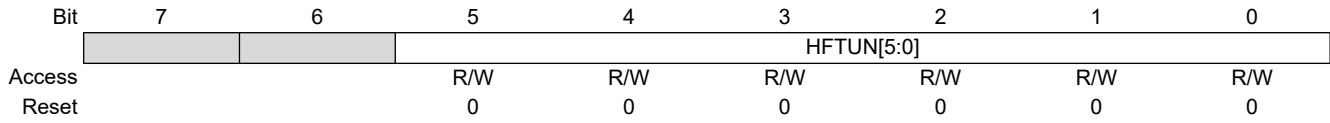
Related Links

[4.7.1 CONFIG1](#)

8.6.6 OSCTUNE

Name: OSCTUNE
 Address: 0x892

HFINTOSC Tuning Register



Bits 5:0 – HFTUN[5:0] HFINTOSC Frequency Tuning bits

| Value | Description |
|---------|---|
| 01 1111 | Maximum frequency |
| 00 0000 | Center frequency. Oscillator module is running at the calibrated frequency (default value). |
| 10 0000 | Minimum frequency |

8.6.7 OSCFRQ

Name: OSCFRQ
Address: 0x893

HFINTOSC Frequency Selection Register



Bits 2:0 – HFFRQ[2:0] HFINTOSC Frequency Selection bits

| FRQ[2:0] | Nominal Frequency (MHz) (NOSC = 110) | 2x PLL Frequency (MHz) (NOSC = 001) |
|----------|---|--|
| 111 | Reserved | Reserved |
| 110 | 32 | |
| 101 | 16 | 32 |
| 100 | 12 | 24 |
| 011 | 8 | 16 |
| 010 | 4 | Reserved |
| 001 | 2 | |
| 000 | 1 | |

Note:

1. When RSTOSC = 110 (HFINTOSC 1 MHz), the FRQ bits will default to '010' upon Reset; when RSTOSC = 001 (HFINTOSC 32 MHz), the FRQ bits will default to '101' upon Reset.

9. REFCLK - Reference Clock Output Module

The reference clock output (REFCLK) module provides the ability to send a clock signal to the clock reference output pin (CLKR). The reference clock output can also be routed internally as a signal for other peripherals, such as the Data Signal Modulator (DSM), memory scanner, and timer module.

The reference clock output module has the following features:

- Selectable Clock Source Using the CLKRCLK Register
- Programmable Clock Divider
- Selectable Duty Cycle

Figure 9-1. Clock Reference Block Diagram

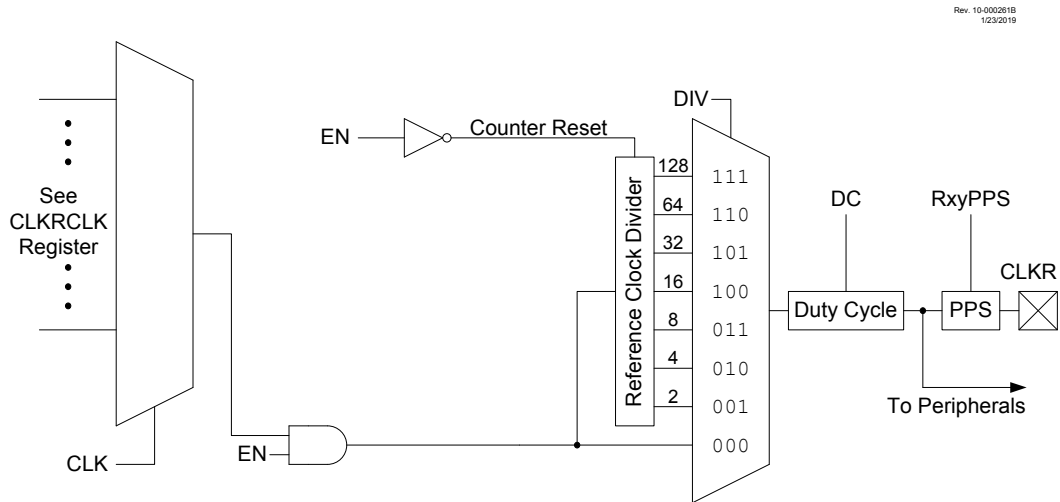
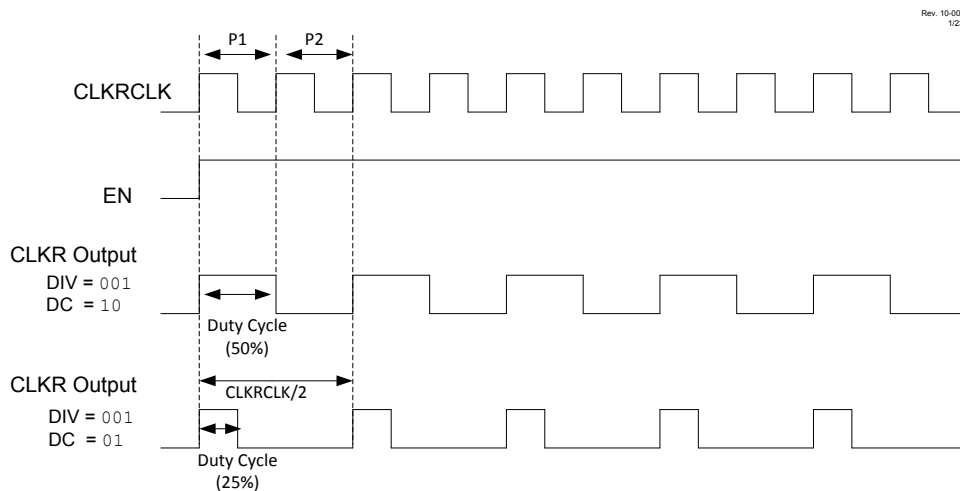


Figure 9-2. Clock Reference Timing



9.1 Clock Source

The clock source of the reference clock peripheral is selected with the **CLK** bits in the CLKRCLK register. The available clock sources are listed below:

- CLCs

- NCO
- Secondary oscillator
- MFINTOSC
- LFINTOSC
- F_{OSC}

9.1.1 Clock Synchronization

The CLKR output signal is ensured to be glitch-free when the EN bit in the CLKRCON register is set to start the module and enable the CLKR output.

When the reference clock output is disabled, the output signal will be disabled immediately.

Clock dividers and clock duty cycles can be changed while the module is enabled but doing so may cause glitches to occur on the output. To avoid possible glitches, clock dividers and clock duty cycles should be changed only when the EN bit is clear.

9.2 Programmable Clock Divider

The module takes the clock input and divides it based on the value of the DIV bits.

The following configurations are available:

- Base F_{OSC} value
- F_{OSC} divided by 2
- F_{OSC} divided by 4
- F_{OSC} divided by 8
- F_{OSC} divided by 16
- F_{OSC} divided by 32
- F_{OSC} divided by 64
- F_{OSC} divided by 128

The clock divider values can be changed while the module is enabled. However, in order to prevent glitches on the output, the DIV bits should only be changed when the module is disabled (EN = 0).

9.3 Selectable Duty Cycle

The DC bits in the CLKRCON register are used to modify the duty cycle of the output clock. A duty cycle of 0%, 25%, 50%, or 75% can be selected for all clock rates when the DIV value is not 000. When DIV = 000 the duty cycle defaults to 50% for all values of DC except 00, in which case the duty cycle is 0% (constant low output).



Important: The DC value at Reset is 10. This makes the default duty cycle 50% and not 0%.



Important: Clock dividers and clock duty cycles can be changed while the module is enabled but doing so may cause glitches to occur on the output. To avoid possible glitches, clock dividers and clock duty cycles should be changed only when the module is disabled (EN = 0).

9.4 Operation in Sleep Mode

The reference clock module continues to operate and provide a signal output in Sleep for all clock source selections except F_{OSC} (CLK=0).

9.5 Register Summary: Reference CLK

| Address | Name | Bit Pos. | | | | | | | |
|-------------|-------------------------|----------|----|--|--|---------|--|----------|--|
| 0x00 ... | Reserved | | | | | | | | |
| 0x0894 | | | | | | | | | |
| 0x0895 | CLKRCON | 7:0 | EN | | | DC[1:0] | | DIV[2:0] | |
| 0x0896 | CLKRCLK | 7:0 | | | | | | CLK[3:0] | |

9.6 Register Definitions: Reference Clock

Related Links

[1.4.2.2 Long Bit Names](#)

9.6.1 CLKRCON

Name: CLKRCON

Address: 0x895

Reference Clock Control Register

| | | | | | | | | |
|--------|-----|---|---|---------|-----|-----|----------|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | EN | | | DC[1:0] | | | DIV[2:0] | |
| Access | R/W | | | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | | | 1 | 0 | 0 | 0 | 0 |

Bit 7 – EN

Reference Clock Module Enable bit

| Value | Description |
|-------|------------------------------------|
| 1 | Reference clock module enabled |
| 0 | Reference clock module is disabled |

Bits 4:3 – DC[1:0]

Reference Clock Duty Cycle bits⁽¹⁾

| Value | Description |
|-------|---------------------------------|
| 11 | Clock outputs duty cycle of 75% |
| 10 | Clock outputs duty cycle of 50% |
| 01 | Clock outputs duty cycle of 25% |
| 00 | Clock outputs duty cycle of 0% |

Bits 2:0 – DIV[2:0]

Reference Clock Divider bits

| Value | Description |
|-------|---------------------------------|
| 111 | Base clock value divided by 128 |
| 110 | Base clock value divided by 64 |
| 101 | Base clock value divided by 32 |
| 100 | Base clock value divided by 16 |
| 011 | Base clock value divided by 8 |
| 010 | Base clock value divided by 4 |
| 001 | Base clock value divided by 2 |
| 000 | Base clock value |

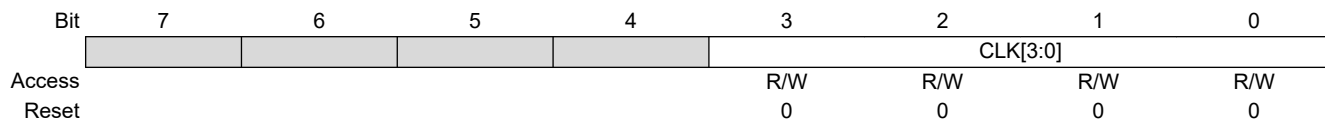
Note:

- Bits are valid for reference clock divider values of two or larger, the base clock cannot be further divided.

9.6.2 CLKRCLK

Name: CLKRCLK
Address: 0x896

Clock Reference Clock Selection MUX



Bits 3:0 – CLK[3:0] CLKR Clock Selection bits

Table 9-1. CLKR Clock Sources

| CLK | Clock Source |
|-----------|--------------------|
| 1111-1011 | Reserved |
| 1010 | CLC4 OUT |
| 1001 | CLC3 OUT |
| 1000 | CLC2 OUT |
| 0111 | CLC1 OUT |
| 0110 | NCO1 OUT |
| 0101 | SOSC |
| 0100 | MFINTOSC (32 kHz) |
| 0011 | MFINTOSC (500 kHz) |
| 0010 | LFINTOSC |
| 0001 | HFINTOSC (32 MHz) |
| 0000 | F _{Osc} |

10. Resets

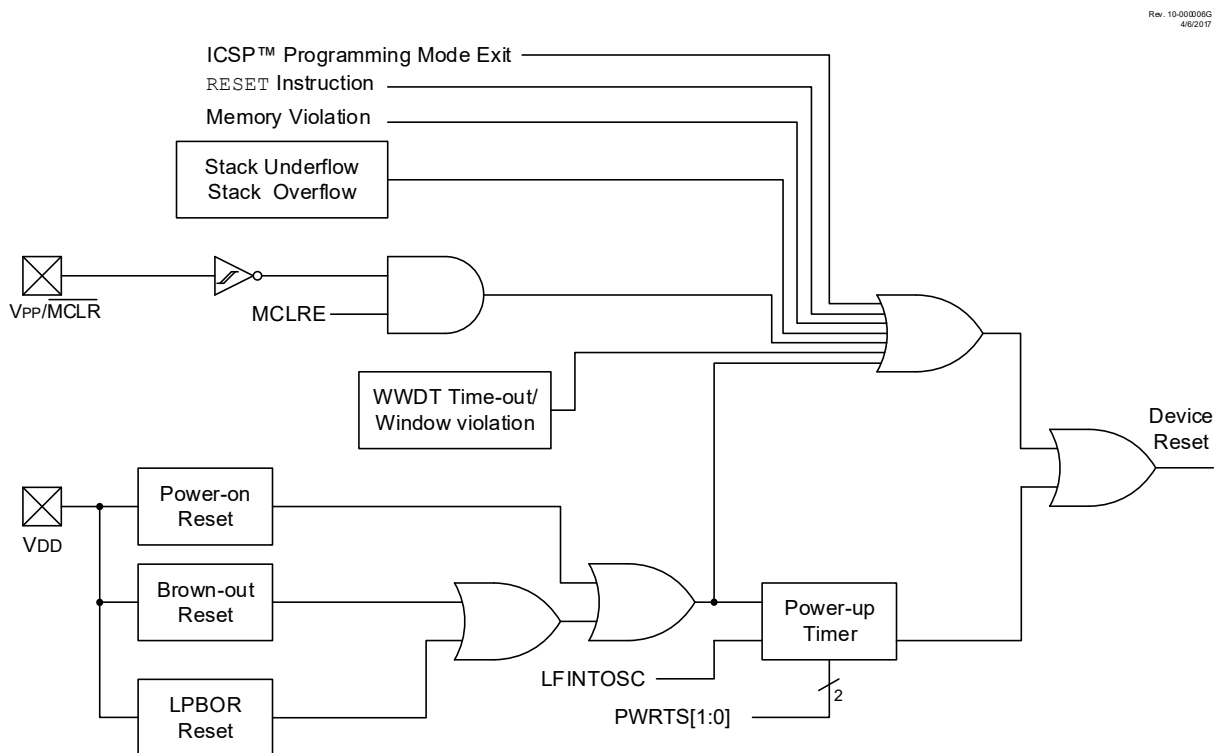
There are multiple ways to reset this device:

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

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

A simplified block diagram of the On-Chip Reset Circuit is shown in the block diagram below.

Figure 10-1. Simplified Block Diagram of On-Chip Reset Circuit



Note: See “BOR Operating Conditions” table for BOR active conditions.

Related Links

[10.2.3 BOR Controlled by Software](#)

10.1 Power-on Reset (POR)

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

Related Links

[10.2.3 BOR Controlled by Software](#)

10.2 Brown-out Reset (BOR)

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

The Brown-out Reset module has four operating modes controlled by the BOREN[1:0] bits in [Configuration Word 2](#). The four operating modes are:

- BOR is always ON
- BOR is OFF when in Sleep
- BOR is controlled by software
- BOR is always OFF

Refer to [BOR Operating Conditions](#) for more information.

The Brown-out Reset voltage level is selectable by configuring the BORV[1:0] bits in [Configuration Word 2](#).

A V_{DD} noise rejection filter prevents the BOR from triggering on small events. If V_{DD} falls below V_{BOR} for a duration greater than parameter T_{BORDC} , the device will reset and the BOR bit in the PCON0 register will be cleared, indicating the Brown-out Reset condition occurred. See [Figure 10-2](#).

10.2.1 BOR is Always ON

When the BOREN bits of [Configuration Word 2](#) are programmed to '11', the BOR is always ON. The device start-up will be delayed until the BOR is ready and V_{DD} is higher than the BOR threshold.

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

10.2.2 BOR is OFF in Sleep

When the BOREN bits of [Configuration Word 2](#) are programmed to '10', the BOR is on, except in Sleep. BOR protection is not active during Sleep, but device wake-up will be delayed until the BOR can determine that V_{DD} is higher than the BOR threshold. The device wake-up will be delayed until the BOR is ready.

10.2.3 BOR Controlled by Software

When the BOREN bits of [Configuration Word 2](#) are programmed to '01', the BOR is controlled by the SBOREN bit. The device wake from Sleep is not delayed by the BOR ready condition or the V_{DD} level only when the SBOREN bit is cleared in software and the device is starting up from a non POR/BOR Reset event.

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

BOR protection is unchanged by Sleep.

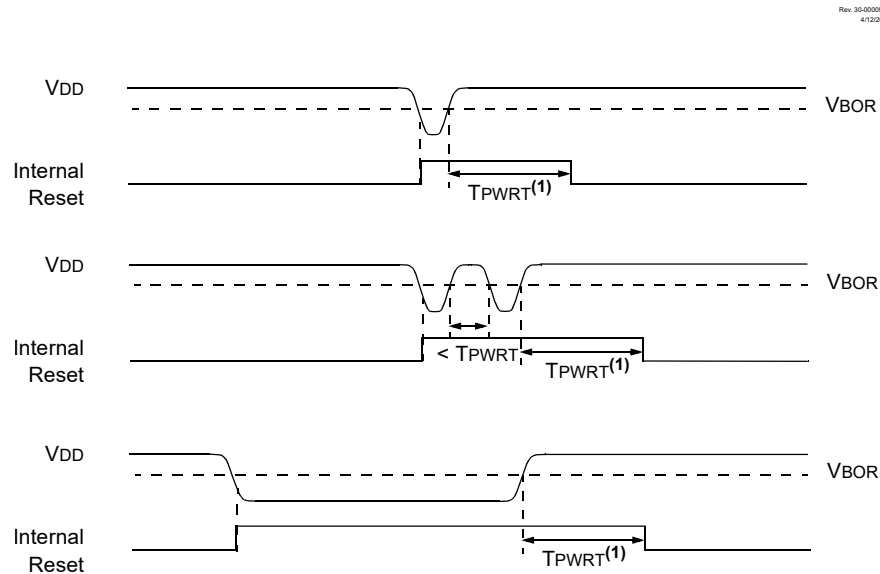
Table 10-1. BOR Operating Conditions

| BOREN[1:0] | SBOREN | Device Mode | BOR Mode | Instruction Execution upon: Release of POR or Wake-up from Sleep |
|------------|--------|-------------|----------|---|
| 11 | X | X | Active | Waits for release of BOR ⁽¹⁾ (BORRDY = 1) |
| 10 | X | Awake | Active | Waits for release of BOR (BORRDY = 1) Waits for BOR Reset release |
| | | Sleep | Disabled | |

.....continued

| BOREN[1:0] | SBOREN | Device Mode | BOR Mode | Instruction Execution upon: Release of POR or Wake-up from Sleep |
|------------|--------|-------------|----------|--|
| 01 | 1 | X | Active | Waits for BOR Reset release (BORRDY = 1) |
| | 0 | X | Disabled | Begins immediately (BORRDY = x) |
| 00 | X | X | Disabled | |

Figure 10-2. Brown-out Situations



Note: T_{PWRT} delay only if PWRTS bit field is programmed to a value different from '11'.

10.2.4 BOR is Always OFF

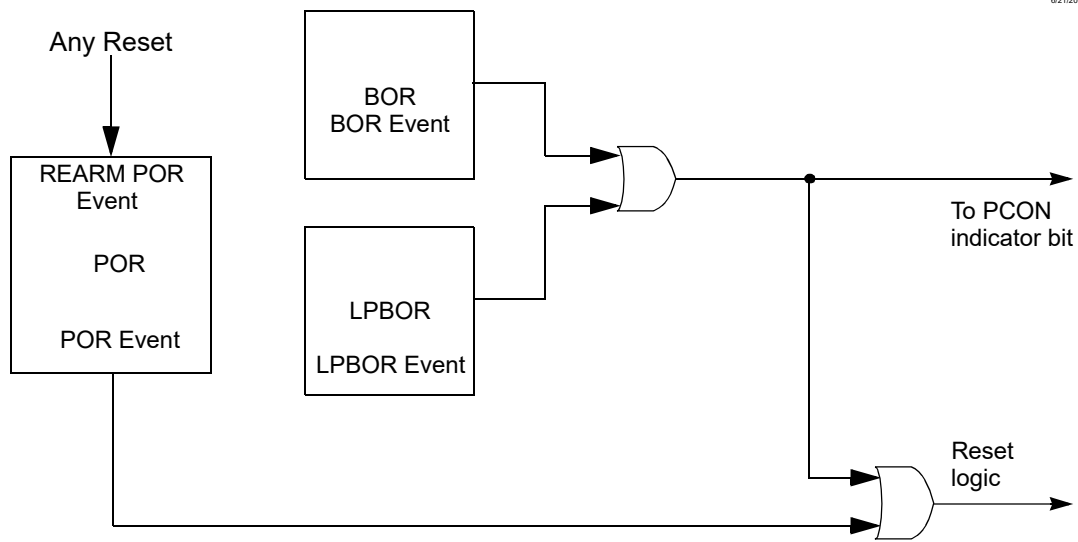
When the BOREN bits of the Configuration Word 2 are programmed to '00', the BOR is always disabled. In the configuration, setting the SWBOREN bit will have no affect on BOR operations.

10.3 Low-Power Brown-out Reset (LPBOR)

The Low-Power Brown-Out Reset (LPBOR) circuit provides alternative protection against Brown-out conditions. When V_{DD} falls below the LPBOR threshold, the device is held in Reset. When this occurs, the BOR bit of the PCON0 register is cleared to indicate that a Brown-out Reset occurred. The BOR bit will be cleared when either the BOR or the LPBOR circuitry detects a BOR condition. The LPBOR feature can be used with or without BOR enabled. When used while BOR is enabled, the LPBOR can be used as a secondary protection circuit in case the BOR circuit fails to detect the BOR condition. Additionally, when BOR is enabled except while in Sleep (BOREN[1:0] = 10), the LPBOR circuit will hold the device in Reset while V_{DD} is lower than the LPBOR threshold, and will also re-arm the POR. (see 40.4.5 Reset, WDT, Oscillator Start-up Timer, Power-up Timer, Brown-Out Reset and Low-Power Brown-Out Reset Specifications for LPBOR Reset voltage levels).

When used without BOR enabled, the LPBOR circuit provides a single Reset trip point with the benefit of reduced current consumption.

Figure 10-3. LPBOR, BOR, POR Relationship



10.3.1 Enabling LPBOR

The LPBOR is controlled by the $\overline{\text{LPBOREN}}$ bit of [Configuration Word 2](#). When the device is erased, the LPBOR module defaults to disabled.

Related Links

[4.7.2 CONFIG2](#)

10.3.2 LPBOR Module Output

The output of the LPBOR module is a signal indicating whether or not a Reset is to be asserted. This signal is OR'd together with the Reset signal of the BOR module to provide the generic $\overline{\text{BOR}}$ signal, which goes to the [PCON0](#) register and to the power control block.

10.4 $\overline{\text{MCLR}}$ Reset

The $\overline{\text{MCLR}}$ is an optional external input that can reset the device. The $\overline{\text{MCLR}}$ function is controlled by the MCLRE bit of [Configuration Word 2](#) and the LVP bit of [Configuration Word 4](#) (see table below). The $\overline{\text{RMCLR}}$ bit in the [PCON0](#) register will be set to '0' if a $\overline{\text{MCLR}}$ has occurred.

Table 10-2. $\overline{\text{MCLR}}$ Configuration

| MCLRE | LVP | $\overline{\text{MCLR}}$ |
|-------|-----|--------------------------|
| x | 1 | Enabled |
| 1 | 0 | Enabled |
| 0 | 0 | Disabled |

10.4.1 $\overline{\text{MCLR}}$ Enabled

When $\overline{\text{MCLR}}$ is enabled and the pin is held low, the device is held in Reset. The $\overline{\text{MCLR}}$ pin is connected to V_{DD} through an internal weak pull-up.

The device has a noise filter in the $\overline{\text{MCLR}}$ Reset path. The filter will detect and ignore small pulses.



Important: An internal Reset event (`RESET` instruction, BOR, WWDT, POR, STKOVF, STKUNF) does not drive the `MCLR` pin low.

Related Links

[2.3 Master Clear \(MCLR\) Pin](#)

10.4.2 `MCLR` Disabled

When `MCLR` is disabled, the `MCLR` becomes input-only and pin functions such as internal weak pull-ups are under software control.

Related Links

[14.3 I/O Priorities](#)

10.5 Windowed Watchdog Timer (WWDT) Reset

The Windowed Watchdog Timer generates a Reset if the firmware does not issue a `CLRWDT` instruction within the time-out period or window set. The `TO` and `PD` bits in the `STATUS` register and the `RWDT` bit are changed to indicate a WDT Reset. The `WDTWV` bit indicates if the WDT Reset has occurred due to a time-out or a window violation.

Related Links

[5.8.4 STATUS](#)

[11. WWDT - Windowed Watchdog Timer](#)

10.6 `RESET` Instruction

A `RESET` instruction will cause a device Reset. The `RI` bit will be set to '0'. See [Reset Status Bits and Their Significance](#) for default conditions after a `RESET` instruction has occurred.

10.7 Stack Overflow/Underflow Reset

The device can reset when the Stack Overflows or Underflows. The `STKOVF` or `STKUNF` bits in `PCON0` register indicate the Reset condition. These Resets are enabled by setting the `STVREN` bit in [Configuration Word 2](#).

Related Links

[4.7.2 CONFIG2](#)

10.8 Programming Mode Exit

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

10.9 Power-up Timer (PWRT)

The Power-up Timer provides a nominal 64 ms time-out on POR or Brown-out Reset. The device is held in Reset as long as `PWRT` is active. The `PWRT` delay allows additional time for the V_{DD} to rise to an acceptable level.

The Power-up Timer is controlled by the `PWRTS` bit field of the [Configuration Word 2](#).

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

For additional information, refer to Application Note AN607, "Power-up Trouble Shooting" (DS00000607).

10.10 Start-up Sequence

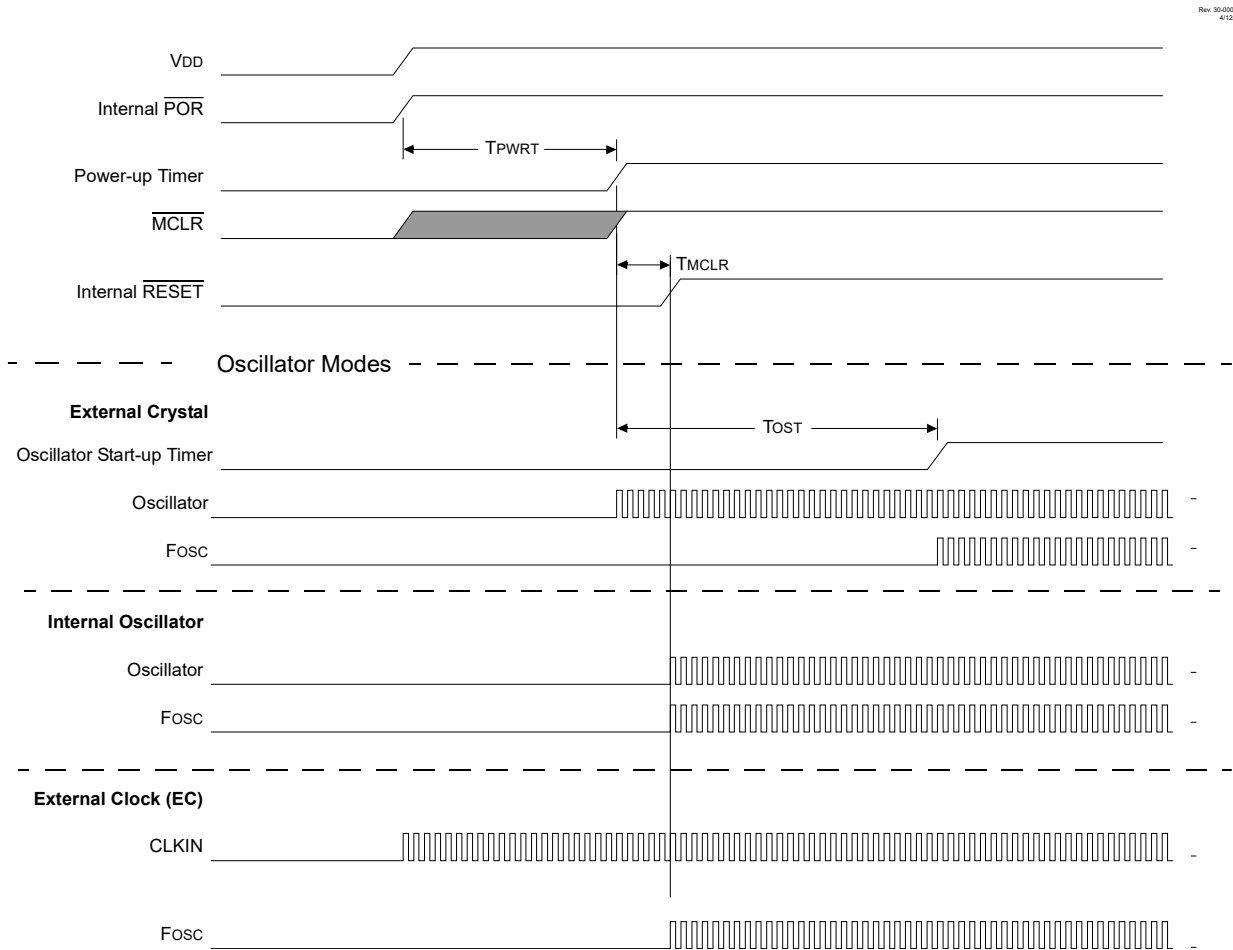
Upon the release of a POR or BOR, the following must occur before the device will begin executing:

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

The total time-out will vary based on oscillator configuration and Power-up Timer configuration.

The Power-up Timer and oscillator start-up timer run independently of MCLR Reset. If MCLR is kept low long enough, the Power-up Timer and oscillator Start-up Timer will expire. Upon bringing MCLR high, the device will begin execution after ten F_{OSC} cycles (see Figure 10-4). This is useful for testing purposes or for synchronizing more than one device operating in parallel.

Figure 10-4. Reset Start-up Sequence



Related Links

[8. OSC - Oscillator Module](#)

10.11 Memory Execution Violation

A Memory Execution Violation Reset occurs if executing an instruction being fetched from outside the valid execution area. The different valid execution areas are defined as follows:

- Flash Memory: The “Device Sizes and Addresses” table shows the addresses available on the PIC16(L)F18426/46 devices based on user Flash size. Execution outside this region generates a memory execution violation.
- Storage Area Flash (SAF): If Storage Area Flash (SAF) is enabled, the SAF area is not a valid execution area.

Prefetched instructions that are not executed do not cause memory execution violations. For example, a `GOTO` instruction in the last memory location will prefetch from an invalid location; this is not an error. If an instruction from an invalid location tries to execute, the memory violation is generated immediately, and any concurrent interrupt requests are ignored. When a memory execution violation is generated, the device is reset and flag `MEMV` is cleared in `PCON1` to signal the cause. The flag needs to be set in code after a memory execution violation.

Related Links

[5.1 Program Memory Organization](#)

[5.2.3 Storage Area Flash](#)

[5.2.5 Memory Violation](#)

10.12 Determining the Cause of a Reset

Upon any Reset, multiple bits in the `STATUS` and `PCON0` registers are updated to indicate the cause of the Reset. The following tables show the Reset conditions of these registers.

Table 10-3. Reset Status Bits and Their Significance

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

Table 10-4. Reset Condition for Special Registers

| Condition | Program Counter | STATUS Register | PCON0 Register | PCON1 Register |
|---|-----------------|-----------------|----------------|----------------|
| Power-on Reset | 0 | ---1 1000 | 0011 110x | ---- --1- |
| Brown-out Reset | 0 | ---1 1000 | 0011 11u0 | ---- --u- |
| \overline{MCLR} Reset during normal operation | 0 | -uuu uuuu | uuuu 0uuu | ---- --1- |
| \overline{MCLR} Reset during Sleep | 0 | ---1 0uuu | uuuu 0uuu | ---- --u- |
| WWDT Time-out Reset | 0 | ---0 uuuu | uuu0 uuuu | ---- --u- |

|continued | | | | |
|--|-----------------------|-----------------|----------------|----------------|
| Condition | Program Counter | STATUS Register | PCON0 Register | PCON1 Register |
| WWDT Wake-up from Sleep | PC + 1 | ---0 0uuu | uuuu uuuu | ---- --u- |
| WWDT Window Violation Reset | 0 | ---u uuuu | uu0u uuuu | ---- --u- |
| Interrupt Wake-up from Sleep | PC + 1 ⁽¹⁾ | ---1 0uuu | uuuu uuuu | ---- --u- |
| RESET Instruction Executed | 0 | ---u uuuu | uuuu u0uu | ---- --u- |
| Stack Overflow Reset (STVREN = 1) | 0 | ---u uuuu | 1uuu uuuu | ---- --u- |
| Stack Underflow Reset (STVREN = 1) | 0 | ---u uuuu | u1uu uuuu | ---- --u- |
| Memory Violation Reset ($\overline{\text{MEMV}}$ = 0) | 0 | -uuu uuuu | uuuu uuuu | ---- --0- |

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

Note:

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

Related Links

[5.8.4 STATUS](#)

10.13 Register Summary - BOR Control and Power Control

| Address | Name | Bit Pos. | | | | | | | | |
|-------------|----------|----------|--------|--------|-------|------|-------|----|------|--------|
| 0x00 ... | Reserved | | | | | | | | | |
| 0x0810 | | | | | | | | | | |
| 0x0811 | BORCON | 7:0 | SBOREN | | | | | | | BORRDY |
| 0x0812 | Reserved | | | | | | | | | |
| 0x0813 | PCON0 | 7:0 | STKOVF | STKUNF | WDTWV | RWDT | RMCLR | RI | POR | BOR |
| 0x0814 | PCON1 | 7:0 | | | | | | | MEMV | |

10.14 Register Definitions: Power Control

10.14.1 BORCON

Name: BORCON

Address: 0x811

Brown-out Reset Control Register

| | | | | | | | | |
|--------|--------|---|---|---|---|---|---|--------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | SBOREN | | | | | | | BORRDY |
| Access | R/W | | | | | | | R |
| Reset | 1 | | | | | | | q |

Bit 7 – SBOREN Software Brown-out Reset Enable bit

Reset States: POR/BOR = 1

All Other Resets = u

| Value | Condition | Description |
|-------|---------------|---|
| – | If BOREN ≠ 01 | SBOREN is read/write, but has no effect on the BOR. |
| 1 | If BOREN = 01 | BOR Enabled |
| 0 | If BOREN = 01 | BOR Disabled |

Bit 0 – BORRDY Brown-out Reset Circuit Ready Status bit

Reset States: POR/BOR = q

All Other Resets = u

| Value | Description |
|-------|--|
| 1 | The Brown-out Reset Circuit is active and armed |
| 0 | The Brown-out Reset Circuit is disabled or is warming up |

Related Links

[4.7.2 CONFIG2](#)

10.14.2 PCON0

Name: PCON0

Address: 0x813

Power Control Register 0

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| | STKOVF | STKUNF | WDTWV | RWDT | RMCLR | RI | POR | BOR |
| Access | R/W/HS | R/W/HS | R/W/HC | R/W/HC | R/W/HC | R/W/HC | R/W/HC | R/W/HC |
| Reset | 0 | 0 | 1 | 1 | 1 | 1 | 0 | q |

Bit 7 – STKOVF Stack Overflow Flag bit

Reset States: POR/BOR = 0

All Other Resets = q

| Value | Description |
|-------|--|
| 1 | A Stack Overflow occurred (more CALLs than fit on the stack) |
| 0 | A Stack Overflow has not occurred or set to '0' by firmware |

Bit 6 – STKUNF Stack Underflow Flag bit

Reset States: POR/BOR = 0

All Other Resets = q

| Value | Description |
|-------|--|
| 1 | A Stack Underflow occurred (more RETURNS than CALLs) |
| 0 | A Stack Underflow has not occurred or set to '0' by firmware |

Bit 5 – WDTWV Watchdog Window Violation Flag bit

Reset States: POR/BOR = 1

All Other Resets = q

| Value | Description |
|-------|--|
| 1 | A WDT window violation has not occurred or set to '1' by firmware |
| 0 | A CLRWDT instruction was issued when the WDT Reset window was closed (set to '0' in hardware when a WDT window violation Reset occurs) |

Bit 4 – RWDT WDT Reset Flag bit

Reset States: POR/BOR = 1

All Other Resets = q

| Value | Description |
|-------|---|
| 1 | A WDT overflow/time-out Reset has not occurred or set to '1' by firmware |
| 0 | A WDT overflow/time-out Reset has occurred (set to '0' in hardware when a WDT Reset occurs) |

Bit 3 – RMCLR MCLR Reset Flag bit

Reset States: POR/BOR = 1

All Other Resets = q

| Value | Description |
|-------|---|
| 1 | A MCLR Reset has not occurred or set to '1' by firmware |
| 0 | A MCLR Reset has occurred (set to '0' in hardware when a MCLR Reset occurs) |

Bit 2 – RI RESET Instruction Flag bit

Reset States: POR/BOR = 1

All Other Resets = q

| Value | Description |
|-------|---|
| 1 | A RESET instruction has not been executed or set to '1' by firmware |
| 0 | A RESET instruction has been executed (set to '0' in hardware upon executing a RESET instruction) |

Bit 1 – POR Power-on Reset Status bit

Reset States: POR/BOR = 0

All Other Resets = u

| Value | Description |
|-------|---|
| 1 | No Power-on Reset occurred or set to '1' by firmware |
| 0 | A Power-on Reset occurred (set to '0' in hardware when a Power-on Reset occurs) |

Bit 0 – $\overline{\text{BOR}}$ Brown-out Reset Status bit

Reset States: POR/BOR = q

All Other Resets = u

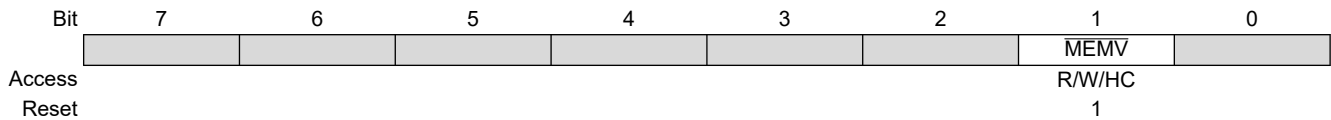
| Value | Description |
|-------|---|
| 1 | No Brown-out Reset occurred or set to '1' by firmware |
| 0 | A Brown-out Reset occurred (set to '0' in hardware when a Brown-out Reset occurs) |

10.14.3 PCON1

Name: PCON1

Address: 0x814

Power Control Register 1



Bit 1 – MEMV Memory Violation Flag bit

Reset States: POR/BOR = 1

All Other Resets = u

| Value | Description |
|-------|---|
| 1 | No Memory Violation Reset occurred or set to '1' by firmware. |
| 0 | A Memory Violation Reset occurred (set to '0' in hardware when a Memory Violation occurs) |

11. WWDT - Windowed Watchdog Timer

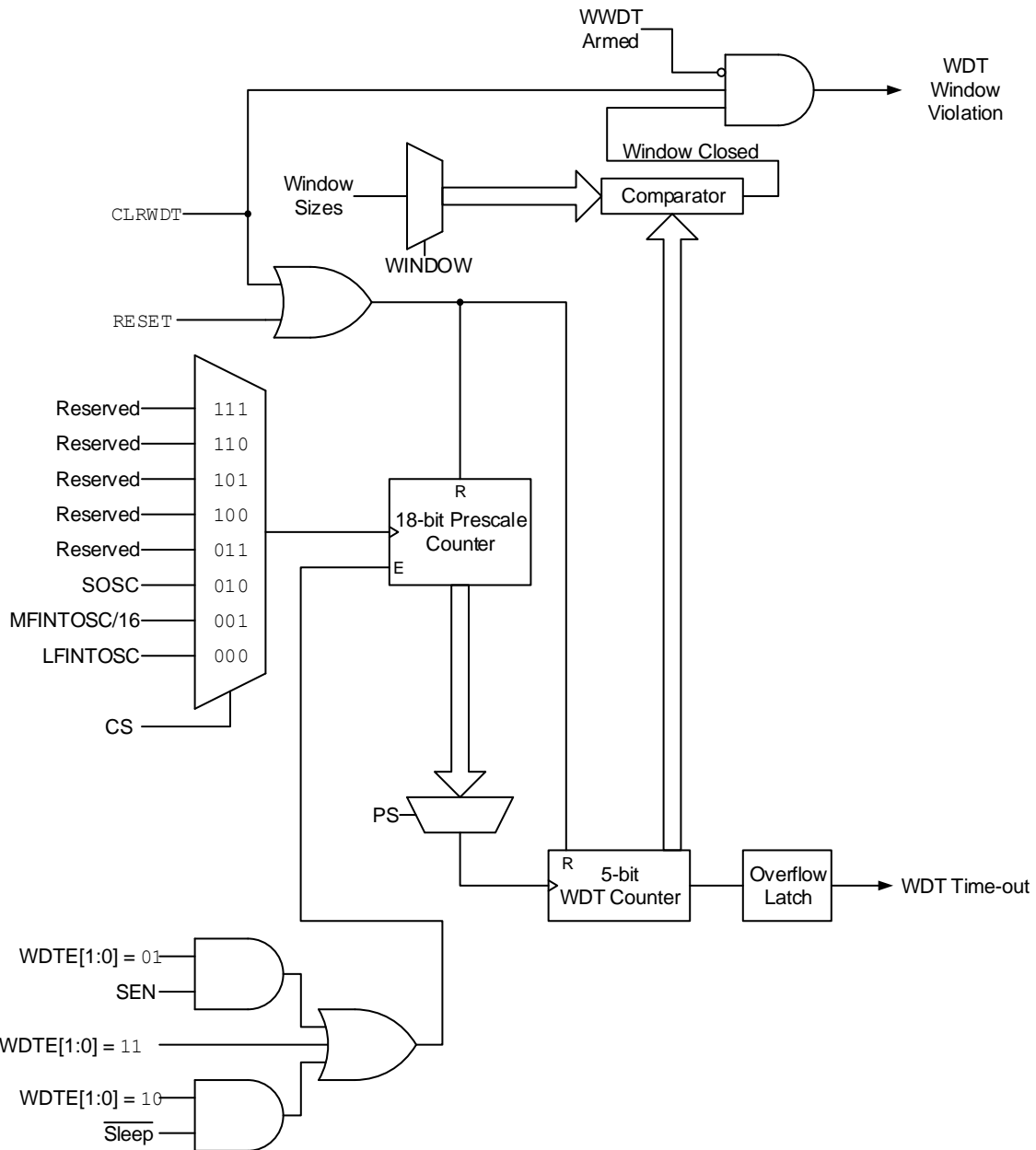
The Watchdog Timer (WDT) is a system timer that generates a Reset if the firmware does not issue a `CLRWDT` instruction within the time-out period. The Watchdog Timer is typically used to recover the system from unexpected events. The Windowed Watchdog Timer (WWDT) differs in that `CLRWDT` instructions are only accepted when they are performed within a specific window during the time-out period.

The WWDT has the following features:

- Selectable Clock Source
- Multiple Operating Modes
 - WWDT is always on
 - WWDT is off when in Sleep
 - WWDT is controlled by software
 - WWDT is always off
- Configurable Time-out Period is from 1 ms to 256s (Nominal)
- Configurable Window Size from 12.5% to 100% of the Time-out Period
- Multiple Reset Conditions

Figure 11-1. Windowed Watchdog Timer Block Diagram

Rev. 10-000 162A
 1/2/2014



11.1 Independent Clock Source

The WWDT can derive its time base from either the 31 kHz LFINTOSC or 31.25 kHz MFINTOSC internal oscillators, depending on the value of WDTE bits in the CONFIG3.

If $WDTE = 'b1x$, then the clock source will be enabled depending on the WDTCCS bits in the CONFIG3.

If $WDTE = 'b01$, the **SEN** bit in the WDTCON0 register should be set by software to enable WWDT, and the clock source is enabled by the **WDTCS** bits in the WDTCON1 register.

Time intervals in this chapter are based on a minimum nominal interval of 1 ms. See [40.4.2 Internal Oscillator Parameters\(1\)](#) section for LFINTOSC and MFINTOSC tolerances.

Related Links

- [4.7.3 CONFIG3](#)
- [40.4.2 Internal Oscillator Parameters\(1\)](#)
- [40. Electrical Specifications](#)

11.2 WWDT Operating Modes

The Windowed Watchdog Timer module has four operating modes controlled by the WDTE bits in the CONFIG3. See [Table 11-1](#).

Table 11-1. WWDT Operating Modes

| WDTE[1:0] | SEN | Device Mode | WWDT Mode |
|-----------|-----|-------------|-----------|
| 11 | X | x | Active |
| 10 | X | Awake | Active |
| | | Sleep | Disabled |
| 01 | 1 | x | Active |
| | 0 | x | Disabled |
| 00 | X | x | Disabled |

11.3 Time-out Period

If the WDTCPs bits in CONFIG3 are set to 0'b111111, then the [WDTPS](#) bits set the time-out period from 1 ms to 256 seconds (nominal). If any value other than the default value is assigned to WDTCPs Configuration bits, then the timer period will be based on the WDTCPs bits in the CONFIG3 register. After a Reset, the default time-out period is 2s.

Related Links

- [4.7.3 CONFIG3](#)

11.4 Watchdog Window

The Windowed Watchdog Timer has an optional Windowed mode that is controlled by the WDTcWS bits in Configuration Word 3 and [WINDOW](#) bits in the WDTCON1 register. In the Windowed mode, the CLRWDT instruction must occur within the allowed window of the WDT period. Any CLRWDT instruction that occurs outside of this window will trigger a window violation and will cause a WWDT Reset, similar to a WWDT time-out. See [Figure 11-2](#) for an example.

The window size is controlled by the Watchdog Timer Window Select ([WINDOW](#)) bits in WDTCON1, if WDTcWS bits in Configuration Word 3 are set to 111.

The WDTTMR bits in the [WDTTMR](#) register are used to determine whether the window is open, as defined by the WINDOW bits.

In the event of a window violation, a Reset will be generated and the $\overline{\text{WDTWV}}$ bit of the PCON0 register will be cleared. This bit is set by a POR or can be set in firmware.

Related Links

- [10.14.2 PCON0](#)

11.5 Clearing the WWDT

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

- Any Reset
- Valid CLRWDT instruction is executed
- Device enters Sleep
- Exit Sleep by Interrupt
- WWDT is disabled
- Oscillator Start-up Timer (OST) is running
- Any write to the WDTCON0 or 11.8.2 WDTCON1 registers

11.5.1 CLRWDT Considerations (Windowed Mode)

When in Windowed mode, the WWDT must be armed before a CLRWDT instruction will clear the timer. This is performed by reading the WDTCON0 register. Executing a CLRWDT instruction without performing such an arming action will trigger a window violation regardless of whether the window is open or not.

See [Table 11-2](#) for more information.

11.6 Operation During Sleep

When the device enters Sleep, the WWDT is cleared. If the WWDT is enabled during Sleep, the WWDT resumes counting. When the device exits Sleep, the WWDT is cleared again.

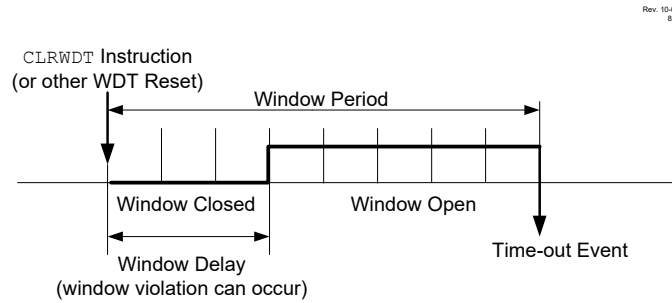
The WWDT remains clear until the Oscillator Start-up Timer (OST) completes, if enabled.

When a WWDT time-out occurs while the device is in Sleep, no Reset is generated. Instead, the device wakes up and resumes operation. The \overline{TO} and \overline{PD} bits in the STATUS register are changed to indicate the event. The \overline{RWDT} bit in the PCON0 register can also be used.

Table 11-2. WWDT Clearing Conditions

| Conditions | WWDT |
|---|------------------------------|
| WDTE = 00 | Cleared |
| WDTE = 01 and SEN = 0 | |
| WDTE = 10 and enter Sleep | |
| CLRWDT Command | |
| Oscillator Fail Detected | |
| Exit Sleep + System Clock = SOSC, EXTRC, INTOSC, EXTCLK | |
| Exit Sleep + System Clock = XT, HS, LP | Cleared until the end of OST |
| Change INTOSC divider (IRCF bits) | Unaffected |

Figure 11-2. Window Period and Delay



Related Links

[8.2.1.3 Oscillator Start-up Timer \(OST\)](#)

[5.8.4 STATUS](#)

[10.14.2 PCON0](#)

[5. Memory Organization](#)

11.7 Register Summary - WDT Control

| Address | Name | Bit Pos. | | | | | | | | | |
|-------------|----------|----------|--|-------------|------------|--|--|-------------|------------|-----|--|
| 0x00 ... | Reserved | | | | | | | | | | |
| 0x080B | | | | | | | | | | | |
| 0x080C | WDTCN0 | 7:0 | | | WDTPS[4:0] | | | | | SEN | |
| 0x080D | WDTCN1 | 7:0 | | WDTCS[2:0] | | | | WINDOW[2:0] | | | |
| 0x080E | WDTPSL | 7:0 | | PSCNTL[7:0] | | | | | | | |
| 0x080F | WDTPSH | 7:0 | | PSCNTH[7:0] | | | | | | | |
| 0x0810 | WDTTMR | 7:0 | | WDTTMR[4:0] | | | | STATE | PSCNT[1:0] | | |

11.8 Register Definitions: Windowed Watchdog Timer Control

11.8.1 WDTCON0

Name: WDTCON0
Address: 0x80C

Watchdog Timer Control Register 0

| | | | | | | | | |
|--------|------------|---|-----|-----|-----|-----|-----|-----|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | WDTPS[4:0] | | | | | | | SEN |
| Access | | | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | | q | q | q | q | q | 0 |

Bits 5:1 – WDTPS[4:0] Watchdog Timer Prescale Select bits⁽¹⁾

Bit Value = Prescale Rate

| Value | Description |
|-------|--|
| 11111 | Reserved. Results in minimum interval (1 ms) |
| 10011 | |
| 10010 | 1:8388608 (2 ²³) (Interval 256s nominal) |
| 10001 | 1:4194304 (2 ²²) (Interval 128s nominal) |
| 10000 | 1:2097152 (2 ²¹) (Interval 64s nominal) |
| 01111 | 1:1048576 (2 ²⁰) (Interval 32s nominal) |
| 01110 | 1:524288 (2 ¹⁹) (Interval 16s nominal) |
| 01101 | 1:262144 (2 ¹⁸) (Interval 8s nominal) |
| 01100 | 1:131072 (2 ¹⁷) (Interval 4s nominal) |
| 01011 | 1:65536 (Interval 2s nominal) (Reset value) |
| 01010 | 1:32768 (Interval 1s nominal) |
| 01001 | 1:16384 (Interval 512 ms nominal) |
| 01000 | 1:8192 (Interval 256 ms nominal) |
| 00111 | 1:4096 (Interval 128 ms nominal) |
| 00110 | 1:2048 (Interval 64 ms nominal) |
| 00101 | 1:1024 (Interval 32 ms nominal) |
| 00100 | 1:512 (Interval 16 ms nominal) |
| 00011 | 1:256 (Interval 8 ms nominal) |
| 00010 | 1:128 (Interval 4 ms nominal) |
| 00001 | 1:64 (Interval 2 ms nominal) |
| 00000 | 1:32 (Interval 1 ms nominal) |

Bit 0 – SEN Software Enable/Disable for Watchdog Timer bit

| Value | Condition | Description |
|-------|--------------|---------------------|
| – | If WDTE = 1x | This bit is ignored |
| 1 | If WDTE = 01 | WDT is turned on |
| 0 | If WDTE = 01 | WDT is turned off |
| – | If WDTE = 00 | This bit is ignored |

Note:

1. Times are approximate. WDT time is based on 31 kHz LFINTOSC.
2. When WDTCP3 in CONFIG3 = 11111, the Reset value (q) of WDTPS is '01011'. Otherwise, the Reset value of WDTPS is equal to WDTCP3 in CONFIG3.
3. When WDTCP3 in CONFIG3L ≠ 11111, these bits are read-only.

11.8.2 WDTCON1

Name: WDTCON1
Address: 0x80D

Watchdog Timer Control Register 1

| | | | | | | | | |
|--------|-------------|-----|-----|-----|-------------|-----|-----|-----|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | WDTCSS[2:0] | | | | WINDOW[2:0] | | | |
| Access | | R/W | R/W | R/W | | R/W | R/W | R/W |
| Reset | | q | q | q | | q | q | q |

Bits 6:4 – WDTCSS[2:0] Watchdog Timer Clock Select bits

| Value | Description |
|------------|--------------------|
| 111 to 010 | Reserved |
| 001 | MFINTOSC 31.25 kHz |
| 000 | LFINTOSC 31 kHz |

Bits 2:0 – WINDOW[2:0] Watchdog Timer Window Select bits

| WINDOW | Window delay Percent of time | Window opening Percent of time |
|--------|------------------------------|--------------------------------|
| 111 | N/A | 100 |
| 110 | 12.5 | 87.5 |
| 101 | 25 | 75 |
| 100 | 37.5 | 62.5 |
| 011 | 50 | 50 |
| 010 | 62.5 | 37.5 |
| 001 | 75 | 25 |
| 000 | 87.5 | 12.5 |

Note:

1. If WDTCCS in CONFIG3 = 111, the Reset value of WDTCSS is '000'.
2. The Reset value (q) of WINDOW is determined by the value of WDTCCS in the CONFIG3 register.
3. If WDTCCS in CONFIG3 ≠ 111, these bits are read-only.
4. If WDTCCS in CONFIG3 ≠ 111, these bits are read-only.

11.8.3 WDTPSL

Name: WDTPSL
Address: 0x80E

WWDT Prescale Select Low Register (Read-Only)

| | | | | | | | | |
|--------|-------------|----|----|----|----|----|----|----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | PSCNTL[7:0] | | | | | | | |
| Access | RO | RO | RO | RO | RO | RO | RO | RO |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bits 7:0 – PSCNTL[7:0] Prescale Select Low Byte bits⁽¹⁾

Note:

1. The 18-bit WDT prescale value, PSCNT[17:0] includes the WDTPSL, WDTPSH and the lower two bits of the WDTTMR registers. PSCNT[17:0] is intended for debug operations and should be read during normal operation.

11.8.4 WDTPSH

Name: WDTPSH
Address: 0x80F

WWDT Prescale Select High Register (Read-Only)

| | | | | | | | | |
|--------|-------------|----|----|----|----|----|----|----|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | PSCNTH[7:0] | | | | | | | |
| Access | RO | RO | RO | RO | RO | RO | RO | RO |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bits 7:0 – PSCNTH[7:0] Prescale Select High Byte bits⁽¹⁾

Note:

1. The 18-bit WDT prescale value, PSCNT[17:0] includes the WDTPSL, WDTPSH and the lower bits of the WDTTMR registers. PSCNT[17:0] is intended for debug operations and should be read during normal operation.

11.8.5 WDTTMR

Name: WDTTMR
Address: 0x810

WDT Timer Register (Read-Only)

| | | | | | | | | | |
|--------|-----|-------------|----|----|----|----|-------|------------|----|
| | Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | WDTTMR[4:0] | | | | | STATE | PSCNT[1:0] | |
| Access | | RO | RO | RO | RO | RO | RO | RO | RO |
| Reset | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bits 7:3 – WDTTMR[4:0] Watchdog Window Value bits

| WINDOW | WDT Window State | | Open Percent |
|--------|------------------|-------------|--------------|
| | Closed | Open | |
| 111 | N/A | 00000-11111 | 100 |
| 110 | 00000-00011 | 00100-11111 | 87.5 |
| 101 | 00000-00111 | 01000-11111 | 75 |
| 100 | 00000-01011 | 01100-11111 | 62.5 |
| 011 | 00000-01111 | 10000-11111 | 50 |
| 010 | 00000-10011 | 10100-11111 | 37.5 |
| 001 | 00000-10111 | 11000-11111 | 25 |
| 000 | 00000-11011 | 11100-11111 | 12.5 |

Bit 2 – STATE WDT Armed Status bit

| Value | Description |
|-------|------------------|
| 1 | WDT is armed |
| 0 | WDT is not armed |

Bits 1:0 – PSCNT[1:0] Prescale Select Upper bits⁽¹⁾

Note:

- The 18-bit WDT prescale value, PSCNT[17:0] includes the WDTPSL, WDTPSH and the lower two bits of the WDTTMR registers. PSCNT[17:0] is intended for debug operations and should be read during normal operation.

12. Power-Saving Operation Modes

The purpose of the Power-Down modes is to reduce power consumption. There are three Power-Down modes:

- Doze mode
- Idle mode
- Sleep mode

12.1 Doze Mode

Doze mode allows for power saving by reducing CPU operation and program memory (PFM) access, without affecting peripheral operation. Doze mode differs from Sleep mode because the bandgap and system oscillators continue to operate, while only the CPU and PFM are affected. The reduced execution saves power by eliminating unnecessary operations within the CPU and memory.

When the Doze Enable bit is set (**DOZEN** = 1), the CPU executes only one instruction cycle out of every N cycles as defined by the **DOZE** bits. For example, if **DOZE** = 100, the instruction cycle ratio is 1:32. The CPU and memory execute for one instruction cycle and then lay Idle for 31 instruction cycles. During the unused cycles, the peripherals continue to operate at the system clock speed.

Related Links

[12.5.2 CPUDOZE](#)

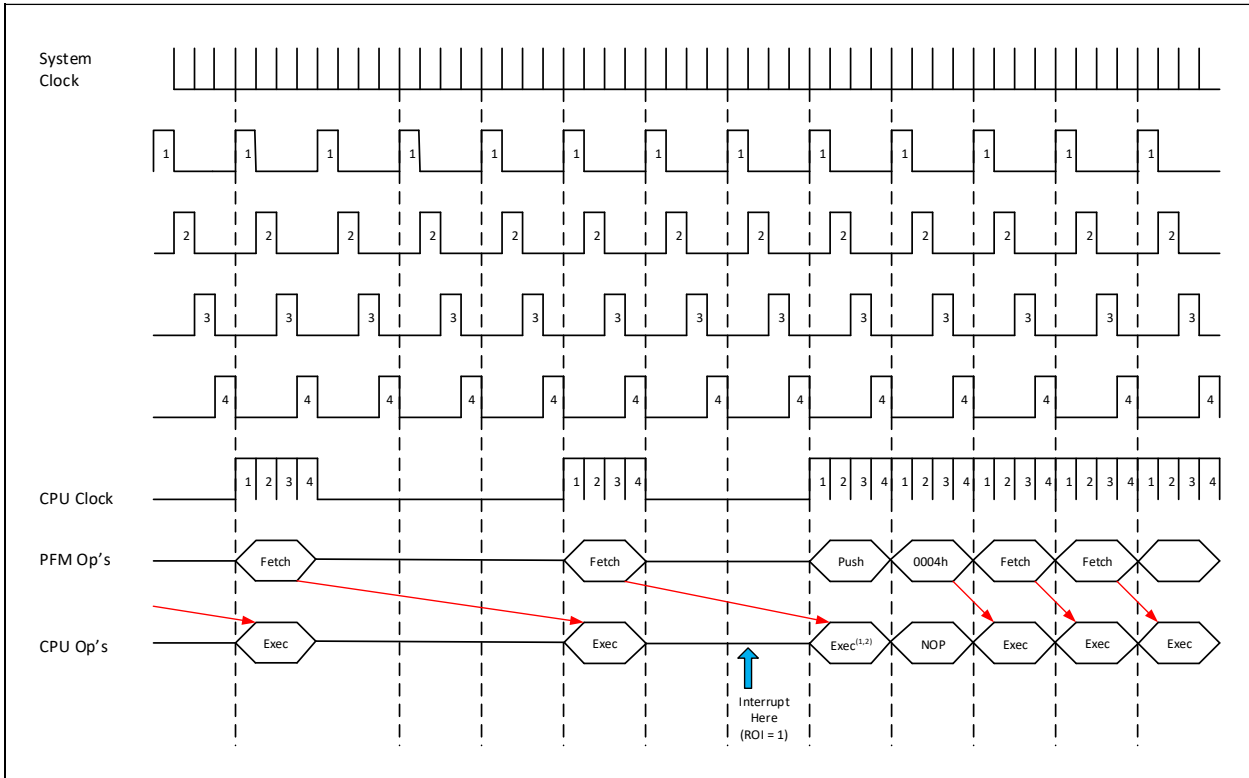
12.1.1 Doze Operation

The Doze operation is illustrated in [Figure 12-1](#). For this example:

- Doze enabled (**DOZEN** = 1)
- **DOZE** = 001 (1:4) ratio
- Recover-on-Interrupt enabled (**ROI** = 1)

As with normal operation, the program memory fetches for the next instruction cycle. The instruction clocks to the peripherals continue throughout.

Figure 12-1. Doze Mode Operation Example



Note:

1. Multicycle instructions are executed to completion before fetching 0004h.
2. If the prefetched instruction clears GIE, the ISR will not occur, but DOZEN is still cleared and the CPU will resume execution at full speed.

12.1.2 Interrupts During Doze

System behavior if an interrupt occurs during DOZE can be configured using the Recover-on-Interrupt (ROI) bit and the Doze-on-Exit (DOE) bit. Refer to the table below for details about system behavior in all cases for a transition from Main to ISR back to Main.

Table 12-1. Interrupts During DOZE

| DOZEN | ROI | Code Flow | | |
|-------|-----|------------------|--|--|
| | | Main | ISR ⁽¹⁾ | Return to Main |
| 0 | 0 | Normal Operation | Normal operation and DOE = DOZEN (in hardware) DOZEN = 0 (unchanged) | |
| 0 | 1 | Normal Operation | Normal operation and DOE = DOZEN (in hardware) DOZEN = 0 (unchanged) | If DOE = 1 when return from interrupt: DOZE operation and DOZEN = 1 (in hardware) If DOE = 0 when return from interrupt: Normal operation and DOZEN = 0 (in hardware) |
| 1 | 0 | DOZE operation | DOZE operation and DOE = DOZEN (in hardware) DOZEN = 1 (unchanged) | |
| 1 | 1 | DOZE operation | Normal operation and DOE = DOZEN (in hardware) DOZEN = 0 (unchanged) | |

Note:

1. User software can change DOE bit in the ISR.

12.2 Sleep Mode

Sleep mode is entered by executing the `SLEEP` instruction, while the Idle Enable (IDLEN) bit of the CPUDOZE register is cleared (IDLEN = 0). If the `SLEEP` instruction is executed while the IDLEN bit is set (IDLEN = 1), the CPU will enter the Idle mode.

Upon entering Sleep mode, the following conditions exist:

1. Resets other than WDT are not affected by Sleep mode; WDT will be cleared but keeps running if enabled for operation during Sleep.
2. The \overline{PD} bit of the STATUS register is cleared.
3. The \overline{TO} bit of the STATUS register is set.
4. The CPU and the System clocks are disabled.
5. 31 kHz LFINTOSC, HFINTOSC and SOSC will remain enabled if any peripheral has requested them as a clock source or if the HFOEN, LFOEN, or SOSSEN bits of the OSCEN register are set.
6. ADC is unaffected if the FRC oscillator is selected. When the ADC clock 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 active.
7. I/O ports maintain the status they had before `SLEEP` was executed (driving high, low, or high-impedance) only if no peripheral connected to the I/O port is active.

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

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

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

I/O pins that are high-impedance inputs should be pulled to V_{DD} or V_{SS} externally to avoid switching currents caused by floating inputs.

Examples of internal circuitry that might be sourcing current include modules such as the DAC and FVR modules.

Related Links

[12.2.3 Low-Power Sleep Mode](#)

[5.8.4 STATUS](#)

[30. FVR - Fixed Voltage Reference](#)

[33. DAC - 5-Bit Digital-to-Analog Converter](#)

12.2.1 Wake-up from Sleep

The device can wake-up from Sleep through one of the following events:

1. External Reset input on \overline{MCLR} pin, if enabled.
2. BOR Reset, if enabled.
3. POR Reset.
4. Windowed Watchdog Timer, if enabled.
5. Any external interrupt.
6. Interrupts by peripherals capable of running during Sleep (see individual peripheral for more information).

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

When the `SLEEP` instruction is being executed, the next instruction ($PC + 1$) is prefetched. For the device to wake-up through an interrupt event, the corresponding interrupt enable bit must be enabled. Wake-up will occur regardless of the state of the GIE bit. If the GIE bit is disabled, the device continues execution at the instruction after the `SLEEP` instruction. If the GIE bit is enabled, the device executes the instruction after the `SLEEP` instruction, the device will then call the Interrupt Service Routine. In cases where the execution of the instruction following `SLEEP` is not desirable, the user should have a `NOP` after the `SLEEP` instruction.

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

Related Links

[10.12 Determining the Cause of a Reset](#)

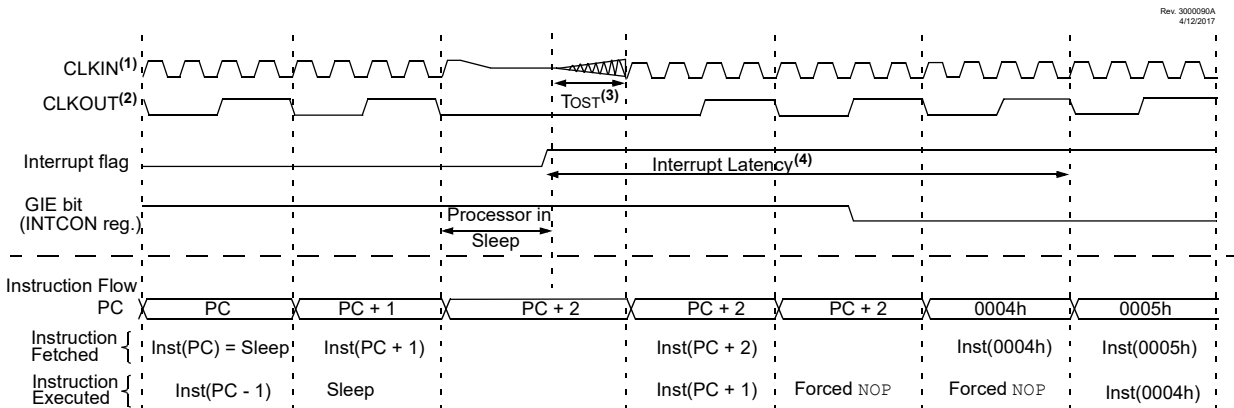
12.2.2 Wake-up Using Interrupts

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

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

Even if the flag bits were checked before executing a `SLEEP` instruction, it may be possible for flag bits to become set before the `SLEEP` instruction completes. To determine whether a `SLEEP` instruction executed, test the \overline{PD} bit. If the \overline{PD} bit is set, the `SLEEP` instruction was executed as a `NOP`.

Figure 12-2. Wake-Up From Sleep Through Interrupt



Note:

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

12.2.3 Low-Power Sleep Mode

The PIC16(L)F18426/46 devices contain an internal Low Dropout (LDO) voltage regulator, which allows the device I/O pins to operate at voltages up to 5.5V while the internal device logic operates at a lower voltage. The LDO and its associated reference circuitry must remain active when the device is in Sleep mode.

The PIC16(L)F18426/46 devices allow the user to optimize the operating current in Sleep, depending on the application requirements.

Low-Power Sleep mode can be selected by setting the VREGPM bit of the VREGCON register. Depending on the configuration of these bits, the LDO and reference circuitry are placed in a low-power state when the device is in Sleep.

12.2.3.1 Sleep Current vs. Wake-up Time

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

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

12.2.3.2 Peripheral Usage in Sleep

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

- Brown-out Reset (BOR)
- Windowed Watchdog Timer (WWDT)
- External interrupt pin/Interrupt-On-Change pins
- Timer1 (with external clock source)

It is the responsibility of the end user to determine what is acceptable for their application when setting the VREGPM settings in order to ensure operation in Sleep.



Important: The LF devices do not have a configurable Low-Power Sleep mode. LFs are unregulated devices and are always in the lowest power state when in Sleep, with no wake-up time penalty. These devices have a lower maximum V_{DD} and I/O voltage than the F devices.

Related Links

[40. Electrical Specifications](#)

12.3 Idle Mode

When the Idle Enable (IDLEN) bit is clear (IDLEN = 0), the `SLEEP` instruction will put the device into full Sleep mode. When IDLEN is set (IDLEN = 1), the `SLEEP` instruction will put the device into Idle mode. In Idle mode, the CPU and memory operations are halted, but the peripheral clocks continue to run. This mode is similar to Doze mode, except that in Idle both the CPU and program memory are shut off.



Important: Peripherals using F_{OSC} will continue running while in Idle (but not in Sleep). Peripherals using HFINTOSC:LFINTOSC will continue running in both Idle and Sleep.



Important: If $\overline{CLKOUTEN}$ is enabled ($\overline{CLKOUTEN} = 0$, Configuration Word 1), the output will continue operating while in Idle.

12.3.1 Idle and Interrupts

Idle mode ends when an interrupt occurs (even if GIE = 0), but IDLEN is not changed. The device can re-enter Idle by executing the `SLEEP` instruction.

If Recover-on-Interrupt is enabled (ROI = 1), the interrupt that brings the device out of Idle also restores full-speed CPU execution when Doze is also enabled.

12.3.2 Idle and WWDT

When in Idle, the WWDT Reset is blocked and will instead wake the device. The WWDT wake-up is not an interrupt, therefore ROI does not apply.



Important: The WWDT can bring the device out of Idle, in the same way it brings the device out of Sleep. The DOZEN bit is not affected.

12.4 Register Summary - Power Savings Control

| Address | Name | Bit Pos. | | | | | | | |
|---------------|----------|----------|-------|-------|-----|-----|--|--------|-----------|
| 0x00 ... | Reserved | | | | | | | | |
| 0x0811 | | | | | | | | | |
| 0x0812 | VREGCON | 7:0 | | | | | | VREGPM | |
| 0x0813 ... | Reserved | | | | | | | | |
| 0x088B | | | | | | | | | |
| 0x088C | CPUDOZE | 7:0 | IDLEN | DOZEN | ROI | DOE | | | DOZE[2:0] |

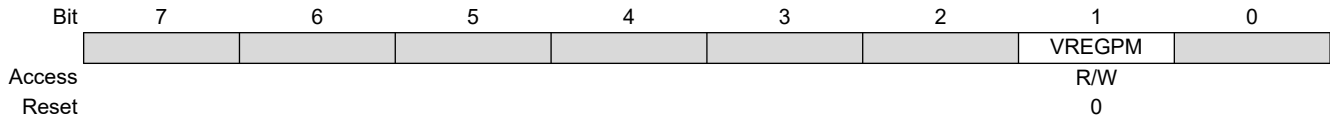
12.5 Register Definitions: Power Savings Control

12.5.1 VREGCON

Name: VREGCON

Address: 0x812

Voltage Regulator Control Register



Bit 1 – VREGPM Voltage Regulator Power Mode Selection bit

This register is available only for F devices.

| Value | Description |
|-------|--|
| 1 | Low-Power Sleep mode enabled in Sleep. Draws lowest current in Sleep, slower wake-up |
| 0 | Normal Power mode enabled in Sleep. Draws higher current in Sleep, faster wake-up |

12.5.2 CPUDOZE

Name: CPUDOZE
Address: 0x88C

Doze and Idle Register

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|-------|-----------|-----|-----------|---|-----------|-----|-----|
| | IDLEN | DOZEN | ROI | DOE | | DOZE[2:0] | | |
| Access | R/W | R/W/HC/HS | R/W | R/W/HC/HS | | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | | 0 | 0 | 0 |

Bit 7 – IDLEN Idle Enable bit

| Value | Description |
|-------|---|
| 1 | A SLEEP instruction places device into Idle mode |
| 0 | A SLEEP instruction places the device into Sleep mode |

Bit 6 – DOZEN Doze Enable bit⁽¹⁾

| Value | Description |
|-------|----------------------------------|
| 1 | Places devices into DOZE setting |
| 0 | Places devices into Normal mode |

Bit 5 – ROI Recover-on-Interrupt bit⁽¹⁾

| Value | Description |
|-------|--|
| 1 | Entering the Interrupt Service Routine (ISR) makes DOZEN = 0 |
| 0 | Entering the Interrupt Service Routine (ISR) does not change DOZEN |

Bit 4 – DOE Doze-on-Exit bit⁽¹⁾

| Value | Description |
|-------|---------------------------------------|
| 1 | Executing the ISR makes DOZEN = 1 |
| 0 | Exiting the ISR does not change DOZEN |

Bits 2:0 – DOZE[2:0] Ratio of CPU Instruction Cycles to Peripheral Instruction Cycles

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

Note:

1. See the link below for more details.

Related Links

[12.1.2 Interrupts During Doze](#)

13. PMD - Peripheral Module Disable

The PIC16(L)F18426/46 devices provide the ability to disable selected modules, placing them into the lowest possible Power mode.



Important: For legacy reasons, all modules are ON by default following any Reset.

13.1 Disabling a Module

A peripheral can be disabled by setting the corresponding peripheral disable bit in the [PMDx](#) register. Disabling a module has the following effects:

- All clock and control inputs to the module are suspended; there are no logic transitions, and the module will not function.
- The module is held in Reset.
 - Writing to the SFRs is disabled.
 - Reading returns 0x00
- Analog outputs are disabled; digital outputs read '0'

Related Links

[16.9.2 xxxPPS](#)

[16.9.1 PPSLOCK](#)

13.2 Enabling a Module

When the register bit is cleared, the module is re-enabled and will be in its Reset state; SFR data will reflect the POR Reset values. Depending on the module, it may take up to one full instruction cycle for the module to become active.



Important: There should be no interaction with the module (e.g., writing to registers) for at least one instruction after it has been re-enabled.

13.3 System Clock Disable

Setting the SYSCMD bit in the [13.5.1 PMD0](#) register disables the system clock F_{OSC} distribution network to the peripherals. Not all peripherals make use of SYSCLK, so not all peripherals are affected. Refer to the specific peripheral description to see if it will be affected by this bit or not.

Related Links

[13.5.1 PMD0](#)

13.4 Register Summary - PMD

| Address | Name | Bit Pos. | | | | | | | | |
|-------------|----------------------|----------|--------|--------|--------|---------|--------|--------|---------|---------|
| 0x00 ... | Reserved | | | | | | | | | |
| 0x0795 | | | | | | | | | | |
| 0x0796 | PMD0 | 7:0 | SYSCMD | FVRMD | | | | NVMMD | CLKRMD | IOCMD |
| 0x0797 | PMD1 | 7:0 | | TMR6MD | TMR5MD | TMR4MD | TMR3MD | TMR2MD | TMR1MD | TMR0MD |
| 0x0798 | PMD2 | 7:0 | NCO1MD | | | | | | | |
| 0x0799 | PMD3 | 7:0 | | DAC1MD | ADCMD | | | C2MD | C1MD | ZCDMD |
| 0x079A | PMD4 | 7:0 | | PWM7MD | PWM6MD | | CCP4MD | CCP3MD | CCP2MD | CCP1MD |
| 0x079B | PMD5 | 7:0 | | CWG2MD | CWG1MD | | | | | |
| 0x079C | PMD6 | 7:0 | | | | UART1MD | | | MSSP2MD | MSSP1MD |
| 0x079D | PMD7 | 7:0 | | | SMT1MD | CLC4MD | CLC3MD | CLC2MD | CLC1MD | DSM1MD |

13.5 Register Definitions: Peripheral Module Disable

13.5.1 PMD0

Name: PMD0
Address: 0x796

PMD Control Register 0

| | | | | | | | | |
|--------|--------|-------|---|---|---|-------|--------|-------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | SYSCMD | FVRMD | | | | NVMMD | CLKRMD | IOCMD |
| Access | R/W | R/W | | | | R/W | R/W | R/W |
| Reset | 0 | 0 | | | | 0 | 0 | 0 |

Bit 7 – SYSCMD Disable Peripheral System Clock Network bit
 Disables the System clock network

| Value | Description |
|-------|---|
| 1 | System clock network disabled (F _{OSC}) |
| 0 | System clock network enabled |

Bit 6 – FVRMD Disable Fixed Voltage Reference bit

| Value | Description |
|-------|---------------------|
| 1 | FVR module disabled |
| 0 | FVR module enabled |

Bit 2 – NVMMD NVM Module Disable bit⁽¹⁾

Disables the NVM module

| Value | Description |
|-------|---|
| 1 | All Memory reading and writing is disabled; NVMCON registers cannot be written; FSR access to these locations returns zero. |
| 0 | NVM module enabled |

Bit 1 – CLKRMD Disable Clock Reference bit

| Value | Description |
|-------|----------------------|
| 1 | CLKR module disabled |
| 0 | CLKR module enabled |

Bit 0 – IOCMD Disable Interrupt-on-Change bit, All Ports

| Value | Description |
|-------|------------------------|
| 1 | IOC module(s) disabled |
| 0 | IOC module(s) enabled |

Note:

1. When enabling NVM, a delay of up to 1 μ s may be required before accessing data.

Related Links

[13.3 System Clock Disable](#)

13.5.2 PMD1

Name: PMD1
Address: 0x797

PMD Control Register 1

| | | | | | | | | |
|--------|---|--------|--------|--------|--------|--------|--------|--------|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit | | TMR6MD | TMR5MD | TMR4MD | TMR3MD | TMR2MD | TMR1MD | TMR0MD |
| Access | | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

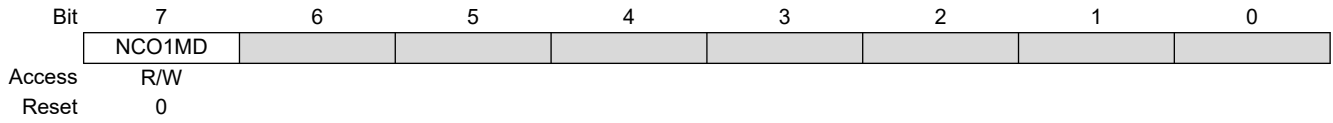
Bits 0, 1, 2, 3, 4, 5, 6 – TMRnMD Disable Timer n bit

| Value | Description |
|-------|----------------------|
| 1 | TMRn module disabled |
| 0 | TMRn module enabled |

13.5.3 PMD2

Name: PMD2
Address: 0x798

PMD Control Register 2



Bit 7 – NCO1MD Disable Numerically Control Oscillator bit

| Value | Description |
|-------|----------------------|
| 1 | NCO1 module disabled |
| 0 | NCO1 module enabled |

13.5.4 PMD3

Name: PMD3
Address: 0x799

PMD Control Register 3

| | | | | | | | | |
|--------|---|--------|-------|---|---|------|------|-------|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | DAC1MD | ADCMD | | | C2MD | C1MD | ZCDMD |
| Access | | R/W | R/W | | | R/W | R/W | R/W |
| Reset | | 0 | 0 | | | 0 | 0 | 0 |

Bit 6 – DAC1MD Disable DAC1 bit

| Value | Description |
|-------|---------------------|
| 1 | DAC module disabled |
| 0 | DAC module enabled |

Bit 5 – ADCMD Disable ADC bit

| Value | Description |
|-------|---------------------|
| 1 | ADC module disabled |
| 0 | ADC module enabled |

Bit 2 – C2MD Disable Comparator C2 bit

| Value | Description |
|-------|--------------------|
| 1 | C2 module disabled |
| 0 | C2 module enabled |

Bit 1 – C1MD Disable Comparator C1 bit

| Value | Description |
|-------|--------------------|
| 1 | C1 module disabled |
| 0 | C1 module enabled |

Bit 0 – ZCDMD Disable Zero-Cross Detect module bit

| Value | Description |
|-------|---------------------|
| 1 | ZCD module disabled |
| 0 | ZCD module enabled |

13.5.5 PMD4

Name: PMD4
Address: 0x79A

PMD Control Register 4

| | | | | | | | | |
|--------|---|--------|--------|---|--------|--------|--------|--------|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | PWM7MD | PWM6MD | | CCP4MD | CCP3MD | CCP2MD | CCP1MD |
| Access | | R/W | R/W | | R/W | R/W | R/W | R/W |
| Reset | | 0 | 0 | | 0 | 0 | 0 | 0 |

Bit 6 – PWM7MD Disable Pulse-Width Modulator PWM7 bit

| Value | Description |
|-------|----------------------|
| 1 | PWM7 module disabled |
| 0 | PWM7 module enabled |

Bit 5 – PWM6MD Disable Pulse-Width Modulator PWM6 bit

| Value | Description |
|-------|----------------------|
| 1 | PWM6 module disabled |
| 0 | PWM6 module enabled |

Bit 3 – CCP4MD Disable Pulse-Width Modulator CCP4 bit

| Value | Description |
|-------|----------------------|
| 1 | CCP4 module disabled |
| 0 | CCP4 module enabled |

Bit 2 – CCP3MD Disable Pulse-Width Modulator CCP3 bit

| Value | Description |
|-------|----------------------|
| 1 | CCP3 module disabled |
| 0 | CCP3 module enabled |

Bit 1 – CCP2MD Disable Pulse-Width Modulator CCP2 bit

| Value | Description |
|-------|----------------------|
| 1 | CCP2 module disabled |
| 0 | CCP2 module enabled |

Bit 0 – CCP1MD Disable Pulse-Width Modulator CCP1 bit

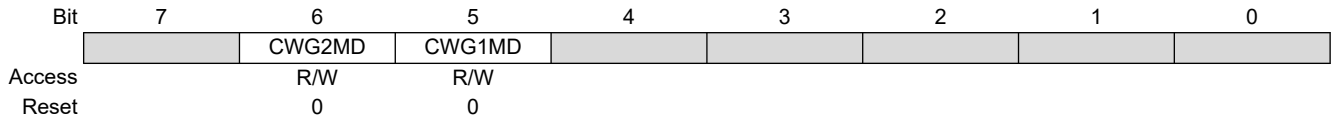
| Value | Description |
|-------|----------------------|
| 1 | CCP1 module disabled |
| 0 | CCP1 module enabled |

13.5.6 PMD5

Name: PMD5

Address: 0x79B

PMD Control Register 5



Bit 6 – CWG2MD Disable CWG2 bit

| Value | Description |
|-------|----------------------|
| 1 | CWG2 module disabled |
| 0 | CWG2 module enabled |

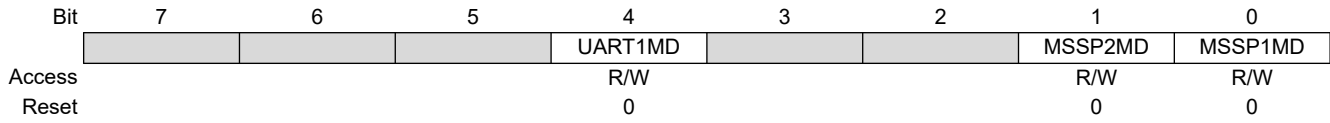
Bit 5 – CWG1MD Disable CWG1 bit

| Value | Description |
|-------|----------------------|
| 1 | CWG1 module disabled |
| 0 | CWG1 module enabled |

13.5.7 PMD6

Name: PMD6
Address: 0x79C

PMD Control Register 6



Bit 4 – UART1MD Disable EUSART1 bit

| Value | Description |
|-------|-------------------------|
| 1 | EUSART1 module disabled |
| 0 | EUSART1 module enabled |

Bit 1 – MSSP2MD Disable MSSP2 bit

| Value | Description |
|-------|-----------------------|
| 1 | MSSP2 module disabled |
| 0 | MSSP2 module enabled |

Bit 0 – MSSP1MD Disable MSSP1 bit

| Value | Description |
|-------|-----------------------|
| 1 | MSSP1 module disabled |
| 0 | MSSP1 module enabled |

13.5.8 PMD7

Name: PMD7
Address: 0x79D

PMD Control Register 7

| | | | | | | | | |
|--------|---|---|--------|--------|--------|--------|--------|--------|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit | | | SMT1MD | CLC4MD | CLC3MD | CLC2MD | CLC1MD | DSM1MD |
| Access | | | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 |

Bit 5 – SMT1MD Disable Signal Measurement Timer1 bit

| Value | Description |
|-------|----------------------|
| 1 | SMT1 module disabled |
| 0 | SMT1 module enabled |

Bits 1, 2, 3, 4 – CLCnMD Disable CLCn bit

| Value | Description |
|-------|----------------------|
| 1 | CLCn module disabled |
| 0 | CLCn module enabled |

Bit 0 – DSM1MD Disable Data Signal Modulator 1 bit

| Value | Description |
|-------|----------------------|
| 1 | DSM1 module disabled |
| 0 | DSM1 module enabled |

14. I/O Ports

14.1 PORT Availability

Table 14-1. PORT Availability Per Device

| PORTs | PORT Description | PIC16(L)F18426 | PIC16(L)F18446 |
|-------|-----------------------------------|----------------|----------------|
| PORTA | 6-bit wide, bidirectional port. | • | • |
| PORTB | 4-bit wide, bidirectional port. | | • |
| PORTC | 6/8-bit wide, bidirectional port. | • | • |

14.2 I/O Ports Description

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

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

Most PORT pins share functions with device peripherals, both analog and digital. In general, when a peripheral is enabled on a PORT pin, that pin cannot be used as a general purpose output; however, the pin can still be read.

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

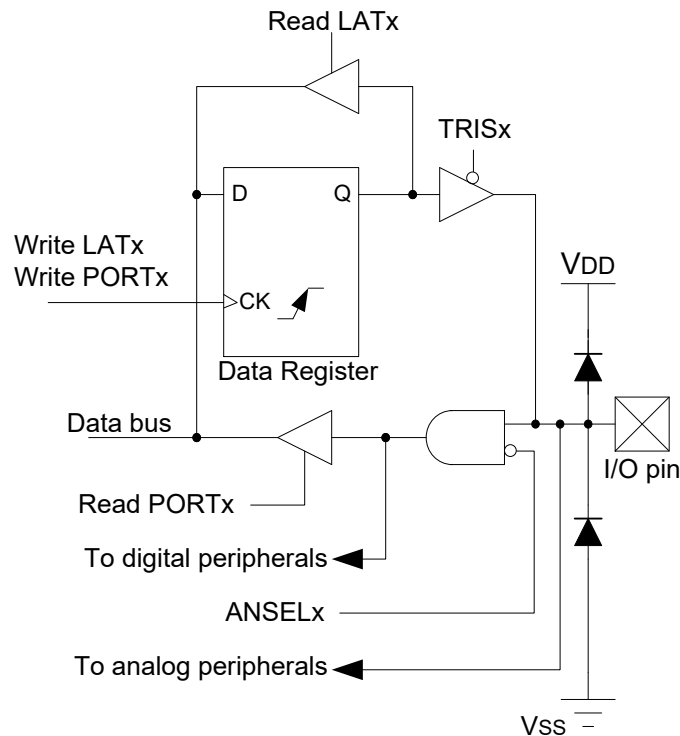
A write operation to the LATx register has the same effect as a write to the corresponding PORTx register. A read of the LATx register reads of the values held in the I/O PORT latches, while a read of the PORTx register reads the actual I/O pin value.

Ports that support analog inputs have an associated ANSELx register. When an ANSELx bit is set, the digital input buffer associated with that bit is disabled.

Disabling the input buffer prevents analog signal levels on the pin between a logic high and low from causing excessive current in the logic input circuitry. A simplified model of a generic I/O port, without the interfaces to other peripherals, is shown in the following figure:

Figure 14-1. Generic I/O Port Operation

Rev. 10-000052A
7/30/2013



14.3 I/O Priorities

Each pin defaults to the PORT data latch after Reset. Other functions are selected with the peripheral pin select logic. See “Peripheral Pin Select (PPS) Module” for more information.

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

Analog outputs, when enabled, take priority over digital outputs and force the digital output driver into a high-impedance state.

The pin function priorities are as follows:

1. Configuration bits
2. Analog outputs (disable the input buffers)
3. Analog inputs
4. Port inputs and outputs from PPS

Related Links

[16. PPS - Peripheral Pin Select Module](#)

14.4 PORTx Registers

In this section the generic names such as PORTx, LATx, TRISx, etc. can be associated with PORTA, PORTB, PORTC, etc. depending on availability per device (see related link below).

Related Links

[14.1 PORT Availability](#)

14.4.1 Data Register

PORTx is a bidirectional port, and its corresponding data direction register is TRISx. Setting a TRISx bit ('1') will make the corresponding PORTx pin an input (i.e., disable the output driver). Clearing a TRISx bit ('0') will make the corresponding PORTx pin an output (i.e., it enables output driver and puts the contents of the output latch on the selected pin). The example below shows how to initialize PORTA.

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

The PORT data latch LATx holds the output port data and contains the latest value of a LATx or PORTx write. The examples below show how to initialize PORTA.

Example 14-1. Initializing PORTA in assembly

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

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

Example 14-2. Initializing PORTA in C

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

PORTA = 0x00;      // Clear PORTA
LATA  = 0x00;      // Clear Data Latch
ANSELA = 0x00;    // Enable digital drivers
TRISA = 0x38;     // Set RA[5:3] as inputs and set others as outputs
```

Related Links

[14.6.1 PORTA](#)

[14.6.4 TRISA](#)

[14.6.7 LATA](#)

14.4.2 Direction Control

The TRISx register controls the PORTx pin output drivers, even when they are being used as analog inputs. The user should ensure the bits in the TRISx register are maintained set when using them as analog inputs. I/O pins configured as analog inputs always read '0'.

14.4.3 Open-Drain Control

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



Important: It is not necessary to set open-drain control when using the pin for I²C; the I²C module controls the pin and makes the pin open-drain.

14.4.4 Slew Rate Control

The SLRCONx register controls the slew rate option for each PORT pin. Slew rate for each port pin can be controlled independently. When an SLRCONx bit is set, the corresponding PORT pin drive is slew rate limited. When an SLRCONx bit is cleared, The corresponding PORT pin drive slews at the maximum rate possible.

14.4.5 Input Threshold Control

The INLVLx register controls the input voltage threshold for each of the available PORTx input pins. A selection between the Schmitt Trigger CMOS or the TTL compatible thresholds is available. The input threshold is important in determining the value of a read of the PORTx register and also the level at which an interrupt-on-change occurs, if that feature is enabled.



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

14.4.6 Analog Control

The ANSELx register is used to configure the Input mode of an I/O pin to analog. Setting the appropriate ANSELx 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 ANSELx bits has no effect on digital output functions. A pin with TRIS clear and ANSEL set will still operate as a digital output, but the Input mode will be analog. This can cause unexpected behavior when executing READ-MODIFY-WRITE instructions on the affected port.



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

14.4.7 Weak Pull-up Control

The WPUx register controls the individual weak pull-ups for each PORT pin.

14.4.8 PORTx Functions and Output Priorities

Each PORTx pin is multiplexed with other functions.

Each pin defaults to the PORT latch data after Reset. Other output functions are selected with the peripheral pin select logic, or by enabling an analog output, such as the DAC. See the link below for more information.

Analog input functions, such as ADC and comparator inputs are not shown in the peripheral pin select lists. Digital output functions may continue to control the pin when it is in Analog mode.

Related Links

[16. PPS - Peripheral Pin Select Module](#)

14.4.9 MCLR/V_{PP} Pin

The $\overline{\text{MCLR}}/\text{V}_{\text{PP}}$ pin is an input-only pin. Its operation is controlled by the MCLRE Configuration bit. When selected as a PORT pin (MCLRE = 0), it functions as a digital input-only pin; as such, it does not have TRISx and LATx bits associated with its operation. Otherwise, it functions as the device's Master Clear input. In either configuration, the $\overline{\text{MCLR}}/\text{V}_{\text{PP}}$ pin also functions as the programming voltage input pin during high voltage programming.

The $\overline{\text{MCLR}}/\text{V}_{\text{PP}}$ pin is a read-only bit and will read '1' when MCLRE = 1 (i.e., Master Clear enabled).



Important: On a Power-on Reset, the $\overline{\text{MCLR}}/V_{\text{PP}}$ pin is enabled as a digital input-only if Master Clear functionality is disabled.

The $\overline{\text{MCLR}}/V_{\text{PP}}$ pin has an individually controlled internal weak pull-up. When set, the corresponding WPU bit enables the pull-up. When the $\overline{\text{MCLR}}/V_{\text{PP}}$ pin is configured as $\overline{\text{MCLR}}$, (MCLRE = 1 and, LVP = 0 in the Configuration Words), or configured for Low-Voltage Programming, (MCLRE = x and LVP = 1), the pull-up is always enabled and the WPU bit has no effect.

14.5 Register Summary - Input/Output

| Address | Name | Bit Pos. | | | | | | | | |
|---------------|----------|----------|---------|---------|---------|---------|---------|---------|---------|---------|
| 0x00 ... | Reserved | | | | | | | | | |
| 0x0B | | | | | | | | | | |
| 0x0C | PORTA | 7:0 | | | RA5 | RA4 | RA3 | RA2 | RA1 | RA0 |
| 0x0D | PORTB | 7:0 | RB7 | RB6 | RB5 | RB4 | | | | |
| 0x0E | PORTC | 7:0 | RC7 | RC6 | RC5 | RC4 | RC3 | RC2 | RC1 | RC0 |
| 0x0F ... | Reserved | | | | | | | | | |
| 0x11 | | | | | | | | | | |
| 0x12 | TRISA | 7:0 | | | TRISA5 | TRISA4 | | TRISA2 | TRISA1 | TRISA0 |
| 0x13 | TRISB | 7:0 | TRISB7 | TRISB6 | TRISB5 | TRISB4 | | | | |
| 0x14 | TRISC | 7:0 | TRISC7 | TRISC6 | TRISC5 | TRISC4 | TRISC3 | TRISC2 | TRISC1 | TRISC0 |
| 0x15 ... | Reserved | | | | | | | | | |
| 0x17 | | | | | | | | | | |
| 0x18 | LATA | 7:0 | | | LATA5 | LATA4 | | LATA2 | LATA1 | LATA0 |
| 0x19 | LATB | 7:0 | LATB7 | LATB6 | LATB5 | LATB4 | | | | |
| 0x1A | LATC | 7:0 | LATC7 | LATC6 | LATC5 | LATC4 | LATC3 | LATC2 | LATC1 | LATC0 |
| 0x1B ... | Reserved | | | | | | | | | |
| 0x1F37 | | | | | | | | | | |
| 0x1F38 | ANSELA | 7:0 | | | ANSELA5 | ANSELA4 | | ANSELA2 | ANSELA1 | ANSELA0 |
| 0x1F39 | WPUA | 7:0 | | | WPUA5 | WPUA4 | WPUA3 | WPUA2 | WPUA1 | WPUA0 |
| 0x1F3A | ODCONA | 7:0 | | | ODCA5 | ODCA4 | | ODCA2 | ODCA1 | ODCA0 |
| 0x1F3B | SLRCONA | 7:0 | | | SLRA5 | SLRA4 | | SLRA2 | SLRA1 | SLRA0 |
| 0x1F3C | INLVLA | 7:0 | | | INLVLA5 | INLVLA4 | INLVLA3 | INLVLA2 | INLVLA1 | INLVLA0 |
| 0x1F3D ... | Reserved | | | | | | | | | |
| 0x1F42 | | | | | | | | | | |
| 0x1F43 | ANSELB | 7:0 | ANSELB7 | ANSELB6 | ANSELB5 | ANSELB4 | | | | |
| 0x1F44 | WPUB | 7:0 | WPUB7 | WPUB6 | WPUB5 | WPUB4 | | | | |
| 0x1F45 | ODCONB | 7:0 | ODCB7 | ODCB6 | ODCB5 | ODCB4 | | | | |
| 0x1F46 | SLRCONB | 7:0 | SLRB7 | SLRB6 | SLRB5 | SLRB4 | | | | |
| 0x1F47 | INVLVB | 7:0 | INVLVB7 | INVLVB6 | INVLVB5 | INVLVB4 | | | | |
| 0x1F48 ... | Reserved | | | | | | | | | |
| 0x1F4D | | | | | | | | | | |
| 0x1F4E | ANSELC | 7:0 | ANSELC7 | ANSELC6 | ANSELC5 | ANSELC4 | ANSELC3 | ANSELC2 | ANSELC1 | ANSELC0 |
| 0x1F4F | WPUC | 7:0 | WPUC7 | WPUC6 | WPUC5 | WPUC4 | WPUC3 | WPUC2 | WPUC1 | WPUC0 |
| 0x1F50 | ODCONC | 7:0 | ODCC7 | ODCC6 | ODCC5 | ODCC4 | ODCC3 | ODCC2 | ODCC1 | ODCC0 |
| 0x1F51 | SLRCONC | 7:0 | SLRC7 | SLRC6 | SLRC5 | SLRC4 | SLRC3 | SLRC2 | SLRC1 | SLRC0 |
| 0x1F52 | INLVLC | 7:0 | INLVLC7 | INLVLC6 | INLVLC5 | INLVLC4 | INLVLC3 | INLVLC2 | INLVLC1 | INLVLC0 |

14.6 Register Definitions: Port Control

14.6.1 PORTA

Name: PORTA

Address: 0x00C

PORTA Register

| | | | | | | | | |
|--------|---|---|-----|-----|-----|-----|-----|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | | RA5 | RA4 | RA3 | RA2 | RA1 | RA0 |
| Access | | | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | | x | x | x | x | x | x |

Bits 0, 1, 2, 3, 4, 5 – RAn Port I/O Value

Reset States: POR/BOR = xxxxxx

All Other Resets = uuuuuu

| Value | Description |
|-------|---------------------------|
| 1 | PORT pin is $\geq V_{IH}$ |
| 0 | PORT pin is $\leq V_{IL}$ |

Note:

- Writes to PORTA are actually written to the corresponding LATA register. Reads from PORTA register return actual I/O pin values.
- Bits RA0, RA1 and RA3 are read-only when \overline{DEBUG} is enabled, and will read '0'.
- Bit RA3 will read '1' when $MCLRE = 1$ (master clear enabled) and '0' when \overline{DEBUG} is enabled.

14.6.2 PORTB

Name: PORTB

Address: 0x00D

PORTB Register

| | | | | | | | | |
|--------|-----|-----|-----|-----|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | RB7 | RB6 | RB5 | RB4 | | | | |
| Access | R/W | R/W | R/W | R/W | | | | |
| Reset | x | x | x | x | | | | |

Bits 4, 5, 6, 7 – RBn Port I/O Value bits

Reset States: POR/BOR = xxxx

All Other Resets = uuuu

| Value | Description |
|-------|---------------------------|
| 1 | PORT pin is $\geq V_{IH}$ |
| 0 | PORT pin is $\leq V_{IL}$ |

Note: Writes to PORTB are actually written to the corresponding LATB register.
Reads from PORTB register return actual I/O pin values.

14.6.3 PORTC

Name: PORTC

Address: 0x00E

PORTC Register

| | | | | | | | | |
|--------|-----|-----|-----|-----|-----|-----|-----|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | RC7 | RC6 | RC5 | RC4 | RC3 | RC2 | RC1 | RC0 |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | x | x | x | x | x | x | x | x |

Bits 0, 1, 2, 3, 4, 5, 6, 7 – RCn Port I/O Value bits

Reset States: POR/BOR = xxxxxxxx

All Other Resets = uuuuuuuu

| Value | Description |
|-------|---------------------------|
| 1 | PORT pin is $\geq V_{IH}$ |
| 0 | PORT pin is $\leq V_{IL}$ |

Note:

- Writes to PORTC are actually written to the corresponding LATC register. Reads from PORTC register return actual I/O pin values.
- Bits RC6 and RC7 available on 20-pin or higher pin count devices only; Bits unimplemented for lower pin count devices.

14.6.4 TRISA

Name: TRISA

Address: 0x012

Tri-State Control Register

| | | | | | | | | |
|--------|---|---|--------|--------|----|--------|--------|--------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | | TRISA5 | TRISA4 | | TRISA2 | TRISA1 | TRISA0 |
| Access | | | R/W | R/W | RO | R/W | R/W | R/W |
| Reset | | | 1 | 1 | 1 | 1 | 1 | 1 |

Bits 4, 5 – TRISAx TRISA Port I/O Tri-state Control bits

| Value | Description |
|-------|---|
| 1 | PORTA pin configured as an input (tri-stated) |
| 0 | PORTA pin configured as an output |

Bits 0, 1, 2 – TRISAn TRISA Port I/O Tri-state Control bits

| Value | Description |
|-------|---|
| 1 | PORTA pin configured as an input (tri-stated) |
| 0 | PORTA pin configured as an output |

Note: Bits TRISA0 and TRISA1 are read-only when $\overline{\text{DEBUG}}$ is enabled, and will read '1'.

14.6.5 TRISB

Name: TRISB

Address: 0x013

Tri-State Control Register

| | | | | | | | | |
|--------|--------|--------|--------|--------|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TRISB7 | TRISB6 | TRISB5 | TRISB4 | | | | |
| Access | R/W | R/W | R/W | R/W | | | | |
| Reset | 1 | 1 | 1 | 1 | | | | |

Bits 4, 5, 6, 7 – TRISBn TRISB Port I/O Tri-state Control bits

| Value | Description |
|-------|---|
| 1 | PORTB pin configured as an input (tri-stated) |
| 0 | PORTB pin configured as an output |

14.6.6 TRISC

Name: TRISC

Address: 0x014

Tri-State Control Register

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| | TRISC7 | TRISC6 | TRISC5 | TRISC4 | TRISC3 | TRISC2 | TRISC1 | TRISC0 |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Bits 0, 1, 2, 3, 4, 5, 6, 7 – TRISCn TRISC Port I/O Tri-state Control bits

| Value | Description |
|-------|---|
| 1 | PORTC pin configured as an input (tri-stated) |
| 0 | PORTC pin configured as an output |

Note: Bits TRISC6 and TRISC7 available on 20-pin or higher pin count devices only; Bits unimplemented for lower pin count devices.

14.6.7 LATA

Name: LATA

Address: 0x018

Output Latch Register

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|---|---|-------|-------|----|-------|-------|-------|
| | | | LATA5 | LATA4 | | LATA2 | LATA1 | LATA0 |
| Access | | | R/W | R/W | RO | R/W | R/W | R/W |
| Reset | | | x | x | 1 | x | x | x |

Bits 4, 5 – LATAn Output Latch A Value bits

Reset States: POR/BOR = xx

All Other Resets = uu

Bits 0, 1, 2 – LATAn Output Latch A Value bits

Reset States: POR/BOR = xxx

All Other Resets = uuu

Note: Writes to LATA are equivalent with writes to the corresponding PORTA register. Reads from LATA register return register values, not I/O pin values.

14.6.8 LATB

Name: LATB

Address: 0x019

Output Latch Register

| | | | | | | | | |
|--------|-------|-------|-------|-------|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | LATB7 | LATB6 | LATB5 | LATB4 | | | | |
| Access | R/W | R/W | R/W | R/W | | | | |
| Reset | x | x | x | x | | | | |

Bits 4, 5, 6, 7 – LATBn Output Latch B Value bits

Reset States: POR/BOR = xxxx

All Other Resets = uuuu

Note: Writes to LATB are equivalent with writes to the corresponding PORTB register. Reads from LATB register return register values, not I/O pin values.

14.6.9 LATC

Name: LATC
Address: 0x01A

Output Latch Register

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|-------|-------|-------|-------|-------|-------|-------|-------|
| | LATC7 | LATC6 | LATC5 | LATC4 | LATC3 | LATC2 | LATC1 | LATC0 |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | x | x | x | x | x | x | x | x |

Bits 0, 1, 2, 3, 4, 5, 6, 7 – LATCn Output Latch C Value bits

Reset States: POR/BOR = xxxxxxxx

All Other Resets = uuuuuuuu

Note:

1. Writes to LATC are equivalent with writes to the corresponding PORTC register. Reads from LATC register return register values, not I/O pin values.
2. Bits LATC6 and LATC7 available on 20-pin or higher pin count devices only; Bits unimplemented for lower pin count devices.

14.6.10 ANSELA

Name: ANSELA

Address: 0x1F38

Analog Select Register

| | | | | | | | | |
|--------|---|---|---------|---------|---|---------|---------|---------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | | ANSELA5 | ANSELA4 | | ANSELA2 | ANSELA1 | ANSELA0 |
| Access | | | R/W | R/W | | R/W | R/W | R/W |
| Reset | | | 1 | 1 | | 1 | 1 | 1 |

Bits 4, 5 – ANSELAn Analog Select on RA Pins

| Value | Description |
|-------|---|
| 1 | Analog input. Pin is assigned as analog input. Digital input buffer disabled. |
| 0 | Digital I/O. Pin is assigned to port or digital special function. |

Bits 0, 1, 2 – ANSELAn Analog Select on RA Pins

| Value | Description |
|-------|---|
| 1 | Analog input. Pin is assigned as analog input. Digital input buffer disabled. |
| 0 | Digital I/O. Pin is assigned to port or digital special function. |

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

14.6.11 ANSELB

Name: ANSELB

Address: 0x1F43

Analog Select Register

| | | | | | | | | |
|--------|---------|---------|---------|---------|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | ANSELB7 | ANSELB6 | ANSELB5 | ANSELB4 | | | | |
| Access | R/W | R/W | R/W | R/W | | | | |
| Reset | 1 | 1 | 1 | 1 | | | | |

Bits 4, 5, 6, 7 – ANSELBn Analog Select on RB Pins

| Value | Description |
|-------|---|
| 1 | Analog input. Pin is assigned as analog input. Digital input buffer disabled. |
| 0 | Digital I/O. Pin is assigned to port or digital special function. |

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

14.6.12 ANSELC

Name: ANSELC

Address: 0x1F4E

Analog Select Register

| | | | | | | | | |
|--------|---------|---------|---------|---------|---------|---------|---------|---------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | ANSELC7 | ANSELC6 | ANSELC5 | ANSELC4 | ANSELC3 | ANSELC2 | ANSELC1 | ANSELC0 |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Bits 0, 1, 2, 3, 4, 5, 6, 7 – ANSEL_{Cn} Analog Select on RC Pins

| Value | Description |
|-------|---|
| 1 | Analog input. Pin is assigned as analog input. Digital input buffer disabled. |
| 0 | Digital I/O. Pin is assigned to port or digital special function. |

Note:

1. When setting a pin to an analog input, the corresponding TRIS bit must be set to Input mode in order to allow external control of the voltage on the pin.
2. Bits ANSC6 and ANSC7 available on 20-pin or higher pin count devices only; Bits unimplemented for lower pin count devices.

14.6.13 WPUA

Name: WPUA
Address: 0x1F39

Weak Pull-up Register

| | | | | | | | | |
|--------|---|---|-------|-------|-------|-------|-------|-------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | | WPUA5 | WPUA4 | WPUA3 | WPUA2 | WPUA1 | WPUA0 |
| Access | | | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 |

Bits 0, 1, 2, 3, 4, 5 – WPUAn Weak Pull-up PORTA Control

| Value | Description |
|-------|-----------------------|
| 1 | Weak Pull-up enabled |
| 0 | Weak Pull-up disabled |

Note:

1. If MCLRE = 1, the weak pull-up in RA3 is always enabled; bit WPUA3 will read as '1'.
2. The weak pull-up device is automatically disabled if the pin is configured as an output.

14.6.14 WPUB

Name: WPUB
Address: 0x1F44

Weak Pull-up Register

| | | | | | | | | |
|--------|-------|-------|-------|-------|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | WPUB7 | WPUB6 | WPUB5 | WPUB4 | | | | |
| Access | R/W | R/W | R/W | R/W | | | | |
| Reset | 0 | 0 | 0 | 0 | | | | |

Bits 4, 5, 6, 7 – WPUBn Weak Pull-up PORTB Control bits

| Value | Description |
|-------|-----------------------|
| 1 | Weak Pull-up enabled |
| 0 | Weak Pull-up disabled |

14.6.15 WPUC

Name: WPUC
Address: 0x1F4F

Weak Pull-up Register

| | | | | | | | | |
|--------|-------|-------|-------|-------|-------|-------|-------|-------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | WPUC7 | WPUC6 | WPUC5 | WPUC4 | WPUC3 | WPUC2 | WPUC1 | WPUC0 |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bits 0, 1, 2, 3, 4, 5, 6, 7 – WPUCn Weak Pull-up PORTC Control bits

| Value | Description |
|-------|-----------------------|
| 1 | Weak Pull-up enabled |
| 0 | Weak Pull-up disabled |

Note: Bits WPUC6 and WPUC7 available on 20-pin or higher pin count devices only; Bits unimplemented for lower pin count devices.

14.6.16 ODCONA

Name: ODCONA

Address: 0x1F3A

Open-Drain Control Register

| | | | | | | | | |
|--------|---|---|-------|-------|---|-------|-------|-------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | | ODCA5 | ODCA4 | | ODCA2 | ODCA1 | ODCA0 |
| Access | | | R/W | R/W | | R/W | R/W | R/W |
| Reset | | | 0 | 0 | | 0 | 0 | 0 |

Bits 4, 5 – ODCAn Open-Drain Configuration on RA Pins

| Value | Description |
|-------|---|
| 1 | PORT pin operates as open-drain drive (sink current only) |
| 0 | PORT pin operates as standard push-pull drive (source and sink current) |

Bits 0, 1, 2 – ODCAn Open-Drain Configuration on RA Pins

| Value | Description |
|-------|---|
| 1 | PORT pin operates as open-drain drive (sink current only) |
| 0 | PORT pin operates as standard push-pull drive (source and sink current) |

14.6.17 ODCONB

Name: ODCONB

Address: 0x1F45

Open-Drain Control Register

| | | | | | | | | |
|--------|-------|-------|-------|-------|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | ODCB7 | ODCB6 | ODCB5 | ODCB4 | | | | |
| Access | R/W | R/W | R/W | R/W | | | | |
| Reset | 0 | 0 | 0 | 0 | | | | |

Bits 4, 5, 6, 7 – ODCBn Open-Drain Configuration on RB Pins

| Value | Description |
|-------|---|
| 1 | PORT pin operates as open-drain drive (sink current only) |
| 0 | PORT pin operates as standard push-pull drive (source and sink current) |

14.6.18 ODCONC

Name: ODCONC

Address: 0x1F50

Open-Drain Control Register

| | | | | | | | | |
|--------|-------|-------|-------|-------|-------|-------|-------|-------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | ODCC7 | ODCC6 | ODCC5 | ODCC4 | ODCC3 | ODCC2 | ODCC1 | ODCC0 |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bits 0, 1, 2, 3, 4, 5, 6, 7 – ODCCn Open-Drain Configuration on RC Pins

| Value | Description |
|-------|---|
| 1 | PORT pin operates as open-drain drive (sink current only) |
| 0 | PORT pin operates as standard push-pull drive (source and sink current) |

Note: Bits ODCC6 and ODCC7 available on 20-pin or higher pin count devices only; Bits unimplemented for lower pin count devices.

14.6.19 SLRCONA

Name: SLRCONA

Address: 0x1F3B

Slew Rate Control Register

| | | | | | | | | |
|--------|---|---|-------|-------|---|-------|-------|-------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | | SLRA5 | SLRA4 | | SLRA2 | SLRA1 | SLRA0 |
| Access | | | R/W | R/W | | R/W | R/W | R/W |
| Reset | | | 1 | 1 | | 1 | 1 | 1 |

Bits 4, 5 – SLRA_n Slew Rate Control on RA Pins

| Value | Description |
|-------|--------------------------------|
| 1 | PORT pin slew rate is limited |
| 0 | PORT pin slews at maximum rate |

Bits 0, 1, 2 – SLRA_n Slew Rate Control on RA Pins

| Value | Description |
|-------|--------------------------------|
| 1 | PORT pin slew rate is limited |
| 0 | PORT pin slews at maximum rate |

14.6.20 SLRCONB

Name: SLRCONB

Address: 0x1F46

Slew Rate Control Register

| | | | | | | | | |
|--------|-------|-------|-------|-------|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | SLRB7 | SLRB6 | SLRB5 | SLRB4 | | | | |
| Access | R/W | R/W | R/W | R/W | | | | |
| Reset | 1 | 1 | 1 | 1 | | | | |

Bits 4, 5, 6, 7 – SLRBn Slew Rate Control on RB Pins

| Value | Description |
|-------|--------------------------------|
| 1 | PORT pin slew rate is limited |
| 0 | PORT pin slews at maximum rate |

14.6.21 SLRCONC

Name: SLRCONC

Address: 0x1F51

Slew Rate Control Register

| | | | | | | | | |
|--------|-------|-------|-------|-------|-------|-------|-------|-------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | SLRC7 | SLRC6 | SLRC5 | SLRC4 | SLRC3 | SLRC2 | SLRC1 | SLRC0 |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Bits 0, 1, 2, 3, 4, 5, 6, 7 – SLRCn Slew Rate Control on RC Pins

| Value | Description |
|-------|--------------------------------|
| 1 | PORT pin slew rate is limited |
| 0 | PORT pin slews at maximum rate |

Note: Bits SLRC6 and SLRC7 available on 20-pin or higher pin count devices only; Bits unimplemented for lower pin count devices.

14.6.22 INLVLA

Name: INLVLA

Address: 0x1F3C

Input Level Control Register

| | | | | | | | | |
|--------|---|---|---------|---------|---------|---------|---------|---------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | | INLVLA5 | INLVLA4 | INLVLA3 | INLVLA2 | INLVLA1 | INLVLA0 |
| Access | | | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | | 1 | 1 | 1 | 1 | 1 | 1 |

Bits 0, 1, 2, 3, 4, 5 – INLVLA_n Input Level Select on RA Pins

| Value | Description |
|-------|---|
| 1 | ST input used for port reads and interrupt-on-change |
| 0 | TTL input used for port reads and interrupt-on-change |

14.6.23 INLVLB

Name: INLVLB

Address: 0x1F47

Input Level Control Register

| | | | | | | | | |
|--------|---------|---------|---------|---------|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | INLVLB7 | INLVLB6 | INLVLB5 | INLVLB4 | | | | |
| Access | R/W | R/W | R/W | R/W | | | | |
| Reset | 1 | 1 | 1 | 1 | | | | |

Bits 4, 5, 6, 7 – INLVLBn Input Level Select on RB Pins

| Value | Description |
|-------|---|
| 1 | ST input used for port reads and interrupt-on-change |
| 0 | TTL input used for port reads and interrupt-on-change |

14.6.24 INLVLC

Name: INLVLC

Address: 0x1F52

Input Level Control Register

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|---------|---------|---------|---------|---------|---------|---------|---------|
| | INLVLC7 | INLVLC6 | INLVLC5 | INLVLC4 | INLVLC3 | INLVLC2 | INLVLC1 | INLVLC0 |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Bits 0, 1, 2, 3, 4, 5, 6, 7 – INLVLCn Input Level Select on RC Pins

| Value | Description |
|-------|---|
| 1 | ST input used for port reads and interrupt-on-change |
| 0 | TTL input used for port reads and interrupt-on-change |

Note: Bits INLVLC6 and INLVLC7 available on 20-pin or higher pin count devices only; Bits unimplemented for lower pin count devices.

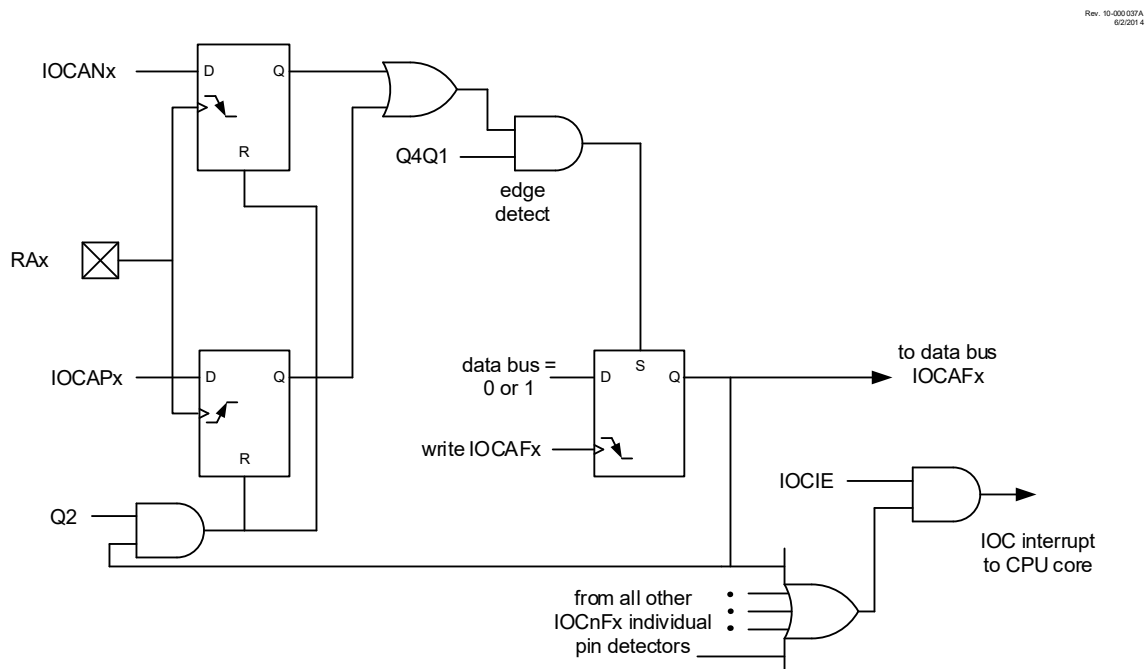
15. IOC - Interrupt-On-Change

All pins on all ports can be configured to operate as Interrupt-On-Change (IOC) pins. An interrupt can be generated by detecting a signal that has either a rising edge or a falling edge. Any individual pin, or combination of pins, can be configured to generate an interrupt. The interrupt-on-change module has the following features:

- Interrupt-on-Change enable
- Individual pin configuration
- Rising and falling edge detection
- Individual pin interrupt flags

Figure 15-1 is a block diagram of the IOC module.

Figure 15-1. Interrupt-on-Change Block Diagram (PORTA Example)



15.1 Enabling the Module

To allow individual port pins to generate an interrupt, the IOCFIE bit of the PIE0 register must be set. If the IOCFIE bit is disabled, the edge detection on the pin will still occur, but an interrupt will not be generated.

Related Links

[7.7.2 PIE0](#)

15.2 Individual Pin Configuration

For each PORT pin, a rising edge detector and a falling edge detector are present. To enable a pin to detect a rising edge, the associated bit of the IOCFxP register is set. To enable a pin to detect a falling edge, the associated bit of the IOCFxN register is set.

A pin can be configured to detect rising and falling edges simultaneously by setting both associated bits of the IOCFxP and IOCFxN registers, respectively.

15.3 Interrupt Flags

The bits located in the IOCxF registers are status flags that correspond to the interrupt-on-change pins of each port. If an expected edge is detected on an appropriately enabled pin, then the status flag for that pin will be set, and an interrupt will be generated if the IOCIE bit is set. The IOCIF bit of the PIR0 register reflects the status of all IOCxF bits.

Related Links

[7.7.11 PIR0](#)

15.3.1 Clearing Interrupt Flags

The individual status flags, (IOCxF register bits), can be cleared by resetting them to zero. If another edge is detected during this clearing operation, the associated status flag will be set at the end of the sequence, regardless of the value actually being written.

In order to ensure that no detected edge is lost while clearing flags, only AND operations masking out known changed bits should be performed. The following sequence is an example of what should be performed.

Example 15-1. Clearing Interrupt Flags (PORTA Example)

```
MOVLW    0xFF
XORWF    IOCAF, W
ANDWF    IOCAF, F
```

15.4 Operation in Sleep

The interrupt-on-change interrupt sequence will wake the device from Sleep mode, if the IOCIE bit is set.

If an edge is detected while in Sleep mode, the IOCxF register will be updated prior to the first instruction executed out of Sleep.

15.5 Register Summary - Interrupt-on-Change

Note: PORTB associated registers as well as IOCCx6 and IOCCx7 bits are available for 20-pin or higher pin count devices only .

| Address | Name | Bit Pos. | | | | | | | | |
|---------------|----------|----------|--------|--------|--------|--------|--------|--------|--------|--------|
| 0x00 ... | Reserved | | | | | | | | | |
| 0x1F3C | | | | | | | | | | |
| 0x1F3D | IOCAP | 7:0 | | | IOCAP5 | IOCAP4 | IOCAP3 | IOCAP2 | IOCAP1 | IOCAP0 |
| 0x1F3E | IOCAN | 7:0 | | | IOCAN5 | IOCAN4 | IOCAN3 | IOCAN2 | IOCAN1 | IOCAN0 |
| 0x1F3F | IOCAF | 7:0 | | | IOCAF5 | IOCAF4 | IOCAF3 | IOCAF2 | IOCAF1 | IOCAF0 |
| 0x1F40 ... | Reserved | | | | | | | | | |
| 0x1F47 | | | | | | | | | | |
| 0x1F48 | IOCBP | 7:0 | IOCBP7 | IOCBP6 | IOCBP5 | IOCBP4 | | | | |
| 0x1F49 | IOCBN | 7:0 | IOCBN7 | IOCBN6 | IOCBN5 | IOCBN4 | | | | |
| 0x1F4A | IOCBF | 7:0 | IOCBF7 | IOCBF6 | IOCBF5 | IOCBF4 | | | | |
| 0x1F4B ... | Reserved | | | | | | | | | |
| 0x1F52 | | | | | | | | | | |
| 0x1F53 | IOCCP | 7:0 | IOCCP7 | IOCCP6 | IOCCP5 | IOCCP4 | IOCCP3 | IOCCP2 | IOCCP1 | IOCCP0 |
| 0x1F54 | IOCCN | 7:0 | IOCCN7 | IOCCN6 | IOCCN5 | IOCCN4 | IOCCN3 | IOCCN2 | IOCCN1 | IOCCN0 |
| 0x1F55 | IOCCF | 7:0 | IOCCF7 | IOCCF6 | IOCCF5 | IOCCF4 | IOCCF3 | IOCCF2 | IOCCF1 | IOCCF0 |

15.6 Register Definitions: Interrupt-on-Change Control

15.6.1 IOCAP

Name: IOCAP
Address: 0x1F3D

Interrupt-on-Change Positive Edge Register

| | | | | | | | | |
|--------|---|---|--------|--------|--------|--------|--------|--------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | | IOCAP5 | IOCAP4 | IOCAP3 | IOCAP2 | IOCAP1 | IOCAP0 |
| Access | | | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 |

Bits 0, 1, 2, 3, 4, 5 – IOCAPn Interrupt-on-Change Positive Edge Enable bits

| Value | Description |
|-------|---|
| 1 | Interrupt-on-Change enabled on the IOCA pin for a positive-going edge. Associated Status bit and interrupt flag will be set upon detecting an edge. |
| 0 | Interrupt-on-Change disabled for the associated pin. |

Note:

1. IOCAP0 and IOCAP1 are not available for use if the debugger is enabled.
2. If MCLRE = 1 or LVP = 1, port functionality is disabled and IOCAP3 is not available.

15.6.2 IOCAN

Name: IOCAN
Address: 0x1F3E

Interrupt-on-Change Negative Edge Register

| | | | | | | | | |
|--------|---|---|--------|--------|--------|--------|--------|--------|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit | | | IOCAN5 | IOCAN4 | IOCAN3 | IOCAN2 | IOCAN1 | IOCAN0 |
| Access | | | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 |

Bits 0, 1, 2, 3, 4, 5 – IOCANn Interrupt-on-Change Negative Edge Enable bits

| Value | Description |
|-------|---|
| 1 | Interrupt-on-Change enabled on the IOCA pin for a negative-going edge. Associated Status bit and interrupt flag will be set upon detecting an edge. |
| 0 | Interrupt-on-Change disabled for the associated pin |

Note:

1. IOCAN0 and IOCAN1 are not available for use if the debugger is enabled.
2. If MCLRE = 1 or LVP = 1, port functionality is disabled and IOCAN3 is not available.

15.6.3 IOCAF

Name: IOCAF

Address: 0x1F3F

PORTA Interrupt-on-Change Flag Register

| | | | | | | | | |
|--------|---|---|--------|--------|--------|--------|--------|--------|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit | | | IOCAF5 | IOCAF4 | IOCAF3 | IOCAF2 | IOCAF1 | IOCAF0 |
| Access | | | R/W/HS | R/W/HS | R/W/HS | R/W/HS | R/W/HS | R/W/HS |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 |

Bits 0, 1, 2, 3, 4, 5 – IOCAF_n Interrupt-on-Change Flag bits

| Value | Condition | Description |
|-------|---------------------------|---|
| 1 | IOCAP[n]=1 | A positive edge was detected on the RA[n] pin |
| 1 | IOCAN[n]=1 | A negative edge was detected on the RA[n] pin |
| 0 | IOCAP[n]=x and IOCAN[n]=x | No change was detected, or the user cleared the detected change |

Note:

1. IOCAF0 and IOCAF1 are not available for use if the debugger is enabled.
2. If MCLRE = 1 or LVP = 1, port functionality is disabled and IOCAF3 is not available.

15.6.4 IOCBP

Name: IOCBP
Address: 0x1F48

Interrupt-on-Change Positive Edge Register

| | | | | | | | | |
|--------|--------|--------|--------|--------|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | IOCBP7 | IOCBP6 | IOCBP5 | IOCBP4 | | | | |
| Access | R/W | R/W | R/W | R/W | | | | |
| Reset | 0 | 0 | 0 | 0 | | | | |

Bits 4, 5, 6, 7 – IOCBPn Interrupt-on-Change Positive Edge Enable bits

| Value | Description |
|-------|---|
| 1 | Interrupt-on-Change enabled on the IOCB pin for a positive-going edge. Associated Status bit and interrupt flag will be set upon detecting an edge. |
| 0 | Interrupt-on-Change disabled for the associated pin. |

Note: PORTB associated registers are available on 20-pin or higher pin count devices only.

15.6.5 IOCBN

Name: IOCBN
 Address: 0x1F49

Interrupt-on-Change Negative Edge Register

| | | | | | | | | |
|--------|--------|--------|--------|--------|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | IOCBN7 | IOCBN6 | IOCBN5 | IOCBN4 | | | | |
| Access | R/W | R/W | R/W | R/W | | | | |
| Reset | 0 | 0 | 0 | 0 | | | | |

Bits 4, 5, 6, 7 – IOCBNn Interrupt-on-Change Negative Edge Enable bits

| Value | Description |
|-------|---|
| 1 | Interrupt-on-Change enabled on the IOCB pin for a negative-going edge. Associated Status bit and interrupt flag will be set upon detecting an edge. |
| 0 | Interrupt-on-Change disabled for the associated pin |

Note: PORTB associated registers are available on 20-pin or higher pin count devices only.

15.6.6 IOCBF

Name: IOCBF
Address: 0x1F4A

PORTB Interrupt-on-Change Flag Register

| | | | | | | | | |
|--------|--------|--------|--------|--------|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | IOCBF7 | IOCBF6 | IOCBF5 | IOCBF4 | | | | |
| Access | R/W/HS | R/W/HS | R/W/HS | R/W/HS | | | | |
| Reset | 0 | 0 | 0 | 0 | | | | |

Bits 4, 5, 6, 7 – IOCBFn Interrupt-on-Change Flag bits

| Value | Condition | Description |
|-------|---------------------------|---|
| 1 | IOCBP[n]=1 | A positive edge was detected on the RB[n] pin |
| 1 | IOCBN[n]=1 | A negative edge was detected on the RB[n] pin |
| 0 | IOCBP[n]=x and IOCBN[n]=x | No change was detected, or the user cleared the detected change |

Note: PORTB associated registers are available on 20-pin or higher pin count devices only.

15.6.7 IOCCP

Name: IOCCP
Address: 0x1F53

Interrupt-on-Change Positive Edge Register

| | | | | | | | | |
|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | IOCCP7 | IOCCP6 | IOCCP5 | IOCCP4 | IOCCP3 | IOCCP2 | IOCCP1 | IOCCP0 |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bits 0, 1, 2, 3, 4, 5, 6, 7 – IOCCPn Interrupt-on-Change Positive Edge Enable bits

| Value | Description |
|-------|---|
| 1 | Interrupt-on-Change enabled on the IOCC pin for a positive-going edge. Associated Status bit and interrupt flag will be set upon detecting an edge. |
| 0 | Interrupt-on-Change disabled for the associated pin. |

Note: IOCCP6 and IOCCP7 are available on 20-pin or higher pin count devices only.

15.6.8 IOCCN

Name: IOCCN
Address: 0x1F54

Interrupt-on-Change Negative Edge Register

| | | | | | | | | |
|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | IOCCN7 | IOCCN6 | IOCCN5 | IOCCN4 | IOCCN3 | IOCCN2 | IOCCN1 | IOCCN0 |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bits 0, 1, 2, 3, 4, 5, 6, 7 – IOCCNn Interrupt-on-Change Negative Edge Enable bits

| Value | Description |
|-------|---|
| 1 | Interrupt-on-Change enabled on the IOCC pin for a negative-going edge. Associated Status bit and interrupt flag will be set upon detecting an edge. |
| 0 | Interrupt-on-Change disabled for the associated pin |

Note: IOCCN6 and IOCCN7 are available on 20-pin or higher pin count devices only.

15.6.9 IOCCF

Name: IOCCF

Address: 0x1F55

PORTC Interrupt-on-Change Flag Register

| | | | | | | | | |
|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | IOCCF7 | IOCCF6 | IOCCF5 | IOCCF4 | IOCCF3 | IOCCF2 | IOCCF1 | IOCCF0 |
| Access | R/W/HS | R/W/HS | R/W/HS | R/W/HS | R/W/HS | R/W/HS | R/W/HS | R/W/HS |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bits 0, 1, 2, 3, 4, 5, 6, 7 – IOCCFn Interrupt-on-Change Flag bits

| Value | Condition | Description |
|-------|---------------------------|---|
| 1 | IOCCP[n]=1 | A positive edge was detected on the RC[n] pin |
| 1 | IOCCN[n]=1 | A negative edge was detected on the RC[n] pin |
| 0 | IOCCP[n]=x and IOCCN[n]=x | No change was detected, or the user cleared the detected change |

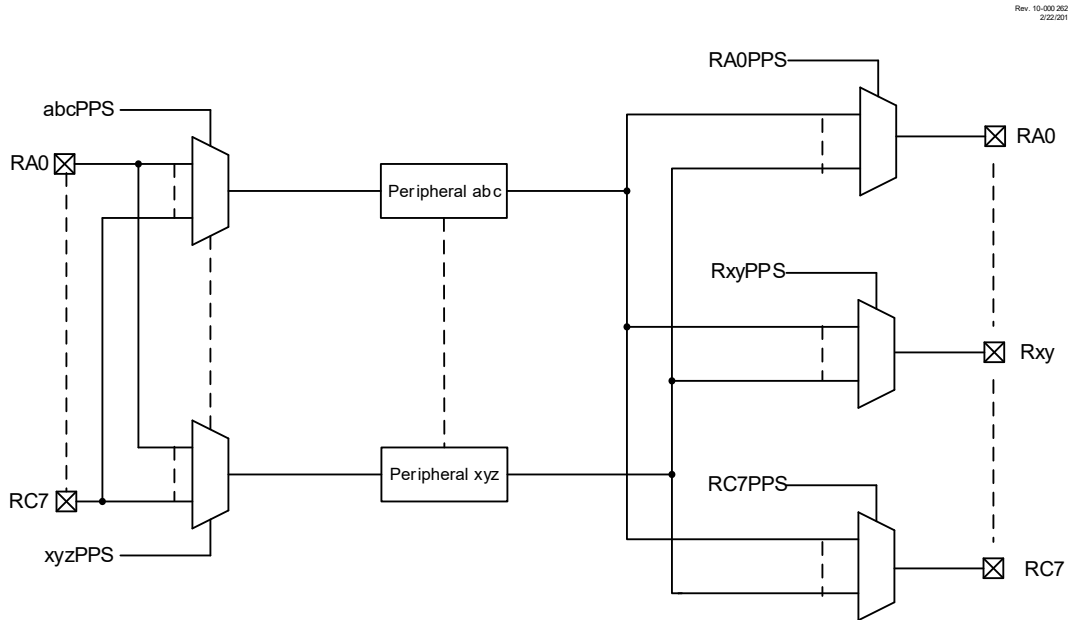
Note: IOCCF6 and IOCCF7 are available on 20-pin or higher pin count devices only.

16. PPS - Peripheral Pin Select Module

The Peripheral Pin Select (PPS) module connects peripheral inputs and outputs to the device I/O pins. Only digital signals are included in the selections.

All analog inputs and outputs remain fixed to their assigned pins. Input and output selections are independent as shown in the [Figure 16-1](#).

Figure 16-1. Simplified PPS Block Diagram



16.1 PPS Inputs

Each peripheral has an PPS register with which the input pin to the peripheral is selected. Although every peripheral has its own PPS input selection register, the selections are identical for every peripheral as shown in [16.9.2 xxxPPS register](#)

Not all ports are available for input as shown in [PPS Input Signal Routing Options](#).

Multiple peripherals can operate from the same source simultaneously. Port reads always return the pin level regardless of peripheral PPS selection. If a pin also has analog functions associated, the ANSEL bit for that pin must be cleared to enable the digital input buffer.



Important: The notation “xxx” in the generic register name is a place holder for the peripheral identifier. For example, xxx = INT for the INTPPS register.

Table 16-1. PPS Input Signal Routing Options

| Input Signal Name | Input Register Name | Default Location at POR (14/16-pin devices) | Reset Value (xxxPPS[4:0]) 14/16-pin devices | Default Location at POR (20-pin devices) | Reset Value (xxxPPS[4:0]) 20-pin devices | PORT From Which Input Is Available | | |
|-------------------|---------------------|---|---|--|--|------------------------------------|---|---|
| INT | INTPPS | RA2 | 00010 | RA2 | 00010 | A | B | — |

PIC16(L)F18426/46

PPS - Peripheral Pin Select Module

.....continued

| Input Signal Name | Input Register Name | Default Location at POR (14/16-pin devices) | Reset Value (xxxPPS[4:0]) 14/16-pin devices | Default Location at POR (20-pin devices) | Reset Value (xxxPPS[4:0]) 20-pin devices | PORT From Which Input Is Available | | |
|-------------------|---------------------|---|---|--|--|------------------------------------|---|---|
| T0CKI | T0CKIPPS | RA2 | 00010 | RA2 | 00010 | A | B | — |
| T1CKI | T1CKIPPS | RA5 | 00101 | RA5 | 00101 | A | — | C |
| T1G | T1GPPS | RA4 | 00100 | RA4 | 00100 | — | B | C |
| T2IN | T2INPPS | RA5 | 00101 | RA5 | 00101 | A | — | C |
| T3CKI | T3CKIPPS | RC5 | 10101 | RC5 | 10101 | — | B | C |
| T3G | T3GPPS | RC4 | 10100 | RC4 | 10100 | A | — | C |
| T4IN | T4INPPS | RC1 | 10001 | RC1 | 10001 | — | B | C |
| T5CKI | T5CKIPPS | RC0 | 10000 | RC0 | 10000 | A | — | C |
| T5G | T5GPPS | RC3 | 10011 | RC3 | 10011 | — | B | C |
| T6IN | T6INPPS | RC2 | 10010 | RC2 | 10010 | — | B | C |
| MDCARL | MDCARLPPS | RC2 | 10010 | RC2 | 10010 | A | — | C |
| MDCARH | MDCARHPPS | RC5 | 10101 | RC5 | 10101 | A | — | C |
| MDSRC | MDSRCPPS | RA1 | 00001 | RA1 | 00001 | A | — | C |
| CCP1IN | CCP1INPPS | RC5 | 10101 | RC5 | 10101 | — | B | C |
| CCP2IN | CCP2INPPS | RC3 | 10011 | RC3 | 10011 | — | B | C |
| CCP3IN | CCP3INPPS | RA2 | 00010 | RA2 | 00010 | — | B | C |
| CCP4IN | CCP4INPPS | RA4 | 00100 | RA4 | 00100 | — | B | C |
| CWG1IN | CWG1INPPS | RA2 | 00010 | RA2 | 00010 | — | B | C |
| CWG2IN | CWG2INPPS | RA2 | 00010 | RA2 | 00010 | — | B | C |
| CLCIN0 | CLCIN0PPS | RC3 | 10011 | RA2 | 00010 | A | — | C |
| CLCIN1 | CLCIN1PPS | RC4 | 10100 | RC3 | 10011 | A | — | C |
| CLCIN2 | CLCIN2PPS | RC1 | 10001 | RB4 | 01100 | — | B | C |
| CLCIN3 | CLCIN3PPS | RA5 | 00101 | RB5 | 01101 | — | B | C |
| ADACT | ADACTPPS | RC2 | 10010 | RC2 | 10010 | — | B | C |
| SCL1/SCK1 | SSP1CLKPPS | RC0 ⁽¹⁾ | 10000 | RB6 ⁽¹⁾ | 01110 | — | B | C |
| SDA1/SDI1 | SSP1DATPPS | RC1 ⁽¹⁾ | 10001 | RB4 ⁽¹⁾ | 01100 | — | B | C |
| SS1 | SS1PPS | RC3 | 10011 | RC6 | 10110 | A | — | C |
| RX1/DT1 | RX1PPS | RC5 | 10101 | RB5 | 01101 | — | B | C |
| TX1/CK1 | CK1PPS | RC4 | 10100 | RB7 | 01111 | — | B | C |
| SMT1SIG | SMT1SIGPPS | RC0 | 10000 | RC0 | 10000 | — | B | C |
| SMT1WIN | SMT1WINPPS | RA5 | 00101 | RA5 | 00101 | — | B | C |

Note:

1. Some pads are configured for I²C logic levels; clock and data signals may be assigned to any of these pins. Assignments to other pins (e.g., RA5) will operate, but logic levels will be standard TTL/ST as selected by the INLVL register.

Table 16-2. PPS Input Register Values

| Desired Input Pin | Value to Write to Register |
|-------------------|----------------------------|
| RC7 | 01 0111 |
| RC6 | 01 0110 |
| RC5 | 01 0101 |
| RC4 | 01 0100 |
| RC3 | 01 0011 |
| RC2 | 01 0010 |
| RC1 | 01 0001 |
| RC0 | 01 0000 |
| RB7 | 00 1111 |
| RB6 | 00 1110 |
| RB5 | 00 1101 |
| RB4 | 00 1100 |
| RA5 | 00 0101 |
| RA4 | 00 0100 |
| RA3 | 00 0011 |
| RA2 | 00 0010 |
| RA1 | 00 0001 |
| RA0 | 00 0000 |

16.2 PPS Outputs

Each I/O pin has a PPS register with which the pin output source is selected. With few exceptions, the port TRIS control associated with that pin retains control over the pin output driver. Peripherals that control the pin output driver as part of the peripheral operation will override the TRIS control as needed. These peripherals include:

- EUSART (synchronous operation)
- MSSP (I²C)

Although every pin has its own PPS peripheral selection register, the selections are identical for every pin as shown in the [16.9.3 RxyPPS](#) register.



Important: The notation “Rxy” is a place holder for the pin identifier. The ‘x’ holds the place of the PORT letter and the ‘y’ holds the place of the bit number. For example, Rxy = RA0 for the RA0PPS register.

[Table 16-3](#) shows detailed output routing options for each peripheral.

Table 16-3. PPS Output Signal Routing Options

| RxyPPS Register Value | Output Signal Name | Remappable to any PORT (14-pin devices) | Remappable to any PORT (20-pin devices) |
|-----------------------|--------------------|--|--|
| 10 0000 | ADCGRDB | X | X |
| 01 1111 | ADCGRDA | X | X |
| 01 1110 | CWG2D | X | X |
| 01 1101 | CWG2C | X | X |
| 01 1100 | CWG2B | X | X |
| 01 1011 | CWG2A | X | X |
| 01 1010 | DSM1OUT | X | X |
| 01 1001 | CLKR | X | X |
| 01 1000 | NCO1OUT | X | X |
| 01 0111 | TMR0OUT | X | X |
| 01 0100 | SDO1/SDA1 | X | X |
| 01 0011 | SCK1/SCL1 | X | X |
| 01 0010 | C2OUT | X | X |
| 01 0001 | C1OUT | X | X |
| 01 0000 | DT1 | X | X |
| 00 1111 | CK1/TX1 | X | X |
| 00 1110 | PWM7OUT | X | X |
| 00 1101 | PWM6OUT | X | X |
| 00 1100 | CCP4OUT | X | X |
| 00 1011 | CCP3OUT | X | X |
| 00 1010 | CCP2OUT | X | X |
| 00 1001 | CCP1OUT | X | X |
| 00 1000 | CWG1D | X | X |
| 00 0111 | CWG1C | X | X |
| 00 0110 | CWG1B | X | X |
| 00 0101 | CWG1A | X | X |
| 00 0100 | CLC4OUT | X | X |
| 00 0011 | CLC3OUT | X | X |
| 00 0010 | CLC2OUT | X | X |
| 00 0001 | CLC1OUT | X | X |

16.3 Bidirectional Pins

PPS selections for peripherals with bidirectional signals on a single pin must be made so that the PPS input and PPS output select the same pin. Peripherals that have bidirectional signals include:

- EUSART (DT/RXxPPS and TX/CKxPPS pins for synchronous operation)
- MSSP (I²C SDA/SSPxDATPPS and SCL/SSPxCLKPPS)



Important: The I²C default inputs, and a limited number of other alternate pins, are I²C and SMBus compatible. Clock and data signals can be routed to any pin, however pins without I²C compatibility will operate at standard TTL/ST logic levels as selected by the INLVL register. See the INLVL register for each port to determine which pins are I²C and SMBus compatible.

16.4 PPS Lock

The PPS includes a mode in which all input and output selections can be locked to prevent inadvertent changes. PPS selections are locked by setting the PPSLOCKED bit of the PPSLOCK register. Setting and clearing this bit requires a special sequence as an extra precaution against inadvertent changes. Examples of setting and clearing the PPSLOCKED bit are shown in the following examples.

Example 16-1. PPS Lock Sequence

```

; suspend interrupts
BCF     INTCON,GIE
BANKSEL PPSLOCK ; set bank
; required sequence, next 5 instructions
MOVLW  0x55
MOVWF  PPSLOCK
MOVLW  0xAA
MOVWF  PPSLOCK
; Set PPSLOCKED bit to disable writes or
BSF     PPSLOCK,PPSLOCKED
; restore interrupts
BSF     INTCON,GIE
    
```

Example 16-2. PPS Unlock Sequence

```

; suspend interrupts
BCF     INTCON,GIE
BANKSEL PPSLOCK ; set bank
; required sequence, next 5 instructions
MOVLW  0x55
MOVWF  PPSLOCK
MOVLW  0xAA
MOVWF  PPSLOCK
; Clear PPSLOCKED bit to enable writes
BCF     PPSLOCK,PPSLOCKED
; restore interrupts
BSF     INTCON,GIE
    
```

Note:

1. The PPSLOCK bit can only be set or cleared after the unlock sequence shown above.
2. If PPS1WAY = 1, the PPSLOCK bit cannot be cleared after it has been set.

16.5 PPS1WAY Bit

The PPS can be locked by setting the PPS1WAY bit of Configuration Word 2. When the PPS1WAY bit is set, the PPSLOCKED bit of the PPSLOCK register can be cleared and set only one time after a device Reset. Once the PPS registers are configured, user software sets the PPSLOCKED bit, preventing any further writes to the PPS registers. the PPS registers can be read at any time, regardless of the PPS1WAY or PPSLOCKED settings.

When the PPS1WAY bit is clear, the PPSLOCKED bit of the PPSLOCK register can be cleared and set multiple times during code execution, but requires the PPS lock/unlock sequence to be performed each time modifications to the PPS registers are made.

Related Links

[16.9.1 PPSLOCK](#)

16.6 Operation During Sleep

PPS input and output selections are unaffected by Sleep.

16.7 Effects of a Reset

A device Power-on-Reset (POR) clears all PPS input and output selections to their default values. All other Resets leave the selections unchanged. Default input selections are shown in [Table 16-1](#). The PPS1WAY is also removed.

16.8 Register Summary - PPS

Note: PORTB associated RxyPPS register as well as RC6PPS and RC7PPS registers are only available for 20-pin or higher pin count devices.

| Address | Name | Bit Pos. | | | | | | | |
|---------|------------|----------|--|--|--|-----------|--|--|-----------|
| 0x00 | Reserved | | | | | | | | |
| ... | | | | | | | | | |
| 0x1E8E | | | | | | | | | |
| 0x1E8F | PPSLOCK | 7:0 | | | | | | | PPSLOCKED |
| 0x1E90 | INTPPS | 7:0 | | | | PORT[1:0] | | | PIN[2:0] |
| 0x1E91 | T0CKIPPS | 7:0 | | | | PORT[1:0] | | | PIN[2:0] |
| 0x1E92 | T1CKIPPS | 7:0 | | | | PORT[1:0] | | | PIN[2:0] |
| 0x1E93 | T1GPPS | 7:0 | | | | PORT[1:0] | | | PIN[2:0] |
| 0x1E94 | T3CKIPPS | 7:0 | | | | PORT[1:0] | | | PIN[2:0] |
| 0x1E95 | T3GPPS | 7:0 | | | | PORT[1:0] | | | PIN[2:0] |
| 0x1E96 | T5CKIPPS | 7:0 | | | | PORT[1:0] | | | PIN[2:0] |
| 0x1E97 | T5GPPS | 7:0 | | | | PORT[1:0] | | | PIN[2:0] |
| 0x1E98 | Reserved | | | | | | | | |
| ... | | | | | | | | | |
| 0x1E9B | | | | | | | | | |
| 0x1E9C | T2INPPS | 7:0 | | | | PORT[1:0] | | | PIN[2:0] |
| 0x1E9D | T4INPPS | 7:0 | | | | PORT[1:0] | | | PIN[2:0] |
| 0x1E9E | T6INPPS | 7:0 | | | | PORT[1:0] | | | PIN[2:0] |
| 0x1E9F | Reserved | | | | | | | | |
| ... | | | | | | | | | |
| 0x1EA0 | | | | | | | | | |
| 0x1EA1 | CCP1PPS | 7:0 | | | | PORT[1:0] | | | PIN[2:0] |
| 0x1EA2 | CCP2PPS | 7:0 | | | | PORT[1:0] | | | PIN[2:0] |
| 0x1EA3 | CCP3PPS | 7:0 | | | | PORT[1:0] | | | PIN[2:0] |
| 0x1EA4 | CCP4PPS | 7:0 | | | | PORT[1:0] | | | PIN[2:0] |
| 0x1EA5 | Reserved | | | | | | | | |
| ... | | | | | | | | | |
| 0x1EA8 | | | | | | | | | |
| 0x1EA9 | SMT1WINPPS | 7:0 | | | | PORT[1:0] | | | PIN[2:0] |
| 0x1EAA | SMT1SIGPPS | 7:0 | | | | PORT[1:0] | | | PIN[2:0] |
| 0x1EAB | Reserved | | | | | | | | |
| ... | | | | | | | | | |
| 0x1EB0 | | | | | | | | | |
| 0x1EB1 | CWG1PPS | 7:0 | | | | PORT[1:0] | | | PIN[2:0] |
| 0x1EB2 | CWG2PPS | 7:0 | | | | PORT[1:0] | | | PIN[2:0] |
| 0x1EB3 | Reserved | | | | | | | | |
| ... | | | | | | | | | |
| 0x1EB7 | | | | | | | | | |
| 0x1EB8 | MDCARLPPS | 7:0 | | | | PORT[1:0] | | | PIN[2:0] |
| 0x1EB9 | MDCARHPPS | 7:0 | | | | PORT[1:0] | | | PIN[2:0] |
| 0x1EBA | MDSRCPPS | 7:0 | | | | PORT[1:0] | | | PIN[2:0] |
| 0x1EBB | CLCIN1PPS | 7:0 | | | | PORT[1:0] | | | PIN[2:0] |
| 0x1EBC | CLCIN2PPS | 7:0 | | | | PORT[1:0] | | | PIN[2:0] |
| 0x1EBD | CLCIN3PPS | 7:0 | | | | PORT[1:0] | | | PIN[2:0] |
| 0x1EBE | CLCIN4PPS | 7:0 | | | | PORT[1:0] | | | PIN[2:0] |
| 0x1EBF | Reserved | | | | | | | | |
| ... | | | | | | | | | |
| 0x1EC2 | | | | | | | | | |
| 0x1EC3 | ADACTPPS | 7:0 | | | | PORT[1:0] | | | PIN[2:0] |
| 0x1EC4 | Reserved | | | | | | | | |
| 0x1EC5 | SSP1CLKPPS | 7:0 | | | | PORT[1:0] | | | PIN[2:0] |
| 0x1EC6 | SSP1DATPPS | 7:0 | | | | PORT[1:0] | | | PIN[2:0] |
| 0x1EC7 | SSP1SSPPS | 7:0 | | | | PORT[1:0] | | | PIN[2:0] |
| 0x1EC8 | SSP2CLKPPS | 7:0 | | | | PORT[1:0] | | | PIN[2:0] |

PIC16(L)F18426/46

PPS - Peripheral Pin Select Module

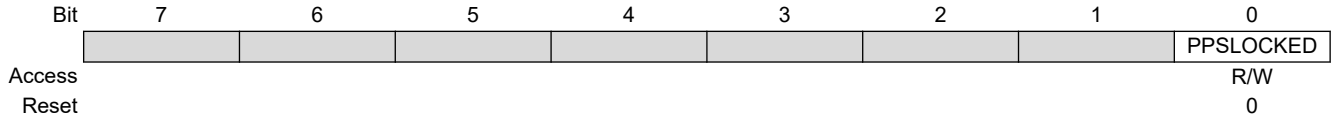
.....continued

| Address | Name | Bit Pos. | | | | | | | |
|---------|------------|----------|--|--|--|-----------|--|--|----------|
| 0x1EC9 | SSP2DATPPS | 7:0 | | | | PORT[1:0] | | | PIN[2:0] |
| 0x1ECA | SSP2SSPPS | 7:0 | | | | PORT[1:0] | | | PIN[2:0] |
| 0x1ECB | RX1PPS | 7:0 | | | | PORT[1:0] | | | PIN[2:0] |
| 0x1ECC | CK1PPS | 7:0 | | | | PORT[1:0] | | | PIN[2:0] |
| 0x1ECD | Reserved | | | | | | | | |
| ... | | | | | | | | | |
| 0x1F0F | | | | | | | | | |
| 0x1F10 | RA0PPS | 7:0 | | | | | | | PPS[5:0] |
| 0x1F11 | RA1PPS | 7:0 | | | | | | | PPS[5:0] |
| 0x1F12 | RA2PPS | 7:0 | | | | | | | PPS[5:0] |
| 0x1F13 | Reserved | | | | | | | | |
| 0x1F14 | RA4PPS | 7:0 | | | | | | | PPS[5:0] |
| 0x1F15 | RA5PPS | 7:0 | | | | | | | PPS[5:0] |
| 0x1F16 | Reserved | | | | | | | | |
| ... | | | | | | | | | |
| 0x1F1B | | | | | | | | | |
| 0x1F1C | RB4PPS | 7:0 | | | | | | | PPS[5:0] |
| 0x1F1D | RB5PPS | 7:0 | | | | | | | PPS[5:0] |
| 0x1F1E | RB6PPS | 7:0 | | | | | | | PPS[5:0] |
| 0x1F1F | RB7PPS | 7:0 | | | | | | | PPS[5:0] |
| 0x1F20 | RC0PPS | 7:0 | | | | | | | PPS[5:0] |
| 0x1F21 | RC1PPS | 7:0 | | | | | | | PPS[5:0] |
| 0x1F22 | RC2PPS | 7:0 | | | | | | | PPS[5:0] |
| 0x1F23 | RC3PPS | 7:0 | | | | | | | PPS[5:0] |
| 0x1F24 | RC4PPS | 7:0 | | | | | | | PPS[5:0] |
| 0x1F25 | RC5PPS | 7:0 | | | | | | | PPS[5:0] |
| 0x1F26 | RC6PPS | 7:0 | | | | | | | PPS[5:0] |
| 0x1F27 | RC7PPS | 7:0 | | | | | | | PPS[5:0] |

16.9 Register Definitions: PPS Input and Output Selection

16.9.1 PPS Lock Register

Name: PPSLOCK
Address: 0x1E8F



Bit 0 – PPSLOCKED PPS Locked bit

| Value | Description |
|-------|---|
| 1 | PPS is locked. PPS selections can not be changed. |
| 0 | PPS is not locked. PPS selections can be changed. |

16.9.2 Peripheral xxx Input Selection

Name: xxxPPS

| | | | | | | | | |
|--------|---|---|---|-----------|-----|----------|-----|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | | | PORT[1:0] | | PIN[2:0] | | |
| Access | | | | R/W | R/W | R/W | R/W | R/W |
| Reset | | | | x | x | g | g | g |

Bits 4:3 – PORT[1:0] Peripheral xxx Input PORT Selection bits
 See the [Table 16-1](#) for a list of available ports and default pin locations.

| Value | Description |
|-------|-------------|
| 10 | PORTC |
| 01 | PORTB |
| 00 | PORTA |

Bits 2:0 – PIN[2:0] Peripheral xxx Input Pin Selection bits

| Value | Description |
|-------|--|
| 111 | Peripheral input is from PORTx Pin 7 (Rx7) |
| 110 | Peripheral input is from PORTx Pin 6 (Rx6) |
| 101 | Peripheral input is from PORTx Pin 5 (Rx5) |
| 100 | Peripheral input is from PORTx Pin 4 (Rx4) |
| 011 | Peripheral input is from PORTx Pin 3 (Rx3) |
| 010 | Peripheral input is from PORTx Pin 2 (Rx2) |
| 001 | Peripheral input is from PORTx Pin 1 (Rx1) |
| 000 | Peripheral input is from PORTx Pin 0 (Rx0) |



Important:
 PORTB is available only for 20-pin or higher pin count devices.

Related Links

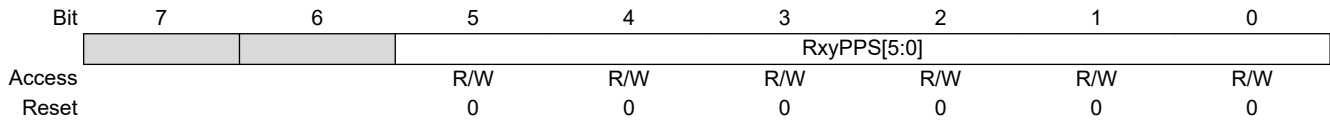
[16.1 PPS Inputs](#)

16.9.3 Pin Rxy Output Source Selection Register

Name: RxyPPS



Important: See [16.8 Register Summary - PPS](#) for the address offset of each individual register.



Bits 5:0 – RxyPPS[5:0] Pin Rxy Output Source Selection bits
 See [Table 16-3](#) for details about source selection.

Related Links

[16.2 PPS Outputs](#)

17. CLC - Configurable Logic Cell

The Configurable Logic Cell (CLC) module provides programmable logic that operates outside the speed limitations of software execution. The logic cell takes up to 64 input signals and, through the use of configurable gates, reduces the 64 inputs to four logic lines that drive one of eight selectable single-output logic functions.

Input sources are a combination of the following:

- I/O pins
- Internal clocks
- Peripherals
- Register bits

The output can be directed internally to peripherals and to an output pin.

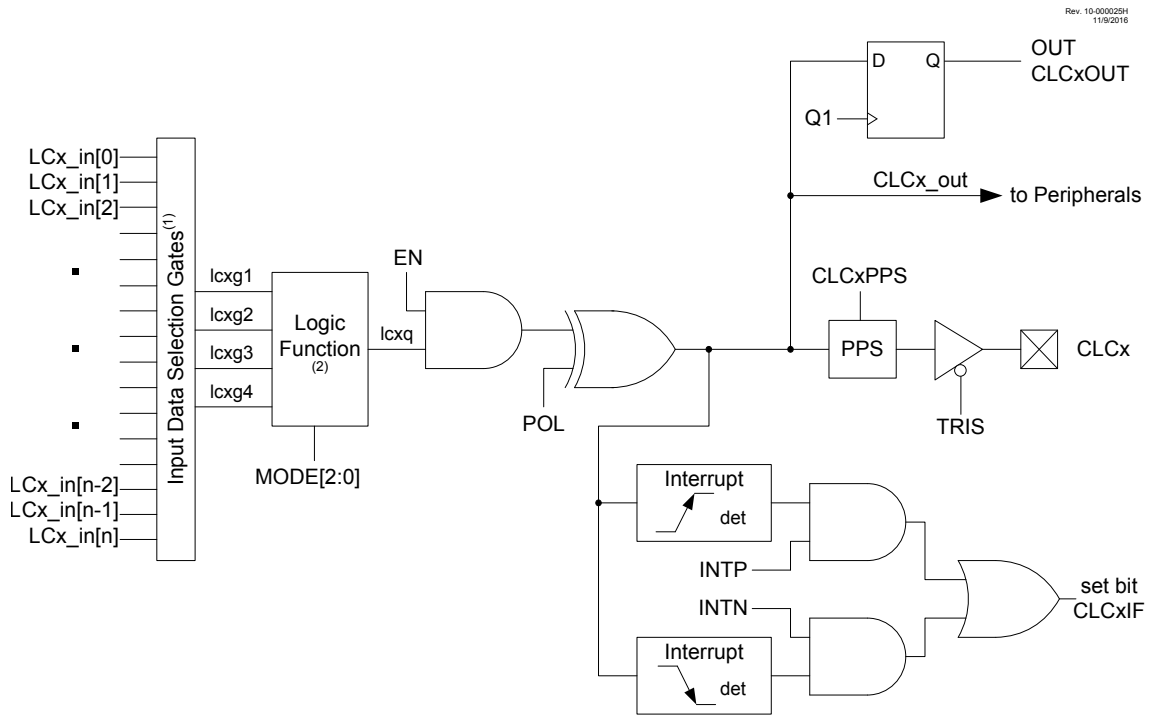


Important: There are several CLC instances on this device. Throughout this section, the lower case 'x' in register names is a generic reference to the CLC instance number. For example, the first instance of the control register is CLC1CON and is generically described in this chapter as CLCxCON.

The following figure is a simplified diagram showing signal flow through the CLC. Possible configurations include:

- Combinatorial Logic:
 - AND
 - NAND
 - AND-OR
 - AND-OR-INVERT
 - OR-XOR
 - OR-XNOR
- Latches:
 - S-R
 - Clocked D with Set and Reset
 - Transparent D with Set and Reset
 - Clocked J-K with Reset

Figure 17-1. CLC Simplified Block Diagram



Note:

1. See [Figure 17-2](#) for input data selection and gating.
2. See [Figure 17-3](#) for programmable logic functions.

17.1 CLC Setup

Programming the CLC module is performed by configuring the four stages in the logic signal flow. The four stages are:

- Data selection
- Data gating
- Logic function selection
- Output polarity

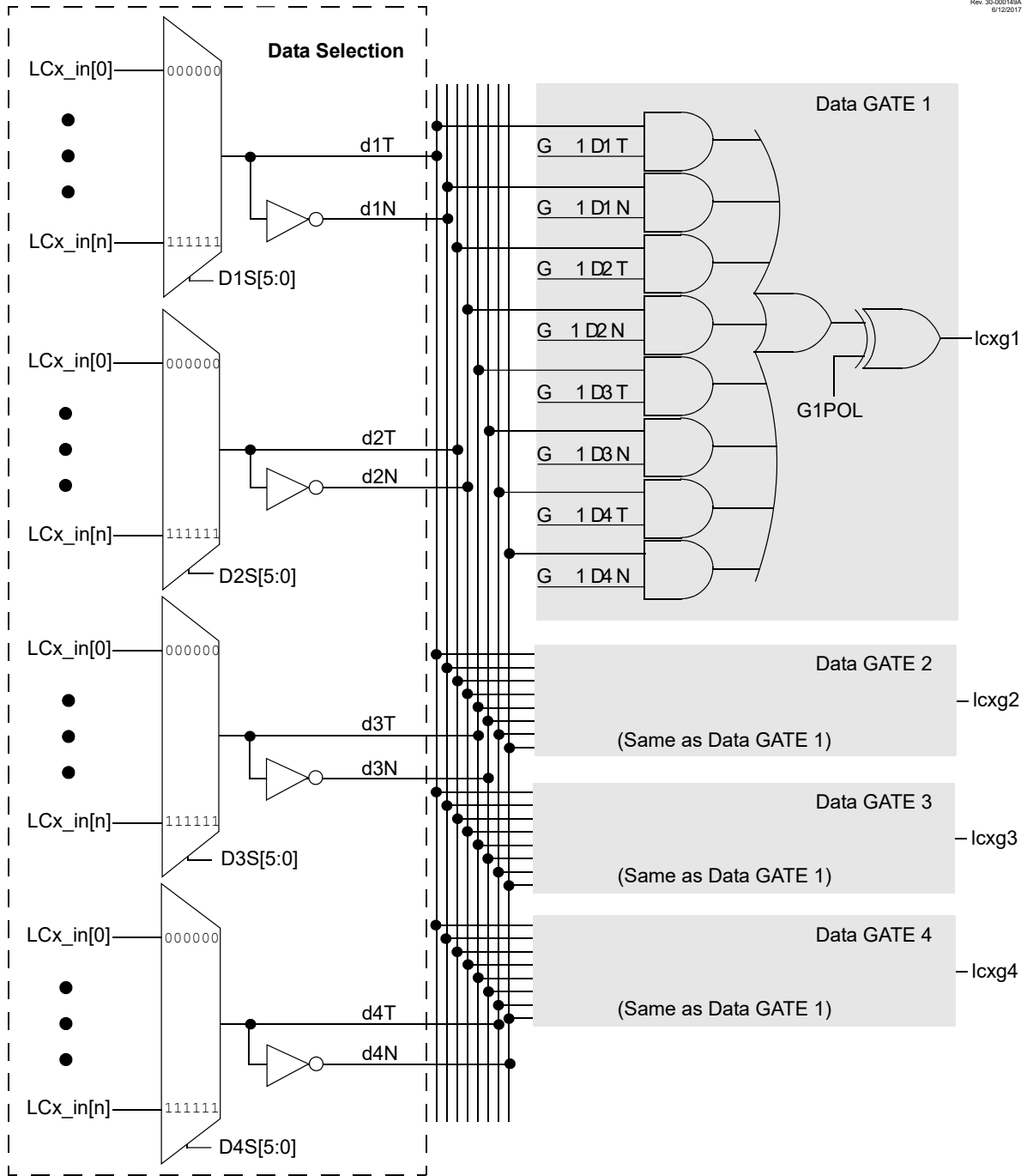
Each stage is setup at run time by writing to the corresponding CLC Special Function Registers. This has the added advantage of permitting logic reconfiguration on-the-fly during program execution.

17.1.1 Data Selection

There are 64 signals available as inputs to the configurable logic. Four 64-input multiplexers are used to select the inputs to pass on to the next stage.

Data selection is through four multiplexers as indicated on the left side of the following diagram. Data inputs in the figure are identified by a generic numbered input name.

Figure 17-2. Input Data Selection and Gating



Note: All controls are undefined at power-up.

The [CLCxSEL0](#), [CLCxSEL1](#), [CLCxSEL2](#), and [CLCxSEL3](#) Data Input Source registers correlate the generic input name to the actual signal for each CLC module. The column labeled 'DyS Value' indicates the MUX selection code for the selected data input. DyS is an abbreviation for the MUX select input codes: D1S through D4S where 'y' is the gate number.



Important: Data selections are undefined at power-up.

17.1.2 Data Gating

Outputs from the input multiplexers are directed to the desired logic function input through the data gating stage. Each data gate can direct any combination of the four selected inputs.

The gate stage is more than just signal direction. The gate can be configured to direct each input signal as inverted or non-inverted data. Directed signals are ANDed together in each gate. The output of each gate can be inverted before going on to the logic function stage.

The gating is in essence a 1-to-4 input AND/NAND/OR/NOR gate. When every input is inverted and the output is inverted, the gate is an AND of all enabled data inputs. When the inputs and output are not inverted, the gate is an OR of all enabled inputs.

The following table summarizes the basic logic that can be obtained in gate 1 by using the gate logic select bits. The table shows the logic of four input variables, but each gate can be configured to use less than four. If no inputs are selected, the output will be zero or one, depending on the gate output polarity bit.

Table 17-1. Data Gating Logic

| CLCxGLSy | GyPOL | Gate Logic |
|----------|-------|------------|
| 0x55 | 1 | AND |
| 0x55 | 0 | NAND |
| 0xAA | 1 | NOR |
| 0xAA | 0 | OR |
| 0x00 | 0 | Logic 0 |
| 0x00 | 1 | Logic 1 |

It is possible (but not recommended) to select both the true and negated values of an input. When this is done, the gate output is zero, regardless of the other inputs, but may emit logic glitches (transient-induced pulses). If the output of the channel must be zero or one, the recommended method is to set all gate bits to zero and use the gate polarity bit to set the desired level.

Data gating is configured with the logic gate select registers as follows:

- Gate 1: [CLCxGLS0](#)
- Gate 2: [CLCxGLS1](#)
- Gate 3: [CLCxGLS2](#)
- Gate 4: [CLCxGLS3](#)

Register number suffixes are different than the gate numbers because other variations of this module have multiple gate selections in the same register.

Data gating is indicated in the right side of [Figure 17-2](#). Only one gate is shown in detail. The remaining three gates are configured identically with the exception that the data enables correspond to the enables for that gate.

17.1.3 Logic Function

There are eight available logic functions including:

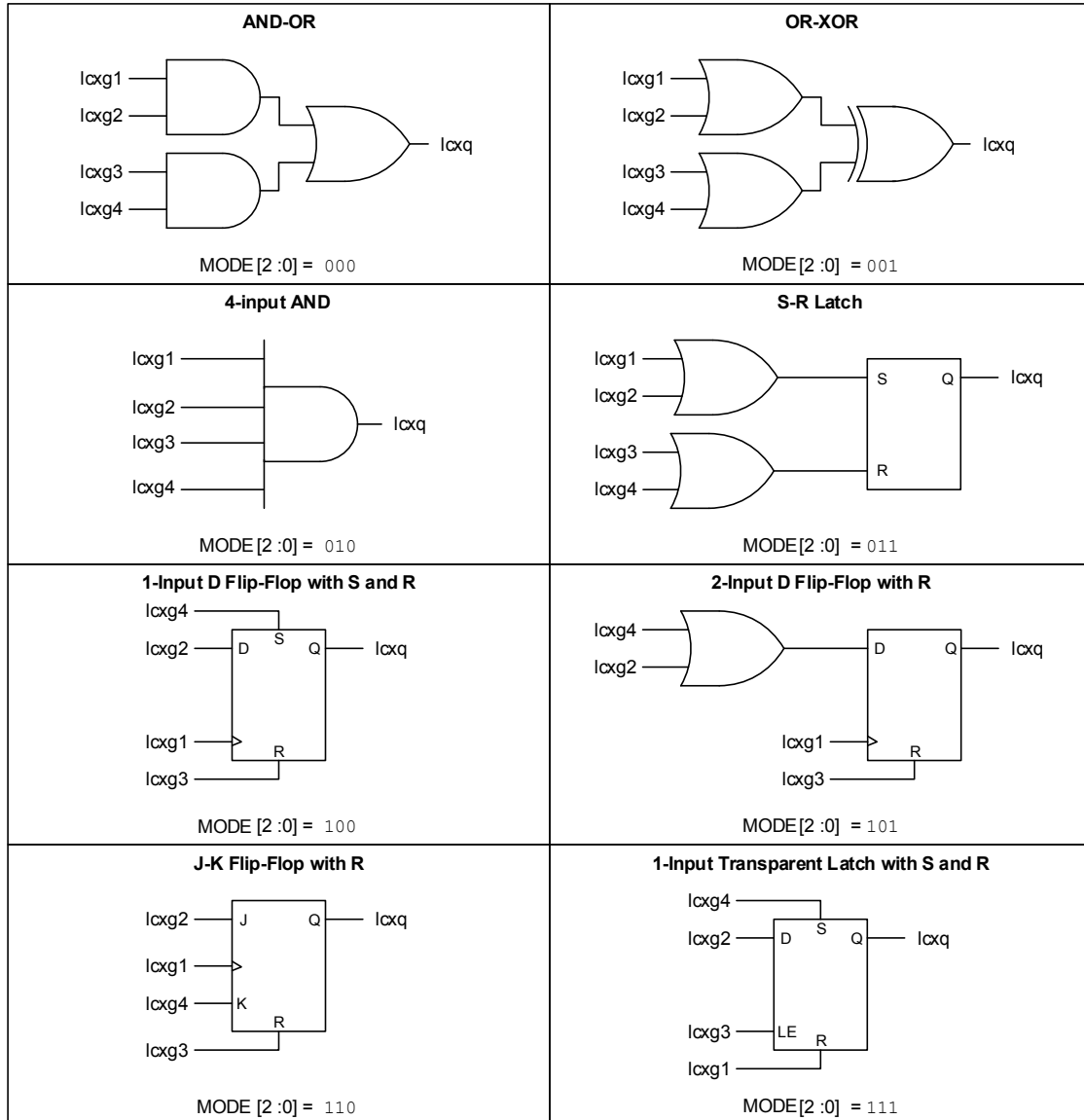
- AND-OR
- OR-XOR
- AND
- S-R Latch
- D Flip-Flop with Set and Reset

- D Flip-Flop with Reset
- J-K Flip-Flop with Reset
- Transparent Latch with Set and Reset

Logic functions are shown in the following diagram. Each logic function has four inputs and one output. The four inputs are the four data gate outputs of the previous stage. The output is fed to the inversion stage and from there to other peripherals, an output pin, and back to the CLC itself.

Figure 17-3. Programmable Logic Functions

Rev: 10-000122B
9/13/2016



17.1.4 Output Polarity

The last stage in the CLC is the output polarity. Setting the **POL** bit in the CLCxPOL register inverts the output signal from the logic stage. Changing the polarity while the interrupts are enabled will cause an interrupt for the resulting output transition.

17.2 CLC Interrupts

An interrupt will be generated upon a change in the output value of the CLCx when the appropriate interrupt enables are set. A rising edge detector and a falling edge detector are present in each CLC for this purpose.

The CLCxIF bit of the associated PIR register will be set when either edge detector is triggered and its associated enable bit is set. The [INTP](#) enables rising edge interrupts and the [INTN](#) bit enables falling edge interrupts.

To fully enable the interrupt, set the following bits:

- CLCxIE bit of the respective PIE register
- [INTP](#) bit (for a rising edge detection)
- [INTN](#) bit (for a falling edge detection)
- If priority interrupts are not used
 - Clear the IPEN bit of the INTCON register
 - Set the GIE bit of the INTCON register
 - Set the PEIE bit of the INTCON register
- If the CLC is a high-priority interrupt
 - Set the IPEN bit of the INTCON register
 - Set the CLCxIP bit of the respective IPR register
 - Set the GIEH bit of the INTCON register
- If the CLC is a low-priority interrupt
 - Set the IPEN bit of the INTCON register
 - Clear the CLCxIP bit of the respective IPR register
 - Set the GIEL bit of the INTCON register

The CLCxIF bit of the respective PIR register, must be cleared in software as part of the interrupt service. If another edge is detected while this flag is being cleared, the flag will still be set at the end of the sequence.

Related Links

[5.8.10 INTCON](#)

[7.7.7 PIE5](#)

[7.7.16 PIR5](#)

17.3 Output Mirror Copies

Mirror copies of all CLCxOUT bits are contained in the [CLCDATA](#) register. Reading this register reads the outputs of all CLCs simultaneously. This prevents any reading skew introduced by testing or reading the OUT bits in the individual CLCxCON registers.

17.4 Effects of a Reset

The CLCxCON register is cleared to zero as the result of a Reset. All other selection and gating values remain unchanged.

17.5 Operation During Sleep

The CLC module operates independently from the system clock and will continue to run during Sleep, provided that the input sources selected remain active.

The HFINTOSC remains active during Sleep when the CLC module is enabled and the HFINTOSC is selected as an input source, regardless of the system clock source selected.

In other words, if the HFINTOSC is simultaneously selected as the system clock and as a CLC input source, when the CLC is enabled, the CPU will go Idle during Sleep, but the CLC will continue to operate and the HFINTOSC will remain active.

This will have a direct effect on the Sleep mode current.

17.6 CLC Setup Steps

The following steps should be followed when setting up the CLC:

- Disable CLC by clearing the [EN](#) bit.
- Select desired inputs using the [CLCxSELO](#) through [CLCxSEL3](#) registers.
- Clear any associated ANSEL bits.
- Set all TRIS bits associated with inputs.
- Enable the chosen inputs through the four gates using the [CLCxGLS0](#) through [CLCxGLS3](#) registers.
- Select the gate output polarities with the [GyPOL](#) bits
- Select the desired logic function with the [MODE](#) bits
- Select the desired polarity of the logic output with the [POL](#) bit. (This step may be combined with the previous gate output polarity step).
- If driving a device pin, set the desired pin PPS control register and also clear the TRIS bit corresponding to that output.
- Configure the interrupts (optional). See [17.2 CLC Interrupts](#)
- Enable the CLC by setting the [EN](#) bit.

17.7 Register Summary - CLC Control

| Address | Name | Bit Pos. | | | | | | | | | |
|-------------|----------|----------|-------|-------|-------|-------|-------|----------|---------|-----------|---------|
| 0x00 ... | Reserved | | | | | | | | | | |
| 0x1E0E | | | | | | | | | | | |
| 0x1E0F | CLCDATA | 7:0 | | | | | | MLC4OUT | MLC3OUT | MLC2OUT | MLC1OUT |
| 0x1E10 | CLC1CON | 7:0 | EN | | OUT | INTP | INTN | | | MODE[2:0] | |
| 0x1E11 | CLC1POL | 7:0 | POL | | | | | G4POL | G3POL | G2POL | G1POL |
| 0x1E12 | CLC1SEL0 | 7:0 | | | | | | D1S[5:0] | | | |
| 0x1E13 | CLC1SEL1 | 7:0 | | | | | | D2S[5:0] | | | |
| 0x1E14 | CLC1SEL2 | 7:0 | | | | | | D3S[5:0] | | | |
| 0x1E15 | CLC1SEL3 | 7:0 | | | | | | D4S[5:0] | | | |
| 0x1E16 | CLC1GLS0 | 7:0 | G1D4T | G1D4N | G1D3T | G1D3N | G1D2T | G1D2N | G1D1T | G1D1N | |
| 0x1E17 | CLC1GLS1 | 7:0 | G2D4T | G2D4N | G2D3T | G2D3N | G2D2T | G2D2N | G2D1T | G2D1N | |
| 0x1E18 | CLC1GLS2 | 7:0 | G3D4T | G3D4N | G3D3T | G3D3N | G3D2T | G3D2N | G3D1T | G3D1N | |
| 0x1E19 | CLC1GLS3 | 7:0 | G4D4T | G4D4N | G4D3T | G4D3N | G4D2T | G4D2N | G4D1T | G4D1N | |
| 0x1E1A | CLC2CON | 7:0 | EN | | OUT | INTP | INTN | | | MODE[2:0] | |
| 0x1E1B | CLC2POL | 7:0 | POL | | | | | G4POL | G3POL | G2POL | G1POL |
| 0x1E1C | CLC2SEL0 | 7:0 | | | | | | D1S[5:0] | | | |
| 0x1E1D | CLC2SEL1 | 7:0 | | | | | | D2S[5:0] | | | |
| 0x1E1E | CLC2SEL2 | 7:0 | | | | | | D3S[5:0] | | | |
| 0x1E1F | CLC2SEL3 | 7:0 | | | | | | D4S[5:0] | | | |
| 0x1E20 | CLC2GLS0 | 7:0 | G1D4T | G1D4N | G1D3T | G1D3N | G1D2T | G1D2N | G1D1T | G1D1N | |
| 0x1E21 | CLC2GLS1 | 7:0 | G2D4T | G2D4N | G2D3T | G2D3N | G2D2T | G2D2N | G2D1T | G2D1N | |
| 0x1E22 | CLC2GLS2 | 7:0 | G3D4T | G3D4N | G3D3T | G3D3N | G3D2T | G3D2N | G3D1T | G3D1N | |
| 0x1E23 | CLC2GLS3 | 7:0 | G4D4T | G4D4N | G4D3T | G4D3N | G4D2T | G4D2N | G4D1T | G4D1N | |
| 0x1E24 | CLC3CON | 7:0 | EN | | OUT | INTP | INTN | | | MODE[2:0] | |
| 0x1E25 | CLC3POL | 7:0 | POL | | | | | G4POL | G3POL | G2POL | G1POL |
| 0x1E26 | CLC3SEL0 | 7:0 | | | | | | D1S[5:0] | | | |
| 0x1E27 | CLC3SEL1 | 7:0 | | | | | | D2S[5:0] | | | |
| 0x1E28 | CLC3SEL2 | 7:0 | | | | | | D3S[5:0] | | | |
| 0x1E29 | CLC3SEL3 | 7:0 | | | | | | D4S[5:0] | | | |
| 0x1E2A | CLC3GLS0 | 7:0 | G1D4T | G1D4N | G1D3T | G1D3N | G1D2T | G1D2N | G1D1T | G1D1N | |
| 0x1E2B | CLC3GLS1 | 7:0 | G2D4T | G2D4N | G2D3T | G2D3N | G2D2T | G2D2N | G2D1T | G2D1N | |
| 0x1E2C | CLC3GLS2 | 7:0 | G3D4T | G3D4N | G3D3T | G3D3N | G3D2T | G3D2N | G3D1T | G3D1N | |
| 0x1E2D | CLC3GLS3 | 7:0 | G4D4T | G4D4N | G4D3T | G4D3N | G4D2T | G4D2N | G4D1T | G4D1N | |
| 0x1E2E | CLC4CON | 7:0 | EN | | OUT | INTP | INTN | | | MODE[2:0] | |
| 0x1E2F | CLC4POL | 7:0 | POL | | | | | G4POL | G3POL | G2POL | G1POL |
| 0x1E30 | CLC4SEL0 | 7:0 | | | | | | D1S[5:0] | | | |
| 0x1E31 | CLC4SEL1 | 7:0 | | | | | | D2S[5:0] | | | |
| 0x1E32 | CLC4SEL2 | 7:0 | | | | | | D3S[5:0] | | | |
| 0x1E33 | CLC4SEL3 | 7:0 | | | | | | D4S[5:0] | | | |
| 0x1E34 | CLC4GLS0 | 7:0 | G1D4T | G1D4N | G1D3T | G1D3N | G1D2T | G1D2N | G1D1T | G1D1N | |
| 0x1E35 | CLC4GLS1 | 7:0 | G2D4T | G2D4N | G2D3T | G2D3N | G2D2T | G2D2N | G2D1T | G2D1N | |
| 0x1E36 | CLC4GLS2 | 7:0 | G3D4T | G3D4N | G3D3T | G3D3N | G3D2T | G3D2N | G3D1T | G3D1N | |
| 0x1E37 | CLC4GLS3 | 7:0 | G4D4T | G4D4N | G4D3T | G4D3N | G4D2T | G4D2N | G4D1T | G4D1N | |

17.8 Register Definitions: Configurable Logic Cell

17.8.1 CLCDATA

Name: CLCDATA
Address: 0x1E0F

CLC Data Output Register

Mirror copy of

| | | | | | | | | |
|--------|---|---|---|---|---------|---------|---------|---------|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | | | | MLC4OUT | MLC3OUT | MLC2OUT | MLC1OUT |
| Access | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | 0 | 0 | 0 | 0 |

Bits 0, 1, 2, 3 – MLCxOUT
 Mirror copy of CLCx_out bit

| Value | Description |
|-------|---------------|
| 1 | CLCx_out is 1 |
| 0 | CLCx_out is 0 |

17.8.2 CLCxCON

Name: CLCxCON
Address: 0x1E10,0x1E1A,0x1E24,0x1E2E

Configurable Logic Cell Control Register

| | | | | | | | | |
|--------|-----|---|-----|------|------|-----------|-----|-----|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | EN | | OUT | INTP | INTN | MODE[2:0] | | |
| Access | R/W | | RO | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | | 0 | 0 | 0 | 0 | 0 | 0 |

Bit 7 – EN

CLC Enable bit

| Value | Description |
|-------|---|
| 1 | Configurable logic cell is enabled and mixing signals |
| 0 | Configurable logic cell is disabled and has logic zero output |

Bit 5 – OUT

Logic cell output data, after LCPOL. Sampled from CLCxOUT

Bit 4 – INTP

Configurable Logic Cell Positive Edge Going Interrupt Enable bit

| Value | Description |
|-------|---|
| 1 | CLCxIF will be set when a rising edge occurs on CLCxOUT |
| 0 | Rising edges on CLCxOUT have no effect on CLCxIF |

Bit 3 – INTN

Configurable Logic Cell Negative Edge Going Interrupt Enable bit

| Value | Description |
|-------|--|
| 1 | CLCxIF will be set when a falling edge occurs on CLCxOUT |
| 0 | Falling edges on CLCxOUT have no effect on CLCxIF |

Bits 2:0 – MODE[2:0]

Configurable Logic Cell Functional Mode Selection bits

| Value | Description |
|-------|--|
| 111 | Cell is 1-input transparent latch with Set and Reset |
| 110 | Cell is J-K flip-flop with Reset |
| 101 | Cell is 2-input D flip-flop with Reset |
| 100 | Cell is 1-input D flip-flop with Set and Reset |
| 011 | Cell is S-R latch |
| 010 | Cell is 4-input AND |
| 001 | Cell is OR-XOR |
| 000 | Cell is AND-OR |

17.8.3 CLCxPOL

Name: CLCxPOL
Address: 0x1E11,0x1E1B,0x1E25,0x1E2F

Signal Polarity Control Register

| | | | | | | | | |
|--------|-----|---|---|---|-------|-------|-------|-------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | POL | | | | G4POL | G3POL | G2POL | G1POL |
| Access | R/W | | | | R/W | R/W | R/W | R/W |
| Reset | 0 | | | | x | x | x | x |

Bit 7 – POL

CLCxOUT Output Polarity Control bit

| Value | Description |
|-------|--|
| 1 | The output of the logic cell is inverted |
| 0 | The output of the logic cell is not inverted |

Bits 0, 1, 2, 3 – GypOL

Gate Output Polarity Control bit

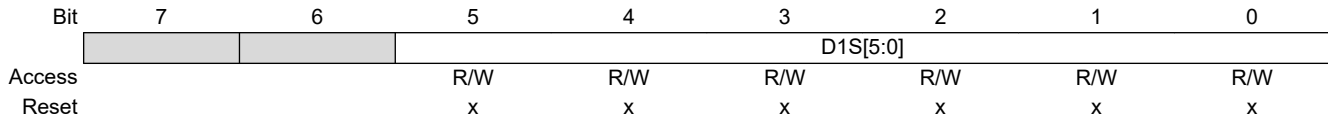
Reset States: Default = xxxx
 POR/BOR = x
 All Other Resets = u

| Value | Description |
|-------|--|
| 1 | The gate output is inverted when applied to the logic cell |
| 0 | The output of the gate is not inverted |

17.8.4 CLCxSELO

Name: CLCxSELO
Address: 0x1E12,0x1E1C,0x1E26,0x1E30

Generic CLCx Data 1 Select Register



Bits 5:0 – D1S[5:0] CLCx Data 1 Input Selection bits

Table 17-2. CLC Data Input Sources

| D1S Value | CLC Input Source | D1S Value | CLC Input Source |
|-------------|------------------|-------------|--------------------|
| 111111 [63] | Reserved | 011111 [31] | DSM1_out |
| 111110 [62] | Reserved | 011110 [30] | IOC_flag |
| 111101 [61] | Reserved | 011101 [29] | ZCD_out |
| 111100 [60] | Reserved | 011100 [28] | C2_out |
| 111011 [59] | Reserved | 011011 [27] | C1_out |
| 111010 [58] | Reserved | 011010 [26] | NCO1_out |
| 111001 [57] | Reserved | 011001 [25] | PWM7_out |
| 111000 [56] | Reserved | 011000 [24] | PWM6_out |
| 110111 [55] | Reserved | 010111 [23] | CCP4_out |
| 110110 [54] | Reserved | 010110 [22] | CCP3_out |
| 110101 [53] | Reserved | 010101 [21] | CCP2_out |
| 110100 [52] | Reserved | 010100 [20] | CCP1_out |
| 110011 [51] | Reserved | 010011 [19] | SMT1_overflow |
| 110010 [50] | Reserved | 010010 [18] | TMR6_out |
| 110001 [49] | Reserved | 010001 [17] | TMR5_overflow |
| 110000 [48] | Reserved | 010000 [16] | TMR4_out |
| 101111 [47] | Reserved | 001111 [15] | TMR3_overflow |
| 101110 [46] | Reserved | 001110 [14] | TMR2_out |
| 101101 [45] | CWG2B_out test | 001101 [13] | TMR1_overflow |
| 101100 [44] | CWG2A_out | 001100 [12] | TMR0_overflow |
| 101011 [43] | CWG1B_out | 001011 [11] | CLKR_out |
| 101010 [42] | CWG1A_out | 001010 [10] | FRC |
| 101001 [41] | MSSP2_clk_out | 001001 [9] | SOSC |
| 101000 [40] | MSSP2_data_out | 001000 [8] | MFINTOSC (32 kHz) |
| 100111 [39] | MSSP1_clk_out | 000111 [7] | MFINTOSC (500 kHz) |

PIC16(L)F18426/46

CLC - Configurable Logic Cell

.....continued

| D1S Value | CLC Input Source | D1S Value | CLC Input Source |
|-------------|------------------|------------|-------------------|
| 100110 [38] | MSSP1_data_out | 000110 [6] | LFINTOSC |
| 100101 [37] | EUSART1_CK_out | 000101 [5] | HFINTOSC (32 MHz) |
| 100100 [36] | EUSART1_DT_out | 000100 [4] | F _{osc} |
| 100011 [35] | CLC4_out | 000011 [3] | CLCIN3PPS |
| 100010 [34] | CLC3_out | 000010 [2] | CLCIN2PPS |
| 100001 [33] | CLC2_out | 000001 [1] | CLCIN1PPS |
| 100000 [32] | CLC1_out | 000000 [0] | CLCIN0PPS |

Reset States: POR/BOR = xxxxxx

All Other Resets = uuuuuu

17.8.5 CLCxSEL1

Name: CLCxSEL1
Address: 0x1E13,0x1E1D,0x1E27,0x1E31

Generic CLCx Data 2 Select Register

| | | | | | | | | |
|--------|----------|---|-----|-----|-----|-----|-----|-----|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | D2S[5:0] | | | | | | | |
| Access | | | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | | x | x | x | x | x | x |

Bits 5:0 – D2S[5:0] CLCx Data 2 Input Selection bits

Table 17-3. CLC Data Input Sources

| D2S Value | CLC Input Source | D2S Value | CLC Input Source |
|-------------|------------------|-------------|--------------------|
| 111111 [63] | Reserved | 011111 [31] | DSM1_out |
| 111110 [62] | Reserved | 011110 [30] | IOC_flag |
| 111101 [61] | Reserved | 011101 [29] | ZCD_out |
| 111100 [60] | Reserved | 011100 [28] | C2_out |
| 111011 [59] | Reserved | 011011 [27] | C1_out |
| 111010 [58] | Reserved | 011010 [26] | NCO1_out |
| 111001 [57] | Reserved | 011001 [25] | PWM7_out |
| 111000 [56] | Reserved | 011000 [24] | PWM6_out |
| 110111 [55] | Reserved | 010111 [23] | CCP4_out |
| 110110 [54] | Reserved | 010110 [22] | CCP3_out |
| 110101 [53] | Reserved | 010101 [21] | CCP2_out |
| 110100 [52] | Reserved | 010100 [20] | CCP1_out |
| 110011 [51] | Reserved | 010011 [19] | SMT1_overflow |
| 110010 [50] | Reserved | 010010 [18] | TMR6_out |
| 110001 [49] | Reserved | 010001 [17] | TMR5_overflow |
| 110000 [48] | Reserved | 010000 [16] | TMR4_out |
| 101111 [47] | Reserved | 001111 [15] | TMR3_overflow |
| 101110 [46] | Reserved | 001110 [14] | TMR2_out |
| 101101 [45] | CWG2B_out test | 001101 [13] | TMR1_overflow |
| 101100 [44] | CWG2A_out | 001100 [12] | TMR0_overflow |
| 101011 [43] | CWG1B_out | 001011 [11] | CLKR_out |
| 101010 [42] | CWG1A_out | 001010 [10] | FRC |
| 101001 [41] | MSSP2_clk_out | 001001 [9] | SOSC |
| 101000 [40] | MSSP2_data_out | 001000 [8] | MFINTOSC (32 kHz) |
| 100111 [39] | MSSP1_clk_out | 000111 [7] | MFINTOSC (500 kHz) |

PIC16(L)F18426/46

CLC - Configurable Logic Cell

.....continued

| D2S Value | CLC Input Source | D2S Value | CLC Input Source |
|-------------|------------------|------------|-------------------|
| 100110 [38] | MSSP1_data_out | 000110 [6] | LFINTOSC |
| 100101 [37] | EUSART1_CK_out | 000101 [5] | HFINTOSC (32 MHz) |
| 100100 [36] | EUSART1_DT_out | 000100 [4] | F _{osc} |
| 100011 [35] | CLC4_out | 000011 [3] | CLCIN3PPS |
| 100010 [34] | CLC3_out | 000010 [2] | CLCIN2PPS |
| 100001 [33] | CLC2_out | 000001 [1] | CLCIN1PPS |
| 100000 [32] | CLC1_out | 000000 [0] | CLCIN0PPS |

Reset States: POR/BOR = xxxxxx

All Other Resets = uuuuuu

17.8.6 CLCxSEL2

Name: CLCxSEL2
Address: 0x1E14,0x1E1E,0x1E28,0x1E32

Generic CLCx Data 3 Select Register

| | | | | | | | | |
|--------|----------|---|-----|-----|-----|-----|-----|-----|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | D3S[5:0] | | | | | | | |
| Access | | | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | | x | x | x | x | x | x |

Bits 5:0 – D3S[5:0] CLCx Data 3 Input Selection bits

Table 17-4. CLC Data Input Sources

| D3S Value | CLC Input Source | D3S Value | CLC Input Source |
|-------------|------------------|-------------|--------------------|
| 111111 [63] | Reserved | 011111 [31] | DSM1_out |
| 111110 [62] | Reserved | 011110 [30] | IOC_flag |
| 111101 [61] | Reserved | 011101 [29] | ZCD_out |
| 111100 [60] | Reserved | 011100 [28] | C2_out |
| 111011 [59] | Reserved | 011011 [27] | C1_out |
| 111010 [58] | Reserved | 011010 [26] | NCO1_out |
| 111001 [57] | Reserved | 011001 [25] | PWM7_out |
| 111000 [56] | Reserved | 011000 [24] | PWM6_out |
| 110111 [55] | Reserved | 010111 [23] | CCP4_out |
| 110110 [54] | Reserved | 010110 [22] | CCP3_out |
| 110101 [53] | Reserved | 010101 [21] | CCP2_out |
| 110100 [52] | Reserved | 010100 [20] | CCP1_out |
| 110011 [51] | Reserved | 010011 [19] | SMT1_overflow |
| 110010 [50] | Reserved | 010010 [18] | TMR6_out |
| 110001 [49] | Reserved | 010001 [17] | TMR5_overflow |
| 110000 [48] | Reserved | 010000 [16] | TMR4_out |
| 101111 [47] | Reserved | 001111 [15] | TMR3_overflow |
| 101110 [46] | Reserved | 001110 [14] | TMR2_out |
| 101101 [45] | CWG2B_out test | 001101 [13] | TMR1_overflow |
| 101100 [44] | CWG2A_out | 001100 [12] | TMR0_overflow |
| 101011 [43] | CWG1B_out | 001011 [11] | CLKR_out |
| 101010 [42] | CWG1A_out | 001010 [10] | FRC |
| 101001 [41] | MSSP2_clk_out | 001001 [9] | SOSC |
| 101000 [40] | MSSP2_data_out | 001000 [8] | MFINTOSC (32 kHz) |
| 100111 [39] | MSSP1_clk_out | 000111 [7] | MFINTOSC (500 kHz) |

PIC16(L)F18426/46

CLC - Configurable Logic Cell

.....continued

| D3S Value | CLC Input Source | D3S Value | CLC Input Source |
|-------------|------------------|------------|-------------------|
| 100110 [38] | MSSP1_data_out | 000110 [6] | LFINTOSC |
| 100101 [37] | EUSART1_CK_out | 000101 [5] | HFINTOSC (32 MHz) |
| 100100 [36] | EUSART1_DT_out | 000100 [4] | F _{osc} |
| 100011 [35] | CLC4_out | 000011 [3] | CLCIN3PPS |
| 100010 [34] | CLC3_out | 000010 [2] | CLCIN2PPS |
| 100001 [33] | CLC2_out | 000001 [1] | CLCIN1PPS |
| 100000 [32] | CLC1_out | 000000 [0] | CLCIN0PPS |

Reset States: POR/BOR = xxxxxx
 All Other Resets = uuuuuu

17.8.7 CLCxSEL3

Name: CLCxSEL3
Address: 0x1E15,0x1E1F,0x1E29,0x1E33

Generic CLCx Data 4 Select Register

| | | | | | | | | |
|--------|----------|---|-----|-----|-----|-----|-----|-----|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | D4S[5:0] | | | | | | | |
| Access | | | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | | x | x | x | x | x | x |

Bits 5:0 – D4S[5:0] CLCx Data 4 Input Selection bits

Table 17-5. CLC Data Input Sources

| D4S Value | CLC Input Source | D4S Value | CLC Input Source |
|-------------|------------------|-------------|--------------------|
| 111111 [63] | Reserved | 011111 [31] | DSM1_out |
| 111110 [62] | Reserved | 011110 [30] | IOC_flag |
| 111101 [61] | Reserved | 011101 [29] | ZCD_out |
| 111100 [60] | Reserved | 011100 [28] | C2_out |
| 111011 [59] | Reserved | 011011 [27] | C1_out |
| 111010 [58] | Reserved | 011010 [26] | NCO1_out |
| 111001 [57] | Reserved | 011001 [25] | PWM7_out |
| 111000 [56] | Reserved | 011000 [24] | PWM6_out |
| 110111 [55] | Reserved | 010111 [23] | CCP4_out |
| 110110 [54] | Reserved | 010110 [22] | CCP3_out |
| 110101 [53] | Reserved | 010101 [21] | CCP2_out |
| 110100 [52] | Reserved | 010100 [20] | CCP1_out |
| 110011 [51] | Reserved | 010011 [19] | SMT1_overflow |
| 110010 [50] | Reserved | 010010 [18] | TMR6_out |
| 110001 [49] | Reserved | 010001 [17] | TMR5_overflow |
| 110000 [48] | Reserved | 010000 [16] | TMR4_out |
| 101111 [47] | Reserved | 001111 [15] | TMR3_overflow |
| 101110 [46] | Reserved | 001110 [14] | TMR2_out |
| 101101 [45] | CWG2B_out test | 001101 [13] | TMR1_overflow |
| 101100 [44] | CWG2A_out | 001100 [12] | TMR0_overflow |
| 101011 [43] | CWG1B_out | 001011 [11] | CLKR_out |
| 101010 [42] | CWG1A_out | 001010 [10] | FRC |
| 101001 [41] | MSSP2_clk_out | 001001 [9] | SOSC |
| 101000 [40] | MSSP2_data_out | 001000 [8] | MFINTOSC (32 kHz) |
| 100111 [39] | MSSP1_clk_out | 000111 [7] | MFINTOSC (500 kHz) |

PIC16(L)F18426/46

CLC - Configurable Logic Cell

.....continued

| D4S Value | CLC Input Source | D4S Value | CLC Input Source |
|-------------|------------------|------------|-------------------|
| 100110 [38] | MSSP1_data_out | 000110 [6] | LFINTOSC |
| 100101 [37] | EUSART1_CK_out | 000101 [5] | HFINTOSC (32 MHz) |
| 100100 [36] | EUSART1_DT_out | 000100 [4] | F _{osc} |
| 100011 [35] | CLC4_out | 000011 [3] | CLCIN3PPS |
| 100010 [34] | CLC3_out | 000010 [2] | CLCIN2PPS |
| 100001 [33] | CLC2_out | 000001 [1] | CLCIN1PPS |
| 100000 [32] | CLC1_out | 000000 [0] | CLCIN0PPS |

Reset States: POR/BOR = xxxxxx
 All Other Resets = uuuuuu

17.8.8 CLCxGLS0

Name: CLCxGLS0
Address: 0x1E16,0x1E20,0x1E2A,0x1E34

CLCx Gate1 Logic Select Register

| | | | | | | | | |
|--------|-------|-------|-------|-------|-------|-------|-------|-------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | G1D4T | G1D4N | G1D3T | G1D3N | G1D2T | G1D2N | G1D1T | G1D1N |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | x | x | x | x | x | x | x | x |

Bits 1, 3, 5, 7 – G1DyT

dyT: Gate1 Data 'y' True (non-inverted) bit

Reset States: Default = xxxx

POR/BOR = x

All Other Resets = u

| Value | Description |
|-------|--------------------------|
| 1 | dyT is gated into g1 |
| 0 | dyT is not gated into g1 |

Bits 0, 2, 4, 6 – G1DyN

dyN: Gate1 Data 'y' Negated (inverted) bit

Reset States: Default = xxxx

POR/BOR = x

All Other Resets = u

| Value | Description |
|-------|--------------------------|
| 1 | dyN is gated into g1 |
| 0 | dyN is not gated into g1 |

17.8.9 CLCxGLS1

Name: CLCxGLS1
Address: 0x1E17,0x1E21,0x1E2B,0x1E35

CLCx Gate2 Logic Select Register

| | | | | | | | | |
|--------|-------|-------|-------|-------|-------|-------|-------|-------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | G2D4T | G2D4N | G2D3T | G2D3N | G2D2T | G2D2N | G2D1T | G2D1N |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | x | x | x | x | x | x | x | x |

Bits 1, 3, 5, 7 – G2DyT

dyT: Gate2 Data 'y' True (non-inverted) bit

Reset States: Default = xxxx

POR/BOR = x

All Other Resets = u

| Value | Description |
|-------|--------------------------|
| 1 | dyT is gated into g2 |
| 0 | dyT is not gated into g2 |

Bits 0, 2, 4, 6 – G2DyN

dyN: Gate2 Data 'y' Negated (inverted) bit

Reset States: Default = xxxx

POR/BOR = x

All Other Resets = u

| Value | Description |
|-------|--------------------------|
| 1 | dyN is gated into g2 |
| 0 | dyN is not gated into g2 |

17.8.10 CLCxGLS2

Name: CLCxGLS2
Address: 0x1E18,0x1E22,0x1E2C,0x1E36

CLCx Gate3 Logic Select Register

| | | | | | | | | |
|--------|-------|-------|-------|-------|-------|-------|-------|-------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | G3D4T | G3D4N | G3D3T | G3D3N | G3D2T | G3D2N | G3D1T | G3D1N |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | x | x | x | x | x | x | x | x |

Bits 1, 3, 5, 7 – G3DyT

dyT: Gate3 Data 'y' True (non-inverted) bit

Reset States: Default = xxxx

POR/BOR = x

All Other Resets = u

| Value | Description |
|-------|--------------------------|
| 1 | dyT is gated into g3 |
| 0 | dyT is not gated into g3 |

Bits 0, 2, 4, 6 – G3DyN

dyN: Gate3 Data 'y' Negated (inverted) bit

Reset States: Default = xxxx

POR/BOR = x

All Other Resets = u

| Value | Description |
|-------|--------------------------|
| 1 | dyN is gated into g3 |
| 0 | dyN is not gated into g3 |

17.8.11 CLCxGLS3

Name: CLCxGLS3
Address: 0x1E19,0x1E23,0x1E2D,0x1E37

CLCx Gate4 Logic Select Register

| | | | | | | | | |
|--------|-------|-------|-------|-------|-------|-------|-------|-------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | G4D4T | G4D4N | G4D3T | G4D3N | G4D2T | G4D2N | G4D1T | G4D1N |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | x | x | x | x | x | x | x | x |

Bits 1, 3, 5, 7 – G4DyT

dyT: Gate4 Data 'y' True (non-inverted) bit

Reset States: Default = xxxx

POR/BOR = x

All Other Resets = u

| Value | Description |
|-------|--------------------------|
| 1 | dyT is gated into g4 |
| 0 | dyT is not gated into g4 |

Bits 0, 2, 4, 6 – G4DyN

dyN: Gate4 Data 'y' Negated (inverted) bit

Reset States: Default = xxxx

POR/BOR = x

All Other Resets = u

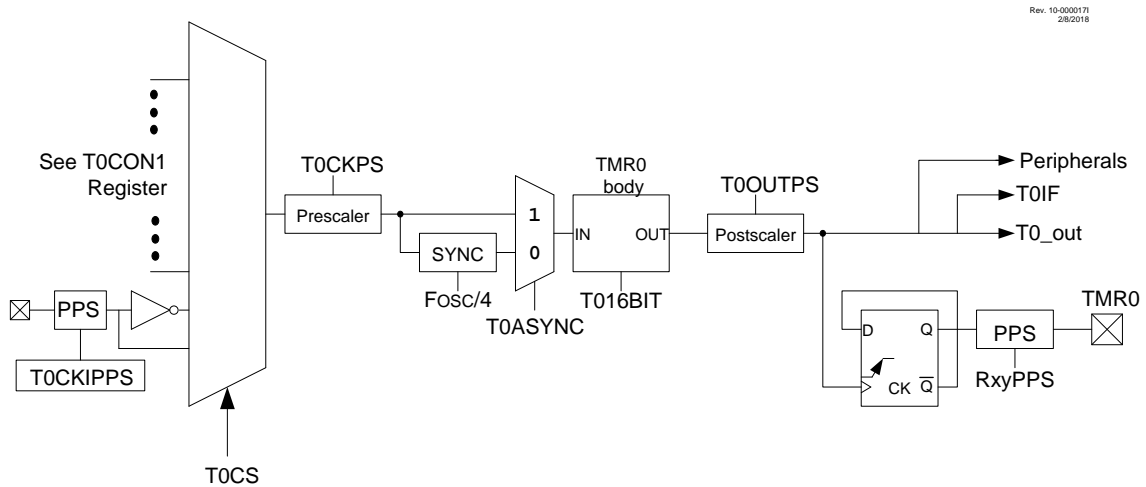
| Value | Description |
|-------|--------------------------|
| 1 | dyN is gated into g4 |
| 0 | dyN is not gated into g4 |

18. TMR0 - Timer0 Module

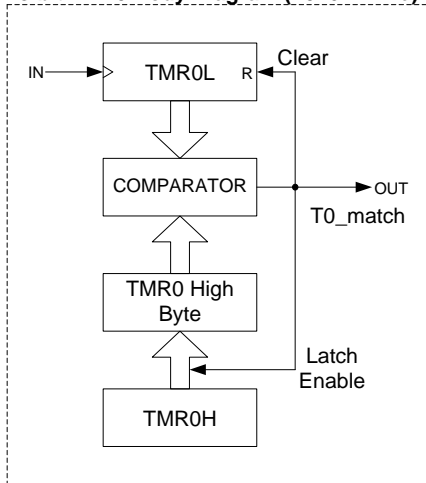
Timer0 module has the following features:

- 8-Bit Timer with Programmable Period
- 16-Bit Timer
- Selectable Clock Sources
- Synchronous and Asynchronous Operation
- Programmable Prescaler and Postscaler
- Interrupt on Match or Overflow
- Output on I/O Pin (via PPS) or to Other Peripherals
- Operation During Sleep

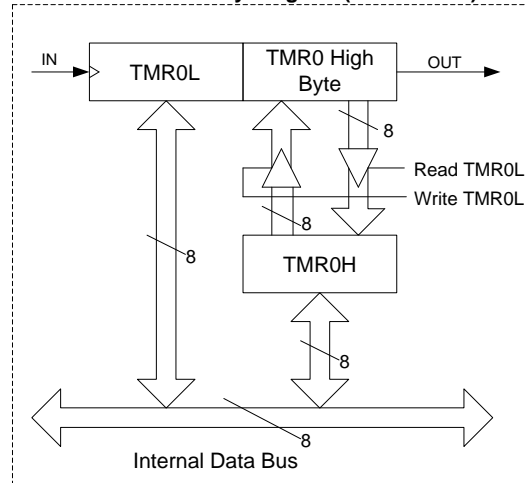
Figure 18-1. Timer0 Block Diagram



8-bit TMR0 Body Diagram (T016BIT = 0)



16-bit TMR0 Body Diagram (T016BIT = 1)



18.1 Timer0 Operation

Timer0 can operate as either an 8-bit or 16-bit timer. The mode is selected with the [T016BIT](#) bit.

18.1.1 8-bit Mode

In this mode Timer0 increments on the rising edge of the selected clock source. A prescaler on the clock input gives several prescale options (see prescaler control bits, [T0CKPS](#)).

In this mode as shown in [Figure 18-1](#), a buffered version of TMR0H is maintained. This is compared with the value of TMR0L on each cycle of the selected clock source. When the two values match, the following events occur:

- TMR0L is reset
- The contents of TMR0H are copied to the TMR0H buffer for next comparison

18.1.2 16-Bit Mode

In this mode Timer0 increments on the rising edge of the selected clock source. A prescaler on the clock input gives several prescale options (see prescaler control bits, [T0CKPS](#)).

In this mode TMR0H:TMR0L form the 16-bit timer value. As shown in [Figure 18-1](#), read and write of the TMR0H register are buffered. TMR0H register is updated with the contents of the high byte of Timer0 during a read of TMR0L register. Similarly, a write to the high byte of Timer0 takes place through the TMR0H buffer register. The high byte is updated with the contents of TMR0H register when a write occurs to TMR0L register. This allows all 16 bits of Timer0 to be read and written at the same time.

Timer0 rolls over to 0x0000 on incrementing past 0xFFFF. This makes the timer free-running. TMR0L/H registers cannot be reloaded in this mode once started.

18.2 Clock Selection

Timer0 has several options for clock source selections, option to operate synchronously/asynchronously and a programmable prescaler.

18.2.1 Clock Source Selection

The [T0CS](#) bits in T0CON1 register are used to select the clock source for Timer0. Refer to [Timer0 Clock Source Selections](#) table for possible clock sources.

18.2.2 Synchronous Mode

When the [T0ASYNC](#) bit is clear, Timer0 clock is synchronized to the system clock ($F_{OSC}/4$). When operating in Synchronous mode, Timer0 clock frequency cannot exceed $F_{OSC}/4$. During Sleep mode system clock is not available and Timer0 cannot operate.

18.2.3 Asynchronous Mode

When the [T0ASYNC](#) bit is set, Timer0 increments with each rising edge of the input source (or output of the prescaler, if used). Asynchronous mode allows Timer0 to continue operation during Sleep mode provided the selected clock source is available.

18.2.4 Programmable Prescaler

Timer0 has 16 programmable input prescaler options ranging from 1:1 to 1:32768. The prescaler values are selected using the [T0CKPS](#) bits.

The prescaler counter is not directly readable or writable. The prescaler counter is cleared on the following events:

- A write to the TMR0L register
- A write to either the T0CON0 or T0CON1 registers
- Any device Reset

Related Links

[10. Resets](#)

18.3 Timer0 Output and Interrupt

18.3.1 Programmable Postscaler

Timer0 has 16 programmable output postscaler options ranging from 1:1 to 1:16. The postscaler values are selected using the **T0OUTPS** bits. The postscaler divides the output of Timer0 by the selected ratio.

The postscaler counter is not directly readable or writable. The postscaler counter is cleared on the following events:

- A write to the TMR0L register
- A write to either the T0CON0 or T0CON1 registers
- Any device Reset

18.3.2 Timer0 Output

TMR0_out is the output of the postscaler. TMR0_out toggles on every match between TMR0L and TMR0H in 8-bit mode, or when TMR0H:TMR0L rolls over in 16-bit mode. If the output postscaler is used, the output is scaled by the ratio selected.

The Timer0 output can be routed to an I/O pin via the RxyPPS output selection register. The Timer0 output can be monitored through software via the **T0OUT** output bit.

Related Links

[16.2 PPS Outputs](#)

18.3.3 Timer0 Interrupt

The Timer0 Interrupt Flag bit (TMR0IF) is set when the TMR0_out toggles. If the Timer0 interrupt is enabled (TMR0IE), the CPU will be interrupted when the TMR0IF bit is set.

When the postscaler bits (T0OUTPS) are set to 1:1 operation (no division), the T0IF flag bit will be set with every TMR0 match or rollover. In general, the TMR0IF flag bit will be set every T0OUTPS + 1 matches or rollovers.

18.3.4 Timer0 Example

Timer0 Configuration:

- Timer0 mode = 16-bit
- Clock Source = $F_{OSC}/4$ (250 kHz)
- Synchronous operation
- Prescaler = 1:1
- Postscaler = 1:2 (T0OUTPS = 1)

In this case the TMR0_out toggles every two rollovers of TMR0H:TMR0L. i.e., $(0xFFFF)*2*(1/250kHz) = 524.28$ ms

18.4 Operation During Sleep

When operating synchronously, Timer0 will halt when the device enters Sleep mode.

When operating asynchronously and selected clock source is active, Timer0 will continue to increment and wake the device from Sleep mode if Timer0 interrupt is enabled.

18.5 Register Summary - Timer0

| Address | Name | Bit Pos. | | | | | | | |
|-------------|----------|----------|------------|--|-------|---------|--------------|--|--|
| 0x00 ... | Reserved | | | | | | | | |
| 0x059B | | | | | | | | | |
| 0x059C | TMR0L | 7:0 | TMR0L[7:0] | | | | | | |
| 0x059D | TMR0H | 7:0 | TMR0H[7:0] | | | | | | |
| 0x059E | TOCON0 | 7:0 | T0EN | | T0OUT | T016BIT | T0OUTPS[3:0] | | |
| 0x059F | TOCON1 | 7:0 | T0CS[2:0] | | | T0ASYNC | T0CKPS[3:0] | | |

18.6 Register Definitions: Timer0 Control

18.6.1 TMR0L

Name: TMR0L

Address: 0x59C

Timer0 Period/Count Low Register

| | | | | | | | | |
|--------|------------|-----|-----|-----|-----|-----|-----|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TMR0L[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bits 7:0 – TMR0L[7:0] TMR0 Least Significant Counter

| Value | Condition | Description |
|----------------------------|-------------|--------------------------------------|
| 11111111 to 00000000 | T016BIT = 0 | 8-bit Timer0 Counter bits |
| 11111111 to 00000000 | T016BIT = 1 | 16-bit Timer0 Least Significant Byte |

18.6.2 TMR0H

Name: TMR0H

Address: 0x59D

Timer0 Period/Count High Register

| | | | | | | | | |
|--------|------------|-----|-----|-----|-----|-----|-----|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TMR0H[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bits 7:0 – TMR0H[7:0] TMR0 Most Significant Counter

| Value | Condition | Description |
|----------------------------|-------------|--|
| 11111111 to 00000000 | T016BIT = 0 | 8-bit Timer0 Period Value. TMR0L continues counting from 0 when this value is reached. |
| 11111111 to 00000000 | T016BIT = 1 | 16-bit Timer0 Most Significant Byte |

18.6.3 T0CON0

Name: T0CON0

Address: 0x59E

Timer0 Control Register 0

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|------|---|-------|---------|-----|--------------|-----|-----|
| | T0EN | | T0OUT | T016BIT | | T0OUTPS[3:0] | | |
| Access | R/W | | R | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | | 0 | 0 | 0 | 0 | 0 | 0 |

Bit 7 – T0EN TMR0 Enable

| Value | Description |
|-------|-------------------------------------|
| 1 | The module is enabled and operating |
| 0 | The module is disabled |

Bit 5 – T0OUT TMR0 Output

Bit 4 – T016BIT TMR0 Operating as 16-Bit Timer Select

| Value | Description |
|-------|------------------------|
| 1 | TMR0 is a 16-bit timer |
| 0 | TMR0 is an 8-bit timer |

Bits 3:0 – T0OUTPS[3:0] TMR0 Output Postscaler (Divider) Select

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

18.6.4 T0CON1

Name: T0CON1

Address: 0x59F

Timer0 Control Register 1

| | | | | | | | | |
|--------|-----------|-----|-----|---------|-------------|-----|-----|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | T0CS[2:0] | | | T0ASYNC | T0CKPS[3:0] | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bits 7:5 – T0CS[2:0] Timer0 Clock Source Select

Table 18-1. Timer0 Clock Source Selections

| T0CS | Clock Source |
|------|--|
| 111 | CLC1_out |
| 110 | SOSC |
| 101 | MFINTOSC(500 kHz) |
| 100 | LFINTOSC |
| 011 | HFINTOSC |
| 010 | F _{OSC} /4 |
| 001 | Pin selected by T0CKIPPS (Inverted) |
| 000 | Pin selected by T0CKIPPS (Noninverted) |

Bit 4 – T0ASYNC TMR0 Input Asynchronization Enable

| Value | Description |
|-------|--|
| 1 | The input to the TMR0 counter is not synchronized to system clocks |
| 0 | The input to the TMR0 counter is synchronized to F _{OSC} /4 |

Bits 3:0 – T0CKPS[3:0] Prescaler Rate Select

| Value | Description |
|-------|-------------|
| 1111 | 1:32768 |
| 1110 | 1:16384 |
| 1101 | 1:8192 |
| 1100 | 1:4096 |
| 1011 | 1:2048 |
| 1010 | 1:1024 |
| 1001 | 1:512 |
| 1000 | 1:256 |
| 0111 | 1:128 |
| 0110 | 1:64 |
| 0101 | 1:32 |
| 0100 | 1:16 |
| 0011 | 1:8 |
| 0010 | 1:4 |
| 0001 | 1:2 |
| 0000 | 1:1 |

19. TMR1 - Timer1 Module with Gate Control

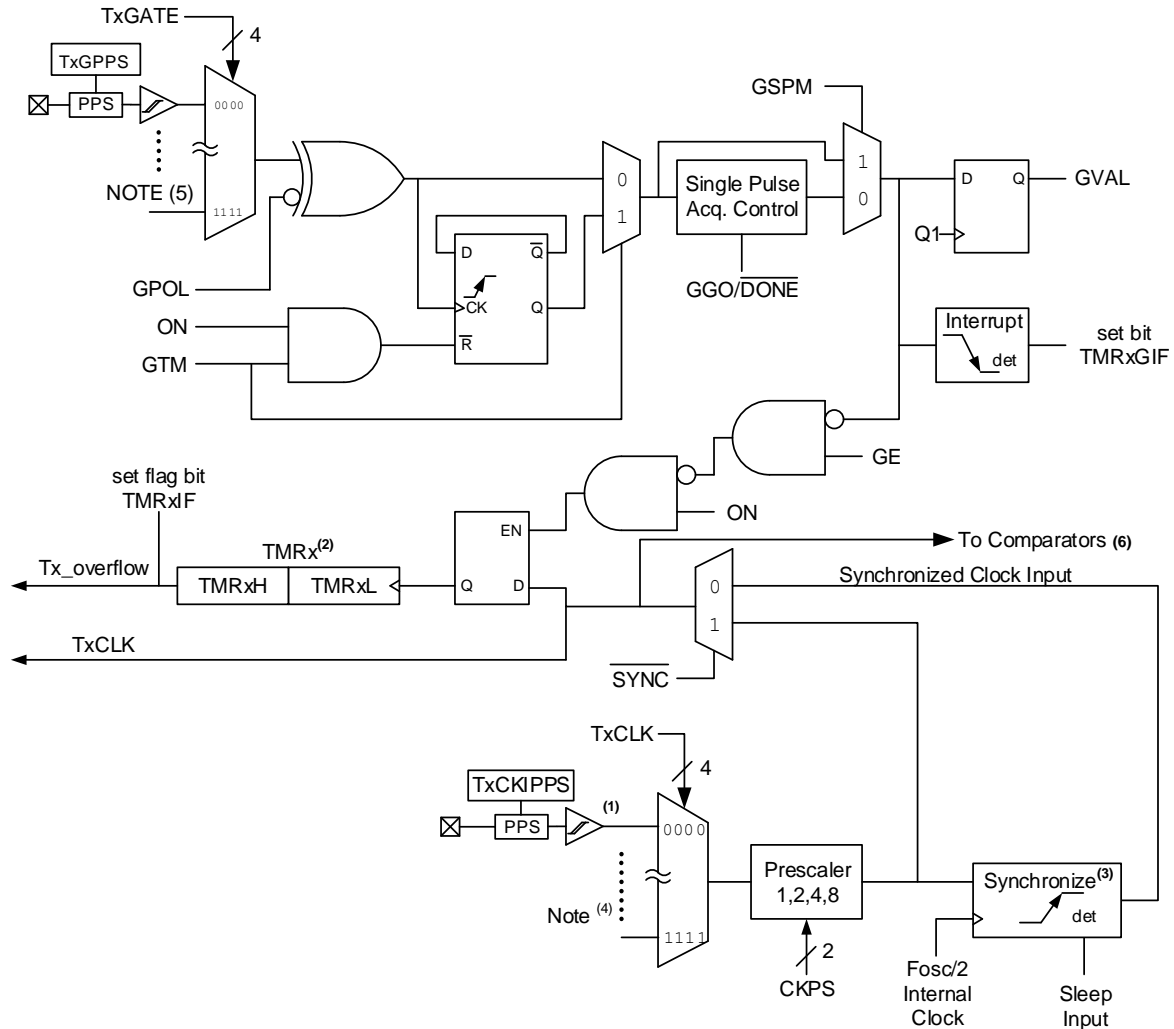
Timer1 module is a 16-bit timer/counter with the following features:

- 16-Bit Timer/Counter Register Pair (TMRxH:TMRxL)
- Programmable Internal or External Clock Source
- 2-Bit Prescaler
- Optionally Synchronized Comparator Out
- Multiple Timer1 Gate (Count Enable) Sources
- Interrupt-on-Overflow
- Wake-Up on Overflow (External Clock, Asynchronous Mode Only)
- 16-Bit Read/Write Operation
- Time Base for the Capture/Compare Function with the CCP Modules
- Special Event Trigger (with CCP)
- Selectable Gate Source Polarity
- Gate Toggle Mode
- Gate Single Pulse Mode
- Gate Value Status
- Gate Event Interrupt



Important: References to module Timer1 apply to all the odd numbered timers on this device.

Figure 19-1. Timer1 Block Diagram



Note:

1. This signal comes from the pin selected by TxCKIPPS.
2. TMRx register increments on rising edge.
3. Synchronize does not operate while in Sleep.
4. See [TMRxCLK](#) for clock source selections.
5. See [TMRxGATE](#) for gate source selection.
6. Synchronized comparator output should not be used in conjunction with synchronized input clock.

19.1 Timer1 Operation

The Timer1 module is a 16-bit incrementing counter that is accessed through the TMRxH:TMRxL register pair. Writes to TMRxH or TMRxL directly update the counter.

When used with an internal clock source, the module is a timer and increments on every instruction cycle. When used with an external clock source, the module can be used as either a timer or counter and increments on every selected edge of the external source.

Timer1 is enabled by configuring the **ON** and **GE** bits in the TxCON and TxGCON registers, respectively. The table below displays the Timer1 enable selections.

Table 19-1. Timer1 Enable Selections

| ON | GE | Timer1 Operation |
|----|----|------------------|
| 1 | 1 | Count Enabled |
| 1 | 0 | Always On |
| 0 | 1 | Off |
| 0 | 0 | Off |

19.2 Clock Source Selection

The **CS** bits select the clock source for Timer1. These bits allow the selection of several possible synchronous and asynchronous clock sources. The possible clock source are listed in [19.14.5 TMRxCLK](#) register.

19.2.1 Internal Clock Source

When the internal clock source is selected the TMRxH:TMRxL register pair will increment on multiples of F_{OSC} as determined by the Timer1 prescaler.

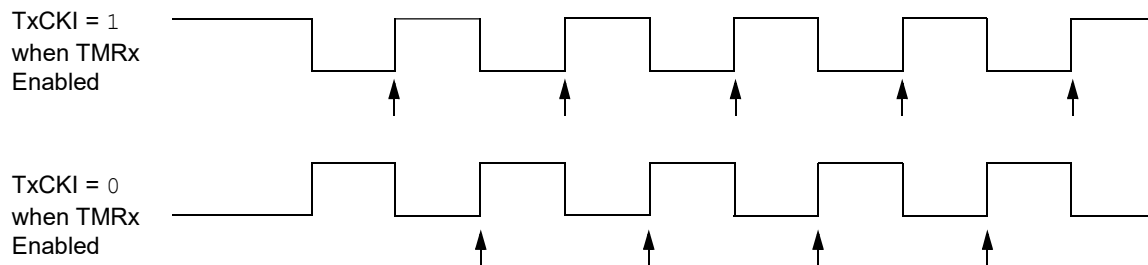
When the F_{OSC} internal clock source is selected, the Timer1 register value will increment by four counts every instruction clock cycle. Due to this condition, a 2 LSB error in resolution will occur when reading the Timer1 value. To utilize the full resolution of Timer1, an asynchronous input signal must be used to gate the Timer1 clock input.



Important: In Counter mode, a falling edge must be registered by the counter prior to the first incrementing rising edge after any one or more of the following conditions:

- Timer1 enabled after POR
- Write to TMRxH or TMRxL
- Timer1 is disabled
- Timer1 is disabled (TMRxON = 0) when TxCKI is high then Timer1 is enabled (TMRxON = 1) when TxCKI is low. Refer to the figure below.

Figure 19-2. 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.

19.2.2 External Clock Source

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

When enabled to count, Timer1 is incremented on the rising edge of the external clock input of the TxCKIPPS pin. This external clock source can be synchronized to the system clock or it can run asynchronously.

19.3 Timer1 Prescaler

Timer1 has four prescaler options allowing 1, 2, 4 or 8 divisions of the clock input. The **CKPS** bits control the prescale counter. The prescale counter is not directly readable or writable; however, the prescaler counter is cleared upon a write to TMRxH or TMRxL.

19.4 Secondary Oscillator

A secondary low-power 32.768 kHz oscillator circuit is built-in between pins SOSC1 (input) and SOSCO (amplifier output). This internal circuit is to be used in conjunction with an external 32.768 kHz crystal. The secondary oscillator is not dedicated only to Timer1; it can also be used by other modules.

The oscillator circuit is enabled by setting the SOSCEN bit of the OSCEN register. This can be used as one of the Timer1 clock sources selected with the **CS** bits. The oscillator will continue to run during Sleep.



Important: The oscillator requires a start-up and stabilization time before use. Thus, the SOSCEN bit of the OSCEN register should be set and a suitable delay observed prior to enabling Timer1. A software check can be performed to confirm if the secondary oscillator is enabled and ready to use. This is done by polling the SOR bit of the OSCSTAT register.

Related Links

[8.2.1.5 Secondary Oscillator](#)

19.5 Timer1 Operation in Asynchronous Counter Mode

When the **SYNC** control bit is set, the external clock input is not synchronized. The timer increments asynchronously to the internal phase clocks. If external clock source is selected then the timer will continue to run during Sleep and can generate an interrupt on overflow, which will wake-up the processor. However, special precautions in software are needed to read/write the timer (see [19.5.1 Reading and Writing Timer1 in Asynchronous Counter Mode](#)).



Important: When switching from synchronous to asynchronous operation, it is possible to skip an increment. When switching from asynchronous to synchronous operation, it is possible to produce an additional increment.

19.5.1 Reading and Writing Timer1 in Asynchronous Counter Mode

Reading TMRxH or TMRxL 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 TMRxH:TMRxL register pair.

19.6 Timer1 16-Bit Read/Write Mode

Timer1 can be configured to read and write all 16 bits of data, to and from, the 8-bit TMRxL and TMRxH registers, simultaneously. The 16-bit read and write operations are enabled by setting the **RD16** bit.

To accomplish this function, the TMRxH register value is mapped to a buffer register called the TMRxH buffer register. While in 16-Bit mode, the TMRxH register is not directly readable or writable and all read and write operations take place through the use of this TMRxH buffer register.

When a read from the TMRxL register is requested, the value of the TMRxH register is simultaneously loaded into the TMRxH buffer register. When a read from the TMRxH register is requested, the value is provided from the TMRxH

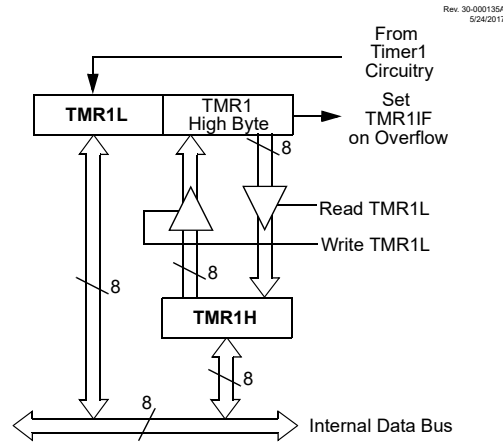
buffer register instead. This provides the user with the ability to accurately read all 16 bits of the Timer1 value from a single instance in time. Refer the figure below for more details.

In contrast, when not in 16-Bit mode, the user must read each register separately and determine if the values have become invalid due to a rollover that may have occurred between the read operations.

When a write request of the TMRxL register is requested, the TMRxH buffer register is simultaneously updated with the contents of the TMRxH register. The value of TMRxH must be preloaded into the TMRxH buffer register prior to the write request for the TMRxL register. This provides the user with the ability to write all 16 bits to the TMRxL:TMRxH register pair at the same time.

Any requests to write to the TMRxH directly does not clear the Timer1 prescaler value. The prescaler value is only cleared through write requests to the TMRxL register.

Figure 19-3. Timer1 16-Bit Read/Write Mode Block Diagram



19.7 Timer1 Gate

Timer1 can be configured to count freely or the count can be enabled and disabled using Timer1 gate circuitry. This is also referred to as Timer1 gate enable.

Timer1 gate can also be driven by multiple selectable sources.

19.7.1 Timer1 Gate Enable

The Timer1 Gate Enable mode is enabled by setting the **GE** bit. The polarity of the Timer1 Gate Enable mode is configured using the **GPOL** bit.

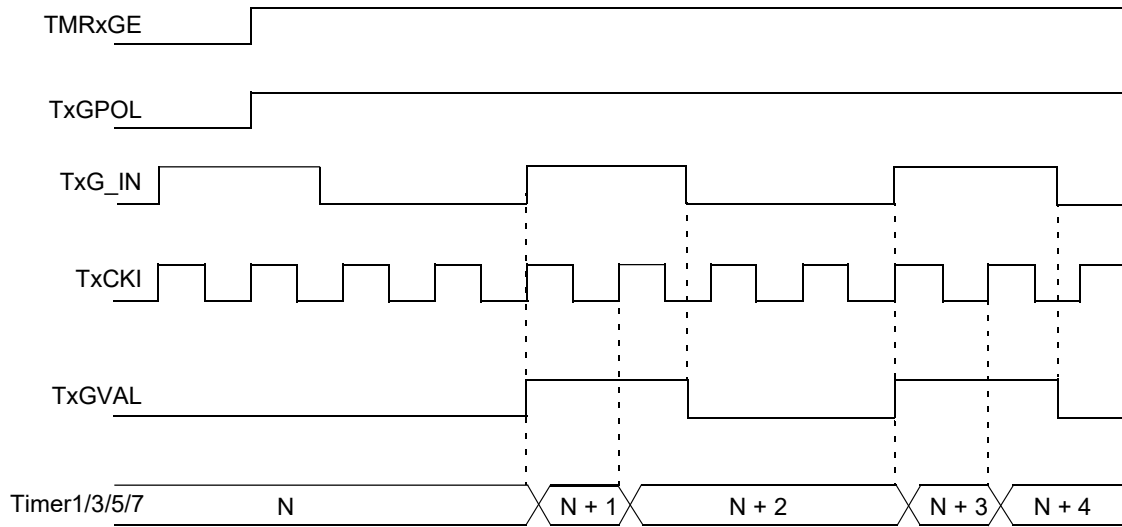
When Timer1 Gate Enable mode is enabled, Timer1 will increment on the rising edge of the Timer1 clock source. When Timer1 Gate signal is inactive, the timer will not increment and hold the current count. Enable mode is disabled, no incrementing will occur and Timer1 will hold the current count. See figure below for timing details.

Table 19-2. Timer1 Gate Enable Selections

| TMRxCLK | GPOL | TxG | Timer1 Operation |
|---------|------|-----|------------------|
| ↑ | 1 | 1 | Counts |
| ↑ | 1 | 0 | Holds Count |
| ↑ | 0 | 1 | Holds Count |
| ↑ | 0 | 0 | Counts |

Figure 19-4. Timer1 Gate Enable Mode

Rev. 30-000137A
5/24/2017



19.7.2 Timer1 Gate Source Selection

The gate source for Timer1 is selected using the [GSS](#) bits. The polarity selection for the gate source is controlled by the [GPOL](#) bit. The following gate source are available:

- Signal Measurement Timers
- Configurable Logic Cells
- Comparators
- PWM
- NCO
- T1CLKIN

For all options for gate source selection refer to the [19.14.4 TMRxGATE](#).

Related Links

[34.4.1 Comparator Output Synchronization](#)

19.7.3 Timer1 Gate Toggle Mode

When Timer1 Gate Toggle mode is enabled, it is possible to measure the full-cycle length of a Timer1 gate signal, as opposed to the duration of a single level pulse.

The Timer1 gate source is routed through a flip-flop that changes state on every incrementing edge of the signal. See figure below for timing details.

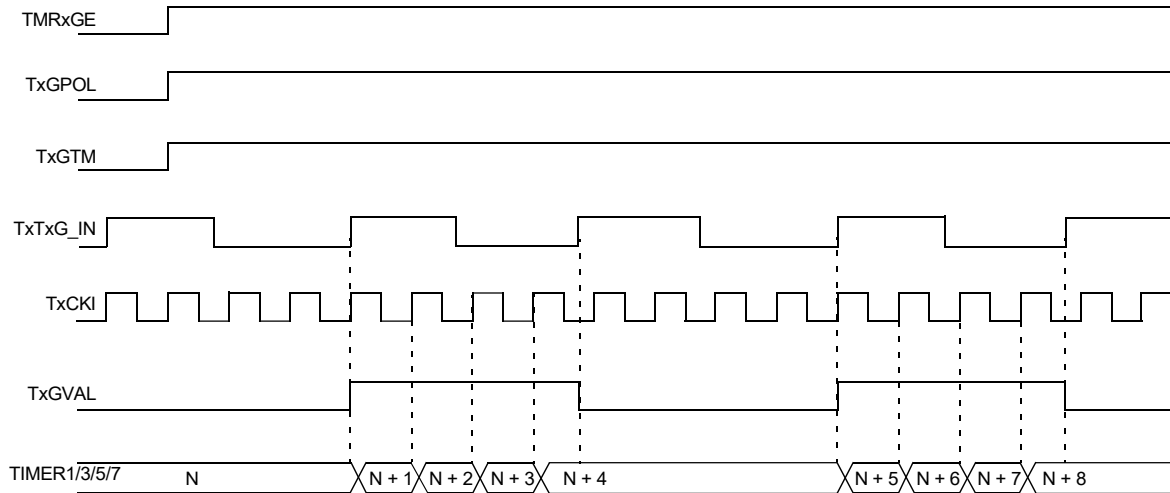
Timer1 Gate Toggle mode is enabled by setting the [GTM](#) bit. When the GTM bit is cleared, the flip-flop is cleared and held clear. This is necessary in order to control which edge is measured.



Important: Enabling Toggle mode at the same time as changing the gate polarity may result in indeterminate operation.

Figure 19-5. Timer1 Gate Toggle Mode

Rev. 30-000138A
5/25/2017



19.7.4 Timer1 Gate Single Pulse Mode

When Timer1 Gate Single Pulse mode is enabled, it is possible to capture a single pulse gate event. Timer1 Gate Single Pulse mode is first enabled by setting the **GSPM** bit in the TxGCON register. Next, the **GGO/DONE** bit in the TxGCON register must be set. The Timer1 will be fully enabled on the next incrementing edge. On the next trailing edge of the pulse, the **GGO/DONE** bit will automatically be cleared. No other gate events will be allowed to increment Timer1 until the **GGO/DONE** bit is once again set in software.

Clearing the GSPM bit will also clear the **GGO/DONE** bit. See figure below for timing details.

Enabling the Toggle mode and the Single Pulse mode simultaneously will permit both sections to work together. This allows the cycle times on the Timer1 gate source to be measured. See figure below for timing details.

Figure 19-6. Timer1 Gate Single Pulse Mode

Rev. 30-00139A
5/25/2017

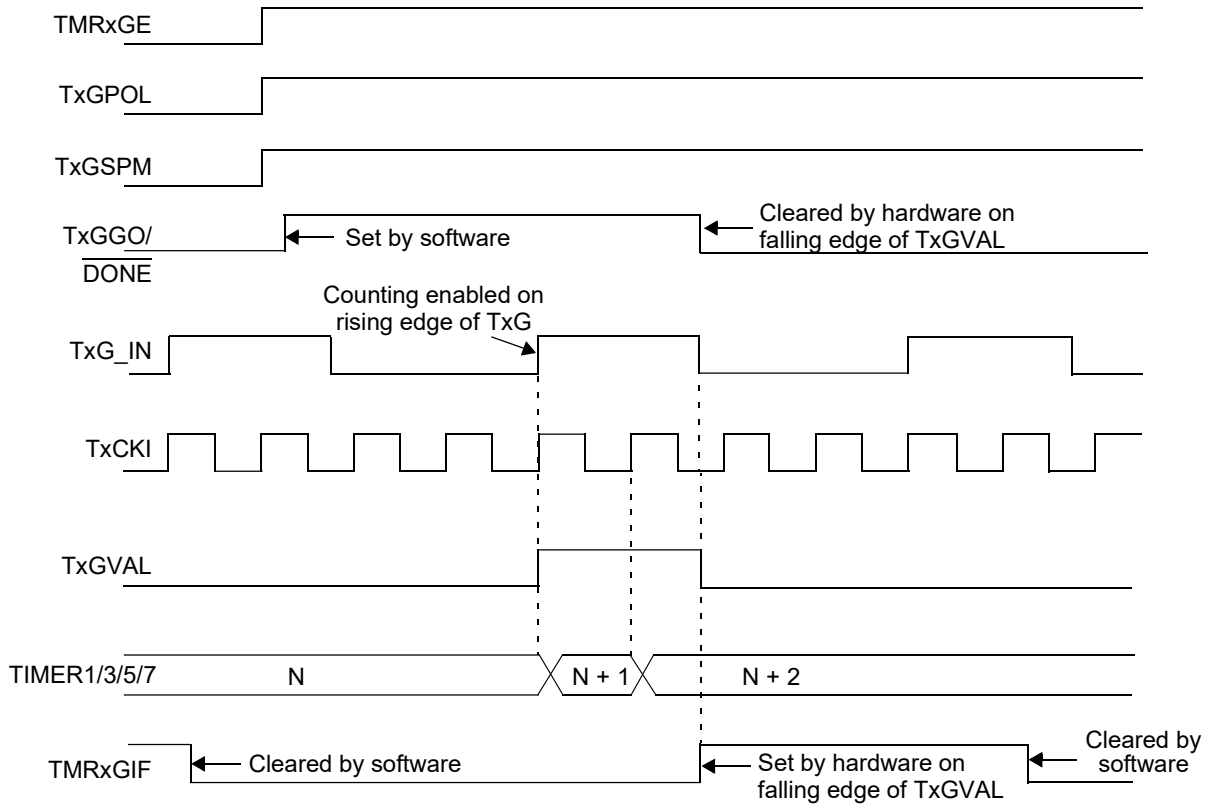
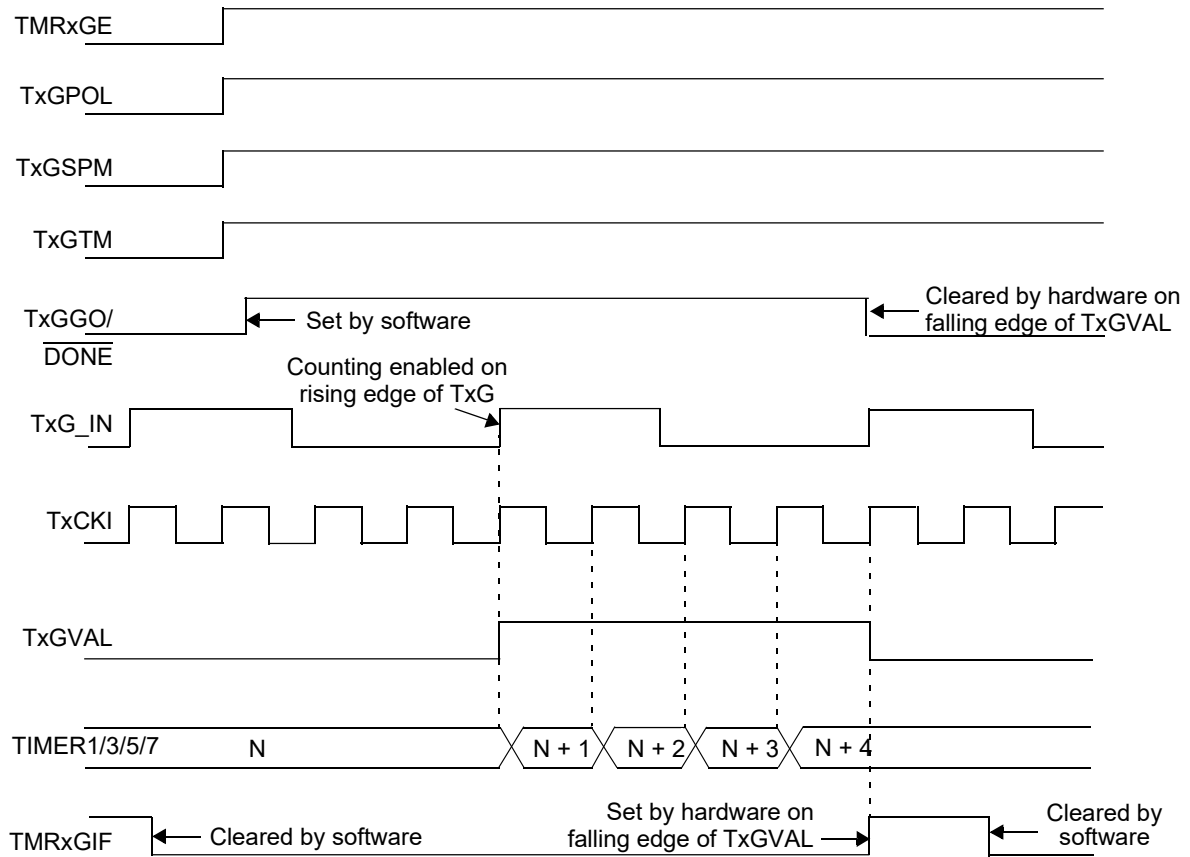


Figure 19-7. Timer1 Gate Single Pulse and Toggle Combined Mode

Rev. 30-000140A
5/25/2017



19.7.5 Timer1 Gate Value Status

When Timer1 Gate Value Status is utilized, it is possible to read the most current level of the gate control value. The value is stored in the GVAL bit in the TxGCON register. The GVAL bit is valid even when the Timer1 gate is not enabled (GE bit is cleared).

19.7.6 Timer1 Gate Event Interrupt

When Timer1 gate event interrupt is enabled, it is possible to generate an interrupt upon the completion of a gate event. When the falling edge of GVAL occurs, the TMRxGIF flag bit in the PIR5 register will be set. If the TMRxGIE bit in the PIE5 register is set, then an interrupt will be recognized.

The TMRxGIF flag bit operates even when the Timer1 gate is not enabled (GE bit is cleared).

For more information on selecting high or low priority status for the Timer1 gate event interrupt see the Interrupts chapter.

19.8 Timer1 Interrupt

The Timer1 register pair (TMRxH:TMRxL) increments to 0xFFFF and rolls over to 0x0000. When Timer1 rolls over, the Timer1 Interrupt Flag (TMR1IF) bit of the PIR4 register is set. To enable the interrupt-on-rollover, the following bits must be set:

- TMRxON bit of the TxCON register
- TMRxIE bits of the PIE4 register

- PEIE/GIEL bit of the INTCON register
- GIE/GIEH bit of the INTCON register

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

For more information on selecting high or low priority status for the Timer1 overflow interrupt, see the Interrupts chapter.



Important: The TMRxH:TMRxL register pair and the TMRxIF bit should be cleared before enabling interrupts.

19.9 Timer1 Operation During Sleep

Timer1 can only operate during Sleep when configured for 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:

- TMRxON bit of the TxCON register must be set
- TMRxIE bit of the PIEx register must be set
- PEIE/GIEL bit of the INTCON register must be set
- $\overline{\text{TxSYNC}}$ bit of the TxCON register must be set
- Configure the TMRxCLK register for using secondary oscillator as the clock source
- Enable the SOSSEN bit of the OSCEN register

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

The secondary oscillator will continue to operate in Sleep regardless of the $\overline{\text{TxSYNC}}$ bit setting.

19.10 CCP Capture/Compare Time Base

The CCP modules use the TMRxH:TMRxL register pair as the time base when operating in Capture or Compare mode.

In Capture mode, the value in the TMRxH:TMRxL register pair is copied into the CCPRxH:CCPRxL register pair on a configured event.

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

For more information, see Capture/Compare/PWM Module(CCP) chapter.

Related Links

[22. Capture/Compare/PWM Module](#)

19.11 CCP Special Event Trigger

When any of the CCPs are configured to trigger a special event, the trigger will clear the TMRxH:TMRxL register pair. This special event does not cause a Timer1 interrupt. The CCP module may still be configured to generate a CCP interrupt.

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

Timer1 should be synchronized and $F_{\text{OSC}}/4$ should be selected as the clock source in order 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 TMRxH or TMRxL coincides with a Special Event Trigger from the CCP, the write will take precedence.

19.12 Peripheral Module Disable

When a peripheral is not used or inactive, the module can be disabled by setting the Module Disable bit in the PMD registers. This will reduce power consumption to an absolute minimum. Setting the PMD bits holds the module in Reset and disconnects the module's clock source. The Module Disable bits for Timer1 (TMR1MD) are in the PMD1 register. See Peripheral Module Disable (PMD) chapter for more information.

Related Links

[13.4 Register Summary - PMD](#)

19.13 Register Summary - Timer1

| Address | Name | Bit Pos. | | | | | | | | |
|-------------|----------|----------|------------|------|-----------|------|----------|----------|------|----|
| 0x00 ... | Reserved | | | | | | | | | |
| 0x020B | | | | | | | | | | |
| 0x020C | TMR1 | 7:0 | TMRx[7:0] | | | | | | | |
| | | 15:8 | TMRx[15:8] | | | | | | | |
| 0x020E | T1CON | 7:0 | | | CKPS[1:0] | | | SYNC | RD16 | ON |
| 0x020F | T1GCON | 7:0 | GE | GPOL | GTM | GSPM | GGO/DONE | GVAL | | |
| 0x0210 | TMR1GATE | 7:0 | | | | | | GSS[4:0] | | |
| 0x0211 | TMR1CLK | 7:0 | | | | | | CS[4:0] | | |
| 0x0212 | TMR3 | 7:0 | TMRx[7:0] | | | | | | | |
| | | 15:8 | TMRx[15:8] | | | | | | | |
| 0x0214 | T3CON | 7:0 | | | CKPS[1:0] | | | SYNC | RD16 | ON |
| 0x0215 | T3GCON | 7:0 | GE | GPOL | GTM | GSPM | GGO/DONE | GVAL | | |
| 0x0216 | TMR3GATE | 7:0 | | | | | | GSS[4:0] | | |
| 0x0217 | TMR3CLK | 7:0 | | | | | | CS[4:0] | | |
| 0x0218 | TMR5 | 7:0 | TMRx[7:0] | | | | | | | |
| | | 15:8 | TMRx[15:8] | | | | | | | |
| 0x021A | T5CON | 7:0 | | | CKPS[1:0] | | | SYNC | RD16 | ON |
| 0x021B | T5GCON | 7:0 | GE | GPOL | GTM | GSPM | GGO/DONE | GVAL | | |
| 0x021C | TMR5GATE | 7:0 | | | | | | GSS[4:0] | | |
| 0x021D | TMR5CLK | 7:0 | | | | | | CS[4:0] | | |

19.14 Register Definitions: Timer1

19.14.1 Timer Register

Name: TMRx
Address: 0x20C,0x212,0x218

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|--------|------------|-----|-----|-----|-----|-----|-----|-----|
| | TMRx[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TMRx[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bits 15:0 – TMRx[15:0] Timer Register Value

Note: The individual bytes in this multi-byte register can be accessed with the following register names:

- TMRxH: Accesses the high byte TMRx[15:8]
- TMRxL: Accesses the low byte TMRx[7:0]

Reset States: POR/BOR = 0000000000000000
 All Other Resets = uuuuuuuuuuuuuuuuu

19.14.2 TxCON

Name: TxCON
Address: 0x20E,0x214,0x21A

Timer Control Register

| | | | | | | | | |
|--------|---|---|-----------|-----|---|------|------|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | | CKPS[1:0] | | | SYNC | RD16 | ON |
| Access | | | R/W | R/W | | R/W | R/W | R/W |
| Reset | | | 0 | 0 | | 0 | 0 | 0 |

Bits 5:4 – CKPS[1:0] Timer Input Clock Prescale Select bits

Reset States: POR/BOR = 00
 All Other Resets = uu

| Value | Description |
|-------|--------------------|
| 11 | 1:8 Prescale value |
| 10 | 1:4 Prescale value |
| 01 | 1:2 Prescale value |
| 00 | 1:1 Prescale value |

Bit 2 – SYNC Timer External Clock Input Synchronization Control bit

Reset States: POR/BOR = 0
 All Other Resets = u

| Value | Condition | Description |
|-------|-------------------------------|---|
| X | $CS = F_{OSC}/4$ or F_{OSC} | This bit is ignored. Timer uses the incoming clock as is. |
| 1 | Else | Do not synchronize external clock input |
| 0 | Else | Synchronize external clock input with system clock |

Bit 1 – RD16 16-Bit Read/Write Mode Enable bit

Reset States: POR/BOR = 0
 All Other Resets = u

| Value | Description |
|-------|--|
| 1 | Enables register read/write of Timer in one 16-bit operation |
| 0 | Enables register read/write of Timer in two 8-bit operations |

Bit 0 – ON Timer On bit

Reset States: POR/BOR = 0
 All Other Resets = u

| Value | Description |
|-------|----------------|
| 1 | Enables Timer |
| 0 | Disables Timer |

19.14.3 TxGCON

Name: TxGCON
Address: 0x20F,0x215,0x21B

Timer Gate Control Register

| | | | | | | | | |
|--------|-----|------|-----|------|----------|------|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | GE | GPOL | GTM | GSPM | GGO/DONE | GVAL | | |
| Access | R/W | R/W | R/W | R/W | R/W | RO | | |
| Reset | 0 | 0 | 0 | 0 | 0 | x | | |

Bit 7 – GE Timer Gate Enable bit

Reset States: POR/BOR = 0
 All Other Resets = u

| Value | Condition | Description |
|-------|-----------|---|
| 1 | ON = 1 | Timer counting is controlled by the Timer gate function |
| 0 | ON = 1 | Timer is always counting |
| X | ON = 0 | This bit is ignored |

Bit 6 – GPOL Timer Gate Polarity bit

Reset States: POR/BOR = 0
 All Other Resets = u

| Value | Description |
|-------|--|
| 1 | Timer gate is active-high (Timer counts when gate is high) |
| 0 | Timer gate is active-low (Timer counts when gate is low) |

Bit 5 – GTM Timer Gate Toggle Mode bit

Timer Gate Flip-Flop Toggles on every rising edge when Toggle mode is enabled.
 Reset States: POR/BOR = 0
 All Other Resets = u

| Value | Description |
|-------|--|
| 1 | Timer Gate Toggle mode is enabled |
| 0 | Timer Gate Toggle mode is disabled and Toggle flip-flop is cleared |

Bit 4 – GSPM Timer Gate Single Pulse Mode bit

Reset States: POR/BOR = 0
 All Other Resets = u

| Value | Description |
|-------|--|
| 1 | Timer Gate Single Pulse mode is enabled and is controlling Timer gate) |
| 0 | Timer Gate Single Pulse mode is disabled |

Bit 3 – GGO/DONE Timer Gate Single Pulse Acquisition Status bit

This bit is automatically cleared when TxGSPM is cleared.
 Reset States: POR/BOR = 0
 All Other Resets = u

| Value | Description |
|-------|--|
| 1 | Timer Gate Single Pulse Acquisition is ready, waiting for an edge |
| 0 | Timer Gate Single Pulse Acquisition has completed or has not been started. |

Bit 2 – GVAL Timer Gate Current State bit

Indicates the current state of the timer gate that could be provided to TMRxH:TMRxL
 Unaffected by Timer Gate Enable (TMRxGE)

19.14.4 TMRxGATE

Name: TMRxGATE
Address: 0x210,0x216,0x21C

Timer Gate Source Selection Register

| | | | | | | | | |
|--------|---|---|---|----------|-----|-----|-----|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | | | GSS[4:0] | | | | |
| Access | | | | R/W | R/W | R/W | R/W | R/W |
| Reset | | | | 0 | 0 | 0 | 0 | 0 |

Bits 4:0 – GSS[4:0] Timer Gate Source Selection bits

Table 19-3. Timer Gate Sources

| GSS | Gate Source | | |
|-------------|------------------------|------------------------|------------------------|
| | Timer1 | Timer3 | Timer5 |
| 11111-11001 | Reserved | Reserved | Reserved |
| 10110 | CLC4_out | CLC4_out | CLC4_out |
| 10101 | CLC3_out | CLC3_out | CLC3_out |
| 10100 | CLC2_out | CLC2_out | CLC2_out |
| 10011 | CLC1_out | CLC1_out | CLC1_out |
| 10010 | ZCD1_output | ZCD1_output | ZCD1_output |
| 10001 | C2OUT_sync | C2OUT_sync | C2OUT_sync |
| 10000 | C1OUT_sync | C1OUT_sync | C1OUT_sync |
| 01111 | NCO1_out | NCO1_out | NCO1_out |
| 01110 | PWM7_out | PWM7_out | PWM7_out |
| 01101 | PWM6_out | PWM6_out | PWM6_out |
| 01100 | CCP4_out | CCP4_out | CCP4_out |
| 01011 | CCP3_out | CCP3_out | CCP3_out |
| 01010 | CCP2_out | CCP2_out | CCP2_out |
| 01001 | CCP1_out | CCP1_out | CCP1_out |
| 01000 | SMT1_overflow | SMT1_overflow | SMT1_overflow |
| 00111 | TMR6_postscaled output | TMR6_postscaled output | TMR6_postscaled output |
| 00110 | Timer5 overflow output | Timer5 overflow output | Reserved |
| 00101 | TMR4_postscaled output | TMR4_postscaled output | TMR4_postscaled output |
| 00100 | Timer3 overflow output | Reserved | Timer3 overflow output |
| 00011 | TMR2_postscaled output | TMR2_postscaled output | TMR2_postscaled output |
| 00010 | Reserved | Timer1 overflow output | Timer1 overflow output |
| 00001 | Timer0 overflow output | Timer0 overflow output | Timer0 overflow output |
| 00000 | T1GPPS | T3GPPS | T5GPPS |

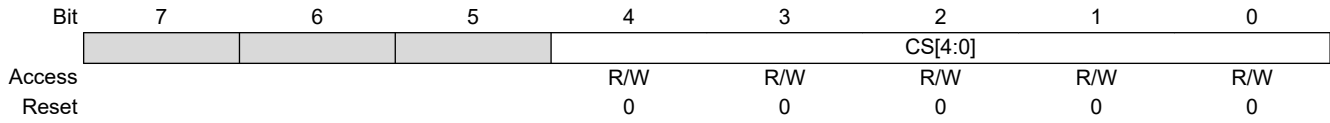
Reset States: POR/BOR = 00000

All Other Resets = uuuuu

19.14.5 TMRxCLK

Name: TMRxCLK
 Address: 0x211,0x217,0x21D

Timer Clock Source Selection Register



Bits 4:0 – CS[4:0] Timer Clock Source Selection bits

Table 19-4. Timer Clock Sources

| CS | Clock Source | | |
|-------------|------------------------|------------------------|------------------------|
| | Timer1 | Timer3 | Timer5 |
| 11111-10001 | Reserved | Reserved | Reserved |
| 10000 | CLC4_out | CLC4_out | CLC4_out |
| 01111 | CLC3_out | CLC3_out | CLC3_out |
| 01110 | CLC2_out | CLC2_out | CLC2_out |
| 01101 | CLC1_out | CLC1_out | CLC1_out |
| 01100 | Timer5 overflow output | Timer5 overflow output | Reserved |
| 01011 | Timer3 overflow output | Reserved | Timer3 overflow output |
| 01010 | Reserved | Timer1 overflow output | Timer1 overflow output |
| 01001 | Timer0 overflow output | Timer0 overflow output | Timer0 overflow output |
| 01000 | CLKR output | CLKR output | CLKR output |
| 00111 | SOSC | SOSC | SOSC |
| 00110 | MFINTOSC (32 kHz) | MFINTOSC (32 kHz) | MFINTOSC (32 kHz) |
| 00101 | MFINTOSC (500 kHz) | MFINTOSC (500 kHz) | MFINTOSC (500 kHz) |
| 00100 | LFINTOSC | LFINTOSC | LFINTOSC |
| 00011 | HFINTOSC | HFINTOSC | HFINTOSC |
| 00010 | F _{Osc} | F _{Osc} | F _{Osc} |
| 00001 | F _{Osc} /4 | F _{Osc} /4 | F _{Osc} /4 |
| 00000 | T1CKIPPS | T3CKIPPS | T5CKIPPS |

Reset States: POR/BOR = 00000
 All Other Resets = uuuuu

20. TMR2 - Timer2 Module

The Timer2 module is a 8-bit timer that incorporates the following features:

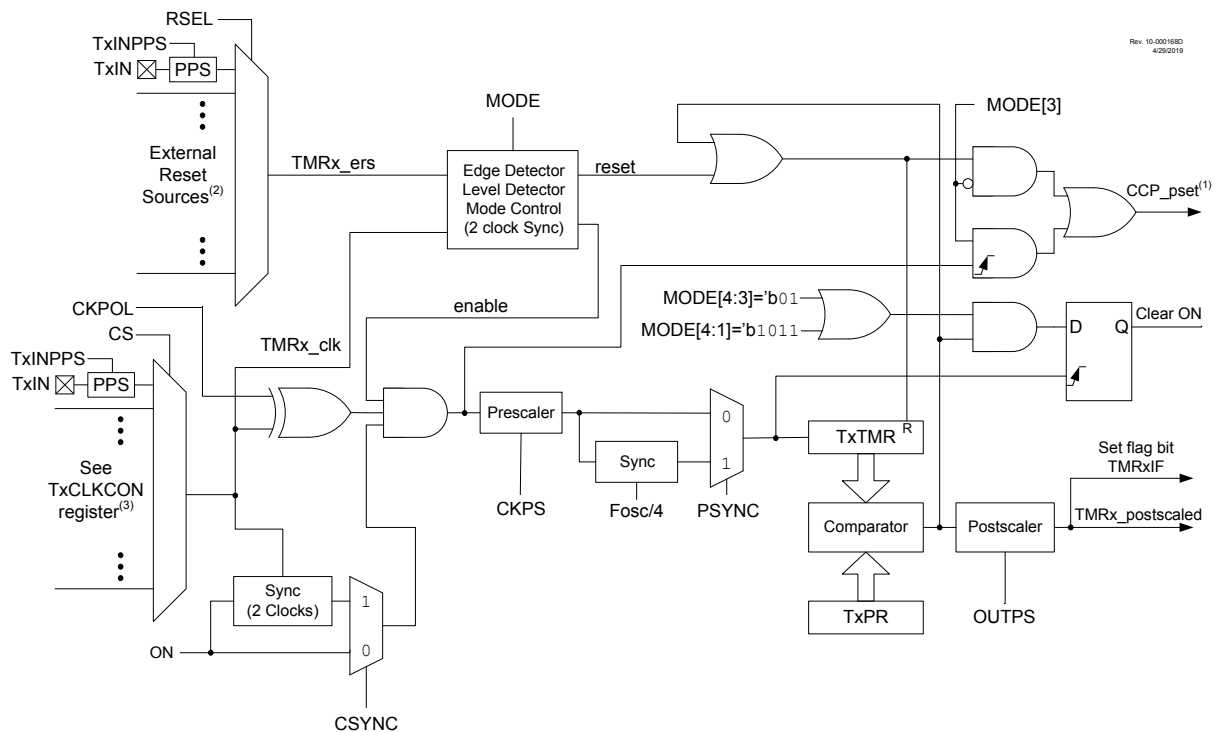
- 8-Bit Timer and Period Registers
- Readable and Writable
- Software Programmable Prescaler (1:1 to 1:128)
- Software Programmable Postscaler (1:1 to 1:16)
- Interrupt on T2TMR Match with T2PR
- One-Shot Operation
- Full Asynchronous Operation
- Includes Hardware Limit Timer (HLT)
- Alternate Clock Sources
- External Timer Reset Signal Sources
- Configurable Timer Reset Operation

See [Figure 20-1](#) for a block diagram of Timer2.



Important: References to module Timer2 apply to all the even numbered timers on this device. (Timer2, Timer4, etc.)

Figure 20-1. Timer2 with Hardware Limit Timer (HLT) Block Diagram



Note:

1. Signal to the CCP to trigger the PWM pulse.
2. See [TxRST](#) for external Reset sources.
3. See [Clock Source Selection](#) table for clock source selection.

20.1 Timer2 Operation

Timer2 operates in three major modes:

- Free Running Period
- One-shot
- Monostable

Within each mode there are several options for starting, stopping, and reset. [Table 20-1](#) lists the options.

In all modes, the T2TMR count register is incremented on the rising edge of the clock signal from the programmable prescaler. When T2TMR equals T2PR, a high level is output to the postscaler counter. T2TMR is cleared on the next clock input.

An external signal from hardware can also be configured to gate the timer operation or force a T2TMR count Reset. In Gate modes the counter stops when the gate is disabled and resumes when the gate is enabled. In Reset modes the T2TMR count is reset on either the level or edge from the external source.

The T2TMR and T2PR registers are both directly readable and writable. The T2TMR register is cleared and the T2PR register initializes to FFh on any device Reset. Both the prescaler and postscaler counters are cleared on the following events:

- A write to the T2TMR register
- A write to the T2CON register
- Any device Reset
- External Reset Source event that resets the timer.



Important: T2TMR is not cleared when T2CON is written.

20.1.1 Free-Running Period Mode

The value of T2TMR is compared to that of the Period register, T2PR, on each clock cycle. When the two values match, the comparator resets the value of T2TMR to 00h on the next cycle and increments the output postscaler counter. When the postscaler count equals the value in the [OUTPS](#) bits of the T2CON register then a one clock period wide pulse occurs on the TMR2_postscaled output, and the postscaler count is cleared.

20.1.2 One-Shot Mode

The One-Shot mode is identical to the Free-Running Period mode except that the ON bit is cleared and the timer is stopped when T2TMR matches T2PR and will not restart until the ON bit is cycled off and on. Postscaler (OUTPS) values other than zero are ignored in this mode because the timer is stopped at the first period event and the postscaler is reset when the timer is restarted.

20.1.3 Monostable Mode

Monostable modes are similar to One-Shot modes except that the ON bit is not cleared and the timer can be restarted by an external Reset event.

20.2 Timer2 Output

The Timer2 module's primary output is TMR2_postscaled, which pulses for a single TMR2_clk period upon each match of the postscaler counter and the OUTPS bits of the T2CON register. The postscaler is incremented each time the T2TMR value matches the T2PR value. This signal can be selected as an input to several other input modules:

- The ADC module, as an auto-conversion trigger
- CWG, as an auto-shutdown source
- The CRC memory scanner, as a trigger for triggered mode

- Gate source for odd numbered timers (Timer1, Timer3, etc.)
- Alternate SPI clock
- Reset signals for other instances of even numbered timers (Timer2, Timer4, etc.)

In addition, the Timer2 is also used by the CCP module for pulse generation in PWM mode. See “*PWM Overview*” and “*Pulse-width Modulation*” sections for more details on setting up Timer2 for use with the CCP and PWM modules.

Related Links

[22.4 PWM Overview](#)

[24. PWM - Pulse-Width Modulation](#)

20.3 External Reset Sources

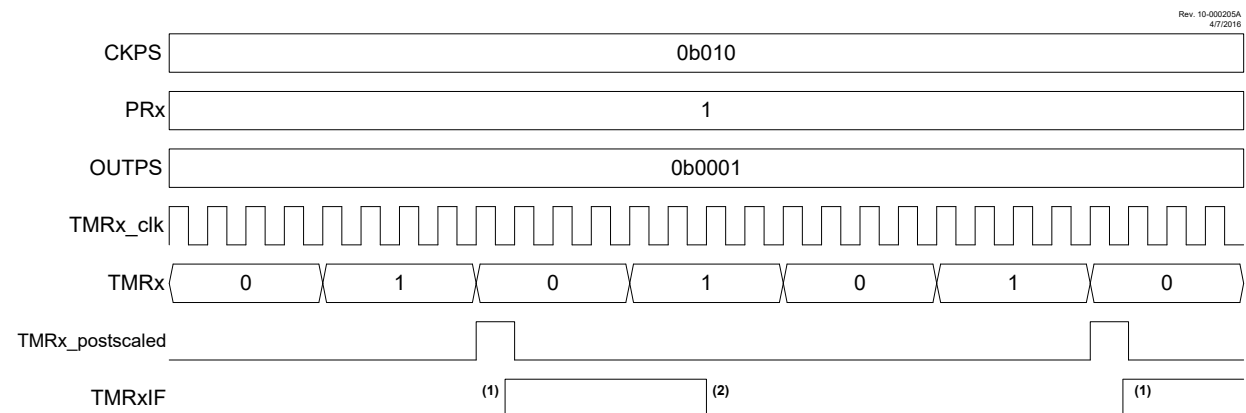
In addition to the clock source, the Timer2 also takes in an external Reset source. This external Reset source is selected for each timer with the corresponding **TxRST** register. This source can control starting and stopping of the timer, as well as resetting the timer, depending on which mode the timer is in. Reset source selections are shown in the following table.

Refer to the [External Reset Sources](#) table for further information.

20.4 Timer2 Interrupt

Timer2 can also generate a device interrupt. The interrupt is generated when the postscaler counter matches with the selected postscaler value (OUTPS bits of T2CON register). The interrupt is enabled by setting the TMR2IE interrupt enable bit. Interrupt timing is illustrated in the figure below.

Figure 20-2. Timer2 Prescaler, Postscaler, and Interrupt Timing Diagram



Note:

1. Setting the interrupt flag is synchronized with the instruction clock.
2. Cleared by software.

20.5 PSYNC bit

Setting the PSYNC bit synchronizes the prescaler output to $F_{OSC}/4$. Setting this bit is required for reading the Timer2 counter register while the selected Timer clock is asynchronous to $F_{OSC}/4$.

Note: Setting PSYNC requires that the output of the prescaler is slower than $F_{OSC}/4$. Setting PSYNC when the output of the prescaler is greater than or equal to $F_{OSC}/4$ may cause unexpected results.

20.6 CSYNC bit

All bits in the Timer2 SFRs are synchronized to $F_{OSC}/4$ by default, not the Timer2 input clock. As such, if the Timer2 input clock is not synchronized to $F_{OSC}/4$, it is possible for the Timer2 input clock to transition at the same time as the

ON bit is set in software, which may cause undesirable behavior and glitches in the counter. Setting the CSYNC bit remedies this problem by synchronizing the ON bit to the Timer2 input clock instead of $F_{OSC}/4$. However, as this synchronization uses an edge of the TMR2 input clock, up to one input clock cycle will be consumed and not counted by the Timer2 when CSYNC is set. Conversely, clearing the CSYNC bit synchronizes the ON bit to $F_{OSC}/4$, which does not consume any clock edges, but has the previously stated risk of glitches.

20.7 Operating Modes

The mode of the timer is controlled by the **MODE** bits of the **T2HLT** register. Edge-Triggered modes require six Timer clock periods between external triggers. Level-Triggered modes require the triggering level to be at least three Timer clock periods long. External triggers are ignored while in Debug mode.

Table 20-1. Operating Modes Table

| Mode | MODE[4:0] | | Output Operation | Operation | Timer Control | | |
|---------------------|-----------|-------|--|---|-------------------------|--------------|--------------------------------------|
| | [4:3] | [2:0] | | | Start | Reset | Stop |
| Free-Running Period | 00 | 000 | Period Pulse | Software gate (Figure 20-3) | ON = 1 | — | ON = 0 |
| | | 001 | | Hardware gate, active-high (Figure 20-4) | ON = 1 and TMRx_ers = 1 | — | ON = 0 or TMRx_ers = 0 |
| | | 010 | | Hardware gate, active-low | ON = 1 and TMRx_ers = 0 | — | ON = 0 or TMRx_ers = 1 |
| | | 011 | Period Pulse with Hardware Reset | Rising or falling edge Reset | ON = 1 | TMRx_ers ↓ | ON = 0 |
| | | 100 | | Rising edge Reset (Figure 20-5) | | TMRx_ers ↑ | |
| | | 101 | | Falling edge Reset | | TMRx_ers ↓ | |
| | | 110 | | Low level Reset | | TMRx_ers = 0 | ON = 0 or TMRx_ers = 0 |
| | | 111 | | High level Reset (Figure 20-6) | | TMRx_ers = 1 | ON = 0 or TMRx_ers = 1 |
| One-shot | 01 | 000 | One-shot | Software start (Figure 20-7) | ON = 1 | — | Next clock after TMRx = PRx (Note 2) |
| | | 001 | Edge-Triggered Start (Note 1) | Rising edge start (Figure 20-8) | ON = 1 and TMRx_ers ↑ | — | |
| | | 010 | | Falling edge start | ON = 1 and TMRx_ers ↓ | — | |
| | | 011 | | Any edge start | ON = 1 and TMRx_ers ↓ | — | |
| | | 100 | Edge-Triggered Start and Hardware Reset (Note 1) | Rising edge start and Rising edge Reset (Figure 20-9) | ON = 1 and TMRx_ers ↑ | TMRx_ers ↑ | |
| | | 101 | | Falling edge start and Falling edge Reset | ON = 1 and TMRx_ers ↓ | TMRx_ers ↓ | |
| | | 110 | | Rising edge start and Low level Reset (Figure 20-10) | ON = 1 and TMRx_ers ↑ | TMRx_ers = 0 | |
| | | 111 | | Falling edge start and High level Reset | ON = 1 and TMRx_ers ↓ | TMRx_ers = 1 | |

.....continued

| Mode | MODE[4:0] | | Output Operation | Operation | Timer Control | | |
|-------------|-----------|-------|--|---|-------------------------|--------------|--|
| | [4:3] | [2:0] | | | Start | Reset | Stop |
| Mono-stable | 10 | 000 | | Reserved | | | |
| | | 001 | Edge-Triggered Start (Note 1) | Rising edge start (Figure 20-11) | ON = 1 and TMRx_ers ↑ | — | ON = 0 or Next clock after TMRx = PRx (Note 3) |
| | | 010 | | Falling edge start | ON = 1 and TMRx_ers ↓ | — | |
| | | 011 | | Any edge start | ON = 1 and TMRx_ers ↓ | — | |
| Reserved | | 100 | Reserved | | | | |
| Reserved | | 101 | Reserved | | | | |
| One-shot | | 110 | Level Triggered Start and Hardware Reset | High level start and Low level Reset (Figure 20-12) | ON = 1 and TMRx_ers = 1 | TMRx_ers = 0 | ON = 0 or Held in Reset (Note 2) |
| | | 111 | | Low level start and High level Reset | ON = 1 and TMRx_ers = 0 | TMRx_ers = 1 | |
| Reserved | 11 | xxx | Reserved | | | | |

Note:

1. If ON = 0 then an edge is required to restart the timer after ON = 1.
2. When T2TMR = T2PR then the next clock clears ON and stops T2TMR at 0x00.
3. When T2TMR = T2PR then the next clock stops T2TMR at 0x00 but does not clear ON.

20.8 Operation Examples

Unless otherwise specified, the following notes apply to the following timing diagrams:

- Both the prescaler and postscaler are set to 1:1 (both the CKPS and OUTPS bits in the T2CON register are cleared).
- The diagrams illustrate any clock except $F_{OSC}/4$ and show clock-sync delays of at least two full cycles for both ON and Timer2_ers. When using $F_{OSC}/4$, the clock-sync delay is at least one instruction period for Timer2_ers; ON applies in the next instruction period.
- ON and Timer2_ers are somewhat generalized, and clock-sync delays may produce results that are slightly different than illustrated.
- The PWM Duty Cycle and PWM output are illustrated assuming that the timer is used for the PWM function of the CCP module as described in the “PWM Overview” section. The signals are not a part of the Timer2 module.

Related Links

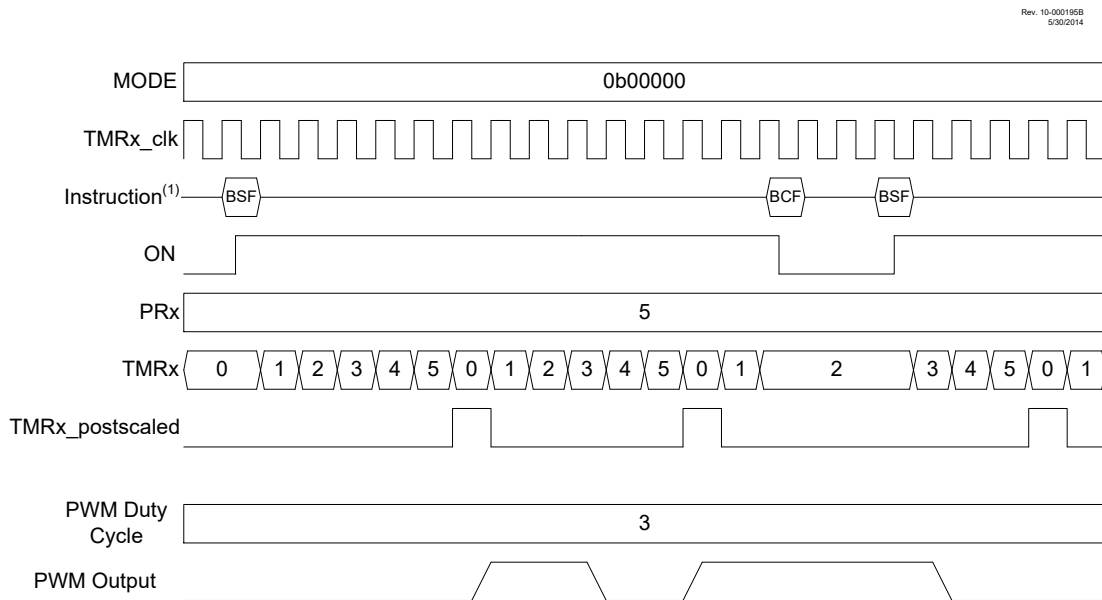
[22.4 PWM Overview](#)

[24. PWM - Pulse-Width Modulation](#)

20.8.1 Software Gate Mode

This mode corresponds to legacy Timer2 operation. The timer increments with each clock input when ON = 1 and does not increment when ON = 0. When the TMRx count equals the PRx period count the timer resets on the next clock and continues counting from 0. Operation with the ON bit software controlled is illustrated in Figure 20-3. With PRx = 5, the counter advances until TMRx = 5, and goes to zero with the next clock.

Figure 20-3. Software Gate Mode Timing Diagram (MODE = 00000)



Note:

1. BSF and BCF represent Bit-Set File and Bit-Clear File instructions executed by the CPU to set or clear the ON bit of TxCON. CPU execution is asynchronous to the timer clock input.

Related Links

[22.4 PWM Overview](#)

[24. PWM - Pulse-Width Modulation](#)

20.8.2 Hardware Gate Mode

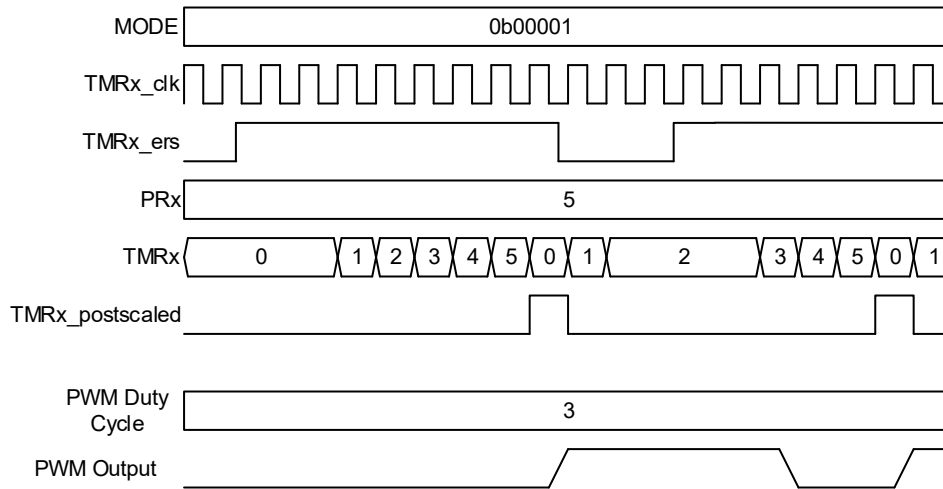
The Hardware Gate modes operate the same as the Software Gate mode except the TMRx_ers external signal can also gate the timer. When used with the CCP, the gating extends the PWM period. If the timer is stopped when the PWM output is high, then the duty cycle is also extended.

When MODE[4:0] = 00001 then the timer is stopped when the external signal is high. When MODE[4:0] = 00010, then the timer is stopped when the external signal is low.

Figure 20-4 illustrates the Hardware Gating mode for MODE[4:0] = 00001 in which a high input level starts the counter.

Figure 20-4. Hardware Gate Mode Timing Diagram (MODE = 00001)

Rev. 10-001198B
5/30/2014



Related Links

[22.4 PWM Overview](#)

[24. PWM - Pulse-Width Modulation](#)

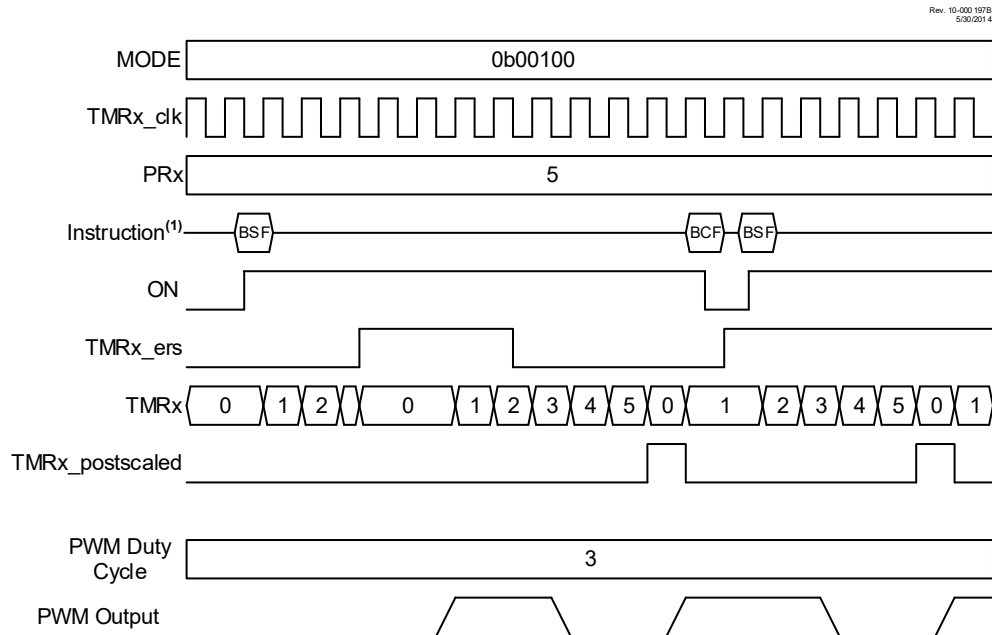
20.8.3 Edge-Triggered Hardware Limit Mode

In Hardware Limit mode, the timer can be reset by the TMRx_ers external signal before the timer reaches the period count. Three types of Resets are possible:

- Reset on rising or falling edge (MODE[4:0] = 00011)
- Reset on rising edge (MODE[4:0] = 00100)
- Reset on falling edge (MODE[4:0] = 00101)

When the timer is used in conjunction with the CCP in PWM mode then an early Reset shortens the period and restarts the PWM pulse after a two clock delay. Refer to [Figure 20-5](#).

Figure 20-5. Edge-Triggered Hardware Limit Mode Timing Diagram
(MODE = 00100)



Note:

1. BSF and BCF represent Bit-Set File and Bit-Clear File instructions executed by the CPU to set or clear the ON bit of TxCON. CPU execution is asynchronous to the timer clock input.

Related Links

[22.4 PWM Overview](#)

[24. PWM - Pulse-Width Modulation](#)

20.8.4 Level-Triggered Hardware Limit Mode

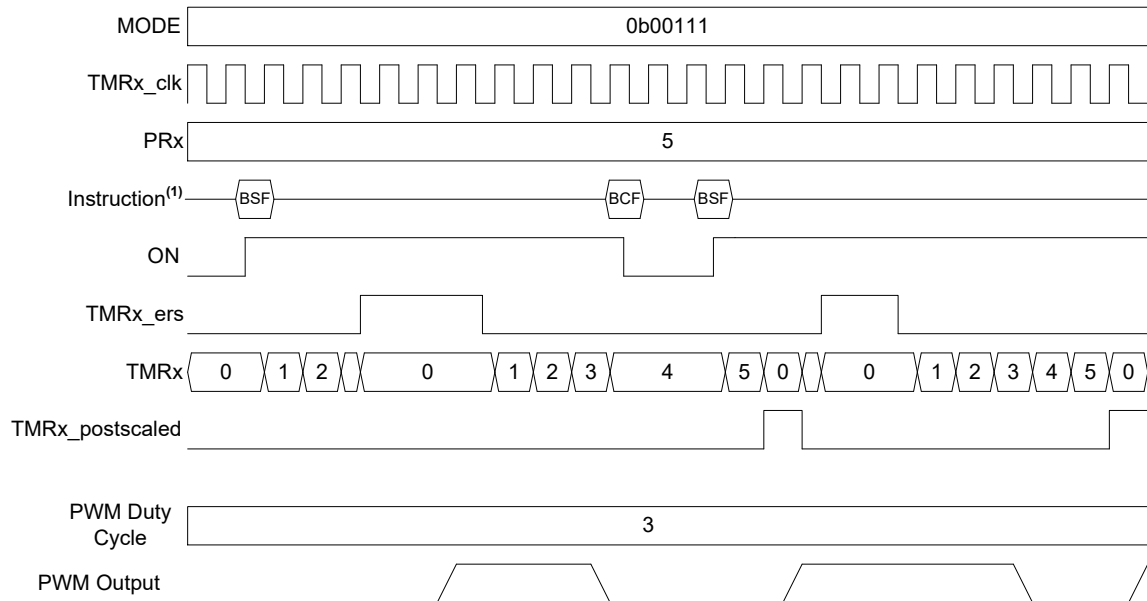
In the Level-Triggered Hardware Limit Timer modes the counter is reset by high or low levels of the external signal TMRx_ers, as shown in [Figure 20-6](#). Selecting MODE[4:0] = 00110 will cause the timer to reset on a low level external signal. Selecting MODE[4:0] = 00111 will cause the timer to reset on a high level external signal. In the example, the counter is reset while TMRx_ers = 1. ON is controlled by BSF and BCF instructions. When ON = 0 the external signal is ignored.

When the CCP uses the timer as the PWM time base then the PWM output will be set high when the timer starts counting and then set low only when the timer count matches the CCPRx value. The timer is reset when either the timer count matches the PRx value or two clock periods after the external Reset signal goes true and stays true.

The timer starts counting, and the PWM output is set high, on either the clock following the PRx match or two clocks after the external Reset signal relinquishes the Reset. The PWM output will remain high until the timer counts up to match the CCPRx pulse width value. If the external Reset signal goes true while the PWM output is high then the PWM output will remain high until the Reset signal is released allowing the timer to count up to match the CCPRx value.

Figure 20-6. Level-Triggered Hardware Limit Mode Timing Diagram
(MODE = 00111)

Rev. 10-000198B
5/30/2014



Note:

1. BSF and BCF represent Bit-Set File and Bit-Clear File instructions executed by the CPU to set or clear the ON bit of TxCON. CPU execution is asynchronous to the timer clock input.

Related Links

- [22.4 PWM Overview](#)
- [24. PWM - Pulse-Width Modulation](#)

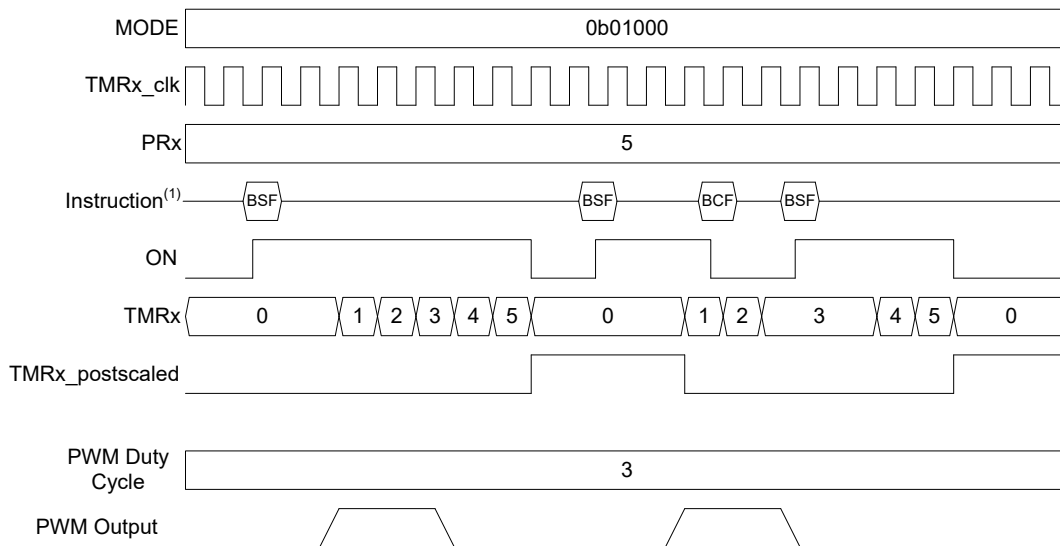
20.8.5 Software Start One-Shot Mode

In One-Shot mode the timer resets and the ON bit is cleared when the timer value matches the PRx period value. The ON bit must be set by software to start another timer cycle. Setting MODE[4:0] = 01000 selects One-Shot mode which is illustrated in Figure 20-7. In the example, ON is controlled by BSF and BCF instructions. In the first case, a BSF instruction sets ON and the counter runs to completion and clears ON. In the second case, a BSF instruction starts the cycle, BCF/BSF instructions turn the counter off and on during the cycle, and then it runs to completion.

When One-Shot mode is used in conjunction with the CCP PWM operation the PWM pulse drive starts concurrent with setting the ON bit. Clearing the ON bit while the PWM drive is active will extend the PWM drive. The PWM drive will terminate when the timer value matches the CCPRx pulse width value. The PWM drive will remain off until software sets the ON bit to start another cycle. If the software clears the ON bit after the CCPRx match but before the PRx match then the PWM drive will be extended by the length of time the ON bit remains cleared. Another timing cycle can only be initiated by setting the ON bit after it has been cleared by a PRx period count match.

Figure 20-7. Software Start One-Shot Mode Timing Diagram (MODE = 01000)

Rev. 10-000198B
4/7/2016



Note:

1. BSF and BCF represent Bit-Set File and Bit-Clear File instructions executed by the CPU to set or clear the ON bit of TxCON. CPU execution is asynchronous to the timer clock input.

Related Links

- [22.4 PWM Overview](#)
- [24. PWM - Pulse-Width Modulation](#)

20.8.6 Edge-Triggered One-Shot Mode

The Edge-Triggered One-Shot modes start the timer on an edge from the external signal input, after the ON bit is set, and clear the ON bit when the timer matches the PRx period value. The following edges will start the timer:

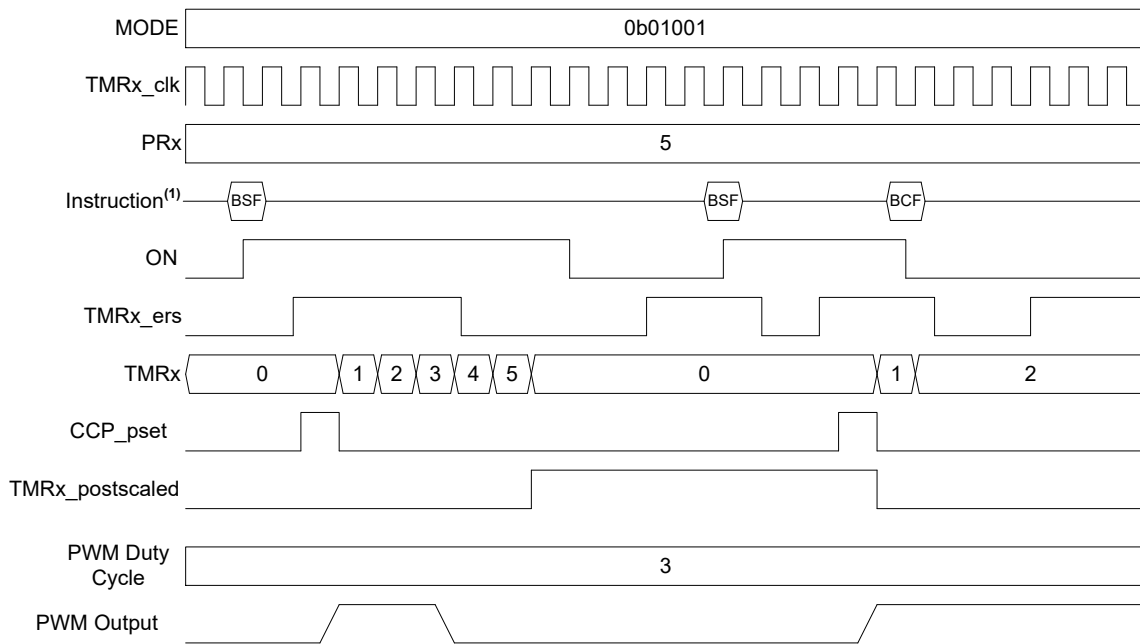
- Rising edge (MODE[4:0] = 01001)
- Falling edge (MODE[4:0] = 01010)
- Rising or Falling edge (MODE[4:0] = 01011)

If the timer is halted by clearing the ON bit then another TMRx_ers edge is required after the ON bit is set to resume counting. [Figure 20-8](#) illustrates operation in the rising edge One-Shot mode.

When Edge-Triggered One-Shot mode is used in conjunction with the CCP then the edge-trigger will activate the PWM drive and the PWM drive will deactivate when the timer matches the CCPRx pulse width value and stay deactivated when the timer halts at the PRx period count match.

Figure 20-8. Edge-Triggered One-Shot Mode Timing Diagram (MODE = 01001)

Rev. 10-000200B
5/19/2016



Note:

1. **BSF** and **BCF** represent Bit-Set File and Bit-Clear File instructions executed by the CPU to set or clear the ON bit of TxCON. CPU execution is asynchronous to the timer clock input.

Related Links

- [22.4 PWM Overview](#)
- [24. PWM - Pulse-Width Modulation](#)

20.8.7 Edge-Triggered Hardware Limit One-Shot Mode

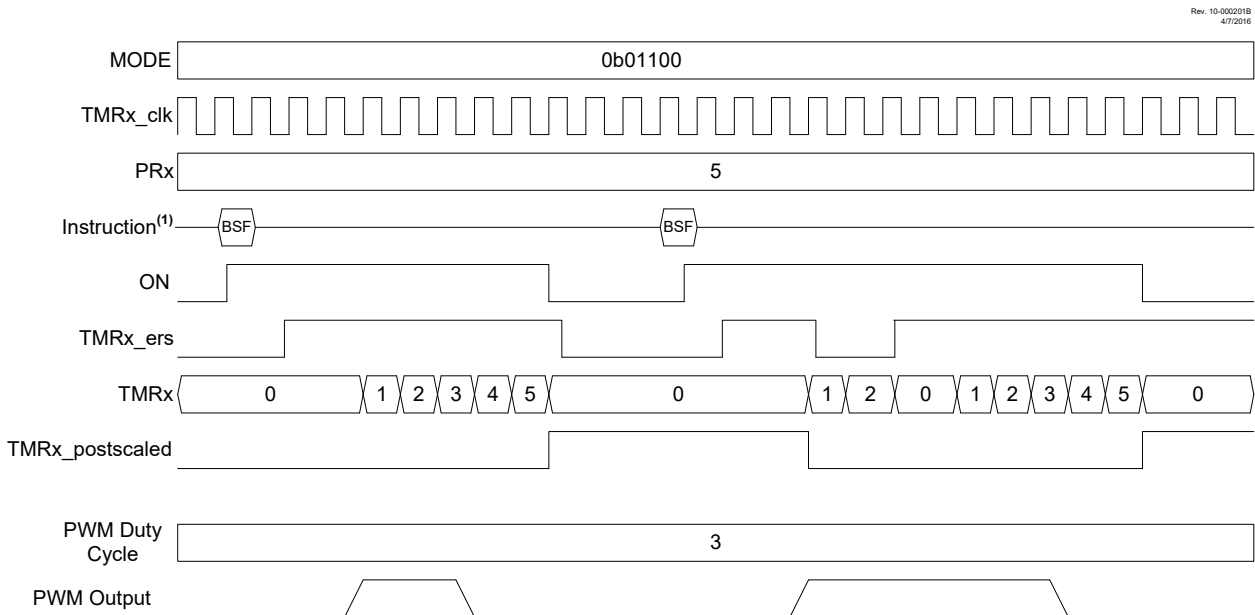
In Edge-Triggered Hardware Limit One-Shot modes the timer starts on the first external signal edge after the ON bit is set and resets on all subsequent edges. Only the first edge after the ON bit is set is needed to start the timer. The counter will resume counting automatically two clocks after all subsequent external Reset edges. Edge triggers are as follows:

- Rising edge start and Reset (MODE[4:0] = 01100)
- Falling edge start and Reset (MODE[4:0] = 01101)

The timer resets and clears the ON bit when the timer value matches the PRx period value. External signal edges will have no effect until after software sets the ON bit. [Figure 20-9](#) illustrates the rising edge hardware limit one-shot operation.

When this mode is used in conjunction with the CCP then the first starting edge trigger, and all subsequent Reset edges, will activate the PWM drive. The PWM drive will deactivate when the timer matches the CCPRx pulse-width value and stay deactivated until the timer halts at the PRx period match unless an external signal edge resets the timer before the match occurs.

Figure 20-9. Edge-Triggered Hardware Limit One-Shot Mode Timing Diagram (MODE = 01100)



Note:

1. BSF and BCF represent Bit-Set File and Bit-Clear File instructions executed by the CPU to set or clear the ON bit of TxCON. CPU execution is asynchronous to the timer clock input.

Related Links

[22.4 PWM Overview](#)

[24. PWM - Pulse-Width Modulation](#)

20.8.8 Level Reset, Edge-Triggered Hardware Limit One-Shot Modes

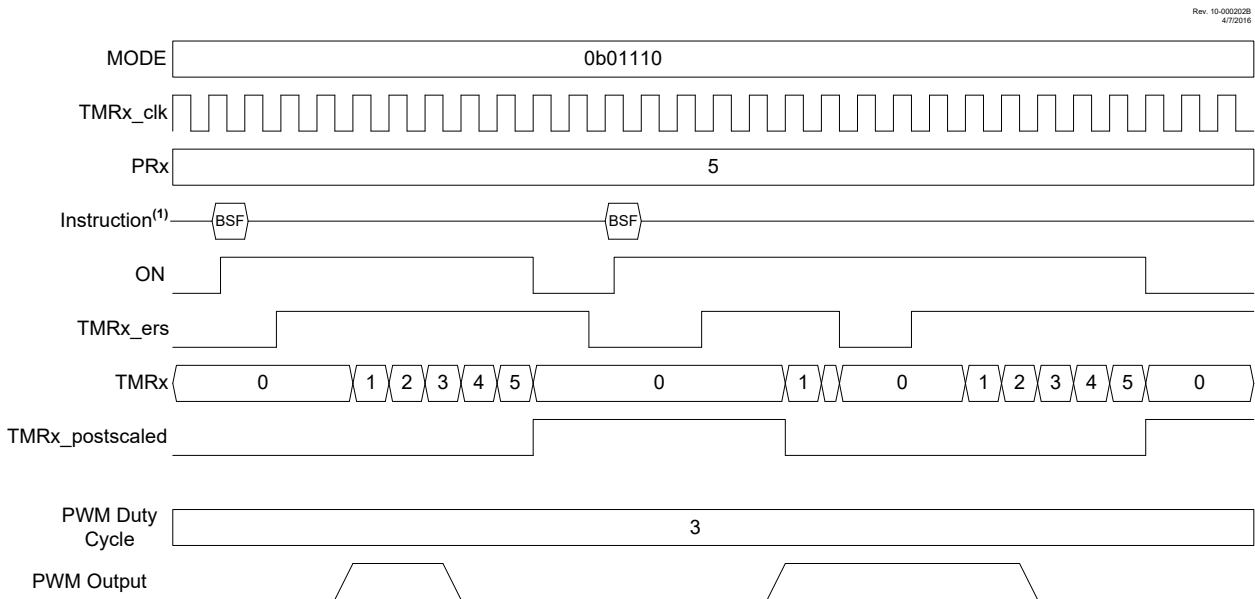
In Level -Triggered One-Shot mode the timer count is reset on the external signal level and starts counting on the rising/falling edge of the transition from Reset level to the active level while the ON bit is set. Reset levels are selected as follows:

- Low Reset level (MODE[4:0] = 01110)
- High Reset level (MODE[4:0] = 01111)

When the timer count matches the PRx period count, the timer is reset and the ON bit is cleared. When the ON bit is cleared by either a PRx match or by software control, a new external signal edge is required after the ON bit is set to start the counter.

When Level-Triggered Reset One-Shot mode is used in conjunction with the CCP PWM operation, the PWM drive goes active with the external signal edge that starts the timer. The PWM drive goes inactive when the timer count equals the CCPRx pulse width count. The PWM drive does not go active when the timer count clears at the PRx period count match.

Figure 20-10. Low Level Reset, Edge-Triggered Hardware Limit One-Shot Mode Timing Diagram (MODE = 01110)



Note:

1. BSF and BCF represent Bit-Set File and Bit-Clear File instructions executed by the CPU to set or clear the ON bit of TxCON. CPU execution is asynchronous to the timer clock input.

Related Links

[22.4 PWM Overview](#)

[24. PWM - Pulse-Width Modulation](#)

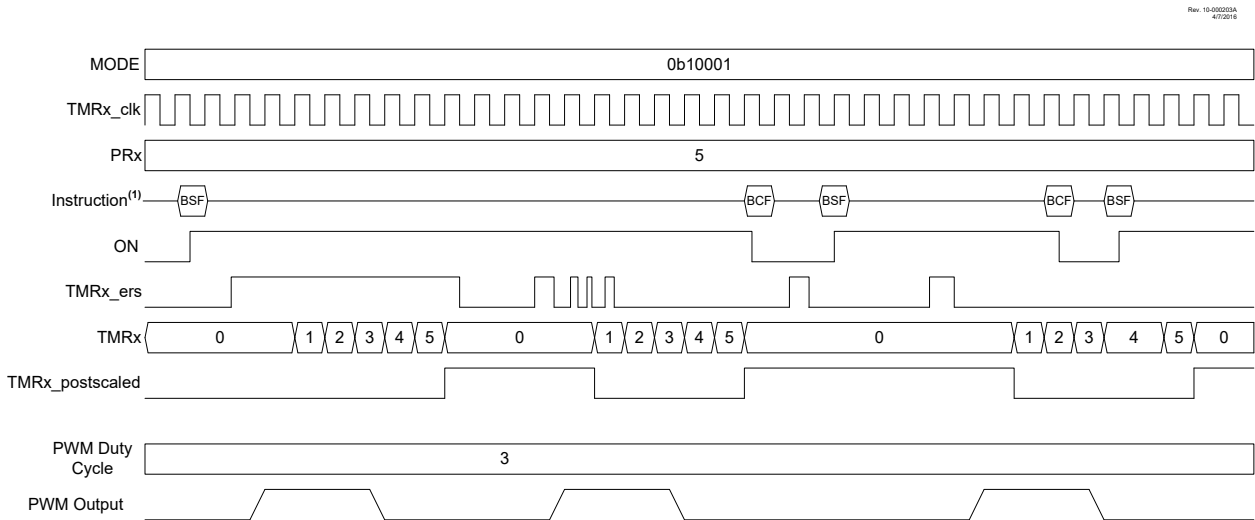
20.8.9 Edge-Triggered Monostable Modes

The Edge-Triggered Monostable modes start the timer on an edge from the external Reset signal input, after the ON bit is set, and stop incrementing the timer when the timer matches the PRx period value. The following edges will start the timer:

- Rising edge (MODE[4:0] = 10001)
- Falling edge (MODE[4:0] = 10010)
- Rising or Falling edge (MODE[4:0] = 10011)

When an Edge-Triggered Monostable mode is used in conjunction with the CCP PWM operation, the PWM drive goes active with the external Reset signal edge that starts the timer, but will not go active when the timer matches the PRx value. While the timer is incrementing, additional edges on the external Reset signal will not affect the CCP PWM.

Figure 20-11. Rising Edge-Triggered Monostable Mode Timing Diagram (MODE = 10001)



Note:

1. BSF and BCF represent Bit-Set File and Bit-Clear File instructions executed by the CPU to set or clear the ON bit of TxCON. CPU execution is asynchronous to the timer clock input.

Related Links

[22.4 PWM Overview](#)

[24. PWM - Pulse-Width Modulation](#)

20.8.10 Level-Triggered Hardware Limit One-Shot Modes

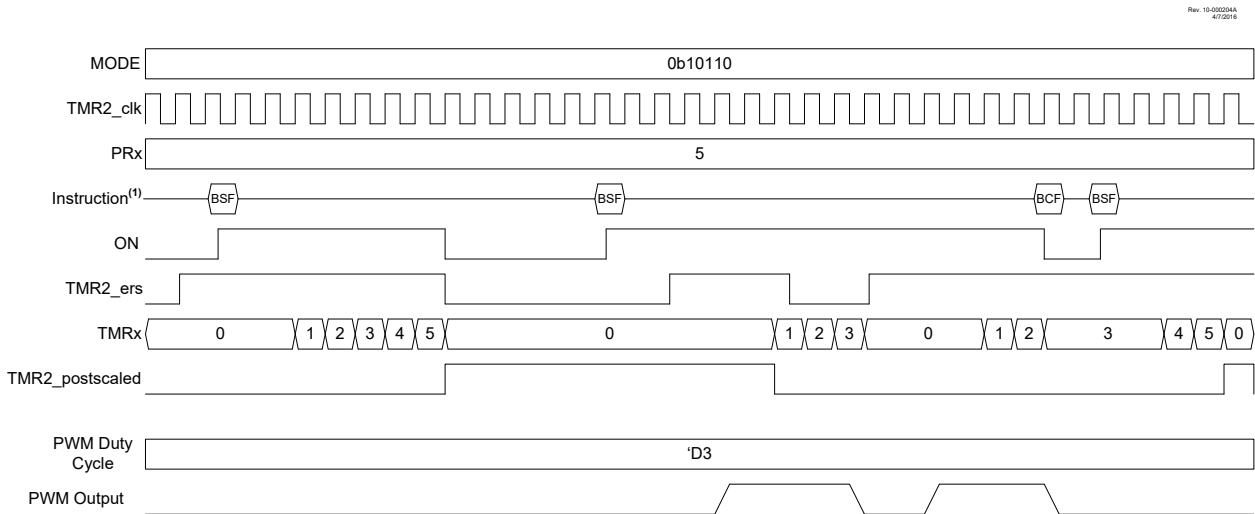
The Level-Triggered Hardware Limit One-Shot modes hold the timer in Reset on an external Reset level and start counting when both the ON bit is set and the external signal is not at the Reset level. If one of either the external signal is not in Reset or the ON bit is set, then the other signal being set/made active will start the timer. Reset levels are selected as follows:

- Low Reset level (MODE[4:0] = 10110)
- High Reset level (MODE[4:0] = 10111)

When the timer count matches the PRx period count, the timer is reset and the ON bit is cleared. When the ON bit is cleared by either a PRx match or by software control, the timer will stay in Reset until both the ON bit is set and the external signal is not at the Reset level.

When Level-Triggered Hardware Limit One-Shot modes are used in conjunction with the CCP PWM operation, the PWM drive goes active with either the external signal edge or the setting of the ON bit, whichever of the two starts the timer.

Figure 20-12. Level-Triggered Hardware Limit one-Shot Mode Timing Diagram (MODE = 10110)



Note:

1. `BSF` and `BCF` represent Bit-Set File and Bit-Clear File instructions executed by the CPU to set or clear the ON bit of TxCON. CPU execution is asynchronous to the timer clock input.

Related Links

[22.4 PWM Overview](#)

[24. PWM - Pulse-Width Modulation](#)

20.9 Timer2 Operation During Sleep

When `PSYNC` = 1, Timer2 cannot be operated while the processor is in Sleep mode. The contents of the T2TMR and T2PR registers will remain unchanged while processor is in Sleep mode.

When `PSYNC` = 0, Timer2 will operate in Sleep as long as the clock source selected is also still running. If any internal oscillator is selected as the clock source, it will stay active during Sleep mode.

20.10 Register Summary - Timer2

| Address | Name | Bit Pos. | | | | | | | |
|-------------|----------|----------|------------|-----------|-------|-----------|------------|--|--|
| 0x00 ... | Reserved | | | | | | | | |
| 0x028B | | | | | | | | | |
| 0x028C | T2TMR | 7:0 | TxTMR[7:0] | | | | | | |
| 0x028D | T2PR | 7:0 | TxPR[7:0] | | | | | | |
| 0x028E | T2CON | 7:0 | ON | CKPS[2:0] | | | OUTPS[3:0] | | |
| 0x028F | T2HLT | 7:0 | PSYNC | CPOL | CSYNC | MODE[4:0] | | | |
| 0x0290 | T2CLKCON | 7:0 | | | | CS[3:0] | | | |
| 0x0291 | T2RST | 7:0 | | | | RSEL[3:0] | | | |
| 0x0292 | T4TMR | 7:0 | TxTMR[7:0] | | | | | | |
| 0x0293 | T4PR | 7:0 | TxPR[7:0] | | | | | | |
| 0x0294 | T4CON | 7:0 | ON | CKPS[2:0] | | | OUTPS[3:0] | | |
| 0x0295 | T4HLT | 7:0 | PSYNC | CPOL | CSYNC | MODE[4:0] | | | |
| 0x0296 | T4CLKCON | 7:0 | | | | CS[3:0] | | | |
| 0x0297 | T4RST | 7:0 | | | | RSEL[3:0] | | | |
| 0x0298 | T6TMR | 7:0 | TxTMR[7:0] | | | | | | |
| 0x0299 | T6PR | 7:0 | TxPR[7:0] | | | | | | |
| 0x029A | T6CON | 7:0 | ON | CKPS[2:0] | | | OUTPS[3:0] | | |
| 0x029B | T6HLT | 7:0 | PSYNC | CPOL | CSYNC | MODE[4:0] | | | |
| 0x029C | T6CLKCON | 7:0 | | | | CS[3:0] | | | |
| 0x029D | T6RST | 7:0 | | | | RSEL[3:0] | | | |

20.11 Register Definitions: Timer2 Control



Notice: References to module Timer2 apply to all the even numbered timers on this device. (Timer2, Timer4, etc.)

Related Links

[1.4.2.2 Long Bit Names](#)

20.11.1 TxTMR

Name: TxTMR
Address: 0x28C,0x292,0x298

Timer Counter Register

| | | | | | | | | |
|--------|------------|-----|-----|-----|-----|-----|-----|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TxTMR[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bits 7:0 – TxTMR[7:0] Timerx Counter bits

20.11.2 TxPR

Name: TxPR
Address: 0x28D,0x293,0x299

Timer Period Register

| | | | | | | | | |
|--------|-----------|-----|-----|-----|-----|-----|-----|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TxPR[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Bits 7:0 – TxPR[7:0] Timer Period Register bits

| Value | Description |
|--------|---|
| 1111 | The timer restarts at '0' when TxTMR reaches TxPR value |
| 1111 – | |
| 0000 | |
| 0000 | |

20.11.3 TxCON

Name: TxCON
Address: 0x28E,0x294,0x29A

Timerx Control Register

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|--------|-----------|-----|-----|------------|-----|-----|-----|
| | ON | CKPS[2:0] | | | OUTPS[3:0] | | | |
| Access | R/W/HC | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bit 7 – ON

Timer On bit⁽¹⁾

| Value | Description |
|-------|---|
| 1 | Timer is on |
| 0 | Timer is off: all counters and state machines are reset |

Bits 6:4 – CKPS[2:0] Timer Clock Prescale Select bits

| Value | Description |
|-------|-----------------|
| 111 | 1:128 Prescaler |
| 110 | 1:64 Prescaler |
| 101 | 1:32 Prescaler |
| 100 | 1:16 Prescaler |
| 011 | 1:8 Prescaler |
| 010 | 1:4 Prescaler |
| 001 | 1:2 Prescaler |
| 000 | 1:1 Prescaler |

Bits 3:0 – OUTPS[3:0] Timer Output Postscaler Select bits

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

Note:

1. In certain modes, the ON bit will be auto-cleared by hardware. See [Table 20-1](#).

20.11.4 TxHLT

Name: TxHLT
Address: 0x28F,0x295,0x29B

Timer Hardware Limit Control Register

| | | | | | | | | |
|--------|-------|------|-------|-----------|-----|-----|-----|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | PSYNC | CPOL | CSYNC | MODE[4:0] | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bit 7 – PSYNC

Timer Prescaler Synchronization Enable bit^(1, 2)

| Value | Description |
|-------|---|
| 1 | Timer Prescaler Output is synchronized to $F_{OSC}/4$ |
| 0 | Timer Prescaler Output is not synchronized to $F_{OSC}/4$ |

Bit 6 – CPOL

Timer Clock Polarity Selection bit⁽³⁾

| Value | Description |
|-------|--|
| 1 | Falling edge of input clock clocks timer/prescaler |
| 0 | Rising edge of input clock clocks timer/prescaler |

Bit 5 – CSYNC

Timer Clock Synchronization Enable bit^(4, 5)

| Value | Description |
|-------|---|
| 1 | ON bit is synchronized to timer clock input |
| 0 | ON bit is not synchronized to timer clock input |

Bits 4:0 – MODE[4:0]

Timer Control Mode Selection bits^(6, 7)

| Value | Description |
|----------------|--------------------------------|
| 00000 to 11111 | See Table 20-1 |

Note:

- Setting this bit ensures that reading TxTMR will return a valid data value.
- When this bit is '1', Timer cannot operate in Sleep mode.
- CKPOL should not be changed while ON = 1.
- Setting this bit ensures glitch-free operation when the ON is enabled or disabled.
- When this bit is set then the timer operation will be delayed by two input clocks after the ON bit is set.
- Unless otherwise indicated, all modes start upon ON = 1 and stop upon ON = 0 (stops occur without affecting the value of TxTMR).
- When TxTMR = TxPR, the next clock clears TxTMR, regardless of the operating mode.

20.11.5 TxCLKCON

Name: TxCLKCON
Address: 0x290,0x296,0x29C

Timer Clock Source Selection Register

| | | | | | | | | |
|--------|---|---|---|---|---------|-----|-----|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | | | | CS[3:0] | | | |
| Access | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | 0 | 0 | 0 | 0 |

Bits 3:0 – CS[3:0] Timer Clock Source Selection bits

Table 20-2. Clock Source Selection

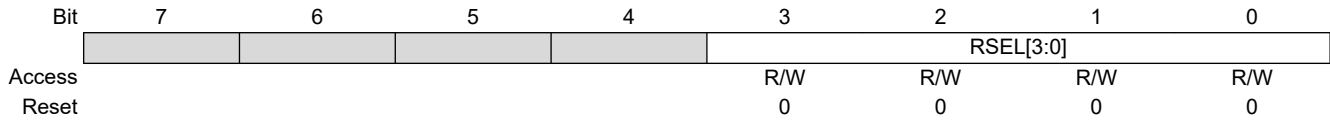
| CS[3:0] | Clock Source | | |
|---------|---------------------|---------------------|---------------------|
| | Timer2 | Timer4 | Timer6 |
| 1111 | Reserved | Reserved | Reserved |
| 1110 | CLC4_out | CLC4_out | CLC4_out |
| 1101 | CLC3_out | CLC3_out | CLC3_out |
| 1100 | CLC2_out | CLC2_out | CLC2_out |
| 1011 | CLC1_out | CLC1_out | CLC1_out |
| 1010 | ZCD1_output | ZCD1_output | ZCD1_output |
| 1001 | NCO1_out | NCO1_out | NCO1_out |
| 1000 | CLKR | CLKR | CLKR |
| 0111 | SOSC | SOSC | SOSC |
| 0110 | MFINTOSC(31.25 kHz) | MFINTOSC(31.25 kHz) | MFINTOSC(31.25 kHz) |
| 0101 | MFINTOSC(500 kHz) | MFINTOSC(500 kHz) | MFINTOSC(500 kHz) |
| 0100 | LFINTOSC | LFINTOSC | LFINTOSC |
| 0011 | HFINTOSC(32 MHz) | HFINTOSC(32 MHz) | HFINTOSC(32 MHz) |
| 0010 | F _{Osc} | F _{Osc} | F _{Osc} |
| 0001 | F _{Osc} /4 | F _{Osc} /4 | F _{Osc} /4 |
| 0000 | T2CKIPPS | T4CKIPPS | T6CKIPPS |

| Value | Description |
|-------|---|
| n | See Clock Source Selection table. |

20.11.6 TxRST

Name: TxRST
Address: 0x291,0x297,0x29D

Timer External Reset Signal Selection Register



Bits 3:0 – RSEL[3:0]

External Reset Source Selection Bits

Table 20-3. External Reset Sources

| RSEL[3:0] | Reset Source | | |
|-----------|--------------|-------------|-------------|
| | TMR2 | TMR4 | TMR6 |
| 1111 | Reserved | Reserved | Reserved |
| 1101 | CLC4_out | CLC4_out | CLC4_out |
| 1100 | CLC3_out | CLC3_out | CLC3_out |
| 1011 | CLC2_out | CLC2_out | CLC2_out |
| 1010 | CLC1_out | CLC1_out | CLC1_out |
| 1001 | ZCD1_output | ZCD1_output | ZCD1_output |
| 1000 | C2OUT_sync | C2OUT_sync | C2OUT_sync |
| 0111 | C1OUT_sync | C1OUT_sync | C1OUT_sync |
| 0110 | PWM7_out | PWM7_out | PWM7_out |
| 0101 | PWM6_out | PWM6_out | PWM6_out |
| 0100 | CCP4_out | CCP4_out | CCP4_out |
| 0011 | CCP3_out | CCP3_out | CCP3_out |
| 0010 | CCP2_out | CCP2_out | CCP2_out |
| 0001 | CCP1_out | CCP1_out | CCP1_out |
| 0000 | T2INPPS | T4INPPS | T6INPPS |

| Value | Description |
|-------|--|
| n | See External Reset Sources table |

21. SMT - Signal Measurement Timer

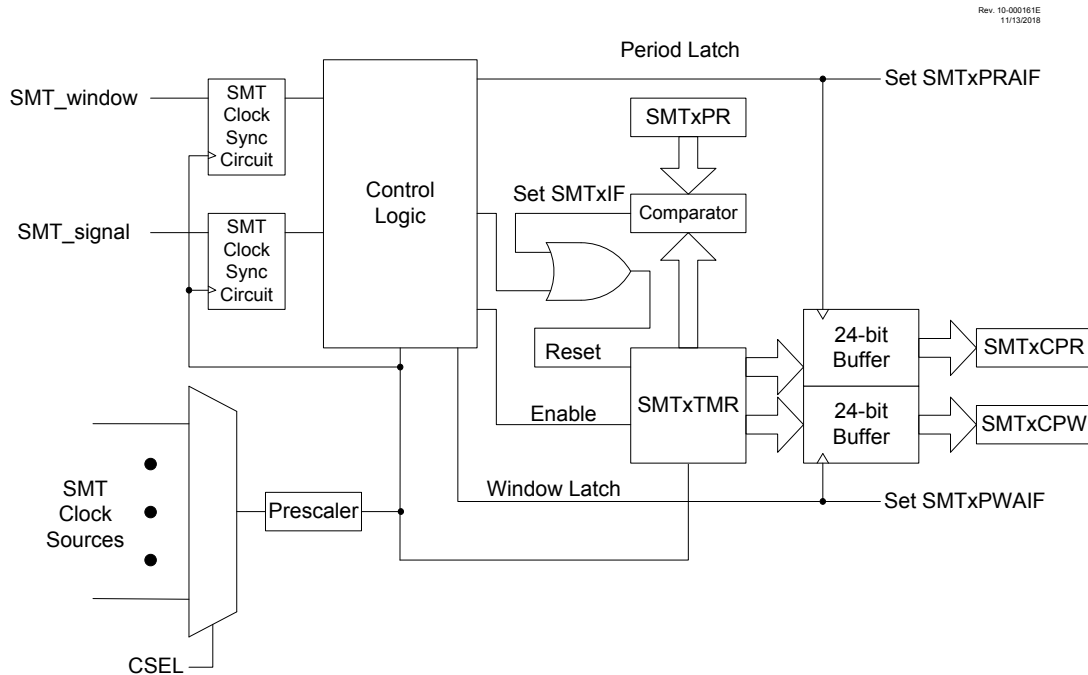
The SMT is a 24-bit counter with advanced clock and gating logic, which can be configured for measuring a variety of digital signal parameters such as pulse width, frequency and duty cycle, and the time difference between edges on two signals.

Features of the SMT include:

- 24-bit timer/counter
- Two 24-bit measurement capture registers
- One 24-bit period match register
- Multi-mode operation, including relative timing measurement
- Interrupt on period match and acquisition complete
- Multiple clock, signal and window sources

Below is the block diagram for the SMT module.

Figure 21-1. Signal Measurement Timer Block Diagram



21.1 SMT Operation

21.1.1 Clock Source Selection

The SMT clock source is selected by configuring the [CSEL](#) bits in the SMTxCLK register. The clock source can be prescaled using the [PS](#) bits of the SMTxCON0 register. The prescaled clock source is used to clock both the counter and any synchronization logic used by the module. Refer the table below for possible clock source options.

The polarity of the clock source can be selected using the [CPOL](#) bit in the SMTxCON0 register.

21.1.2 Signal and Window Source Selection

The SMT signal and window sources are selected by configuring the [SSEL](#) bits in the SMTxSIG register and the [WSEL](#) bits in the SMTxWIN register. Refer the tables below for the possible selections.

The polarity of the signal and window sources can be selected using the **SPOL** and **WPOL** bits in the SMTxCON0 register.

21.1.3 Time Base

The SMTxTMR is the 24-bit counter/timer used for measurement in each of the modes of the SMT. It can be reset to 0x000000 by setting the **RST** bit of the SMTxSTAT register. It can be written to and read by software. It is not guarded for atomic access, therefore reads and writes to the SMTxTMR should be made when the **GO** = 0.

The counter can be prevented from a rollover using the **STP** bit in the SMTxCON0 register. When **STP** = 1, SMTxTMR will remain equal to SMTxPR. When **STP** = 0, SMTxTMR resets to 0x000000.

21.1.4 Capture Pulse Width and Period Registers

The SMTxCPW and SMTxCPR registers are used to latch in the value of the SMTxTMR based on the mode of SMT operation. These registers can also be updated with the current value of the SMTxTMR value by setting the **CPWUP** and **CPRUP** bits of the SMTxSTAT register, respectively.

21.1.5 Status Information

The SMT provides input status information for the user without requiring the need to deal with the polarity of the incoming signals.

Go Status: Timer run status is determined by the **TS** bit of the SMTxSTAT register, and will be delayed in time by synchronizer delays in non-Counter modes.

Signal Status: Signal status is determined by the **AS** bit of the SMTxSTAT register. This bit is used in all modes except Window Measure, Time of Flight and Capture modes, and is only valid when **TS** = 1, and will be delayed in time by synchronizer delays in non-Counter modes.

Window Status: Window status is determined by the **WS** bit of the SMTxSTAT register. This bit is only used in Windowed Measure, Gated Counter and Gated Window Measure modes, and is only valid when **TS** = 1, and will be delayed in time by synchronizer delays in non-Counter modes.

21.1.6 Modes of Operation

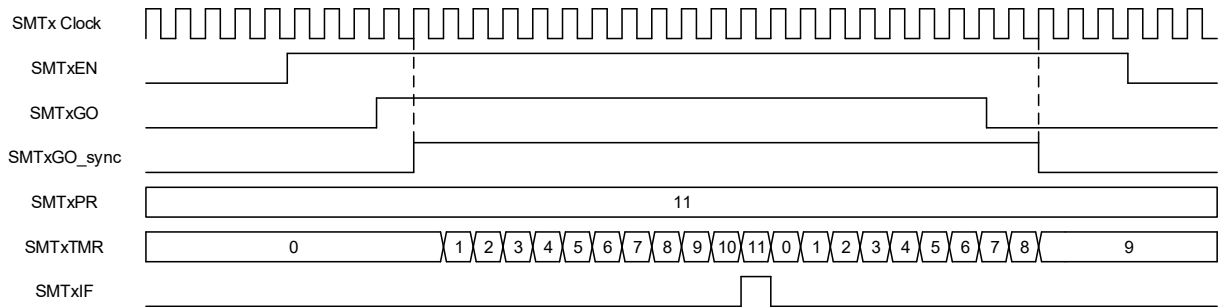
The mode of operation is selected by configuring **MODE** bits in SMTxCON1 register. The following sections provide descriptions and examples of how the modes can be used. Note that all waveforms assume **WPOL/SPOL/CPOL** = 0. For all modes, the **REPEAT** bit controls whether the acquisition is repeated or single. When **REPEAT** = 0 (Single Acquisition mode), the timer will stop incrementing and the SMTxGO bit will be reset upon the completion of an acquisition. Otherwise, the timer will continue and allow for continued acquisitions to overwrite the previous ones until the timer is stopped in software.

21.1.6.1 Timer Mode

Timer mode is the basic mode of operation where the SMTxTMR is used as a 24-bit timer. No data acquisition takes place in this mode. The timer increments as long as the SMTxGO bit has been set by software. No SMT window or SMT signal events affect the SMTxGO bit. Everything is synchronized to the SMT clock source. When the timer experiences a period match (SMTxTMR = SMTxPR), SMTxTMR is reset and the period match interrupt trips. See figure below.

Figure 21-2. Timer Mode Timing Diagram

Rev. 10-000 19A
12/19/2013



21.1.6.2 Gated Timer Mode

Gated Timer mode uses the signal input (**SSEL**) to control whether or not the SMTxTMR will increment. Upon a falling edge of the signal, the SMTxCPW register will update to the current value of the SMTxTMR. Example waveforms for both repeated and single acquisitions are provided in figures below.

Figure 21-3. Gated Timer Mode, Repeat Acquisition Timing Diagram

Rev. 10-000 19A
12/19/2013

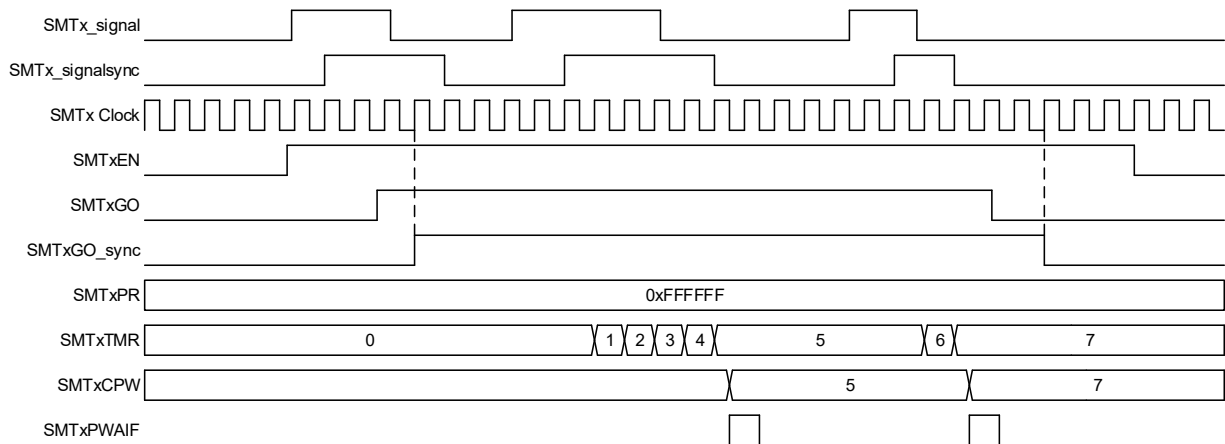
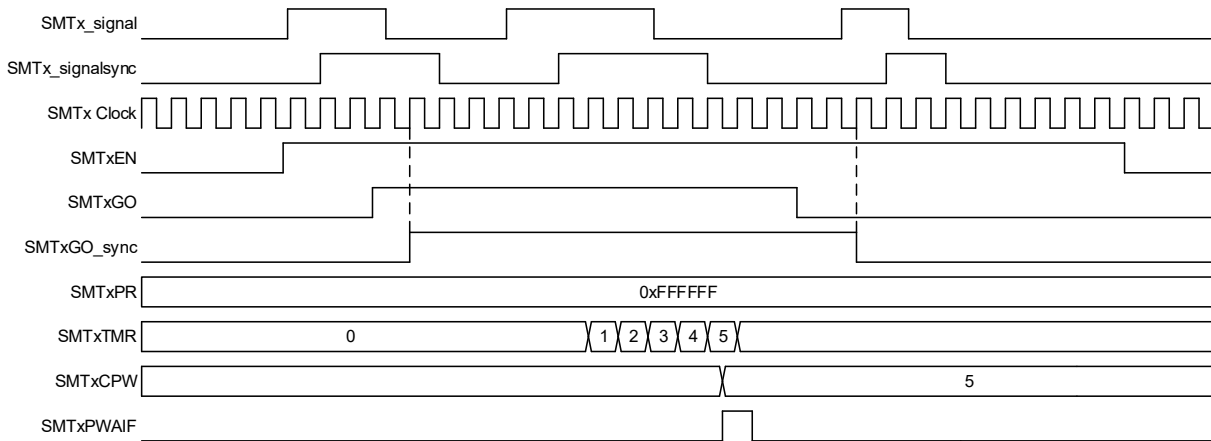


Figure 21-4. Gated Timer Mode, Single Acquisition Timing Diagram

Rev. 10-000-175A
12/19/2013



21.1.6.3 Period and Duty Cycle Measurement Mode

In this mode, either the duty cycle or period (depending on polarity) of the input signal can be acquired relative to the SMT clock. The CPW register is updated on a falling edge of the signal, and the CPR register is updated on a rising edge of the signal, along with the SMTxTMR resetting to 0x000001. In addition, the SMTxGO bit is reset on a rising edge when the SMT is in single acquisition mode. See figures below.

Figure 21-5. Period and Duty-Cycle, Repeat Acquisition Mode Timing Diagram

Rev. 10-000-177A
12/19/2013

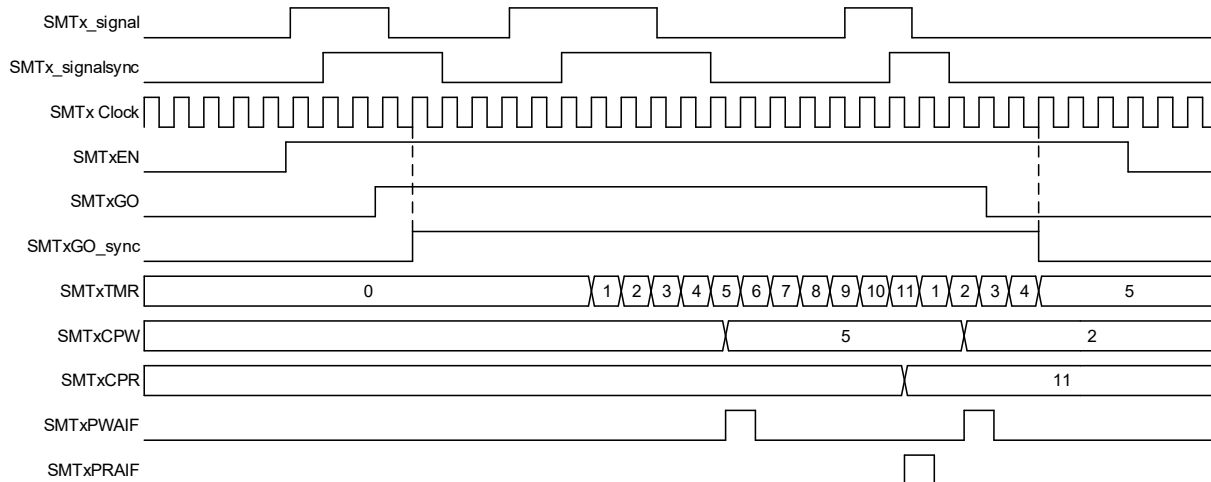
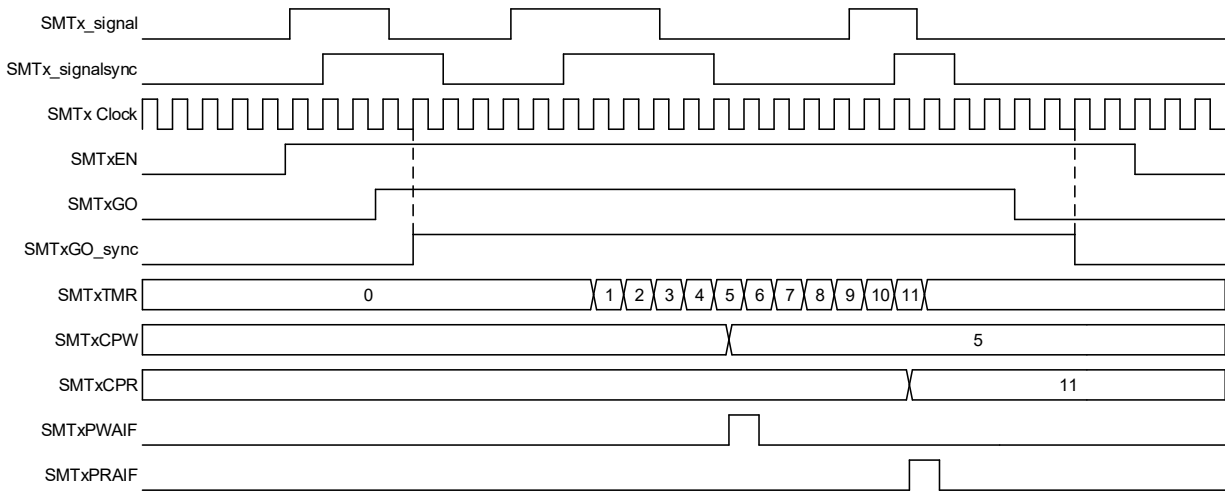


Figure 21-6. Period and Duty-Cycle, Single Acquisition Mode Timing Diagram

Rev. 10,000 18A
10/19/2013



21.1.6.4 High and Low Measurement Mode

This mode measures the high and low pulse time of the signal relative to the SMT clock. It begins incrementing the SMTxTMR on a rising edge on the input signal, then updates the SMTxCPW register with the value and resets the SMTxTMR on a falling edge, starting to increment again. Upon observing another rising edge, it updates the SMTxCPR register with its current value and once again resets the SMTxTMR value and begins incrementing again. See the figures below.

Figure 21-7. High and Low Measurement Mode, Repeat Acquisition Timing Diagram

Rev. 10,000 18A
10/19/2013

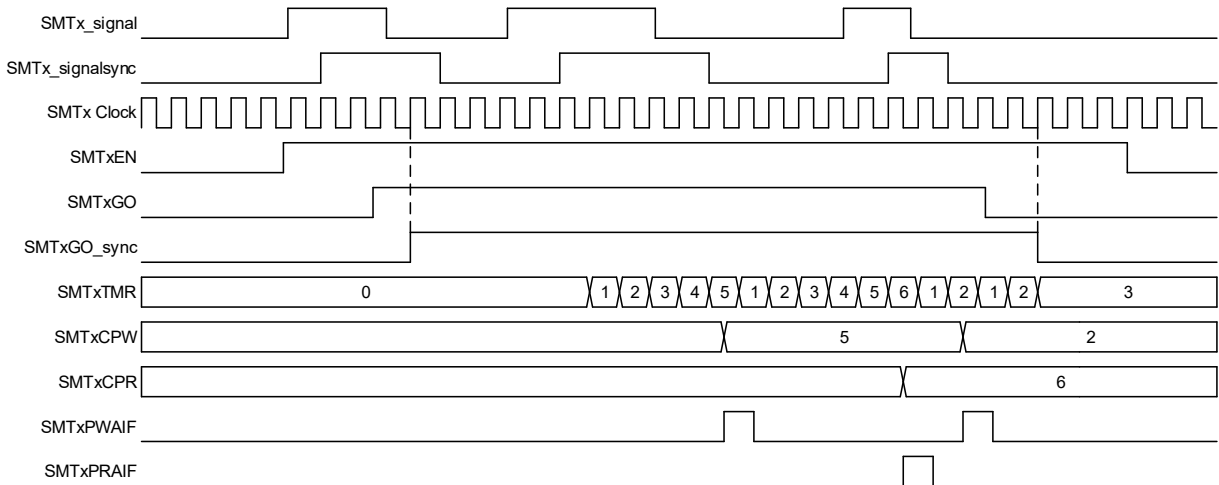
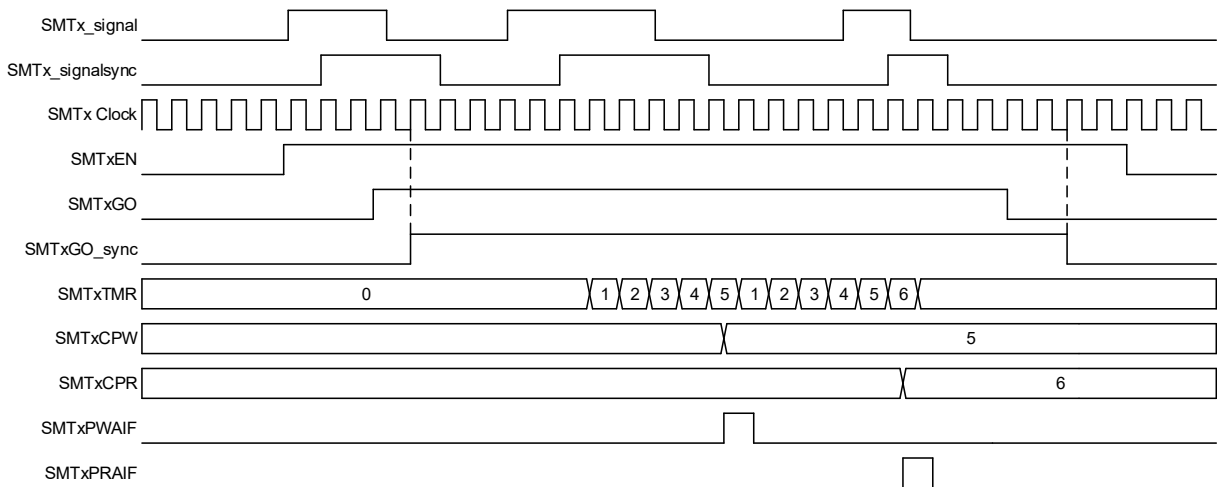


Figure 21-8. High and Low Measurement Mode, Single Acquisition Timing Diagram

Rev. 10-000 179A
12/19/2013



21.1.6.5 Windowed Measurement Mode

This mode measures the duration of the window input (**WSEL**) to the SMT. It begins incrementing the timer on a rising edge of the window input and updates the **SMTxCPR** register with the value of the timer and resets the timer on a second rising edge. See figures below.

Figure 21-9. Windowed Measurement Mode, Repeat Acquisition Timing Diagram

Rev. 10-000 185A
12/19/2013

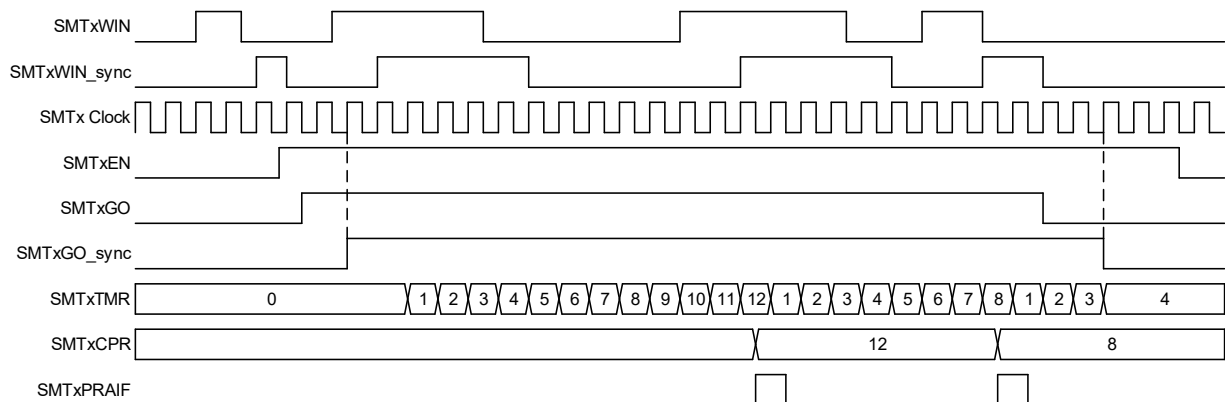
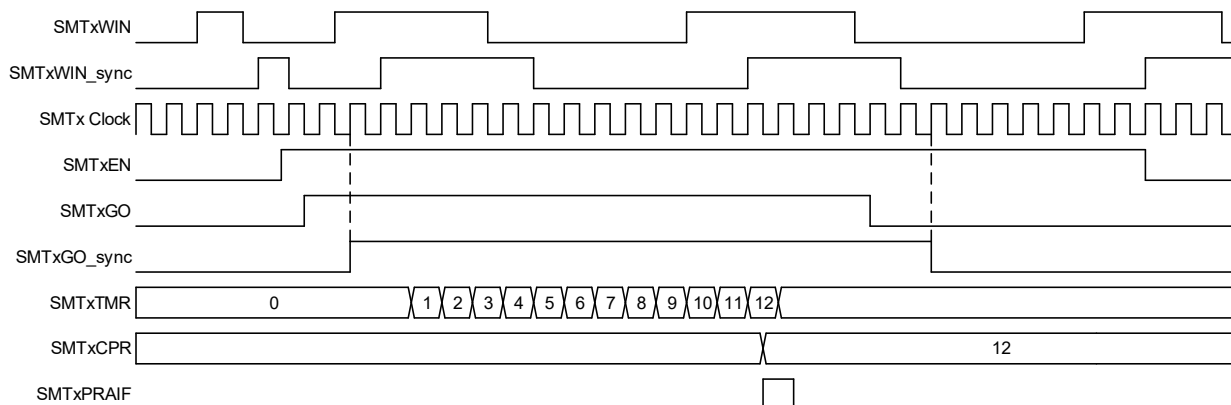


Figure 21-10. Windowed Measurement Mode, Single Acquisition Timing Diagram

Rev. 10-000-181A
12/18/2013



21.1.6.6 Gated Window Measurement Mode

This mode measures the duty cycle of the signal input over a known input window. It does so by incrementing the timer on each pulse of the clock signal while the signal input is high, updating the SMTxCPR register and resetting the timer on every rising edge of the window input after the first. See figures below.

Figure 21-11. Gated Windowed Measurement Mode, Repeat Acquisition Timing Diagram

Rev. 10-000-181A
12/18/2013

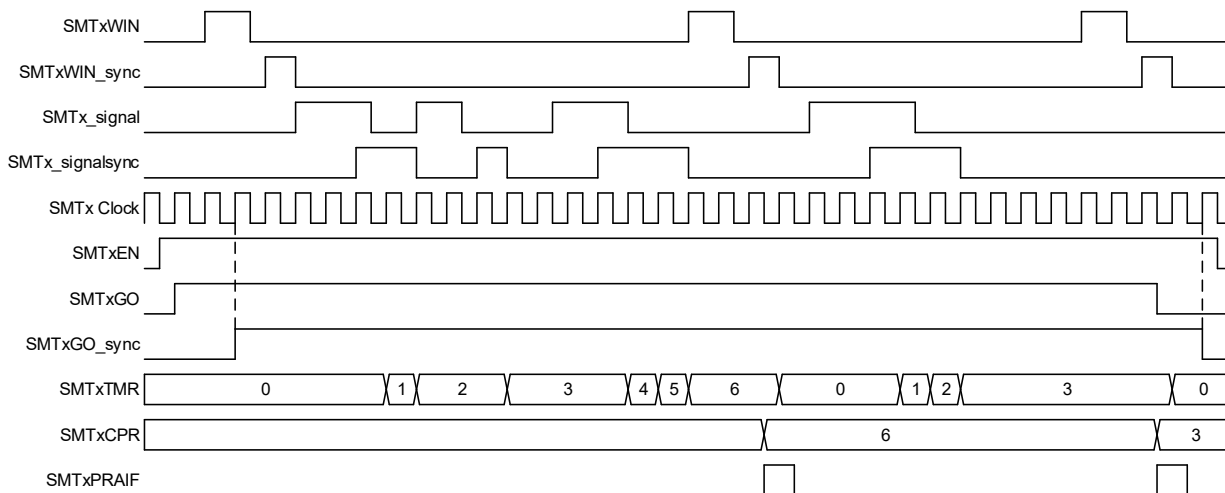
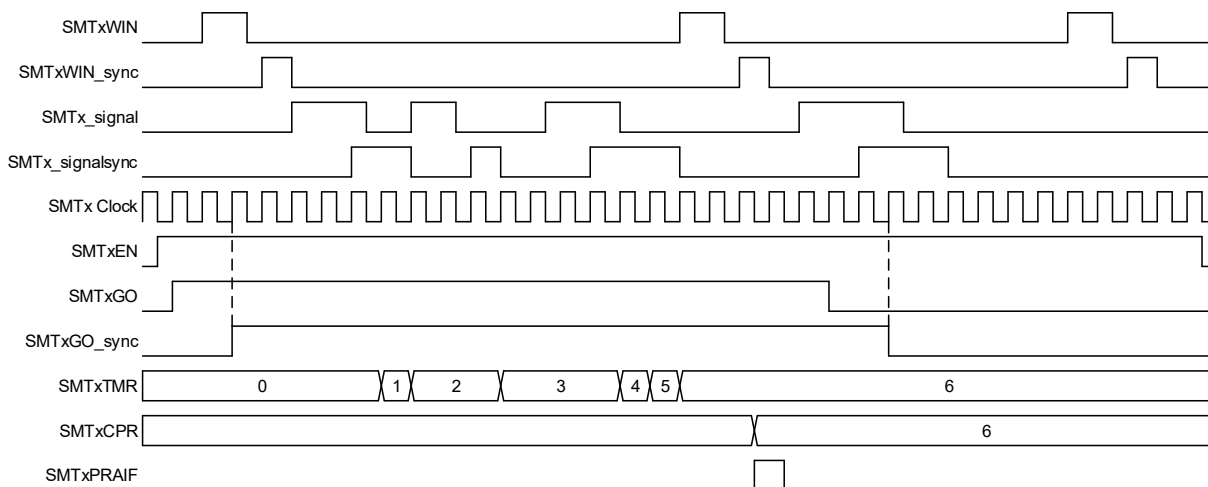


Figure 21-12. Gated Windowed Measurement Mode, Single Acquisition Timing Diagram

Rev. 10-000183A
12/19/2013



21.1.6.7 Time of Flight Measurement Mode

This mode measures the time interval between a rising edge on the window input and a rising edge on the signal input, beginning to increment the timer upon observing a rising edge on the window input, while updating the SMTxCPR register and resetting the timer upon observing a rising edge on the signal input. In the event of two rising edges of the window signal without a signal rising edge, it will update the SMTxCPW register with the current value of the timer and reset the timer value. See figures below.

Figure 21-13. Time of Flight Mode, Repeat Acquisition Timing Diagram

Rev. 10-000186A
4/22/2016

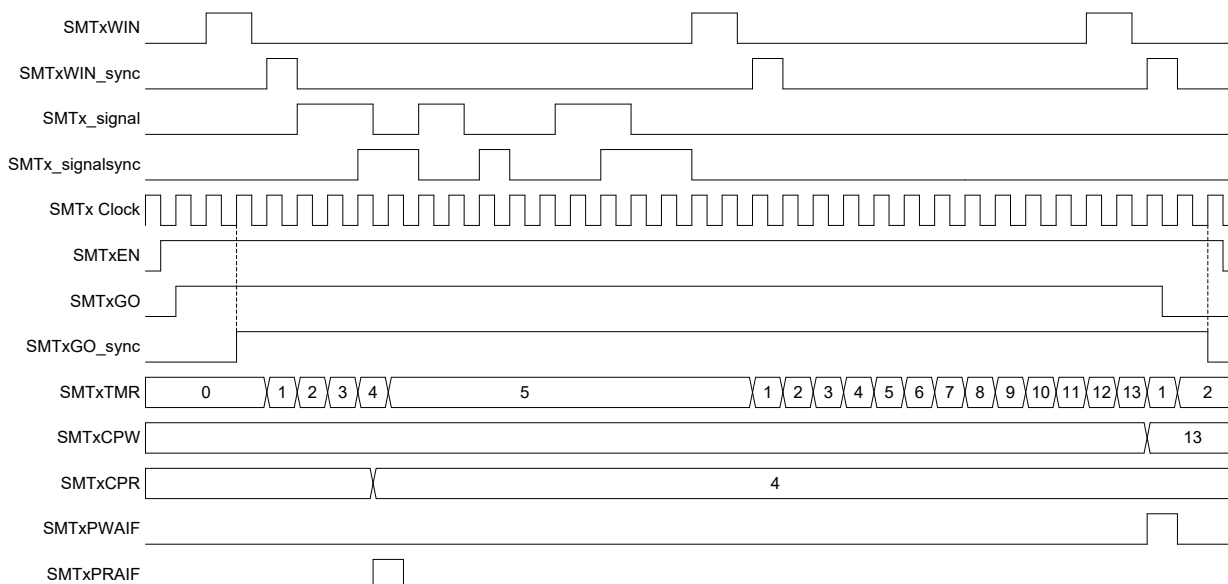
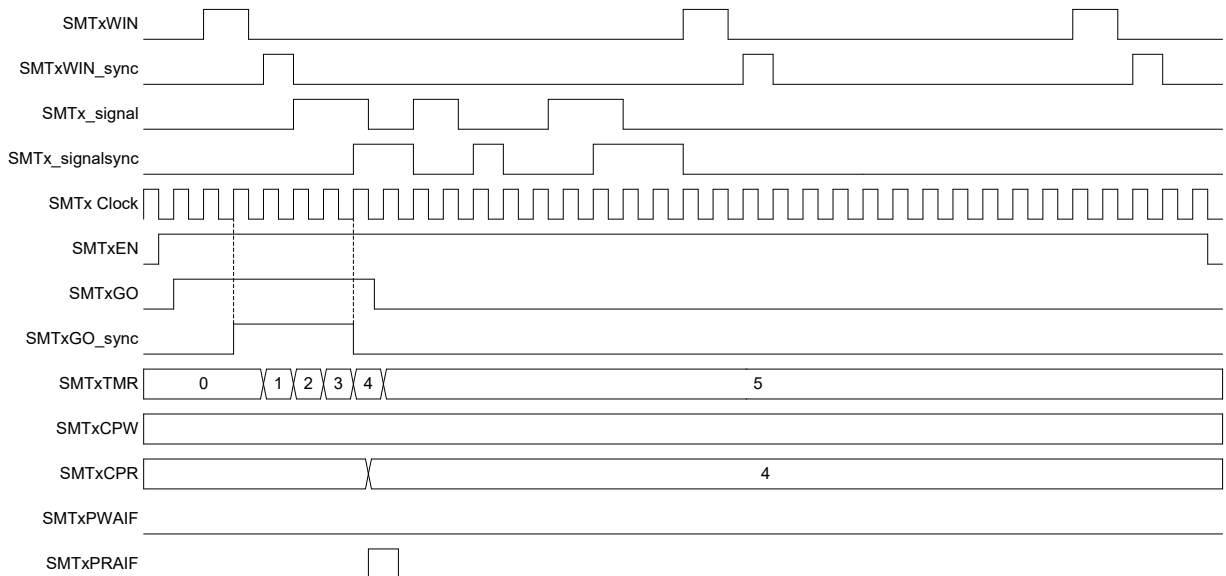


Figure 21-14. Time of Flight Mode, Single Acquisition Timing Diagram

Rev. 10-000185A
4/20/2014



21.1.6.8 Capture Mode

This mode captures the timer value based on a rising or falling edge on the window input and triggers an interrupt. This mimics the capture feature of a CCP module. The timer begins incrementing upon the SMTxGO bit being set, and updates the value of the SMTxCPR register on each rising edge of window signal, and updates the value of the SMTxCPW register on each falling edge of the window signal. The timer is not reset by any hardware conditions in this mode and must be reset by software, if desired. See figures below.

Figure 21-15. Capture Mode, Repeat Acquisition Timing Diagram

Rev. 10-000185A
10/16/2013

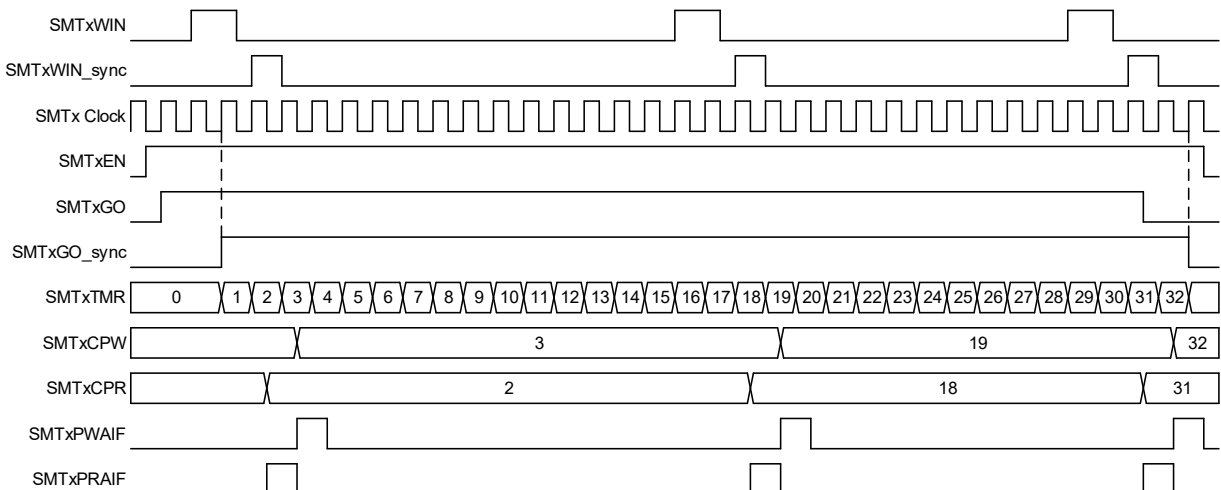
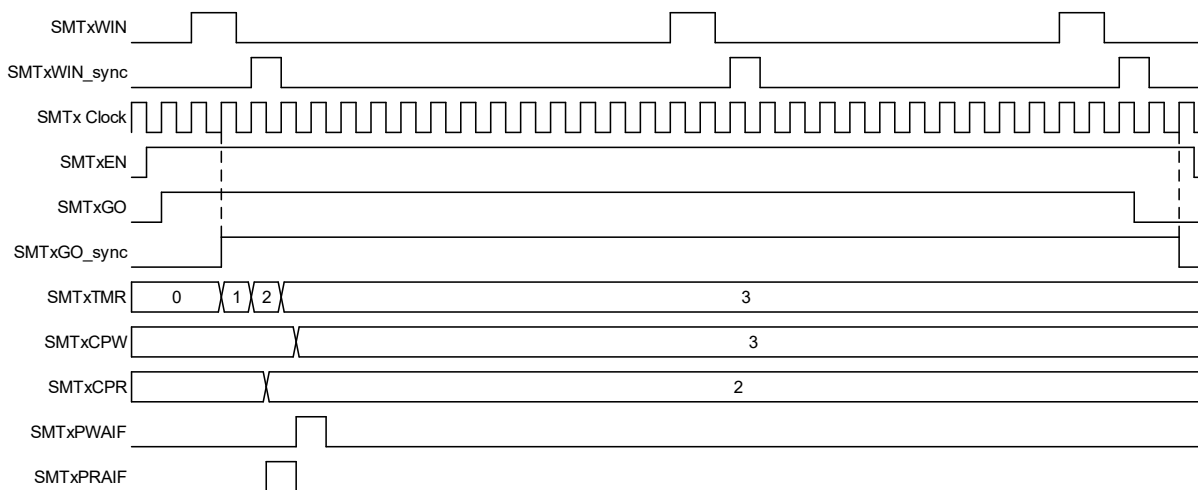


Figure 21-16. Capture Mode, Single Acquisition Timing Diagram

Rev. 10-000187A
10/16/2013

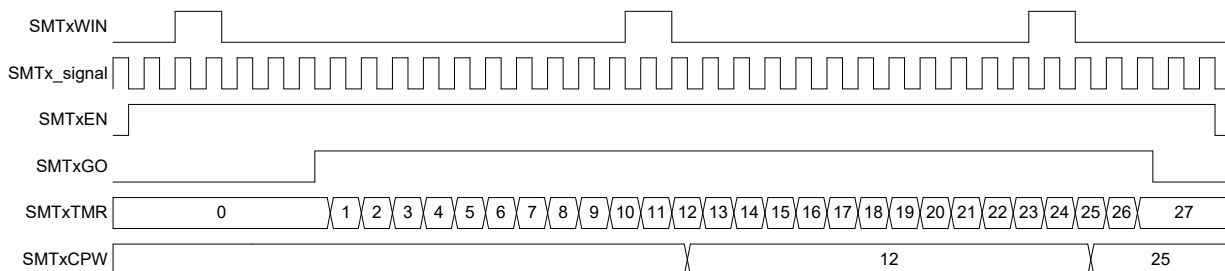


21.1.6.9 Counter Mode

This mode increments the timer on each pulse of the signal input. This mode is asynchronous to the SMT clock and uses the signal input as a time source. The SMTxCPW register will be updated with the current SMTxTMR value on the falling edge of the window input. See figure below.

Figure 21-17. Counter Mode Timing Diagram

Rev. 10-000188A
4/12/2016



21.1.6.10 Gated Counter Mode

This mode counts pulses on the signal input, gated by the window input. It begins incrementing the timer upon seeing a rising edge of the window input and updates the SMTxCPW register upon a falling edge on the window input. See figures below.

Figure 21-18. Gated Counter Mode, Repeat Acquisition Timing Diagram

Rev. 10.000190A
12/18/2013

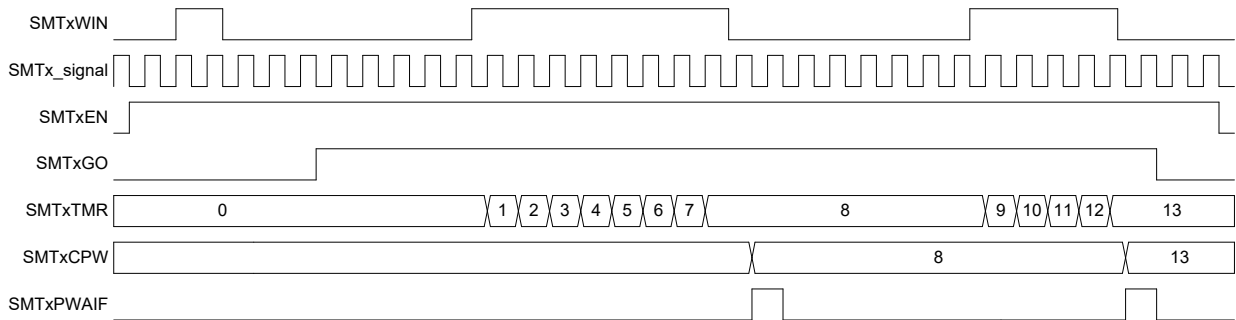
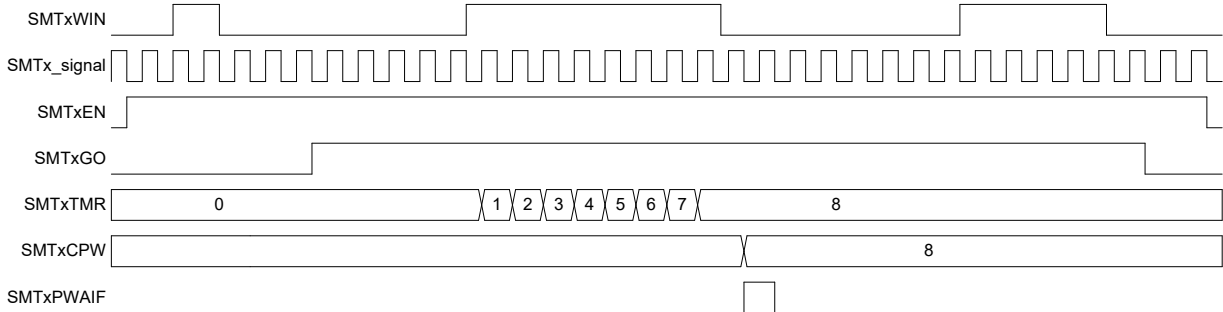


Figure 21-19. Gated Counter Mode, Single Acquisition Timing Diagram

Rev. 10.000191A
12/18/2013



21.1.6.11 Windowed Counter Mode

This mode counts pulses on the signal input, within a window dictated by the window input. It begins counting upon seeing a rising edge of the window input, updates the SMTxCPW register on a falling edge of the window input, and updates the SMTxCPR register on each rising edge of the window input after the first. See figures below.

Figure 21-20. Windowed Counter Mode, Repeat Acquisition Timing Diagram

Rev. 10.000192A
10/18/2013

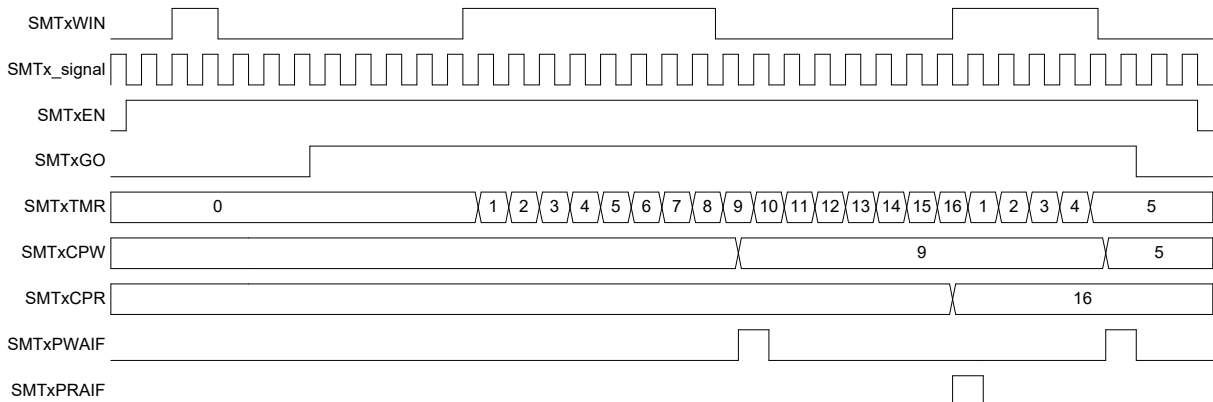
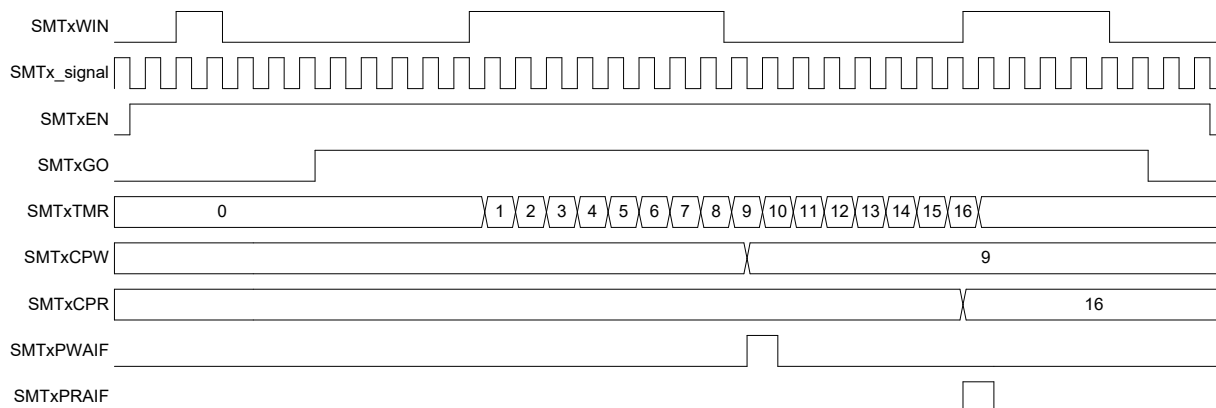


Figure 21-21. Windowed Counter Mode, Single Acquisition Timing Diagram

Rev. 10-000193A
12/16/2013



21.1.7 Interrupts

The SMT has three interrupts:

- **Pulse Width Acquisition Interrupt (SMTxPWAIF):** Interrupt triggers when SMTxCPW is updated
- **Period Acquisition Interrupt (SMTxPRAIF):** Interrupt triggers when SMTxCPR is updated
- **Counter Period Match Interrupt (SMTxIF):** Interrupt triggers when SMTxTMR equals SMTxPR

Each of the above interrupts can be enabled/disabled using the corresponding bits in the PEx register.

21.1.8 Operation During Sleep

The SMT can operate during Sleep, Idle, and Doze modes; provided that the clock and signal sources continue to function. System clock sources, like F_{OSC} and $F_{OSC}/4$, are disabled in Sleep.

21.2 Register Summary - SMT Control

| Address | Name | Bit Pos. | | | | | | | | |
|-------------|----------|----------|------------|--------|-----|------|------|-----------|---------|----|
| 0x00 ... | Reserved | | | | | | | | | |
| 0x048B | | | | | | | | | | |
| 0x048C | SMT1TMR | 7:0 | TMR[7:0] | | | | | | | |
| | | 15:8 | TMR[15:8] | | | | | | | |
| | | 23:16 | TMR[23:16] | | | | | | | |
| 0x048F | SMT1CPR | 7:0 | CPR[7:0] | | | | | | | |
| | | 15:8 | CPR[15:8] | | | | | | | |
| | | 23:16 | CPR[23:16] | | | | | | | |
| 0x0492 | SMT1CPW | 7:0 | CPW[7:0] | | | | | | | |
| | | 15:8 | CPW[15:8] | | | | | | | |
| | | 23:16 | CPW[23:16] | | | | | | | |
| 0x0495 | SMT1PR | 7:0 | PR[7:0] | | | | | | | |
| | | 15:8 | PR[15:8] | | | | | | | |
| | | 23:16 | PR[23:16] | | | | | | | |
| 0x0498 | SMT1CON0 | 7:0 | EN | | STP | WPOL | SPOL | CPOL | PS[1:0] | |
| 0x0499 | SMT1CON1 | 7:0 | GO | REPEAT | | | | MODE[3:0] | | |
| 0x049A | SMT1STAT | 7:0 | CPRUP | CPWUP | | RST | | TS | WS | AS |
| 0x049B | SMT1CLK | 7:0 | | | | | | CSEL[2:0] | | |
| 0x049C | SMT1SIG | 7:0 | | | | | | SSEL[4:0] | | |
| 0x049D | SMT1WIN | 7:0 | | | | | | WSEL[4:0] | | |

21.3 Register Definitions: SMT Control

21.3.1 SMTxTMR

Name: SMTxTMR
Address: 0x048C

SMT Timer Register

| | | | | | | | | | |
|--------|-----|------------|-----|-----|-----|-----|-----|-----|-----|
| | Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | | TMR[23:16] | | | | | | | |
| Access | | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | | | | | | | | |
| | Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| | | TMR[15:8] | | | | | | | |
| Access | | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | | | | | | | | |
| | Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | TMR[7:0] | | | | | | | |
| Access | | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bits 23:0 – TMR[23:0] SMT timer value

Note: The individual bytes in this multi-byte register can be accessed with the following register names:

- SMTxTMRU: Accesses the upper byte TMR[23:16]
- SMTxTMRH: Accesses the high byte TMR[15:8]
- SMTxTMRL: Accesses the low byte TMR[7:0]

21.3.3 SMTxCPW

Name: SMTxCPW
Address: 0x0492

SMT Captured Pulse Width Register

| | | | | | | | | | |
|--------|-----|------------|----|----|----|----|----|----|----|
| | Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | | CPW[23:16] | | | | | | | |
| Access | | R | R | R | R | R | R | R | R |
| Reset | | x | x | x | x | x | x | x | x |
| | | | | | | | | | |
| | Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| | | CPW[15:8] | | | | | | | |
| Access | | R | R | R | R | R | R | R | R |
| Reset | | x | x | x | x | x | x | x | x |
| | | | | | | | | | |
| | Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | CPW[7:0] | | | | | | | |
| Access | | R | R | R | R | R | R | R | R |
| Reset | | x | x | x | x | x | x | x | x |

Bits 23:0 – CPW[23:0] SMTxTMR value at time of capture event
 Reset States: POR/BOR = xxxxxxxxxxxxxxxxxxxxxxxxx
 All Other Resets = uuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuu

Note: The individual bytes in this multi-byte register can be accessed with the following register names:

- SMTxCPWU: Accesses the upper byte CPW[23:16]
- SMTxCPWH: Accesses the high byte CPW[15:8]
- SMTxCPWL: Accesses the low byte CPW[7:0]

21.3.4 SMTxPR

Name: SMTxPR
Address: 0x0495

SMT Period Register

| | | | | | | | | |
|--------|-----------|-----|-----|-----|-----|-----|-----|-----|
| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | PR[23:16] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| | PR[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | PR[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Bits 23:0 – PR[23:0] The SMTxTMR value at which the SMTxTMR resets to zero

Note: The individual bytes in this multi-byte register can be accessed with the following register names:

- SMTxPRU: Accesses the upper byte PR[23:16]
- SMTxPRH: Accesses the high byte PR[15:8]
- SMTxPRL: Accesses the low byte PR[7:0]

21.3.5 SMTxCON0

Name: SMTxCON0
Address: 0x0498

SMT Control Register 0

| | | | | | | | | |
|--------|-----|---|-----|------|------|------|---------|-----|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | EN | | STP | WPOL | SPOL | CPOL | PS[1:0] | |
| Access | R/W | | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | | 0 | 0 | 0 | 0 | 0 | 0 |

Bit 7 – EN SMT Enable bit

| Value | Description |
|-------|---|
| 1 | SMT is enabled |
| 0 | SMT is disabled; internal states are reset, clock requests are disabled |

Bit 5 – STP SMT Counter Halt Enable bit

| Value | Condition | Description |
|-------|-----------------------|--|
| 1 | When SMTxTMR = SMTxPR | Counter remains SMTxPR; period match interrupt occurs when clocked |
| 0 | When SMTxTMR = SMTxPR | Counter resets to 0x000000; period match interrupt occurs when clocked |

Bit 4 – WPOL SMTxWIN Input Polarity Control bit

| Value | Description |
|-------|--|
| 1 | Window signal is active-low/falling edge enabled |
| 0 | Window signal is active-high/rising edge enabled |

Bit 3 – SPOL SMTxSIG Input Polarity Control bit

| Value | Description |
|-------|---|
| 1 | SMT Signal is active-low/falling edge enabled |
| 0 | SMT Signal is active-high/rising edge enabled |

Bit 2 – CPOL SMT Clock Input Polarity Control bit

| Value | Description |
|-------|---|
| 1 | SMTxTMR increments on the falling edge of the selected clock signal |
| 0 | SMTxTMR increments on the rising edge of the selected clock signal |

Bits 1:0 – PS[1:0] SMT Prescale Select bits

| Value | Description |
|-------|-----------------|
| 11 | Prescaler = 1:8 |
| 10 | Prescaler = 1:4 |
| 01 | Prescaler = 1:2 |
| 00 | Prescaler = 1:1 |

21.3.6 SMTxCON1

Name: SMTxCON1
Address: 0x0499

SMT Control Register 1

| | | | | | | | | |
|--------|-----|--------|---|---|-----------|-----|-----|-----|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | GO | REPEAT | | | MODE[3:0] | | | |
| Access | R/W | R/W | | | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | | | 0 | 0 | 0 | 0 |

Bit 7 – GO SMT GO Data Acquisition bit

| Value | Description |
|-------|--|
| 1 | Incrementing, acquiring data is enabled |
| 0 | Incrementing, acquiring data is disabled |

Bit 6 – REPEAT SMT Repeat Acquisition Enable bit

| Value | Description |
|-------|---|
| 1 | Repeat Data Acquisition mode is enabled |
| 0 | Single Acquisition mode is enabled |

Bits 3:0 – MODE[3:0] SMT Operation Mode Select bits

| Value | Description |
|-------|-----------------------------------|
| 1111 | Reserved |
| 1110 | Reserved |
| 1101 | Reserved |
| 1100 | Reserved |
| 1011 | Reserved |
| 1010 | Windowed counter |
| 1001 | Gated counter |
| 1000 | Counter |
| 0111 | Capture |
| 0110 | Time of flight |
| 0101 | Gated windowed measurement |
| 0100 | Windowed measurement |
| 0011 | High and low time measurement |
| 0010 | Period and Duty-Cycle Acquisition |
| 0001 | Gated Timer |
| 0000 | Timer |

21.3.7 SMTxSTAT

Name: SMTxSTAT
Address: 0x049A

SMT Status Register

| | | | | | | | | |
|--------|--------|--------|---|-----|---|----|----|----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | CPRUP | CPWUP | | RST | | TS | WS | AS |
| Access | R/W/HC | R/W/HC | | R/W | | RO | RO | RO |
| Reset | 0 | 0 | | 0 | | 0 | 0 | 0 |

Bit 7 – CPRUP SMT Manual Period Buffer Update bit

| Value | Description |
|-------|--------------------------------------|
| 1 | Request update to SMTxCPR registers |
| 0 | SMTxCPR registers update is complete |

Bit 6 – CPWUP SMT Manual Pulse Width Buffer Update bit

| Value | Description |
|-------|--------------------------------------|
| 1 | Request update to SMTxCPW registers |
| 0 | SMTxCPW registers update is complete |

Bit 4 – RST SMT Manual Timer Reset bit

| Value | Description |
|-------|--------------------------------------|
| 1 | Request Reset to SMTxTMR registers |
| 0 | SMTxTMR registers update is complete |

Bit 2 – TS SMT GO Value Status bit

| Value | Description |
|-------|-----------------------------|
| 1 | SMTxTMR is incrementing |
| 0 | SMTxTMR is not incrementing |

Bit 1 – WS SMT Window Status bit

| Value | Description |
|-------|----------------------|
| 1 | SMT window is open |
| 0 | SMT window is closed |

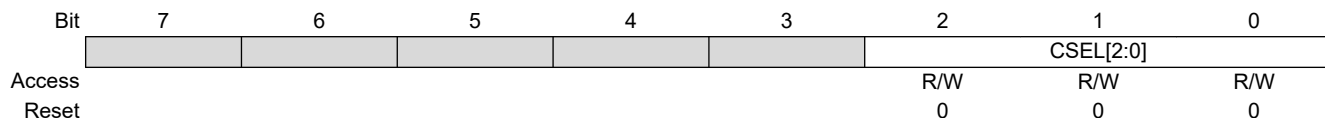
Bit 0 – AS SMT Signal Value Status bit

| Value | Description |
|-------|------------------------------------|
| 1 | SMT acquisition is in progress |
| 0 | SMT acquisition is not in progress |

21.3.8 SMTxCLK

Name: SMTxCLK
Address: 0x049B

SMT Clock Selection Register



Bits 2:0 – CSEL[2:0] SMT Clock Selection bits

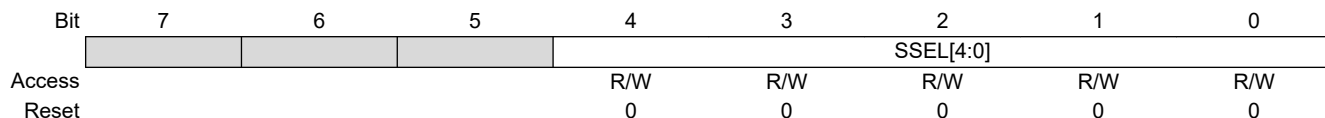
Table 21-1. SMT Clock Source Selection

| CSEL[2:0] | Clock Source |
|-----------|---------------------|
| 111 | CLKREF output |
| 110 | SOSC |
| 101 | MFINTOSC (31.25kHz) |
| 100 | MFINTOSC (500kHz) |
| 011 | LFINTOSC |
| 010 | HFINTOSC |
| 001 | F _{osc} |
| 000 | F _{osc} /4 |

21.3.9 SMTxSIG

Name: SMTxSIG
Address: 0x049C

SMT Signal Selection bits



Bits 4:0 – SSEL[4:0] SMT Signal Selection bits

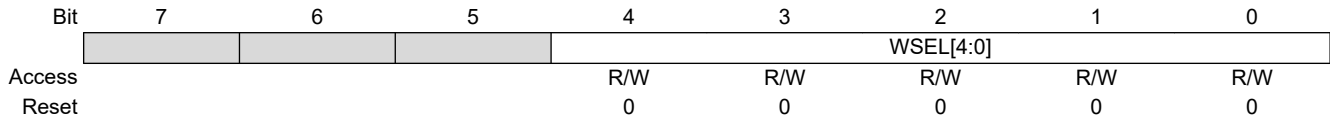
Table 21-2. SMT Signal Selection

| SSEL[4:0] | SMT1 Signal Source |
|-------------|----------------------------|
| 11111-10110 | Reserved |
| 10101 | CLC4OUT |
| 10100 | CLC3OUT |
| 10011 | CLC2OUT |
| 10010 | CLC1OUT |
| 10001 | ZCDOUT |
| 10000 | C2OUT |
| 01111 | C1OUT |
| 01110 | NCO1OUT |
| 01101 | PWM7OUT |
| 01100 | PWM6OUT |
| 01011 | CCP4OUT |
| 01010 | CCP3OUT |
| 01001 | CCP2OUT |
| 01000 | CCP1OUT |
| 00111 | TMR6 postscaled output |
| 00110 | TMR5 overflow |
| 00101 | TMR4 postscaled output |
| 00100 | TMR3 overflow |
| 00011 | TMR2 postscaled output |
| 00010 | TMR1 overflow |
| 00001 | TMR0 overflow |
| 00000 | Pin Selected by SMT1SIGPPS |

21.3.10 SMTxWIN

Name: SMTxWIN
Address: 0x049D

SMT Window Input Select Register



Bits 4:0 – WSEL[4:0] SMT Window Selection bits

Table 21-3. SMT Window Selection

| WSEL[4:0] | SMT1 Window Source |
|-------------|----------------------------|
| 11111-11000 | Reserved |
| 10111 | NCO1OUT |
| 10110 | Reserved |
| 10101 | CLKREFOUT |
| 10100 | CLC4OUT |
| 10011 | CLC3OUT |
| 10010 | CLC2OUT |
| 10001 | CLC1OUT |
| 10000 | ZCDOUT |
| 01111 | C2OUT |
| 01110 | C1OUT |
| 01101 | PWM7OUT |
| 01100 | PWM6OUT |
| 01011 | CCP4OUT |
| 01010 | CCP3OUT |
| 01001 | CCP2OUT |
| 01000 | CCP1OUT |
| 00111 | TMR6_postscaled_out |
| 00110 | TMR4_postscaled_out |
| 00101 | TMR2_postscaled_out |
| 00100 | TMR0_overflow |
| 00011 | SOSC |
| 00010 | MFINTOSC (31.25kHz) |
| 00001 | LFINTOSC (31.25kHz) |
| 00000 | Pin Selected by SMT1WINPPS |

22. Capture/Compare/PWM Module

The Capture/Compare/PWM module is a peripheral that allows the user to time and control different events, and to generate Pulse-Width Modulation (PWM) signals. In Capture mode, the peripheral allows the timing of the duration of an event. The Compare mode allows the user to trigger an external event when a predetermined amount of time has expired. The PWM mode can generate Pulse-Width Modulated signals of varying frequency and duty cycle.

This family of devices contains four standard Capture/Compare/PWM modules (CCP1, CCP2, CCP3, and CCP4). It should be noted that the Capture/Compare mode operation is described with respect to TMR1 and the PWM mode operation is described with respect to T2TMR in the following sections.

The Capture and Compare functions are identical for all CCP modules.



Important:

1. In devices with more than one CCP module, it is very important to pay close attention to the register names used. A number placed after the module acronym is used to distinguish between separate modules. For example, the CCP1CON and CCP2CON control the same operational aspects of two completely different CCP modules.
2. Throughout this section, generic references to a CCP module in any of its operating modes may be interpreted as being equally applicable to CCPx module. Register names, module signals, I/O pins, and bit names may use the generic designator 'x' to indicate the use of a numeral to distinguish a particular module, when required.

22.1 CCP Module Configuration

Each Capture/Compare/PWM module is associated with a control register ([CCPxCON](#)), a capture input selection register ([CCPxCAP](#)) and a data register ([CCPRx](#)). The data register, in turn, is comprised of two 8-bit registers: CCPRxL (low byte) and CCPRxH (high byte).

22.1.1 CCP Modules and Timer Resources

The CCP modules utilize Timers 1 through 6 that vary with the selected mode. Various timers are available to the CCP modules in Capture, Compare or PWM modes, as shown in the table below.

Table 22-1. CCP Mode - Timer Resources

| CCP Mode | Timer Resource |
|----------|--------------------------|
| Capture | Timer1, Timer3 or Timer5 |
| Compare | |
| PWM | Timer2, Timer4 or Timer6 |

The assignment of a particular timer to a module is determined by the timer to CCP enable bits in the [CCPTMRS0](#) and/or [CCPTMRS1](#) registers. All of the modules may be active at once and may share the same timer resource if they are configured to operate in the same mode (Capture/Compare or PWM) at the same time.

22.1.2 Open-Drain Output Option

When operating in Output mode (the Compare or PWM modes), the drivers for the CCPx pins can be optionally configured as open-drain outputs. This feature allows the voltage level on the pin to be pulled to a higher level through an external pull-up resistor and allows the output to communicate with external circuits without the need for additional level shifters.

22.2 Capture Mode

Capture mode makes use of the 16-bit odd numbered timer resources (Timer1, Timer3, etc.). When an event occurs on the capture source, the 16-bit CCPRx register captures and stores the 16-bit value of the TMRx register. An event is defined as one of the following and is configured by the **MODE** bits:

- Every falling edge of CCPx input
- Every rising edge of CCPx input
- Every 4th rising edge of CCPx input
- Every 16th rising edge of CCPx input
- Every edge of CCPx input (rising or falling)

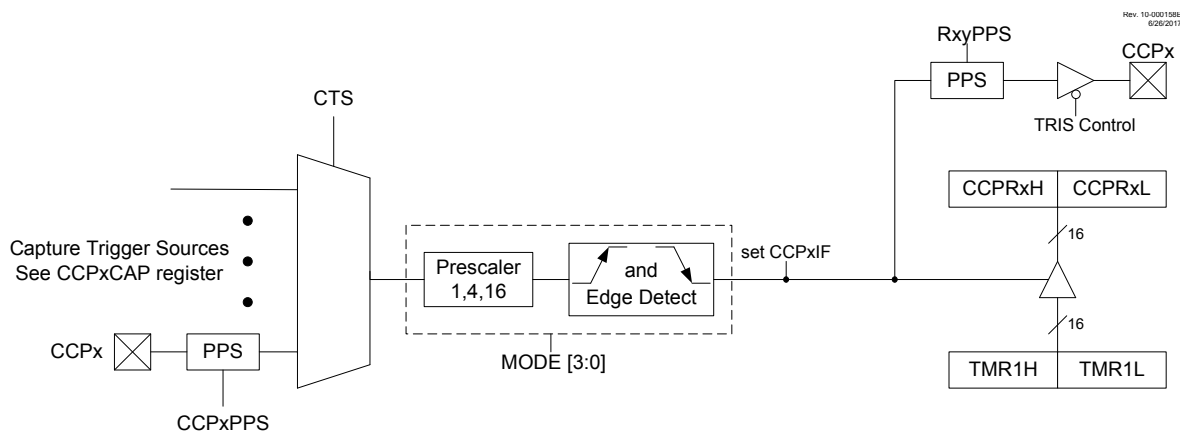
When a capture is made, the Interrupt Request Flag bit CCPxIF of the PIR6 register is set. The interrupt flag must be cleared in software. If another capture occurs before the value in the CCPRx register is read, the old captured value is overwritten by the new captured value.



Important: If an event occurs during a 2-byte read, the high and low-byte data will be from different events. It is recommended while reading the CCPRxH:CCPRxL register pair to either disable the module or read the register pair twice for data integrity.

The following figure shows a simplified diagram of the capture operation.

Figure 22-1. Capture Mode Operation Block Diagram



22.2.1 Capture Sources

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



Important: If the CCPx pin is configured as an output, a write to the port can cause a capture condition.

The capture source is selected by configuring the **CTS** bits as shown in the following table:

Table 22-2. Capture Trigger Sources

| CTS | Source |
|-----|----------|
| 111 | CLC4_out |

|continued | |
|----------------|-------------------------|
| CTS | Source |
| 110 | CLC3_out |
| 101 | CLC2_out |
| 100 | CLC1_out |
| 011 | IOC_interrupt |
| 010 | C2_out |
| 001 | C1_out |
| 000 | Pin selected by CCPxPPS |

22.2.2 Timer1 Mode Resource

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

See section “*Timer1 Module with Gate Control*” for more information on configuring Timer1.

Related Links

[19. TMR1 - Timer1 Module with Gate Control](#)

22.2.3 Software Interrupt Mode

When the Capture mode is changed, a false capture interrupt may be generated. The user should keep the CCPxIE Interrupt Priority bit of the PIE6 register clear to avoid false interrupts. Additionally, the user should clear the CCPxIF Interrupt Flag bit of the PIR6 register following any change in Operating mode.



Important: Clocking Timer1 from the system clock (F_{OSC}) should not be used in Capture mode. In order for Capture mode to recognize the trigger event on the CCPx pin, Timer1 must be clocked from the instruction clock ($F_{OSC}/4$) or from an external clock source.

22.2.4 CCP Prescaler

There are four prescaler settings specified by the **MODE** bits. 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. The example below demonstrates the code to perform this function.

Example 22-1. Changing Between Capture Prescalers

```
BANKSEL CCP1CON      ;(only needed when CCP1CON is not in ACCESS space)
CLRF  CCP1CON        ;Turn CCP module off
MOVLW  NEW_CAPT_PS   ;CCP ON and Prescaler select → W
MOVWF  CCP1CON       ;Load CCP1CON with this value
```

22.2.5 Capture During Sleep

Capture mode depends upon the Timer1 module for proper operation. There are two options for driving the Timer1 module in Capture mode. It can be driven by the instruction clock ($F_{OSC}/4$), or by an external clock source.

When Timer1 is clocked by $F_{OSC}/4$, Timer1 will not increment during Sleep. When the device wakes from Sleep, Timer1 will continue from its previous state.

Capture mode will operate during Sleep when Timer1 is clocked by an external clock source.

22.3 Compare Mode

The Compare mode function described in this section is available and identical for all CCP modules.

Compare mode makes use of the 16-bit odd numbered Timer resources (Timer1, Timer3, etc.). The 16-bit value of the CCPRx register is constantly compared against the 16-bit value of the TMRx register. When a match occurs, one of the following events can occur:

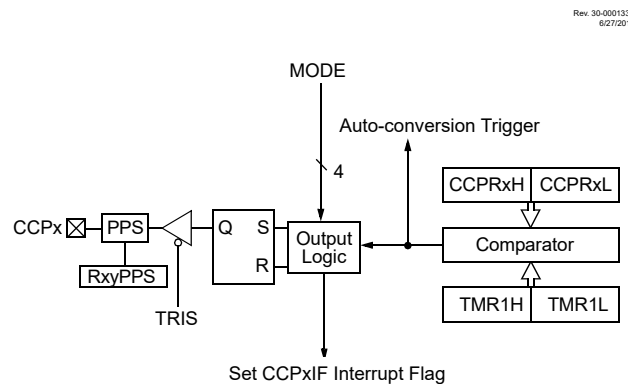
- Toggle the CCPx output and clear TMRx
- Toggle the CCPx output without clearing TMRx
- Set the CCPx output
- Clear the CCPx output
- Pulse output
- Pulse output and clear TMRx

The action on the pin is based on the value of the **MODE** control bits. At the same time, the interrupt flag CCPxIF bit is set, and an ADC conversion can be triggered, if selected.

All Compare modes can generate an interrupt and trigger an ADC conversion. When **MODE** = '0001' or '1011', the CCP resets the TMRx register.

The following figure shows a simplified diagram of the compare operation.

Figure 22-2. Compare Mode Operation Block Diagram



22.3.1 CCPx Pin Configuration

The software must configure the CCPx pin as an output by clearing the associated TRIS bit and defining the appropriate output pin through the RxyPPS registers. See section "Peripheral Pin Select (PPS) Module" for more details.

The CCP output can also be used as an input for other peripherals.



Important: Clearing the CCPxCON register will force the CCPx compare output latch to the default low level. This is not the PORT I/O data latch.

Related Links

[16. PPS - Peripheral Pin Select Module](#)

22.3.2 Timer1 Mode Resource

In Compare mode, Timer1 must be running in either Timer mode or Synchronized Counter mode. The compare operation may not work in Asynchronous Counter mode.

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



Important: Clocking Timer1 from the system clock (F_{OSC}) should not be used in Compare mode. In order for Compare mode to generate the trigger event on the CCPx pin, Timer1 must be clocked from the instruction clock ($F_{OSC}/4$) or from an external clock source.

22.3.3 Auto-Conversion Trigger

All CCPx modes set the CCP Interrupt Flag (CCPxIF). When this flag is set and a match occurs, an auto-conversion trigger can take place if the CCP module is selected as the conversion trigger source.

Refer to Section “Auto-Conversion Trigger” for more information.



Important: Removing the match condition by changing the contents of the CCPRxH and CCPRxL register pair, between the clock edge that generates the Auto-conversion Trigger and the clock edge that generates the Timer1 Reset, will preclude the Reset from occurring.

Related Links

[32.2.6 Auto-Conversion Trigger](#)

22.3.4 Compare During Sleep

Since F_{OSC} is shut down during Sleep mode, the Compare mode will not function properly during Sleep, unless the timer is running. The device will wake on interrupt (if enabled).

22.4 PWM Overview

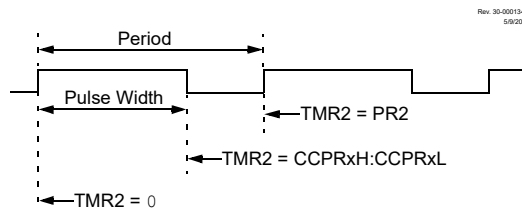
Pulse-Width Modulation (PWM) is a scheme that provides power to a load by switching quickly between fully ON and fully OFF states. The PWM signal resembles a square wave where the high portion of the signal is considered the ON state and the low portion of the signal is considered the OFF state. The high portion, also known as the pulse width, can vary in time and is defined in steps. A larger number of steps applied, which lengthens the pulse width, also supplies more power to the load. Lowering the number of steps applied, which shortens the pulse width, supplies less power. The PWM period is defined as the duration of one complete cycle or the total amount of ON and OFF time combined.

PWM resolution defines the maximum number of steps that can be present in a single PWM period. A higher resolution allows for more precise control of the pulse-width time and in turn the power that is applied to the load.

The term duty cycle describes the proportion of the ON time to the OFF time and is expressed in percentages, where 0% is fully OFF and 100% is fully ON. A lower duty cycle corresponds to less power applied and a higher duty cycle corresponds to more power applied.

The figure below shows a typical waveform of the PWM signal.

Figure 22-3. CCP PWM Output Signal



22.4.1 Standard PWM Operation

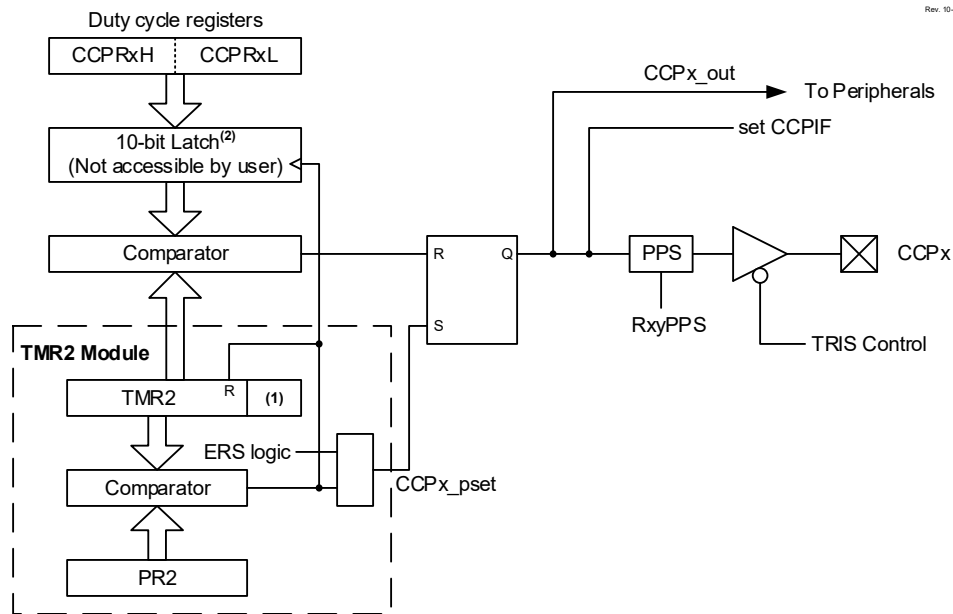
The standard PWM function described in this section is available and identical for all CCP modules.

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

- Even numbered TxPR registers (T2PR, T4PR, etc)
- Even numbered TxCON registers (T2CON, T4CON, etc)
- 16-bit CCPRx registers
- CCPxCON registers

It is required to have $F_{OSC}/4$ as the clock input to TxTMR for correct PWM operation. The following figure shows a simplified block diagram of PWM operation.

Figure 22-4. Simplified PWM Block Diagram



Note:

1. 8-bit timer is concatenated with two bits generated by F_{OSC} or two bits of the internal prescaler to create 10-bit time base.
2. The alignment of the 10 bits from the CCPRx register is determined by the CCPxFMT bit.



Important: The corresponding TRIS bit must be cleared to enable the PWM output on the CCPx pin.

22.4.2 Setup for PWM Operation

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

1. Use the desired output pin RxyPPS control to select CCPx as the source and disable the CCPx pin output driver by setting the associated TRIS bit.
2. Load the T2PR register with the PWM period value.
3. Configure the CCP module for the PWM mode by loading the CCPxCON register with the appropriate values.
4. Load the CCPRx register with the PWM duty cycle value and configure the FMT bit to set the proper register alignment.
5. Configure and start Timer2:
 - Clear the TMR2IF interrupt flag bit of the PIR4 register. See Note below.
 - Select the timer clock source to be as $F_{OSC}/4$ using the TxCLKCON register. This is required for correct operation of the PWM module.

- Configure the T2CKPS bits of the T2CON register with the timer prescale value.
 - Enable the timer by setting the T2ON bit.
6. Enable PWM output pin:
- Wait until the timer overflows and the TMR2IF bit of the PIR4 register is set. See Note below.
 - Enable the CCPx pin output driver by clearing the associated TRIS bit.



Important: In order to send a complete duty cycle and period on the first PWM output, the above steps must be included in the setup sequence. If it is not critical to start with a complete PWM signal on the first output, then step 6 may be ignored.

Related Links

[20.11.3 TxCON](#)

22.4.3 Timer2 Timer Resource

The PWM standard mode makes use of the 8-bit Timer2 timer resources to specify the PWM period.

22.4.4 PWM Period

The PWM period is specified by the T2PR register of Timer2. The PWM period can be calculated using the formula in the equation below.

Equation 22-1. PWM Period

$$PWMPeriod = [(T2PR + 1)] \cdot 4 \cdot T_{OSC} \cdot (TMR2PrescaleValue)$$

where $T_{OSC} = 1/F_{OSC}$

When T2TMR is equal to T2PR, the following three events occur on the next increment cycle:

- T2TMR 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 transferred from the CCPRx register into a 10-bit buffer.



Important: The Timer postscaler (see “*Timer2 Interrupt*”) is not used in the determination of the PWM frequency.

Related Links

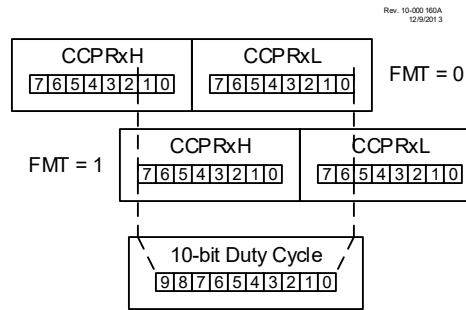
[20.4 Timer2 Interrupt](#)

22.4.5 PWM Duty Cycle

The PWM duty cycle is specified by writing a 10-bit value to the CCPRx register. The alignment of the 10-bit value is determined by the FMT bit (see [Figure 22-5](#)). The CCPRx register can be written to at any time. However, the duty cycle value is not latched into the 10-bit buffer until after a match between T2PR and T2TMR.

The equations below are used to calculate the PWM pulse width and the PWM duty cycle ratio.

Figure 22-5. PWM 10-Bit Alignment



Equation 22-2. Pulse Width

$$Pulse\ Width = (CCPRxH:CCPRxL\ register\ value) \cdot T_{OSC} \cdot (TMR2\ Prescale\ Value)$$

Equation 22-3. Duty Cycle

$$DutyCycleRatio = \frac{(CCPRxH:CCPRxL\ register\ value)}{4(T2PR + 1)}$$

The CCPRx register is used to double buffer the PWM duty cycle. This double buffering is essential for glitchless PWM operation.

The 8-bit timer T2TMR register is concatenated with either the 2-bit internal system clock (F_{OSC}), or two bits of the prescaler, to create the 10-bit time base. The system clock is used if the Timer2 prescaler is set to 1:1.

When the 10-bit time base matches the CCPRx register, then the CCPx pin is cleared (see Figure 22-4).

22.4.6 PWM Resolution

The resolution determines the number of available duty cycles for a given period. For example, a 10-bit resolution will result in 1024 discrete duty cycles, whereas an 8-bit resolution will result in 256 discrete duty cycles.

The maximum PWM resolution is ten bits when T2PR is 255. The resolution is a function of the T2PR register value as shown below.

Equation 22-4. PWM Resolution

$$Resolution = \frac{\log[4(T2PR + 1)]}{\log(2)}\ bits$$



Important: If the pulse-width value is greater than the period, the assigned PWM pin(s) will remain unchanged.

Table 22-3. Example PWM Frequencies and Resolutions ($F_{OSC} = 20\ MHz$)

| PWM Frequency | 1.22 kHz | 4.88 kHz | 19.53 kHz | 78.12 kHz | 156.3 kHz | 208.3 kHz |
|---------------------------|----------|----------|-----------|-----------|-----------|-----------|
| Timer Prescale | 16 | 4 | 1 | 1 | 1 | 1 |
| T2PR Value | 0xFF | 0xFF | 0xFF | 0x3F | 0x1F | 0x17 |
| Maximum Resolution (bits) | 10 | 10 | 10 | 8 | 7 | 6.6 |

Table 22-4. Example PWM Frequencies and Resolutions ($F_{OSC} = 8\ MHz$)

| PWM Frequency | 1.22 kHz | 4.90 kHz | 19.61 kHz | 76.92 kHz | 153.85 kHz | 200.0 kHz |
|----------------|----------|----------|-----------|-----------|------------|-----------|
| Timer Prescale | 16 | 4 | 1 | 1 | 1 | 1 |

|continued | | | | | | |
|---------------------------|----------|----------|-----------|-----------|------------|-----------|
| PWM Frequency | 1.22 kHz | 4.90 kHz | 19.61 kHz | 76.92 kHz | 153.85 kHz | 200.0 kHz |
| T2PR Value | 0x65 | 0x65 | 0x65 | 0x19 | 0x0C | 0x09 |
| Maximum Resolution (bits) | 8 | 8 | 8 | 6 | 5 | 5 |

22.4.7 Operation in Sleep Mode

In Sleep mode, the T2TMR 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, T2TMR will continue from the previous state.

22.4.8 Changes in System Clock Frequency

The PWM frequency is derived from the system clock frequency. Any changes in the system clock frequency will result in changes to the PWM frequency. See the “*Oscillator Module (with Fail-Safe Clock Monitor)*” section for additional details.

Related Links

[8. OSC - Oscillator Module](#)

22.4.9 Effects of Reset

Any Reset will force all ports to Input mode and the CCP registers to their Reset states.

22.5 Register Summary - CCP Control

| Address | Name | Bit Pos. | | | | | | | |
|-----------------------|----------|----------|------------|--|-----|-----|-----------|--|--|
| 0x00 ... 0x030B | Reserved | | | | | | | | |
| 0x030C | CCPR1 | 7:0 | CCPR[7:0] | | | | | | |
| | | 15:8 | CCPR[15:8] | | | | | | |
| 0x030E | CCP1CON | 7:0 | EN | | OUT | FMT | MODE[3:0] | | |
| 0x030F | CCP1CAP | 7:0 | | | | | CTS[2:0] | | |
| 0x0310 | CCPR2 | 7:0 | CCPR[7:0] | | | | | | |
| | | 15:8 | CCPR[15:8] | | | | | | |
| 0x0312 | CCP2CON | 7:0 | EN | | OUT | FMT | MODE[3:0] | | |
| 0x0313 | CCP2CAP | 7:0 | | | | | CTS[2:0] | | |
| 0x0314 | CCPR3 | 7:0 | CCPR[7:0] | | | | | | |
| | | 15:8 | CCPR[15:8] | | | | | | |
| 0x0316 | CCP3CON | 7:0 | EN | | OUT | FMT | MODE[3:0] | | |
| 0x0317 | CCP3CAP | 7:0 | | | | | CTS[2:0] | | |
| 0x0318 | CCPR4 | 7:0 | CCPR[7:0] | | | | | | |
| | | 15:8 | CCPR[15:8] | | | | | | |
| 0x031A | CCP4CON | 7:0 | EN | | OUT | FMT | MODE[3:0] | | |
| 0x031B | CCP4CAP | 7:0 | | | | | CTS[2:0] | | |

22.6 Register Definitions: CCP Control

Long bit name prefixes for the CCP peripherals are shown in the following table. Refer to the “*Long Bit Names*” section for more information.

Table 22-5. CCP Long Bit Name Prefixes

| Peripheral | Bit Name Prefix |
|------------|-----------------|
| CCP1 | CCP1 |
| CCP2 | CCP2 |
| CCP3 | CCP3 |
| CCP4 | CCP4 |

Related Links

[1.4.2.2 Long Bit Names](#)

22.6.1 CCPxCON

Name: CCPxCON
Address: 0x30E,0x312,0x316,0x31A

CCP Control Register

| | | | | | | | | |
|--------|-----|---|-----|-----|-----------|-----|-----|-----|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | EN | | OUT | FMT | MODE[3:0] | | | |
| Access | R/W | | RO | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | | x | 0 | 0 | 0 | 0 | 0 |

Bit 7 – EN CCP Module Enable bit

| Value | Description |
|-------|-----------------|
| 1 | CCP is enabled |
| 0 | CCP is disabled |

Bit 5 – OUT CCP Output Data bit (read-only)

Bit 4 – FMT CCPW (pulse-width) Value Alignment bit

| Value | Condition | Description |
|-------|--------------|----------------------|
| x | Capture mode | Not used |
| x | Compare mode | Not used |
| 1 | PWM mode | Left-aligned format |
| 0 | PWM mode | Right-aligned format |

Bits 3:0 – MODE[3:0] CCP Mode Select bits

Table 22-6. CCPx Mode Select Bits

| MODE | Operating Mode | Operation | Set CCPxIF |
|------|----------------|--|------------|
| 11xx | PWM | PWM Operation | Yes |
| 1011 | Compare | Pulse output; clear TMR1 ⁽²⁾ | Yes |
| 1010 | | Pulse output | Yes |
| 1001 | | Clear output ⁽¹⁾ | Yes |
| 1000 | | Set output ⁽¹⁾ | Yes |
| 0111 | Capture | Every 16 th rising edge of CCPx input | Yes |
| 0110 | | Every 4 th rising edge of CCPx input | Yes |
| 0101 | | Every rising edge of CCPx input | Yes |
| 0100 | | Every falling edge of CCPx input | Yes |
| 0011 | | Every edge of CCPx input | Yes |
| 0010 | Compare | Toggle output | Yes |
| 0001 | | Toggle output; clear TMR1 ⁽²⁾ | Yes |
| 0000 | Disabled | | — |

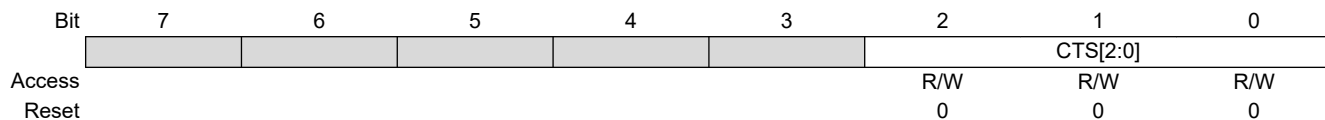
Note:

1. The set and clear operations of the Compare mode are Reset by setting MODE = '0000'.
2. When MODE = '0001' or '1011', then the timer associated with the CCP module is cleared. TMR1 is the default selection for the CCP module, so it is used for indication purpose only.

22.6.2 CCPxCAP

Name: CCPxCAP
Address: 0x30F,0x313,0x317,0x31B

Capture Trigger Input Selection Register



Bits 2:0 – CTS[2:0] Capture Trigger Input Selection bits

Table 22-7. Capture Trigger Sources

| CTS | Source |
|-----|-------------------------|
| 111 | CLC4_out |
| 110 | CLC3_out |
| 101 | CLC2_out |
| 100 | CLC1_out |
| 011 | IOC_interrupt |
| 010 | C2_out |
| 001 | C1_out |
| 000 | Pin selected by CCPxPPS |

22.6.3 CCPRx

Name: CCPRx
Address: 0x30C,0x310,0x314,0x318

Capture/Compare/Pulse Width Register

| | | | | | | | | | |
|--------|-----|------------|-----|-----|-----|-----|-----|-----|-----|
| | Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| | | CCPR[15:8] | | | | | | | |
| Access | | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | x | x | x | x | x | x | x | x |
| | | | | | | | | | |
| | Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | CCPR[7:0] | | | | | | | |
| Access | | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | x | x | x | x | x | x | x | x |

Bits 15:0 – CCPR[15:0] Capture/Compare/Pulse Width

Note: The individual bytes in this multi-byte register can be accessed with the following register names:

- When **MODE** = Capture or Compare
 - CCPRxH: Accesses the high byte CCPR[15:8]
 - CCPRxL: Accesses the low byte CCPR[7:0]
- When **MODE** = PWM and **FMT**=0
 - CCPRxH[7:2]: Not used
 - CCPRxH[1:0]: Accesses the two Most Significant bits CCPR[9:8]
 - CCPRxL: Accesses the eight Least Significant bits CCPR[7:0]
- When **MODE** = PWM and **FMT**=1
 - CCPRxH: Accesses the eight Most Significant bits CCPR[9:2]
 - CCPRxL[7:6]: Accesses the two Least Significant bits CCPR[1:0]
 - CCPRxL[5:0]: Not used

23. CCP/PWM Timer Resource Selection

Each CCP/PWM module has an independent timer selection which can be accessed using the CxTSEL or PxTSEL bits in the [CCPTMRS0](#) and/or [CCPTMRS1](#) registers. The default timer selection is Timer1 when using Capture/Compare mode and Timer2 when using PWM mode in the CCPx module. The default timer selection for the PWM module is always Timer2.

23.1 Register Summary - Timer Selection Registers for CCP/PWM

| Address | Name | Bit Pos. | | | | | | |
|-------------|--------------------------|----------|-------------|-------------|-------------|-------------|--|--|
| 0x00 ... | Reserved | | | | | | | |
| 0x021D | | | | | | | | |
| 0x021E | CCPTMRS0 | 7:0 | C4TSEL[1:0] | C3TSEL[1:0] | C2TSEL[1:0] | C1TSEL[1:0] | | |
| 0x021F | CCPTMRS1 | 7:0 | | P7TSEL[1:0] | P6TSEL[1:0] | | | |

23.2 Register Definitions: CCP/PWM Timer Selection

23.2.1 CCPTMRS0

Name: CCPTMRS0

Address: 0x21E

CCP Timers Selection Register0

| | | | | | | | | |
|--------|-------------|-----|-------------|-----|-------------|-----|-------------|-----|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | C4TSEL[1:0] | | C3TSEL[1:0] | | C2TSEL[1:0] | | C1TSEL[1:0] | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |

Bits 0:1, 2:3, 4:5, 6:7 – CxTSEL CCPx Timer Selection bits

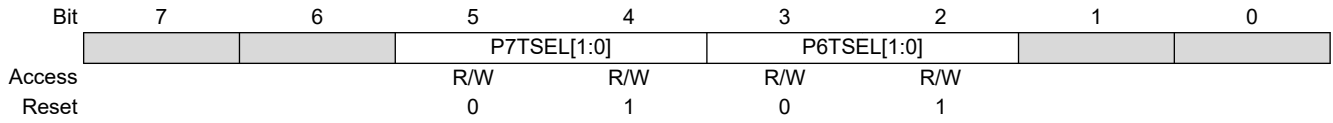
| Value | Description |
|-------|---|
| 11 | CCPx is based off Timer5 in Capture/Compare mode and Timer6 in PWM mode |
| 10 | CCPx is based off Timer3 in Capture/Compare mode and Timer4 in PWM mode |
| 01 | CCPx is based off Timer1 in Capture/Compare mode and Timer2 in PWM mode |
| 00 | Reserved |

23.2.2 CCPTMRS1

Name: CCPTMRS1

Address: 0x21F

CCP Timers Control Register



Bits 2:3, 4:5 – PxTSEL PWMx Timer Selection bits

| Value | Description |
|-------|--------------------|
| 11 | PWMx based on TMR6 |
| 10 | PWMx based on TMR4 |
| 01 | PWMx based on TMR2 |
| 00 | Reserved |

24. PWM - Pulse-Width Modulation

The PWM module generates a Pulse-Width Modulated signal determined by the duty cycle, period, and resolution that are configured by the following registers:

- TxPR
- TxCON
- PWMxDC
- PWMxCON



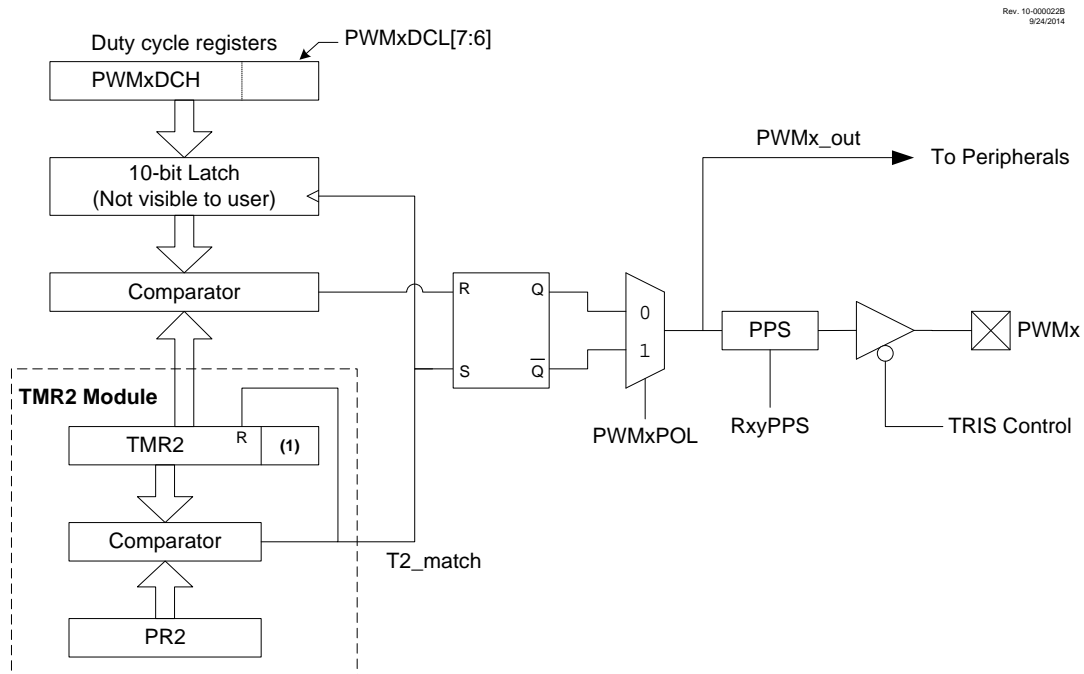
Important: The corresponding TRIS bit must be cleared to enable the PWM output on the PWMx pin.

Each PWM module can select the timer source that controls the module. Note that the PWM mode operation is described with respect to TMR2 in the following sections.

Figure 24-1 shows a simplified block diagram of PWM operation.

Figure 24-2 shows a typical waveform of the PWM signal.

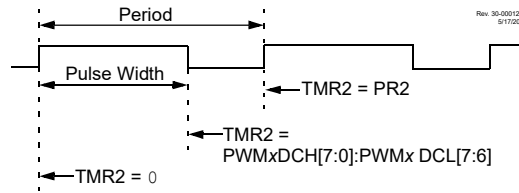
Figure 24-1. Simplified PWM Block Diagram



Note:

1. 8-bit timer is concatenated with two bits generated by Fosc or two bits of the internal prescaler to create 10-bit time base.

Figure 24-2. PWM Output



For a step-by-step procedure on how to set up this module for PWM operation, refer to [24.9 Setup for PWM Operation using PWMx Output Pins](#).

24.1 Fundamental Operation

The PWM module produces a 10-bit resolution output. The PWM timer can be selected using the PXTSEL bits in the CCPTMRS register. The default selection for PWMx is TMR2. Note that the PWM module operation in the following sections is described with respect to TMR2. Timer2 and T2PR set the period of the PWM. The PWMxDCH and PWMxDCL registers configure the duty cycle. The period is common to all PWM modules, whereas the duty cycle is independently controlled.



Important: The Timer2 postscaler is not used in the determination of the PWM frequency. The postscaler could be used to have a servo update rate at a different frequency than the PWM output.

All PWM outputs associated with Timer2 are set when T2TMR is cleared. Each PWMx is cleared when TxTMR is equal to the value specified in the corresponding PWMxDCH (8 MSb) and PWMxDCL[7:6] (2 LSb) registers. When the value is greater than or equal to T2PR, the PWM output is never cleared (100% duty cycle).



Important: The PWMxDCH and PWMxDCL registers are double-buffered. The buffers are updated when T2TMR matches T2PR. Care should be taken to update both registers before the timer match occurs.

24.2 PWM Output Polarity

The output polarity is inverted by setting the POL bit in the PWMxCON register.

24.3 PWM Period

The PWM period is specified by the TxPR register. The PWM period can be calculated using the formula of [Equation 24-1](#). It is required to have $F_{OSC}/4$ as the selected clock input to the timer for correct PWM operation.

Equation 24-1. PWM Period

$$PWMPeriod = [(T2PR) + 1] \cdot 4 \cdot T_{osc} \cdot (TMR2 \text{ PrescaleValue})$$

Note: $T_{osc} = 1/F_{osc}$

When T2TMR is equal to T2PR, the following three events occur on the next increment cycle:

- T2TMR is cleared
- The PWM output is active. (Exception: When the PWM duty cycle = 0%, the PWM output will remain inactive.)
- The PWMxDCH and PWMxDCL register values are latched into the buffers.



Important: The Timer2 postscaler has no effect on the PWM operation.

24.4 PWM Duty Cycle

The PWM duty cycle is specified by writing a 10-bit value to the PWMxDCH and PWMxDCL register pair. The PWMxDCH register contains the eight MSBs and the PWMxDCL[7:6], the two LSbs. The PWMxDCH and PWMxDCL registers can be written to at any time.

The formulas below are used to calculate the PWM pulse width and the PWM duty cycle ratio.

Equation 24-2. Pulse Width

$$PulseWidth = (PWMxDCH:PWMxDCL[7:6]) \cdot T_{osc} \cdot (TMR2PrescaleValue)$$

Note: $T_{osc} = 1/F_{osc}$

Equation 24-3. Duty Cycle Ratio

$$DutyCycleRatio = \frac{(PWMxDCH:PWMxDCL[7:6])}{4(T2PR + 1)}$$

The 8-bit timer T2TMR register is concatenated with the two Least Significant bits of $1/F_{osc}$, adjusted by the Timer2 prescaler to create the 10-bit time base. The system clock is used if the Timer2 prescaler is set to 1:1.

24.5 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 ten bits when T2PR is 255. The resolution is a function of the T2PR register value as shown below.

Equation 24-4. PWM Resolution

$$Resolution = \frac{\log[4(T2PR + 1)]}{\log(2)} \text{ bits}$$



Important: If the pulse-width value is greater than the period, the assigned PWM pin(s) will remain unchanged.

Table 24-1. Example PWM Frequencies and Resolutions ($F_{osc} = 20 \text{ MHz}$)

| PWM Frequency | 0.31 kHz | 4.88 kHz | 19.53 kHz | 78.12 kHz | 156.3 kHz | 208.3 kHz |
|---------------------------|----------|----------|-----------|-----------|-----------|-----------|
| Timer Prescale | 64 | 4 | 1 | 1 | 1 | 1 |
| T2PR Value | 0xFF | 0xFF | 0xFF | 0x3F | 0x1F | 0x17 |
| Maximum Resolution (bits) | 10 | 10 | 10 | 8 | 7 | 6.6 |

Table 24-2. Example PWM Frequencies and Resolutions ($F_{osc} = 8 \text{ MHz}$)

| PWM Frequency | 0.31 kHz | 4.90 kHz | 19.61 kHz | 76.92 kHz | 153.85 kHz | 200.0 kHz |
|----------------|----------|----------|-----------|-----------|------------|-----------|
| Timer Prescale | 64 | 4 | 1 | 1 | 1 | 1 |
| T2PR Value | 0x65 | 0x65 | 0x65 | 0x19 | 0x0C | 0x09 |

|continued | | | | | | |
|---------------------------|----------|----------|-----------|-----------|------------|-----------|
| PWM Frequency | 0.31 kHz | 4.90 kHz | 19.61 kHz | 76.92 kHz | 153.85 kHz | 200.0 kHz |
| Maximum Resolution (bits) | 8 | 8 | 8 | 6 | 5 | 5 |

24.6 Operation in Sleep Mode

In Sleep mode, the T2TMR register will not increment and the state of the module will not change. If the PWMx pin is driving a value, it will continue to drive that value. When the device wakes up, T2TMR will continue from its previous state.

24.7 Changes in System Clock Frequency

The PWM frequency is derived from the system clock frequency (F_{OSC}). Any changes in the system clock frequency will result in changes to the PWM frequency.

Related Links

[8. OSC - Oscillator Module](#)

24.8 Effects of Reset

Any Reset will force all ports to Input mode and the PWM registers to their Reset states.

24.9 Setup for PWM Operation using PWMx Output Pins

The following steps should be taken when configuring the module for PWM operation using the PWMx pins:

1. Disable the PWMx pin output driver(s) by setting the associated TRIS bit(s).
2. Clear the PWMxCON register.
3. Load the T2PR register with the PWM period value.
4. Load the PWMxDCH register and bits [7:6] of the PWMxDCL register with the PWM duty cycle value.
5. Configure and start Timer2:
 - Clear the TMR2IF interrupt flag bit of the PIR4 register.⁽¹⁾
 - Select the timer clock source to be as $F_{OSC}/4$ using the TxCLKCON register. This is required for correct operation of the PWM module.
 - Configure the T2CKPS bits of the T2CON register with the Timer2 prescale value.
 - Enable Timer2 by setting the T2ON bit of the T2CON register.
6. Enable PWM output pin and wait until Timer2 overflows, TMR2IF bit of the PIR4 register is set.⁽²⁾
7. Enable the PWMx pin output driver(s) by clearing the associated TRIS bit(s) and setting the desired pin PPS control bits.
8. Configure the PWM module by loading the PWMxCON register with the appropriate values.

Note:

1. In order to send a complete duty cycle and period on the first PWM output, the above steps must be followed in the order given. If it is not critical to start with a complete PWM signal, then move Step 8 to replace Step 4.
2. For operation with other peripherals only, disable PWMx pin outputs.

24.9.1 PWMx Pin Configuration

All PWM outputs are multiplexed with the PORT data latch. The user must configure the pins as outputs by clearing the associated TRIS bits.

24.10 Setup for PWM Operation to Other Device Peripherals

The following steps should be taken when configuring the module for PWM operation to be used by other device peripherals:

1. Disable the PWMx pin output driver(s) by setting the associated TRIS bit(s).
2. Clear the PWMxCON register.
3. Load the T2PR register with the PWM period value.
4. Load the PWMxDCH register and bits [7:6] of the PWMxDCL register with the PWM duty cycle value.
5. Configure and start Timer2:
 - Clear the TMR2IF interrupt flag bit of the PIR4 register.⁽¹⁾
 - Select the timer clock source to be as $F_{OSC}/4$ using the TxCLKCON register. This is required for correct operation of the PWM module.
 - Configure the T2CKPS bits of the T2CON register with the Timer2 prescale value.
 - Enable Timer2 by setting the T2ON bit of the T2CON register.
6. Wait until Timer2 overflows, TMR2IF bit of the PIR4 register is set.⁽¹⁾
7. Configure the PWM module by loading the PWMxCON register with the appropriate values.

Note:

1. In order to send a complete duty cycle and period on the first PWM output, the above steps must be included in the setup sequence. If it is not critical to start with a complete PWM signal on the first output, then step 6 may be ignored.

24.11 Register Summary - Registers Associated with PWM

| Address | Name | Bit Pos. | | | | | | | | |
|-------------|----------|----------|----------|--|-----|-----|--|--|--|--|
| 0x00 ... | Reserved | | | | | | | | | |
| 0x038B | | | | | | | | | | |
| 0x038C | PWM6DC | 7:0 | DCL[1:0] | | | | | | | |
| | | 15:8 | DCH[7:0] | | | | | | | |
| 0x038E | PWM6CON | 7:0 | EN | | OUT | POL | | | | |
| 0x038F | Reserved | | | | | | | | | |
| 0x0390 | PWM7DC | 7:0 | DCL[1:0] | | | | | | | |
| | | 15:8 | DCH[7:0] | | | | | | | |
| 0x0392 | PWM7CON | 7:0 | EN | | OUT | POL | | | | |

24.12 Register Definitions: PWM Control

24.12.1 PWMxDC

Name: PWMxDC
Address: 0x38C,0x390

PWM Duty Cycle Register

| | | | | | | | | | |
|--------|-----|----------|----|----|----|----|----|---|---|
| | Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| | | DCH[7:0] | | | | | | | |
| Access | | | | | | | | | |
| Reset | | x | x | x | x | x | x | x | x |
| | Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | DCL[1:0] | | | | | | | |
| Access | | | | | | | | | |
| Reset | | x | x | | | | | | |

Bits 15:8 – DCH[7:0] PWM Duty Cycle Most Significant bits
 These bits are the MSBs of the PWM duty cycle.
 Reset States: POR/BOR = xxxxxxxx
 All Other Resets = uuuuuuuu

Bits 7:6 – DCL[1:0] PWM Duty Cycle Least Significant bits
 These bits are the LSbs of the PWM duty cycle.
 Reset States: POR/BOR = xx
 All Other Resets = uu

24.12.2 PWMxCON

Name: PWMxCON
Address: 0x38E,0x392

PWM Control Register

| | | | | | | | | |
|--------|-----|---|-----|-----|---|---|---|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | EN | | OUT | POL | | | | |
| Access | R/W | | RO | R/W | | | | |
| Reset | 0 | | 0 | 0 | | | | |

Bit 7 – EN PWM Module Enable bit

| Value | Description |
|-------|------------------------|
| 1 | PWM module is enabled |
| 0 | PWM module is disabled |

Bit 5 – OUT PWM Module Output Level When Bit is Read

Bit 4 – POL PWM Output Polarity Select bit

| Value | Description |
|-------|------------------------|
| 1 | PWM output is inverted |
| 0 | PWM output is normal |

25. CWG - Complementary Waveform Generator

The Complementary Waveform Generator (CWG) produces half-bridge, full-bridge, and steering of PWM waveforms. It is backwards compatible with previous CCP functions. The PIC16(L)F18426/46 family has 2 instance(s) of the CWG module.

The CWG has the following features:

- Six Operating Modes:
 - Synchronous Steering mode
 - Asynchronous Steering mode
 - Full-Bridge mode, Forward
 - Full-Bridge mode, Reverse
 - Half-Bridge mode
 - Push-Pull mode
- Output Polarity Control
- Output Steering
- Independent 6-Bit Rising and Falling Event Dead-Band Timers:
 - Clocked dead band
 - Independent rising and falling dead-band enables
- Auto-Shutdown Control With:
 - Selectable shutdown sources
 - Auto-restart option
 - Auto-shutdown pin override control

25.1 Fundamental Operation

The CWG generates two output waveforms from the selected input source.

The off-to-on transition of each output can be delayed from the on-to-off transition of the other output, thereby, creating a time delay immediately where neither output is driven. This is referred to as dead time and is covered in section [25.7 Dead-Band Control](#).

It may be necessary to guard against the possibility of circuit faults or a feedback event arriving too late or not at all. In this case, the active drive must be terminated before the Fault condition causes damage. This is referred to as auto-shutdown and is covered in section [25.11 Auto-Shutdown](#).

25.2 Operating Modes

The CWG module can operate in six different modes, as specified by the [MODE](#) bits:

- Half-Bridge mode
- Push-Pull mode
- Asynchronous Steering mode
- Synchronous Steering mode
- Full-Bridge mode, Forward
- Full-Bridge mode, Reverse

All modes accept a single pulse data input, and provide up to four outputs as described in the following sections.

All modes include auto-shutdown control as described in [25.11 Auto-Shutdown](#).



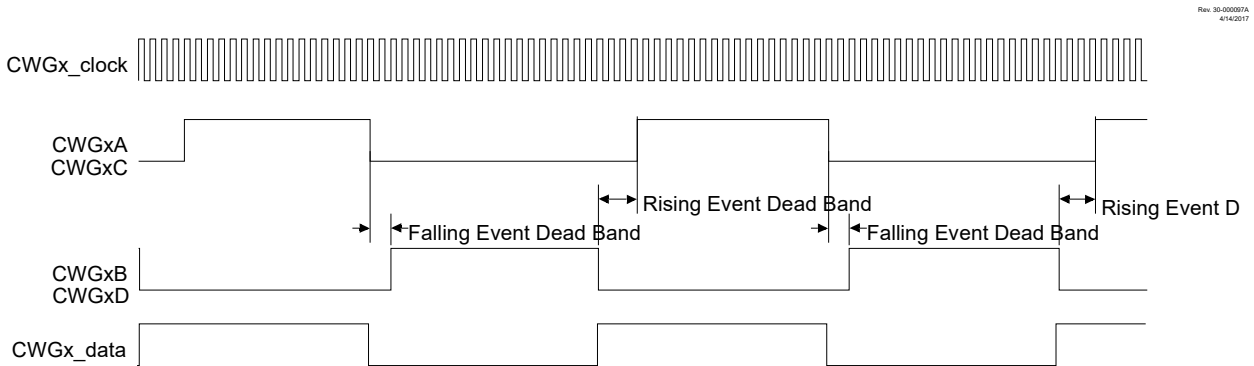
Important: Except as noted for Full-Bridge mode (25.2.3 Full-Bridge Modes), mode changes should only be performed while EN = 0.

25.2.1 Half-Bridge Mode

In Half-Bridge mode, two output signals are generated as true and inverted versions of the input as illustrated in Figure 25-1. A non-overlap (dead-band) time is inserted between the two outputs to prevent shoot-through current in various power supply applications. Dead-band control is described in 25.7 Dead-Band Control. The output steering feature cannot be used in this mode. A basic block diagram of this mode is shown in Figure 25-2.

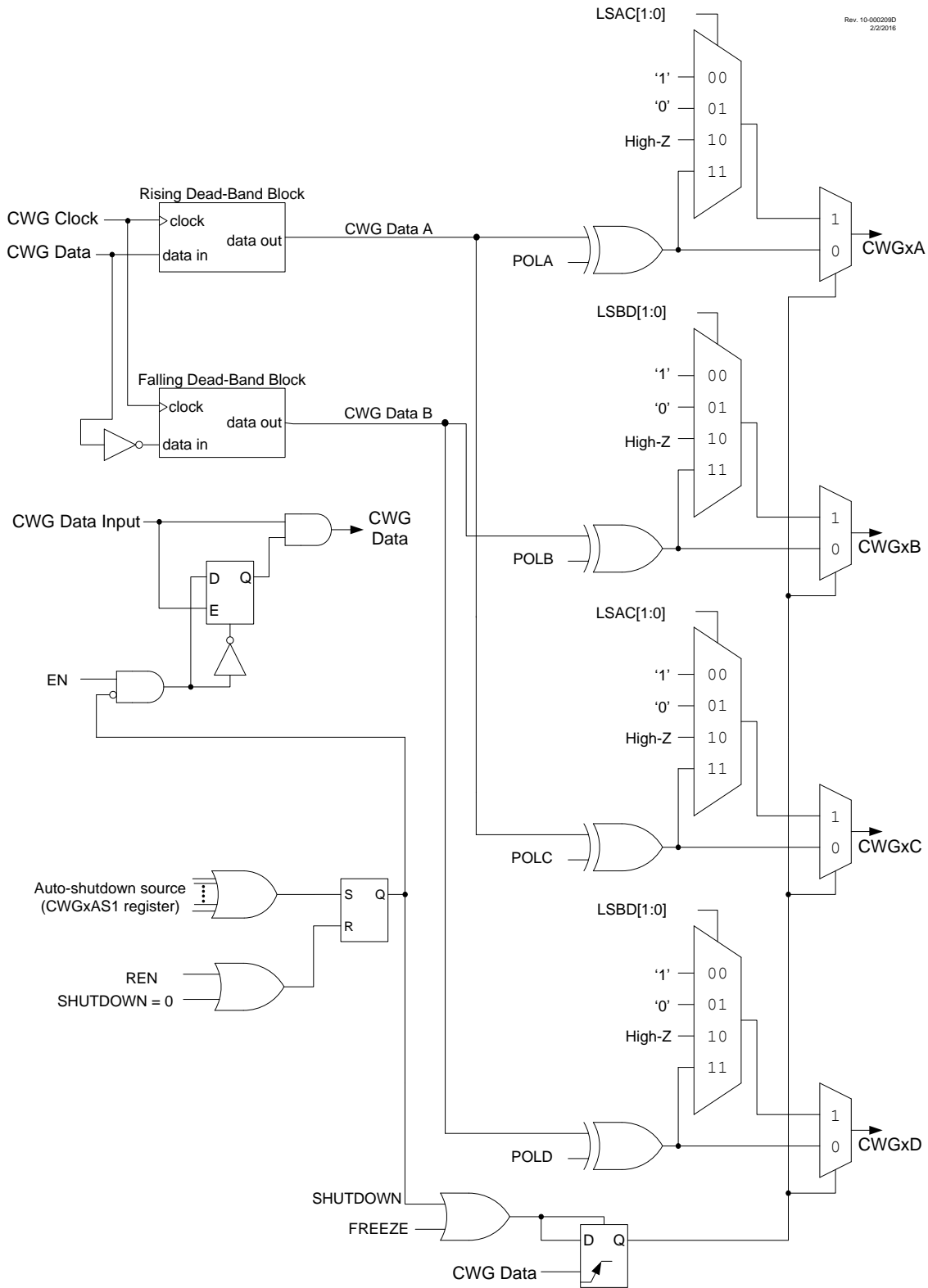
The unused outputs CWGxC and CWGxD drive similar signals, with polarity independently controlled by the POLC and POLD bits, respectively.

Figure 25-1. CWG Half-Bridge Mode Operation



Note: CWGx_rising_src = CCP1_out, CWGx_falling_src = ~CCP1_out

Figure 25-2. Simplified CWG Block Diagram (Half-Bridge Mode, MODE[2:0] = 100)



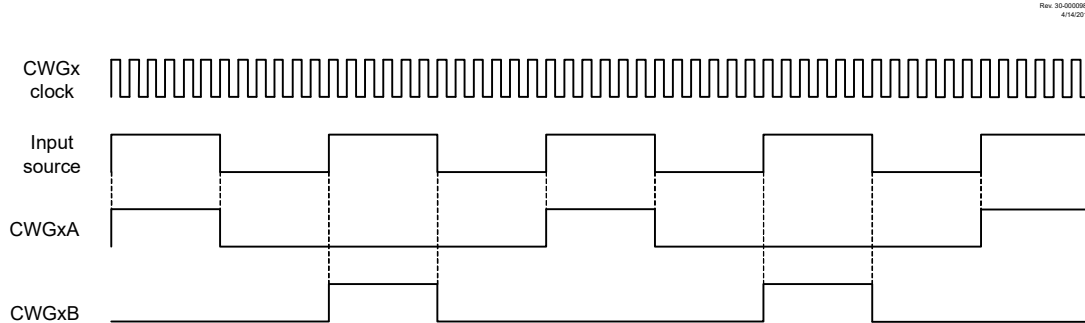
25.2.2 Push-Pull Mode

In Push-Pull mode, two output signals are generated, alternating copies of the input as illustrated in [Figure 25-3](#). This alternation creates the push-pull effect required for driving some transformer-based power supply designs. Steering modes are not used in Push-Pull mode. A basic block diagram for the Push-Pull mode is shown in [Figure 25-4](#).

The push-pull sequencer is reset whenever EN = 0 or if an auto-shutdown event occurs. The sequencer is clocked by the first input pulse, and the first output appears on CWG1A.

The unused outputs CWGxC and CWGxD drive copies of CWGxA and CWGxB, respectively, but with polarity controlled by the POLC and POLD bits of the CWGxCON1 register, respectively.

Figure 25-3. CWG Push-Pull Mode Operation

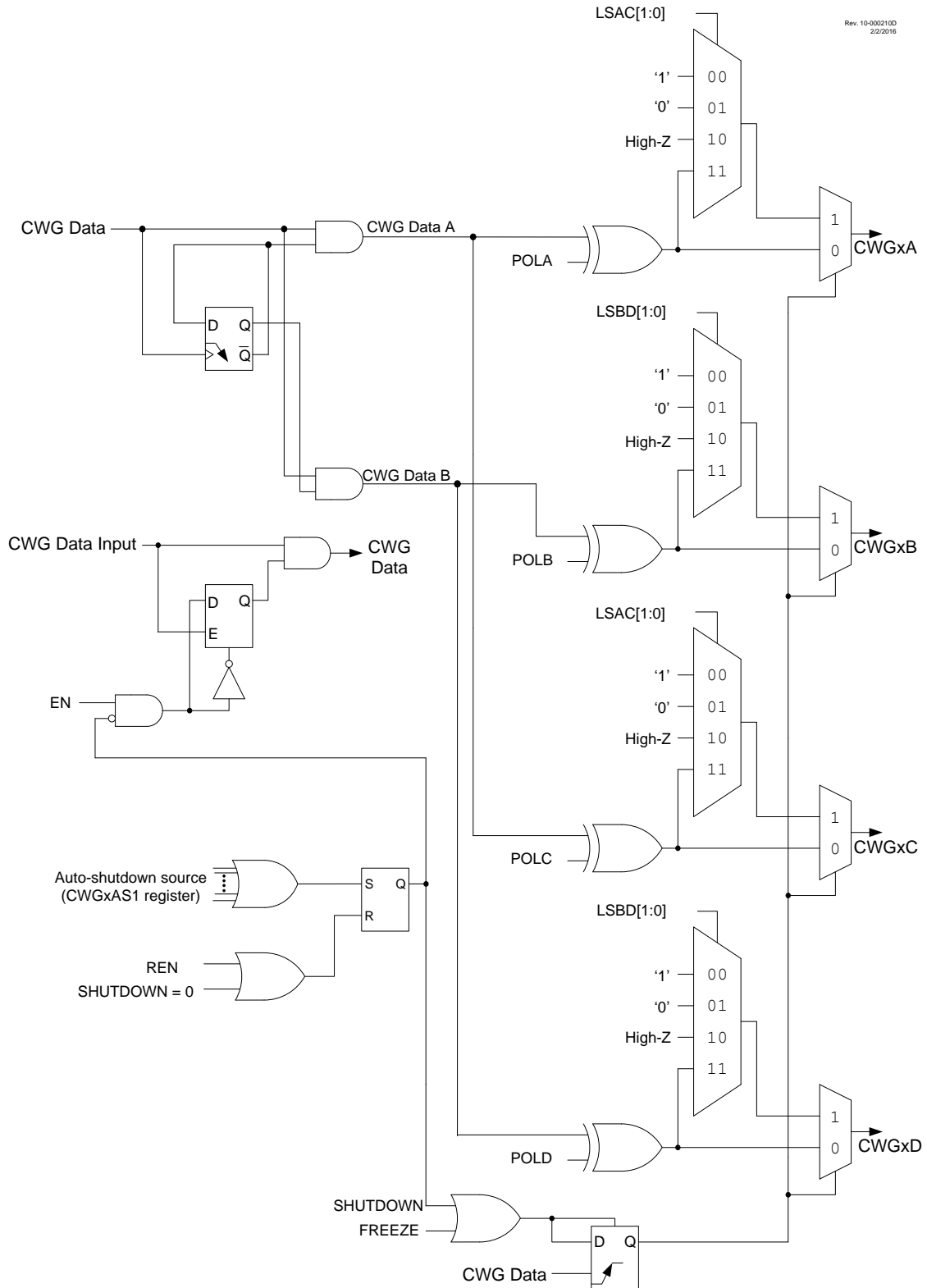


Rev. 30-000088A
4/14/2017

PIC16(L)F18426/46

CWG - Complementary Waveform Generator

Figure 25-4. Simplified CWG Block Diagram (Push-Pull Mode, MODE[2:0] = 101)



25.2.3 Full-Bridge Modes

In Forward and Reverse Full-Bridge modes, three outputs drive static values while the fourth is modulated by the input data signal. The mode selection may be toggled between forward and reverse by toggling the MODE[0] bit of the CWGxCON0 while keeping MODE[2:1] static, without disabling the CWG module. When connected, as shown in Figure 25-5, the outputs are appropriate for a full-bridge motor driver. Each CWG output signal has independent polarity control, so the circuit can be adapted to high-active and low-active drivers. A simplified block diagram for the Full-Bridge modes is shown in Figure 25-6.

Figure 25-5. Example of Full-Bridge Application

Rev. 10-00263A
1/28/2015

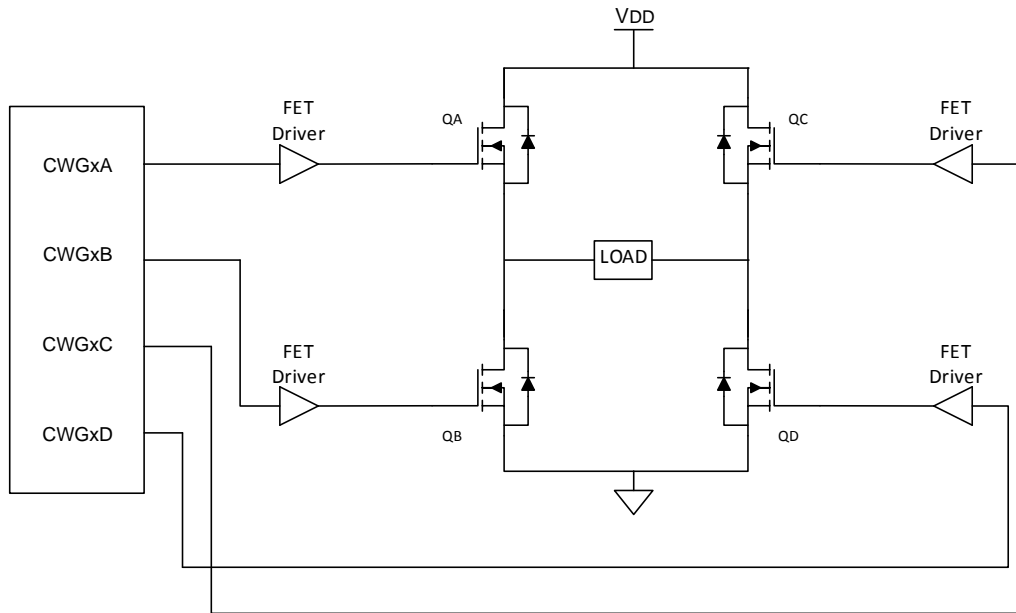
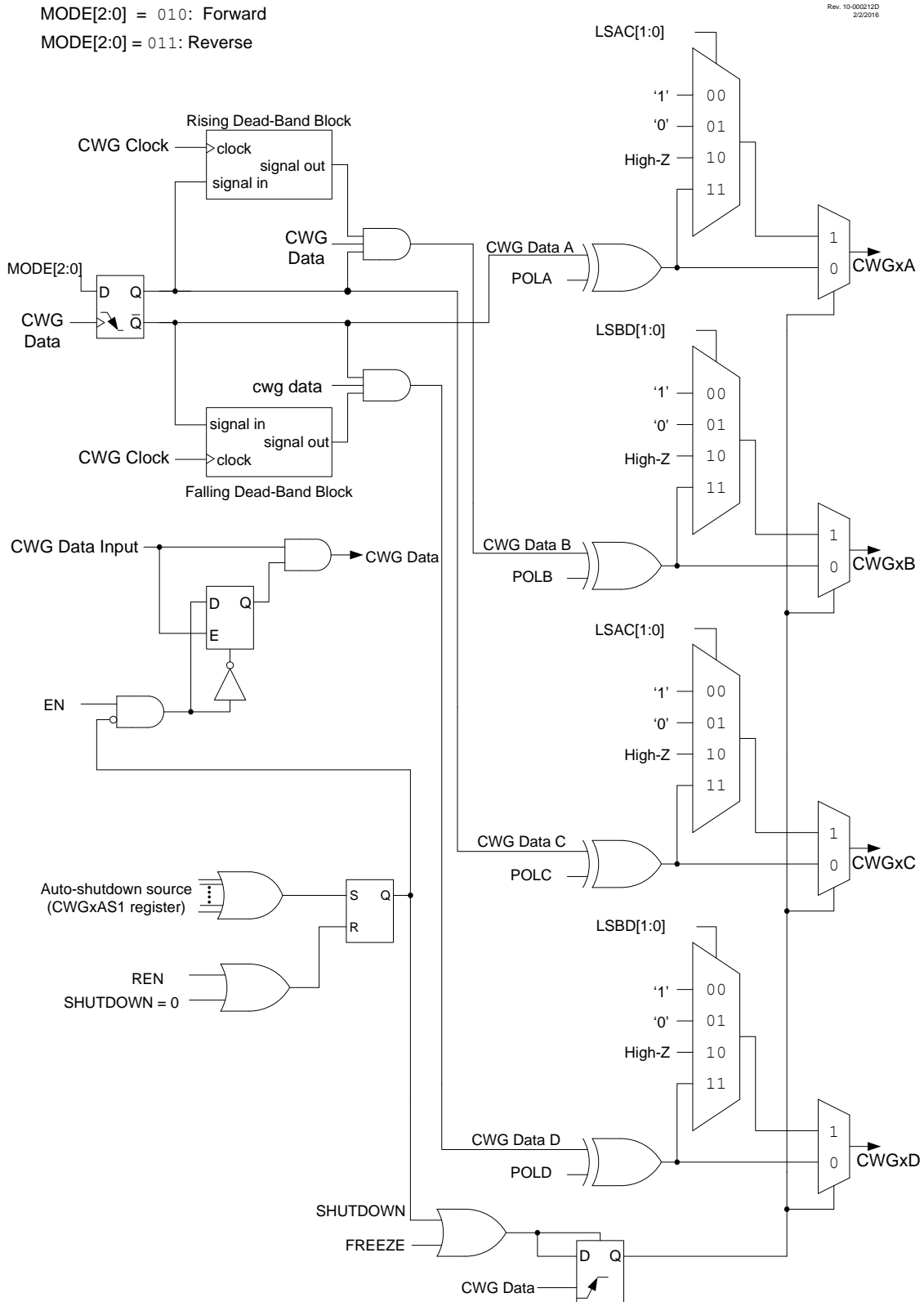


Figure 25-6. Simplified CWG Block Diagram (Forward and Reverse Full-Bridge Modes)

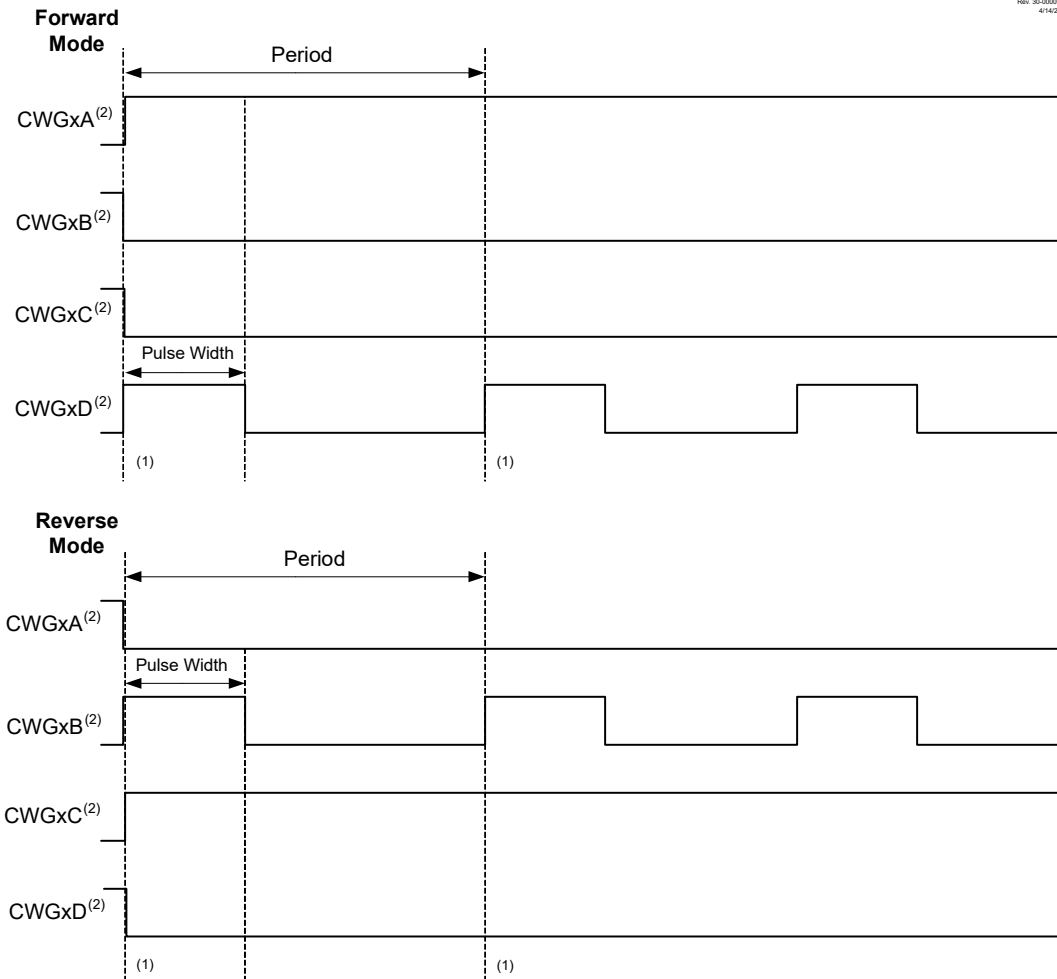


In Forward Full-Bridge mode (**MODE** = 010), CWGxA is driven to its Active state, CWGxB and CWGxC are driven to their Inactive state, and CWGxD is modulated by the input signal, as shown in [Figure 25-7](#).

In Reverse Full-Bridge mode ($MODE = 011$), CWGxC is driven to its Active state, CWGxA and CWGxD are driven to their Inactive states, and CWGxB is modulated by the input signal, as shown in Figure 25-7.

In Full-Bridge mode, the dead-band period is used when there is a switch from forward to reverse or vice versa. This dead-band control is described in 25.7 Dead-Band Control, with additional details in 25.8 Rising Edge and Reverse Dead Band and 25.9 Falling Edge and Forward Dead Band. Steering modes are not used with either of the Full-Bridge modes. The mode selection may be toggled between forward and reverse toggling the $MODE[0]$ bit of the CWGxCON0 while keeping $MODE[2:1]$ static, without disabling the CWG module.

Figure 25-7. Example of Full-Bridge Output



Note:

1. A rising CWG data input creates a rising event on the modulated output.
2. Output signals shown as active-high; all POL bits are clear.

25.2.3.1 Direction Change in Full-Bridge Mode

In Full-Bridge mode, changing $MODE$ controls the forward/reverse direction. Direction changes occur on the next rising edge of the modulated input.

A direction change is initiated in software by changing the $MODE$ bits. The sequence is illustrated in Figure 25-8.

- The associated active output CWGxA and the inactive output CWGxC are switched to drive in the opposite direction.
- The previously modulated output CWGxD is switched to the Inactive state, and the previously inactive output CWGxB begins to modulate.
- CWG modulation resumes after the direction-switch dead band has elapsed.

25.2.3.2 Dead-Band Delay in Full-Bridge Mode

Dead-band delay is important when either of the following conditions is true:

1. The direction of the CWG output changes when the duty cycle of the data input is at or near 100%, or
2. The turn-off time of the power switch, including the power device and driver circuit, is greater than the turn-on time.

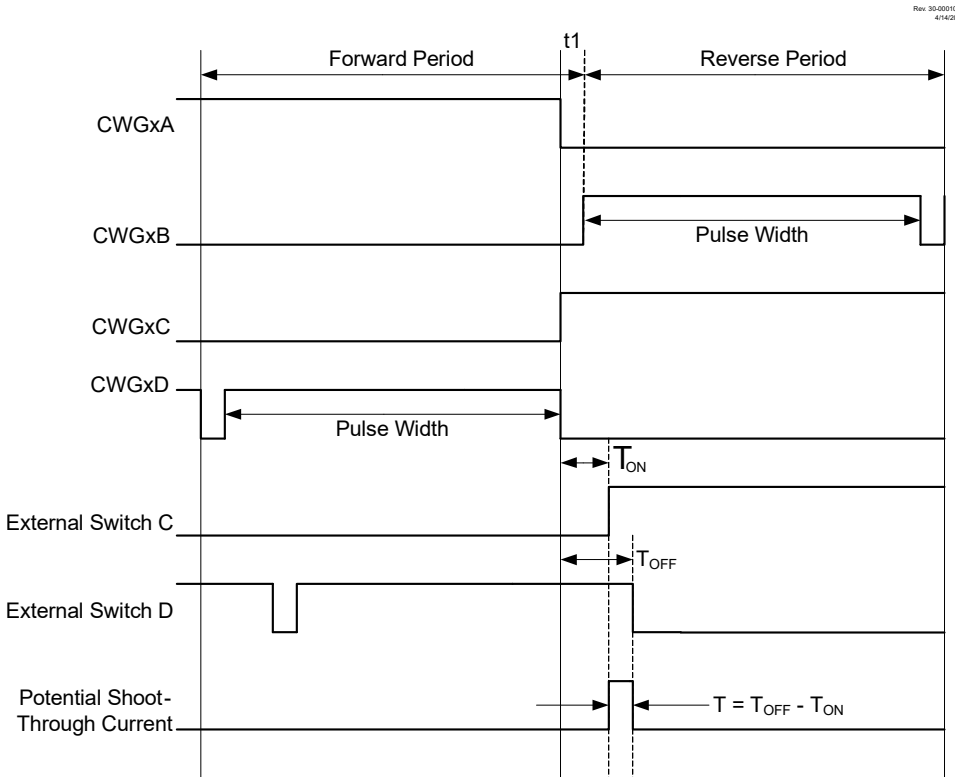
The dead-band delay is inserted only when changing directions, and only the modulated output is affected. The statically-configured outputs (CWGxA and CWGxC) are not afforded dead band, and switch essentially simultaneously.

The following figure shows an example of the CWG outputs changing directions from forward to reverse, at near 100% duty cycle. In this example, at time t1, the output of CWGxA and CWGxD become inactive, while output CWGxC 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 for the duration of 't'. The same phenomenon will occur to power devices QA and QB for the CWG direction change from reverse to forward.

When changing the CWG direction at high duty cycle is required for an application, two possible solutions for eliminating the shoot-through current are:

1. Reduce the CWG duty cycle for one period before changing directions.
2. Use switch drivers that can drive the switches off faster than they can drive them on.

Figure 25-8. Example of PWM Direction Change at Near 100% Duty Cycle



25.2.4 Steering Modes

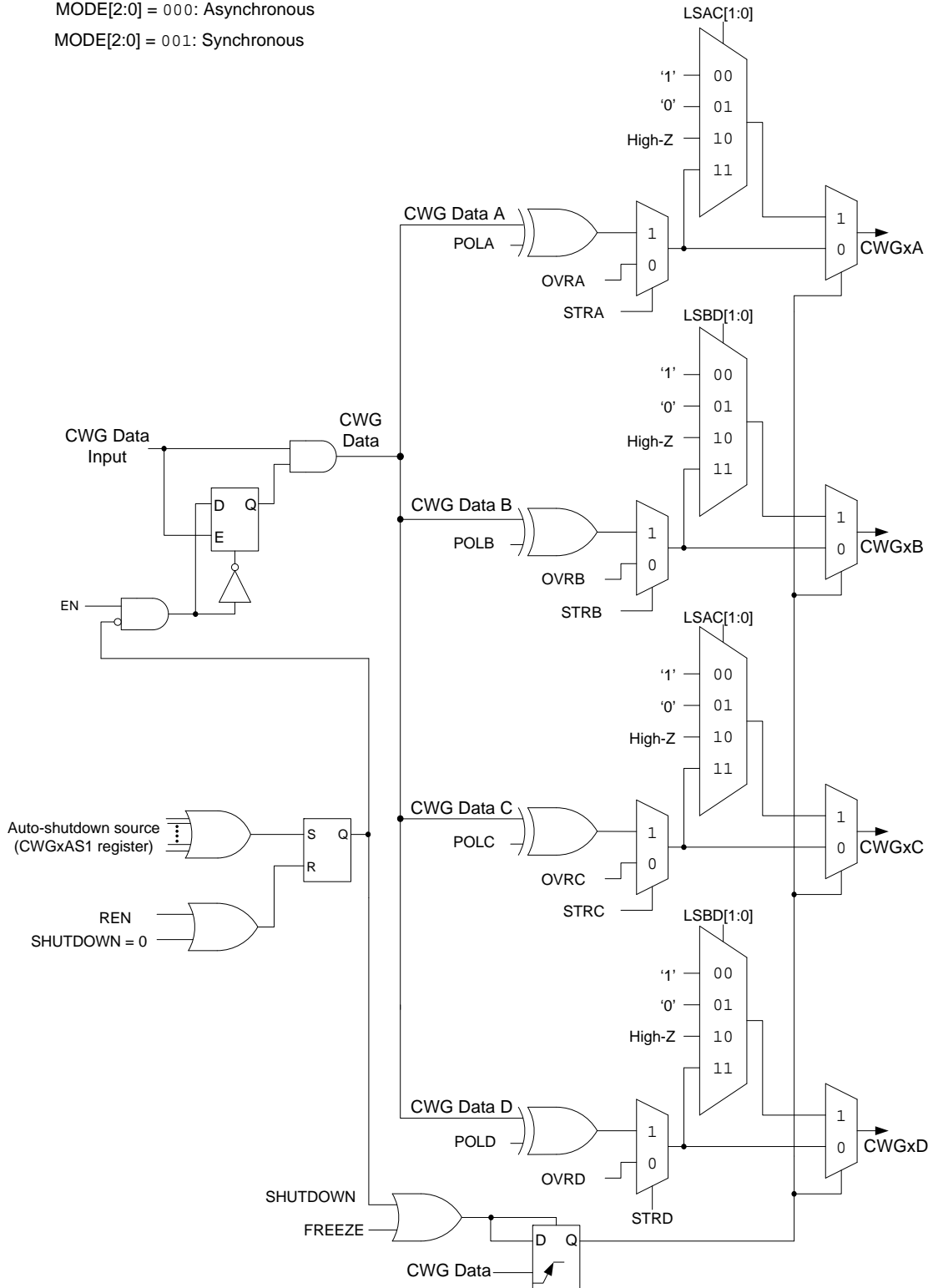
In both Synchronous and Asynchronous Steering modes, the modulated input signal can be steered to any combination of four CWG outputs. A fixed-value will be presented on all the outputs not used for the PWM output. Each output has independent polarity, steering, and shutdown options. Dead-band control is not used in either steering mode.

Figure 25-9. Simplified CWG Block Diagram (Output Steering Modes)

MODE[2:0] = 000: Asynchronous

MODE[2:0] = 001: Synchronous

Rev. 10-000211D
5/30/2017



For example, when $STRA = 0$ then the corresponding pin is held at the level defined by $OVRA$. When $STRA = 1$, then the pin is driven by the modulated input signal.

The $POLy$ bits control the signal polarity only when $STRy = 1$.

The CWG auto-shutdown operation also applies in Steering modes as described in “25.11 Auto-Shutdown”. An auto-shutdown event will only affect pins that have $STRy = 1$.

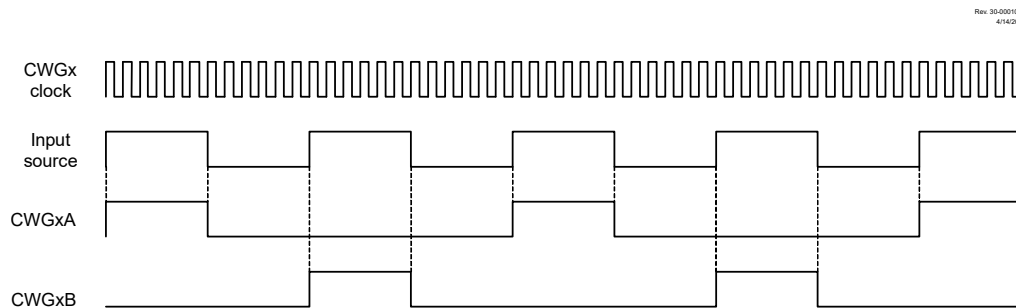
25.2.4.1 Synchronous Steering Mode

In Synchronous Steering mode ($MODE = 001$), changes to steering selection registers take effect on the next rising edge of the modulated data input (see figure below). In Synchronous Steering mode, the output will always produce a complete waveform.



Important: Only the $STRx$ bits are synchronized; the $OVRx$ bits are not synchronized.

Figure 25-10. Example of Synchronous Steering ($MODE = 001$)

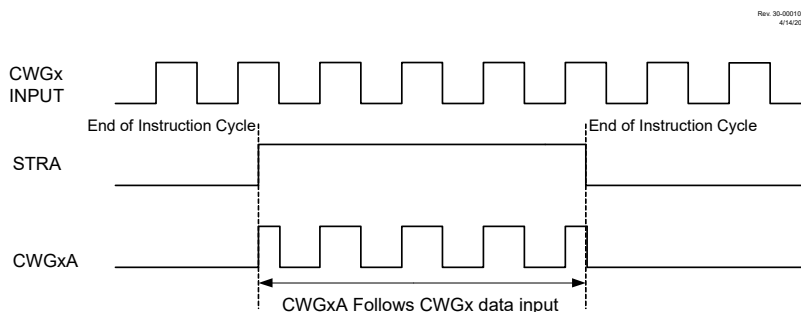


Rev. 30-00101A
4/14/2017

25.2.4.2 Asynchronous Steering Mode

In Asynchronous mode ($MODE = 000$), steering takes effect at the end of the instruction cycle that writes to $STRx$. In Asynchronous Steering mode, the output signal may be an incomplete waveform (see figure below). This operation may be useful when the user firmware needs to immediately remove a signal from the output pin.

Figure 25-11. Example of Asynchronous Steering ($MODE = 000$)



Rev. 30-00102A
4/14/2017

25.3 Start-up Considerations

The application hardware must use the proper external pull-up and/or pull-down resistors on the CWG output pins. This is required because all I/O pins are forced to high-impedance at Reset.

The polarity control bits ($POLy$) allow the user to choose whether the output signals are active-high or active-low.

25.4 Clock Source

The clock source is used to drive the dead-band timing circuits. The CWG module allows the following clock sources to be selected:

- F_{OSC} (system clock)
- HFINTOSC

When the HFINTOSC is selected, the HFINTOSC will be kept running during Sleep. Therefore, CWG modes requiring dead band can operate in Sleep, provided that the CWG data input is also active during Sleep. The clock sources are selected using the **CS** bit. The system clock F_{OSC} is disabled in Sleep and thus dead-band control cannot be used.

25.5 Selectable Input Sources

The CWG generates the output waveforms having the following input sources:

- Analog Comparators output
- Data Signal Modulator
- CLCs
- NCO
- PWM
- CCP
- Pin selected by CWGxINPPS

The input sources are selected using the **ISM** bits in the CWGxISM register.

25.6 Output Control

25.6.1 CWG Outputs

Each CWG output can be routed to a Peripheral Pin Select (PPS) output via the RxyPPS register.

Related Links

[16. PPS - Peripheral Pin Select Module](#)

25.6.2 Polarity Control

The polarity of each CWG output can be selected independently. When the output polarity bit is set, the corresponding output is active-high. Clearing the output polarity bit configures the corresponding output as active-low. However, polarity does not affect the override levels. Output polarity is selected with the **POLy** bits. Auto-shutdown and steering options are unaffected by polarity.

25.7 Dead-Band Control

The dead-band control provides non-overlapping PWM signals to prevent shoot-through current in PWM switches. Dead-band operation is employed for Half-Bridge and Full-Bridge modes. The CWG contains two 6-bit dead-band counters. One is used for the rising edge of the input source control in Half-Bridge mode or for reverse dead-band Full-Bridge mode. The other is used for the falling edge of the input source control in Half-Bridge mode or for forward dead band in Full-Bridge mode.

Dead band is timed by counting CWG clock periods from zero up to the value in the rising or falling dead-band counter registers.

25.7.1 Dead-Band Functionality in Half-Bridge mode

In Half-Bridge mode, the dead-band counters dictate the delay between the falling edge of the normal output and the rising edge of the inverted output. This can be seen in [Figure 25-1](#).

25.7.2 Dead-Band Functionality in Full-Bridge mode

In Full-Bridge mode, the dead-band counters are used when undergoing a direction change. The [MODE\[0\]](#) bit can be set or cleared while the CWG is running, allowing for changes from Forward to Reverse mode. The CWGxA and CWGxC signals will change immediately upon the first rising input edge following a direction change, but the modulated signals (CWGxB or CWGxD, depending on the direction of the change) will experience a delay dictated by the dead-band counters.

25.8 Rising Edge and Reverse Dead Band

In Half-Bridge mode, the rising edge dead band delays the turn-on of the CWGxA output after the rising edge of the CWG data input. In Full-Bridge mode, the reverse dead-band delay is only inserted when changing directions from Forward mode to Reverse mode, and only the modulated output CWGxB is affected.

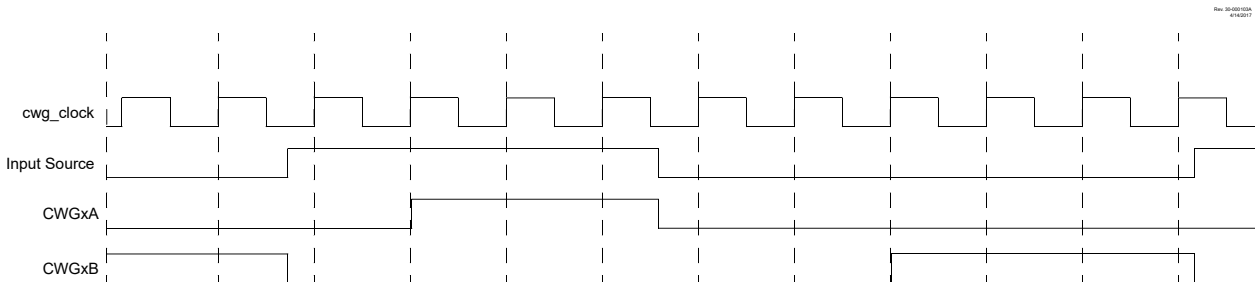
The [25.15.3 CWGxDBR](#) register determines the duration of the dead-band interval on the rising edge of the input source signal. This duration is from 0 to 64 periods of the CWG clock.

Dead band is always initiated on the edge of the input source signal. A count of zero indicates that no dead band is present.

If the input source signal reverses polarity before the dead-band count is completed, then no signal will be seen on the respective output.

The CWGxDBR register value is double-buffered. When [EN](#) = 0, the buffer is loaded when CWGxDBR is written. When [EN](#) = 1, then the buffer will be loaded at the rising edge following the first falling edge of the data input, after the [LD](#) bit is set. Refer to the following figure for an example.

Figure 25-12. Dead-Band Operation, CWGxDBR = 0x01, CWGxDBF = 0x02



25.9 Falling Edge and Forward Dead Band

In Half-Bridge mode, the falling edge dead band delays the turn-on of the CWGxB output at the falling edge of the CWG data input. In Full-Bridge mode, the forward dead-band delay is only inserted when changing directions from Reverse mode to Forward mode, and only the modulated output CWGxD is affected.

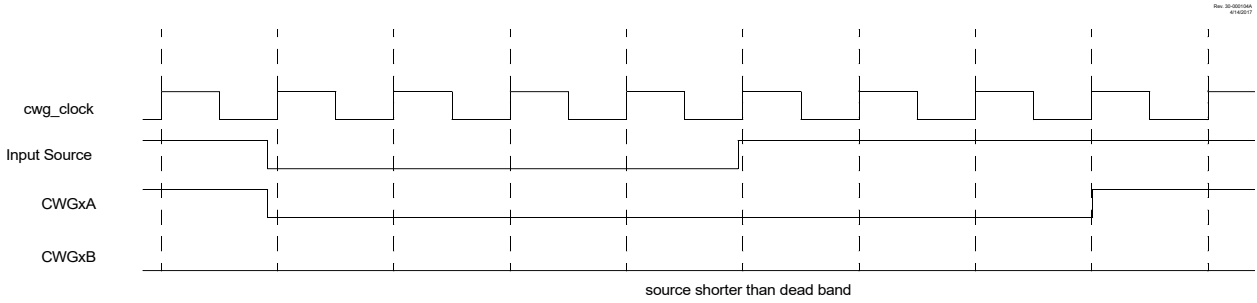
The [25.15.4 CWGxDBF](#) register determines the duration of the dead-band interval on the falling edge of the input source signal. This duration is from zero to 64 periods of CWG clock.

Dead-band delay is always initiated on the edge of the input source signal. A count of zero indicates that no dead band is present.

If the input source signal reverses polarity before the dead-band count is completed, then no signal will be seen on the respective output.

The CWGxDBF register value is double-buffered. When [EN](#) = 0, the buffer is loaded when CWGxDBF is written. When [EN](#) = 1, then the buffer will be loaded at the rising edge following the first falling edge of the data input after the [LD](#) is set. Refer to the following figure for an example.

Figure 25-13. Dead-Band Operation, CWGxDBR = 0x03, CWGxDBF = 0x06, Source Shorter Than Dead Band



25.10 Dead-Band Jitter

When the rising and falling edges of the input source are asynchronous to the CWG clock, it creates jitter in the dead-band time delay. The maximum jitter is equal to one CWG clock period. Refer to the equations below for more details.

Equation 25-1. Dead-Band Delay Time Calculation

$$T_{DEAD - BAND_MIN} = \frac{1}{F_{CWG_CLOCK}} \cdot DBx < 5:0 >$$

$$T_{DEAD - BAND_MAX} = \frac{1}{F_{CWG_CLOCK}} \cdot DBx < 5:0 > + 1$$

$$T_{JITTER} = T_{DEAD - BAND_MAX} - T_{DEAD - BAND_MIN}$$

$$T_{JITTER} = \frac{1}{F_{CWG_CLOCK}}$$

$$T_{DEAD - BAND_MAX} = T_{DEAD - BAND_MIN} + T_{JITTER}$$

Equation 25-2. Dead-Band Delay Example Calculation

$$DBx < 5:0 > = 0x0A = 10$$

$$F_{CWG_CLOCK} = 8 \text{ MHz}$$

$$T_{JITTER} = \frac{1}{8 \text{ MHz}} = 125 \text{ ns}$$

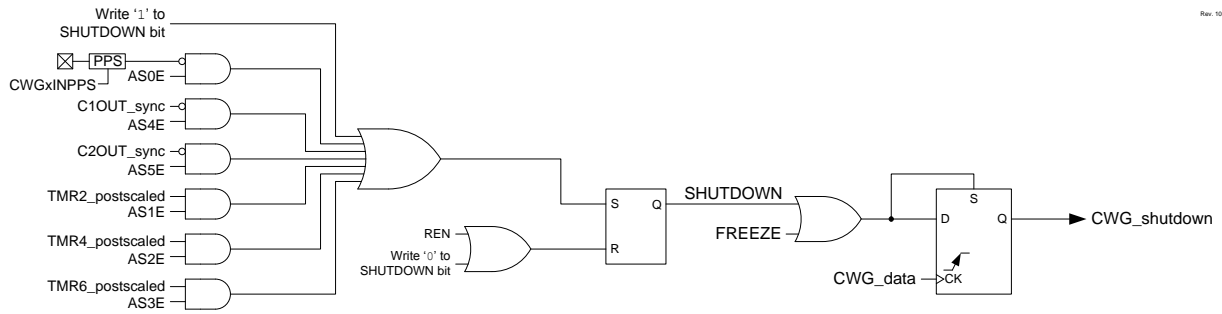
$$T_{DEAD - BAND_MIN} = 125 \text{ ns} \cdot 10 = 1.25 \mu\text{s}$$

$$T_{DEAD - BAND_MAX} = 1.25 \mu\text{s} + 0.125 \mu\text{s} = 1.37 \mu\text{s}$$

25.11 Auto-Shutdown

Auto-shutdown is a method to immediately override the CWG output levels with specific overrides that allow for safe shutdown of the circuit. The shutdown state can be either cleared automatically or held until cleared by software. The auto-shutdown circuit is illustrated in the following figure.

Figure 25-14. CWG Shutdown Block Diagram



25.11.1 Shutdown

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

- Software Generated
- External Input

25.11.1.1 Software Generated Shutdown

Setting the **SHUTDOWN** bit will force the CWG into the shutdown state.

When the auto-restart is disabled, the shutdown state will persist as long as the SHUTDOWN bit is set.

When auto-restart is enabled, the SHUTDOWN bit will clear automatically and resume operation on the next rising edge event. The SHUTDOWN bit indicates when a shutdown condition exists. The bit may be set or cleared in software or by hardware.

25.11.1.2 External Input Source

External shutdown inputs provide the fastest way to safely suspend CWG operation in the event of a Fault condition. When any of the selected shutdown inputs goes active, the CWG outputs will immediately go to the selected override levels without software delay. The override levels are selected by the **LSBD** and **LSAC** bits in CWGxAS0 register. The following input sources can be selected to cause a shutdown condition:

- CLC2_out/CLC3_out (low causes shutdown)
- CMP2_out (low causes shutdown)
- CMP1_out (low causes shutdown)
- TMR6_postscaled (high causes shutdown)
- TMR4_postscaled (high causes shutdown)
- TMR2_postscaled (high causes shutdown)
- Pin selected by CWGxPPS (low causes shutdown)

The shutdown input sources are individually enabled by the **ASyE** bits in CWGxAS1 register.



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

25.11.1.3 Pin Override Levels

The levels driven to the CWG outputs during an auto-shutdown event are controlled by the **LSBD** and **LSAC** bits. The **LSBD** bits control CWGxB/D output levels, while the **LSAC** bits control the CWGxA/C output levels.

25.11.1.4 Auto-Shutdown Interrupts

When an auto-shutdown event occurs, either by software or hardware setting SHUTDOWN, the CWGxIF flag bit of the PIR7 register is set.

Related Links

[7.7.18 PIR7](#)

25.11.2 Auto-Shutdown Restart

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

- Software controlled
- Auto-restart

In either case, the shutdown source must be cleared before the restart can take place. That is, either the shutdown condition must be removed, or the corresponding ASyE bit must be cleared.

25.11.2.1 Software-Controlled Restart

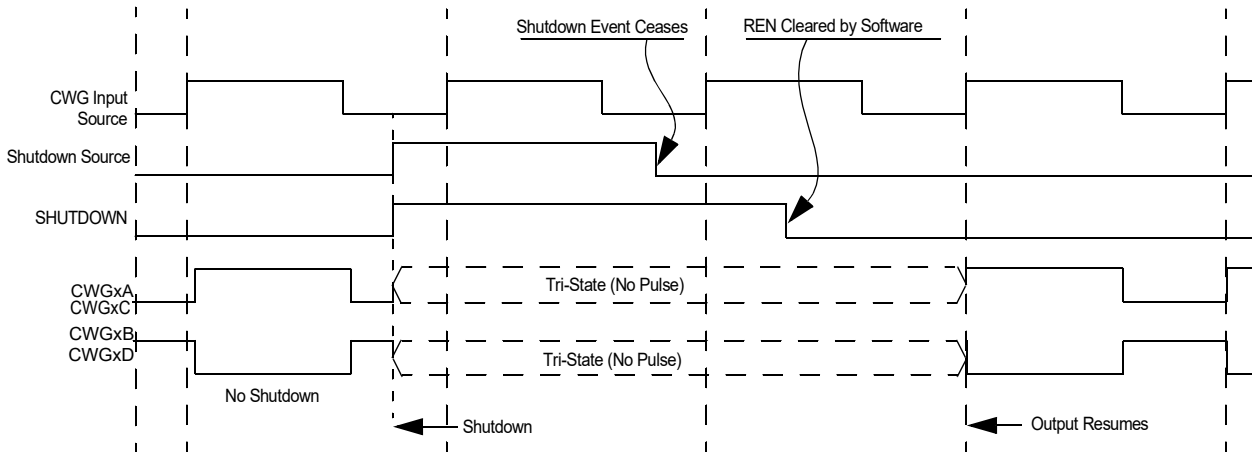
When the REN bit is clear ($REN = 0$), the CWG module must be restarted after an auto-shutdown event through software.

Once all auto-shutdown sources are removed, the software must clear SHUTDOWN. Once SHUTDOWN is cleared, the CWG module will resume operation upon the first rising edge of the CWG data input.



Important: The SHUTDOWN bit cannot be cleared in software if the auto-shutdown condition is still present.

Figure 25-15. Shutdown Functionality, Auto-Restart Disabled ($REN = 0$, $LSAC = 01$, $LSBD = 01$)



25.11.2.2 Auto-Restart

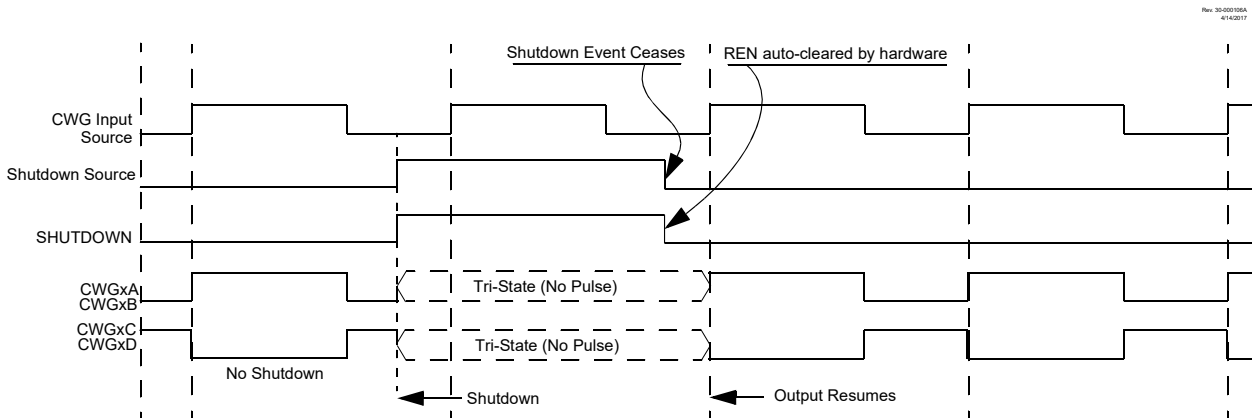
When the REN bit is set ($REN = 1$), the CWG module will restart from the shutdown state automatically.

Once all auto-shutdown conditions are removed, the hardware will automatically clear SHUTDOWN. Once SHUTDOWN is cleared, the CWG module will resume operation upon the first rising edge of the CWG data input.



Important: The SHUTDOWN bit cannot be cleared in software if the auto-shutdown condition is still present.

Figure 25-16. Shutdown Functionality, Auto-Restart Enabled (REN = 1, LSAC = 01, LSBD = 01)



25.12 Operation During Sleep

The CWG module operates independently from the system clock and will continue to run during Sleep, provided that the clock and input sources selected remain active.

The HFINTOSC remains active during Sleep when all the following conditions are met:

- CWG module is enabled
- Input source is active
- HFINTOSC is selected as the clock source, regardless of the system clock source selected.

In other words, if the HFINTOSC is simultaneously selected as the system clock and the CWG clock source, when the CWG is enabled and the input source is active, then the CPU will go idle during Sleep, but the HFINTOSC will remain active and the CWG will continue to operate. This will have a direct effect on the Sleep mode current.

25.13 Configuring the CWG

1. Ensure that the TRIS control bits corresponding to CWG outputs are set so that all are configured as inputs, ensuring that the outputs are inactive during setup. External hardware should ensure that pin levels are held to safe levels.
2. Clear the **EN** bit, if not already cleared.
3. Configure the **MODE** bits to set the output operating mode.
4. Configure the **POLy** bits to set the output polarities.
5. Configure the **ISM** bits to select the data input source.
6. If a steering mode is selected, configure the **STRy** bits to select the desired output on the CWG outputs.
7. Configure the **LSBD** and **LSAC** bits to select the auto-shutdown output override states (this is necessary even if not using auto-shutdown because start-up will be from a shutdown state).
8. If auto-restart is desired, set the **REN** bit.
9. If auto-shutdown is desired, configure the **ASyE** bits to select the shutdown source.
10. Set the desired rising and falling dead-band times with the **CWGxDBR** and **CWGxDBF** registers.
11. Select the clock source with the **CS** bits.
12. Set the **EN** bit to enable the module.
13. Clear the TRIS bits that correspond to the CWG outputs to set them as outputs.

If auto-restart is to be used, set the **REN** bit and the **SHUTDOWN** bit will be cleared automatically. Otherwise, clear the **SHUTDOWN** bit in software to start the CWG.

25.14 Register Summary - CWG Control

| Address | Name | Bit Pos. | | | | | | | | |
|-------------|----------|----------|----------|------|-----------|------|-----------|-----------|------|------|
| 0x00 ... | Reserved | | | | | | | | | |
| 0x060B | | | | | | | | | | |
| 0x060C | CWG1CLK | 7:0 | | | | | | | | CS |
| 0x060D | CWG1ISM | 7:0 | | | | | | ISM[3:0] | | |
| 0x060E | CWG1DBR | 7:0 | | | | | | DBR[5:0] | | |
| 0x060F | CWG1DBF | 7:0 | | | | | | DBF[5:0] | | |
| 0x0610 | CWG1CON0 | 7:0 | EN | LD | | | | MODE[2:0] | | |
| 0x0611 | CWG1CON1 | 7:0 | | | IN | | POLD | POLC | POLB | POLA |
| 0x0612 | CWG1AS0 | 7:0 | SHUTDOWN | REN | LSBD[1:0] | | LSAC[1:0] | | | |
| 0x0613 | CWG1AS1 | 7:0 | | | AS5E | AS4E | AS3E | AS2E | AS1E | AS0E |
| 0x0614 | CWG1STR | 7:0 | OVRD | OVRC | OVRB | OVRA | STRD | STRC | STRB | STRA |
| 0x0615 | Reserved | | | | | | | | | |
| 0x0616 | CWG2CLK | 7:0 | | | | | | | | CS |
| 0x0617 | CWG2ISM | 7:0 | | | | | | ISM[3:0] | | |
| 0x0618 | CWG2DBR | 7:0 | | | | | | DBR[5:0] | | |
| 0x0619 | CWG2DBF | 7:0 | | | | | | DBF[5:0] | | |
| 0x061A | CWG2CON0 | 7:0 | EN | LD | | | | MODE[2:0] | | |
| 0x061B | CWG2CON1 | 7:0 | | | IN | | POLD | POLC | POLB | POLA |
| 0x061C | CWG2AS0 | 7:0 | SHUTDOWN | REN | LSBD[1:0] | | LSAC[1:0] | | | |
| 0x061D | CWG2AS1 | 7:0 | | | AS5E | AS4E | AS3E | AS2E | AS1E | AS0E |
| 0x061E | CWG2STR | 7:0 | OVRD | OVRC | OVRB | OVRA | STRD | STRC | STRB | STRA |

25.15 Register Definitions: CWG Control

Long bit name prefixes for the CWG peripherals are shown in the table below. Refer to the “*Long Bit Names Section*” for more information.

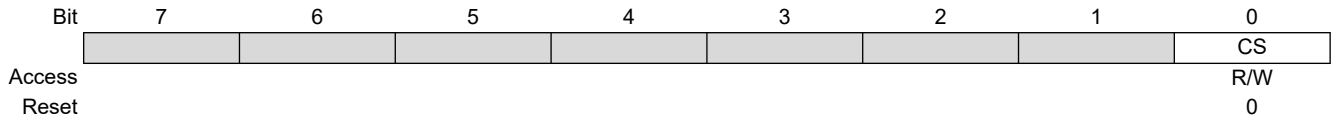
Table 25-1. CWG Bit Name Prefixes

| Peripheral | Bit Name Prefix |
|------------|-----------------|
| CWG1 | CWG1 |
| CWG2 | CWG2 |

25.15.1 CWGxCLK

Name: CWGxCLK
Address: 0x60C,0x616

CWGx Clock Input Selection Register



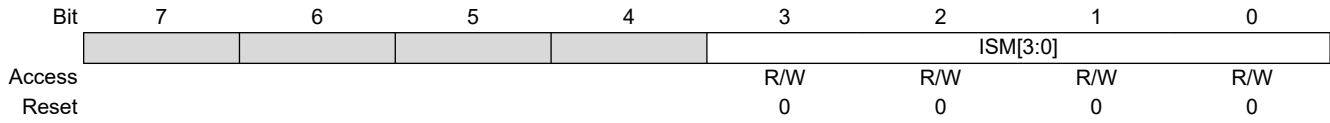
Bit 0 – CS Clock Source
 CWG Clock Source Selection Select bits

| Value | Description |
|-------|---|
| 1 | HFINTOSC (remains operating during Sleep) |
| 0 | Fosc |

25.15.2 CWGxISM

Name: CWGxISM
Address: 0x60D,0x617

CWGx Input Selection Register



Bits 3:0 – ISM[3:0] CWG Data Input Source Select bits

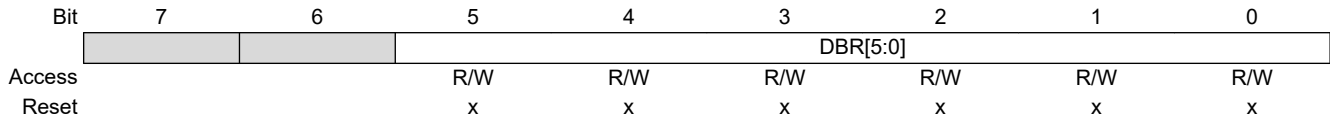
Table 25-2. CWG Data Input Sources

| ISM | Data Source |
|------|---------------------------|
| 1111 | Reserved |
| 1110 | CLC4_out |
| 1101 | CLC3_out |
| 1100 | CLC2_out |
| 1011 | CLC1_out |
| 1010 | DSM1_out |
| 1001 | C2_out |
| 1000 | C1_out |
| 0111 | NCO1_out |
| 0110 | PWM7_out |
| 0101 | PWM6_out |
| 0100 | CCP4_out |
| 0011 | CCP3_out |
| 0010 | CCP2_out |
| 0001 | CCP1_out |
| 0000 | Pin selected by CWGxINPPS |

25.15.3 CWGxDBR

Name: CWGxDBR
Address: 0x60E,0x618

CWG Rising Dead-Band Count Register



Bits 5:0 – DBR[5:0] CWG Rising Edge-Triggered Dead-Band Count bits

Reset States: POR/BOR = xxxxxx

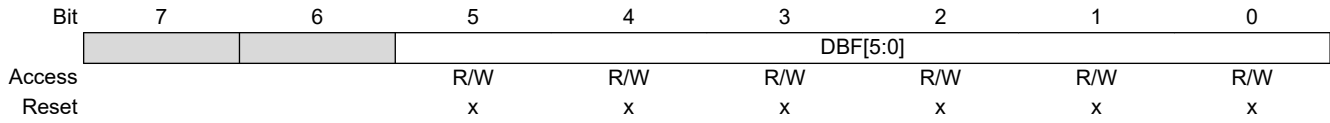
All Other Resets = uuuuuu

| Value | Description |
|-------|---|
| n | Dead band is active no less than n, and no more than n+1, CWG clock periods after the rising edge |
| 0 | 0 CWG clock periods. Dead-band generation is bypassed |

25.15.4 CWGxDBF

Name: CWGxDBF
Address: 0x60F,0x619

CWG Falling Dead-Band Count Register



Bits 5:0 – DBF[5:0] CWG Falling Edge-Triggered Dead-Band Count bits

Reset States: POR/BOR = xxxxxx

All Other Resets = uuuuuu

| Value | Description |
|-------|--|
| n | Dead band is active no less than n, and no more than n+1, CWG clock periods after the falling edge |
| 0 | 0 CWG clock periods. Dead-band generation is bypassed |

25.15.5 CWGxCON0

Name: CWGxCON0
Address: 0x610,0x61A

CWG Control Register 0

| | | | | | | | | |
|--------|-----|--------|---|---|---|-----|-----------|-----|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | EN | LD | | | | | MODE[2:0] | |
| Access | R/W | R/W/HC | | | | R/W | R/W | R/W |
| Reset | 0 | 0 | | | | 0 | 0 | 0 |

Bit 7 – EN CWG1 Enable bit

| Value | Description |
|-------|--------------------|
| 1 | Module is enabled |
| 0 | Module is disabled |

Bit 6 – LD CWG1 Load Buffers bit⁽¹⁾

| Value | Description |
|-------|--|
| 1 | Dead-band count buffers to be loaded on CWG data rising edge, following first falling edge after this bit is set |
| 0 | Buffers remain unchanged |

Bits 2:0 – MODE[2:0] CWG1 Mode bits

| Value | Description |
|-------|---|
| 111 | Reserved |
| 110 | Reserved |
| 101 | CWG outputs operate in Push-Pull mode |
| 100 | CWG outputs operate in Half-Bridge mode |
| 011 | CWG outputs operate in Reverse Full-Bridge mode |
| 010 | CWG outputs operate in Forward Full-Bridge mode |
| 001 | CWG outputs operate in Synchronous Steering mode |
| 000 | CWG outputs operate in Asynchronous Steering mode |

Note:

- This bit can only be set after EN = 1; it cannot be set in the same cycle when EN is set.

25.15.6 CWGxCON1

Name: CWGxCON1
Address: 0x611,0x61B

CWG Control Register 1

| | | | | | | | | |
|--------|---|---|----|---|------|------|------|------|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | | IN | | POLD | POLC | POLB | POLA |
| Access | | | RO | | R/W | R/W | R/W | R/W |
| Reset | | | x | | 0 | 0 | 0 | 0 |

Bit 5 – IN CWG Input Value bit (read-only)

| Value | Description |
|-------|------------------------|
| 1 | CWG input is a logic 1 |
| 0 | CWG input is a logic 0 |

Bits 0, 1, 2, 3 – POLy CWG Output 'y' Polarity bit

| Value | Description |
|-------|------------------------------------|
| 1 | Signal output is inverted polarity |
| 0 | Signal output is normal polarity |

25.15.7 CWGxAS0

Name: CWGxAS0
Address: 0x612,0x61C

CWG Auto-Shutdown Control Register 0

| | | | | | | | | | |
|--------|-----|-----------|-----|-----------|-----|-----------|-----|---|---|
| | Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | SHUTDOWN | REN | LSBD[1:0] | | LSAC[1:0] | | | |
| Access | | R/W/HS/HC | R/W | R/W | R/W | R/W | R/W | | |
| Reset | | 0 | 0 | 0 | 1 | 0 | 1 | | |

Bit 7 – SHUTDOWN Auto-Shutdown Event Status bit^(1,2)

| Value | Description |
|-------|-------------------------------------|
| 1 | An auto-shutdown state is in effect |
| 0 | No auto-shutdown event has occurred |

Bit 6 – REN Auto-Restart Enable bit

| Value | Description |
|-------|--------------------------|
| 1 | Auto-restart is enabled |
| 0 | Auto-restart is disabled |

Bits 5:4 – LSBD[1:0] CWGxB and CWGxD Auto-Shutdown State Control bits

| Value | Description |
|-------|---|
| 11 | A logic '1' is placed on CWGxB/D when an auto-shutdown event occurs. |
| 10 | A logic '0' is placed on CWGxB/D when an auto-shutdown event occurs. |
| 01 | Pin is tri-stated on CWGxB/D when an auto-shutdown event occurs. |
| 00 | The Inactive state of the pin, including polarity, is placed on CWGxB/D after the required dead-band interval when an auto-shutdown event occurs. |

Bits 3:2 – LSAC[1:0] CWGxA and CWGxC Auto-Shutdown State Control bits

| Value | Description |
|-------|---|
| 11 | A logic '1' is placed on CWGxA/C when an auto-shutdown event occurs. |
| 10 | A logic '0' is placed on CWGxA/C when an auto-shutdown event occurs. |
| 01 | Pin is tri-stated on CWGxA/C when an auto-shutdown event occurs. |
| 00 | The Inactive state of the pin, including polarity, is placed on CWGxA/C after the required dead-band interval when an auto-shutdown event occurs. |

Note:

1. This bit may be written while EN = 0 ([25.15.5 CWGxCON0](#)), to place the outputs into the shutdown configuration.
2. The outputs will remain in Auto-shutdown state until the next rising edge of the CWG data input after this bit is cleared.

25.15.8 CWGxAS1

Name: CWGxAS1
Address: 0x613,0x61D

CWG Auto-Shutdown Control Register 1

| | | | | | | | | |
|--------|---|---|------|------|------|------|------|------|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit | | | AS5E | AS4E | AS3E | AS2E | AS1E | AS0E |
| Access | | | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 |

Bits 0, 1, 2, 3, 4, 5 – ASyE CWG Auto-shutdown Source ASyE Enable bit(1)

Table 25-3. Shutdown Sources

| ASyE | Source |
|------|---|
| AS6E | CLC2_out/CLC3_out (low causes shutdown) |
| AS5E | CMP2_out (low causes shutdown) |
| AS4E | CMP1_out (low causes shutdown) |
| AS3E | TMR6_postscaled (high causes shutdown) |
| AS2E | TMR4_postscaled (high causes shutdown) |
| AS1E | TMR2_postscaled (high causes shutdown) |
| AS0E | Pin selected by CWGxPPS (low causes shutdown) |

| Value | Description |
|-------|---|
| 1 | Auto-shutdown for source ASyE is enabled |
| 0 | Auto-shutdown for source ASyE is disabled |

Note: 1. This bit may be written while EN = 0 ([25.15.5 CWGxCON0](#)), to place the outputs into the shutdown configuration.

The outputs will remain in auto-shutdown state until the next rising edge of the CWG data input after this bit is cleared.

25.15.9 CWGxSTR

Name: CWGxSTR
Address: 0x614,0x61E

CWG Steering Control Register ⁽¹⁾

| | | | | | | | | |
|--------|------|------|------|------|------|------|------|------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | OVRD | OVRC | OVRB | OVRA | STRD | STRC | STRB | STRA |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bits 4, 5, 6, 7 – OVRy Steering Data OVR'y' bit

| Value | Condition | Description |
|-------|-----------------------|--|
| x | STRy = 1 | CWGx'y' output has the CWG data input waveform with polarity control from POLy bit |
| 1 | STRy = 0 and POLy = x | CWGx'y' output is high |
| 0 | STRy = 0 and POLy = x | CWGx'y' output is low |

Bits 0, 1, 2, 3 – STRy STR'y' Steering Enable bit⁽²⁾

| Value | Description |
|-------|--|
| 1 | CWGx'y' output has the CWG data input waveform with polarity control from POLy bit |
| 0 | CWGx'y' output is assigned to value of OVRy bit |

Note:

1. The bits in this register apply only when MODE = '00x' (25.15.5 CWGxCON0, Steering modes).
2. This bit is double-buffered when MODE = '001'.

26. NCO - Numerically Controlled Oscillator

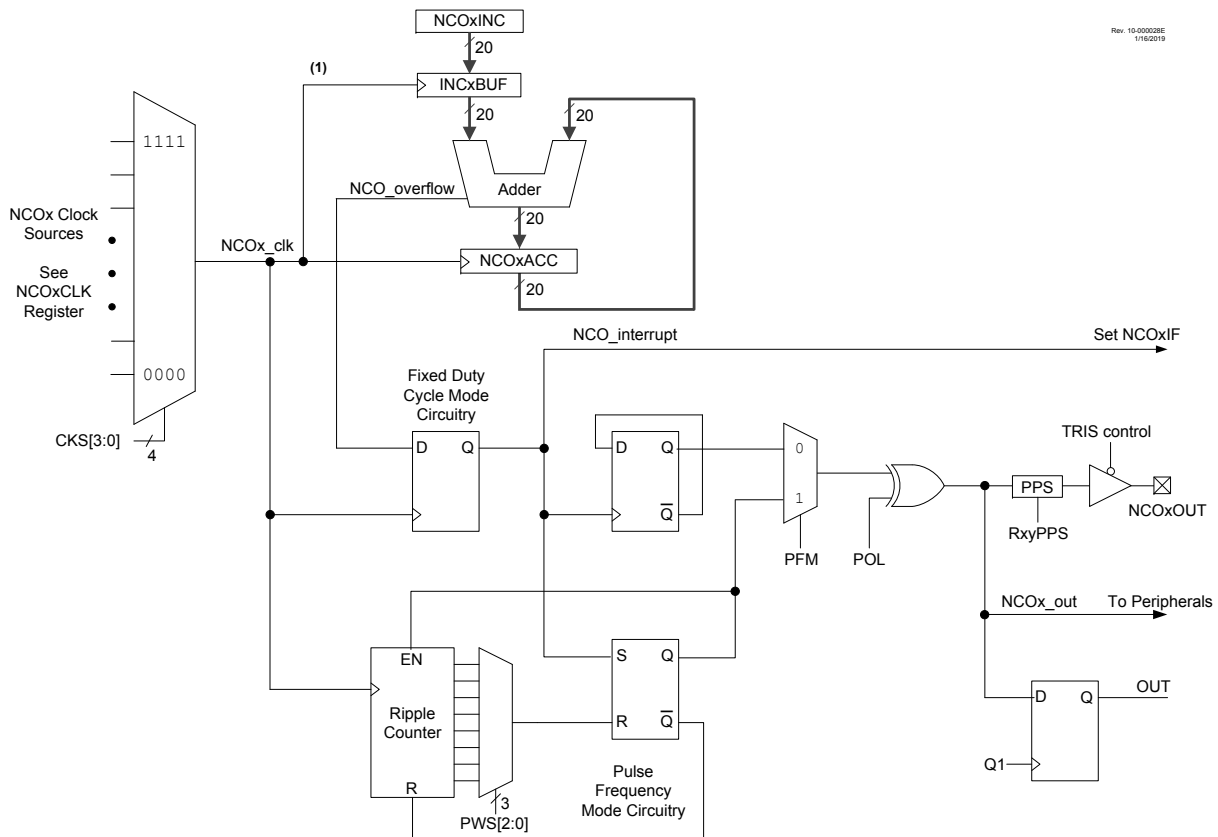
The Numerically Controlled Oscillator (NCO) module is a timer that uses overflow from the addition of an increment value to divide the input frequency. The advantage of the addition method over simple counter driven timer is that the output frequency resolution does not vary with the divider value. The NCO is most useful for application that requires frequency accuracy and fine resolution at a fixed duty cycle.

Features of the NCO include:

- 20-bit Increment Function
- Fixed Duty Cycle mode (FDC) Mode
- Pulse Frequency (PF) Mode
- Output Pulse Width Control
- Multiple Clock Input Sources
- Output polarity Control
- Interrupt Capability

Figure 26-1 is a simplified block diagram of the NCO module.

Figure 26-1. Numerically Controlled Oscillator Module Simplified Block Diagram



Note 1: The increment registers are double-buffered to allow for value changes to be made without first disabling the NCO module. The full increment value is loaded into the buffer registers on the second rising edge of the NCOx_clk signal that occurs immediately after a write to the NCOxINCL register. The buffers are not user-accessible and are shown here for reference.

26.1 NCO Operation

The NCO operates by repeatedly adding a fixed value to an accumulator. Additions occur at the input clock rate. The accumulator will overflow with a carry periodically, which is the raw NCO output (NCO_overflow). This effectively reduces the input clock by the ratio of the addition value to the maximum accumulator value. Refer to [Equation 26-1](#) for details.

The NCO output can be further modified by stretching the pulse or toggling a flip-flop. The modified NCO output is then distributed internally to other peripherals and can optionally be output to a pin. The accumulator overflow also generates an interrupt (NCO_interrupt). The NCO period changes in discrete steps to create an average frequency.

Equation 26-1. NCO Overflow Frequency

$$F_{OVERFLOW} = \frac{NCO\ Clock\ Frequency \times Increment\ Value}{2^{20}}$$

26.1.1 NCO Clock Sources

Clock sources available to the NCO include:

- HFINTOSC
- FOSC
- LC1_out

The NCO clock source is selected by configuring the [CKS](#) bits in the NCOxCLK register.

Related Links

[26.9.4 NCOxCLK](#)

26.1.2 Accumulator

The accumulator is a 20-bit register. Read and write access to the accumulator is available through three registers:

- NCOxACCL
- NCOxACCH
- NCOxACCU

Related Links

[26.9.1 NCOxACC](#)

26.1.3 Adder

The NCO Adder is a full adder, which operates independently from the system clock. The addition of the previous result and the increment value replaces the accumulator value on the rising edge of each input clock.

26.1.4 Increment Registers

The increment value is stored in three registers making up a 20-bit incrementer. In order of LSB to MSB they are:

- NCOxINCL
- NCOxINCH
- NCOxINCU

When the NCO module is enabled, the NCOxINCU and NCOxINCH registers should be written first, then the NCOxINCL register. Writing to the NCOxINCL register initiates the increment buffer registers to be loaded simultaneously on the second rising edge of the NCO_clk signal.

The registers are readable and writable. The increment registers are double-buffered to allow value changes to be made without first disabling the NCO module.

When the NCO module is disabled, the increment buffers are loaded immediately after a write to the increment registers.



Important: The increment buffer registers are not user-accessible.

Related Links

[26.9.2 NCOXINC](#)

26.2 Fixed Duty Cycle Mode

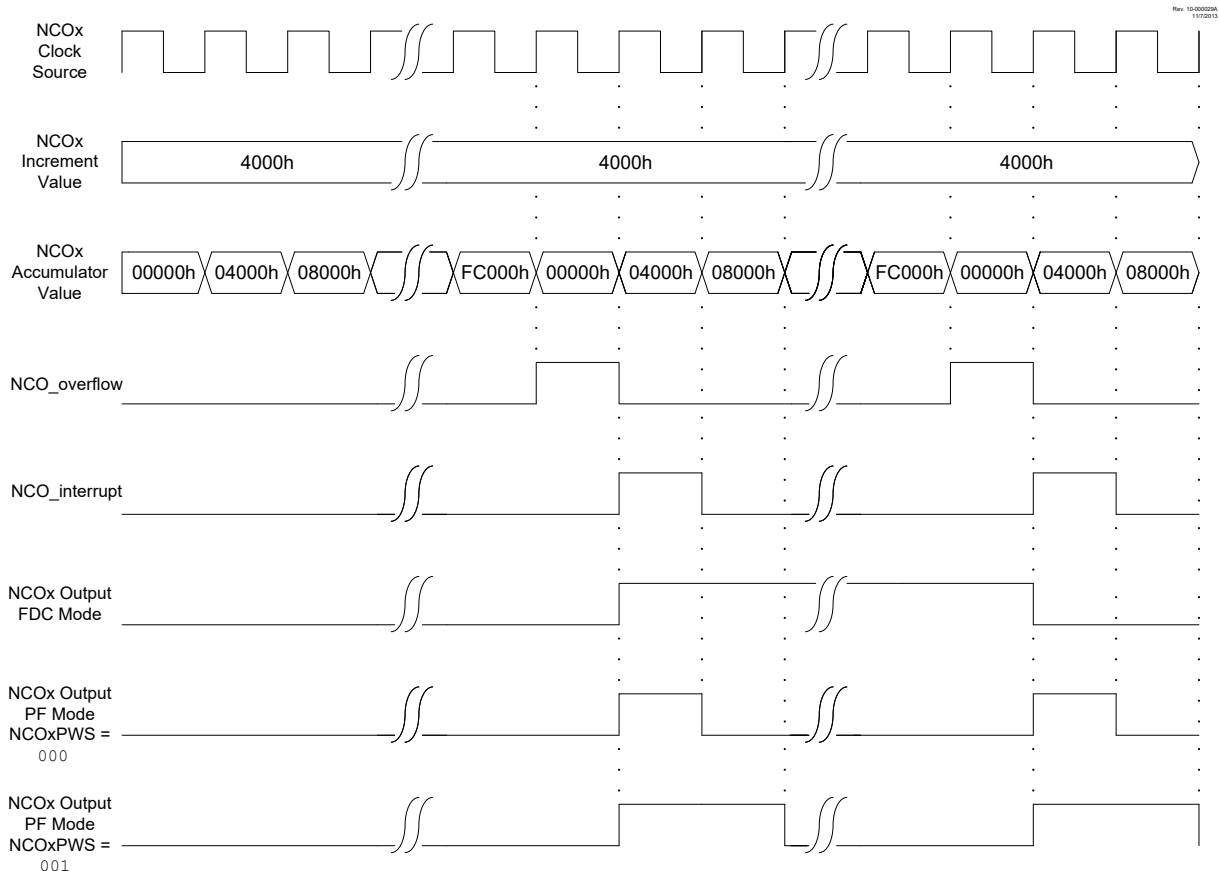
In Fixed Duty Cycle (FDC) mode, every time the accumulator overflows (NCO_overflow), the output is toggled. This provides a 50% duty cycle with a constant frequency, provided that the increment value remains constant.

The FDC frequency can be calculated using [Equation 26-2](#). The FDC frequency is half of the overflow frequency since it takes two overflow events to generate one FDC clock period. For more information, refer to [Figure 26-2](#).

Equation 26-2. FDC Frequency

$$F_{fdc} = F_{overflow} / 2$$

Figure 26-2. FDC Output Mode Operation Diagram



Related Links

[26.9.3 NCOxCON](#)

26.3 Pulse Frequency Mode

In Pulse Frequency (PF) mode, every time the Accumulator overflows, the output becomes active for one or more clock periods. Once the clock period expires, the output returns to an inactive state. This provides a pulsed output. The output becomes active on the rising clock edge immediately following the overflow event. For more information, refer to [Figure 26-2](#).

The value of the active and inactive states depends on the [POL](#) bit in the NCOxCON register.

The PF mode is selected by setting the [PFM](#) bit in the NCOxCON register.

Related Links

[26.9.3 NCOxCON](#)

26.3.1 Output Pulse Width Control

When operating in PF mode, the active state of the output can vary in width by multiple clock periods. Various pulse widths are selected with the [PWS](#) bits.

When the selected pulse width is greater than the Accumulator overflow time frame, then NCO output does not toggle.

Related Links

[26.9.4 NCOxCLK](#)

26.4 Output Polarity Control

The last stage in the NCO module is the output polarity. The [POL](#) bit in the NCOxCON register selects the output polarity. Changing the polarity while the interrupts are enabled will cause an interrupt for the resulting output transition.

The NCO output can be used internally by source code or other peripherals. Accomplish this by reading the NxOUT (read-only) bit of the NCOxCON register. The NCO output signal (NCOx_out) is available to the following peripherals:

- CLC
- CWG
- Timer1
- Timer2
- CLKR

26.5 Interrupts

When the accumulator overflows (NCO_overflow), the NCO Interrupt Flag bit, NCO1IF, of the PIR7 register is set. To enable the interrupt event (NCO_interrupt), the following bits must be set:

- [EN](#) bit
- NCO1IE bit of the PIE7 register
- PEIE bit of the INTCON register
- GIE bit of the INTCON register

The interrupt must be cleared by software by clearing the NCO1IF bit in the Interrupt Service Routine.

Related Links

[26.9.3 NCOxCON](#)

[5.8.10 INTCON](#)

[7.7.18 PIR7](#)

[7.7.9 PIE7](#)

26.6 Effects of a Reset

All of the NCO registers are cleared to zero as the result of a Reset.

26.7 Operation in Sleep

The NCO module operates independently from the system clock and will continue to run during Sleep, provided that the clock source selected remains active.

The HFINTOSC remains active during Sleep when the NCO module is enabled and the HFINTOSC is selected as the clock source, regardless of the system clock source selected.

In other words, if the HFINTOSC is simultaneously selected as the system clock and the NCO clock source, when the NCO is enabled, the CPU will go Idle during Sleep, but the NCO will continue to operate and the HFINTOSC will remain active.

This will have a direct effect on the Sleep mode current.

26.8 Register Summary - NCO

| Address | Name | Bit Pos. | | | | | | | | |
|-------------|----------|----------|------------|--|-----|-----|----------|--|--|-----|
| 0x00 ... | Reserved | | | | | | | | | |
| 0x058B | | | | | | | | | | |
| 0x058C | NCO1ACC | 7:0 | ACC[7:0] | | | | | | | |
| | | 15:8 | ACC[15:8] | | | | | | | |
| | | 23:16 | ACC[23:16] | | | | | | | |
| 0x058F | NCO1INC | 7:0 | INC[7:0] | | | | | | | |
| | | 15:8 | INC[15:8] | | | | | | | |
| | | 23:16 | INC[19:16] | | | | | | | |
| 0x0592 | NCO1CON | 7:0 | EN | | OUT | POL | | | | PFM |
| 0x0593 | NCO1CLK | 7:0 | PWS[2:0] | | | | CKS[3:0] | | | |

26.9 Register Definitions: NCO

26.9.1 NCOxACC

Name: NCOxACC
Address: 0x058C

NCO Accumulator Register

| | | | | | | | | |
|--------|------------|-----|-----|-----|-----|-----|-----|-----|
| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | ACC[23:16] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| | ACC[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | ACC[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bits 23:0 – ACC[23:0] Accumulated sum of NCO additions

Note:

- The individual bytes in this multi-byte register can be accessed with the following register names:
 - NCOxACCU: Accesses the upper byte ACC[23:16]
 - NCOxACCH: Accesses the high byte ACC[15:8]
 - NCOxACCL: Accesses the low byte ACC[7:0]
- The accumulator spans registers NCOxACCU:NCOxACCH:NCOxACCL. The 24 bits are reserved but not all are used. This register updates in real-time, asynchronously to the CPU; there is no provision to ensure atomic access to this 24-bit space using an 8-bit bus. Writing to this register while the module is operating will produce undefined results.

26.9.2 NCOxINC

Name: NCOxINC
Address: 0x058F

NCO Increment Register

| | | | | | | | | | |
|--------|-----|------------|-----|-----|-----|-----|-----|-----|-----|
| | Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | | INC[19:16] | | | | | | | |
| Access | | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | | 0 | 0 | 0 | 0 |
| | | | | | | | | | |
| | Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| | | INC[15:8] | | | | | | | |
| Access | | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | | | | | | | | |
| | Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | INC[7:0] | | | | | | | |
| Access | | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bits 19:0 – INC[19:0] Value by which the NCOxACC is increased by each NCO clock^(1,2)

Note:

1. The individual bytes in this multi-byte register can be accessed with the following register names:
 - NCOxINC_U: Accesses the upper byte INC[19:16]
 - NCOxINC_H: Accesses the high byte INC[15:8]
 - NCOxINC_L: Accesses the low byte INC[7:0]
2. The logical increment spans NCOxINC_U:NCOxINC_H:NCOxINC_L.
3. NCOxINC is double-buffered as INCBUF
 - INCBUF is updated on the next falling edge of NCOxCLK after writing to NCOxINCL
 - NCOxINC_U and NCOxINC_H should be written prior to writing NCOxINCL

26.9.3 NCOxCON

Name: NCOxCON

Address: 0x0592

NCO Control Register

| | | | | | | | | |
|--------|-----|---|-----|-----|---|---|---|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | EN | | OUT | POL | | | | PFM |
| Access | R/W | | RO | R/W | | | | R/W |
| Reset | 0 | | 0 | 0 | | | | 0 |

Bit 7 – EN NCO Enable bit

| Value | Description |
|-------|------------------------|
| 1 | NCO module is enabled |
| 0 | NCO module is disabled |

Bit 5 – OUT NCO Output bit

Displays the current output value of the NCO module.

Bit 4 – POL NCO Polarity bit

| Value | Description |
|-------|-----------------------------------|
| 1 | NCO output signal is inverted |
| 0 | NCO output signal is not inverted |

Bit 0 – PFM NCO Pulse Frequency Mode bit

| Value | Description |
|-------|--|
| 1 | NCO operates in Pulse Frequency mode |
| 0 | NCO operates in Fixed Duty Cycle mode, divide by 2 |

26.9.4 NCOxCLK

Name: NCOxCLK

Address: 0x0593

NCO Input Clock Control Register

| | | | | | | | | |
|--------|----------|-----|-----|---|----------|-----|-----|-----|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | PWS[2:0] | | | | CKS[3:0] | | | |
| Access | R/W | R/W | R/W | | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | | 0 | 0 | 0 | 0 |

Bits 7:5 – PWS[2:0] NCO Output Pulse Width Select bits(1)

| Value | Description |
|-------|--|
| 111 | NCO output is active for 128 input clock periods |
| 110 | NCO output is active for 64 input clock periods |
| 101 | NCO output is active for 32 input clock periods |
| 100 | NCO output is active for 16 input clock periods |
| 011 | NCO output is active for 8 input clock periods |
| 010 | NCO output is active for 4 input clock periods |
| 001 | NCO output is active for 2 input clock periods |
| 000 | NCO output is active for 1 input clock periods |

Bits 3:0 – CKS[3:0] NCO Clock Source Select bits

| Description | Clock Source |
|--------------------------------------|--------------------|
| Table 26-1. NCO Clock Sources | |
| CKS | Clock Source |
| 1111-1011 | Reserved |
| 1010 | CLC4_out |
| 1001 | CLC3_out |
| 1000 | CLC2_out |
| 0111 | CLC1_out |
| 0110 | CLKR |
| 0101 | SOSC |
| 0100 | MFINTOSC (32 kHz) |
| 0011 | MFINTOSC (500 kHz) |
| 0010 | LFINTOSC |
| 0001 | HFINTOSC |
| 0000 | F _{Osc} |

Note:

1. PWS applies only when operating in Pulse Frequency mode.

27. DSM - Data Signal Modulator Module

The Data Signal Modulator (DSM) is a peripheral that allows the user to mix a data stream, also known as a modulator signal, with a carrier signal to produce a modulated output.

Both the carrier and the modulator signals are supplied to the DSM module either internally, from the output of a peripheral, or externally through an input pin.

The modulated output signal is generated by performing a logical “AND” operation of both the carrier and modulator signals and then provided to the MDOUT pin.

The carrier signal is comprised of two distinct and separate signals. A Carrier High (CARH) signal and a Carrier Low (CARL) signal. During the time in which the modulator (MOD) signal is in a logic high state, the DSM mixes the CARH signal with the modulator signal. When the modulator signal is in a logic low state, the DSM mixes the CARL signal with the modulator signal.

Using this method, the DSM can generate the following types of key modulation schemes:

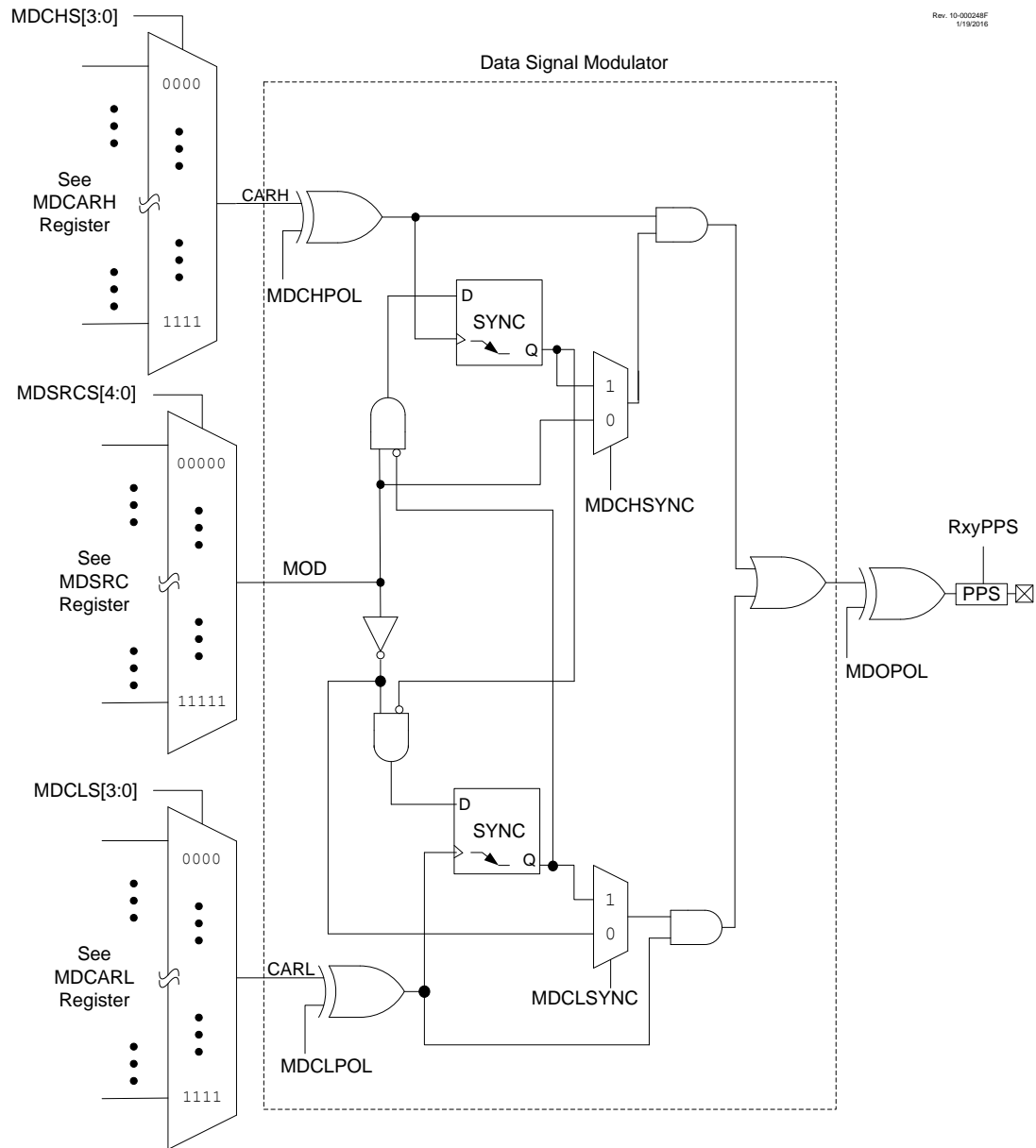
- Frequency-Shift Keying (FSK)
- Phase-Shift Keying (PSK)
- On-Off Keying (OOK)

Additionally, the following features are provided within the DSM module:

- Carrier Synchronization
- Carrier Source Polarity Select
- Programmable Modulator Data
- Modulated Output Polarity Select
- Peripheral Module Disable, which provides the ability to place the DSM module in the lowest power consumption mode

The figure below shows a simplified block diagram of the data signal modulator peripheral.

Figure 27-1. Simplified Block Diagram of the Data Signal Modulator



27.1 DSM Operation

The DSM module can be enabled by setting the **EN** bit in the MDCON0 register. Clearing the EN bit, disables the output of the module but retain the carrier and source signal selections. The module will resume operation when the EN bit is set again. The output of the DSM module can be rerouted to several pins using the RxyPPS register. When the EN bit is cleared the output pin is held low.

27.2 Modulator Signal Sources

The modulator signal can be supplied from the following sources:

- EUSART

- MSSP
- CLC
- NCO
- CCP/PWM
- NCO
- internal MDBIT
- I/O pin

The sources are selected using the **SRCS** bits in the MDxSRC register.

27.3 Carrier Signal Sources

The carrier high signal and carrier low signal can be supplied from the following sources:

- CCPs
- CLCs
- PWMs
- NCO
- Reference clocks
- System clocks
- I/O pin

The carrier high signal is selected by using the **CHS** bits in the MDxCARH register.

The carrier low signal is selected by using the **CLS** bits in the MDxCARL register.

27.4 Carrier Synchronization

During the time when the DSM switches between carrier high and carrier low signal sources, the carrier data in the modulated output signal can become truncated. To prevent this, the carrier signal can be synchronized to the modulator signal. When synchronization is enabled, the carrier pulse that is being mixed at the time of the transition is allowed to transition low before the DSM switches over to the next carrier source.

Synchronization is enabled separately for the carrier high and carrier low signal sources. Synchronization for the carrier high signal is enabled by setting the **CHSYNC** bit. Synchronization for the carrier low signal is enabled by setting the **CLSYNC** bit.

The figures below show the timing diagrams of using various synchronization methods.

Figure 27-2. On Off Keying (OOK) Synchronization

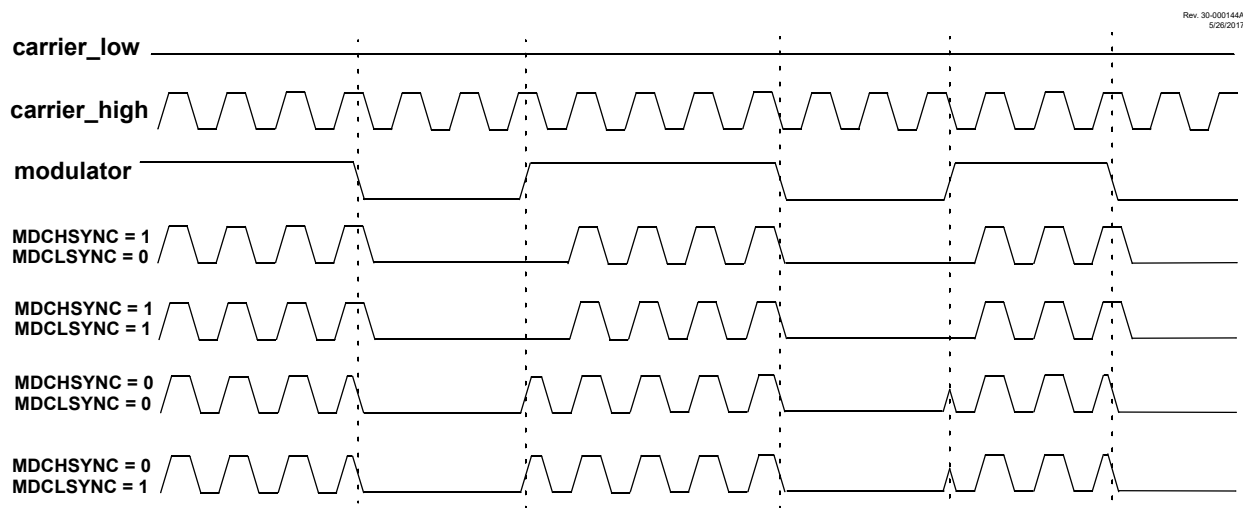


Figure 27-3. No Synchronization (MDCHSYNC = 0, MDCLSYNC = 0)

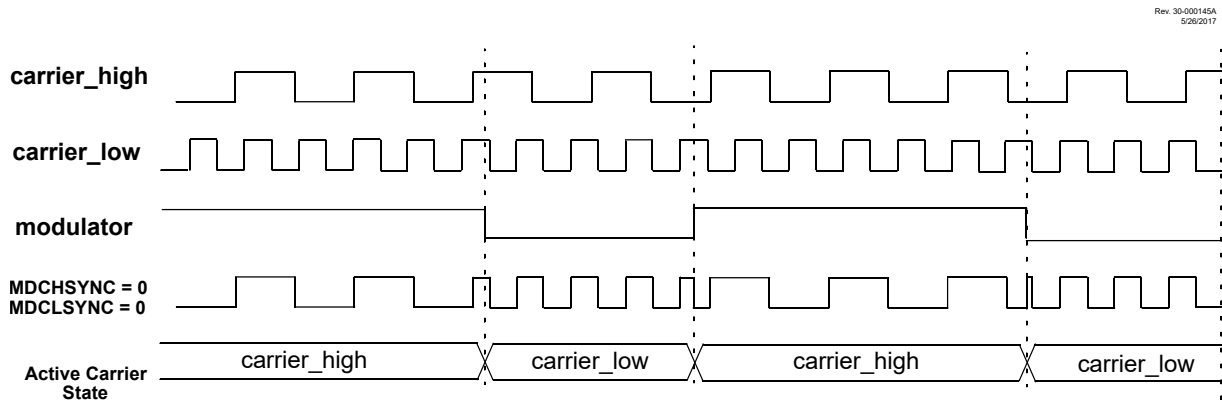


Figure 27-4. Carrier High Synchronization (MDCHSYNC = 1, MDCLSYNC = 0)

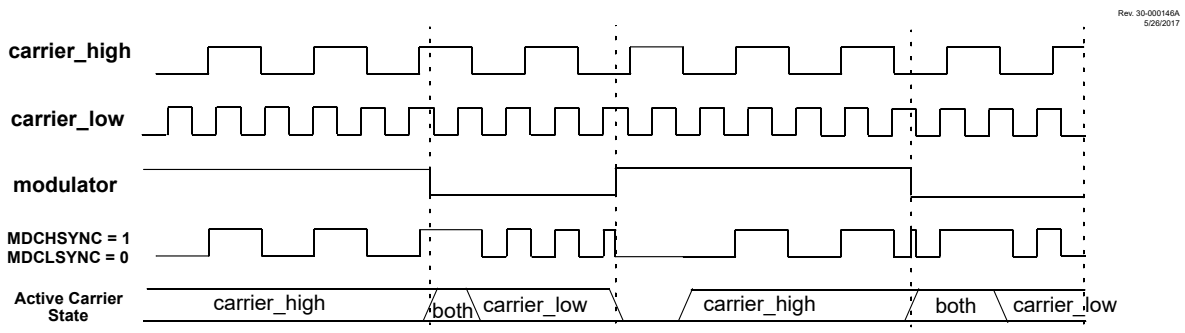


Figure 27-5. Carrier Low Synchronization (MDCHSYNC = 0, MDCLSYNC = 1)

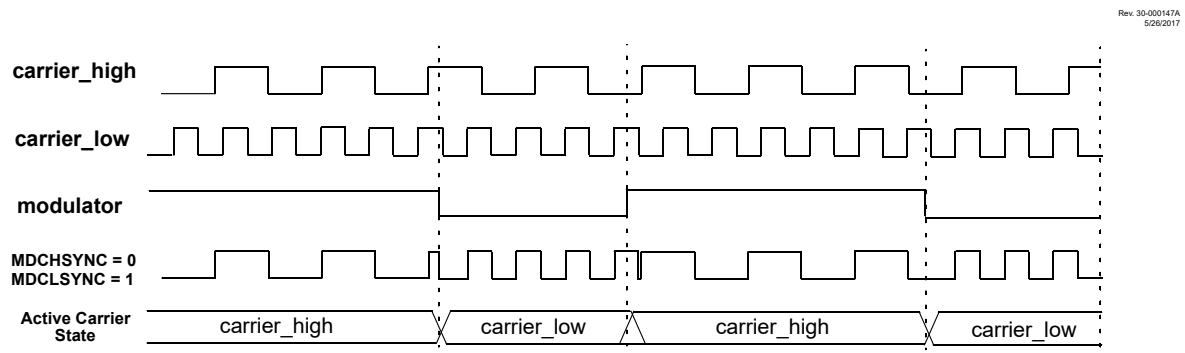
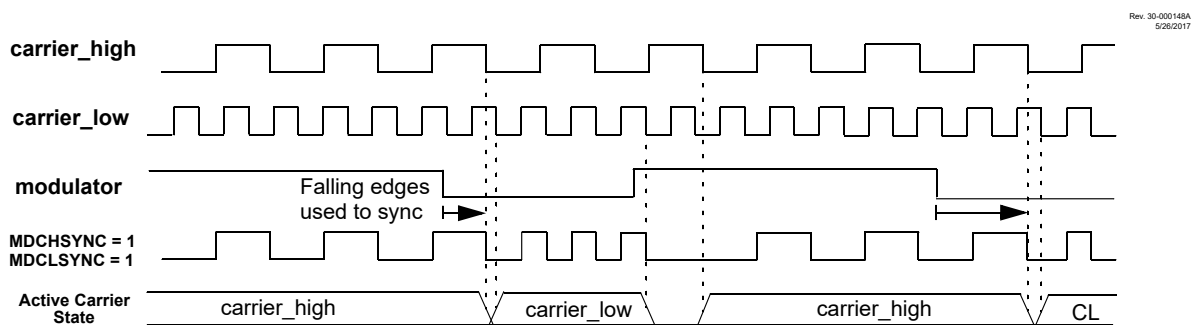


Figure 27-6. Full Synchronization (MDCHSYNC = 1, MDCLSYNC = 1)



27.5 Carrier Source Polarity Select

The signal provided from any selected input source for the carrier high and carrier low signals can be inverted. Inverting the signal for the carrier high and low source is enabled by setting the [CHPOL](#) bit and the [CLPOL](#) bit, respectively.

27.6 Programmable Modulator Data

The [BIT](#) control bit can be selected as the modulation source. This gives the user the ability to provide software driven modulation.

27.7 Modulated Output Polarity

The modulated output signal provided on the DSM pin can also be inverted. Inverting the modulated output signal is enabled by setting the [OPOL](#) bit.

27.8 Operation in Sleep Mode

The DSM can still operate during Sleep, if the carrier and modulator input sources are also still operable during Sleep. Refer to “*Power-Saving Operation Modes*” for more details.

Related Links

[12. Power-Saving Operation Modes](#)

27.9 Effects of a Reset

Upon any device Reset, the DSM module is disabled. The user's firmware is responsible for initializing the module before enabling the output. All the registers are reset to their default values.

27.10 Peripheral Module Disable

The DSM module can be completely disabled using the PMD module to achieve maximum power saving. When the DSM1MD bit of PMD7 register is set, the DSM module is completely disabled. This puts the module in its lowest Power Consumption state. When enabled again all the registers of the DSM module default to POR status.

Related Links

[13.5 Register Definitions: Peripheral Module Disable](#)

27.11 Register Summary - DSM

| Address | Name | Bit Pos. | | | | | | | | |
|-------------|-------------------------|----------|----|--|-------|--------|--|-----------|-------|--------|
| 0x00 ... | Reserved | | | | | | | | | |
| 0x0896 | | | | | | | | | | |
| 0x0897 | MD1CON0 | 7:0 | EN | | OUT | OPOL | | | | BIT |
| 0x0898 | MD1CON1 | 7:0 | | | CHPOL | CHSYNC | | | CLPOL | CLSYNC |
| 0x0899 | MD1SRC | 7:0 | | | | | | SRCS[4:0] | | |
| 0x089A | MD1CARL | 7:0 | | | | | | CLS[3:0] | | |
| 0x089B | MD1CARH | 7:0 | | | | | | CHS[3:0] | | |

27.12 Register Definitions: Modulation Control

27.12.1 MDxCON0

Name: MDxCON0
Address: 0x0897

Modulation Control Register 0

| | | | | | | | | |
|--------|-----|---|-----|------|---|---|---|-----|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | EN | | OUT | OPOL | | | | BIT |
| Access | R/W | | R/W | R/W | | | | R/W |
| Reset | 0 | | 0 | 0 | | | | 0 |

Bit 7 – EN Modulator Module Enable bit

| Value | Description |
|-------|--|
| 1 | Modulator module is enabled and mixing input signals |
| 0 | Modulator module is disabled and has no output |

Bit 5 – OUT Modulator Output bit

Displays the current output value of the modulator module.

Bit 4 – OPOL Modulator Output Polarity Select bit

| Value | Description |
|-------|--|
| 1 | Modulator output signal is inverted; idle high output |
| 0 | Modulator output signal is not inverted; idle low output |

Bit 0 – BIT Modulation Source Select Input bit

Allows software to manually set modulation source input to module

Note:

1. The modulated output frequency can be greater and asynchronous from the clock that updates this register bit, the bit value may not be valid for higher speed modulator or carrier signals.
2. MDBIT must be selected as the modulation source in the MDxSRC register for this operation.

27.12.2 MDxCON1

Name: MDxCON1
Address: 0x0898

Modulation Control Register 1

| | | | | | | | | |
|--------|---|---|-------|--------|---|---|-------|--------|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | | CHPOL | CHSYNC | | | CLPOL | CLSYNC |
| Access | | | R/W | R/W | | | R/W | R/W |
| Reset | | | 0 | 0 | | | 0 | 0 |

Bit 5 – CHPOL Modulator High Carrier Polarity Select bit

| Value | Description |
|-------|--|
| 1 | Selected high carrier signal is inverted |
| 0 | Selected high carrier signal is not inverted |

Bit 4 – CHSYNC Modulator High Carrier Synchronization Enable bit

| Value | Description |
|-------|---|
| 1 | Modulator waits for a falling edge on the high time carrier signal before allowing a switch to the low time carrier |
| 0 | Modulator output is not synchronized to the high time carrier signal |

Bit 1 – CLPOL Modulator Low Carrier Polarity Select bit

| Value | Description |
|-------|---|
| 1 | Selected low carrier signal is inverted |
| 0 | Selected low carrier signal is not inverted |

Bit 0 – CLSYNC Modulator Low Carrier Synchronization Enable bit

| Value | Description |
|-------|---|
| 1 | Modulator waits for a falling edge on the low time carrier signal before allowing a switch to the high time carrier |
| 0 | Modulator output is not synchronized to the low time carrier signal |

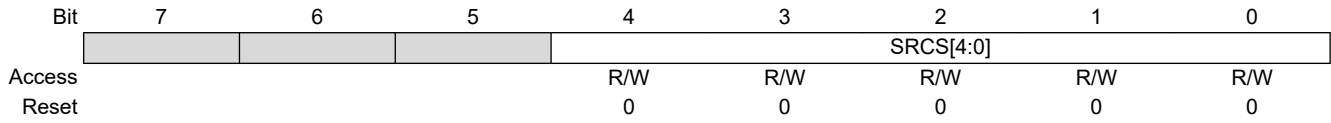
Note:

1. Narrowed carrier pulse widths or spurs may occur in the signal stream if the carrier is not synchronized.

27.12.3 MDxSRC

Name: MDxSRC
Address: 0x0899

Modulation Source Control Register



Bits 4:0 – SRCS[4:0] Modulator Source Selection bits

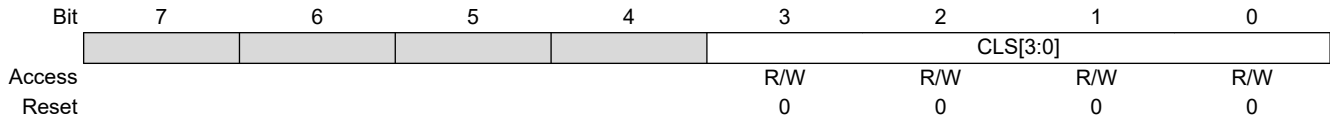
Table 27-1. MDSRC Selection MUX Connections

| SRCS[4:0] | Connection |
|-------------|---------------------------|
| 11111-10010 | Reserved |
| 10001 | MSSP2 - SDO |
| 10000 | MSSP1 - SDO |
| 01111 | EUSART1 TX (TX/CK output) |
| 01110 | CLC4 OUT |
| 01101 | CLC3 OUT |
| 01100 | CLC2 OUT |
| 01011 | CLC1 OUT |
| 01010 | C2 OUT |
| 01001 | C1 OUT |
| 01000 | NCO1 OUT |
| 00111 | PWM7 OUT |
| 00110 | PWM6 OUT |
| 00101 | CCP4 OUT |
| 00100 | CCP3 OUT |
| 00011 | CCP2 OUT |
| 00010 | CCP1 OUT |
| 00001 | MDBIT |
| 00000 | Pin selected by MDSRCPPS |

27.12.4 MDxCARL

Name: MDxCARL
Address: 0x089A

Modulation Low Carrier Control Register



Bits 3:0 – CLS[3:0] Modulator Carrier Low Input Selection bits

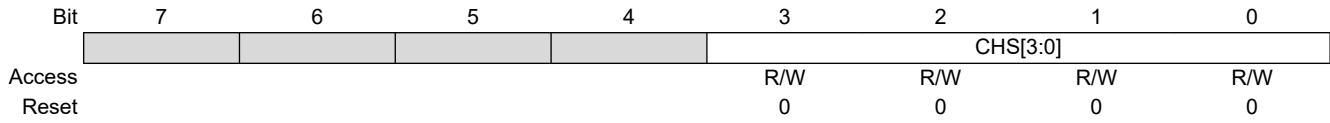
Table 27-2. MDCARL Source Selections

| MDCARL | |
|----------|---------------------------------|
| CLS[3:0] | Connection |
| 1111 | Reserved |
| 1110 | CLC4 OUT |
| 1101 | CLC3 OUT |
| 1100 | CLC2 OUT |
| 1011 | CLC1 OUT |
| 1010 | NCO1 OUT |
| 1001 | PWM7 OUT |
| 1000 | PWM6 OUT |
| 0111 | CCP4 OUT |
| 0110 | CCP3 OUT |
| 0101 | CCP2 OUT |
| 0100 | CCP1 OUT |
| 0011 | CLKREF output |
| 0010 | HFINTOSC |
| 0001 | F _{OSC} (system clock) |
| 0000 | Pin selected by MDCARLPPS |

27.12.5 MDxCARH

Name: MDxCARH
Address: 0x089B

Modulation High Carrier Control Register



Bits 3:0 – CHS[3:0] Modulator Carrier High Selection bits

Table 27-3. MDCARH Source Selections

| MDCARH | |
|----------|---------------------------------|
| CHS[3:0] | Connection |
| 1111 | Reserved |
| 1110 | CLC4 OUT |
| 1101 | CLC3 OUT |
| 1100 | CLC2 OUT |
| 1011 | CLC1 OUT |
| 1010 | NCO1 OUT |
| 1001 | PWM7 OUT |
| 1000 | PWM6 OUT |
| 0111 | CCP4 OUT |
| 0110 | CCP3 OUT |
| 0101 | CCP2 OUT |
| 0100 | CCP1 OUT |
| 0011 | CLKREF output |
| 0010 | HFINTOSC |
| 0001 | F _{OSC} (system clock) |
| 0000 | Pin selected by MDCARHPPS |

28. EUSART - Enhanced Universal Synchronous Asynchronous Receiver Transmitter

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 28-1](#) and [Figure 28-2](#).

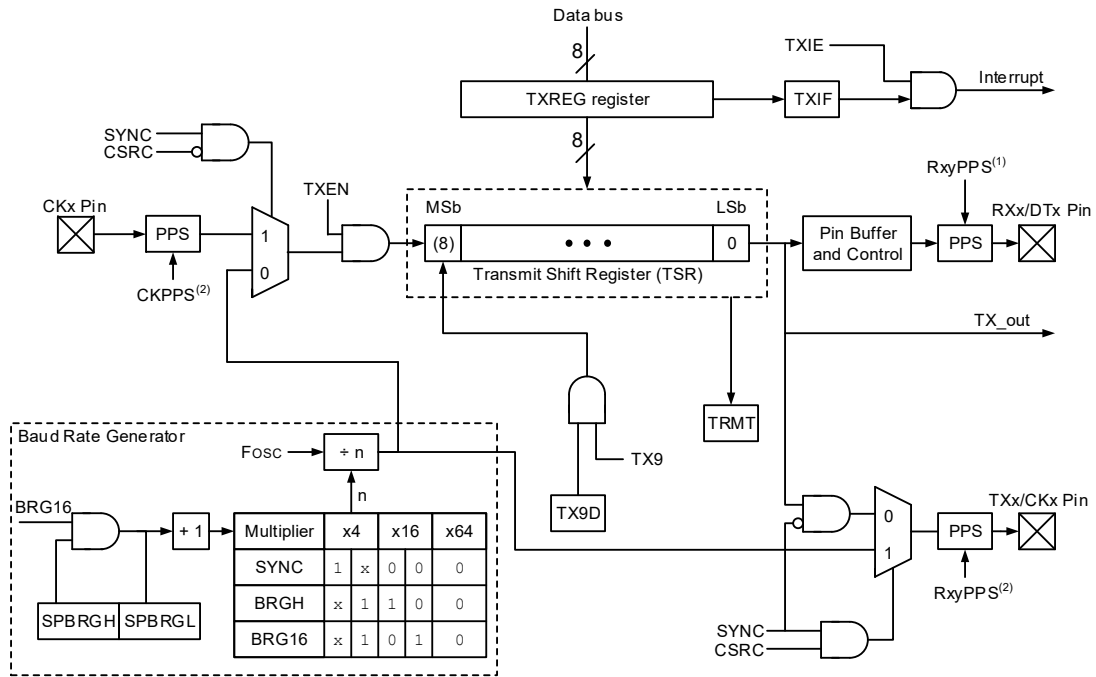
The operation of the EUSART module consists of six registers:

- Transmit Status and Control ([28.6.5 TXxSTA](#))
- Receive Status and Control ([28.6.4 RCxSTA](#))
- Baud Rate Control ([28.6.6 BAUDxCON](#))
- Baud Rate Value ([28.6.3 SPxBRG](#))
- Receive Data Register ([28.6.1 RCxREG](#))
- Transmit Data Register ([28.6.2 TXxREG](#))

The RXx/DTx and TXx/CKx input pins are selected with the RXxPPS and TXxPPS registers, respectively. TXx, CKx, and DTx output pins are selected with each pin's RxyPPS register. Since the RX input is coupled with the DT output in Synchronous mode, it is the user's responsibility to select the same pin for both of these functions when operating in Synchronous mode. The EUSART control logic will control the data direction drivers automatically.

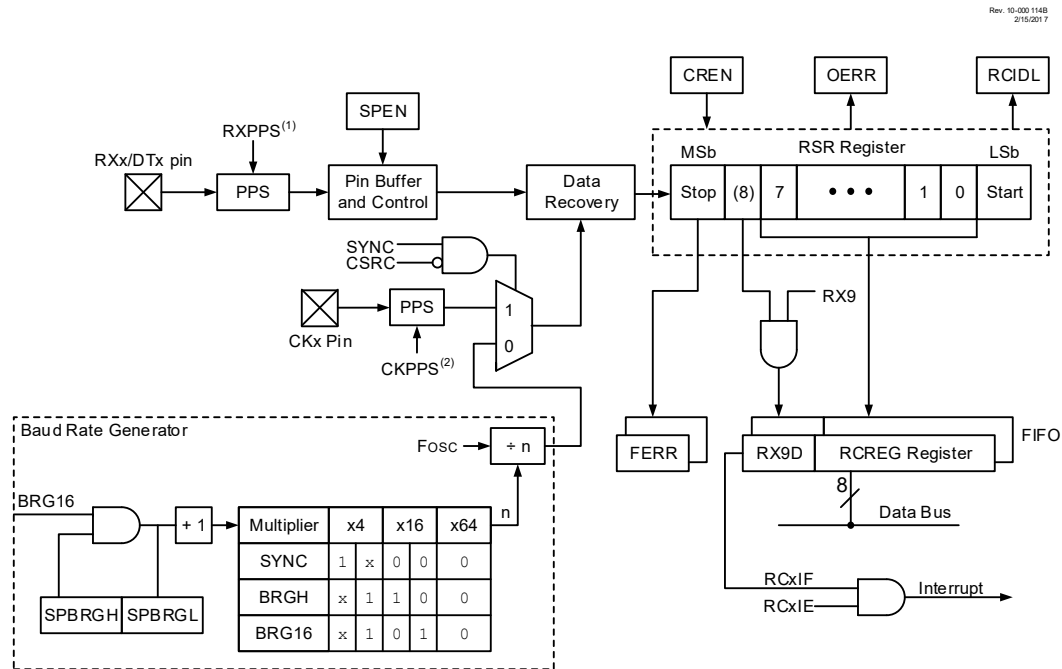
Figure 28-1. EUSART Transmit Block Diagram

Rev. 10-00-113C
2/15/2017



- Note 1:** In Synchronous mode, the DT output and RX input PPS selections should enable the same pin.
2: In Master Synchronous mode the TX output and CK input PPS selections should enable the same pin.

Figure 28-2. EUSART Receive Block Diagram



Note 1: In Synchronous mode, the DT output and RX input PPS selections should enable the same pin.
Note 2: In Master Synchronous mode the TX output and CK input PPS selections should enable the same pin.

28.1 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 V_{OH} Mark state which represents a '1' data bit, and a V_{OL} 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 eight bits. Each transmitted bit persists for a period of $1/(\text{Baud Rate})$. 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 28-2 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.

28.1.1 EUSART Asynchronous Transmitter

Figure 28-1 is a simplified representation of the transmitter. 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 TXxREG register.

28.1.1.1 Enabling the Transmitter

The EUSART transmitter is enabled for asynchronous operations by configuring the following three control bits:

- TXEN bit in the TXxSTA register is set to '1' to enable the transmitter circuitry of the EUSART
- SYNC bit in the TXxSTA register is set to '0' to configure the EUSART for asynchronous operation
- SPEN bit in the RCxSTA register is set to '1' to enable the EUSART interface and to enable automatically the output drivers for the RxyPPS selected as the TXx/CKx output

All other EUSART control bits are assumed to be in their default state.

If the TXx/CKx pin is shared with an analog peripheral, the analog I/O function must be disabled by clearing the corresponding ANSEL bit.



Important: The TXxIF Transmitter Interrupt flag in the PIR3 register is set when the TXEN enable bit in the PIE3 register is set and the Transmit Shift register (TSR) is Idle.

28.1.1.2 Transmitting Data

A transmission is initiated by writing a character to the TXxREG register. If this is the first character, or the previous character has been completely flushed from the TSR, the data in the TXxREG 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 TXxREG until the Stop bit of the previous character has been transmitted. The pending character in the TXxREG is then transferred to the TSR in one T_{CY} 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 TXxREG.

28.1.1.3 Transmit Data Polarity

The polarity of the transmit data can be controlled with the SCKP bit of the BAUDxCON register. The default state of this bit is '0' which selects high true transmit idle and data bits. Setting the SCKP bit to '1' will invert the transmit data resulting in low true idle and data bits. The SCKP bit controls transmit data polarity in Asynchronous mode only. In Synchronous mode, the SCKP bit has a different function. See the [28.3.1.2 Clock Polarity](#) section for more details.

28.1.1.4 Transmit Interrupt Flag

The TXxIF interrupt flag bit of the PIR3 register is set whenever the EUSART transmitter is enabled and no character is being held for transmission in the TXxREG. In other words, the TXxIF bit is only cleared when the TSR is busy with a character and a new character has been queued for transmission in the TXxREG. The TXxIF flag bit is not cleared immediately upon writing TXxREG. TXxIF becomes valid in the second instruction cycle following the write execution. Polling TXxIF immediately following the TXxREG write will return invalid results. The TXxIF bit is read-only, it cannot be set or cleared by software.

The TXxIF interrupt can be enabled by setting the TXxIE interrupt enable bit of the PIE3 register. However, the TXxIF flag bit will be set whenever the TXxREG is empty, regardless of the state of TXxIE enable bit.

To use interrupts when transmitting data, set the TXxIE bit only when there is more data to send. Clear the TXxIE interrupt enable bit upon writing the last character of the transmission to the TXxREG.

28.1.1.5 TSR Status

The TRMT bit of the TXxSTA 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 TXxREG. 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 needs to poll this bit to determine the TSR status.



Important: The TSR register is not mapped in data memory, so it is not available to the user.

28.1.1.6 Transmitting 9-Bit Characters

The EUSART supports 9-bit character transmissions. When the TX9 bit of the TXxSTA register is set, the EUSART will shift nine bits out for each character transmitted. The TX9D bit of the TXxSTA register is the ninth, and Most Significant data bit. When transmitting 9-bit data, the TX9D data bit must be written before writing the eight Least Significant bits into the TXxREG. All nine bits of data will be transferred to the TSR shift register immediately after the TXxREG is written.

A special 9-bit Address mode is available for use with multiple receivers. See the [28.1.2.7 Address Detection](#) section for more information on the Address mode.

28.1.1.7 Asynchronous Transmission Setup

1. Initialize the SPxBRGH, SPxBRGL register pair and the BRGH and BRG16 bits to achieve the desired baud rate (see section 28.2 EUSART Baud Rate Generator (BRG)).
2. Select the transmit output pin by writing the appropriate value to the RxyPPS register.
3. Enable the asynchronous serial port by clearing the SYNC bit and setting the SPEN bit.
4. If 9-bit transmission is desired, set the TX9 control bit in the TXxSTA register. That will indicate that the eight Least Significant data bits are an address when the receiver is set for address detection.
5. Set SCKP bit if inverted transmit is desired.
6. Enable the transmission by setting the TXEN control bit. This will cause the TXxIF interrupt bit to be set.
7. If interrupts are desired, set the TXxIE interrupt enable bit of the PIE3 register.
8. An interrupt will occur immediately provided that the GIE and PEIE bits of the INTCON register are also set.
9. If 9-bit transmission is selected, the ninth bit should be loaded into the TX9D data bit.
10. Load 8-bit data into the TXxREG register. This will start the transmission.

Figure 28-3. Asynchronous Transmission

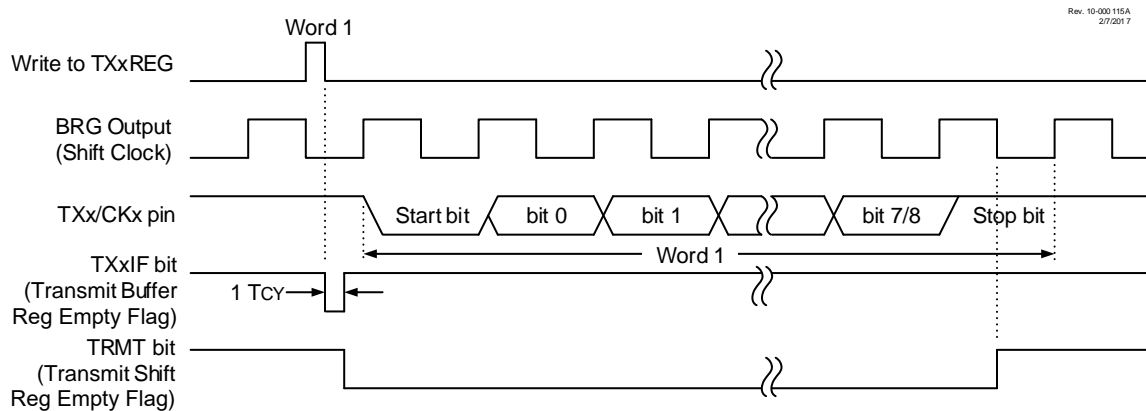
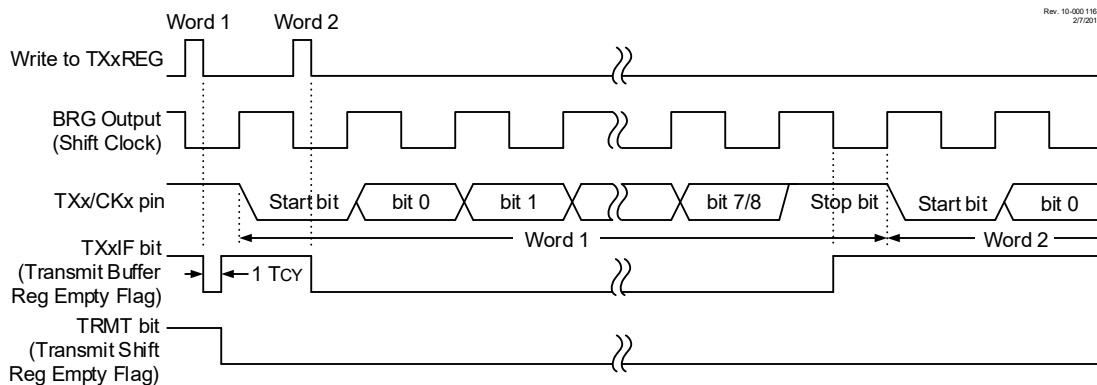


Figure 28-4. Asynchronous Transmission (Back-to-Back)



28.1.2 EUSART Asynchronous Receiver

The Asynchronous mode is typically used in RS-232 systems. A simplified representation of the receiver is shown in the Figure 28-2. The data is received on the RXx/DTx 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 eight or nine 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 RCxREG register.

28.1.2.1 Enabling the Receiver

The EUSART receiver is enabled for asynchronous operation by configuring the following three control bits:

- CREN bit in the RCxSTA register is set to '1' to enables the receiver circuitry of the EUSART
- SYNC bit in the TXxSTA register is set to '0' to configure the EUSART for asynchronous operation
- SPEN bit in the RCxSTA register is set to '1' to enable the EUSART interface

All other EUSART control bits are assumed to be in their default state.

The user must set the RXxPPS register to select the RXx/DTx I/O pin and set the corresponding TRIS bit to configure the pin as an input.



Important: If the RX/DT function is on an analog pin, the corresponding ANSEL bit must be cleared for the receiver to function.

28.1.2.2 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 the [28.1.2.4 Receive Framing Error](#) section 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 RCxIF interrupt flag bit of the PIR3 register is set. The top character in the FIFO is transferred out of the FIFO by reading the RCxREG register.



Important: If the receive FIFO is overrun, no additional characters will be received until the overrun condition is cleared. See the [28.1.2.5 Receive Overrun Error](#) section for more information.

28.1.2.3 Receive Interrupts

The RCxIF interrupt flag bit of the PIR3 register is set whenever the EUSART receiver is enabled and there is an unread character in the receive FIFO. The RCxIF interrupt flag bit is read-only, it cannot be set or cleared by software.

RCxIF interrupts are enabled by setting all of the following bits:

- RCxIE, Interrupt Enable bit of the PIE3 register
- PEIE, Peripheral Interrupt Enable bit of the INTCON register
- GIE, Global Interrupt Enable bit of the INTCON register

The RCxIF interrupt flag bit will be set when there is an unread character in the FIFO, regardless of the state of interrupt enable bits.

28.1.2.4 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 RCxSTA 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 RCxREG.

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 RCxSTA register which resets the EUSART. Clearing the CREN bit of the RCxSTA register does not affect the FERR bit. A framing error by itself does not generate an interrupt.



Important: If all receive characters in the receive FIFO have framing errors, repeated reads of the RCxREG will not clear the FERR bit.

28.1.2.5 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 RCxSTA 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 RCxSTA register or by resetting the EUSART by clearing the SPEN bit of the RCxSTA register.

28.1.2.6 Receiving 9-Bit Characters

The EUSART supports 9-bit character reception. When the RX9 bit of the RCxSTA register is set the EUSART will shift nine bits into the RSR for each character received. The RX9D bit of the RCxSTA 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 eight Least Significant bits from the RCxREG.

28.1.2.7 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 RCxSTA 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 RCxIF 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.

28.1.2.8 Asynchronous Reception Setup

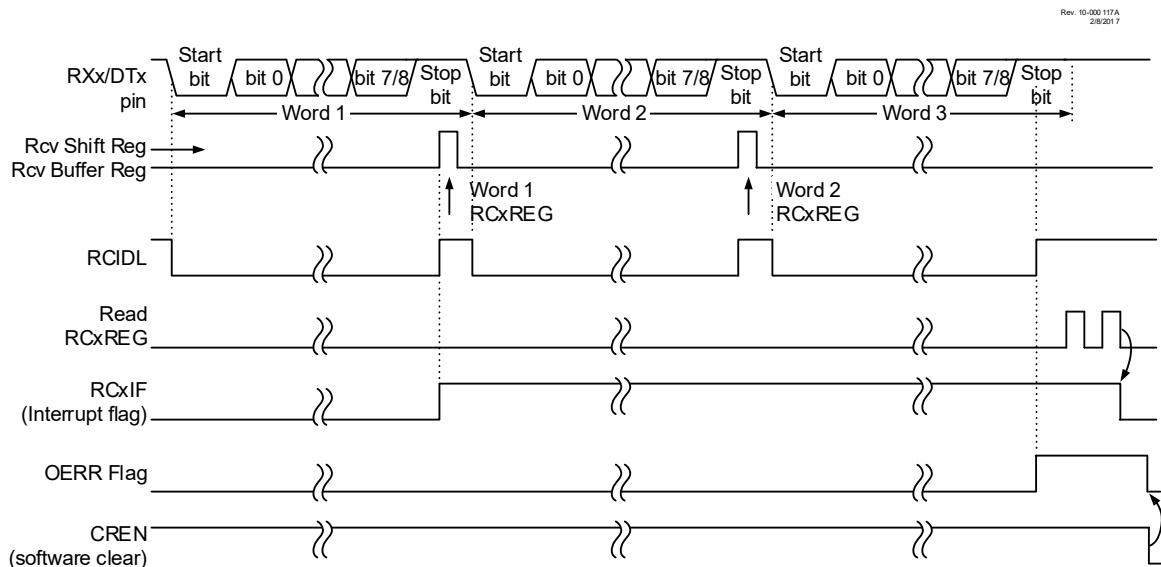
1. Initialize the SPxBRGH:SPxBRGL register pair and the BRGH and BRG16 bits to achieve the desired baud rate (see the [28.2 EUSART Baud Rate Generator \(BRG\)](#) section).
2. Set the RXxPPS register to select the RXx/DTx input pin.
3. Clear the ANSEL bit for the RXx pin (if applicable).
4. Enable the serial port by setting the SPEN bit. The SYNC bit must be clear for asynchronous operation.
5. If interrupts are desired, set the RCxIE bit of the PIE3 register and the GIE and PEIE bits of the INTCON register.
6. If 9-bit reception is desired, set the RX9 bit.
7. Enable reception by setting the CREN bit.
8. The RCxIF 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 RCxIE interrupt enable bit was also set.
9. Read the RCxSTA register to get the error flags and, if 9-bit data reception is enabled, the ninth data bit.
10. Get the received eight Least Significant data bits from the receive buffer by reading the RCxREG register.
11. If an overrun occurred, clear the OERR flag by clearing the CREN receiver enable bit.

28.1.2.9 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 follow these steps:

1. Initialize the SPxBRGH:SPxBRGL register pair and the BRGH and BRG16 bits to achieve the desired baud rate (see the [28.2 EUSART Baud Rate Generator \(BRG\)](#) section).
2. Set the RXxPPS register to select the RXx input pin.
3. Clear the ANSEL bit for the RXx pin (if applicable).
4. Enable the serial port by setting the SPEN bit. The SYNC bit must be cleared for asynchronous operation.
5. If interrupts are desired, set the RCxIE bit of the PIE3 register and the GIE and PEIE bits of the INTCON register.
6. Enable 9-bit reception by setting the RX9 bit.
7. Enable address detection by setting the ADDEN bit.
8. Enable reception by setting the CREN bit.
9. The RCxIF 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 RCxIE Interrupt Enable bit is also set.
10. Read the RCxSTA register to get the error flags. The ninth data bit will always be set.
11. Get the received eight Least Significant data bits from the receive buffer by reading the RCxREG register. Software determines if this is the device's address.
12. If an overrun occurred, clear the OERR flag by clearing the CREN receiver enable bit.
13. If the device has been addressed, clear the ADDEN bit to allow all received data into the receive buffer and generate interrupts.

Figure 28-5. Asynchronous Reception



Note: This timing diagram shows three bytes appearing on the RXx input. The OERR flag is set because the RCxREG is not read before the third word is received.

28.1.3 Clock Accuracy with Asynchronous Operation

The factory calibrates the internal oscillator block output (INTOSC). However, the INTOSC frequency may drift as V_{DD} 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.

The other method adjusts the value in the Baud Rate Generator. This can be done automatically with the Auto-Baud Detect feature (see [28.2.1 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.

28.2 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 BAUDxCON register selects 16-bit mode.

The SPxBRGH, SPxBRGL 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 TXxSTA register and the BRG16 bit of the BAUDxCON register. In Synchronous mode, the BRGH bit is ignored.

Table 28-1 contains the formulas for determining the baud rate. Equation 28-1 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 and are shown in Table 28-2. 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. The BRGH bit is used to achieve very high baud rates.

Writing a new value to the SPxBRGH, SPxBRGL 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.

Equation 28-1. Calculating Baud Rate Error

For a device with Fosc of 16 MHz, desired baud rate of 9600, Asynchronous mode, 8-bit BRG:

$$DesiredBaudrate = \frac{F_{OSC}}{(64 \times SPxBRG) + 1}$$

Solving for SPxBRG:

$$SPxBRG = \frac{F_{OSC}}{64 \times DesiredBaudrate} - 1$$

$$SPxBRG = \frac{16000000}{64 \times 9600} - 1$$

$$SPxBRG = 25.042 \approx 25$$

$$CalculatedBaudrate = \frac{16000000}{64 \times (25 + 1)}$$

$$CalculatedBaudrate = 9615$$

$$Error = \frac{CalculatedBaudrate - DesiredBaudrate}{DesiredBaudrate}$$

$$Error = \frac{9615 - 9600}{9600}$$

$$Error = 0.16\%$$

Table 28-1. Baud Rate Formulas

| Configuration Bits | | | BRG/EUSART Mode | Baud Rate Formula |
|--------------------|-------|------|---------------------|----------------------|
| SYNC | BRG16 | BRGH | | |
| 0 | 0 | 0 | 8-bit/Asynchronous | $F_{OSC}/[64 (n+1)]$ |
| 0 | 0 | 1 | 8-bit/Asynchronous | $F_{OSC}/[16 (n+1)]$ |
| 0 | 1 | 0 | 16-bit/Asynchronous | |

.....continued

| Configuration Bits | | | BRG/EUSART Mode | Baud Rate Formula |
|--------------------|-------|------|---------------------|--------------------|
| SYNC | BRG16 | BRGH | | |
| 0 | 1 | 1 | 16-bit/Asynchronous | $F_{OSC}/[4(n+1)]$ |
| 1 | 0 | x | 8-bit/Synchronous | |
| 1 | 1 | x | 16-bit/Synchronous | |

Note: x = Don't care, n = value of SPxBRGH:SPxBRGL register pair.

Table 28-2. Sample Baud Rates for Asynchronous Modes

| BAUD RATE | SYNC = 0, BRGH = 0, BRG16 = 0 | | | | | | | | | | | |
|-----------|-------------------------------|---------|-----------------------|-------------------------------|---------|-----------------------|-------------------------------|---------|-----------------------|--------------------------------|---------|-----------------------|
| | F _{OSC} = 32.000 MHz | | | F _{OSC} = 20.000 MHz | | | F _{OSC} = 18.432 MHz | | | F _{OSC} = 11.0592 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 |
| 2400 | 2404 | 0.16 | 207 | 2404 | 0.16 | 129 | 2400 | 0.00 | 119 | 2400 | 0.00 | 71 |
| 9600 | 9615 | 0.16 | 51 | 9470 | -1.36 | 32 | 9600 | 0.00 | 29 | 9600 | 0.00 | 17 |
| 10417 | 10417 | 0.00 | 47 | 10417 | 0.00 | 29 | 10286 | -1.26 | 27 | 10165 | -2.42 | 16 |
| 19.2k | 19.23k | 0.16 | 25 | 19.53k | 1.73 | 15 | 19.20k | 0.00 | 14 | 19.20k | 0.00 | 8 |
| 57.6k | 55.55k | -3.55 | 3 | — | — | — | 57.60k | 0.00 | 7 | 57.60k | 0.00 | 2 |
| 115.2k | — | — | — | — | — | — | — | — | — | — | — | — |

| BAUD RATE | SYNC = 0, BRGH = 0, BRG16 = 0 | | | | | | | | | | | |
|-----------|-------------------------------|---------|-----------------------|------------------|---------|-----------------------|-------------------|---------|-----------------------|------------------|---------|-----------------------|
| | Fosc = 8.000 MHz | | | Fosc = 4.000 MHz | | | Fosc = 3.6864 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 | 51 |
| 1200 | 1202 | 0.16 | 103 | 1202 | 0.16 | 51 | 1200 | 0.00 | 47 | 1202 | 0.16 | 12 |
| 2400 | 2404 | 0.16 | 51 | 2404 | 0.16 | 25 | 2400 | 0.00 | 23 | — | — | — |
| 9600 | 9615 | 0.16 | 12 | — | — | — | 9600 | 0.00 | 5 | — | — | — |
| 10417 | 10417 | 0.00 | 11 | 10417 | 0.00 | 5 | — | — | — | — | — | — |
| 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 = 32.000 MHz | | | Fosc = 20.000 MHz | | | Fosc = 18.432 MHz | | | Fosc = 11.0592 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 | — | — | — | — | — | — | — | — | — | — | — | — |
| 9600 | 9615 | 0.16 | 207 | 9615 | 0.16 | 129 | 9600 | 0.00 | 119 | 9600 | 0.00 | 71 |
| 10417 | 10417 | 0.00 | 191 | 10417 | 0.00 | 119 | 10378 | -0.37 | 110 | 10473 | 0.53 | 65 |
| 19.2k | 19.23k | 0.16 | 103 | 19.23k | 0.16 | 64 | 19.20k | 0.00 | 59 | 19.20k | 0.00 | 35 |
| 57.6k | 57.14k | -0.79 | 34 | 56.82k | -1.36 | 21 | 57.60k | 0.00 | 19 | 57.60k | 0.00 | 11 |
| 115.2k | 117.64k | 2.12 | 16 | 113.64k | -1.36 | 10 | 115.2k | 0.00 | 9 | 115.2k | 0.00 | 5 |

| BAUD RATE | SYNC = 0, BRGH = 1, BRG16 = 0 | | | | | | | | | | | |
|-----------|-------------------------------|---------|-----------------------|------------------|---------|-----------------------|-------------------|---------|-----------------------|------------------|---------|-----------------------|
| | Fosc = 8.000 MHz | | | Fosc = 4.000 MHz | | | Fosc = 3.6864 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 | 51 |
| 2400 | 2404 | 0.16 | 207 | 2404 | 0.16 | 103 | 2400 | 0.00 | 95 | 2404 | 0.16 | 25 |
| 9600 | 9615 | 0.16 | 51 | 9615 | 0.16 | 25 | 9600 | 0.00 | 23 | — | — | — |
| 10417 | 10417 | 0.00 | 47 | 10417 | 0.00 | 23 | 10473 | 0.53 | 21 | 10417 | 0.00 | 5 |
| 19.2k | 19231 | 0.16 | 25 | 19.23k | 0.16 | 12 | 19.2k | 0.00 | 11 | — | — | — |
| 57.6k | 55556 | -3.55 | 8 | — | — | — | 57.60k | 0.00 | 3 | — | — | — |
| 115.2k | — | — | — | — | — | — | 115.2k | 0.00 | 1 | — | — | — |

| BAUD RATE | SYNC = 0, BRGH = 0, BRG16 = 1 | | | | | | | | | | | |
|-----------|-------------------------------|---------|-----------------------|-------------------|---------|-----------------------|-------------------|---------|-----------------------|--------------------|---------|-----------------------|
| | Fosc = 32.000 MHz | | | Fosc = 20.000 MHz | | | Fosc = 18.432 MHz | | | Fosc = 11.0592 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 | 6666 | 300.0 | -0.01 | 4166 | 300.0 | 0.00 | 3839 | 300.0 | 0.00 | 2303 |
| 1200 | 1200 | -0.02 | 3332 | 1200 | -0.03 | 1041 | 1200 | 0.00 | 959 | 1200 | 0.00 | 575 |
| 2400 | 2401 | -0.04 | 832 | 2399 | -0.03 | 520 | 2400 | 0.00 | 479 | 2400 | 0.00 | 287 |
| 9600 | 9615 | 0.16 | 207 | 9615 | 0.16 | 129 | 9600 | 0.00 | 119 | 9600 | 0.00 | 71 |
| 10417 | 10417 | 0.00 | 191 | 10417 | 0.00 | 119 | 10378 | -0.37 | 110 | 10473 | 0.53 | 65 |
| 19.2k | 19.23k | 0.16 | 103 | 19.23k | 0.16 | 64 | 19.20k | 0.00 | 59 | 19.20k | 0.00 | 35 |

| | | | | | | | | | | | | |
|--------|--------|-------|----|---------|-------|----|--------|------|----|--------|------|----|
| 57.6k | 57.14k | -0.79 | 34 | 56.818 | -1.36 | 21 | 57.60k | 0.00 | 19 | 57.60k | 0.00 | 11 |
| 115.2k | 117.6k | 2.12 | 16 | 113.636 | -1.36 | 10 | 115.2k | 0.00 | 9 | 115.2k | 0.00 | 5 |

| BAUD RATE | SYNC = 0, BRGH = 0, BRG16 = 1 | | | | | | | | | | | |
|-----------|-------------------------------|---------|-----------------------|------------------|---------|-----------------------|-------------------|---------|-----------------------|------------------|---------|-----------------------|
| | Fosc = 8.000 MHz | | | Fosc = 4.000 MHz | | | Fosc = 3.6864 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 | 299.9 | -0.02 | 1666 | 300.1 | 0.04 | 832 | 300.0 | 0.00 | 767 | 300.5 | 0.16 | 207 |
| 1200 | 1199 | -0.08 | 416 | 1202 | 0.16 | 207 | 1200 | 0.00 | 191 | 1202 | 0.16 | 51 |
| 2400 | 2404 | 0.16 | 207 | 2404 | 0.16 | 103 | 2400 | 0.00 | 95 | 2404 | 0.16 | 25 |
| 9600 | 9615 | 0.16 | 51 | 9615 | 0.16 | 25 | 9600 | 0.00 | 23 | — | — | — |
| 10417 | 10417 | 0.00 | 47 | 10417 | 0.00 | 23 | 10473 | 0.53 | 21 | 10417 | 0.00 | 5 |
| 19.2k | 19.23k | 0.16 | 25 | 19.23k | 0.16 | 12 | 19.20k | 0.00 | 11 | — | — | — |
| 57.6k | 55556 | -3.55 | 8 | — | — | — | 57.60k | 0.00 | 3 | — | — | — |
| 115.2k | — | — | — | — | — | — | 115.2k | 0.00 | 1 | — | — | — |

| BAUD RATE | SYNC = 0, BRGH = 1, BRG16 = 1 or SYNC = 1, BRG16 = 1 | | | | | | | | | | | |
|-----------|--|---------|-----------------------|-------------------|---------|-----------------------|-------------------|---------|-----------------------|--------------------|---------|-----------------------|
| | Fosc = 32.000 MHz | | | Fosc = 20.000 MHz | | | Fosc = 18.432 MHz | | | Fosc = 11.0592 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 | 26666 | 300.0 | 0.00 | 16665 | 300.0 | 0.00 | 15359 | 300.0 | 0.00 | 9215 |
| 1200 | 1200 | 0.00 | 6666 | 1200 | -0.01 | 4166 | 1200 | 0.00 | 3839 | 1200 | 0.00 | 2303 |
| 2400 | 2400 | 0.01 | 3332 | 2400 | 0.02 | 2082 | 2400 | 0.00 | 1919 | 2400 | 0.00 | 1151 |
| 9600 | 9604 | 0.04 | 832 | 9597 | -0.03 | 520 | 9600 | 0.00 | 479 | 9600 | 0.00 | 287 |
| 10417 | 10417 | 0.00 | 767 | 10417 | 0.00 | 479 | 10425 | 0.08 | 441 | 10433 | 0.16 | 264 |
| 19.2k | 19.18k | -0.08 | 416 | 19.23k | 0.16 | 259 | 19.20k | 0.00 | 239 | 19.20k | 0.00 | 143 |
| 57.6k | 57.55k | -0.08 | 138 | 57.47k | -0.22 | 86 | 57.60k | 0.00 | 79 | 57.60k | 0.00 | 47 |
| 115.2k | 115.9k | 0.64 | 68 | 116.3k | 0.94 | 42 | 115.2k | 0.00 | 39 | 115.2k | 0.00 | 23 |

| BAUD RATE | SYNC = 0, BRGH = 1, BRG16 = 1 or SYNC = 1, BRG16 = 1 | | | | | | | | | | | |
|-----------|--|---------|-----------------------|------------------|---------|-----------------------|-------------------|---------|-----------------------|------------------|---------|-----------------------|
| | Fosc = 8.000 MHz | | | Fosc = 4.000 MHz | | | Fosc = 3.6864 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 | 0.00 | 6666 | 300.0 | 0.01 | 3332 | 300.0 | 0.00 | 3071 | 300.1 | 0.04 | 832 |
| 1200 | 1200 | -0.02 | 1666 | 1200 | 0.04 | 832 | 1200 | 0.00 | 767 | 1202 | 0.16 | 207 |
| 2400 | 2401 | 0.04 | 832 | 2398 | 0.08 | 416 | 2400 | 0.00 | 383 | 2404 | 0.16 | 103 |

| | | | | | | | | | | | | |
|--------|--------|-------|-----|--------|-------|-----|--------|------|----|--------|------|----|
| 9600 | 9615 | 0.16 | 207 | 9615 | 0.16 | 103 | 9600 | 0.00 | 95 | 9615 | 0.16 | 25 |
| 10417 | 10417 | 0 | 191 | 10417 | 0.00 | 95 | 10473 | 0.53 | 87 | 10417 | 0.00 | 23 |
| 19.2k | 19.23k | 0.16 | 103 | 19.23k | 0.16 | 51 | 19.20k | 0.00 | 47 | 19.23k | 0.16 | 12 |
| 57.6k | 57.14k | -0.79 | 34 | 58.82k | 2.12 | 16 | 57.60k | 0.00 | 15 | — | — | — |
| 115.2k | 117.6k | 2.12 | 16 | 111.1k | -3.55 | 8 | 115.2k | 0.00 | 7 | — | — | — |

28.2.1 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 BAUDxCON register starts the auto-baud calibration sequence. 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 SPxBRG begins counting up using the BRG counter clock as shown in Figure 28-6. The fifth rising edge will occur on the RXx pin at the end of the eighth bit period. At that time, an accumulated value totaling the proper BRG period is left in the SPxBRGH, SPxBRGL register pair, the ABDEN bit is automatically cleared and the RCxIF interrupt flag is set. The value in the RCxREG needs to be read to clear the RCxIF interrupt. RCxREG content should be discarded. When calibrating for modes that do not use the SPxBRGH register the user can verify that the SPxBRGL register did not overflow by checking for 00h in the SPxBRGH register.

The BRG auto-baud clock is determined by the BRG16 and BRGH bits as shown in Table 28-3. During ABD, both the SPxBRGH and SPxBRGL registers are used as a 16-bit counter, independent of the BRG16 bit setting. While calibrating the baud rate period, the SPxBRGH and SPxBRGL 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:

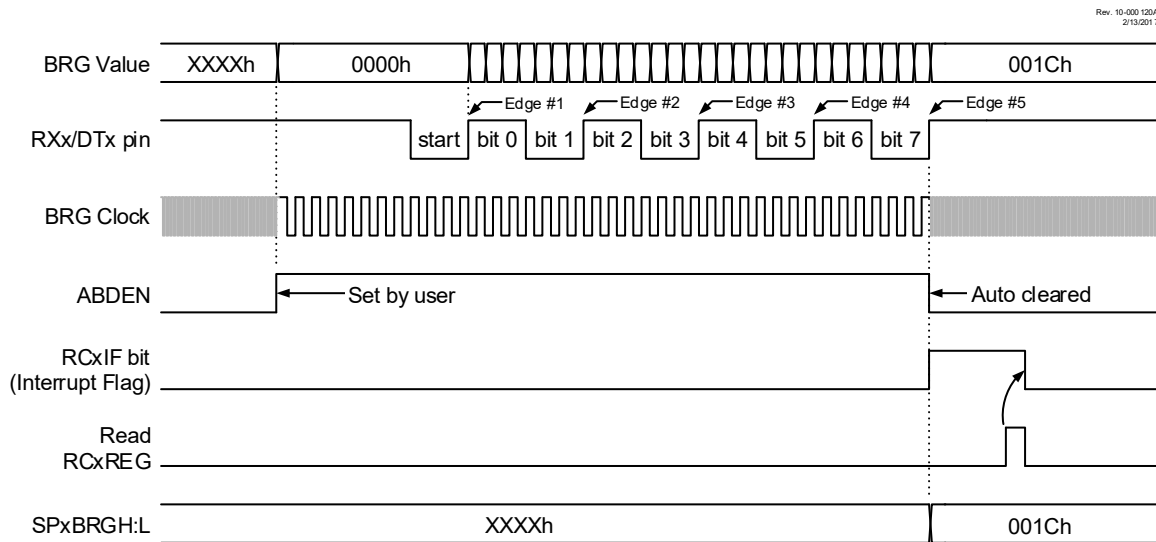
1. If the WUE bit is set with the ABDEN bit, auto-baud detection will occur on the byte following the Break character (see 28.2.3 Auto-Wake-up on Break).
2. It is up to the user to determine that the incoming character baud rate is within the range of the selected BRG clock source. Some combinations of oscillator frequency and EUSART baud rates are not possible.
3. During the auto-baud process, the auto-baud counter starts counting at one. Upon completion of the auto-baud sequence, to achieve maximum accuracy, subtract 1 from the SPxBRGH:SPxBRGL register pair.

Table 28-3. BRG Counter Clock Rates

| BRG16 | BRGH | BRG Base Clock | BRG ABD Clock |
|-------|------|----------------------|-----------------------|
| 1 | 1 | F _{osc} /4 | F _{osc} /32 |
| 1 | 0 | F _{osc} /16 | F _{osc} /128 |
| 0 | 1 | F _{osc} /16 | F _{osc} /128 |
| 0 | 0 | F _{osc} /64 | F _{osc} /512 |

Note: During the ABD sequence, SPxBRGL and SPxBRGH registers are both used as a 16-bit counter, independent of the BRG16 setting.

Figure 28-6. Automatic Baud Rate Calibration



28.2.2 Auto-Baud Overflow

During the course of automatic baud detection, the ABDOVF bit of the BAUDxCON register will be set if the baud rate counter overflows before the fifth rising edge is detected on the RXx pin. The ABDOVF bit indicates that the counter has exceeded the maximum count that can fit in the 16 bits of the SPxBRGH:SPxBRGL register pair. After the ABDOVF bit has been set, the counter continues to count until the fifth rising edge is detected on the RXx pin. Upon detecting the fifth RX edge, the hardware will set the RCxIF interrupt flag and clear the ABDEN bit of the BAUDxCON register. The RCxIF flag can be subsequently cleared by reading the RCxREG register. The ABDOVF flag of the BAUDxCON register can be cleared by software directly.

To terminate the auto-baud process before the RCxIF flag is set, clear the ABDEN bit then clear the ABDOVF bit of the BAUDxCON register. The ABDOVF bit will remain set if the ABDEN bit is not cleared first.

28.2.3 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 BAUDxCON 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 RCxIF interrupt coincident with the wake-up event. The interrupt is generated synchronously to the Q clocks in normal CPU operating modes as shown in Figure 28-7, and asynchronously if the device is in Sleep mode, as shown in Figure 28-8. The interrupt condition is cleared by reading the RCxREG 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.

28.2.3.1 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 ten or more bit times, 13-bit times recommended for LIN bus, or any number of bit times for standard RS-232 devices.

Oscillator Start-up Time

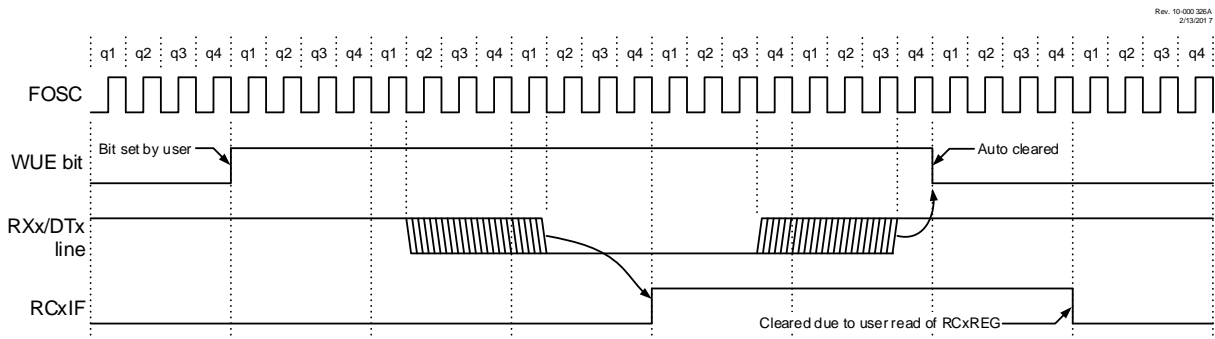
Oscillator start-up time must be considered, especially in applications using oscillators with longer start-up intervals (i.e., LP, XT or HS/PLL mode). The Sync Break (or wake-up signal) character must be of sufficient length, and be followed by a sufficient interval, to allow enough time for the selected oscillator to start and provide proper initialization of the EUSART.

WUE Bit

The wake-up event causes a receive interrupt by setting the RCxIF 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 RCxREG register and discarding its contents.

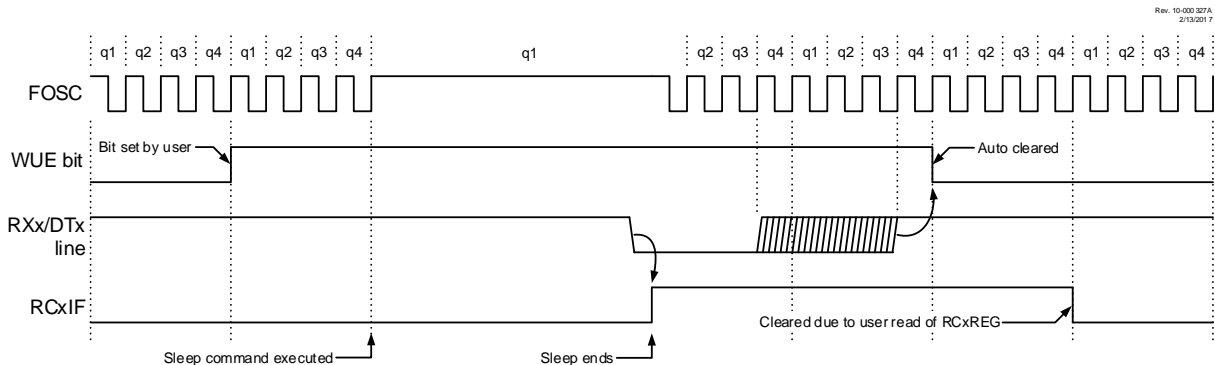
To ensure that no actual data is lost, check the RCIDL bit to verify that a receive operation is not in process before setting the WUE bit. If a receive operation is not occurring, the WUE bit may then be set just prior to entering the Sleep mode.

Figure 28-7. Auto-Wake-up Bit (WUE) Timing During Normal Operation



Note 1: The EUSART remains in Idle while the WUE bit is set.

Figure 28-8. Auto-Wake-up Bit (WUE) Timings During Sleep



Note 1: The EUSART remains in Idle while the WUE bit is set.

28.2.4 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 TXxSTA register. The Break character transmission is then initiated by a write to the TXxREG. The value of data written to TXxREG 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 TXxSTA register indicates when the transmit operation is active or idle, just as it does during normal transmission. See [Figure 28-9](#) for more details.

28.2.4.1 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 TXxREG with a dummy character to initiate transmission (the value is ignored).
4. Write '55h' to TXxREG 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 TXxREG becomes empty, as indicated by the TXxIF, the next data byte can be written to TXxREG.

28.2.5 Receiving a Break Character

The EUSART module can receive a Break character in two ways.

The first method to detect a Break character uses the FERR bit of the RCxSTA register and the received data as indicated by RCxREG. The Baud Rate Generator is assumed to have been initialized to the expected baud rate.

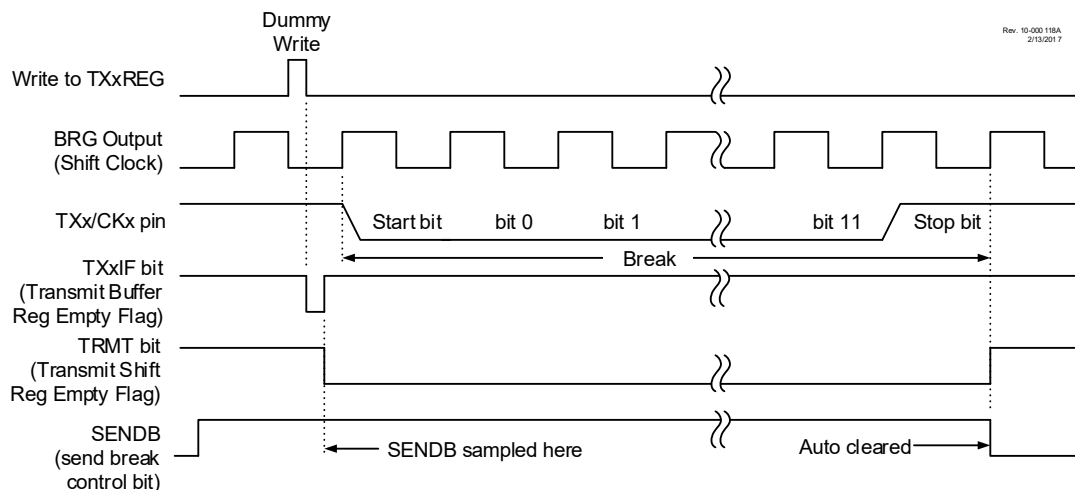
A Break character has been received when all three of the following conditions are true:

- RCxIF bit is set
- FERR bit is set
- RCxREG = 00h

The second method uses the Auto-Wake-up feature described in [28.2.3 Auto-Wake-up on Break](#). By enabling this feature, the EUSART will sample the next two transitions on RX/DT, cause an RCxIF 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 BAUDxCON register before placing the EUSART in Sleep mode.

Figure 28-9. Send Break Character Sequence



28.3 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. The 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.

28.3.1 Synchronous Master Mode

The following bits are used to configure the EUSART for synchronous master operation:

- SYNC bit in the TXxSTA register is set to '1' to configure the EUSART for synchronous operation
- CSRC bit in the TXxSTA register is set to '1' to configure the EUSART as the master
- SREN bit in the RCxSTA register is set to '0' for transmit; SREN = 1 for receive (recommended setting to receive 1 byte)
- CREN bit in the RCxSTA register is set to '0' for transmit; CREN = 1 to receive continuously
- SPEN bit in the RCxSTA register is set to '1' to enable the EUSART interface



Important: Clearing the SREN and CREN bits of the RCxSTA register ensures that the device is in the Transmit mode, otherwise the device will be configured to receive.

28.3.1.1 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 TXx/CKx 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.

28.3.1.2 Clock Polarity

A clock polarity option is provided for Microwire compatibility. Clock polarity is selected with the SCKP bit of the BAUDxCON 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.

28.3.1.3 Synchronous Master Transmission

Data is transferred out of the device on the RXx/DTx pin. The RXx/DTx and TXx/CKx 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 TXxREG register. If the TSR still contains all or part of a previous character the new character data is held in the TXxREG 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 TXxREG is immediately transferred to the TSR. The transmission of the character commences immediately following the transfer of the data to the TSR from the TXxREG.

Each data bit changes on the leading edge of the master clock and remains valid until the subsequent leading clock edge.

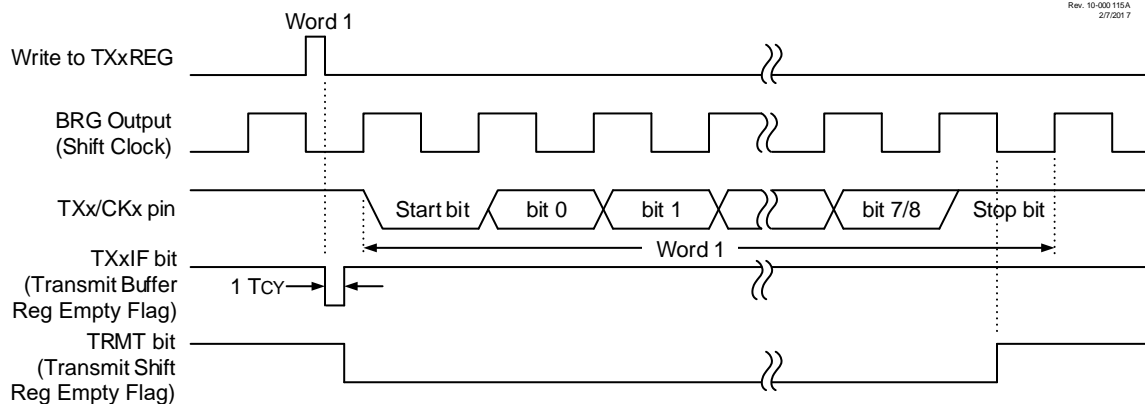
Note: The TSR register is not mapped in data memory, so it is not available to the user.

28.3.1.4 Synchronous Master Transmission Setup

1. Initialize the SPxBRGH, SPxBRGL register pair and the BRG16 bit to achieve the desired baud rate (see section [28.2 EUSART Baud Rate Generator \(BRG\)](#)).

2. Select the transmit output pin by writing the appropriate values to the RxyPPS register and RXxPPS register. Both selections should enable the same pin.
3. Select the clock output pin by writing the appropriate values to the RxyPPS register and CKxPPS register. Both selections should enable the same pin.
4. Enable the synchronous master serial port by setting bits SYNC, SPEN and CSRC.
5. Disable Receive mode by clearing the SREN and CREN bits.
6. Enable Transmit mode by setting the TXEN bit.
7. If 9-bit transmission is desired, set the TX9 bit.
8. If interrupts are desired, set the TXxIE bit of the PIE3 register and the GIE and PEIE bits of the INTCON register.
9. If 9-bit transmission is selected, the ninth bit should be loaded in the TX9D bit.
10. Start transmission by loading data to the TXxREG register.

Figure 28-10. Synchronous Transmission



28.3.1.5 Synchronous Master Reception

Data is received at the RXx/DTx pin. The RXx/DTx 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 RCxSTA register) or the Continuous Receive Enable bit (CREN of the RCxSTA 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 RXx/DTx 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 RCxIF 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 RCxREG. The RCxIF bit remains set as long as there are unread characters in the receive FIFO.

Note: If the RX/DT function is on an analog pin, the corresponding ANSEL bit must be cleared for the receiver to function.

28.3.1.6 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 RCxREG is read to access the FIFO. When this happens the OERR bit of the RCxSTA 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 RCxREG. If the overrun occurred when the CREN bit is set then the error condition is cleared by either clearing the CREN bit of the RCxSTA register or by clearing the SPEN bit which resets the EUSART.

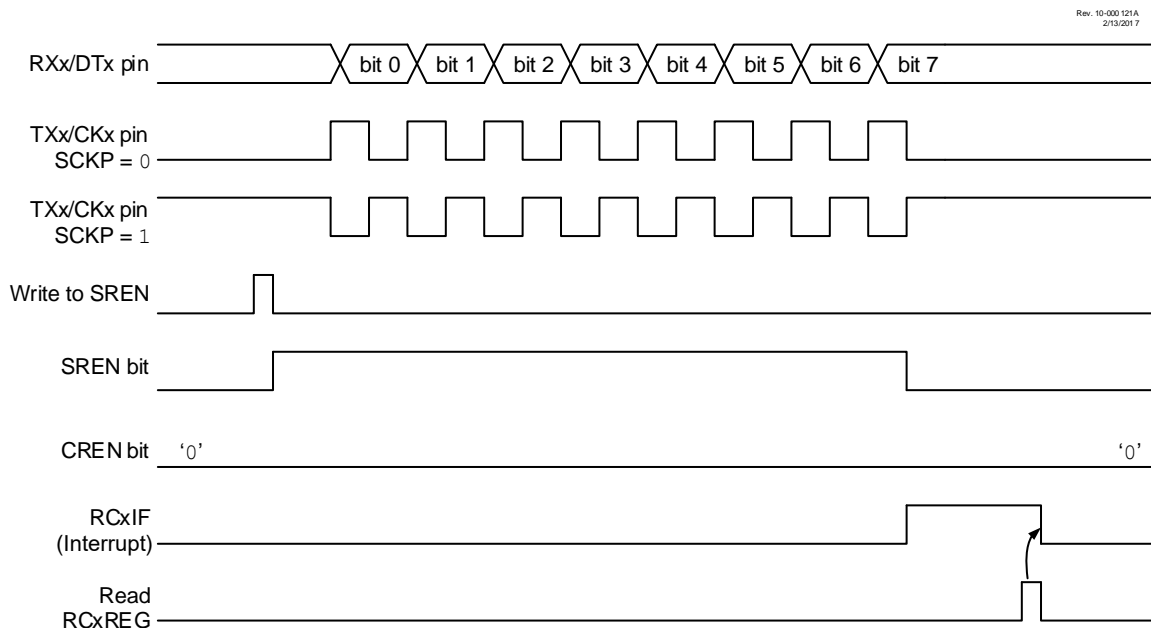
28.3.1.7 Receiving 9-Bit Characters

The EUSART supports 9-bit character reception. When the RX9 bit of the RCxSTA register is set the EUSART will shift nine bits into the RSR for each character received. The RX9D bit of the RCxSTA 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 eight Least Significant bits from the RCxREG.

28.3.1.8 Synchronous Master Reception Setup

1. Initialize the SPxBRGH:SPxBRGL register pair and set or clear the BRG16 bit, as required, to achieve the desired baud rate.
2. Select the receive input pin by writing the appropriate values to the RxyPPS register and RXxPPS register. Both selections should enable the same pin.
3. Select the clock output pin by writing the appropriate values to the RxyPPS register and CKxPPS register. Both selections should enable the same pin.
4. Clear the ANSEL bit for the RXx pin (if applicable).
5. Enable the synchronous master serial port by setting bits SYNC, SPEN and CSRC.
6. Ensure that CREN and SREN bits are cleared.
7. If interrupts are desired, set the RCxIE bit of the PIE3 register and the GIE and PEIE bits of the INTCON register.
8. If 9-bit reception is desired, set bit RX9.
9. Start reception by setting the SREN bit or for continuous reception, set the CREN bit.
10. Interrupt flag bit RCxIF will be set when reception of a character is complete. An interrupt will be generated if the enable bit RCxIE was set.
11. Read the RCxSTA register to get the ninth bit (if enabled) and determine if any error occurred during reception.
12. Read the 8-bit received data by reading the RCxREG register.
13. If an overrun error occurs, clear the error by either clearing the CREN bit of the RCxSTA register or by clearing the SPEN bit which resets the EUSART.

Figure 28-11. Synchronous Reception (Master Mode, SREN)



28.3.2 Synchronous Slave Mode

The following bits are used to configure the EUSART for synchronous slave operation:

- SYNC = 1 (configures the EUSART for synchronous operation.)
- CSRC = 0 (configures the EUSART as a slave)

- SREN = 0 (for transmit); SREN = 1 (for single byte receive)
- CREN = 0 (for transmit); CREN = 1 (recommended setting for continuous receive)
- SPEN = 1 (enables the EUSART)



Important: Clearing the SREN and CREN bits of the RCxSTA register ensures that the device is in the Transmit mode, otherwise the device will be configured to receive.

28.3.2.1 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 TXx/CKx 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.



Important: If the device is configured as a slave and the TX/CK function is on an analog pin, the corresponding ANSEL bit must be cleared.

28.3.2.2 EUSART Synchronous Slave Transmit

The operation of the Synchronous Master and Slave modes are identical (see [28.3.1.3 Synchronous Master Transmission](#)), except in the case of the Sleep mode.

If two words are written to the TXxREG 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 the TXxREG register.
3. The TXxIF bit will not be set.
4. After the first character has been shifted out of TSR, the TXxREG register will transfer the second character to the TSR and the TXxIF bit will now be set.
5. If the PEIE and TXxIE 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.

28.3.2.3 Synchronous Slave Transmission Setup

1. Set the SYNC and SPEN bits and clear the CSRC bit.
2. Select the transmit output pin by writing the appropriate values to the RxyPPS register and RXxPPS register. Both selections should enable the same pin.
3. Select the clock input pin by writing the appropriate value to the CKxPPS register.
4. Clear the ANSEL bit for the CKx pin (if applicable).
5. Clear the CREN and SREN bits.
6. If interrupts are desired, set the TXxIE bit of the PIE3 register and the GIE and PEIE bits of the INTCON register.
7. If 9-bit transmission is desired, set the TX9 bit.
8. Enable transmission by setting the TXEN bit.
9. If 9-bit transmission is selected, insert the Most Significant bit into the TX9D bit.
10. Prepare for transmission by writing the Least Significant eight bits to the TXxREG register. The word will be transmitted in response to the Master clocks at the CKx pin.

28.3.2.4 EUSART Synchronous Slave Reception

The operation of the Synchronous Master and Slave modes is identical (see [28.3.1.5 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 RCxREG register. If the RCxIE 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.

28.3.2.5 Synchronous Slave Reception Setup:

1. Set the SYNC and SPEN bits and clear the CSRC bit.
2. Select the receive input pin by writing the appropriate value to the RXxPPS register.
3. Select the clock input pin by writing the appropriate values to the CKxPPS register.
4. Clear the ANSEL bit for both the TXx/CKx and RXx/DTx pins (if applicable).
5. If interrupts are desired, set the RCxIE bit of the PIE3 register and the GIE and PEIE bits of the INTCON register.
6. If 9-bit reception is desired, set the RX9 bit.
7. Set the CREN bit to enable reception.
8. The RCxIF bit will be set when reception is complete. An interrupt will be generated if the RCxIE bit was set.
9. If 9-bit mode is enabled, retrieve the Most Significant bit from the RX9D bit of the RCxSTA register.
10. Retrieve the eight Least Significant bits from the receive FIFO by reading the RCxREG register.
11. If an overrun error occurs, clear the error by either clearing the CREN bit of the RCxSTA register or by clearing the SPEN bit which resets the EUSART.

28.4 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.

28.4.1 Synchronous Receive During Sleep

To receive during Sleep, all the following conditions must be met before entering Sleep mode:

- RCxSTA and TXxSTA Control registers must be configured for Synchronous Slave Reception (see [28.3.2.5 Synchronous Slave Reception Setup](#)).
- If interrupts are desired, set the RCxIE bit of the PIE3 register and the GIE and PEIE bits of the INTCON register.
- The RCxIF interrupt flag must be cleared by reading RCxREG 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 RXx/DTx and TXx/CKx pins, respectively. When the data word has been completely clocked in by the external device, the RCxIF interrupt flag bit of the PIR3 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 Global Interrupt Enable (GIE) bit of the INTCON register is also set, then the Interrupt Service Routine at address 0x0004 will be called.

28.4.2 Synchronous Transmit During Sleep

To transmit during Sleep, all the following conditions must be met before entering Sleep mode:

- The RCxSTA and TXxSTA Control registers must be configured for synchronous slave transmission (see [28.3.2.3 Synchronous Slave Transmission Setup](#)).
- The TXxIF interrupt flag must be cleared by writing the output data to the TXxREG, thereby filling the TSR and transmit buffer.
- The TXxIE interrupt enable bits of the PIE3 register and PEIE of the INTCON register must be written to '1'.

- If interrupts are desired, set the GIE bit of the INTCON register.

Upon entering Sleep mode, the device will be ready to accept clocks on the TXx/CKx pin and transmit data on the RXx/DTx pin. When the data word in the TSR register has been completely clocked out by the external device, the pending byte in the TXxREG will transfer to the TSR and the TXxIF flag will be set. Thereby, waking the processor from Sleep. At this point, the TXxREG is available to accept another character for transmission. Writing TXxREG will clear the TXxIF flag.

Upon waking from Sleep, the instruction following the `SLEEP` instruction will be executed. If the Global Interrupt Enable (GIE) bit is also set then the Interrupt Service Routine at address 0x0004 will be called.

28.5 Register Summary - EUSART

| Address | Name | Bit Pos. | | | | | | | | |
|-------------|----------|----------|-------------|-------|------|------|-------|------|------|-------|
| 0x00 ... | Reserved | | | | | | | | | |
| 0x0118 | | | | | | | | | | |
| 0x0119 | RC1REG | 7:0 | RCREG[7:0] | | | | | | | |
| 0x011A | TX1REG | 7:0 | TXREG[7:0] | | | | | | | |
| 0x011B | SP1BRG | 7:0 | SPBRG[7:0] | | | | | | | |
| | | 15:8 | SPBRG[15:8] | | | | | | | |
| 0x011D | RC1STA | 7:0 | SPEN | RX9 | SREN | CREN | ADDEN | FERR | OERR | RX9D |
| 0x011E | TX1STA | 7:0 | CSRC | TX9 | TXEN | SYNC | SENDB | BRGH | TRMT | TX9D |
| 0x011F | BAUD1CON | 7:0 | ABDOVF | RCIDL | | SCKP | BRG16 | | WUE | ABDEN |

28.6 Register Definitions: EUSART Control

28.6.1 RCxREG

Name: RCxREG

Address: 0x0119

Receive Data Register

| | | | | | | | | |
|--------|------------|----|----|----|----|----|----|----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | RCREG[7:0] | | | | | | | |
| Access | RO | RO | RO | RO | RO | RO | RO | RO |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bits 7:0 – RCREG[7:0] Receive data

28.6.2 TXxREG

Name: TXxREG

Address: 0x011A

Transmit Data Register

| | | | | | | | | |
|--------|------------|-----|-----|-----|-----|-----|-----|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TXREG[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bits 7:0 – TXREG[7:0] Transmit Data

28.6.3 SPxBRG

Name: SPxBRG

Address: 0x011B

UART BAUD Rate Generator

| | | | | | | | | |
|--------|-------------|-----|-----|-----|-----|-----|-----|-----|
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| | SPBRG[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | SPBRG[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bits 15:0 – SPBRG[15:0] Baud Rate Register

Note: The individual bytes in this multi-byte register can be accessed with the following register names:

- SPxBRGH: Accesses the high byte SPBRG[15:8]
- SPxBRGL: Accesses the low byte SPBRG[7:0]

28.6.4 RCxSTA

Name: RCxSTA
Address: 0x011D

Receive Status and Control Register

| | | | | | | | | |
|--------|------|-----|------|------|-------|------|------|------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | SPEN | RX9 | SREN | CREN | ADDEN | FERR | OERR | RX9D |
| Access | R/W | R/W | R/W | R/W | R/W | RO | R/HC | R/HC |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bit 7 – SPEN Serial Port Enable bit

| Value | Description |
|-------|--------------------------------------|
| 1 | Serial port enabled |
| 0 | Serial port disabled (held in Reset) |

Bit 6 – RX9 9-Bit Receive Enable bit

| Value | Description |
|-------|-------------------------|
| 1 | Selects 9-bit reception |
| 0 | Selects 8-bit reception |

Bit 5 – SREN Single Receive Enable bit

Controls reception. This bit is cleared by hardware when reception is complete

| Value | Condition | Description |
|-------|-----------------------|----------------------------|
| 1 | SYNC = 1 AND CSRC = 1 | Start single receive |
| 0 | SYNC = 1 AND CSRC = 1 | Single receive is complete |
| X | SYNC = 0 OR CSRC = 0 | Don't care |

Bit 4 – CREN Continuous Receive Enable bit

| Value | Condition | Description |
|-------|-----------|---|
| 1 | SYNC = 1 | Enables continuous receive until enable bit CREN is cleared (CREN overrides SREN) |
| 0 | SYNC = 1 | Disables continuous receive |
| 1 | SYNC = 0 | Enables receiver |
| 0 | SYNC = 0 | Disables receiver |

Bit 3 – ADDEN Address Detect Enable bit

| Value | Condition | Description |
|-------|----------------------|---|
| 1 | SYNC = 0 AND RX9 = 1 | The receive buffer is loaded and the interrupt occurs only when the ninth received bit is set |
| 0 | SYNC = 0 AND RX9 = 1 | All bytes are received and interrupt always occurs. Ninth bit can be used as parity bit |
| X | RX9 = 0 OR SYNC = 1 | Don't care |

Bit 2 – FERR Framing Error bit

| Value | Description |
|-------|--|
| 1 | Unread byte in 28.6.1 RCxREG has a framing error |
| 0 | Unread byte in 28.6.1 RCxREG does not have a framing error |

Bit 1 – OERR Overrun Error bit

| Value | Description |
|-------|--|
| 1 | Overrun error (can be cleared by clearing either SPEN or CREN bit) |
| 0 | No overrun error |

Bit 0 – RX9D Ninth bit of Received Data

This can be address/data bit or a parity bit which is determined by user firmware.

28.6.5 TXxSTA

Name: TXxSTA

Address: 0x011E

Transmit Status and Control Register

| | | | | | | | | |
|--------|------|-----|------|------|-------|------|------|------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | CSRC | TX9 | TXEN | SYNC | SENDB | BRGH | TRMT | TX9D |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | RO | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

Bit 7 – CSRC Clock Source Select bit

| Value | Condition | Description |
|-------|-----------|---|
| 1 | SYNC = 1 | Master mode (clock generated internally from BRG) |
| 0 | SYNC = 1 | Slave mode (clock from external source) |
| X | SYNC = 0 | Don't care |

Bit 6 – TX9 9-bit Transmit Enable bit

| Value | Description |
|-------|----------------------------|
| 1 | Selects 9-bit transmission |
| 0 | Selects 8-bit transmission |

Bit 5 – TXEN Transmit Enable bit

Enables transmitter⁽¹⁾

| Value | Description |
|-------|-------------------|
| 1 | Transmit enabled |
| 0 | Transmit disabled |

Bit 4 – SYNC EUSART Mode Select bit

| Value | Description |
|-------|-------------------|
| 1 | Synchronous mode |
| 0 | Asynchronous mode |

Bit 3 – SENDB Send Break Character bit

| Value | Condition | Description |
|-------|-----------|--|
| 1 | SYNC = 0 | Send Sync Break on next transmission (cleared by hardware upon completion) |
| 0 | SYNC = 0 | Sync Break transmission disabled or completed |
| X | SYNC = 1 | Don't care |

Bit 2 – BRGH High Baud Rate Select bit

| Value | Condition | Description |
|-------|-----------|---|
| 1 | SYNC = 0 | High speed, if BRG16 = 1, baud rate is baudclk/4; else baudclk/16 |
| 0 | SYNC = 0 | Low speed |
| X | SYNC = 1 | Don't care |

Bit 1 – TRMT Transmit Shift Register (TSR) Status bit

| Value | Description |
|-------|------------------|
| 1 | TSR is empty |
| 0 | TSR is not empty |

Bit 0 – TX9D Ninth bit of Transmit Data

Can be address/data bit or a parity bit.

Note:

1. **SREN** and **CREN** bits override TXEN in Sync mode.

28.6.6 BAUDxCON

Name: BAUDxCON

Address: 0x011F

Baud Rate Control Register

| | | | | | | | | |
|--------|--------|-------|---|------|-------|---|-----|-------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | ABDOVF | RCIDL | | SCKP | BRG16 | | WUE | ABDEN |
| Access | RO | RO | | RW | RW | | RW | RW |
| Reset | 0 | 0 | | 0 | 0 | | 0 | 0 |

Bit 7 – ABDOVF Auto-Baud Detect Overflow bit

| Value | Condition | Description |
|-------|-----------|----------------------------------|
| 1 | SYNC = 0 | Auto-baud timer overflowed |
| 0 | SYNC = 0 | Auto-baud timer did not overflow |
| X | SYNC = 1 | Don't care |

Bit 6 – RCIDL Receive Idle Flag bit

| Value | Condition | Description |
|-------|-----------|---|
| 1 | SYNC = 0 | Receiver is Idle |
| 0 | SYNC = 0 | Start bit has been received and the receiver is receiving |
| X | SYNC = 1 | Don't care |

Bit 4 – SCKP Synchronous Clock Polarity Select bit

| Value | Condition | Description |
|-------|-----------|--|
| 1 | SYNC = 0 | Idle state for transmit (TX) is a low level (transmit data inverted) |
| 0 | SYNC = 0 | Idle state for transmit (TX) is a high level (transmit data is non-inverted) |
| 1 | SYNC = 1 | Data is clocked on rising edge of the clock |
| 0 | SYNC = 1 | Data is clocked on falling edge of the clock |

Bit 3 – BRG16 16-bit Baud Rate Generator Select bit

| Value | Description |
|-------|------------------------------------|
| 1 | 16-bit Baud Rate Generator is used |
| 0 | 8-bit Baud Rate Generator is used |

Bit 1 – WUE Wake-up Enable bit

| Value | Condition | Description |
|-------|-----------|--|
| 1 | SYNC = 0 | Receiver is waiting for a falling edge. Upon falling edge no character will be received and flag RCxIF will be set. WUE will automatically clear after RCxIF is set. |
| 0 | SYNC = 0 | Receiver is operating normally |
| X | SYNC = 1 | Don't care |

Bit 0 – ABDEN Auto-Baud Detect Enable bit

| Value | Condition | Description |
|-------|-----------|--|
| 1 | SYNC = 0 | Auto-Baud Detect mode is enabled (clears when auto-baud is complete) |
| 0 | SYNC = 0 | Auto-Baud Detect is complete or mode is disabled |
| X | SYNC = 1 | Don't care |

29. MSSP - Master Synchronous Serial Port Module

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 (I²C)

The SPI interface can operate in Master or Slave mode and supports the following features:

- Selectable Clock Parity
- Slave Select Synchronization (Slave Mode Only)
- Daisy-Chain Connection of Slave Devices

The I²C interface can operate in Master or Slave mode and supports the following modes and features:

- Byte NACKing (Slave Mode)
- Limited Multi-Master Support
- 7-Bit and 10-Bit Addressing
- Start and Stop Interrupts
- Interrupt Masking
- Clock Stretching
- Bus Collision Detection
- General Call Address Matching
- Address Masking
- Address Hold and Data Hold Modes
- Selectable SDA Hold Times

29.1 SPI Mode Overview

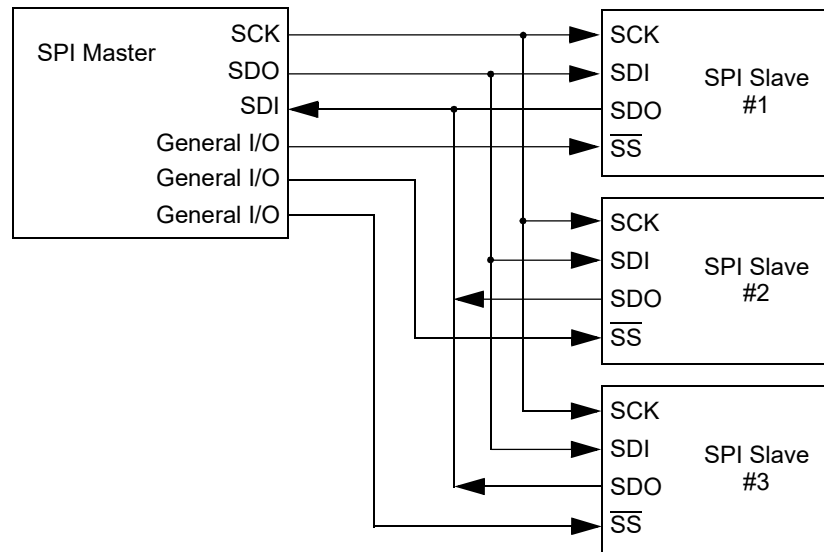
The Serial Peripheral Interface (SPI) bus is a synchronous serial data communication bus that operates in Full-Duplex mode. Devices communicate in a master/slave environment where the master device initiates the communication. A slave device is controlled through a Chip Select known as Slave Select.

The SPI bus specifies four signal connections:

- Serial Clock (SCK)
- Serial Data Out (SDO)
- Serial Data In (SDI)
- Slave Select (\overline{SS})

Figure 29-1 shows the block diagram of the MSSP module when operating in SPI mode.

Figure 29-2. SPI Master and Multiple Slave Connection

Rev. 30-000012A
3/31/2017

29.1.1 SPI Mode Registers

The MSSP module has five registers for SPI mode operation. These are:

- MSSP STATUS register (SSPxSTAT)
- MSSP Control register 1 (SSPxCON1)
- MSSP Control register 3 (SSPxCON3)
- MSSP Data Buffer register (SSPxBUF)
- MSSP Address register (SSPxADD)
- MSSP Shift register (SSPSR)
(Not directly accessible)

SSPxCON1 and SSPxSTAT are the control and STATUS registers for SPI mode operation. The SSPxCON1 register is readable and writable. The lower six bits of the SSPxSTAT are read-only. The upper two bits of the SSPxSTAT are read/write.

One of the five SPI Master modes uses the SSPxADD value to determine the Baud Rate Generator clock frequency. More information on the Baud Rate Generator is available in [29.7 Baud Rate Generator](#).

SSPSR is the Shift register used for shifting data in and out. SSPxBUF provides indirect access to the SSPSR register. SSPxBUF is the Buffer register to which data bytes are written, and from which data bytes are read.

In receive operations, SSPSR and SSPxBUF together create a buffered receiver. When SSPSR receives a complete byte, it is transferred to SSPxBUF and the SSPxIF interrupt is set.

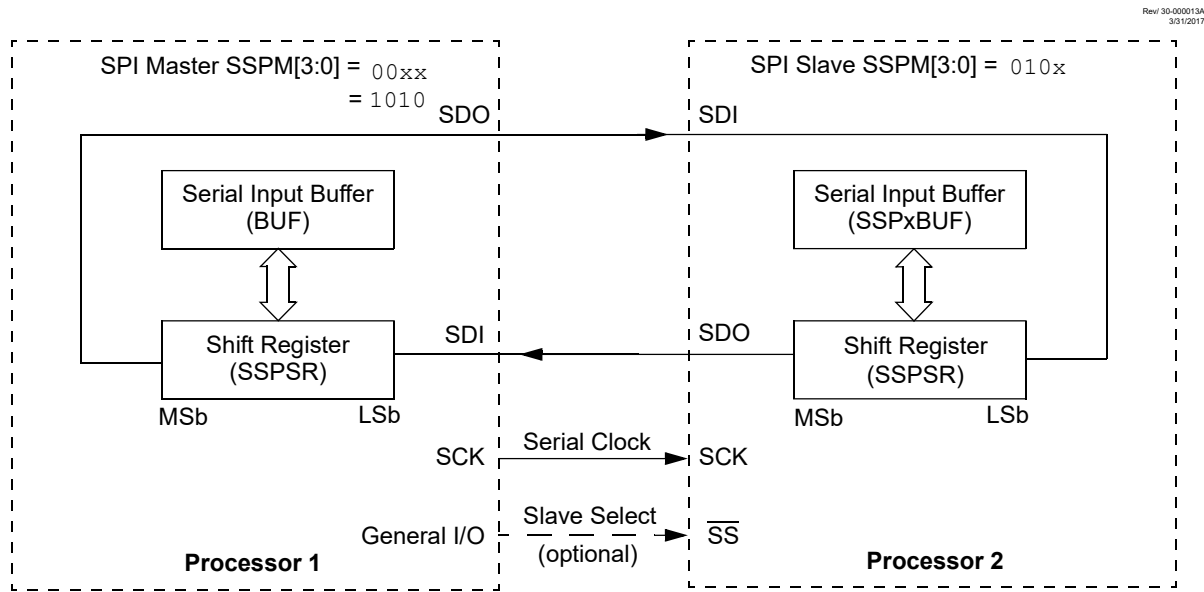
During transmission, the SSPxBUF is not buffered. A write to SSPxBUF will write to both SSPxBUF and SSPSR.

29.2 SPI Mode Operation

Transmissions involve two Shift registers, eight bits in size, one in the master and one in the slave. With either the master or the slave device, data is always shifted out one bit at a time, with the Most Significant bit (MSb) shifted out first. At the same time, a new Least Significant bit (LSb) is shifted into the same register.

The following figure shows a typical connection between two processors configured as master and slave devices.

Figure 29-3. SPI Master/Slave Connection



Data is shifted out of both Shift registers on the programmed clock edge and latched on the opposite edge of the clock.

The master device transmits information out on its SDO output pin, which is connected to and received by the slave's SDI input pin. The slave device transmits information out on its SDO output pin, which is connected to and received by the master's SDI input pin.

To begin communication the master device first sends out the clock signal. Both the master and the slave devices should be configured for the same clock polarity.

The master device starts a transmission by sending out the MSb from its Shift register. The slave device reads this bit from that same line and saves it into the LSb position of its Shift register.

During each SPI clock cycle, a full-duplex data transmission occurs. This means that while the master device is sending out the MSb from its Shift register (on its SDO pin) and the slave device is reading this bit and saving it as the LSb of its Shift register, the slave device is also sending out the MSb from its Shift register (on its SDO pin) and the master device is reading this bit and saving it as the LSb of its Shift register.

After eight bits have been shifted out, the master and slave have exchanged register values.

If there is more data to exchange, the Shift registers are loaded with new data and the process repeats itself.

Whether the data is meaningful or not (dummy data), depends on the application software. This leads to three scenarios for data transmission:

- Master sends useful data and slave sends dummy data.
- Master sends useful data and slave sends useful data.
- Master sends dummy data and slave sends useful data.

Transmissions may involve any number of clock cycles. When there is no more data to be transmitted, the master stops sending the clock signal and it deselects the slave.

Every slave device connected to the bus that has not been selected through its slave select line must disregard the clock and transmission signals and must not transmit out any data of its own.

When initializing the SPI several options need to be specified. This is done by programming the appropriate control bits ($SSPxCON1[5:0]$ and $SSPxSTAT[7:6]$). These control bits allow the following to be specified:

- Master mode (SCK is the clock output)
- Slave mode (SCK is the clock input)
- Clock Polarity (Idle state of SCK)

- Data Input Sample Phase (middle or end of data output time)
- Clock Edge (output data on rising/falling edge of SCK)
- Clock Rate (Master mode only)
- Slave Select mode (Slave mode only)

To enable the serial port the SSP Enable bit, **SSPEN**, must be set. To reset or reconfigure SPI mode, clear the SSPEN bit, re-initialize the SSPxCONx registers and then set the SSPEN bit. The SDI, SDO, SCK and \overline{SS} serial port pins are selected with the PPS controls. For the pins to behave as the serial port function, some must have their data direction bits (in the TRIS register) appropriately programmed as follows:

- SDI must have corresponding TRIS bit set
- SDO must have corresponding TRIS bit cleared
- SCK (Master mode) must have corresponding TRIS bit cleared
- SCK (Slave mode) must have corresponding TRIS bit set
- The RxyPPS and SSPxCLKPPS controls must select the same pin
- \overline{SS} must have corresponding TRIS bit set

Any serial port function that is not desired may be overridden by programming the corresponding data direction (TRIS) register to the opposite value.

The MSSP consists of a transmit/receive shift register (SSPSR) and a buffer register (SSPxBUF). The SSPSR shifts the data in and out of the device, MSb first. The SSPxBUF holds the data that was written to the SSPSR until the received data is ready. Once the eight bits of data have been received, that byte is moved to the SSPxBUF register. Then, the Buffer Full Detect bit, **BF**, and the Interrupt Flag bit, SSPxIF, are set. This double-buffering of the received data (SSPxBUF) allows the next byte to start reception before reading the data that was just received. Any write to the SSPxBUF register during transmission/reception of data will be ignored and the write collision detect bit, **WCOL**, will be set. User software must clear the WCOL bit to allow the following write(s) to the SSPxBUF register to complete successfully.

When the application software is expecting to receive valid data, the SSPxBUF should be read before the next byte of data to transfer is written to the SSPxBUF. The Buffer Full bit, **BF**, indicates when SSPxBUF has been loaded with the received data (transmission is complete). When the SSPxBUF is read, the BF bit is cleared. This data may be irrelevant if the SPI is only a transmitter. Generally, the MSSP interrupt is used to determine when the transmission/reception has completed. If the interrupt method is not going to be used, then software polling can be done to ensure that a write collision does not occur.

The SSPSR is not directly readable or writable and can only be accessed by addressing the SSPxBUF register. Additionally, the SSPxSTAT register indicates the various Status conditions.

29.2.1 SPI Master Mode

The master can initiate the data transfer at any time because it controls the SCK line. The master determines when the slave (Processor 2, [Figure 29-3](#)) is to broadcast data by the software protocol.

In Master mode, the data is transmitted/received as soon as the SSPxBUF register is written to. If the SPI is only going to receive, the SDO output could be disabled (programmed as an input). The SSPSR register will continue to shift in the signal present on the SDI pin at the programmed clock rate. As each byte is received, it will be loaded into the SSPxBUF register as if a normal received byte (interrupts and Status bits appropriately set).

The clock polarity is selected by appropriately programming the **CKP** bit and the **CKE** bit. This then, would give waveforms for SPI communication as shown in [Figure 29-4](#), [Figure 29-6](#), [Figure 29-7](#) and [Figure 29-8](#), where the MSB is transmitted first. In Master mode, the SPI clock rate (bit rate) is user programmable to be one of the following:

- $F_{OSC}/4$ (or T_{CY})
- $F_{OSC}/16$ (or $4 * T_{CY}$)
- $F_{OSC}/64$ (or $16 * T_{CY}$)
- Timer2 output/2
- $F_{OSC}/(4 * (SSPxADD + 1))$

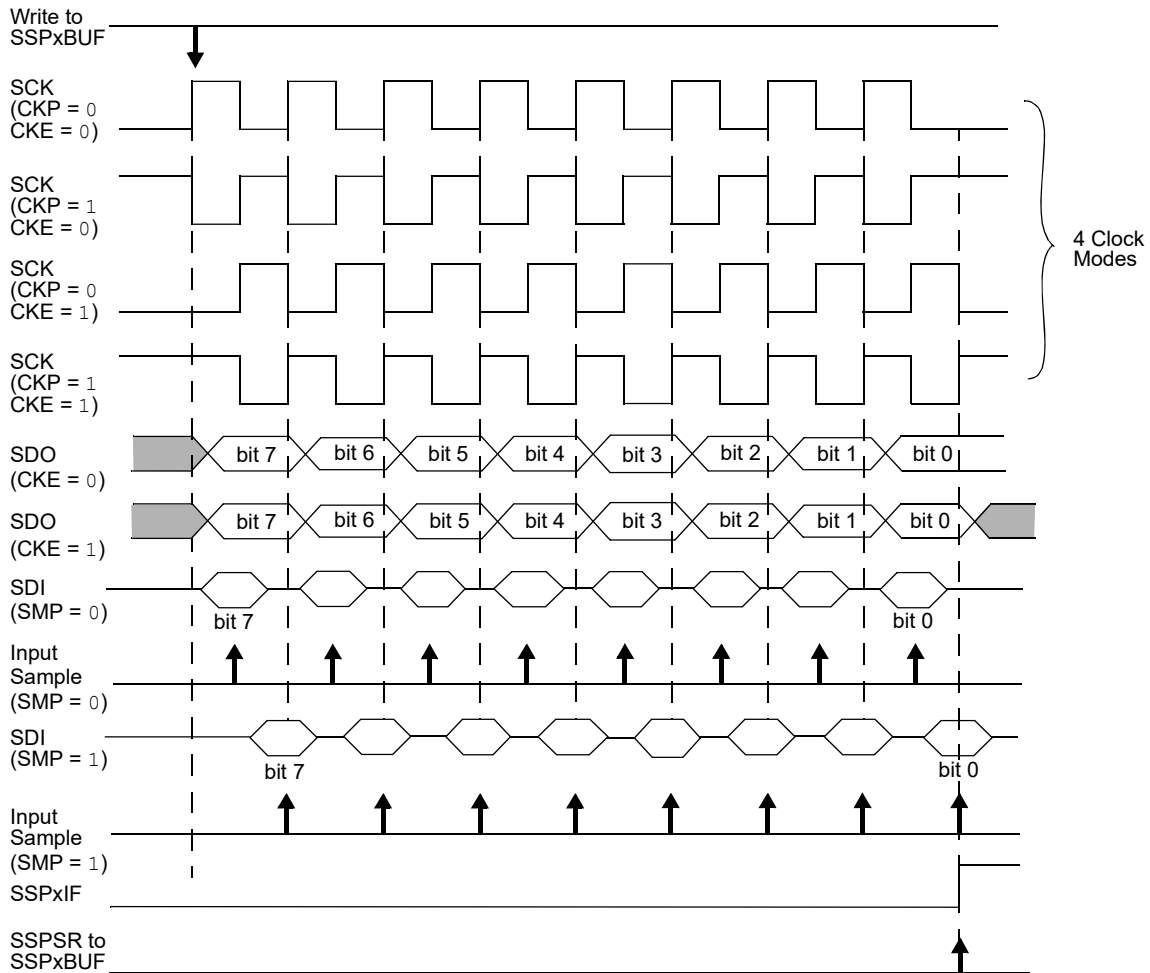
[Figure 29-4](#) shows the waveforms for Master mode.

When the CKE bit is set, the SDO data is valid before there is a clock edge on SCK. The change of the input sample is shown based on the state of the SMP bit. The time when the SSPxBUF is loaded with the received data is shown.



Important: In Master mode the clock signal output to the SCK pin is also the clock signal input to the peripheral. The pin selected for output with the RxyPPS register must also be selected as the peripheral input with the SSPxCLKPPS register. The pin that is selected using the SSPxCLKPPS register should also be made a digital I/O. This is done by clearing the corresponding ANSEL bit.

Figure 29-4. SPI Mode Waveform (Master Mode)



29.2.2 SPI Slave Mode

In Slave mode, the data is transmitted and received as external clock pulses appear on SCK. When the last bit is latched, the SSPxIF Interrupt Flag bit is set.

Before enabling the module in SPI Slave mode, the clock line must match the proper Idle state. The clock line can be observed by reading the SCK pin. The Idle state is determined by the CKP bit.

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” chapter.

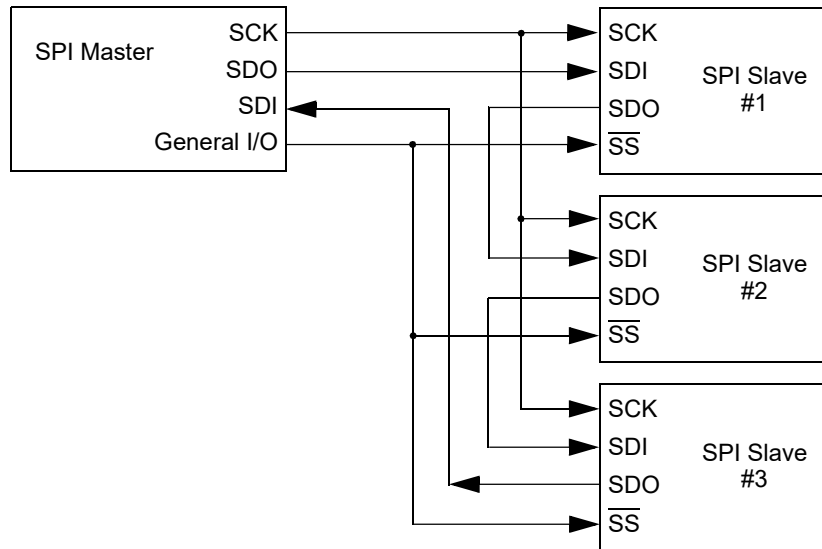
While in Sleep mode, the slave can transmit/receive data. The Shift register is clocked from the SCK pin input and when a byte is received, the device will generate an interrupt. If enabled, the device will wake-up from Sleep.

29.2.3 Daisy-Chain Configuration

The SPI bus can sometimes be connected in a daisy-chain configuration. The first slave output is connected to the second slave input, the second slave output is connected to the third slave input, and so on. The final slave output is connected to the master input. Each slave sends out, during a second group of clock pulses, an exact copy of what was received during the first group of clock pulses. The whole chain acts as one large communication shift register. The daisy-chain feature only requires a single Slave Select line from the master device.

The following figure shows the block diagram of a typical daisy-chain connection when operating in SPI mode.

Figure 29-5. SPI Daisy-Chain Connection



Rev. 30-000015A
3/31/2017

In a daisy-chain configuration, only the most recent byte on the bus is required by the slave. Setting the **BOEN** bit will enable writes to the SSPxBUF register, even if the previous byte has not been read. This allows the software to ignore data that may not apply to it.

29.2.4 Slave Select Synchronization

The Slave Select can also be used to synchronize communication. The Slave Select line is held high until the master device is ready to communicate. When the Slave Select line is pulled low, the slave knows that a new transmission is starting.

If the slave fails to receive the communication properly, it will be reset at the end of the transmission, when the Slave Select line returns to a high state. The slave is then ready to receive a new transmission when the Slave Select line is pulled low again. If the Slave Select line is not used, there is a risk that the slave will eventually become out of sync with the master. If the slave misses a bit, it will always be one bit off in future transmissions. Use of the Slave Select line allows the slave and master to align themselves at the beginning of each transmission.

The \overline{SS} pin allows a Synchronous Slave mode. The SPI must be in Slave mode with \overline{SS} pin control enabled (**SSPM** = 0100).

When the \overline{SS} pin is low, transmission and reception are enabled and the SDO pin is driven.

When the \overline{SS} pin goes high, the SDO pin is no longer driven, even if in the middle of a transmitted byte and becomes a floating output. External pull-up/pull-down resistors may be desirable depending on the application.

Note:

1. When the SPI is in Slave mode with \overline{SS} pin control enabled (**SSPM** = 0100), the SPI module will reset if the \overline{SS} pin is set to V_{DD} .
2. When the SPI is used in Slave mode with **CKE** set; the user must enable \overline{SS} pin control.
3. While operated in SPI Slave mode the **SMP** bit must remain clear.

When the SPI module resets, the bit counter is forced to '0'. This can be done by either forcing the \overline{SS} pin to a high level or clearing the SSPEN bit.

Figure 29-6. Slave Select Synchronous Waveform

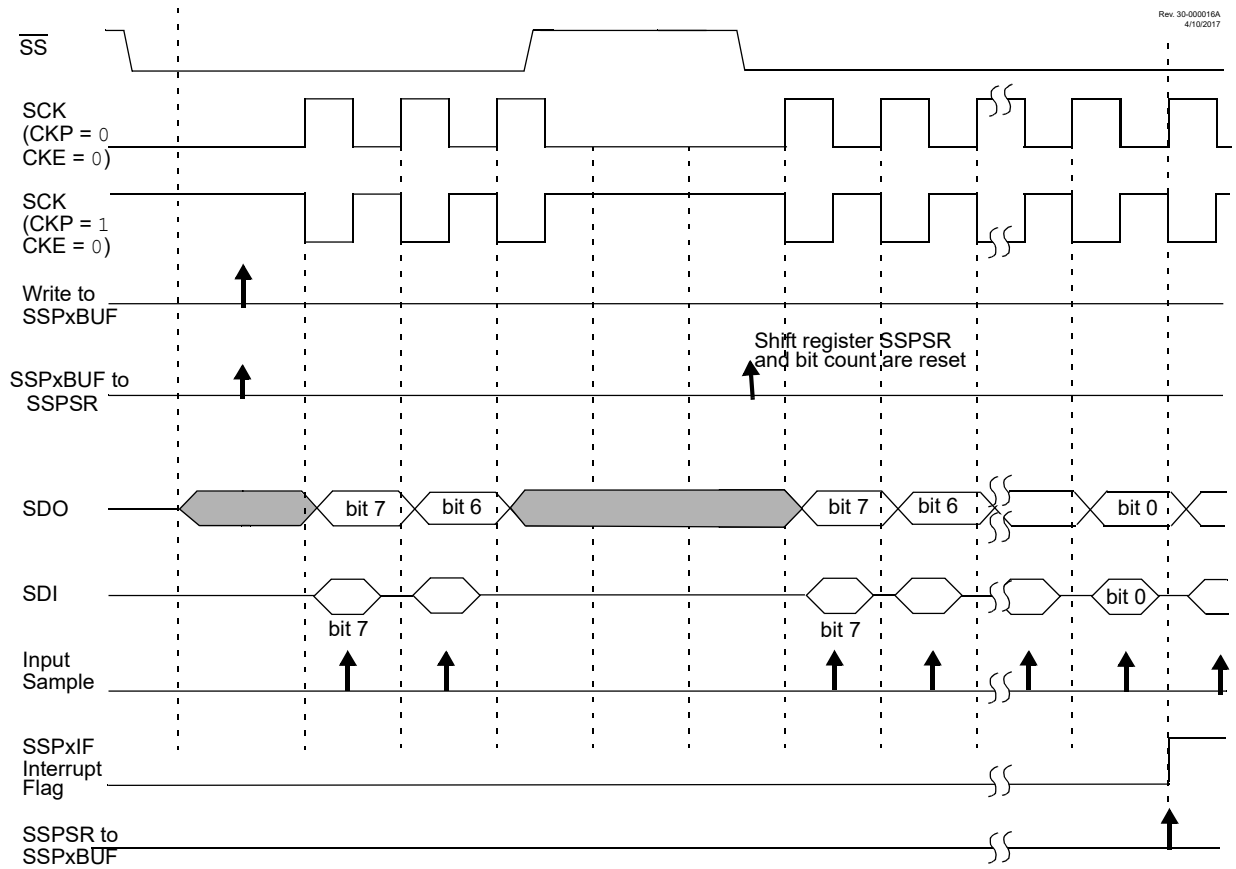


Figure 29-7. SPI Mode Waveform (Slave Mode with CKE = 0)

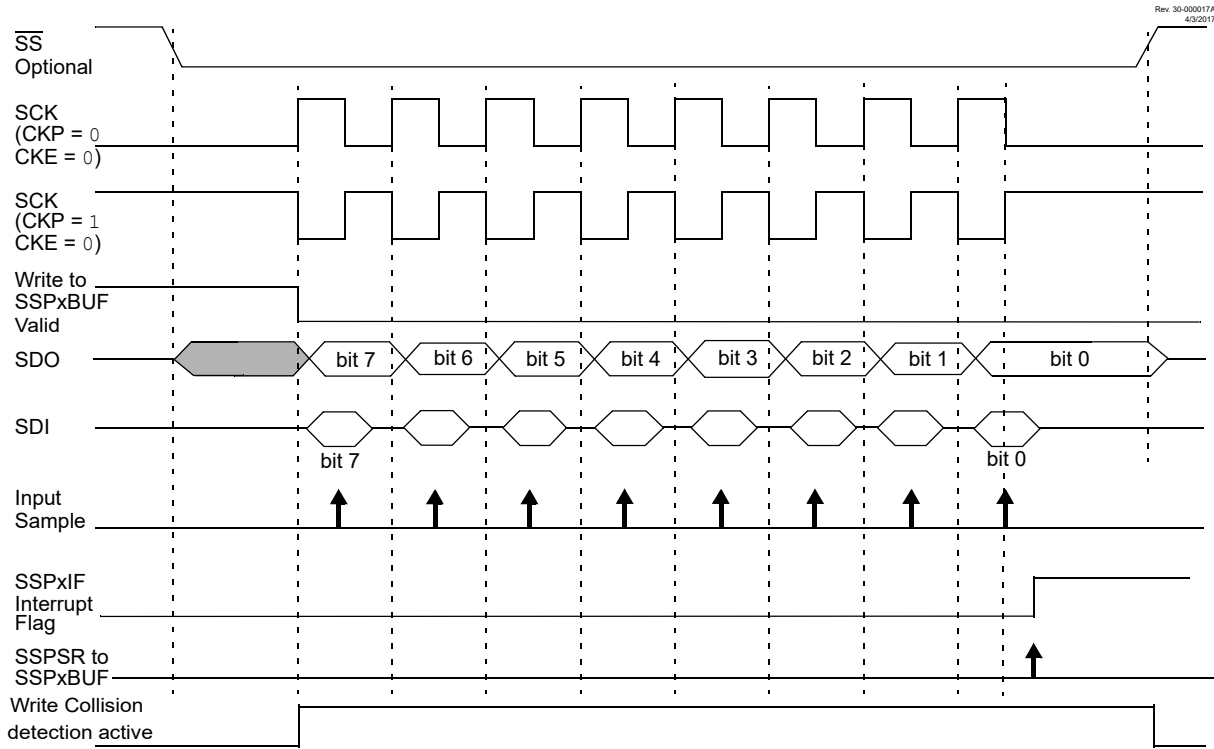
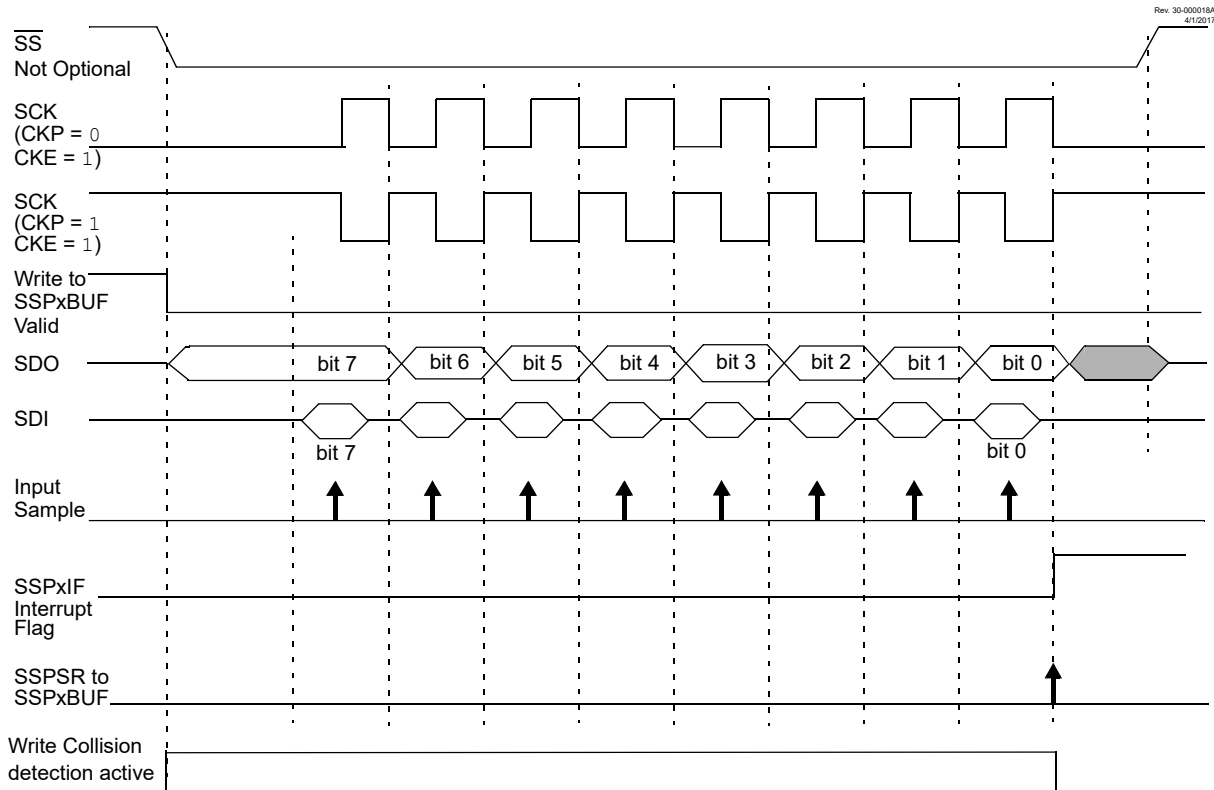


Figure 29-8. SPI Mode Waveform (Slave Mode with CKE = 1)



29.2.5 SPI Operation in Sleep Mode

In SPI Master mode, module clocks may be operating at a different speed than when in Full-Power mode; in the case of the Sleep mode, all clocks are halted.

Special care must be taken by the user when the MSSP clock is much faster than the system clock.

In Slave mode, when MSSP interrupts are enabled, after the master completes sending data, an MSSP interrupt will wake the controller from Sleep.

If an exit from Sleep mode is not desired, MSSP interrupts should be disabled.

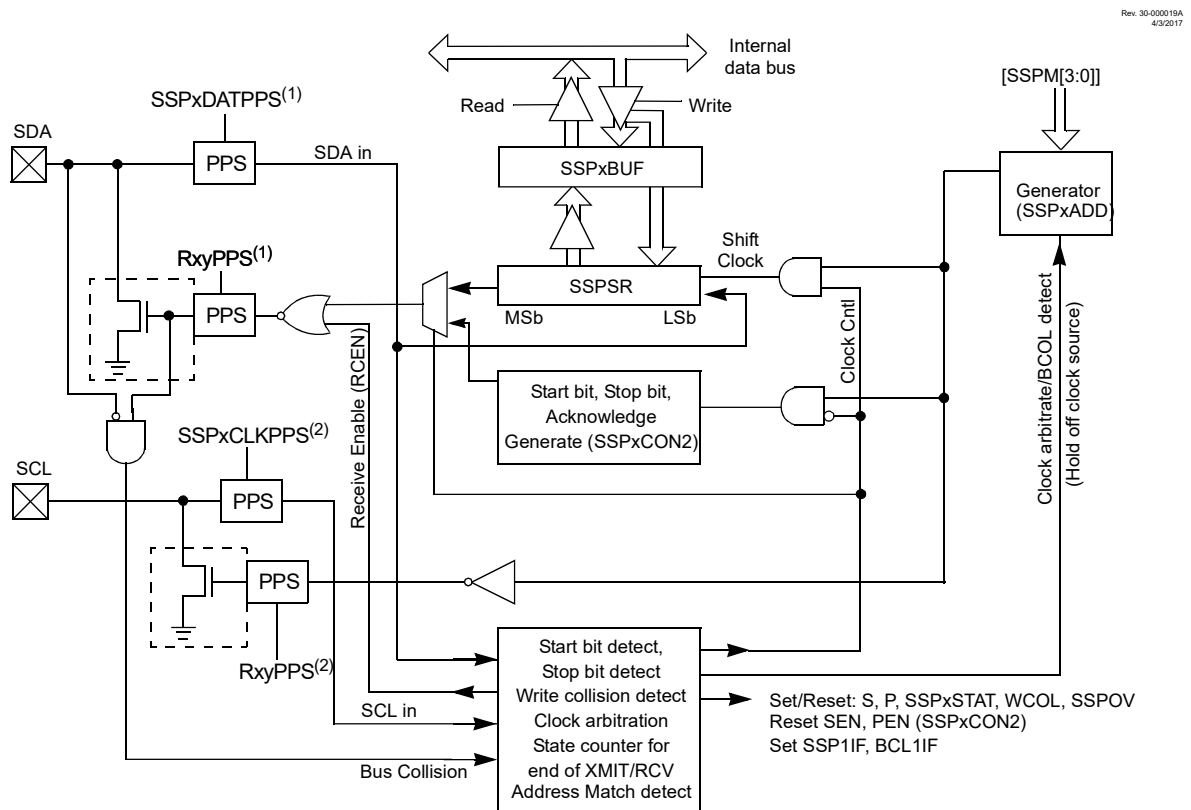
In SPI Master mode, when the Sleep mode is selected, all module clocks are halted and the transmission/reception will remain in that state until the device wakes. After the device returns to Run mode, the module will resume transmitting and receiving data.

In SPI Slave mode, the SPI Transmit/Receive Shift register operates asynchronously to the device. This allows the device to be placed in Sleep mode and data to be shifted into the SPI Transmit/Receive Shift register. When all eight bits have been received, the MSSP interrupt flag bit will be set and if enabled, will wake the device.

29.3 I²C Mode Overview

The Inter-Integrated Circuit (I²C) bus is a multimaster serial data communication bus. Devices communicate in a master/slave environment where the master devices initiate the communication. A slave device is controlled through addressing. The following two diagrams show block diagrams of the I²C Master and Slave modes, respectively.

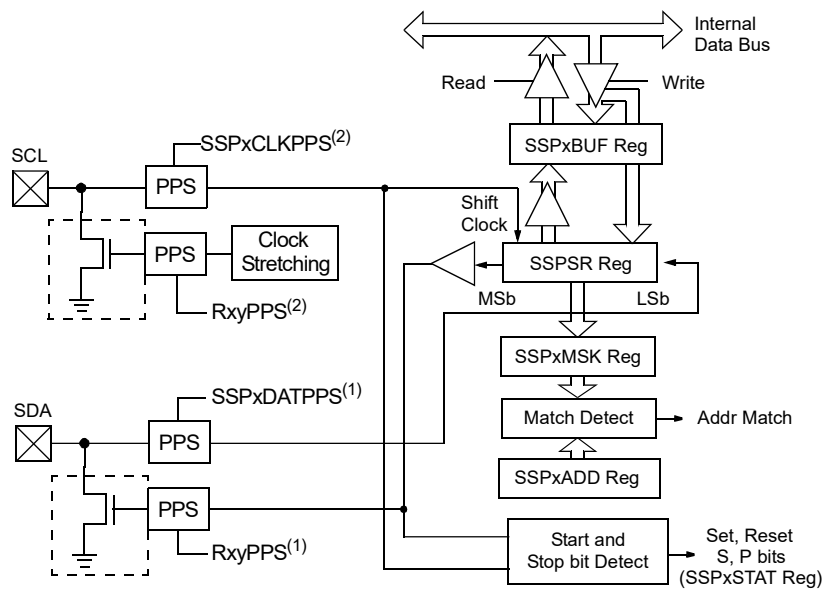
Figure 29-9. MSSP Block Diagram (I²C Master mode)



Note 1: SDA pin selections must be the same for input and output

Note 2: SCL pin selections must be the same for input and output

Figure 29-10. MSSP Block Diagram (I²C Slave mode)



Rev. 30-000020A
4/3/2017

Note 1: SDA pin selections must be the same for input and output

2: SCL pin selections must be the same for input and output

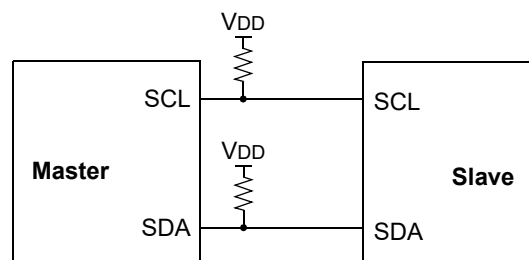
The I²C bus specifies two signal connections:

- Serial Clock (SCL)
- Serial Data (SDA)

Both the SCL and SDA connections are bidirectional open-drain lines, each requiring pull-up resistors for the supply voltage. Pulling the line to ground is considered a logical zero and letting the line float is considered a logical one.

The following diagram shows a typical connection between two processors configured as master and slave devices.

Figure 29-11. I²C Master/Slave Connection



Rev. 30-000021A
4/3/2017

The I²C bus can operate with one or more master devices and one or more slave devices.

There are four potential modes of operation for a given device:

- Master Transmit mode
(master is transmitting data to a slave)
- Master Receive mode
(master is receiving data from a slave)
- Slave Transmit mode
(slave is transmitting data to a master)

- Slave Receive mode
(slave is receiving data from the master)

To begin communication, a master device starts out in Master Transmit mode. The master device sends out a Start bit followed by the address byte of the slave it intends to communicate with. This is followed by a single Read/Write bit, which determines whether the master intends to transmit to or receive data from the slave device.

If the requested slave exists on the bus, it will respond with an Acknowledge bit, otherwise known as an \overline{ACK} . The master then continues in either Transmit mode or Receive mode and the slave continues in the complement, either in Receive mode or Transmit mode, respectively.

A Start bit is indicated by a high-to-low transition of the SDA line while the SCL line is held high. Address and data bytes are sent out, Most Significant bit (MSb) first. The Read/Write bit is sent out as a logical one when the master intends to read data from the slave, and is sent out as a logical zero when it intends to write data to the slave.

The Acknowledge bit (\overline{ACK}) is an active-low signal, which holds the SDA line low to indicate to the transmitter that the slave device has received the transmitted data and is ready to receive more.

The transition of a data bit is always performed while the SCL line is held low. Transitions that occur while the SCL line is held high are used to indicate Start and Stop bits.

If the master intends to write to the slave, then it repeatedly sends out a byte of data, with the slave responding after each byte with an \overline{ACK} bit. In this example, the master device is in Master Transmit mode and the slave is in Slave Receive mode.

If the master intends to read from the slave, then it repeatedly receives a byte of data from the slave, and responds after each byte with an \overline{ACK} bit. In this example, the master device is in Master Receive mode and the slave is in Slave Transmit mode.

On the last byte of data communicated, the master device may end the transmission by sending a Stop bit. If the master device is in Receive mode, it sends the Stop bit in place of the last \overline{ACK} bit. A Stop bit is indicated by a low-to-high transition of the SDA line while the SCL line is held high.

In some cases, the master may want to maintain control of the bus and re-initiate another transmission. If so, the master device may send another Start bit in place of the Stop bit or last \overline{ACK} bit when it is in Receive mode.

The I²C bus specifies three message protocols:

- Single message where a master writes data to a slave.
- Single message where a master reads data from a slave.
- Combined message where a master initiates a minimum of two writes, or two reads, or a combination of writes and reads, to one or more slaves.

When one device is transmitting a logical one, or letting the line float, and a second device is transmitting a logical zero, or holding the line low, the first device can detect that the line is not a logical one. This detection, when used on the SCL line, is called clock stretching. Clock stretching gives slave devices a mechanism to control the flow of data. When this detection is used on the SDA line, it is called arbitration. Arbitration ensures that there is only one master device communicating at any single time.

29.3.1 Register Definitions: I²C Mode

The MSSPx module has seven registers for I²C operation.

These are:

- MSSP Status register (SSPxSTAT)
- MSSP Control register 1 (SSPxCON1)
- MSSP Control register 2 (SSPxCON2)
- MSSP Control register 3 (SSPxCON3)
- Serial Receive/Transmit Buffer register (SSPxBUF)
- MSSP Address register (SSPxADD)
- I²C Slave Address Mask register (SSPxMSK)
- MSSP Shift register (SSPSR) – not directly accessible

SSPxCON1, SSPxCON2, SSPxCON3 and SSPxSTAT are the Control and STATUS registers in I²C mode operation. The SSPxCON1, SSPxCON2, and SSPxCON3 registers are readable and writable. The lower six bits of the SSPxSTAT are read-only. The upper two bits of the SSPxSTAT are read/write. SSPSR is the Shift register used for shifting data in or out. SSPxBUF is the buffer register to which data bytes are written to or read from. SSPxADD contains the slave device address when the MSSP is configured in I²C Slave mode. When the MSSP is configured in Master mode, the lower seven bits of SSPxADD act as the Baud Rate Generator reload value.

SSPxMSK holds the slave address mask value when the module is configured for 7-Bit Address Masking mode. While it is a separate register, it shares the same SFR address as SSPxADD; it is only accessible when the SSPM[3:0] bits are specifically set to permit access. In receive operations, SSPSR and SSPxBUF together, create a double-buffered receiver. When SSPSR receives a complete byte, it is transferred to SSPxBUF and the SSPxIF interrupt is set. During transmission, the SSPxBUF is not double-buffered. A write to SSPxBUF will write to both SSPxBUF and SSPSR.

29.4 I²C Mode Operation

All MSSP I²C communication is byte oriented and shifted out MSb first. Six SFR registers and two interrupt flags interface the module with the PIC[®] microcontroller and user software. Two pins, SDA and SCL, are exercised by the module to communicate with other external I²C devices.

29.4.1 Clock Stretching

When a slave device has not completed processing data, it can delay the transfer of more data through the process of clock stretching. An addressed slave device may hold the SCL clock line low after receiving or sending a bit, indicating that it is not yet ready to continue. The master that is communicating with the slave will attempt to raise the SCL line in order to transfer the next bit, but will detect that the clock line has not yet been released. Because the SCL connection is open-drain, the slave has the ability to hold that line low until it is ready to continue communicating.

Clock stretching allows receivers that cannot keep up with a transmitter to control the flow of incoming data.

29.4.2 Arbitration

Each master device must monitor the bus for Start and Stop bits. If the device detects that the bus is busy, it cannot begin a new message until the bus returns to an Idle state.

However, two master devices may try to initiate a transmission on or about the same time. When this occurs, the process of arbitration begins. Each transmitter checks the level of the SDA data line and compares it to the level that it expects to find. The first transmitter to observe that the two levels do not match, loses arbitration, and must stop transmitting on the SDA line.

For example, if one transmitter holds the SDA line to a logical one (lets it float) and a second transmitter holds it to a logical zero (pulls it low), the result is that the SDA line will be low. The first transmitter then observes that the level of the line is different than expected and concludes that another transmitter is communicating.

The first transmitter to notice this difference is the one that loses arbitration and must stop driving the SDA line. If this transmitter is also a master device, it also must stop driving the SCL line. It then can monitor the lines for a Stop condition before trying to reissue its transmission. In the meantime, the other device that has not noticed any difference between the expected and actual levels on the SDA line continues with its original transmission. It can do so without any complications, because so far, the transmission appears exactly as expected with no other transmitter disturbing the message.

Slave Transmit mode can also be arbitrated, when a master addresses multiple slaves, but this is less common.

If two master devices are sending a message to two different slave devices at the address stage, the master sending the lower slave address always wins arbitration. When two master devices send messages to the same slave address, and addresses can sometimes refer to multiple slaves, the arbitration process must continue into the data stage.

Arbitration usually occurs very rarely, but it is a necessary process for proper multi-master support.

29.4.3 Byte Format

All communication in I²C is done in 9-bit segments. A byte is sent from a master to a slave or vice versa, followed by an Acknowledge bit sent back. After the eighth falling edge of the SCL line, the device outputting data on the SDA changes that pin to an input and reads in an acknowledge value on the next clock pulse.

The clock signal, SCL, is provided by the master. Data is valid to change while the SCL signal is low, and sampled on the rising edge of the clock. Changes on the SDA line while the SCL line is high define special conditions on the bus, explained below.

29.4.4 Definition of I²C Terminology

There is language and terminology in the description of I²C communication that have definitions specific to I²C. That word usage is defined below and may be used in the rest of this document without explanation. This table was adapted from the Philips I²C specification.

| TERM | Description |
|------------------|---|
| Transmitter | The device that shifts data out onto the bus. |
| Receiver | The device that shifts data in from the bus. |
| Master | The device that initiates a transfer, generates clock signals and terminates a transfer. |
| Slave | The device addressed by the master. |
| Multi-master | A bus with more than one device that can initiate data transfers. |
| Arbitration | Procedure to ensure that only one master at a time controls the bus. Winning arbitration ensures that the message is not corrupted. |
| Synchronization | Procedure to synchronize the clocks of two or more devices on the bus. |
| Idle | No master is controlling the bus, and both SDA and SCL lines are high. |
| Active | Any time one or more master devices are controlling the bus. |
| Addressed Slave | Slave device that has received a matching address and is actively being clocked by a master. |
| Matching Address | Address byte that is clocked into a slave that matches the value stored in SSPxADD. |
| Write Request | Slave receives a matching address with R/W bit clear, and is ready to clock in data. |
| Read Request | Master sends an address byte with the R/W bit set, indicating that it wishes to clock data out of the Slave. This data is the next and all following bytes until a Restart or Stop. |
| Clock Stretching | When a device on the bus hold SCL low to stall communication. |
| Bus Collision | Any time the SDA line is sampled low by the module while it is outputting and expected high state. |

29.4.5 SDA and SCL Pins

Selection of any I²C mode with the [SSPEN](#) bit set, forces the SCL and SDA pins to be open-drain. These pins should be set by the user to inputs by setting the appropriate TRIS bits.

Note:

1. SDA is tied to output zero when an I²C mode is enabled.
2. Any device pin can be selected for SDA and SCL functions with the PPS peripheral. These functions are bidirectional. The SDA input is selected with the SSPxDATPPS registers. The SCL input is selected with the SSPxCLKPPS registers. Outputs are selected with the RxyPPS registers. It is the user's responsibility to make the selections so that both the input and the output for each function is on the same pin.

29.4.6 SDA Hold Time

The hold time of the SDA pin is selected by the [SDAHT](#) bit. Hold time is the time SDA is held valid after the falling edge of SCL. Setting the SDAHT bit selects a longer 300 ns minimum hold time and may help on buses with large capacitance.

29.4.7 Start Condition

The I²C specification defines a Start condition as a transition of SDA from a high to a low state while SCL line is high. A Start condition is always generated by the master and signifies the transition of the bus from an Idle to an Active state. Figure 29-12 shows wave forms for Start and Stop conditions.

A bus collision can occur on a Start condition if the module samples the SDA line low before asserting it low. This does not conform to the I²C Specification that states no bus collision can occur on a Start.

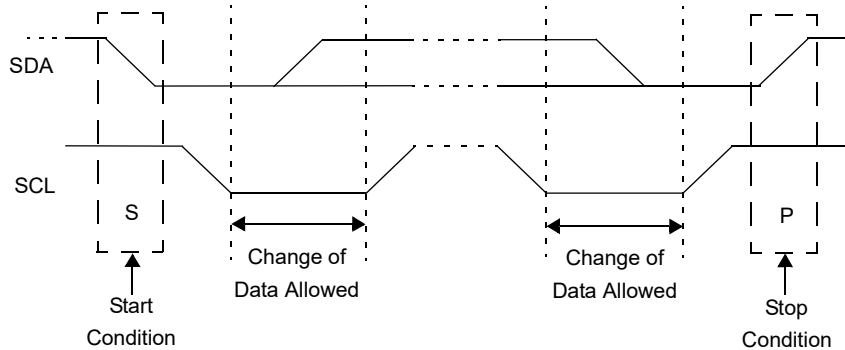
29.4.8 Stop Condition

A Stop condition is a transition of the SDA line from low-to-high state while the SCL line is high.



Important: At least one SCL low time must appear before a Stop is valid, therefore, if the SDA line goes low then high again while the SCL line stays high, only the Start condition is detected.

Figure 29-12. I²C Start and Stop Conditions



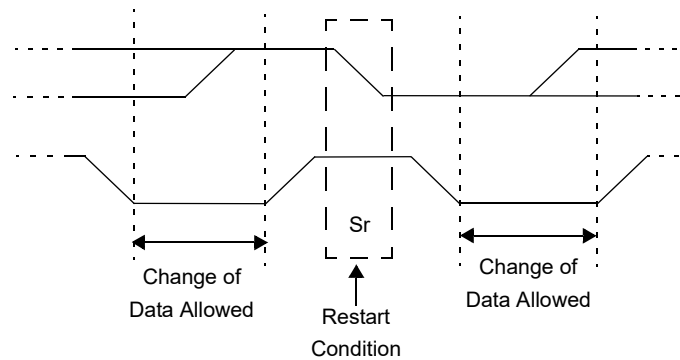
29.4.9 Restart Condition

A Restart is valid any time that a Stop would be valid. A master can issue a Restart if it wishes to hold the bus after terminating the current transfer. A Restart has the same effect on the slave that a Start would, resetting all slave logic and preparing it to clock in an address. The master may want to address the same or another slave. Figure 29-13 shows the wave form for a Restart condition.

In 10-bit Addressing Slave mode a Restart is required for the master to clock data out of the addressed slave. Once a slave has been fully addressed, matching both high and low address bytes, the master can issue a Restart and the high address byte with the R/W bit set. The slave logic will then hold the clock and prepare to clock out data.

After a full match with R/W clear in 10-bit mode, a prior match flag is set and maintained until a Stop condition, a high address with R/W clear, or high address match fails.

Figure 29-13. I²C Restart Condition



29.4.10 Start/Stop Condition Interrupt Masking

The [SCIE](#) and [PCIE](#) bits can enable the generation of an interrupt in Slave modes that do not typically support this function. These bits will have no effect in Slave modes where interrupt on Start and Stop detect are already enabled.

29.4.11 Acknowledge Sequence

The ninth SCL pulse for any transferred byte in I²C is dedicated as an Acknowledge. It allows receiving devices to respond back to the transmitter by pulling the SDA line low. The transmitter must release control of the line during this time to shift in the response. The Acknowledge (\overline{ACK}) is an active-low signal, pulling the SDA line low indicates to the transmitter that the device has received the transmitted data and is ready to receive more.

The result of an \overline{ACK} is placed in the [ACKSTAT](#) bit in the SSPxCON2 register.

The slave software, when the [AHEN](#) and [DHEN](#) bits in the SSPxCON3 register are set, allows the user to set the \overline{ACK} value sent back to the transmitter. The [ACKDT](#) bit in the SSPxCON2 register is set/cleared to determine the response.

The slave hardware will generate an \overline{ACK} response if both the AHEN and DHEN bits are clear. However, if the [BF](#) bit in the SSPxSTAT register or the [SSPOV](#) bit in the SSPxCON1 register are set when a byte is received then the \overline{ACK} will not be sent by the slave.

When the module is addressed, after the eighth falling edge of SCL on the bus, the [ACKTIM](#) bit in the SSPxCON3 register is set. The ACKTIM bit indicates the acknowledge time of the active bus. The ACKTIM Status bit is only active when either the AHEN bit or DHEN bit is enabled.

29.5 I²C Slave Mode Operation

The MSSP Slave mode operates in one of four modes selected by the [SSPM](#) bits. The modes can be divided into 7-bit and 10-bit Addressing mode. 10-bit Addressing modes operate the same as 7-bit with some additional overhead for handling the larger addresses.

Modes with Start and Stop bit interrupts operate the same as the other modes with SSPxIF additionally getting set upon detection of a Start, Restart, or Stop condition.

29.5.1 Slave Mode Addresses

The SSPxADD register contains the Slave mode address. The first byte received after a Start or Restart condition is compared against the value stored in this register. If the byte matches, the value is loaded into the SSPxBUF register and an interrupt is generated. If the value does not match, the module goes Idle and no indication is given to the software that anything happened.

The SSPxMSK register affects the address matching process. See [29.5.9 SSP Mask Register](#) for more information.

29.5.1.1 I²C Slave 7-bit Addressing Mode

In 7-bit Addressing mode, the LSB of the received data byte is ignored when determining if there is an address match.

29.5.1.2 I²C Slave 10-bit Addressing Mode

In 10-bit Addressing mode, the first received byte is compared to the binary value of '1 1 1 0 A9 A8 0'. A9 and A8 are the two MSBs of the 10-bit address and stored in bits 2 and 1 of the SSPxADD register.

After the acknowledge of the high byte the **UA** bit is set and SCL is held low until the user updates SSPxADD with the low address. The low address byte is clocked in and all eight bits are compared to the low address value in SSPxADD. Even if there is not an address match; SSPxIF and UA are set, and SCL is held low until SSPxADD is updated to receive a high byte again. When SSPxADD is updated the UA bit is cleared. This ensures the module is ready to receive the high address byte on the next communication.

A high and low address match as a write request is required at the start of all 10-bit addressing communication. A transmission can be initiated by issuing a Restart once the slave is addressed, and clocking in the high address with the R/W bit set. The slave hardware will then acknowledge the read request and prepare to clock out data. This is only valid for a slave after it has received a complete high and low address byte match.

29.5.2 Slave Reception

When the R/W bit of a matching received address byte is clear, the **R/W** bit is cleared. The received address is loaded into the SSPxBUF register and acknowledged.

When the overflow condition exists for a received address, then Not Acknowledge (NACK) is given. An overflow condition is defined as either the **BF** bit is set, or the **SSPOV** bit is set. The **BOEN** bit modifies this operation. For more information see SSPxCON3.

An MSSP interrupt is generated for each transferred data byte. The SSPxIF flag bit must be cleared by software.

When the **SEN** bit is set, SCL will be held low (clock stretch) following each received byte. The clock must be released by setting the **CKP** bit, except sometimes in 10-bit mode. See [29.5.6.2 10-bit Addressing Mode](#) for more details.

29.5.2.1 7-bit Addressing Reception

This section describes a standard sequence of events for the MSSP module configured as an I²C slave in 7-bit Addressing mode. [Figure 29-14](#) and [Figure 29-15](#) are used as a visual reference for this description.

This is a step by step process of what typically must be done to accomplish I²C communication.

1. Start bit detected.
2. **S** bit is set; SSPxIF is set if interrupt on Start detect is enabled.
3. Matching address with **R/W** bit clear is received.
4. The slave pulls SDA low sending an $\overline{\text{ACK}}$ to the master, and sets SSPxIF bit.
5. Software clears the SSPxIF bit.
6. Software reads received address from SSPxBUF clearing the BF flag.
7. If **SEN** = 1; Slave software sets **CKP** bit to release the SCL line.
8. The master clocks out a data byte.
9. Slave drives SDA low sending an $\overline{\text{ACK}}$ to the master, and sets SSPxIF bit.
10. Software clears SSPxIF.
11. Software reads the received byte from SSPxBUF clearing BF.
12. Steps 8-12 are repeated for all received bytes from the master.
13. Master sends Stop condition, setting **P** bit, and the bus goes Idle.

29.5.2.2 7-bit Reception with AHEN and DHEN

Slave device reception with AHEN and DHEN set operate the same as without these options with extra interrupts and clock stretching added after the eighth falling edge of SCL. These additional interrupts allow the slave software to decide whether it wants to $\overline{\text{ACK}}$ the receive address or data byte, rather than the hardware. This functionality adds support for PMBus™ that was not present on previous versions of this module.

This list describes the steps that need to be taken by slave software to use these options for I²C communication. [Figure 29-16](#) displays a module using both address and data holding. [Figure 29-17](#) includes the operation with the SEN bit of the SSPxCON2 register set.

1. **S** bit is set; SSPxIF is set if interrupt on Start detect is enabled.

2. Matching address with **R/W** bit clear is clocked in. **SSPxIF** is set and **CKP** cleared after the eighth falling edge of SCL.
3. Slave clears the **SSPxIF**.
4. Slave can look at the **ACKTIM** bit to determine if the **SSPxIF** was after or before the $\overline{\text{ACK}}$.
5. Slave reads the address value from **SSPxBUF**, clearing the **BF** flag.
6. Slave sets $\overline{\text{ACK}}$ value clocked out to the master by setting **ACKDT**.
7. Slave releases the clock by setting **CKP**.
8. **SSPxIF** is set after an $\overline{\text{ACK}}$, not after a **NACK**.
9. If **SEN** = 1, the slave hardware will stretch the clock after the $\overline{\text{ACK}}$.
10. Slave clears **SSPxIF**.



Important: **SSPxIF** is still set after the ninth falling edge of SCL even if there is no clock stretching and **BF** has been cleared. Only if **NACK** is sent to master is **SSPxIF** not set

11. **SSPxIF** set and **CKP** cleared after eighth falling edge of SCL for a received data byte.
12. Slave looks at **ACKTIM** bit to determine the source of the interrupt.
13. Slave reads the received data from **SSPxBUF** clearing **BF**.
14. Steps 7-14 are the same for each received data byte.
15. Communication is ended by either the slave sending an $\overline{\text{ACK}} = 1$, or the master sending a Stop condition. If a Stop is sent and Interrupt on Stop Detect is disabled, the slave will only know by polling the **P** bit.

Figure 29-14. I²C Slave, 7-bit Address, Reception (SEN = 0, AHEN = 0, DHEN = 0)

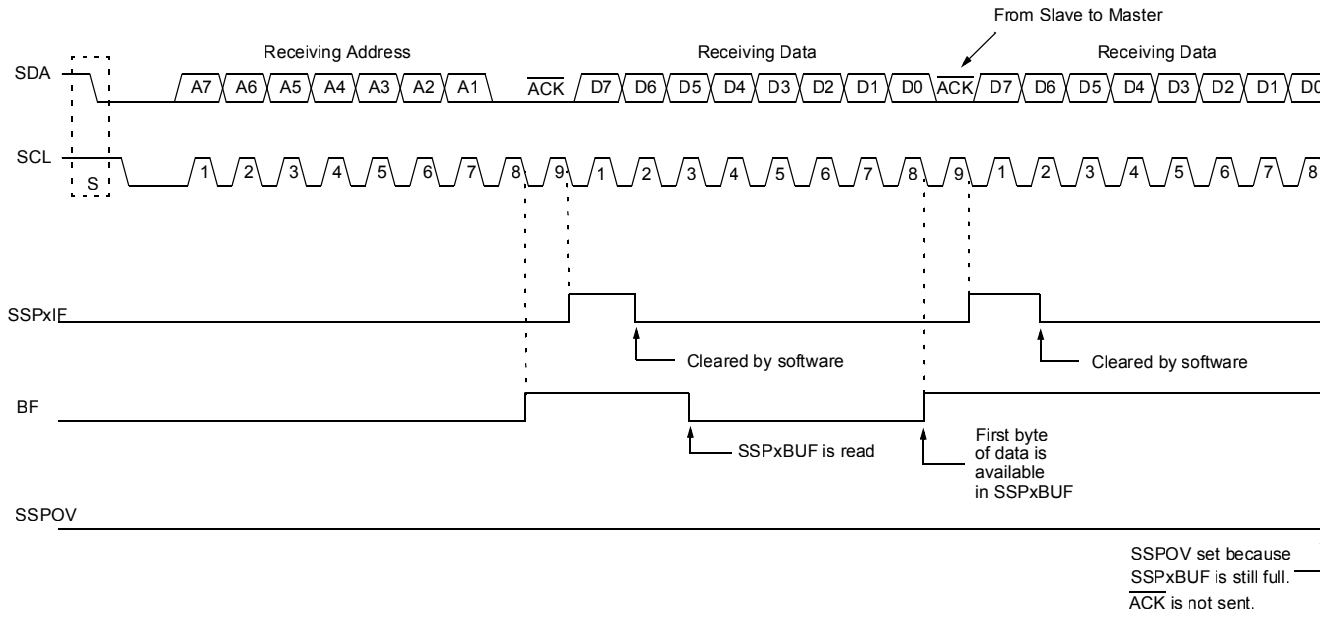


Figure 29-15. I²C Slave, 7-bit Address, Reception (SEN = 1, AHEN = 0, DHEN = 0)

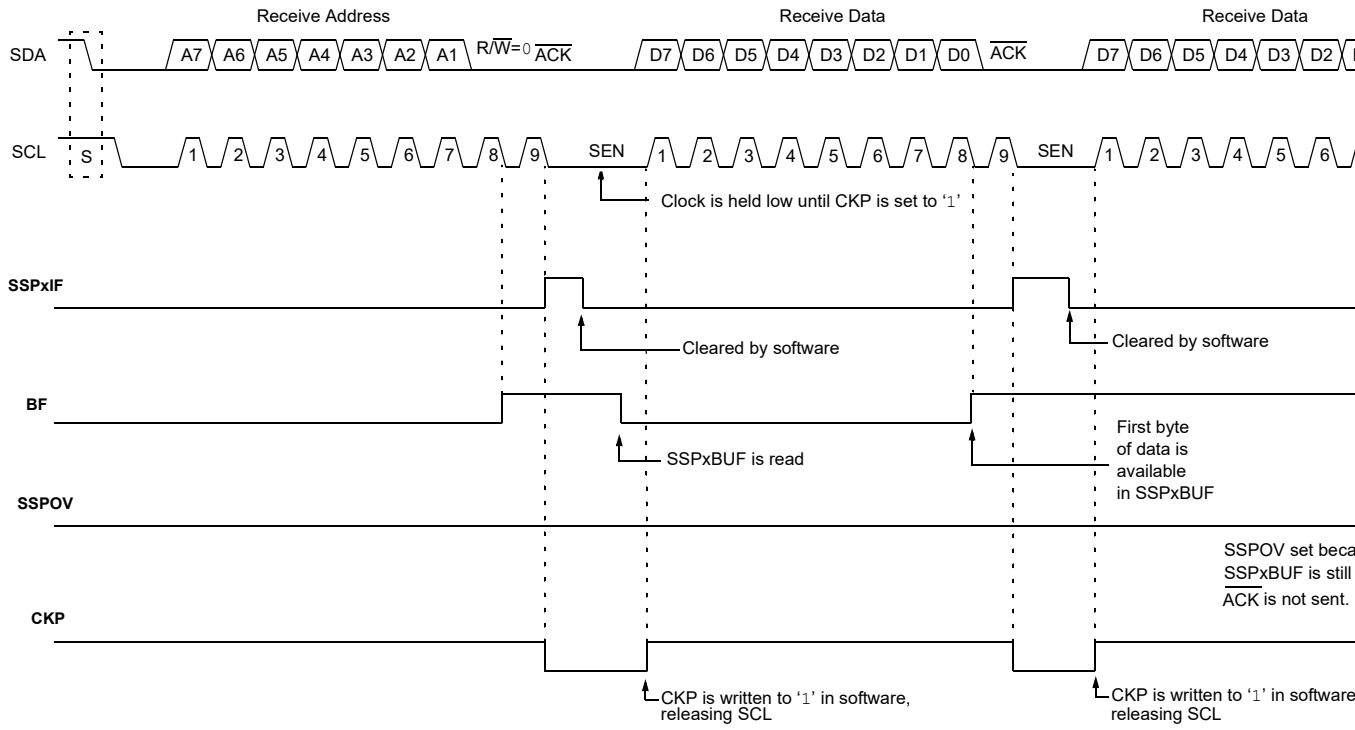


Figure 29-16. I²C Slave, 7-bit Address, Reception (SEN = 0, AHEN = 1, DHEN = 1)

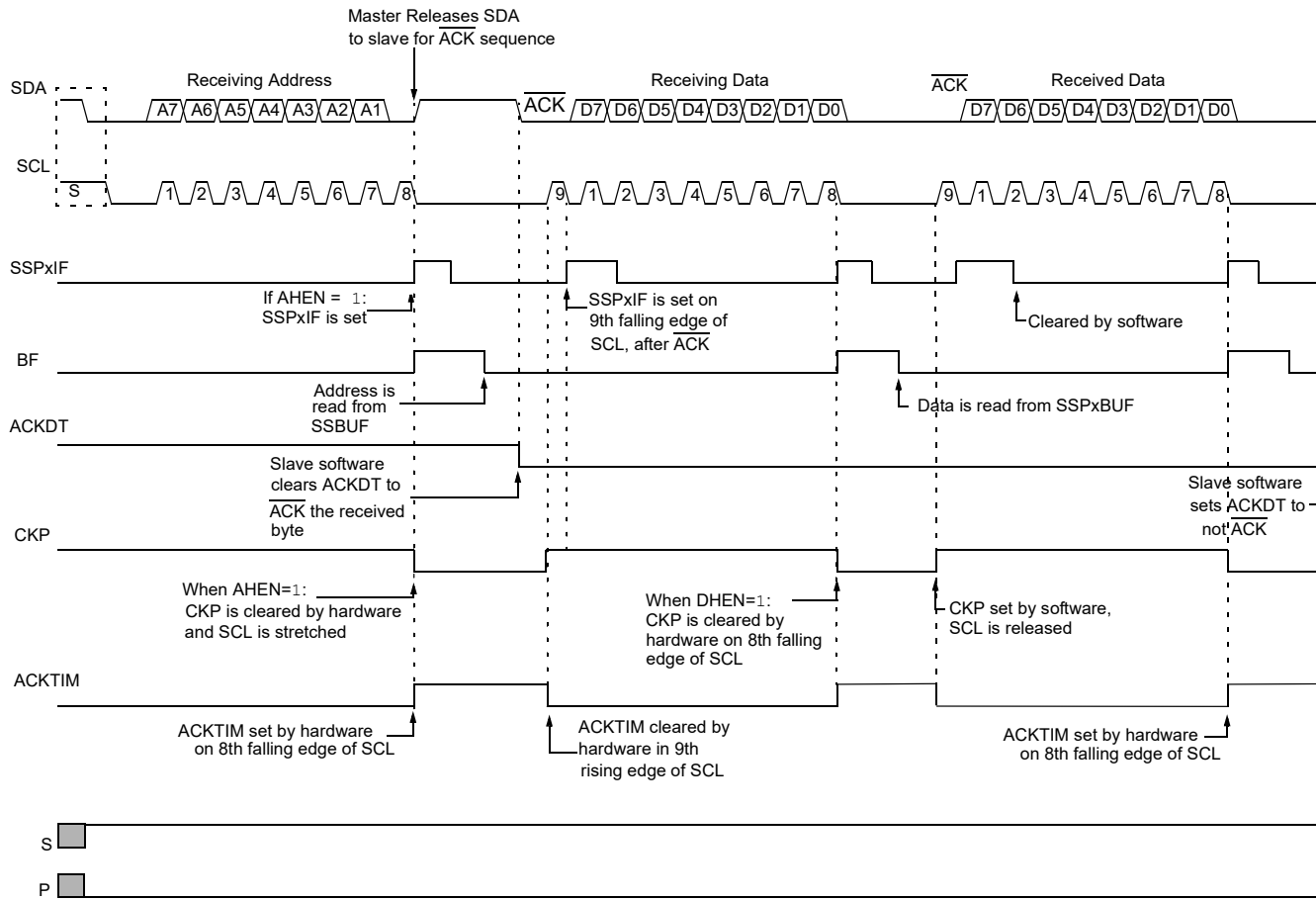
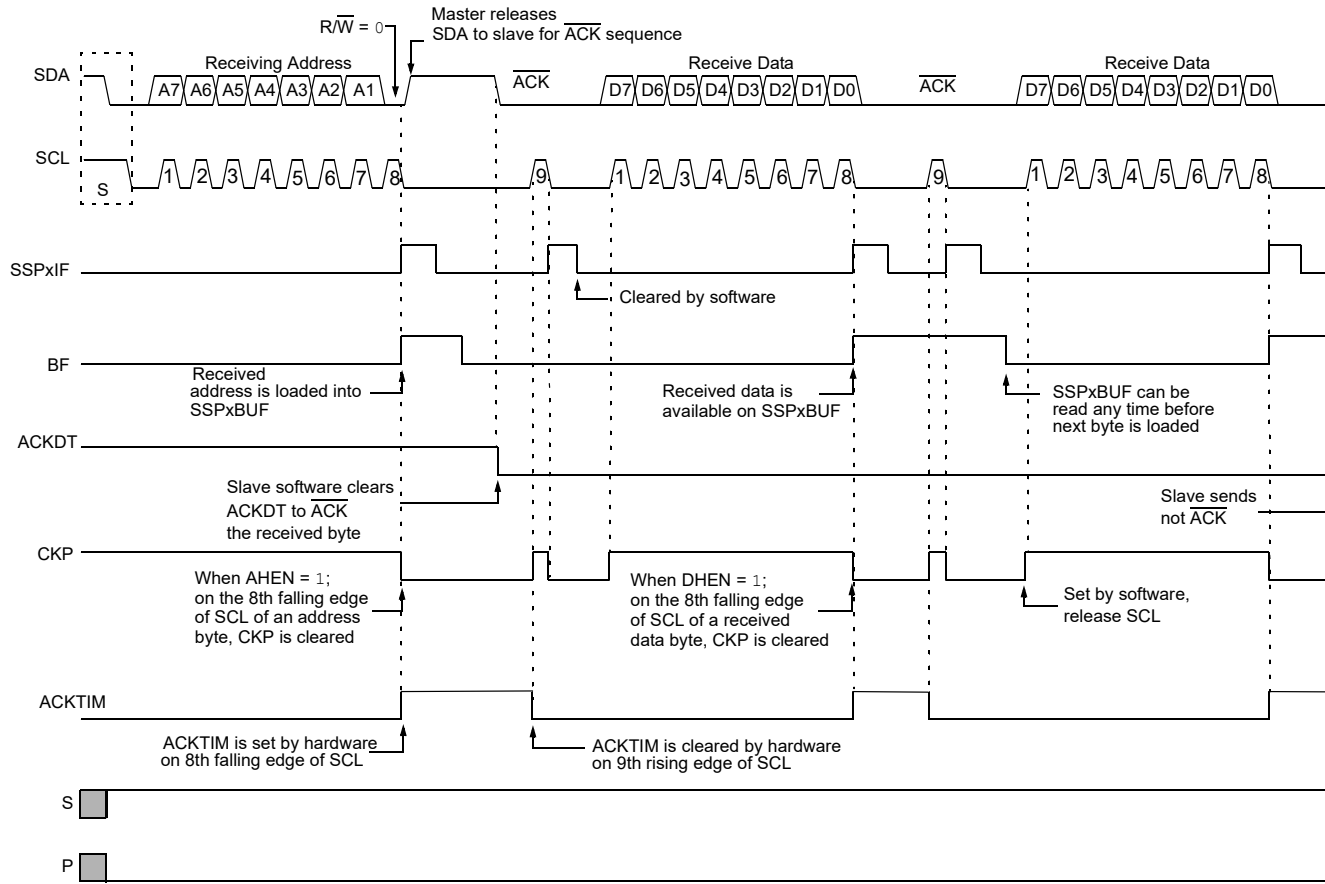


Figure 29-17. I²C Slave, 7-bit Address, Reception (SEN = 1, AHEN = 1, DHEN = 1)



29.5.3 Slave Transmission

When the R/\overline{W} bit of the incoming address byte is set and an address match occurs, the R/W bit is set. The received address is loaded into the SSPxBUF register, and an \overline{ACK} pulse is sent by the slave on the ninth bit.

Following the \overline{ACK} , slave hardware clears the CKP bit and the SCL pin is held low (see 29.5.6 Clock Stretching for more details). By stretching the clock, the master will be unable to assert another clock pulse until the slave is done preparing the transmit data.

The transmit data must be loaded into the SSPxBUF register, which also loads the SSPSR register. Then the SCL pin should be released by setting the CKP bit. The eight data bits are shifted out on the falling edge of the SCL input. This ensures that the SDA signal is valid during the SCL high time.

The \overline{ACK} pulse from the master receiver is latched on the rising edge of the ninth SCL input pulse. This \overline{ACK} value is copied to the $ACKSTAT$ bit. If $ACKSTAT$ is set (not \overline{ACK}), then the data transfer is complete. In this case, when the not \overline{ACK} is latched by the slave, the slave goes Idle and waits for another occurrence of the Start bit. If the SDA line was low (\overline{ACK}), the next transmit data must be loaded into the SSPxBUF register. Again, the SCL pin must be released by setting bit CKP .

An MSSP interrupt is generated for each data transfer byte. The SSPxIF bit must be cleared by software and the SSPxSTAT register is used to determine the status of the byte. The SSPxIF bit is set on the falling edge of the ninth clock pulse.

29.5.3.1 Slave Mode Bus Collision

A slave receives a read request and begins shifting data out on the SDA line. If a bus collision is detected and the $SBCDE$ bit is set, the BCLxIF bit of the PIR3 register is set. Once a bus collision is detected, the slave goes Idle and waits to be addressed again. User software can use the BCLxIF bit to handle a slave bus collision.

29.5.3.2 7-bit Transmission

A master device can transmit a read request to a slave, and then clock data out of the slave. The list below outlines what software for a slave will need to do to accomplish a standard transmission. Figure 29-18 can be used as a reference to this list.

1. Master sends a Start condition on SDA and SCL.
2. S bit is set; SSPxIF is set if interrupt on Start detect is enabled.
3. Matching address with R/\overline{W} bit set is received by the Slave setting SSPxIF bit.
4. Slave hardware generates an \overline{ACK} and sets SSPxIF.
5. SSPxIF bit is cleared by user.
6. Software reads the received address from SSPxBUF, clearing BF.
7. R/\overline{W} is set so CKP was automatically cleared after the \overline{ACK} .
8. The slave software loads the transmit data into SSPxBUF.
9. CKP bit is set releasing SCL, allowing the master to clock the data out of the slave.
10. SSPxIF is set after the \overline{ACK} response from the master is loaded into the $ACKSTAT$ register.
11. SSPxIF bit is cleared.
12. The slave software checks the $ACKSTAT$ bit to see if the master wants to clock out more data.

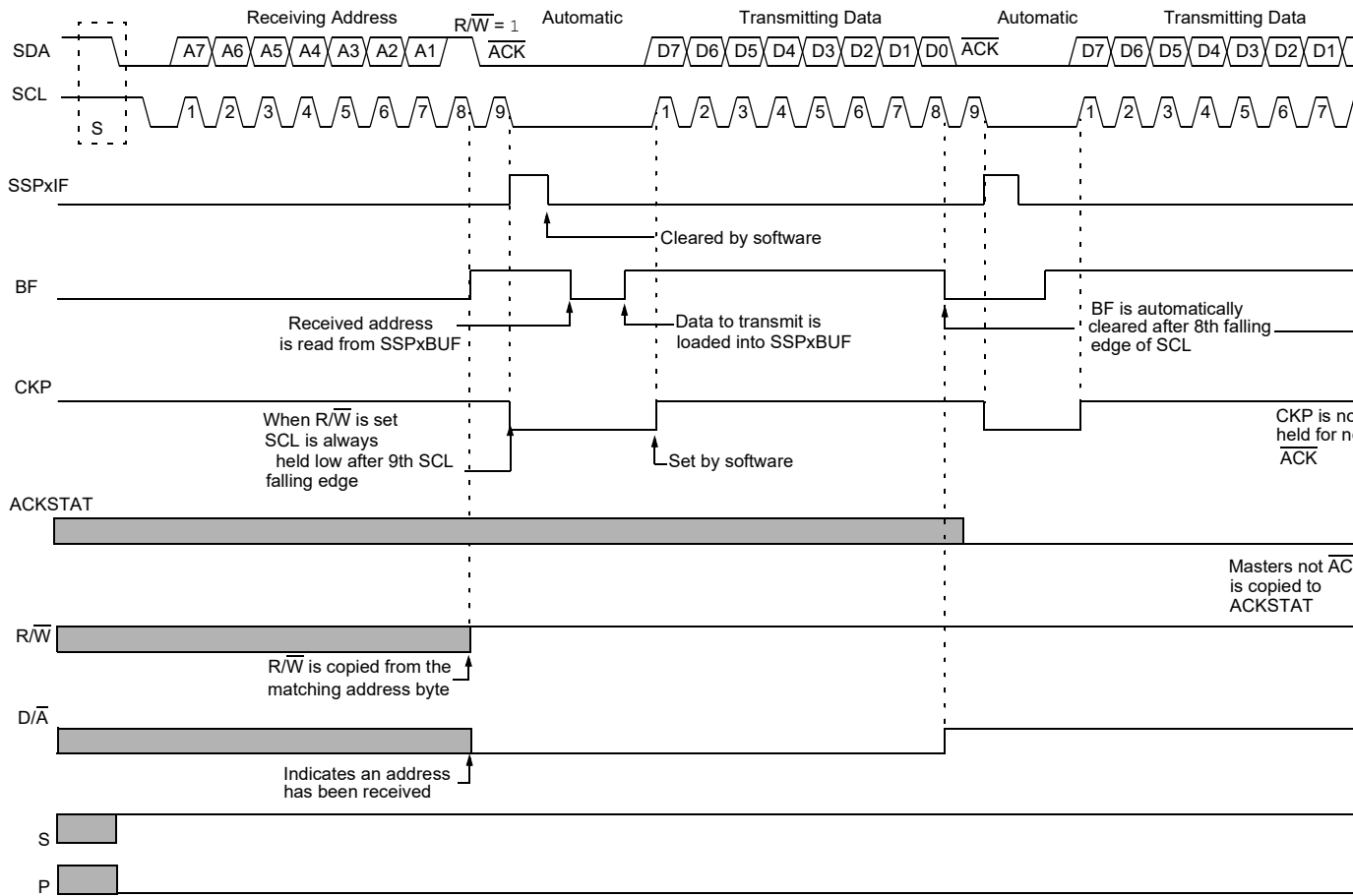


Important:

1. If the master \overline{ACK} s then the clock will be stretched.
2. $ACKSTAT$ is the only bit updated on the rising edge of the ninth SCL clock instead of the falling edge.

13. Steps 9-13 are repeated for each transmitted byte.
14. If the master sends a not \overline{ACK} ; the clock is not held, but SSPxIF is still set.
15. The master sends a Restart condition or a Stop.
16. The slave is no longer addressed.

Figure 29-18. I²C Slave, 7-bit Address, Transmission (AHEN = 0)



29.5.3.3 7-bit Transmission with Address Hold Enabled

Setting the **AHEN** bit enables additional clock stretching and interrupt generation after the eighth falling edge of a received matching address. Once a matching address has been clocked in, **CKP** is cleared and the **SSPxIF** interrupt is set.

Figure 29-19 displays a standard waveform of a 7-bit address slave transmission with **AHEN** enabled.

1. Bus starts Idle.
2. Master sends Start condition; the **S** bit is set; **SSPxIF** is set if interrupt on Start detect is enabled.
3. Master sends matching address with **R/W** bit set. After the eighth falling edge of the **SCL** line the **CKP** bit is cleared and **SSPxIF** interrupt is generated.
4. Slave software clears **SSPxIF**.
5. Slave software reads the **ACKTIM**, **R/W** and **D/A** bits to determine the source of the interrupt.
6. Slave reads the address value from the **SSPxBUF** register clearing the **BF** bit.
7. Slave software decides from this information if it wishes to $\overline{\text{ACK}}$ or not $\overline{\text{ACK}}$ and sets the **ACKDT** bit accordingly.
8. Slave sets the **CKP** bit releasing **SCL**.
9. Master clocks in the $\overline{\text{ACK}}$ value from the slave.
10. Slave hardware automatically clears the **CKP** bit and sets **SSPxIF** after the $\overline{\text{ACK}}$ if the **R/W** bit is set.
11. Slave software clears **SSPxIF**.
12. Slave loads value to transmit to the master into **SSPxBUF** setting the **BF** bit.



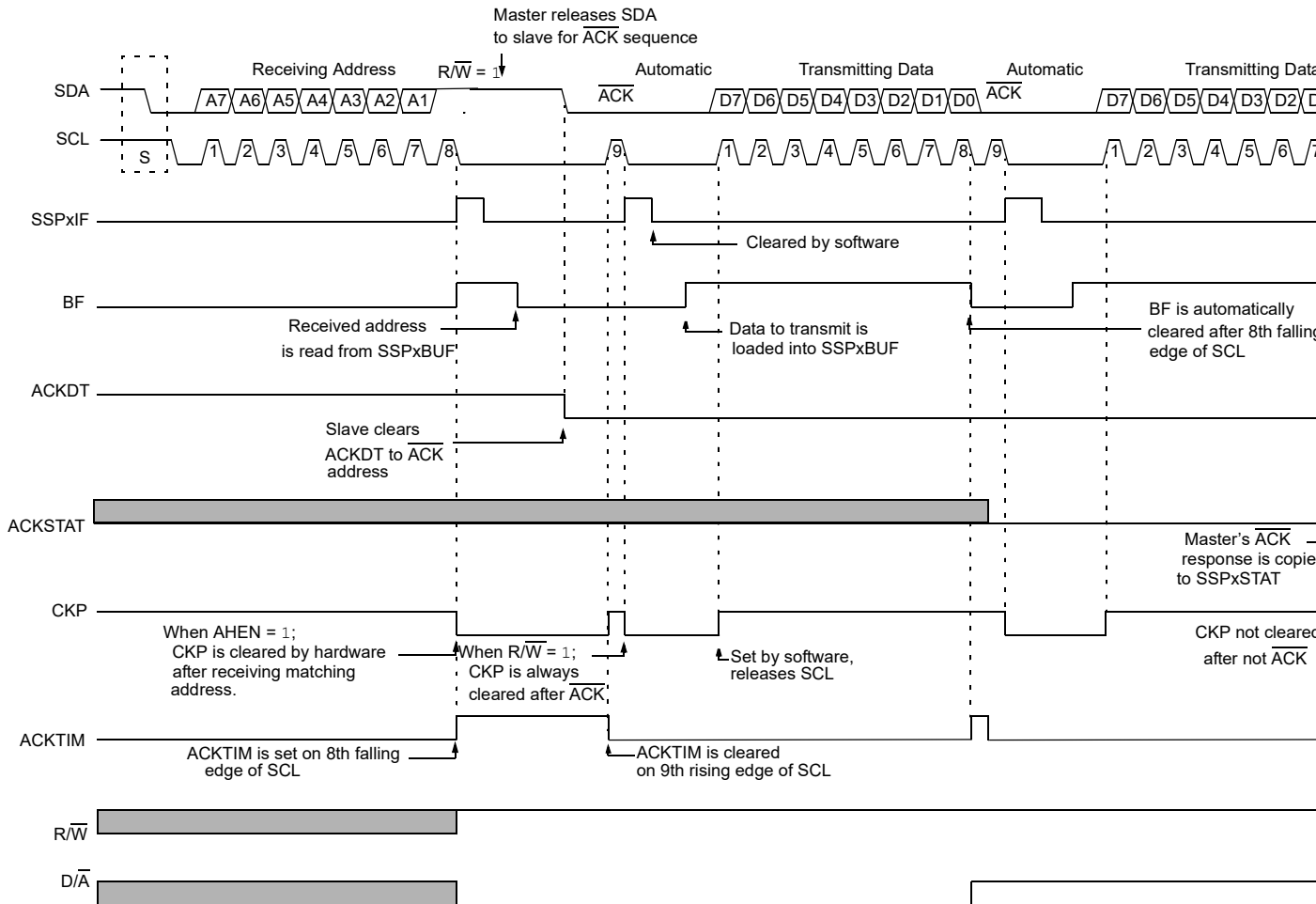
Important: **SSPxBUF** cannot be loaded until after the $\overline{\text{ACK}}$.

13. Slave sets the **CKP** bit releasing the clock.
14. Master clocks out the data from the slave and sends an $\overline{\text{ACK}}$ value on the ninth **SCL** pulse.
15. Slave hardware copies the $\overline{\text{ACK}}$ value into the **ACKSTAT** bit.
16. Steps 10-15 are repeated for each byte transmitted to the master from the slave.
17. If the master sends a not $\overline{\text{ACK}}$ the slave releases the bus allowing the master to send a Stop and end the communication.



Important: Master must send a not $\overline{\text{ACK}}$ on the last byte to ensure that the slave releases the **SCL** line to receive a Stop.

Figure 29-19. I²C Slave, 7-bit Address, Transmission (AHEN = 1)



29.5.4 Slave Mode 10-bit Address Reception

This section describes a standard sequence of events for the MSSP module configured as an I²C slave in 10-bit Addressing mode.

Figure 29-20 is used as a visual reference for this description.

This is a step-by-step process of what must be done by the slave software to accomplish I²C communication.

1. Bus starts Idle.
2. Master sends Start condition; **S** bit is set; SSPxIF is set if interrupt on Start detect is enabled.
3. Master sends matching high address with **R/W** bit clear; **UA** bit is set.
4. Slave sends $\overline{\text{ACK}}$ and SSPxIF is set.
5. Software clears the SSPxIF bit.
6. Software reads received address from SSPxBUF clearing the BF flag.
7. Slave loads low address into SSPxADD, releasing SCL.
8. Master sends matching low address byte to the slave; **UA** bit is set.



Important: Updates to the SSPxADD register are not allowed until after the $\overline{\text{ACK}}$ sequence.

9. Slave sends $\overline{\text{ACK}}$ and SSPxIF is set.



Important: If the low address does not match, SSPxIF and UA are still set so that the slave software can set SSPxADD back to the high address. BF is not set because there is no match. **CKP** is unaffected.

10. Slave clears SSPxIF.
11. Slave reads the received matching address from SSPxBUF clearing BF.
12. Slave loads high address into SSPxADD.
13. Master clocks a data byte to the slave and clocks out the slaves $\overline{\text{ACK}}$ on the ninth SCL pulse; SSPxIF is set.
14. If **SEN** bit is set, **CKP** is cleared by hardware and the clock is stretched.
15. Slave clears SSPxIF.
16. Slave reads the received byte from SSPxBUF clearing BF.
17. If **SEN** is set the slave sets **CKP** to release the SCL.
18. Steps 13-17 repeat for each received byte.
19. Master sends Stop to end the transmission.

29.5.5 10-bit Addressing with Address or Data Hold

Reception using 10-bit addressing with **AHEN** or **DHEN** set is the same as with 7-bit modes. The only difference is the need to update the SSPxADD register using the **UA** bit. All functionality, specifically when the **CKP** bit is cleared and SCL line is held low are the same. Figure 29-21 can be used as a reference of a slave in 10-bit addressing with **AHEN** set.

Figure 29-22 shows a standard waveform for a slave transmitter in 10-bit Addressing mode.

Figure 29-20. I²C Slave, 10-bit Address, Reception (SEN = 1, AHEN = 0, DHEN = 0)

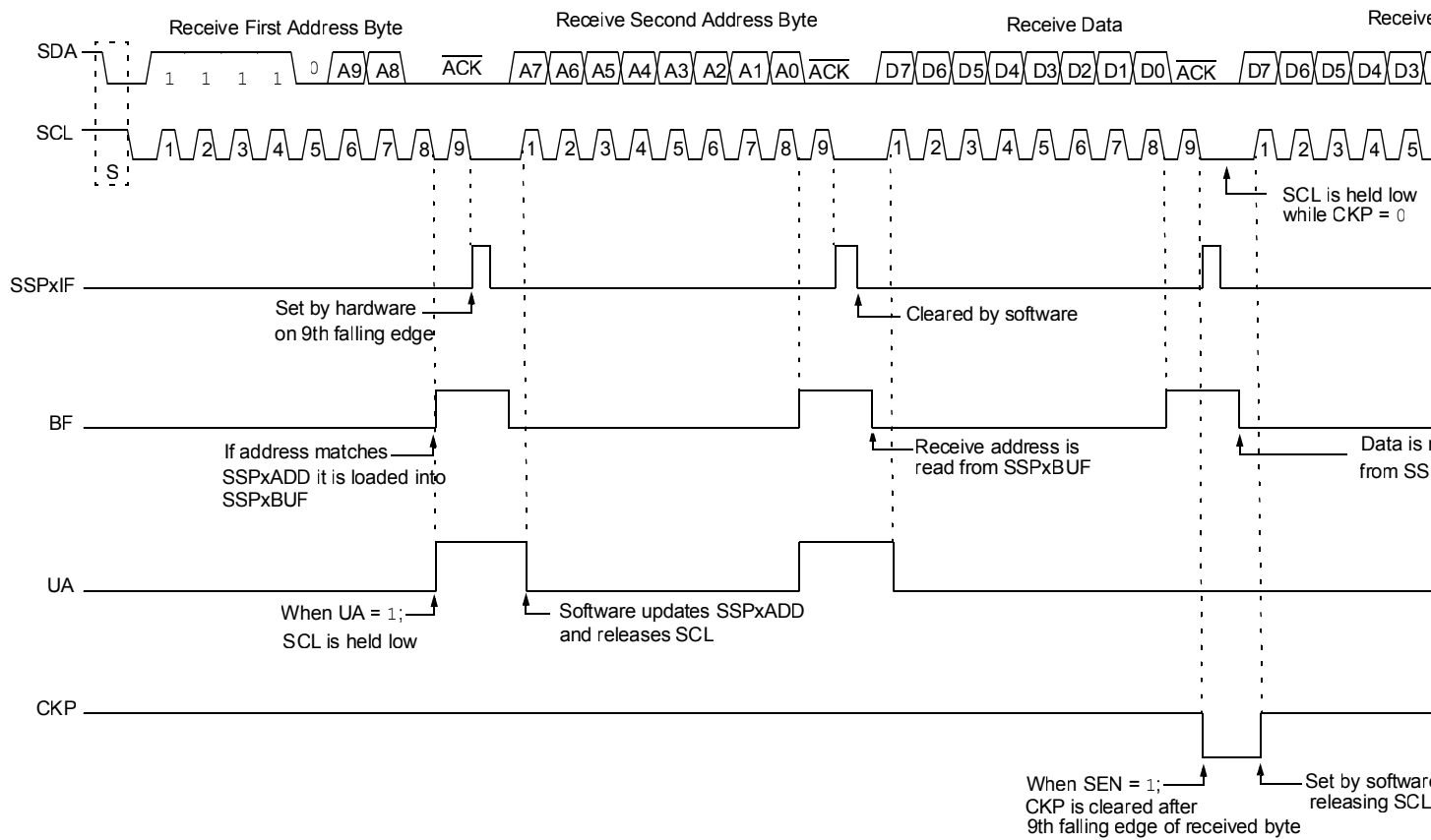


Figure 29-21. I²C Slave, 10-bit Address, Reception (SEN = 0, AHEN = 1, DHEN = 0)

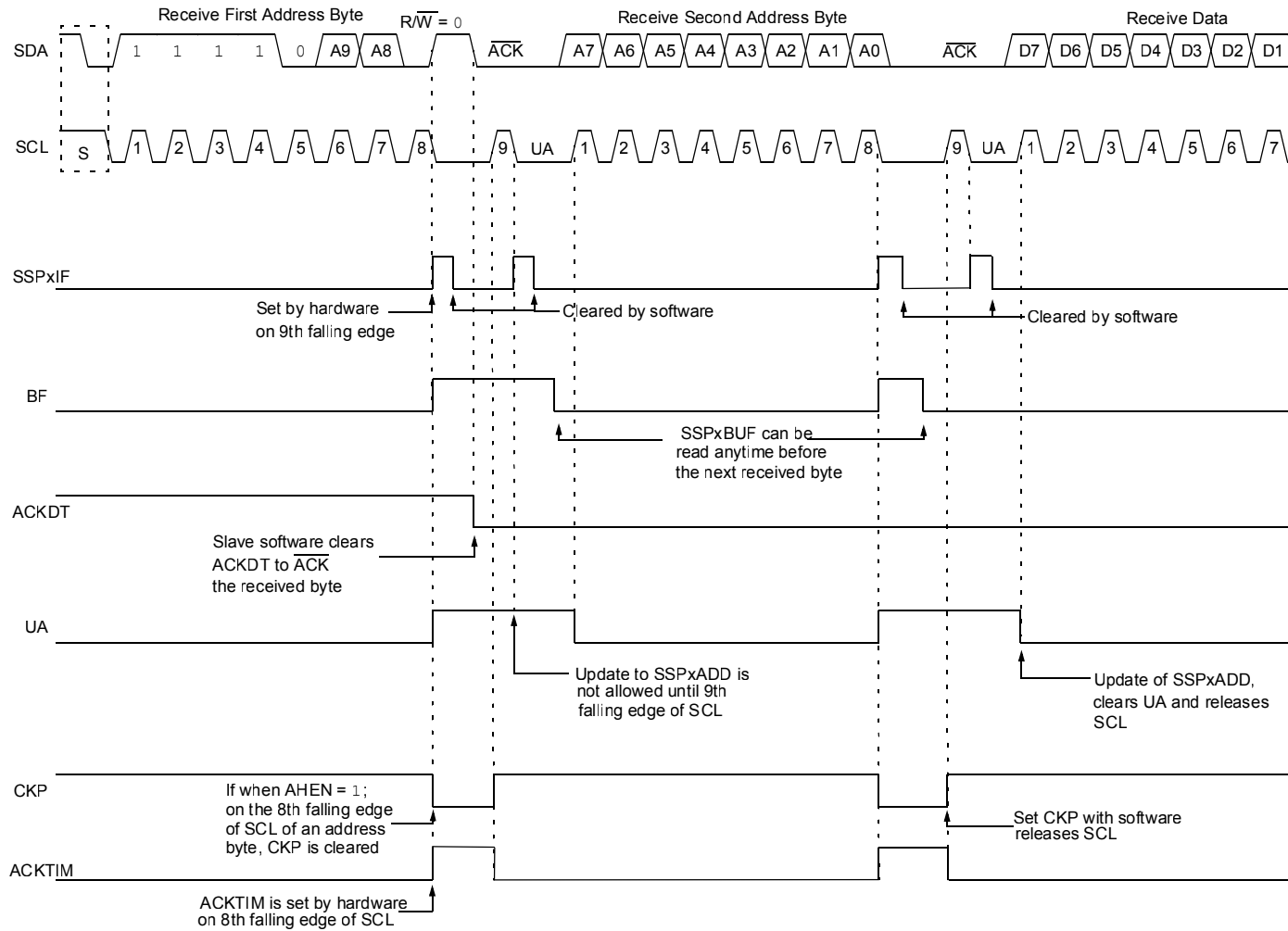
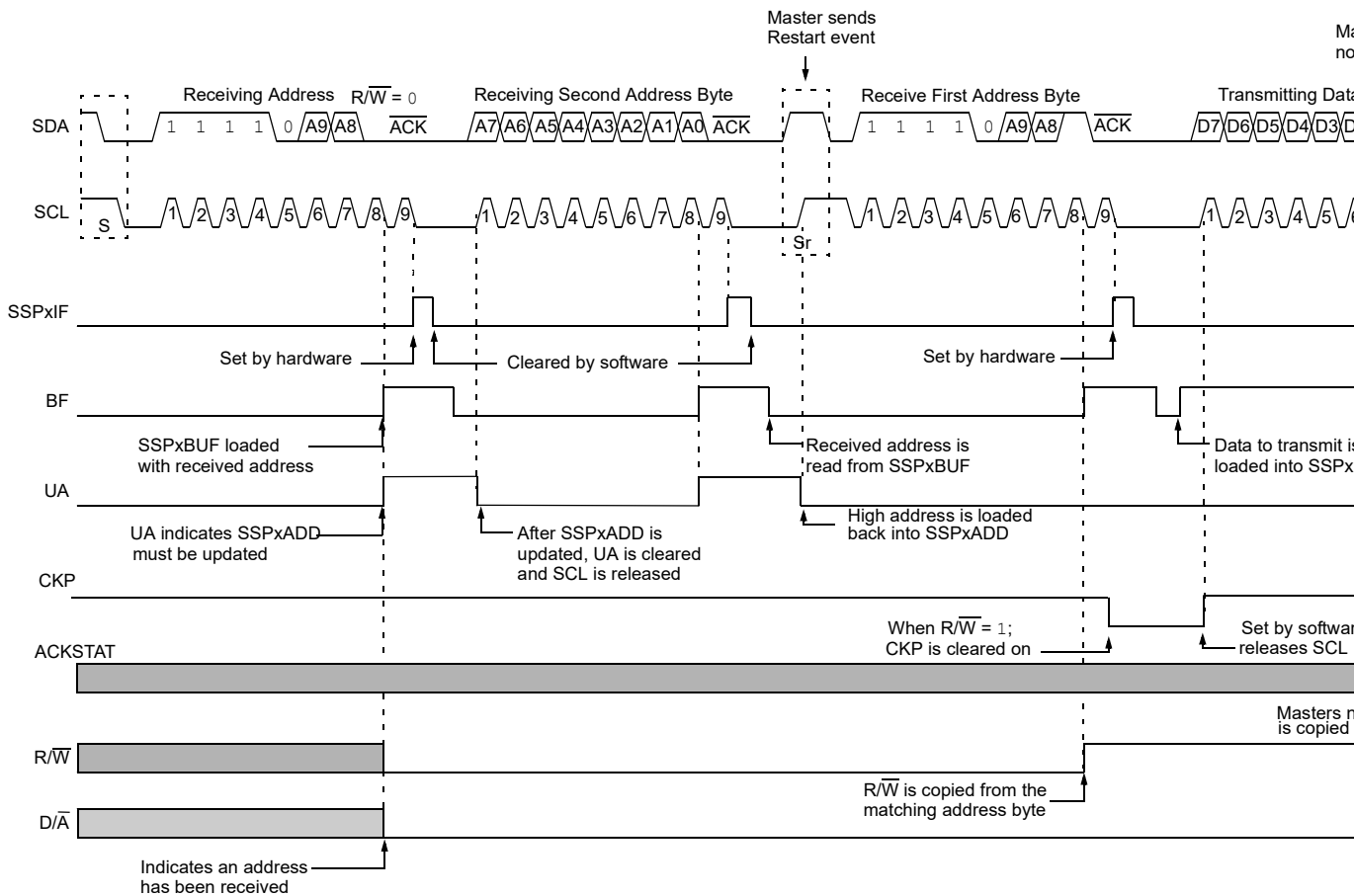


Figure 29-22. I²C Slave, 10-bit Address, Transmission (SEN = 0, AHEN = 0, DHEN = 0)



Ma
no

Masters n
is copied

29.5.6 Clock Stretching

Clock stretching occurs when a device on the bus holds the SCL line low, effectively pausing communication. The slave may stretch the clock to allow more time to handle data or prepare a response for the master device. A master device is not concerned with stretching as anytime it is active on the bus and not transferring data it is stretching. Any stretching done by a slave is invisible to the master software and handled by the hardware that generates SCL.

The **CKP** bit is used to control stretching in software. Any time the CKP bit is cleared, the module will wait for the SCL line to go low and then hold it. Setting CKP will release SCL and allow more communication.

29.5.6.1 Normal Clock Stretching

Following an $\overline{\text{ACK}}$ if the **R/W** bit is set, a read request, the slave hardware will clear CKP. This allows the slave time to update SSPxBUF with data to transfer to the master. If the **SEN** bit is set, the slave hardware will always stretch the clock after the $\overline{\text{ACK}}$ sequence. Once the slave is ready; **CKP** is set by software and communication resumes.



Important:

1. The **BF** bit has no effect on whether or not the clock will be stretched. This is different than previous versions of the module that would not stretch the clock, clear CKP, if SSPxBUF was read before the ninth falling edge of SCL.
2. Previous versions of the module did not stretch the clock for a transmission if SSPxBUF was loaded before the ninth falling edge of SCL. It is now always cleared for read requests.

29.5.6.2 10-bit Addressing Mode

In 10-bit Addressing mode, when the **UA** bit is set, the clock is always stretched. This is the only time the SCL is stretched without CKP being cleared. SCL is released immediately after a write to SSPxADD.



Important: Previous versions of the module did not stretch the clock if the second address byte did not match.

29.5.6.3 Byte NACKing

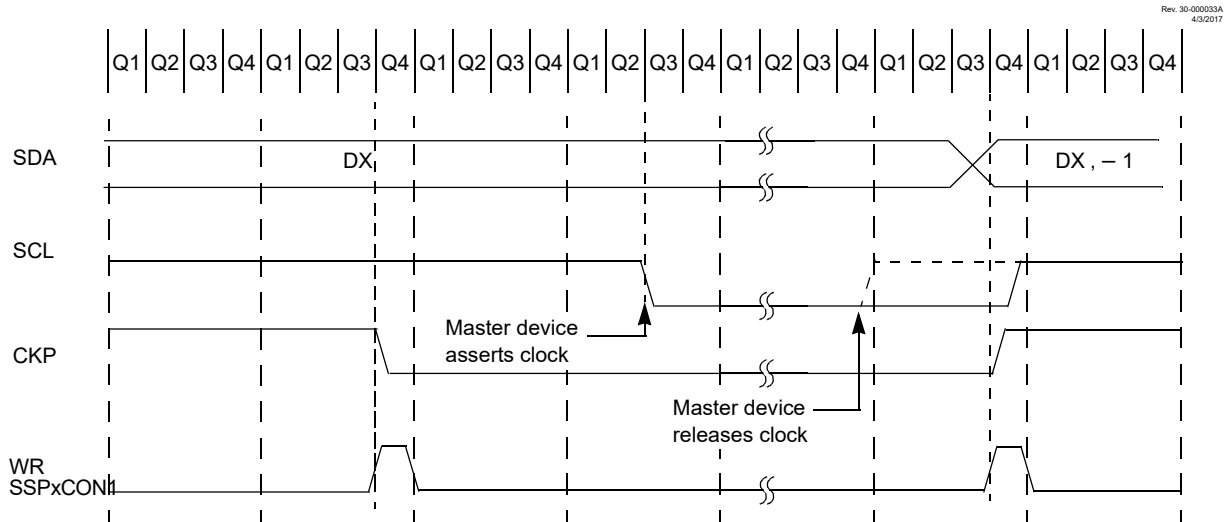
When the **AHEN** bit is set, CKP is cleared by hardware after the eighth falling edge of SCL for a received matching address byte. When the **DHEN** bit is set, CKP is cleared after the eighth falling edge of SCL for received data.

Stretching after the eighth falling edge of SCL allows the slave to look at the received address or data and decide if it wants to ACK the received data.

29.5.7 Clock Synchronization and the CKP bit

Any time the **CKP** bit is cleared, the module will wait for the SCL line to go low and then hold it. However, clearing the CKP bit will not assert the SCL output low until the SCL output is already sampled low. Therefore, the CKP bit will not assert the SCL line until an external I²C master device has already asserted the SCL line. The SCL output will remain low until the CKP bit is set and all other devices on the I²C bus have released SCL. This ensures that a write to the CKP bit will not violate the minimum high time requirement for SCL (see the following figure).

Figure 29-23. Clock Synchronization Timing

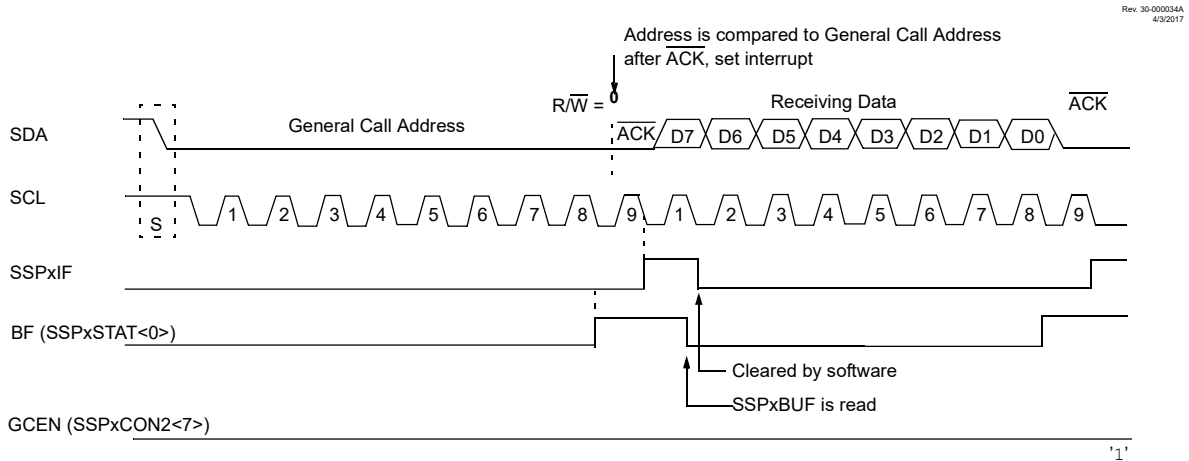


29.5.8 General Call Address Support

The addressing procedure for the I²C bus is such that the first byte after the Start condition usually determines which device will be the slave addressed by the master device. The exception is the general call address that can address all devices. When this address is used, all devices should, in theory, respond with an acknowledge.

The general call address is a reserved address in the I²C protocol, defined as address 0x00. When the GCEN bit is set, the slave module will automatically ACK the reception of this address regardless of the value stored in SSPxADD. After the slave clocks in an address of all zeros with the R/W bit clear, an interrupt is generated and slave software can read SSPxBUF and respond. The following figure shows a general call reception sequence.

Figure 29-24. Slave Mode General Call Address Sequence



In 10-bit Address mode, the UA bit will not be set on the reception of the general call address. The slave will prepare to receive the second byte as data, just as it would in 7-bit mode.

If the AHEN bit is set, just as with any other address reception, the slave hardware will stretch the clock after the eighth falling edge of SCL. The slave must then set its ACKEN value and release the clock with communication progressing as it would normally.

29.5.9 SSP Mask Register

An SSP Mask register (SSPxMSK) is available in I²C Slave mode as a mask for the value held in the SSPSR register during an address comparison operation. A zero ('0') bit in the SSPxMSK register has the effect of making the corresponding bit of the received address a "don't care".

This register is reset to all '1's upon any Reset condition and, therefore, has no effect on standard SSP operation until written with a mask value.

The SSP Mask register is active during:

- 7-bit Address mode: address compare of A[7:1].
- 10-bit Address mode: address compare of A[7:0] only. The SSP mask has no effect during the reception of the first (high) byte of the address.

29.6 I²C Master Mode

Master mode is enabled by setting and clearing the appropriate SSPM bits and setting the SSPEN bit. In Master mode, the SDA and SCK pins must be configured as inputs. The MSSP peripheral hardware will override the output driver TRIS controls when necessary to drive the pins low.

Master mode of operation is supported by interrupt generation on the detection of the Start and Stop conditions. The Stop (P) and Start (S) bits are cleared from a Reset or when the MSSP module is disabled. Control of the I²C bus may be taken when the P bit is set, or the bus is Idle.

In Firmware Controlled Master mode, user code conducts all I²C bus operations based on Start and Stop bit condition detection. Start and Stop condition detection is the only active circuitry in this mode. All other communication is done by the user software directly manipulating the SDA and SCL lines.

The following events will cause the SSP Interrupt Flag bit, SSPxIF, to be set (SSP interrupt, if enabled):

- Start condition detected
- Stop condition detected
- Data transfer byte transmitted/received
- Acknowledge transmitted/received
- Repeated Start generated



Important:

1. The MSSP module, when configured in I²C Master mode, does not allow queuing of events. For instance, the user is not allowed to initiate a Start condition and immediately write the SSPxBUF register to initiate transmission before the Start condition is complete. In this case, the SSPxBUF will not be written to and the WCOL bit will be set, indicating that a write to the SSPxBUF did not occur.
2. Master mode suspends Start/Stop detection when sending the Start/Stop condition by means of the SEN/PEN control bits. The SSPxIF bit is set at the end of the Start/Stop generation when hardware clears the control bit.

29.6.1 I²C Master Mode Operation

The master device generates all of the serial clock pulses and the Start and Stop conditions. A transfer is ended with a Stop condition or with a Repeated Start condition. Since the Repeated Start condition is also the beginning of the next serial transfer, the I²C bus will not be released.

In Master Transmitter mode, serial data is output through SDA, while SCL outputs the serial clock. The first byte transmitted contains the slave address of the receiving device (7 bits) and the R/W bit. In this case, the R/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/W bit. In this case, the R/W bit will be logic '1'. Thus, the first byte transmitted is a 7-bit slave address followed

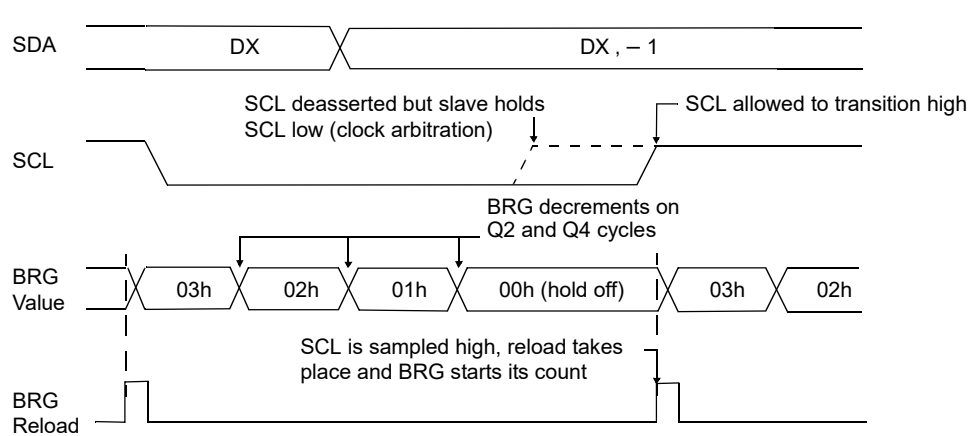
by a '1' to indicate the receive bit. Serial data is received via SDA, while SCL outputs the serial clock. Serial data is received eight bits at a time. After each byte is received, an Acknowledge bit is transmitted. Start and Stop conditions indicate the beginning and end of transmission.

A Baud Rate Generator is used to set the clock frequency output on SCL. See 29.7 Baud Rate Generator for more details.

29.6.2 Clock Arbitration

Clock arbitration occurs when the master, during any receive, transmit or Repeated Start/Stop condition, releases the SCL pin (SCL allowed to float high). When the SCL pin is allowed to float high, the Baud Rate Generator (BRG) is suspended from counting until the SCL pin is actually sampled high. When the SCL pin is sampled high, the Baud Rate Generator is reloaded with the contents of 29.9.2 SSPxADD 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 as shown in the following figure.

Figure 29-25. Baud Rate Generator Timing with Clock Arbitration



29.6.3 WCOL Status Flag

If the user writes the SSPxBUF when a Start, Restart, Stop, Receive or Transmit sequence is in progress, the WCOL bit is set and the contents of the buffer are unchanged (the write does not occur). Any time the WCOL bit is set it indicates that an action on SSPxBUF was attempted while the module was not Idle.



Important: Because queuing of events is not allowed, writing to the lower five bits of SSPxCON2 is disabled until the Start condition is complete.

29.6.4 I²C Master Mode Start Condition Timing

To initiate a Start condition (Figure 29-26), the user sets the SEN Start Enable bit. If the SDA and SCL pins are sampled high, the Baud Rate Generator is reloaded with the contents of SSPxADD and starts its count. If SCL and SDA are both sampled high when the Baud Rate Generator times out (T_{BRG}), 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 to be set. Following this, the Baud Rate Generator is reloaded with the contents of SSPxADD and resumes its count. When the Baud Rate Generator times out (T_{BRG}), the SEN bit will be automatically cleared by hardware; the Baud Rate Generator is suspended, leaving the SDA line held low and the Start condition is complete.

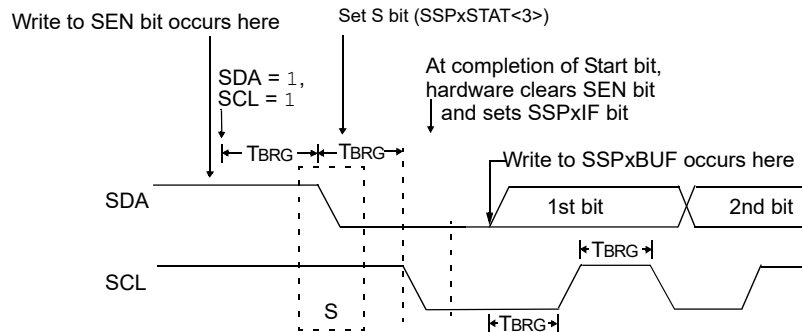


Important:

1. If at the beginning of the Start condition, the SDA and SCL pins are already sampled low, or if during the Start condition, the SCL line is sampled low before the SDA line is driven low, a bus collision occurs, the Bus Collision Interrupt Flag, BCLxIF, is set, the Start condition is aborted and the I²C module is reset into its Idle state.
2. The Philips I²C specification states that a bus collision cannot occur on a Start.

Figure 29-26. First Start Bit Timing

Rev. 30-000036A
4/3/2017



29.6.5 I²C Master Mode Repeated Start Condition Timing

A Repeated Start condition (Figure 29-27) occurs when the RSEN bit is programmed high and the master state machine is no longer active. When the RSEN bit is set, the SCL pin is asserted low. When the SCL pin is sampled low, the Baud Rate Generator is loaded and begins counting. The SDA pin is released (brought high) for one Baud Rate Generator count (T_{BRG}). When the Baud Rate Generator times out, if SDA is sampled high, the SCL pin will be deasserted (brought high). When SCL is sampled high, the Baud Rate Generator is reloaded and begins counting. SDA and SCL must be sampled high for one T_{BRG} . This action is then followed by assertion of the SDA pin ($SDA = 0$) for one T_{BRG} while SCL is high. SCL is asserted low. Following this, the RSEN bit 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 will be set. The SSPxIF bit will not be set until the Baud Rate Generator has timed out.

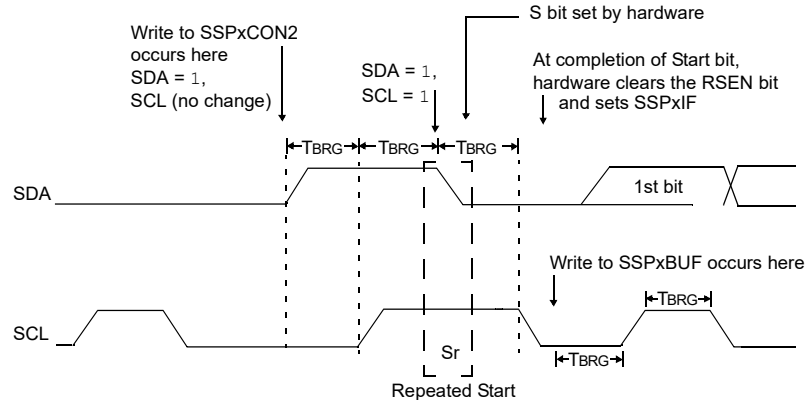


Important:

1. If RSEN is programmed while any other event is in progress, it will not take effect.
2. A bus collision during the Repeated Start condition occurs if:
 - SDA is sampled low when SCL goes from low-to-high.
 - SCL goes low before SDA is asserted low. This may indicate that another master is attempting to transmit a data '1'.

Figure 29-27. Repeated Start Condition Waveform

Rev. 30-000037A
4/10/2017



29.6.6 I²C Master Mode Transmission

Transmission of a data byte, a 7-bit address or the other half of a 10-bit address is accomplished by simply writing a value to the SSPxBUF register. This action will set the Buffer Full flag bit, BF, and allow the Baud Rate Generator to begin counting and start the next transmission. Each bit of address/data will be shifted out onto the SDA pin after the falling edge of SCL is asserted. SCL is held low for one Baud Rate Generator rollover count (T_{BRG}). Data should be valid before SCL is released high. When the SCL pin is released high, it is held that way for T_{BRG} . The data on the SDA pin must remain stable for that duration and some hold time after the next falling edge of SCL. After the eighth bit is shifted out (the falling edge of the eighth clock), the BF flag is cleared and the master releases SDA. This allows the slave device being addressed to respond with an \overline{ACK} bit during the ninth bit time if an address match occurred, or if data was received properly. The status of \overline{ACK} is written into the **ACKSTAT** bit on the rising edge of the ninth clock. If the master receives an Acknowledge, the Acknowledge Status bit, **ACKSTAT**, is cleared. If not, the bit is set. After the ninth clock, the SSPxIF bit is set and the master clock (Baud Rate Generator) is suspended until the next data byte is loaded into the SSPxBUF, leaving SCL low and SDA unchanged (Figure 29-28).

After the write to the SSPxBUF, each bit of the address will be shifted out on the falling edge of SCL until all seven address bits and the R/W bit are completed. On the falling edge of the eighth clock, the master will release the SDA pin, allowing the slave to respond with an Acknowledge. On the falling edge of the ninth clock, the master will sample the SDA pin to see if the address was recognized by a slave. The status of the \overline{ACK} bit is loaded into the **ACKSTAT** Status bit of the SSPxCON2 register. Following the falling edge of the ninth clock transmission of the address, the SSPxIF is set, the BF flag is cleared and the Baud Rate Generator is turned off until another write to the SSPxBUF takes place, holding SCL low and allowing SDA to float.

29.6.6.1 BF Status Flag

In Transmit mode, the **BF** bit is set when the CPU writes to SSPxBUF and is cleared when all eight bits are shifted out.

29.6.6.2 WCOL Status Flag

If the user writes the SSPxBUF when a transmit is already in progress (i.e., SSPSR is still shifting out a data byte), the **WCOL** bit is set and the contents of the buffer are unchanged (the write does not occur).

The **WCOL** bit must be cleared by software before the next transmission.

29.6.6.3 ACKSTAT Status Flag

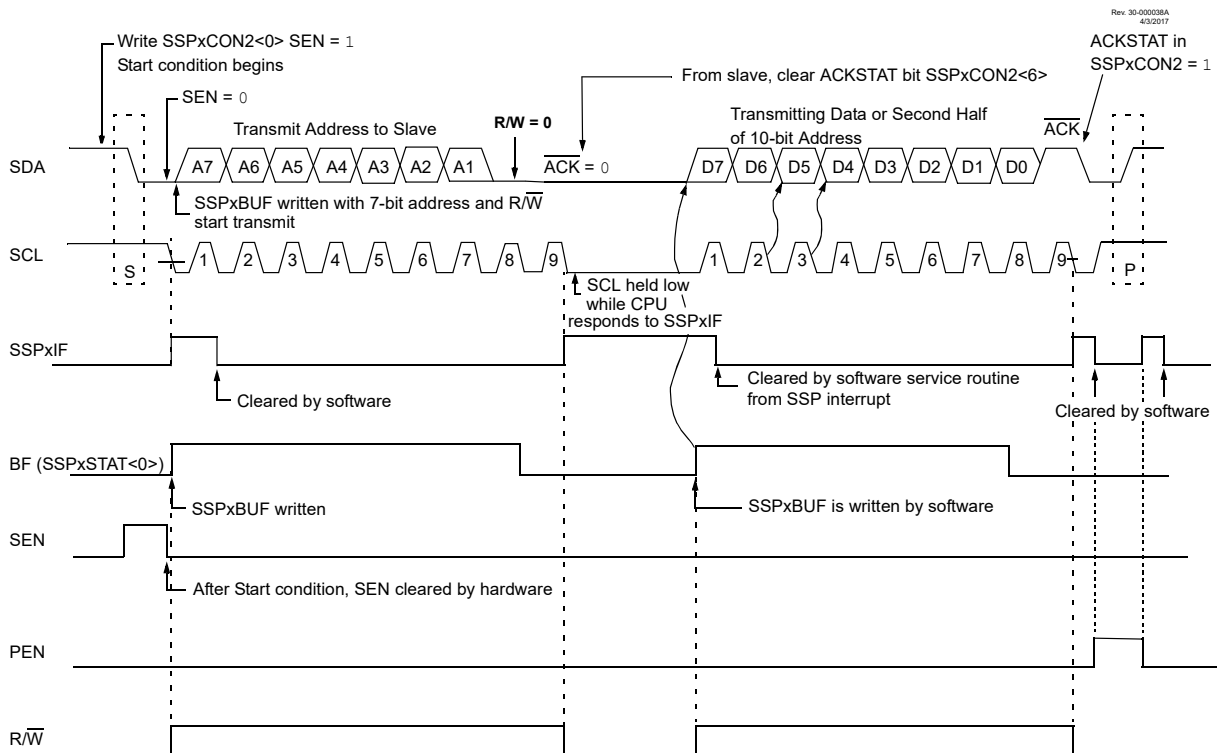
In Transmit mode, the **ACKSTAT** bit is cleared when the slave has sent an Acknowledge ($\overline{ACK} = 0$) and is set when the slave does not Acknowledge ($\overline{ACK} = 1$). A slave sends an Acknowledge when it has recognized its address (including a general call), or when the slave has properly received its data.

29.6.6.4 Typical Transmit Sequence:

1. The user generates a Start condition by setting the **SEN** bit.

2. SSPxIF is set by hardware on completion of the Start.
3. SSPxIF is cleared by software.
4. The MSSP module will wait the required start time before any other operation takes place.
5. The user loads the SSPxBUF with the slave address to transmit.
6. Address is shifted out the SDA pin until all eight bits are transmitted. Transmission begins as soon as SSPxBUF is written to.
7. The MSSP module shifts in the \overline{ACK} bit from the slave device and writes its value into the **ACKSTAT** bit.
8. The MSSP module generates an interrupt at the end of the ninth clock cycle by setting the SSPxIF bit.
9. The user loads the SSPxBUF with eight bits of data.
10. Data is shifted out the SDA pin until all eight bits are transmitted.
11. The MSSP module shifts in the \overline{ACK} bit from the slave device and writes its value into the **ACKSTAT** bit.
12. Steps 8-11 are repeated for all transmitted data bytes.
13. The user generates a Stop or Restart condition by setting the **PEN** or **RSEN** bits. Interrupt is generated once the Stop/Restart condition is complete.

Figure 29-28. I²C Master Mode Waveform (Transmission, 7 or 10-bit Address)



29.6.7 I²C Master Mode Reception

Master mode reception (Figure 29-29) is enabled by programming the **RCEN** Receive Enable bit.



Important: 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 all the following events occur:

- The receive enable flag is automatically cleared
- The contents of the SSPSR are loaded into the SSPxBUF

- The BF flag bit is set
- The SSPxIF flag bit is set
- The Baud Rate Generator is suspended from counting
- The SCL pin is held low

The MSSP is now in Idle state awaiting the next command. When the buffer is read by the CPU, the BF flag bit is automatically cleared. The user can then send an Acknowledge bit at the end of reception by setting the Acknowledge Sequence Enable, **ACKEN** bit.

29.6.7.1 BF Status Flag

In receive operation, the BF bit is set when an address or data byte is loaded into SSPxBUF from SSPSR. It is cleared when the SSPxBUF register is read.

29.6.7.2 SSPOV Status Flag

In receive operation, the **SSPOV** bit is set when eight bits are received into the SSPSR while the BF flag bit is already set from a previous reception.

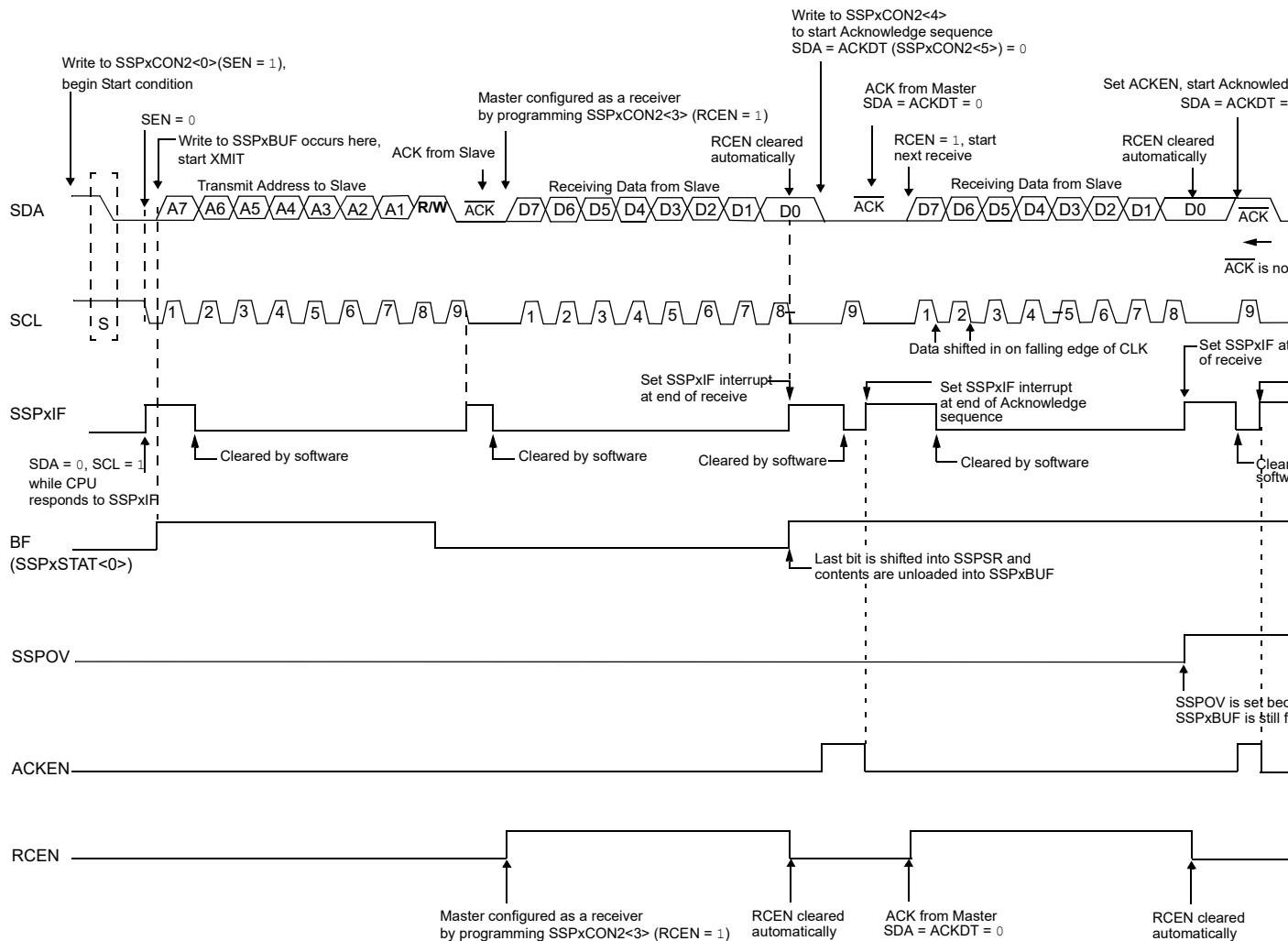
29.6.7.3 WCOL Status Flag

If the user writes the SSPxBUF when a receive is already in progress (i.e., SSPSR is still shifting in a data byte), the **WCOL** bit is set and the contents of the buffer are unchanged (the write does not occur).

29.6.7.4 Typical Receive Sequence:

1. The user generates a Start condition by setting the **SEN** bit.
2. SSPxIF is set by hardware on completion of the Start.
3. SSPxIF is cleared by software.
4. User writes SSPxBUF with the slave address to transmit and the **R/W** bit set.
5. Address is shifted out the SDA pin until all eight bits are transmitted. Transmission begins as soon as SSPxBUF is written to.
6. The MSSP module shifts in the \overline{ACK} bit from the slave device and writes its value into the **ACKSTAT** bit.
7. The MSSP module generates an interrupt at the end of the ninth clock cycle by setting the SSPxIF bit.
8. User sets the **RCEN** bit and the master clocks in a byte from the slave.
9. After the eighth falling edge of SCL, SSPxIF and BF are set.
10. Master clears SSPxIF and reads the received byte from SSPUF which clears BF.
11. Master sets the \overline{ACK} value to be sent to slave in the **ACKDT** bit and initiates the \overline{ACK} by setting the **ACKEN** bit.
12. Master's \overline{ACK} is clocked out to the slave and SSPxIF is set.
13. User clears SSPxIF.
14. Steps 8-13 are repeated for each received byte from the slave.
15. Master sends a not \overline{ACK} or Stop to end communication.

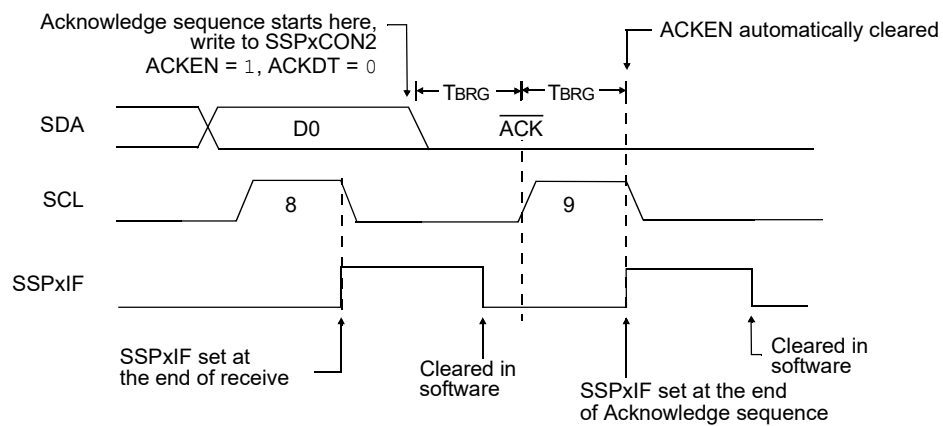
Figure 29-29. I²C Master Mode Waveform (Reception, 7-bit Address)



29.6.8 Acknowledge Sequence Timing

An Acknowledge sequence is enabled by setting the Acknowledge Sequence Enable **ACKEN** bit. When this bit is set, the SCL pin is pulled low and the contents of the Acknowledge data bit are presented on the SDA pin. If the user wishes to generate an Acknowledge, then the **ACKDT** bit should be cleared. If not, the user should set the **ACKDT** bit before starting an Acknowledge sequence. The Baud Rate Generator then counts for one rollover period (T_{BRG}) and the SCL pin is deasserted (pulled high). When the SCL pin is sampled high (clock arbitration), the Baud Rate Generator counts for T_{BRG} . 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 29-30. Acknowledge Sequence Waveform



Note: T_{BRG} = one Baud Rate Generator period.

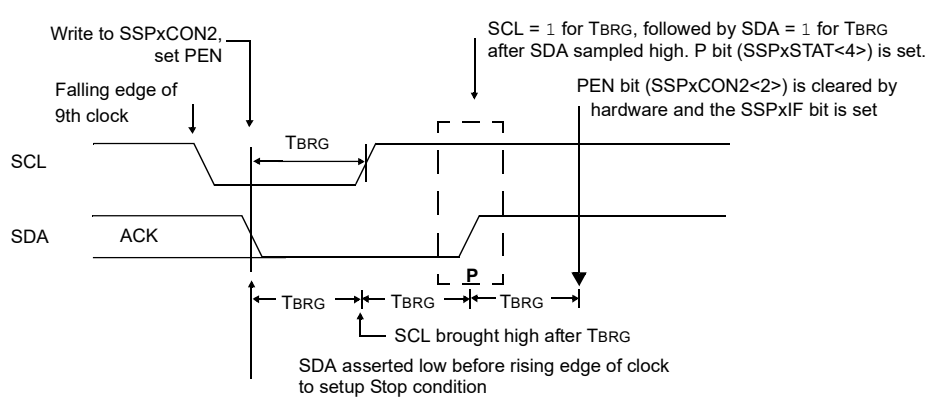
29.6.8.1 Acknowledge Write Collision

If the user writes the **SSPxBUF** when an Acknowledge sequence is in progress, then the **WCOL** bit is set and the contents of the buffer are unchanged (the write does not occur).

29.6.9 Stop Condition Timing

A Stop bit is asserted on the SDA pin at the end of a receive/transmit by setting the Stop Sequence Enable **PEN** bit. 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 T_{BRG} (Baud Rate Generator rollover count) later, the SDA pin will be deasserted. When the SDA pin is sampled high while SCL is high, the **P** bit is set. One T_{BRG} later, the **PEN** bit is cleared and the **SSPxIF** bit is set.

Figure 29-31. Stop Condition in Receive or Transmit Mode



Note: T_{BRG} = one Baud Rate Generator period.

29.6.9.1 Write Collision on Stop

If the user writes the SSPxBUF when a Stop sequence is in progress, then the WCOL bit is set and the contents of the buffer are unchanged (the write does not occur).

29.6.10 Sleep Operation

While in Sleep mode, the I²C slave module can receive addresses or data and when an address match or complete byte transfer occurs, wake the processor from Sleep (if the MSSP interrupt is enabled).

29.6.11 Effects of a Reset

A Reset disables the MSSP module and terminates the current transfer.

29.6.12 Multi-Master Mode

In Multi-Master mode, the interrupt generation on the detection of the Start and Stop conditions allows the determination of when the bus is free. The Stop (P) and Start (S) bits are cleared from a Reset or when the MSSP module is disabled. Control of the I²C bus may be taken when the P bit is set, or the bus is Idle, with both the S and P bits cleared. When the bus is busy, enabling the SSP interrupt will generate the interrupt when the Stop condition occurs.

In multi-master operation, the SDA line must be monitored for arbitration to see if the signal level is the expected output level. This check is performed by hardware with the result placed in the BCLxIF bit.

The states where arbitration can be lost are:

- Address Transfer
- Data Transfer
- A Start Condition
- A Repeated Start Condition
- An Acknowledge Condition

29.6.13 Multi-Master Communication, Bus Collision and Bus Arbitration

Multi-Master mode support is achieved by bus arbitration. When the master outputs address/data bits onto the SDA pin, arbitration takes place when the master outputs a '1' on SDA, by letting SDA float high and another master asserts a '0'. When the SCL pin floats high, data should be stable. If the expected data on SDA is a '1' and the data sampled on the SDA pin is '0', then a bus collision has taken place. The master will set the Bus Collision Interrupt Flag, BCLxIF and reset the I²C port to its Idle state (Figure 29-32).

If a transmit was in progress when the bus collision occurred, the transmission is halted, the BF flag is cleared, the SDA and SCL lines are deasserted and the SSPxBUF can be written to. When the user services the bus collision Interrupt Service Routine and if the I²C bus is free, the user can resume communication by asserting a Start condition.

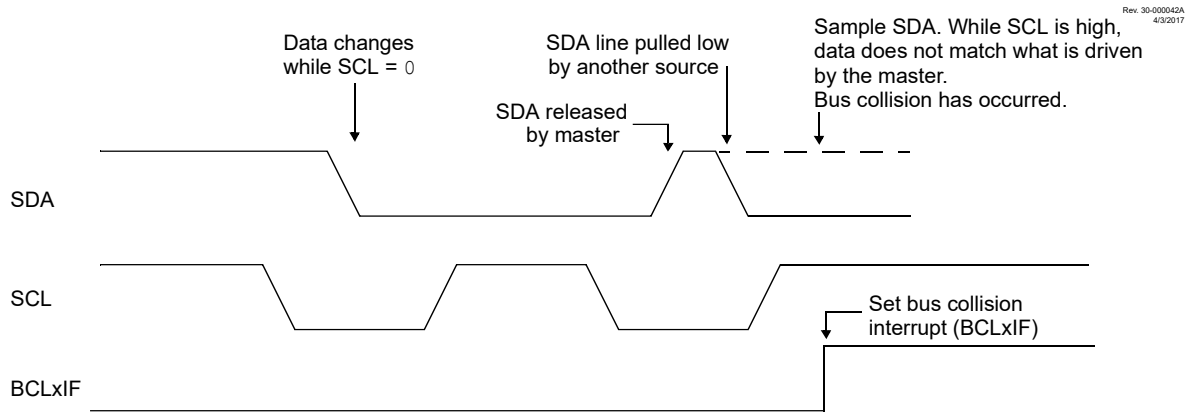
If a Start, Repeated Start, Stop or Acknowledge condition was in progress when the bus collision occurred, the condition is aborted, the SDA and SCL lines are deasserted and the respective control bits in the SSPxCON2 register are cleared. When the user services the bus collision Interrupt Service Routine and if the I²C bus is free, the user can resume communication by asserting a Start condition.

The master will continue to monitor the SDA and SCL pins. If a Stop condition occurs, the SSPxIF bit will be set.

A write to the SSPxBUF will start the transmission of data at the first data bit, regardless of where the transmitter left off when the bus collision occurred.

In Multi-Master mode, the interrupt generation on the detection of Start and Stop conditions allows the determination of when the bus is free. Control of the I²C bus can be taken when the P bit is set, or the bus is Idle and the S and P bits are cleared.

Figure 29-32. Bus Collision Timing for Transmit and Acknowledge



29.6.13.1 Bus Collision During a Start Condition

During a Start condition, a bus collision occurs if:

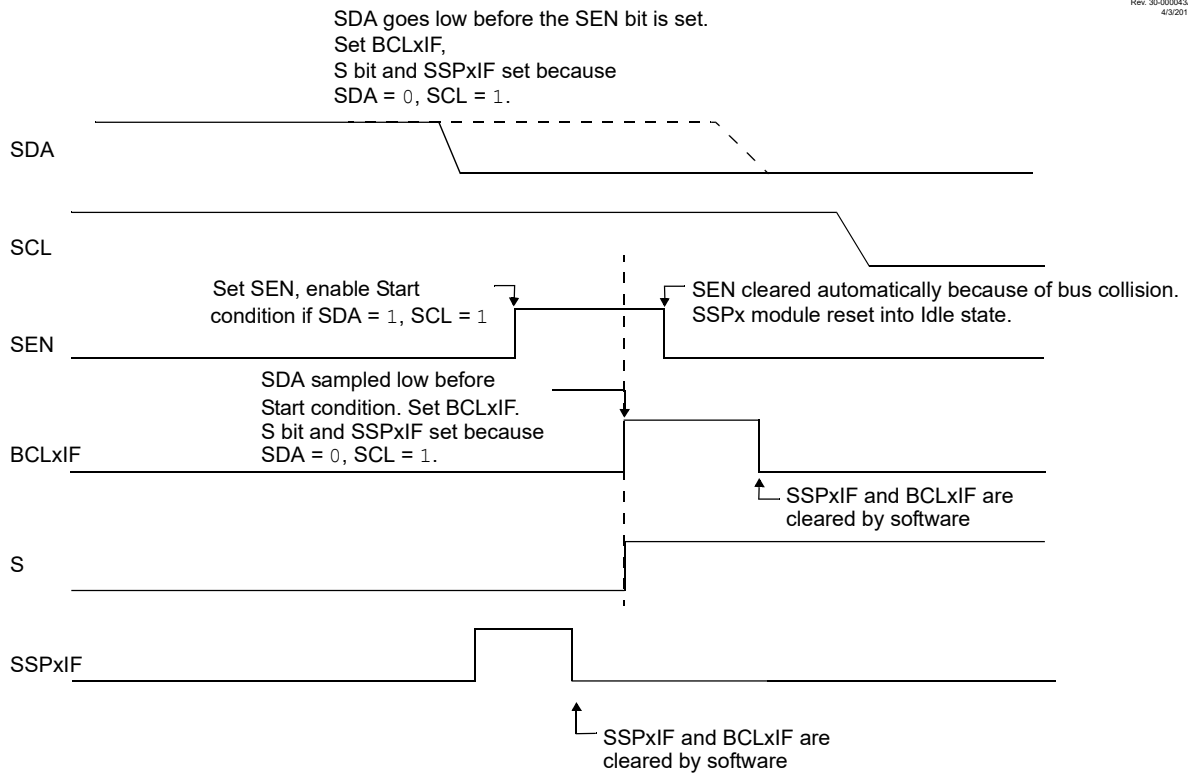
1. SDA or SCL are sampled low at the beginning of the Start condition (Figure 29-33).
2. SCL is sampled low before SDA is asserted low (Figure 29-34).

During a Start condition, both the SDA and the SCL pins are monitored.

If the SDA pin is already low, or the SCL pin is already low, then all of the following occur:

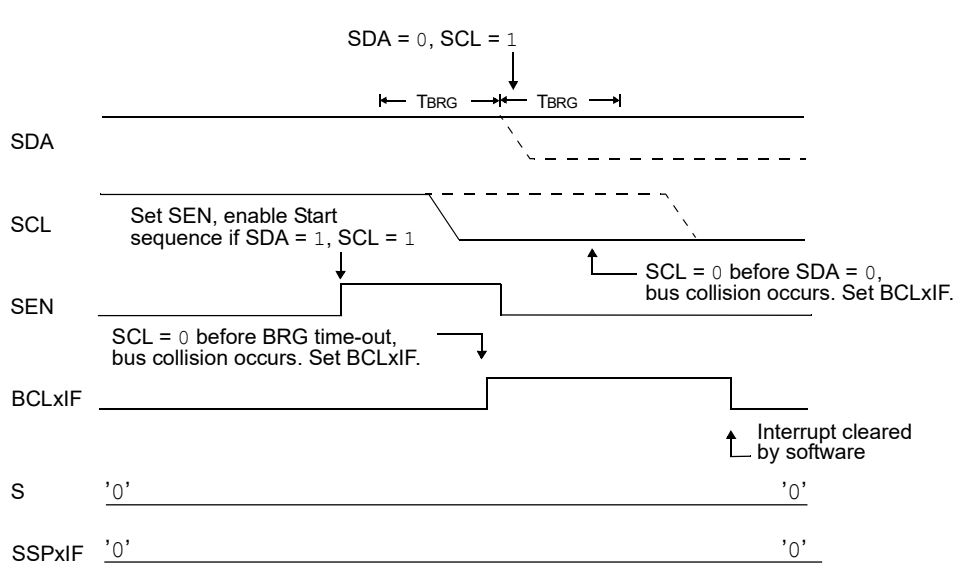
- the Start condition is aborted,
- the BCLxIF flag is set and
- the MSSP module is reset to its Idle state (Figure 29-33).

Figure 29-33. Bus Collision During Start Condition (SDA Only)



The Start condition begins with the SDA and SCL pins deasserted. When the SDA pin is sampled high, the Baud Rate Generator is loaded and counts down. If the SCL pin is sampled low while SDA is high, a bus collision occurs because it is assumed that another master is attempting to drive a data '1' during the Start condition.

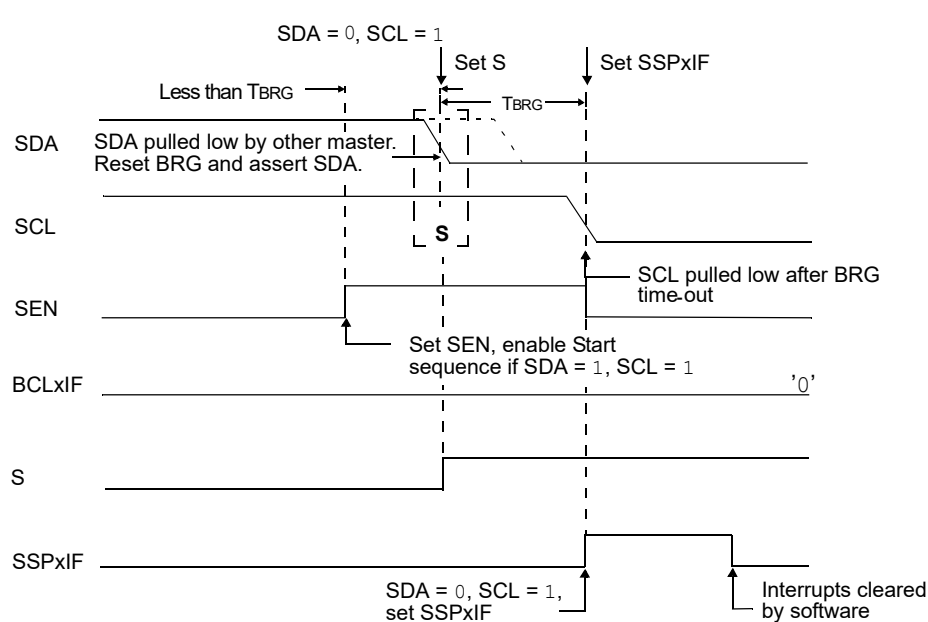
Figure 29-34. Bus Collision During Start Condition (SCL = 0)



If the SDA pin is sampled low during this count, the BRG is reset and the SDA line is asserted early (Figure 29-35). If, however, a '1' is sampled on the SDA pin, the SDA pin is asserted low at the end of the BRG count. The Baud Rate

Generator is then reloaded and counts down to zero; if the SCL pin is sampled as '0' during this time, a bus collision does not occur. At the end of the BRG count, the SCL pin is asserted low.

Figure 29-35. BRG Reset Due to SDA Arbitration During Start Condition



Important: 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.

29.6.13.2 Bus Collision During a Repeated Start Condition

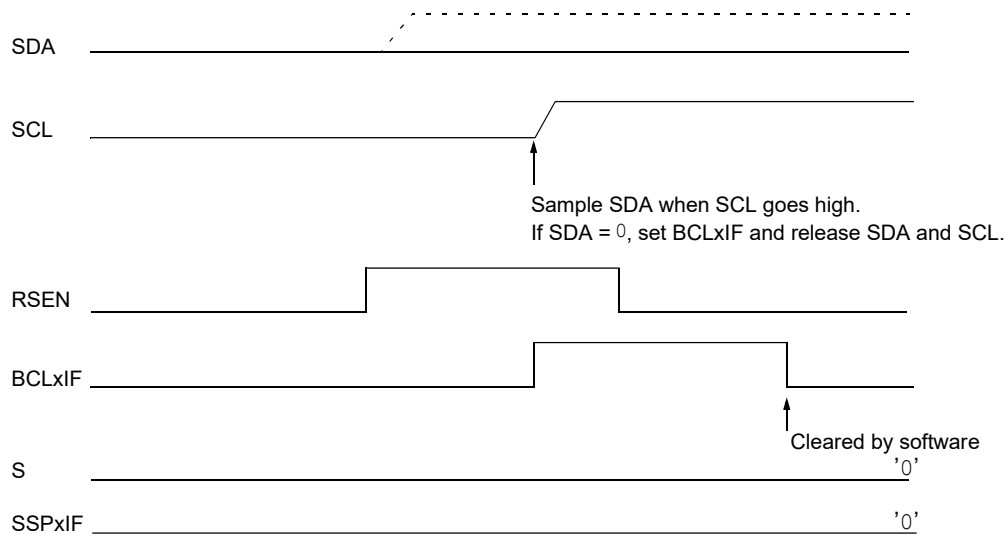
During a Repeated Start condition, a bus collision occurs if:

1. A low level is sampled on SDA when SCL goes from low level to high level (Case 1).
2. SCL goes low before SDA is asserted low, indicating that another master is attempting to transmit a data '1' (Case 2).

When the user releases SDA and the pin is allowed to float high, the BRG is loaded with SSPxADD and counts down to zero. The SCL pin is then deasserted and when sampled high, the SDA pin is sampled.

If SDA is low, a bus collision has occurred (i.e., another master is attempting to transmit a data '0', [Figure 29-36](#)). If SDA is sampled high, the BRG is reloaded and begins counting. If SDA goes from high-to-low before the BRG times out, no bus collision occurs because no two masters can assert SDA at exactly the same time.

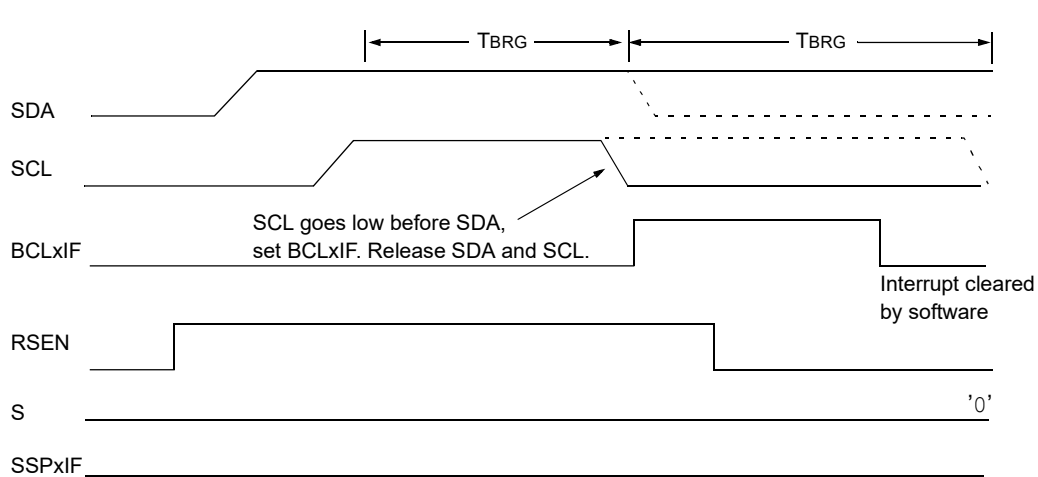
Figure 29-36. Bus Collision During a Repeated Start Condition (Case 1)



If SCL goes from high-to-low before the BRG times out and SDA has not already been asserted, a bus collision occurs. In this case, another master is attempting to transmit a data '1' during the Repeated Start condition, see Figure 29-37.

If, at the end of the BRG time-out, both SCL and SDA are still high, the SDA pin is driven low and the BRG is reloaded and begins counting. At the end of the count, regardless of the status of the SCL pin, the SCL pin is driven low and the Repeated Start condition is complete.

Figure 29-37. Bus Collision During Repeated Start Condition (Case 2)



29.6.13.3 Bus Collision During a Stop Condition

Bus collision occurs during a Stop condition if:

1. After the SDA pin has been deasserted and allowed to float high, SDA is sampled low after the BRG has timed out (Case 1).
2. After the SCL pin is deasserted, SCL is sampled low before SDA goes high (Case 2).

The Stop condition begins with SDA asserted low. When SDA is sampled low, the SCL pin is allowed to float. When the pin is sampled high (clock arbitration), the Baud Rate Generator is loaded with SSPxADD and counts down to zero. After the BRG times out, SDA is sampled. If SDA is sampled low, a bus collision has occurred. This is due to another master attempting to drive a data '0' (Figure 29-38). If the SCL pin is sampled low before SDA is allowed to float high, a bus collision occurs. This is another case of another master attempting to drive a data '0' (Figure 29-39).

Figure 29-38. Bus Collision During a Stop Condition (Case 1)

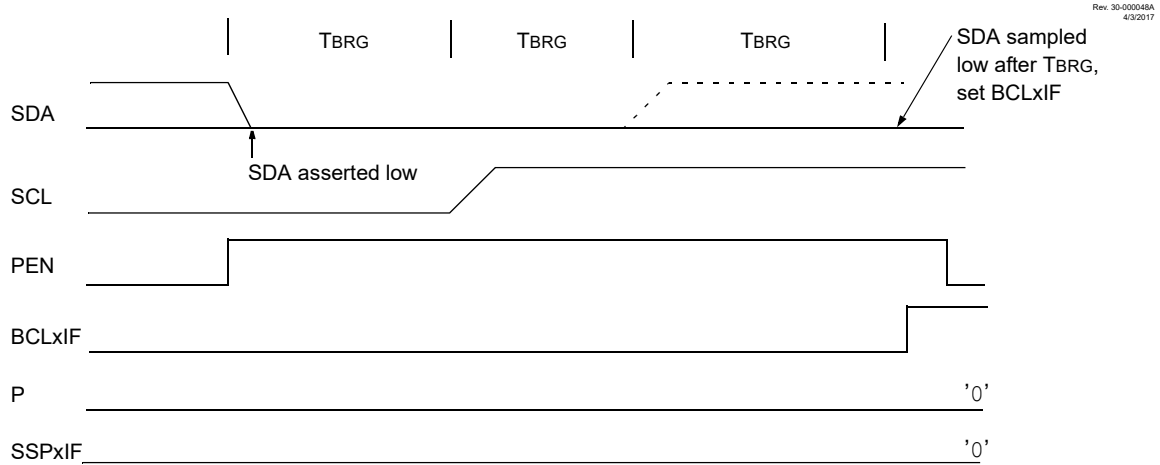
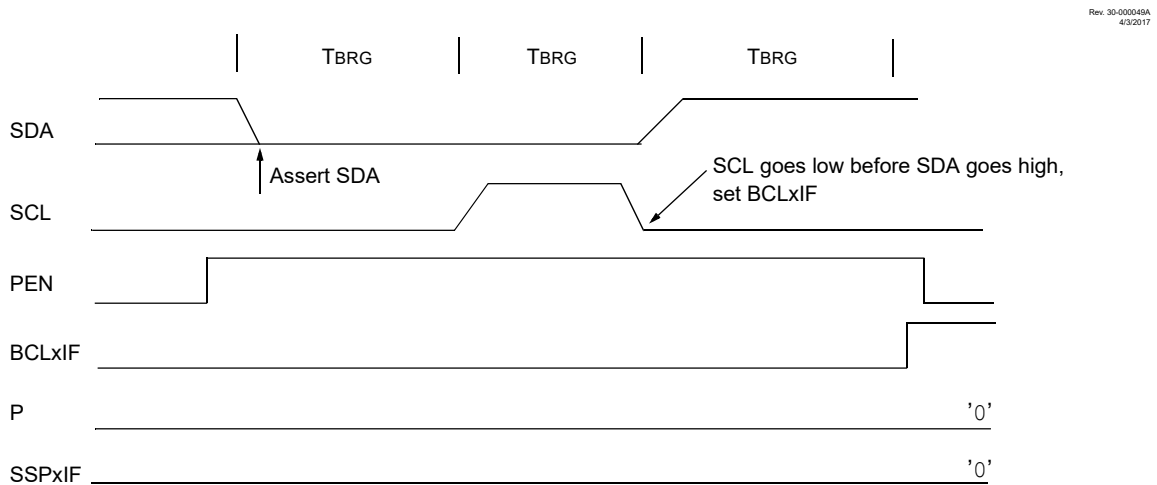


Figure 29-39. Bus Collision During a Stop Condition (Case 2)



29.7 Baud Rate Generator

The MSSP module has a Baud Rate Generator available for clock generation in both I²C and SPI Master modes. The Baud Rate Generator (BRG) reload value is placed in the SSPxADD register. When a write occurs to SSPxBUF, the Baud Rate Generator will automatically begin counting down.

Once the given operation is complete, the internal clock will automatically stop counting and the clock pin will remain in its last state.

An internal signal “Reload”, shown in Figure 29-40, triggers the value from SSPxADD to be loaded into the BRG counter. This occurs twice for each oscillation of the module clock line. The logic dictating when the reload signal is asserted depends on the mode in which the MSSP is being operated.

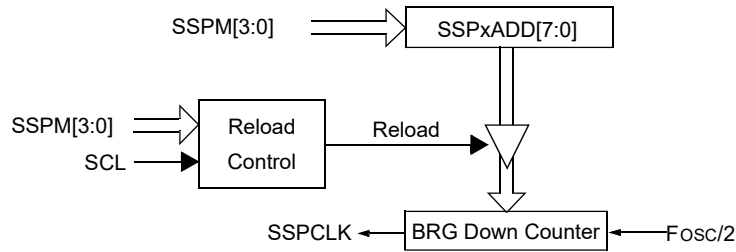
Table 29-1 illustrates clock rates based on instruction cycles and the BRG value loaded into SSPxADD.

Example 29-1. MSSP Baud Rate Generator Frequency Equation

$$F_{CLOCK} = \frac{F_{OSC}}{4 \times (SSPxADD + 1)}$$

Figure 29-40. Baud Rate Generator Block Diagram

Rev. 30-00050A
4/5/2017



Important: Values of 0x00, 0x01 and 0x02 are not valid for SSPxADD when used as a Baud Rate Generator for I²C. This is an implementation limitation.

Table 29-1. MSSP Clock Rate w/BRG

| F _{osc} | F _{CY} | BRG Value | F _{clock} (2 Rollovers of BRG) |
|------------------|-----------------|-----------|--|
| 32 MHz | 8 MHz | 13h | 400 kHz |
| 32 MHz | 8 MHz | 19h | 308 kHz |
| 32 MHz | 8 MHz | 4Fh | 100 kHz |
| 16 MHz | 4 MHz | 09h | 400 kHz |
| 16 MHz | 4 MHz | 0Ch | 308 kHz |
| 16 MHz | 4 MHz | 27h | 100 kHz |
| 4 MHz | 1 MHz | 09h | 100 kHz |

Note: Refer to the I/O port electrical specifications in the “*Electrical Specifications*” section, Internal Oscillator Parameters, to ensure the system is designed to support all requirements.

29.8 Register Summary: MSSP Control

| Address | Name | Bit Pos. | | | | | | | | | | |
|---------------|----------|----------|----------|---------|-------|-------|-----------|-------|------|------|--|------|
| 0x00 ... | Reserved | | | | | | | | | | | |
| 0x018B | | | | | | | | | | | | |
| 0x018C | SSP1BUF | 7:0 | BUF[7:0] | | | | | | | | | |
| 0x018D | SSP1ADD | 7:0 | ADD[7:0] | | | | | | | | | |
| 0x018E | SSP1MSK | 7:0 | MSK[6:0] | | | | | | | | | MSK0 |
| 0x018F | SSP1STAT | 7:0 | SMP | CKE | D/Ā | P | S | R/W | UA | BF | | |
| 0x0190 | SSP1CON1 | 7:0 | WCOL | SSPOV | SSPEN | CKP | SSPM[3:0] | | | | | |
| 0x0191 | SSP1CON2 | 7:0 | GCEN | ACKSTAT | ACKDT | ACKEN | RCEN | PEN | RSEN | SEN | | |
| 0x0192 | SSP1CON3 | 7:0 | ACKTIM | PCIE | SCIE | BOEN | SDAHT | SBCDE | AHEN | DHEN | | |
| 0x0193 ... | Reserved | | | | | | | | | | | |
| 0x0195 | | | | | | | | | | | | |
| 0x0196 | SSP2BUF | 7:0 | BUF[7:0] | | | | | | | | | |
| 0x0197 | SSP2ADD | 7:0 | ADD[7:0] | | | | | | | | | |
| 0x0198 | SSP2MSK | 7:0 | MSK[6:0] | | | | | | | | | MSK0 |
| 0x0199 | SSP2STAT | 7:0 | SMP | CKE | D/Ā | P | S | R/W | UA | BF | | |
| 0x019A | SSP2CON1 | 7:0 | WCOL | SSPOV | SSPEN | CKP | SSPM[3:0] | | | | | |
| 0x019B | SSP2CON2 | 7:0 | GCEN | ACKSTAT | ACKDT | ACKEN | RCEN | PEN | RSEN | SEN | | |
| 0x019C | SSP2CON3 | 7:0 | ACKTIM | PCIE | SCIE | BOEN | SDAHT | SBCDE | AHEN | DHEN | | |

29.9 Register Definitions: MSSP Control

29.9.1 SSPxBUF

Name: SSPxBUF
Address: 0x18C,0x196

MSSP Data Buffer Register

| | | | | | | | | |
|--------|----------|-----|-----|-----|-----|-----|-----|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | BUF[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | x | x | x | x | x | x | x | x |

Bits 7:0 – BUF[7:0] MSSP Input and Output Data Buffer bits

29.9.2 SSPxADD

Name: SSPxADD
Address: 0x18D,0x197

MSSP Baud Rate Divider and Address Register

| | | | | | | | | |
|--------|----------|-----|-----|-----|-----|-----|-----|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | ADD[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bits 7:0 – ADD[7:0]

- SPI and I²C Master: Baud rate divider
- I²C Slave: Address bits

| Value | Mode | Description |
|---------------------------|--|--|
| 11111111 -0000001 1 | SPI and I ² C Master | Baud rate divider. SCK/SCL pin clock period = ((n + 1) *4)/F _{OSC} . Values less than 3 are not valid. |
| xxxxx11x - xxxxx00x | I ² C 10-bit Slave MS Address | Bits 7-3 and Bit 0 are not used and are don't care. Bits 2:1 are bits 9:8 of the 10-bit Slave Most Significant Address |
| 11111111 -0000000 0 | I ² C 10-bit Slave LS Address | Bits 7:0 of 10-Bit Slave Least Significant Address |
| 1111111x -0000000 x | I ² C 7-bit Slave | Bit 0 is not used and is don't care. Bits 7:1 are the 7-bit Slave Address |

29.9.3 SSPxMSK

Name: SSPxMSK
Address: 0x18E,0x198

MSSP Address Mask Register

| | | | | | | | | |
|--------|----------|-----|-----|-----|-----|-----|-----|------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | MSK[6:0] | | | | | | | MSK0 |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Bits 7:1 – MSK[6:0] Mask bits

| Value | Mode | Description |
|-------|------------------------|--|
| 1 | I ² C Slave | The received address bit n is compared to SSPxADD bit n to detect I ² C address match |
| 0 | I ² C Slave | The received address bit n is not used to detect I ² C address match |

Bit 0 – MSK0

Mask bit for I²C 10-bit Slave mode

| Value | Mode | Description |
|-------|-------------------------------|--|
| 1 | I ² C 10-bit Slave | The received address bit 0 is compared to SSPxADD bit 0 to detect I ² C address match |
| 0 | I ² C 10-bit Slave | The received address bit 0 is not used to detect I ² C address match |
| x | SPI or I ² C 7-bit | Don't care |

29.9.4 SSPxSTAT

Name: SSPxSTAT
Address: 0x18F,0x199

MSSP Status Register

| | | | | | | | | |
|--------|-----|-----|--------------|----|----|--------------|----|----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | SMP | CKE | D/ \bar{A} | P | S | R/ \bar{W} | UA | BF |
| Access | R/W | R/W | RO | RO | RO | RO | RO | RO |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bit 7 – SMP Slew Rate Control bit

| Value | Mode | Description |
|-------|------------------|---|
| 1 | SPI Master | Input data is sampled at the end of data output time |
| 0 | SPI Master | Input data is sampled at the middle of data output time |
| 0 | SPI Slave | Keep this bit cleared in SPI Slave mode |
| 1 | I ² C | Slew rate control is disabled for Standard Speed mode (100 kHz and 1 MHz) |
| 0 | I ² C | Slew rate control is enabled for High-Speed mode (400 kHz) |

Bit 6 – CKE

SPI: Clock select bit⁽⁴⁾ I²C: SMBus Select bit

| Value | Mode | Description |
|-------|------------------|---|
| 1 | SPI | Transmit occurs on the transition from active to Idle clock state |
| 0 | SPI | Transmit occurs on the transition from Idle to active clock state |
| 1 | I ² C | Enables SMBus-specific inputs |
| 0 | I ² C | Disables SMBus-specific inputs |

Bit 5 – D/ \bar{A}

Data/Address bit

| Value | Mode | Description |
|-------|--------------------------------|--|
| x | SPI or I ² C Master | Reserved |
| 1 | I ² C Slave | Indicates that the last byte received or transmitted was data |
| 0 | I ² C Slave | Indicates that the last byte received or transmitted was address |

Bit 4 – P

Stop bit⁽¹⁾

| Value | Mode | Description |
|-------|------------------|--------------------------------|
| x | SPI | Reserved |
| 1 | I ² C | Stop bit was detected last |
| 0 | I ² C | Stop bit was not detected last |

Bit 3 – S

Start bit⁽¹⁾

| Value | Mode | Description |
|-------|------------------|---------------------------------|
| x | SPI | Reserved |
| 1 | I ² C | Start bit was detected last |
| 0 | I ² C | Start bit was not detected last |

Bit 2 – R/ \bar{W}

Read/ \bar{W} rite Information bit^(2,3)

| Value | Mode | Description |
|-------|------------------------|-------------|
| x | SPI | Reserved |
| 1 | I ² C Slave | Read |
| 0 | I ² C Slave | Write |

| Value | Mode | Description |
|-------|-------------------------|-----------------------------|
| 1 | I ² C Master | Transmit is in progress |
| 0 | I ² C Master | Transmit is not in progress |

Bit 1 – UA Update Address bit (10-Bit Slave mode only)

| Value | Mode | Description |
|-------|-------------------------------|---|
| x | All other modes | Reserved |
| 1 | I ² C 10-bit Slave | Indicates that the user needs to update the address in the SSPxADD register |
| 0 | I ² C 10-bit Slave | Address does not need to be updated |

Bit 0 – BF

Buffer Full Status bit⁽⁵⁾

| Value | Mode | Description |
|-------|----------------------------------|---|
| 1 | I ² C Transmit | Character written to SSPxBUF has not been sent |
| 0 | I ² C Transmit | SSPxBUF is ready for next character |
| 1 | SPI and I ² C Receive | Received character in SSPxBUF has not been read |
| 0 | SPI and I ² C Receive | Received character in SSPxBUF has been read |

Note:

1. This bit is cleared on Reset and when **SSPEN** is cleared.
2. In I²C Slave mode 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 \overline{ACK} bit.
3. ORing this bit with SEN, RSEN, PEN, RCEN or ACKEN will indicate if the MSSP is in Active mode.
4. Polarity of clock state is set by the **CKP** bit.
5. I²C receive status does not include \overline{ACK} and Stop bits.

29.9.5 SSPxCON1

Name: SSPxCON1
Address: 0x190,0x19A

MSSP Control Register 1

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|--------|--------|-------|-----|-----------|-----|-----|-----|
| | WCOL | SSPOV | SSPEN | CKP | SSPM[3:0] | | | |
| Access | R/W/HS | R/W/HS | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bit 7 – WCOL

Write Collision Detect bit

| Value | Mode | Description |
|-------|--|---|
| 1 | SPI | A write to the SSPxBUF register was attempted while the previous byte was still transmitting (must be cleared by software) |
| 1 | I ² C Master transmit | A write to the SSPxBUF register was attempted while the I ² C conditions were not valid for a transmission to be started (must be cleared by software) |
| 1 | I ² C Slave transmit | The SSPxBUF register is written while it is still transmitting the previous word (must be cleared in software) |
| 0 | SPI or I ² C Master or Slave transmit | No collision |
| x | Master or Slave receive | Don't care |

Bit 6 – SSPOV

Receive Overflow Indicator bit⁽¹⁾

| Value | Mode | Description |
|-------|--|--|
| 1 | SPI Slave | A byte is received while the SSPxBUF register is still holding the previous byte. The user must read SSPxBUF, even if only transmitting data, to avoid setting overflow. (must be cleared in software) |
| 1 | I ² C Receive | A byte is received while the SSPxBUF register is still holding the previous byte (must be cleared in software) |
| 0 | SPI Slave or I ² C Receive | No overflow |
| x | SPI Master or I ² C Master transmit | Don't care |

Bit 5 – SSPEN

Master Synchronous Serial Port Enable bit.⁽²⁾

| Value | Mode | Description |
|-------|------------------|---|
| 1 | SPI | Enables the serial port. The SCKx, SDOx, SDIx, and SSx pin selections must be made with the PPS controls. Each signal must be configured with the corresponding TRIS control to the direction appropriate for the mode selected. |
| 1 | I ² C | Enables the serial port. The SDAx and SCLx pin selections must be made with the PPS controls. Since both signals are bidirectional the PPS input pin and PPS output pin selections must be made that specify the same pin. Both pins must be configured as inputs with the corresponding TRIS controls. |
| 0 | All | Disables serial port and configures these pins as I/O PORT pins |

Bit 4 – CKP

SCK Release Control bit

| Value | Mode | Description |
|-------|-------------------------|---|
| 1 | SPI | Idle state for the clock is a high level |
| 0 | SPI | Idle state for the clock is a low level |
| 1 | I ² C Slave | Releases clock |
| 0 | I ² C Slave | Holds clock low (clock stretch), used to ensure data setup time |
| x | I ² C Master | Unused in this mode |

Bits 3:0 – SSPM[3:0]Master Synchronous Serial Port Mode Select bits⁽⁴⁾

| Value | Description |
|-------|---|
| 1111 | I ² C Slave mode: 10-bit address with Start and Stop bit interrupts enabled |
| 1110 | I ² C Slave mode: 7-bit address with Start and Stop bit interrupts enabled |
| 1101 | Reserved - do not use |
| 1100 | Reserved - do not use |
| 1011 | I ² C Firmware Controlled Master mode (slave Idle) |
| 1010 | SPI Master mode: Clock = $F_{OSC}/(4*(SSPxADD+1))$. SSPxADD must be greater than 0. ⁽³⁾ |
| 1001 | Reserved - do not use |
| 1000 | I ² C Master mode: Clock = $F_{OSC}/(4 * (SSPxADD + 1))$ |
| 0111 | I ² C Slave mode: 10-bit address |
| 0110 | I ² C Slave mode: 7-bit address |
| 0101 | SPI Slave mode: Clock = SCKx pin. \overline{SSx} pin control is disabled |
| 0100 | SPI Slave mode: Clock = SCKx pin. \overline{SSx} pin control is enabled |
| 0011 | SPI Master mode: Clock = TMR2 output/2 |
| 0010 | SPI Master mode: Clock = $F_{osc}/64$ |
| 0001 | SPI Master mode: Clock = $F_{osc}/16$ |
| 0000 | SPI Master mode: Clock = $F_{osc}/4$ |

Note:

1. In Master mode, the overflow bit is not set since each new reception (and transmission) is initiated by writing to the SSPxBUF register.
2. When enabled, these pins must be properly configured as inputs or outputs.
3. SSPxADD = 0 is not supported.
4. Bit combinations not specifically listed here are either reserved or implemented in I²C mode only.

29.9.6 SSPxCON2

Name: SSPxCON2
Address: 0x191,0x19B

Control Register for I²C Operation Only
 MSSP Control Register 2

| | | | | | | | | |
|--------|------|---------|-------|-------|------|-----|------|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | GCEN | ACKSTAT | ACKDT | ACKEN | RCEN | PEN | RSEN | SEN |
| Access | R/W | R/W/HC | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bit 7 – GCEN

General Call Enable bit (Slave mode only)

| Value | Mode | Description |
|-------|-------------|-----------------------------|
| x | Master mode | Don't care |
| 1 | Slave mode | General call is enabled |
| 0 | Slave mode | General call is not enabled |

Bit 6 – ACKSTAT Acknowledge Status bit (Master Transmit mode only)

| Value | Description |
|-------|---|
| 1 | Acknowledge was not received from slave |
| 0 | Acknowledge was received from slave |

Bit 5 – ACKDT

Acknowledge Data bit (Master Receive mode only)⁽¹⁾

| Value | Description |
|-------|-----------------|
| 1 | Not Acknowledge |
| 0 | Acknowledge |

Bit 4 – ACKEN

Acknowledge Sequence Enable bit⁽²⁾

| Value | Description |
|-------|--|
| 1 | Initiates Acknowledge sequence on SDAx and SCLx pins and transmits ACKDT data bit; automatically cleared by hardware |
| 0 | Acknowledge sequence is Idle |

Bit 3 – RCEN

Receive Enable bit (Master Receive mode only)⁽²⁾

| Value | Description |
|-------|---|
| 1 | Enables Receive mode for I ² C |
| 0 | Receive is Idle |

Bit 2 – PEN

Stop Condition Enable bit (Master mode only)⁽²⁾

| Value | Description |
|-------|---|
| 1 | Initiates Stop condition on SDAx and SCLx pins; automatically cleared by hardware |
| 0 | Stop condition is Idle |

Bit 1 – RSEN

Repeated Start Condition Enable bit (Master mode only)⁽²⁾

| Value | Description |
|-------|---|
| 1 | Initiates Repeated Start condition on SDAx and SCLx pins; automatically cleared by hardware |
| 0 | Repeated Start condition is Idle |

PIC16(L)F18426/46

MSSP - Master Synchronous Serial Port Module

Bit 0 – SEN

Start Condition Enable bit (Master mode only)⁽²⁾

| Value | Description |
|-------|--|
| 1 | Initiates Start condition on SDAx and SCLx pins; automatically cleared by hardware |
| 0 | Start condition is Idle |

Note:

1. The value that will be transmitted when the user initiates an Acknowledge sequence at the end of a receive.
2. If the I²C module is active, these bits may not be set (no spooling) and the SSPxBUF may not be written (or writes to the SSPxBUF are disabled).

29.9.7 SSPxCON3

Name: SSPxCON3
Address: 0x192,0x19C

MSSP Control Register 3

| | | | | | | | | |
|--------|---------|------|------|------|-------|-------|------|------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | ACKTIM | PCIE | SCIE | BOEN | SDAHT | SBCDE | AHEN | DHEN |
| Access | R/HS/HC | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bit 7 – ACKTIM Acknowledge Time Status bit
Unused in Master mode.

| Value | Mode | Description |
|-------|---|---|
| x | SPI or I ² C Master | This bit is not used |
| 1 | I ² C Slave and AHEN = 1 or DHEN = 1 | Eighth falling edge of SCL has occurred and the $\overline{\text{ACK}}$ /NACK state is active |
| 0 | I ² C Slave | $\overline{\text{ACK}}$ /NACK state is not active. Transitions low on ninth rising edge of SCL. |

Bit 6 – PCIE
Stop Condition Interrupt Enable bit⁽¹⁾

| Value | Mode | Description |
|-------|-----------------------------|---|
| x | SPI or SSPM = 1111 or 0111 | Don't care |
| 1 | SSPM ≠ 1111 and SSPM ≠ 0111 | Enable interrupt on detection of Stop condition |
| 0 | SSPM ≠ 1111 and SSPM ≠ 0111 | Stop detection interrupts are disabled |

Bit 5 – SCIE Start Condition Interrupt Enable bit

| Value | Mode | Description |
|-------|-----------------------------|--|
| x | SPI or SSPM = 1111 or 0111 | Don't care |
| 1 | SSPM ≠ 1111 and SSPM ≠ 0111 | Enable interrupt on detection of Start condition |
| 0 | SSPM ≠ 1111 and SSPM ≠ 0111 | Start detection interrupts are disabled |

Bit 4 – BOEN
Buffer Overwrite Enable bit⁽²⁾

| Value | Mode | Description |
|-------|------------------|--|
| 1 | SPI | SSPxBUF is updated every time a new data byte is available, ignoring the BF bit |
| 0 | SPI | If a new byte is receive with BF set then SSPOV is set and SSPxBUF is not updated |
| 1 | I ² C | SSPxBUF is updated every time a new data byte is available, ignoring the SSPOV effect on updating the buffer |
| 0 | I ² C | SSPxBUF is only updated when SSPOV is clear |

Bit 3 – SDAHT SDA Hold Time Selection bit

| Value | Mode | Description |
|-------|------------------|--|
| x | SPI | Not used in SPI mode |
| 1 | I ² C | Minimum of 300 ns hold time on SDA after the falling edge of SCL |
| 0 | I ² C | Minimum of 100 ns hold time on SDA after the falling edge of SCL |

Bit 2 – SBCDE Slave Mode Bus Collision Detect Enable bit
Unused in Master mode.

| Value | Mode | Description |
|-------|--------------------------------|------------------------------------|
| x | SPI or I ² C Master | Don't care |
| 1 | I ² C Slave | Collision detection is enabled |
| 0 | I ² C Slave | Collision detection is not enabled |

Bit 1 – AHEN Address Hold Enable bit

| Value | Mode | Description |
|-------|--------------------------------|--|
| x | SPI or I ² C Master | Don't care |
| 1 | I ² C Slave | Address hold is enabled. As a result CKP is cleared after the eighth falling SCL edge of an address byte reception. Software must set the CKP bit to resume operation. |
| 0 | I ² C Slave | Address hold is not enabled |

Bit 0 – DHEN Data Hold Enable bit

| Value | Mode | Description |
|-------|--------------------------------|---|
| x | SPI or I ² C Master | Don't care |
| 1 | I ² C Slave | Data hold is enabled. As a result CKP is cleared after the eighth falling SCL edge of a data byte reception. Software must set the CKP bit to resume operation. |
| 0 | I ² C Slave | Data hold is not enabled |

Note:

1. This bit has no effect in Slave modes that Start and Stop condition detection is explicitly listed as enabled.
2. For daisy-chained SPI operation; allows the user to ignore all but the last received byte. SSPOV is still set when a new byte is received and BF = 1, but hardware continues to write the most recent byte to SSPxBUF.

30. FVR - Fixed Voltage Reference

The Fixed Voltage Reference (FVR) is a stable voltage reference, independent of V_{DD} , with the following selectable output levels:

- 1.024V
- 2.048V
- 4.096V

The output of the FVR can be configured to supply a reference voltage to the following:

- ADC input channel
- ADC positive reference
- Comparator input
- Digital-to-Analog Converter (DAC)

The FVR can be enabled by setting the FVREN bit of the FVRCON register.



Important: Fixed Voltage Reference output cannot exceed V_{DD} .

Related Links

[30.4.1 FVRCON](#)

30.1 Independent Gain Amplifiers

The output of the FVR, which is connected to the ADC, Comparators, and DAC, is routed through two independent programmable gain amplifiers. Each amplifier can be programmed for a gain of 1x, 2x or 4x, to produce the three possible voltage levels.

The ADFVR[1:0] bits of the FVRCON register are used to enable and configure the gain amplifier settings for the reference supplied to the ADC module.

The CDAFVR[1:0] bits of the FVRCON register are used to enable and configure the gain amplifier settings for the reference supplied to the DAC and comparator module.

Related Links

[32. ADC2 - Analog-to-Digital Converter](#)

[34. CMP - Comparator Module](#)

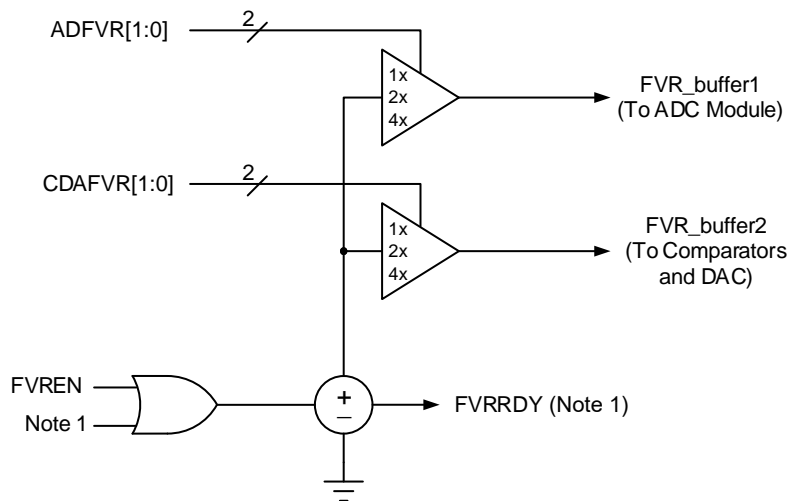
[33. DAC - 5-Bit Digital-to-Analog Converter](#)

30.2 FVR Stabilization Period

When the Fixed Voltage Reference module is enabled, it requires time for the reference and amplifier circuits to stabilize. Once the circuits stabilize and are ready for use, the FVRRDY bit of the FVRCON register will be set.

Figure 30-1. Voltage Reference Block Diagram

Rev. 10-000-053C
12/20/2013



Note:

1. Any peripheral requiring the fixed reference.

30.3 Register Summary - FVR

| Address | Name | Bit Pos. | | | | | | | |
|-------------|---------------|----------|-------|--------|------|-------|-------------|------------|--|
| 0x00 ... | Reserved | | | | | | | | |
| 0x090B | | | | | | | | | |
| 0x090C | FVRCON | 7:0 | FVREN | FVRRDY | TSEN | TSRNG | CDAFVR[1:0] | ADFVR[1:0] | |

30.4 Register Definitions: FVR Control

30.4.1 FVRCON

Name: FVRCON
Address: 0x90C

Fixed Voltage Reference Control Register

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|-------|--------|------|-------|-------------|-----|------------|-----|
| | FVREN | FVRRDY | TSEN | TSRNG | CDAFVR[1:0] | | ADFVR[1:0] | |
| Access | R/W | R | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bit 7 – FVREN Fixed Voltage Reference Enable bit

| Value | Description |
|-------|-------------------------------------|
| 1 | Fixed Voltage Reference is enabled |
| 0 | Fixed Voltage Reference is disabled |

Bit 6 – FVRRDY Fixed Voltage Reference Ready Flag bit (3)

| Value | Description |
|-------|--|
| 1 | Fixed Voltage Reference output is ready for use |
| 0 | Fixed Voltage Reference output is not ready or not enabled |

Bit 5 – TSEN Temperature Indicator Enable bit(2)

| Value | Description |
|-------|-----------------------------------|
| 1 | Temperature Indicator is enabled |
| 0 | Temperature Indicator is disabled |

Bit 4 – TSRNG Temperature Indicator Range Selection bit(2)

| Value | Description |
|-------|---|
| 1 | $V_{OUT} = V_{DD} - 4 V_T$ (High Range) |
| 0 | $V_{OUT} = V_{DD} - 2 V_T$ (Low Range) |

Bits 3:2 – CDAFVR[1:0] Comparator FVR Buffer Gain Selection bits

| Value | Description |
|-------|---|
| 11 | Comparator FVR Buffer Gain is 4x, (4.096V) ⁽¹⁾ |
| 10 | Comparator FVR Buffer Gain is 2x, (2.048V) ⁽¹⁾ |
| 01 | Comparator FVR Buffer Gain is 1x, (1.024V) |
| 00 | Comparator FVR Buffer is off |

Bits 1:0 – ADFVR[1:0] ADC FVR Buffer Gain Selection bit

| Value | Description |
|-------|--|
| 11 | ADC FVR Buffer Gain is 4x, (4.096V) ⁽¹⁾ |
| 10 | ADC FVR Buffer Gain is 2x, (2.048V) ⁽¹⁾ |
| 01 | ADC FVR Buffer Gain is 1x, (1.024V) |
| 00 | ADC FVR Buffer is off |

Note:

- Fixed Voltage Reference output cannot exceed V_{DD} .
- See “Temperature Indicator Module” section for additional information.
- FVRRDY is always ‘1’.

Related Links

[31. Temperature Indicator Module](#)

31. Temperature Indicator Module

This family of devices is equipped with a temperature circuit designed to measure the operating temperature of the silicon die.

The circuit's range of operating temperature falls between -40°C and $+125^{\circ}\text{C}$. The output is a voltage that is proportional to the device temperature. The output of the temperature indicator is internally connected to the device ADC.

The circuit may be used as a temperature threshold detector or a more accurate temperature indicator, depending on the level of calibration performed. A one point calibration allows the circuit to indicate a temperature closely surrounding that point. A two-point calibration allows the circuit to sense the entire range of temperature more accurately.

Reference Application Note AN2092, "Using the Temperature Indicator Module" (DS00002092) for more details regarding the calibration process.

31.1 Module Operation

The figure below shows a simplified block diagram of the temperature circuit. The proportional voltage output is achieved by measuring the forward voltage drop across multiple silicon junctions.

The following equation describes the output characteristics of the temperature indicator.

Equation 31-1. V_{OUT} Ranges

High Range: $V_{OUT} = V_{DD} - 4V_T$

Low Range: $V_{OUT} = V_{DD} - 2V_T$

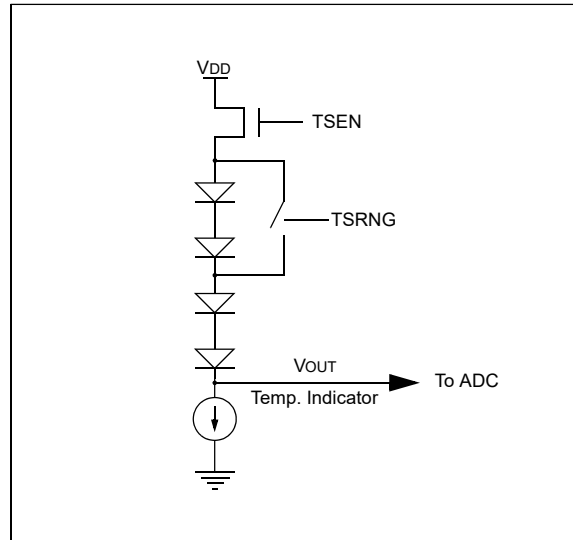
The temperature sense circuit is integrated with the Fixed Voltage Reference (FVR) module. See "Fixed Voltage Reference (FVR)" section for more information.

The circuit is enabled by setting the TSEN bit of the FVRCON register. When disabled, the circuit draws no current.

The circuit operates in either high or low range. The high range, selected by setting the TSRNG bit of the FVRCON register, provides a wider output voltage. This provides more resolution over the temperature range. This range requires a higher bias voltage to operate and thus, a higher V_{DD} is needed.

The low range is selected by clearing the TSRNG bit of the FVRCON register. The low range generates a lower voltage drop and thus, a lower V_{DD} voltage is needed to operate the circuit. The low range is provided for low voltage operation.

Figure 31-1. Temperature Circuit Block Diagram



Related Links

- [30. FVR - Fixed Voltage Reference](#)
- [31.3 Temperature Indicator Range](#)
- [32. ADC2 - Analog-to-Digital Converter](#)

31.2 Minimum Operating V_{DD}

When the temperature circuit is operated in low range, the device may be operated at any operating voltage that is within specifications. When the temperature circuit is operated in high range, the device operating voltage, V_{DD} , must be high enough to ensure that the temperature circuit is correctly biased.

The following table shows the recommended minimum V_{DD} vs. range setting.

Table 31-1. Recommended V_{DD} vs. Range

| Min. V_{DD} , TSRNG = 1 (High Range) | Min. V_{DD} , TSRNG = 0 (Low Range) |
|--|---------------------------------------|
| ≥3.6V | ≥1.8V |

31.3 Temperature Indicator Range

The temperature indicator circuit operates in either High or Low range. The High range, selected by setting the TSRNG bit of the FVRCON register, provides a wider output voltage. This provides more resolution over the temperature range. High range requires a higher-bias voltage to operate and thus, a higher V_{DD} is needed. The Low range is selected by clearing the TSRNG bit of the FVRCON register. The Low range generates a lower sensor voltage and thus, a lower V_{DD} voltage is needed to operate the circuit.

The output voltage of the sensor is the highest value at -40°C and the lowest value at $+125^{\circ}\text{C}$.

- **High Range:** The High range is used in applications with the reference for the ADC, $V_{REF} = 2.048\text{V}$. This range may not be suitable for battery-powered applications. The ADC reading (in counts) at 90°C for the high range setting is stored in the [DIA Table](#) as parameter TSHR2.
- **Low Range:** This mode is useful in applications in which the V_{DD} is too low for high-range operation. The V_{DD} in this mode can be as low as 1.8V. V_{DD} must, however, be at least 0.5V higher than the maximum sensor voltage depending on the expected low operating temperature. The ADC reading (in counts) at 90°C for the low range setting is stored in the [DIA Table](#) as parameter TSLR2.

31.4 Estimation of Temperature

This section describes the steps involved in estimating the die temperature, T_{MEAS} :

1. Obtain the ADC count value of the measured analog voltage: The analog output voltage, V_{MEAS} is converted to a digital count value by the Analog-to-Digital Converter (ADC) and is referred to as ADC_{MEAS} .
2. Obtain the ADC count value, ADC_{DIA} at 90°C, from the [DIA Table](#). This parameter is TSLR2 for the low range setting or TSHR2 for the high range setting of the temperature indicator module.
3. Obtain the output analog voltage (in mV) value of the Fixed Reference Voltage (FVR) for 2x setting, from the DIA table. This parameter is referred to as FVRA2X in the [DIA Table](#).
4. Obtain the value of the temperature indicator voltage sensitivity, parameter M_v , from the “*Electrical Specifications*” section.

The following equation provides an estimate of the die temperature based on the above parameters:

Equation 31-2. Sensor Temperature

$$T_{MEAS} = 90 + \frac{(ADC_{MEAS} - ADC_{DIA}) \times FVRA2X}{(2^N - 1) \times M_v}$$

Note: Where:

ADC_{MEAS} = ADC reading at temperature being estimated

ADC_{DIA} = ADC reading stored in the DIA

FVRA2X = FVR value stored in the DIA for 2x setting

N = Resolution of the ADC

M_v = Temperature Indicator voltage sensitivity (mV/°C)

Note: It is recommended to take the average of ten measurements of ADC_{MEAS} to reduce noise and improve accuracy.

Related Links

[40.4.6 Temperature Indicator Requirements](#)

31.4.1 Calibration

31.4.1.1 Higher-Order Calibration

If the application requires more precise temperature measurement, additional calibrations steps will be necessary. For these applications, two-point or three-point calibration is recommended. An Application Note will be released in future that demonstrates higher-order calibration process. An Application Note that demonstrates higher-order calibration process will be released in the future.

31.4.2 Temperature Resolution

The resolution of the ADC reading, Ma (°C/count), depends on both the ADC resolution N and the reference voltage used for conversion, as shown in the equation below. It is recommended to use the smallest V_{REF} value, such as the ADC FVR1 Output Voltage for 2x setting (FVRA2X) value from the [DIA Table](#).

Related Links

[40.4.11 Fixed Voltage Reference \(FVR\) Specifications](#)

31.5 ADC Acquisition Time

To ensure accurate temperature measurements, the user must wait at least 200 ms after the ADC input multiplexer is connected to the temperature indicator output before the conversion is performed.

32. ADC² - Analog-to-Digital Converter

The Analog-to-Digital Converter with Computation (ADC²) allows conversion of an analog input signal to a 12-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 12-bit binary result via successive approximation and stores the conversion result into the ADC result registers (ADRESH:ADRESL register pair).

Additionally, the following features are provided within the ADC module:

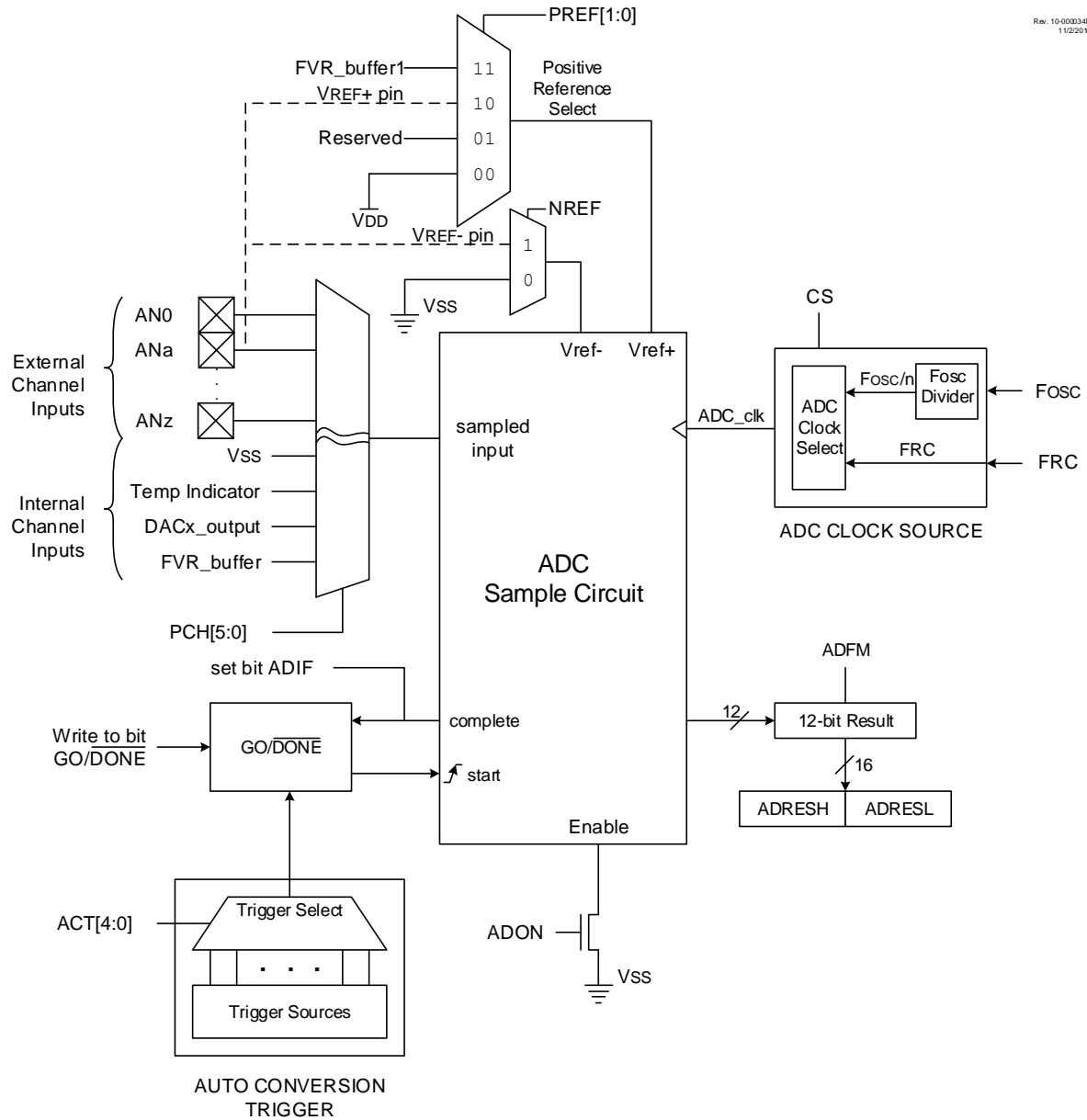
- 13-bit Acquisition Timer
- Hardware Capacitive Voltage Divider (CVD) Support:
 - 13-bit precharge timer
 - Adjustable sample and hold capacitor array
 - Guard ring digital output drive
- Automatic Repeat and Sequencing:
 - Automated double sample conversion for CVD
 - Two sets of result registers (Result and Previous result)
 - Auto-conversion trigger
 - Internal retrigger
- Computation Features:
 - Averaging and low-pass filter functions
 - Reference comparison
 - 2-level threshold comparison
 - Selectable interrupts

The figure below shows the block diagram of the ADC.

The ADC voltage reference is software selectable to be either internally generated or externally supplied.

The ADC can generate an interrupt upon completion of a conversion and upon threshold comparison. These interrupts can be used to wake-up the device from Sleep.

Figure 32-1. ADC² Block Diagram



32.1 ADC Configuration

When configuring and using the ADC the following functions must be considered:

- Port Configuration
- Channel Selection
- ADC Voltage Reference Selection
- ADC Conversion Clock Source
- Interrupt Control
- Result Formatting
- Conversion Trigger Selection

- ADC Acquisition Time
- ADC Precharge Time
- Additional Sample-and-Hold Capacitor
- Single/Double Sample Conversion
- Guard Ring Outputs

32.1.1 Port Configuration

The ADC can be used to convert both analog and digital signals. When converting analog signals, the I/O pin should be configured for analog by setting the associated TRIS and ANSEL bits. Refer to the “I/O Ports” section for more information.



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

Related Links

[14. I/O Ports](#)

32.1.2 Channel Selection

The [ADPCH](#) register determines which channel is connected to the sample and hold circuit for conversion. When changing channels, a delay is required before starting the next conversion. Refer to the [32.2 ADC Operation](#) section for more information.

The channel selections available using [PCH](#) bits in [ADPCH](#) register.

Related Links

[32.2 ADC Operation](#)

[30. FVR - Fixed Voltage Reference](#)

[31. Temperature Indicator Module](#)

[33. DAC - 5-Bit Digital-to-Analog Converter](#)

32.1.3 ADC Voltage Reference

The [PREF](#) bits provide control of the positive voltage reference. The positive voltage reference can be:

- V_{REF+} pin
- V_{DD}
- FVR 1.024V
- FVR 2.048V
- FVR 4.096V

The [NREF](#) bit provides control of the negative voltage reference. The negative voltage reference can be:

- V_{REF-} pin
- V_{SS}

Related Links

[30. FVR - Fixed Voltage Reference](#)

[32.8.20 ADREF](#)

32.1.4 Conversion Clock

The source of the conversion clock is software selectable using [CS](#) bit in the [ADCON0](#) register. If F_{OSC} is selected as the ADC clock ($CS = 0$), a prescaler can be used to divide the input clock in order to meet the ADC clock period specification. The prescaler options are available via [ADCLK](#) register.

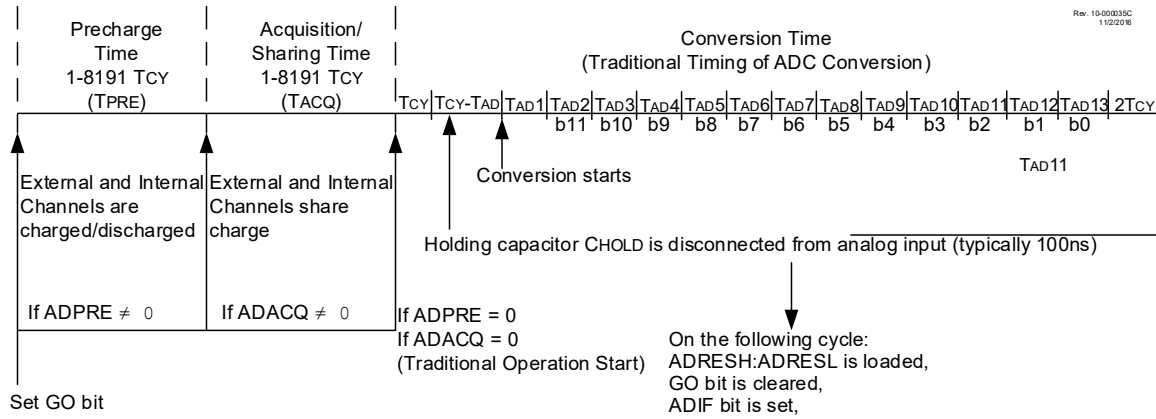
The available ADC clock source options are the following:

- $CS = 0$: $F_{OSC}/2^{(n+1)}$, where ‘n’ is from 0 to 127

- CS = 1: F_{RC} (dedicated RC oscillator)

The time to complete one bit conversion is defined as the T_{AD}. The following figure shows the complete timing details of the ADC conversion.

Figure 32-2. Analog-to-Digital Conversion T_{AD} Cycles



For correct conversion, the appropriate T_{AD} specification must be met. Access the “ADC Timing Specifications” section for more information. The following table below gives examples of appropriate ADC clock selections.



Important:

1. Unless using the F_{RC}, any changes in the system clock frequency will change the ADC clock frequency, which may adversely affect the ADC result.
2. The internal control logic of the ADC runs off of the clock selected by the CS bit. What this can mean is when the CS is set to ‘1’ (ADC runs on FRC), there may be unexpected delays in operation when setting ADC control bits.

Table 32-1. ADC Clock Period (T_{AD}) Vs. Device Operating Frequencies^(1,4)

| ADC Clock Period (T _{AD}) | | Device Frequency (F _{OSC}) | | | | | | |
|-------------------------------------|-------------------|--------------------------------------|-------------------------|-----------------------|-----------------------|------------------------|------------------------|-------------------------|
| ADC Clock Source | ADCLK | 64 MHz | 32 MHz | 20 MHz | 16 MHz | 8 MHz | 4 MHz | 1 MHz |
| FRC | CS(ADCON0[4]) = 1 | 1.0-6.0 μs | 1.0-6.0 μs | 1.0-6.0 μs | 1.0-6.0 μs | 1.0-6.0 μs | 1.0-6.0 μs | 1.0-6.0 μs |
| FOSC/128 | 111111 | 2.0 μs | 4.0 μs | 6.4 μs | 8.0 μs | 16.0 μs ⁽³⁾ | 32.0 μs ⁽³⁾ | 128.0 μs ⁽³⁾ |
| ... | ... | ... | ... | ... | ... | ... | ... | ... |
| ... | ... | ... | ... | ... | ... | ... | ... | ... |
| FOSC/10 | 000100 | 250 ns ⁽²⁾ | 500 ns | 800 ns | 1.0 μs | 2.0 μs | 4.0 μs | 16.0 μs ⁽³⁾ |
| FOSC/8 | 000011 | 125 ns ⁽²⁾ | 250 ns ⁽²⁾ | 400 ns ⁽²⁾ | 500 ns | 1.0 μs | 2.0 μs | 8.0 μs |
| FOSC/6 | 000010 | 93.75 ns ⁽²⁾ | 187.5 ns ⁽²⁾ | 300 ns ⁽²⁾ | 375 ns ⁽²⁾ | 750 ns | 1.5 μs | 6.0 μs |
| FOSC/4 | 000001 | 62.5 ns ⁽²⁾ | 125 ns ⁽²⁾ | 200 ns ⁽²⁾ | 250 ns ⁽²⁾ | 500 ns | 1.0 μs | 4.0 μs |
| FOSC/2 | 000000 | 31.25 ns ⁽²⁾ | 62.5 ns ⁽²⁾ | 100 ns ⁽²⁾ | 125 ns ⁽²⁾ | 250 ns ⁽²⁾ | 500 ns | 2.0 μs |

.....continued

| ADC Clock Period (T_{AD}) | | Device Frequency (F_{OSC}) | | | | | | |
|-------------------------------|-------|--------------------------------|--------|--------|--------|-------|-------|-------|
| ADC Clock Source | ADCLK | 64 MHz | 32 MHz | 20 MHz | 16 MHz | 8 MHz | 4 MHz | 1 MHz |

Note:

1. See T_{AD} parameter in the “*Electrical Specifications*” section for F_{RC} source typical T_{AD} value.
2. These values violate the required T_{AD} time.
3. Outside the recommended T_{AD} time.
4. The ADC clock period (T_{AD}) and total ADC conversion time can be minimized when the ADC clock is derived from the system clock F_{OSC} . However, the F_{RC} oscillator source must be used when conversions are to be performed with the device in Sleep mode.

Related Links

[32.8.15 ADCON0](#)

[40.4.8 Analog-to-Digital Converter \(ADC\) Conversion Timing Specifications](#)

[40. Electrical Specifications](#)

32.1.5 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.



Important:

1. The ADIF bit is set at the completion of every conversion, regardless of whether or not the ADC interrupt is enabled.
2. The ADC operates during Sleep only when the F_{RC} oscillator is selected.

This interrupt can be generated while the device is operating or while in Sleep. If the device is in Sleep, the interrupt will wake-up the device. Upon waking from Sleep, the next instruction following the `SLEEP` instruction is always executed. If the user is attempting to wake-up from Sleep and resume in-line code execution, the ADIE bit and the PEIE bit of the INTCON register must both be set and the GIE bit of the INTCON register must be cleared. If all three of these bits are set, the execution will switch to the Interrupt Service Routine.

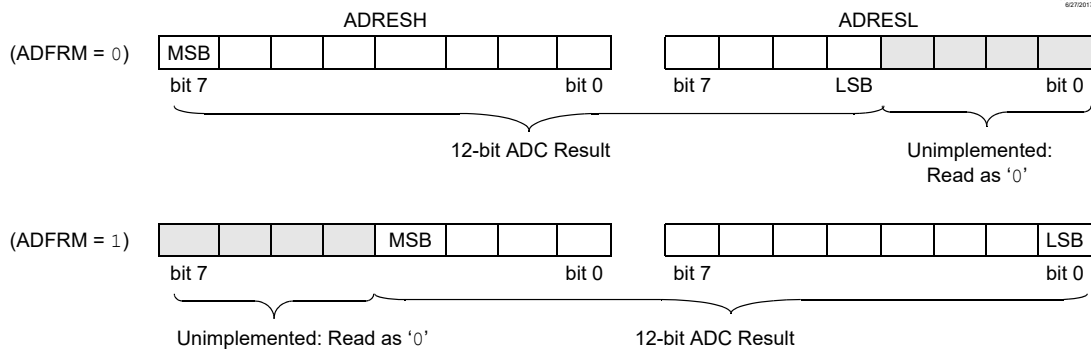
32.1.6 Result Formatting

The 12-bit ADC conversion result can be supplied in two formats, left justified or right justified. The `FRM` bit controls the output format.

The figure below shows the two output formats.

Writes to the ADRES register pair are always right justified regardless of the selected format mode. Therefore, data read after writing to ADRES when `FRM = 0` will be shifted left four places.

Figure 32-3. 12-Bit ADC Conversion Result Format



32.2 ADC Operation

32.2.1 Starting a Conversion

To enable the ADC module, the **ON** bit in the ADCON0 register must be set to a '1'. A conversion may be started by any of the following:

- Software setting the **GO** bit to '1'
- An external trigger (source selected by [32.8.21 ADACT](#))
- A continuous-mode retrigger (see “*Continuous Sampling mode*” section.)



Important: The **GO** bit should not be set in the same instruction that turns on the ADC.

Related Links

- [32.2.7 ADC Conversion Procedure \(Basic Mode\)](#)
- [32.6.8 Continuous Sampling Mode](#)
- [32.8.15 ADCON0](#)
- [32.8.17 ADCON2](#)

32.2.2 Completion of a Conversion

When any individual conversion is complete, the value already in ADRES is written into ADPREV (if **PSIS** = 1) and the new conversion result appears in ADRES. When the conversion completes, the ADC module will:

- Clear the **GO** bit (unless the **CONT** bit is set)
- Set the ADIF Interrupt Flag bit
- Set the **MATH** bit in the ADSTAT register
- Update ADACC

When **DSEN** = 0 then after every conversion, or when **DSEN** = 1 then after every other conversion, the following events occur:

- ADERR is calculated
- ADTIF interrupt is set if ADERR calculation meets threshold comparison

Importantly, filter and threshold computations occur after the conversion itself is complete. As such, interrupt handlers responding to ADIF should check ADTIF before reading filter and threshold results.

32.2.3 Terminating a Conversion

If a conversion must be terminated before completion, the **GO** bit can be cleared in software. The ADRESH and ADRESL registers will be updated with the partially complete Analog-to-Digital conversion sample. Incomplete bits will match the last bit converted. In this case, filter and/or threshold occur.



Important: A device Reset forces all registers to their Reset state. Thus, the ADC module is turned off and any pending conversion is terminated.

32.2.4 ADC Operation During Sleep

The ADC module can operate during Sleep. This requires the ADC clock source to be set to internal RC oscillator (**CS** bit in the ADCON0 register is set to '1'). When the F_{RC} is selected as ADC clock source, 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 **ON** bit remains set.

32.2.5 External Trigger During Sleep

If the external trigger is received during Sleep while the ADC clock source is set to the FRC, the ADC module will perform the conversion and set the ADIF bit upon completion.

If an external trigger is received when the ADC clock source is something other than FRC, the trigger will be recorded, but the conversion will not begin until the device exits Sleep.

32.2.6 Auto-Conversion Trigger

The auto-conversion trigger allows periodic ADC measurements without software intervention. When a rising edge of the selected source occurs, the **GO** bit is set by hardware.

The auto-conversion trigger source is selected by the **ACT** bits in the ADOACT register.

Using the Auto-conversion Trigger does not assure proper ADC timing. It is the user's responsibility to ensure that the ADC timing requirements are met. See [ADC Auto-Conversion Trigger Sources](#) table for auto-conversion sources.

32.2.7 ADC Conversion Procedure (Basic Mode)

This is an example procedure for using the ADC to perform an Analog-to-Digital conversion:

1. Configure Port:
 - 1.1. Disable pin output driver (Refer to the TRISx register)
 - 1.2. Configure pin as analog (Refer to the ANSELx register)
2. Configure the ADC module:
 - 2.1. Select ADC conversion clock
 - 2.2. Configure voltage reference
 - 2.3. Select ADC input channel (precharge+acquisition)
 - 2.4. Turn on ADC module
3. Configure ADC interrupt (optional):
 - 3.1. Clear ADC interrupt flag
 - 3.2. Enable ADC interrupt
 - 3.3. Enable peripheral interrupt (PIE bit)
 - 3.4. Enable global interrupt (GIE bit) (see Note 1 below)
4. If ADACQ = 0, software must wait the required acquisition time (see Note 2 below).
5. Start conversion by setting the **GO** bit.
6. Wait for ADC conversion to complete by one of the following:
 - 6.1. Polling the **GO** bit

- 6.2. Polling the ADIF bit
- 6.3. Waiting for the ADC interrupt (interrupts enabled)
7. Read ADC Result.
8. Clear the ADC interrupt flag (required if interrupt is enabled).



Important:

1. The global interrupt can be disabled if the user is attempting to wake-up from Sleep and resume in-line code execution.
2. Refer to the “ADC Acquisition Requirements” section.

Example 32-1. ADC Conversion (assembly)

```
; This code block configures the ADC for polling, VDD and VSS references,  
; FRC oscillator, and AN0 input.  
; Conversion start & polling for completion are included.  
;  
BANKSEL ADCON1 ;  
movlw B'11110000' ;Right justify, FRC oscillator  
movwf ADCON1 ;Vdd and Vss Vref  
BANKSEL TRISA ;  
bsf TRISA,0 ;Set RA0 to input  
BANKSEL ANSEL ;  
bsf ANSEL,0 ;Set RA0 to analog  
BANKSEL ADCON0 ;  
movlw B'00000001' ;Select channel AN0  
movwf ADCON0 ;Turn ADC On  
call SampleTime ;Acquisiton delay  
bsf ADCON0,ADGO ;Start conversion  
btfsc ADCON0,ADGO ;Is conversion done?  
goto $-1 ;No, test again  
BANKSEL ADRESH ;  
movf ADRESH,W ;Read upper 2 bits  
movwf RESULTHI ;store in GPR space  
movf ADRESL,W ;Read lower 8 bits  
movwf RESULTLO ;Store in GPR space
```

Example 32-2. ADC Conversion (C)

```
/*This code block configures the ADC for polling, VDD and VSS references, FRC  
oscillator and AN0 input.  
Conversion start & polling for completion are included.  
*/  
void main() {  
//System Initialize  
initializeSystem();  
  
//Setup ADC  
ADCON0bits.FM = 1; //right justify  
ADCON0bits.CS = 1; //FRC Clock  
ADPCH = 0x00; //RA0 is Analog channel  
TRISAbits.TRISA0 = 1; //Set RA0 to input  
ANSELAbits.ANSELA0 = 1; //Set RA0 to analog  
ADCON0bits.ON = 1; //Turn ADC On  
  
while (1) {  
ADCON0bits.GO = 1; //Start conversion  
while (ADCON0bits.GO); //Wait for conversion done  
resultHigh = ADRESH; //Read result  
resultLow = ADRESL; //Read result  
}  
}
```

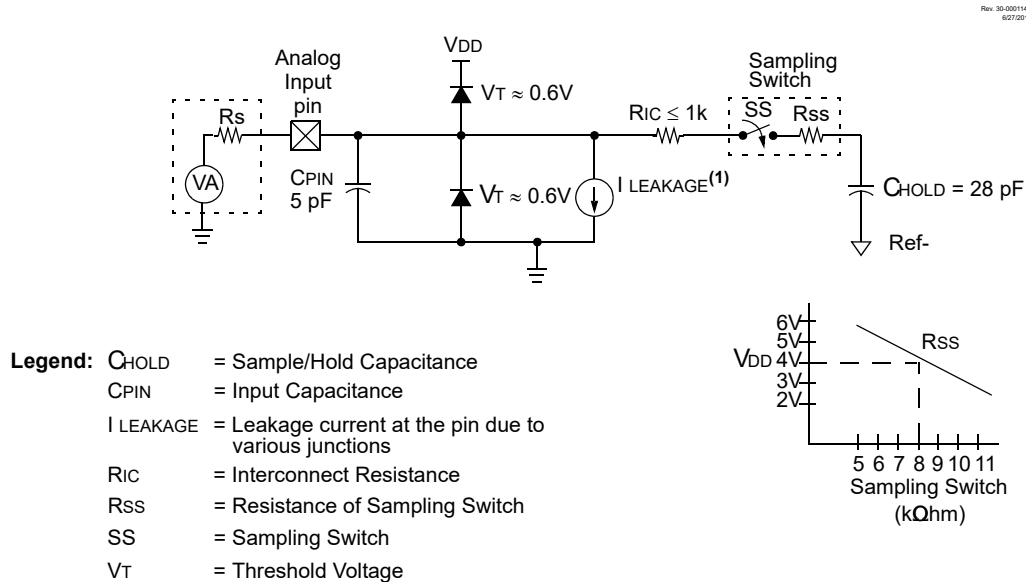
Related Links

32.3 ADC Acquisition Requirements

32.3 ADC Acquisition Requirements

For the ADC to meet its specified accuracy, the charge holding capacitor (C_{HOLD}) must be allowed to fully charge to the input channel voltage level. The Analog Input model is shown in the following figure. The source impedance (R_S) and the internal sampling switch (R_{SS}) impedance directly affect the time required to charge the capacitor C_{HOLD} . The sampling switch (R_{SS}) impedance varies over the device voltage (V_{DD}), refer to the following figure.

Figure 32-4. Analog Input Model



Note:

1. Refer to "I/O Ports Electrical Specifications".



Important: The maximum recommended impedance for analog sources is 10 kΩ.

If the source impedance is decreased, the acquisition time may be decreased. After the analog input channel is selected (or changed), an ADC acquisition must be completed before the conversion can be started. To calculate the minimum acquisition time, the following equation may be used. This equation assumes that 1/2 LSB error is used (4,096 steps for the ADC). The 1/2 LSB error is the maximum error allowed for the ADC to meet its specified resolution.

Equation 32-1. Acquisition Time Example

Assumptions: Temperature = 50°C and external impedance pf 10 kΩ 5.0V V_{DD}

T_{ACQ} = Amplifier Settling Time + Hold Capacitor Charging Time + Temperature Coefficient

$$= T_{AMP} + T_C + T_{COFF}$$

$$= 2 \mu s + T_C + [(Temperature - 25^\circ C)(0.05 \mu s/^\circ C)]$$

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

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

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

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

Note: Where n = number of bits of the ADC.

Solving for T_C :

$$T_C = -C_{HOLD}(R_{IC} + R_{SS} + R_S) \ln(1/8191)$$

$$T_C = -28pF(1k\Omega + 7k\Omega + 10k\Omega) \ln(0.0001221)$$

$$T_C = 4.54\mu s$$

Therefore:

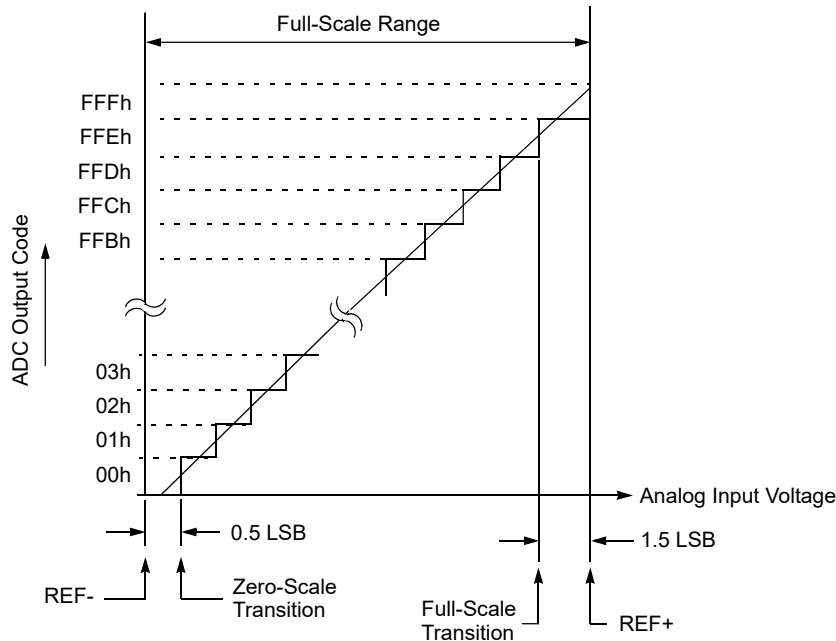
$$T_{ACQ} = 2\mu s + 4.54\mu s + [(50^\circ C - 25^\circ C)(0.05\mu s/^\circ C)]$$

$$T_{ACQ} = 7.79\mu s$$

Note:

1. The reference voltage (V_{REF}) has no effect on the equation, since it cancels itself out.
2. The charge holding capacitor (C_{HOLD}) is not discharged after each conversion.
3. The maximum recommended impedance for analog sources is 10 k Ω . This is required to meet the pin leakage specification.

Figure 32-5. ADC Transfer Function



Related Links

[40.4.4 I/O and CLKOUT Timing Specifications](#)

32.4 ADC Charge Pump

The ADC module has a dedicated charge pump that can be controlled through the ADCPCON0 register. The primary purpose of the charge pump is to supply a constant voltage to the gates of transistor devices in the A/D converter, signal and reference input pass-gates, to prevent degradation of transistor performance at low operating voltage.

The charge pump can be enabled by setting the **CPON** bit. Once enabled, the pump will undergo a start-up time to stabilize the charge pump output. Once the output stabilizes and is ready for use, the **CPRDY** bit will be set.

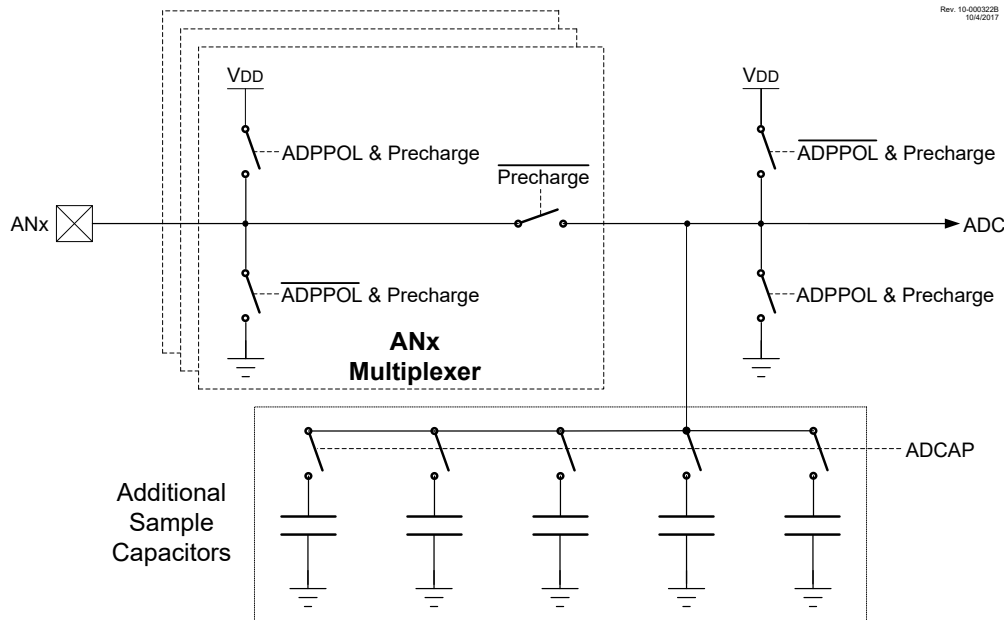
Related Links

[32.8.23 ADCPCON0](#)

32.5 Capacitive Voltage Divider (CVD) Features

The ADC module contains several features that allow the user to perform a relative capacitance measurement on any ADC channel using the internal ADC sample and hold capacitance as a reference. This relative capacitance measurement can be used to implement capacitive touch or proximity sensing applications. The following figure shows the basic block diagram of the CVD portion of the ADC module.

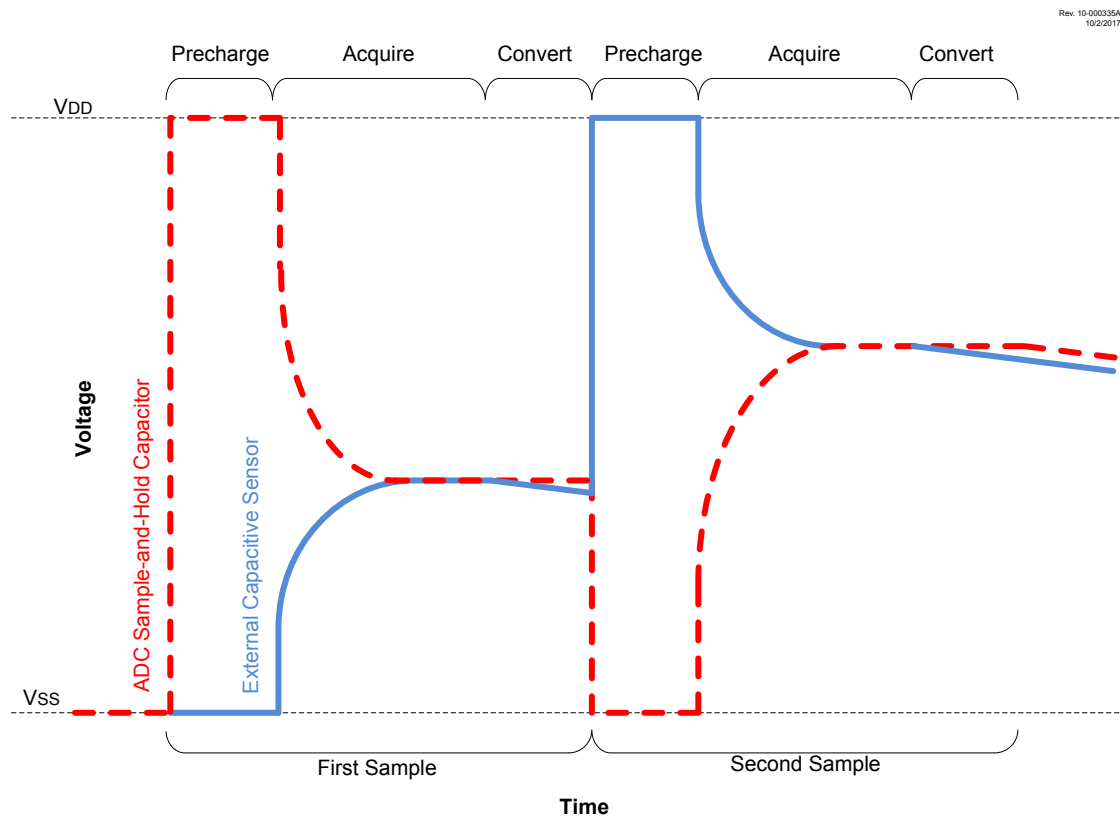
Figure 32-6. Hardware Capacitive Voltage Divider Block Diagram



32.5.1 CVD Operation

A CVD operation begins with the ADC's internal Sample-and-Hold capacitor (C_{HOLD}) being disconnected from the path which connects it to the external capacitive sensor node. While disconnected, C_{HOLD} is precharged to V_{DD} or V_{SS} the sensor node is also charged to V_{SS} or V_{DD} , respectively to the level opposite that of C_{HOLD} . When the precharge phase is complete, the V_{DD}/V_{SS} bias paths for the two nodes are shut off and the paths between C_{HOLD} and the external sensor node is reconnected, at which time the acquisition phase of the CVD operation begins. During acquisition, a capacitive voltage divider is formed between the precharged C_{HOLD} and sensor nodes, which results in a final voltage level setting on C_{HOLD} which is determined by the capacitances and precharge levels of the two nodes. After acquisition, the ADC converts the voltage level on C_{HOLD} . This process is then repeated with the selected precharge levels inverted for both the C_{HOLD} and the sensor nodes. The waveform for two CVD measurements, which is known as differential CVD measurement, is shown in the following figure.

Figure 32-7. Differential CVD Measurement Waveform



32.5.2 Precharge Control

The Precharge stage is an optional period of time that brings the external channel and internal sample and hold capacitor to known voltage levels. Precharge is enabled by writing a non-zero value to the ADPRE register. This stage is initiated when an ADC conversion begins, either from setting the GO bit, a special event trigger, or a conversion restart from the computation functionality. If the ADPRE register is cleared when an ADC conversion begins, this stage is skipped.

During the precharge time, C_{HOLD} is disconnected from the outer portion of the sample path that leads to the external capacitive sensor and is connected to either V_{DD} or V_{SS} , depending on the value of the PPOL bit. At the same time, the port pin logic of the selected analog channel is overridden to drive a digital high or low out, in order to precharge the outer portion of the ADC's sample path, which includes the external sensor. The output polarity of this override is also determined by the PPOL bit. The amount of time that this charging needs is controlled by the ADPRE register.



Important: The external charging overrides the TRIS setting of the respective I/O pin. If there is a device attached to this pin, Precharge should not be used.

Related Links

[32.8.16 ADCON1](#)
[32.8.14 ADPRE](#)

32.5.3 Acquisition Control for CVD

The Acquisition stage allows time for the voltage on the internal sample and hold capacitor to charge or discharge from the selected analog channel. This acquisition time is controlled by the ADACQ register. If $ADPRE = 0$,

acquisition starts at the beginning of conversion. When $ADPRE \neq 0$, the acquisition stage begins when precharge ends.

At the start of the acquisition stage, the port pin logic of the selected analog channel is overridden to turn off the digital high/low output drivers so they do not affect the final result of the charge averaging. Also, the selected ADC channel is connected to C_{HOLD} . This allows charge averaging to proceed between the precharged channel and the C_{HOLD} capacitor.



Important: When $ADPRE \neq 0$ setting $ADACQ$ to '0' will set a maximum acquisition time (8191 ADC clock cycles). When $ADPRE = 0$, setting $ADACQ$ to '0' will disable hardware acquisition time control.

32.5.4 Guard Ring Outputs

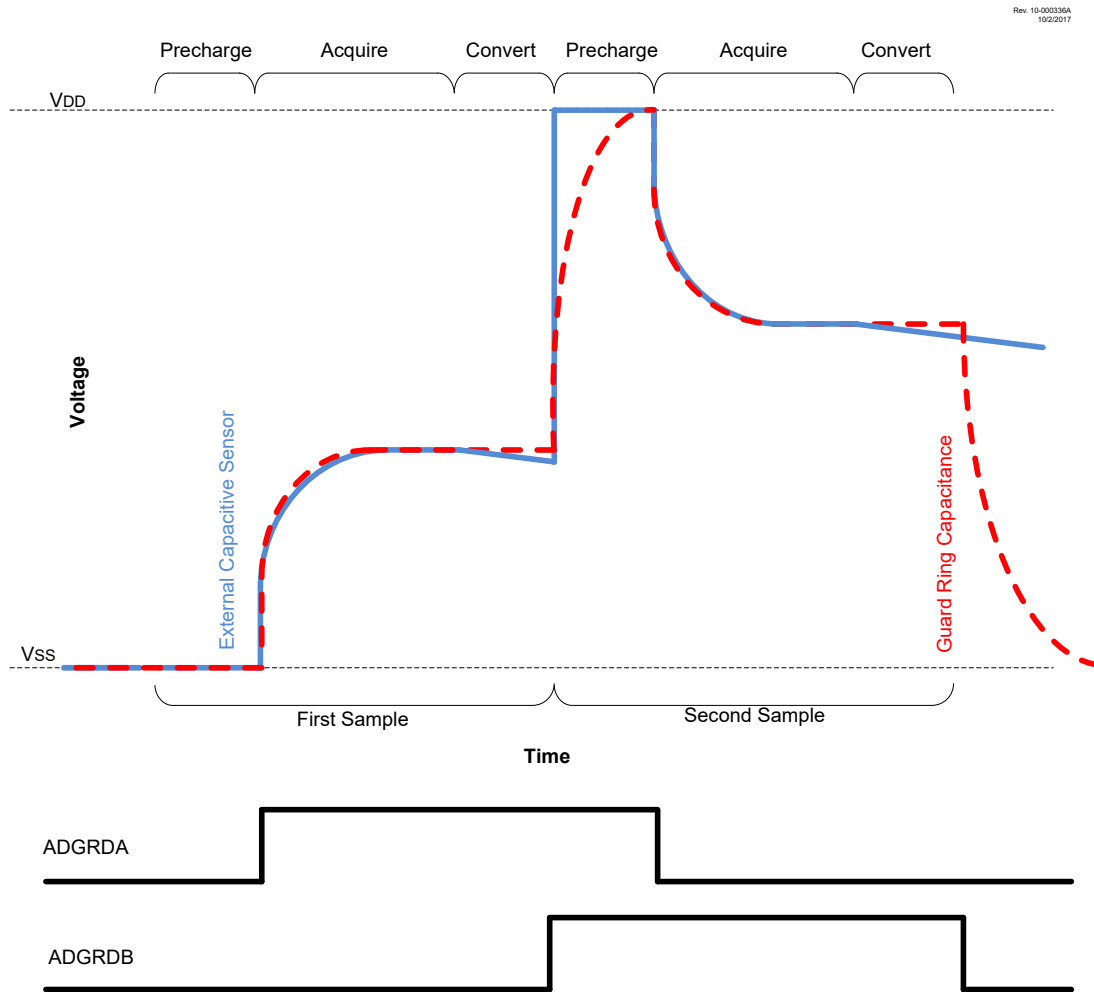
The following figure shows a typical guard ring circuit. C_{GUARD} represents the capacitance of the guard ring trace placed on the PCB board. The user selects values for R_A and R_B that will create a voltage profile on C_{GUARD} , which will match the selected acquisition channel.

The purpose of the guard ring is to generate a signal in phase with the CVD sensing signal to minimize the effects of the parasitic capacitance on sensing electrodes. It also can be used as a mutual drive for mutual capacitive sensing. For more information about active guard and mutual drive, see [Application Note AN1478, "mTouch™ Sensing Solution Acquisition Methods Capacitive Voltage Divider"](#).

The ADC has two guard ring drive outputs, $ADGRDA$ and $ADGRDB$. These outputs can be routed through PPS controls to I/O pins (see "[Peripheral Pin Select \(PPS\) Module](#)" for details) and the polarity of these outputs are controlled by the $GPOL$ and $IPEN$ bits.

At the start of the first precharge stage, both outputs are set to match the $GPOL$ bit. Once the acquisition stage begins, $ADGRDA$ changes polarity, while $ADGRDB$ remains unchanged. When performing a double sample conversion, setting the $IPEN$ bit causes both guard ring outputs to transition to the opposite polarity of $GPOL$ at the start of the second precharge stage, and $ADGRDA$ toggles again for the second acquisition. For more information on the timing of the guard ring output, refer to the two following figures.

Figure 32-8. Differential CVD with Guard Ring Output Waveform



Related Links

- [32.8.16 ADCON1](#)
- [16. PPS - Peripheral Pin Select Module](#)

32.5.5 Additional Sample and Hold Capacitance

Additional capacitance can be added in parallel with the internal sample and hold capacitor (C_{HOLD}) by using the ADCAP register. This register selects a digitally programmable capacitance, which is added to the ADC conversion bus, increasing the effective internal capacitance of the sample and hold capacitor in the ADC module. This is used to improve the match between internal and external capacitance for a better sensing performance. The additional capacitance does not affect analog performance of the ADC because it is not connected during conversion.

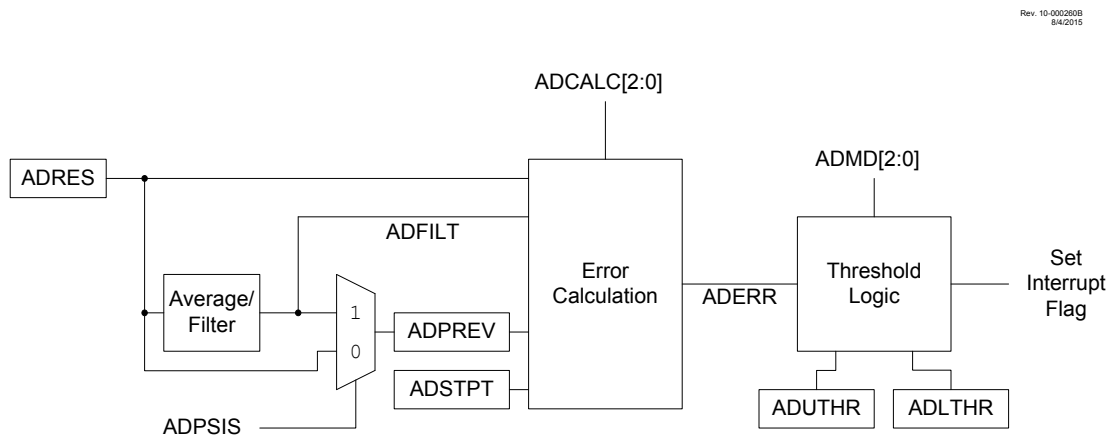
Related Links

- [32.6 Computation Operation](#)
- [32.8.13 ADCAP](#)

32.6 Computation Operation

The ADC module hardware is equipped with post conversion computation features. These features provide data post-processing functions that can be applied to the ADC conversion result, including digital filtering/averaging and threshold comparison functions.

Figure 32-9. Computational Features Simplified Block Diagram



The operation of the ADC computational features is controlled by **MD** bits.

The module can be operated in one of five modes:

- **Basic:** This is a legacy mode. In this mode, ADC conversion occurs on single (**DSEN** = 0) or double (**DSEN** = 1) samples. **ADIF** is set after all the conversions are complete.
- **Accumulate:** With each trigger, the ADC conversion result is added to accumulator and **ADCNT** increments. **ADIF** is set after each conversion. **ADTIF** is set according to the calculation mode.
- **Average:** With each trigger, the ADC conversion result is added to the accumulator. When the **ADRPT** number of samples have been accumulated, a threshold test is performed. Upon the next trigger, the accumulator is cleared. For the subsequent tests, additional **ADRPT** samples are required to be accumulated.
- **Burst Average:** At the trigger, the accumulator is cleared. The ADC conversion results are then collected repetitively until **ADRPT** samples are accumulated and finally the threshold is tested.
- **Low-Pass Filter (LPF):** With each trigger, the ADC conversion result is sent through a filter. When **ADRPT** samples have occurred, a threshold test is performed. Every trigger after that, the ADC conversion result is sent through the filter and another threshold test is performed.

The five modes are summarized in following table.

Table 32-2. Computation Modes

| Mode | MD | Bit Clear Conditions | Value after Trigger completion | | Threshold Operations | | | Va |
|-----------------|-----|--|--|--|-------------------------------|------------------|-------------------|-------------|
| | | ADACC and ADCNT | ADACC | ADCNT | Retrigger | Threshold Test | Interrupt | |
| Low-pass Filter | 100 | ADACL _R = 1 | $S + ADACC - \frac{ADACC}{2^{ADC_{RS}}}$ or $(S2 - S1) + ADACC - \frac{ADACC}{2^{ADC_{RS}}}$ | If (ADCNT=FF): ADCNT, otherwise: ADCNT+1 | No | If ADCNT ≥ ADRPT | If threshold=true | ADA Over |
| Burst Average | 011 | ADACL _R = 1 or ADGO set or retrigger | Each repetition: same as Average End with sum of all samples | Each repetition: same as Average End with ADCNT=ADRPT | Repeat while ADCNT < ADRPT | If ADCNT ≥ ADRPT | If threshold=true | ADA Over |
| Average | 010 | ADACL _R = 1 or ADCNT ≥ ADRPT at ADGO or retrigger | S + ADACC or (S2-S1) + ADACC | If (ADCNT=FF): ADCNT, otherwise: ADCNT+1 | No | If ADCNT ≥ ADRPT | If threshold=true | ADA Over |
| Accumulate | 001 | ADACL _R = 1 | S + ADACC or (S2-S1) + ADACC | If (ADCNT=FF): ADCNT, otherwise: ADCNT+1 | No | Every Sample | If threshold=true | ADA Over |
| Basic | 000 | ADACL _R = 1 | Unchanged | Unchanged | No | Every Sample | If threshold=true | N |

Note: S1 and S2 are abbreviations for Sample 1 and Sample 2, respectively. When **DSEN** = 0, S1 = ADRES; When **DSEN** = 1, S1 = ADPREV

32.6.1 Digital Filter/Average

The digital filter/average module consists of an accumulator with data feedback options, and control logic to determine when threshold tests need to be applied. The accumulator is a 16-bit wide register that can be accessed through the ADACCH:ADACCL register pair.

Upon each trigger event (the GO bit set or external event trigger), the ADC conversion result is added to the accumulator. If the accumulated result exceeds $2^{(\text{accumulator_width})-1}$, the OV Accumulator Overflow bit is set.

The number of samples to be accumulated is determined by the ADRPT (A/D Repeat Setting) register. Each time a sample is added to the accumulator, the ADCNT register is incremented. Once ADRPT samples are accumulated (ADCNT = ADRPT), an accumulator clear command can be issued by the software by setting the ACLR bit. Setting the ACLR bit will also clear the OV bit, as well as the ADCNT register. The ACLR bit is cleared by the hardware when accumulator clearing action is complete.



Important: When ADC is operating from FRC, five FRC clock cycles are required to execute the ADACC clearing operation.

The CRS bits control the data shift on the accumulator result, which effectively divides the value in accumulator (ADACCU:ADACCH:ADACCL) register pair. For the Accumulate mode of the digital filter, the shift provides a simple scaling operation. For the Average/Burst Average mode, the Shift bits are used to determine the number of logical right shifts to be performed on the accumulated result. For the Low-pass Filter mode, the shift is an integral part of the filter, and determines the cut-off frequency of the filter. The table below shows the -3 dB cut-off frequency in ωT (radians) and the highest signal attenuation obtained by this filter at nyquist frequency ($\omega T = \pi$).

Table 32-3. Low-pass Filter -3 dB Cut-off Frequency

| CRS | ωT (radians) @ -3 dB Frequency | dB @ $F_{\text{nyquist}}=1/(2T)$ |
|-----|--|----------------------------------|
| 1 | 0.72 | -9.5 |
| 2 | 0.284 | -16.9 |
| 3 | 0.134 | -23.5 |
| 4 | 0.065 | -29.8 |
| 5 | 0.032 | -36.0 |
| 6 | 0.016 | -42.0 |
| 7 | 0.0078 | -48.1 |

32.6.2 Basic Mode

Basic mode ($MD = 000$) disables all additional computation features. In this mode, no accumulation occurs but threshold error comparison is performed. Double sampling, Continuous mode, and all CVD features are still available, but no features involving the digital filter/average features are used.

32.6.3 Accumulate Mode

In Accumulate mode ($MD = 001$), after every conversion, the ADC result is added to the ADACC register. The ADACC register is right-shifted by the value of the CRS bits. This right-shifted value is copied into the ADFLT register. The Formatting mode does not affect the right-justification of the ADFLT value. Upon each sample, ADCNT is also incremented, incrementing the number of samples accumulated. After each sample and accumulation, the ADACC value has a threshold comparison performed on it and the ADTIF interrupt may trigger.

Related Links

[32.6.7 Threshold Comparison](#)

32.6.4 Average Mode

In Average Mode ($MD = 010$), the ADACC registers accumulate with each ADC sample, much as in Accumulate mode, and the ADCNT register increments with each sample. The ADFLT register is also updated with the right-

shifted value of the ADACC register. The value of the **CRS** bits govern the number of right shifts. However, in Average mode, the threshold comparison is performed upon ADCNT being greater than or equal to a user-defined ADRPT value. In this mode when $ADCNT = 2^{ADCRS}$, then the final accumulated value will be divided by number of samples, allowing for a threshold comparison operation on the average of all gathered samples.

32.6.5 Burst Average Mode

The Burst Average mode (**MD** = 011) acts the same as the Average mode in most respects. The one way it differs is that it continuously retriggers ADC sampling until the ADCNT value is greater than or equal to ADRPT, even if Continuous Sampling mode is not enabled. This allows for a threshold comparison on the average of a short burst of ADC samples.

Related Links

[32.6.8 Continuous Sampling Mode](#)

32.6.6 Low-pass Filter Mode

The Low-pass Filter mode (**MD** = 100) acts similarly to the Average mode in how it handles samples (accumulates samples until ADCNT value greater than or equal to ADRPT, then triggers threshold comparison), but instead of a simple average, it performs a low-pass filter operation on all of the samples, reducing the effect of high-frequency noise on the average, then performs a threshold comparison on the results. In this mode, the **CRS** bits determine the cut-off frequency of the low-pass filter.

Related Links

[32.6.1 Digital Filter/Average](#)

[32.6 Computation Operation](#)

32.6.7 Threshold Comparison

At the end of each computation:

- The conversion results are latched and held stable at the end-of-conversion.
- The error (ADERR) is calculated based on a difference calculation which is selected by the **CALC** bits in the ADCON3 register. The value can be one of the following calculations:
 - The first derivative of single measurements
 - The CVD result in CVD mode
 - The current result vs. a setpoint
 - The current result vs. the filtered/average result
 - The first derivative of the filtered/average value
 - Filtered/average value vs. a setpoint
- The result of the calculation (ADERR) is compared to the upper and lower thresholds, ADUTH[UTHH:UTHL] and ADLTH[LTHH:LTHL] registers, to set the UTHR and LTHR flag bits. The threshold logic is selected by **TMD** bits. The threshold trigger option can be one of the following:
 - Never interrupt
 - Error is less than lower threshold
 - Error is greater than or equal to lower threshold
 - Error is between thresholds (inclusive)
 - Error is outside of thresholds
 - Error is less than or equal to upper threshold
 - Error is greater than upper threshold
 - Always interrupt regardless of threshold test results
 - If the threshold condition is met, the threshold interrupt flag ADTIF is set.

Note:

1. The threshold tests are signed operations.
2. If **OV** is set, a threshold interrupt is signaled. It is good practice for threshold interrupt handlers to verify the validity of the threshold by checking ADAOV.

See [ADC Error Calculation Mode](#) table for further details.

32.6.8 Continuous Sampling Mode

Setting the **CONT** bit automatically retriggers a new conversion cycle after updating the ADACC register. The **GO** bit remains set and retriggering occurs automatically.

If **SOI** = 1, a threshold interrupt condition will clear **GO** and the conversions will stop.

32.6.9 Double Sample Conversion

Double sampling is enabled by setting the **DSEN** bit. When this bit is set, two conversions are required before the module will calculate threshold error (each conversion must still be triggered separately). The first conversion will set the **MATH** bit and update ADACC, but will not calculate ADERR or trigger ADTIF. When the second conversion completes, the first value is transferred to ADPREV (depending on the setting of **PSIS**) and the value of the second conversion is placed into ADRES. Only upon the completion of the second conversion is ADERR calculated and ADTIF triggered (depending on the value of **CALC**).

32.7 Register Summary - ADC Control

| Address | Name | Bit Pos. | | | | | | | | |
|-------------------------|----------|----------|------------|------|-----------|------|------|-----|--|-----------|
| 0x00 ... 0x8B | Reserved | | | | | | | | | |
| 0x8C | ADLTH | 7:0 | LTH[7:0] | | | | | | | |
| | | 15:8 | LTH[15:8] | | | | | | | |
| 0x8E | ADUTH | 7:0 | UTH[7:0] | | | | | | | |
| | | 15:8 | UTH[15:8] | | | | | | | |
| 0x90 | ADERR | 7:0 | ERR[7:0] | | | | | | | |
| | | 15:8 | ERR[15:8] | | | | | | | |
| 0x92 | ADSTPT | 7:0 | STPT[7:0] | | | | | | | |
| | | 15:8 | STPT[15:8] | | | | | | | |
| 0x94 | ADFLTR | 7:0 | FLTR[7:0] | | | | | | | |
| | | 15:8 | FLTR[15:8] | | | | | | | |
| 0x96 | ADACC | 7:0 | ACC[7:0] | | | | | | | |
| | | 15:8 | ACC[15:8] | | | | | | | |
| | | 23:16 | ACC[17:16] | | | | | | | |
| 0x99 | ADCNT | 7:0 | CNT[7:0] | | | | | | | |
| 0x9A | ADRPT | 7:0 | RPT[7:0] | | | | | | | |
| 0x9B | ADPREV | 7:0 | PREV[7:0] | | | | | | | |
| | | 15:8 | PREV[15:8] | | | | | | | |
| 0x9D | ADRES | 7:0 | RES[7:0] | | | | | | | |
| | | 15:8 | RES[15:8] | | | | | | | |
| 0x9F | ADPCH | 7:0 | PCH[5:0] | | | | | | | |
| 0xA0 ... 0x010B | Reserved | | | | | | | | | |
| 0x010C | ADACQ | 7:0 | ACQ[7:0] | | | | | | | |
| | | 15:8 | ACQ[12:8] | | | | | | | |
| 0x010E | ADCAP | 7:0 | CAP[4:0] | | | | | | | |
| 0x010F | ADPRE | 7:0 | PRE[7:0] | | | | | | | |
| | | 15:8 | PRE[12:8] | | | | | | | |
| 0x0111 | ADCON0 | 7:0 | ON | CONT | | CS | | FRM | | GO |
| 0x0112 | ADCON1 | 7:0 | PPOL | IPEN | GPOL | | | | | DSEN |
| 0x0113 | ADCON2 | 7:0 | PSIS | | CRS[2:0] | | ACLR | | | MD[2:0] |
| 0x0114 | ADCON3 | 7:0 | | | CALC[2:0] | | SOI | | | TMD[2:0] |
| 0x0115 | ADSTAT | 7:0 | OV | UTHR | LTHR | MATH | | | | STAT[2:0] |
| 0x0116 | ADREF | 7:0 | | | | NREF | | | | PREF[1:0] |
| 0x0117 | ADACT | 7:0 | | | | | | | | ACT[4:0] |
| 0x0118 | ADCLK | 7:0 | | | | | | | | CS[5:0] |
| 0x0119 ... 0x029E | Reserved | | | | | | | | | |
| 0x029F | ADPCON0 | 7:0 | CPON | | | | | | | CPRDY |

32.8 Register Definitions: ADC Control

32.8.1 ADLTH

Name: ADLTH
Address: 0x08C

ADC Lower Threshold Register

ADLTH and ADUTH are compared with ADERR to set the **UTHR** and **LTHR** bits. Depending on the setting of **TMD**, an interrupt may be triggered by the results of this comparison.

| | | | | | | | | | |
|--------|-----|-----------|-----|-----|-----|-----|-----|-----|-----|
| | Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| | | LTH[15:8] | | | | | | | |
| Access | | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | | | | | | | | |
| | Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | LTH[7:0] | | | | | | | |
| Access | | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bits 15:0 – LTH[15:0] ADC Lower Threshold - Signed 2's complement

Note: The individual bytes in this multi-byte register can be accessed with the following register names:

- ADLTHH: Accesses the high byte ADLTH[15:8]
- ADLTHL: Accesses the low byte ADLTH[7:0]

32.8.2 ADUTH

Name: ADUTH
Address: 0x08E

ADC Upper Threshold Register

ADLTH and ADUTH are compared with ADERR to set the **UTHR** and **LTHR** bits. Depending on the setting of **TMD**, an interrupt may be triggered by the results of this comparison.

| | | | | | | | | | |
|--------|-----|-----------|-----|-----|-----|-----|-----|-----|-----|
| | Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| | | UTH[15:8] | | | | | | | |
| Access | | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | | | | | | | | |
| | Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | UTH[7:0] | | | | | | | |
| Access | | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bits 15:0 – UTH[15:0] ADC Upper Threshold - Signed 2's complement

Note: The individual bytes in this multi-byte register can be accessed with the following register names:

- ADUTHH: Accesses the high byte ADUTH[15:8]
- ADUTHL: Accesses the low byte ADUTH[7:0]

32.8.3 ADERR

Name: ADERR

Address: 0x090

ADC Setpoint Error Register

ADC Setpoint Error calculation is determined by the [CALC](#) bits.

| | | | | | | | | | |
|--------|-----|-----------|----|----|----|----|----|---|---|
| | Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| | | ERR[15:8] | | | | | | | |
| Access | | R | R | R | R | R | R | R | R |
| Reset | | x | x | x | x | x | x | x | x |
| | | | | | | | | | |
| | Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | ERR[7:0] | | | | | | | |
| Access | | R | R | R | R | R | R | R | R |
| Reset | | x | x | x | x | x | x | x | x |

Bits 15:0 – ERR[15:0] ADC Setpoint Error - Signed 2's complement

Note: The individual bytes in this multi-byte register can be accessed with the following register names:

- ADERRH: Accesses the high byte ADERR[15:8]
- ADERRL: Accesses the low byte ADERR[7:0]

32.8.4 ADSTPT

Name: ADSTPT

Address: 0x092

ADC Threshold Setpoint Register

Depending on the CALC bits, it may be used to determine ADERR. See [ADC Error Calculation Mode](#) table for more details.

| | | | | | | | | |
|--------|------------|-----|-----|-----|-----|-----|-----|-----|
| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| | STPT[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | | | | | | | |
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | STPT[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bits 15:0 – STPT[15:0] ADC Threshold Setpoint - Signed 2's complement

Note: The individual bytes in this multi-byte register can be accessed with the following register names:

- ADSTPTH: Accesses the high byte ADSTPT[15:8]
- ADSTPLH: Accesses the low byte ADSTPT[7:0]

32.8.5 ADFLTR

Name: ADFLTR
Address: 0x094

ADC Filter Register

| | | | | | | | | |
|--------|------------|----|----|----|----|----|----|----|
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| | FLTR[15:8] | | | | | | | |
| Access | RO | RO | RO | RO | RO | RO | RO | RO |
| Reset | x | x | x | x | x | x | x | x |
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | FLTR[7:0] | | | | | | | |
| Access | RO | RO | RO | RO | RO | RO | RO | RO |
| Reset | x | x | x | x | x | x | x | x |

Bits 15:0 – FLTR[15:0] ADC Filter Output - Signed 2's complement

In Accumulate, Average, and Burst Average mode, this is equal to ADACC register right shifted by the **CRS** bits. In LPF mode, this is the output of the low-pass filter.

Note: The individual bytes in this multi-byte register can be accessed with the following register names:

- ADFLTRH: Accesses the high byte ADFLTR[15:8]
- ADFLTRL: Accesses the low byte ADFLTR[7:0]

32.8.6 ADACC

Name: ADACC
Address: 0x096

ADC Accumulator Register
 See [Computation Modes](#) for more details.



Important: This register contains signed 2's complement accumulator value and the upper unused bits contain copies of the sign bit.

| | | | | | | | | |
|--------|-----------|-----|-----|-----|-----|-----|------------|-----|
| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | | | | | | | ACC[17:16] | |
| Access | | | | | | | R/W | R/W |
| Reset | | | | | | | x | x |
| | | | | | | | | |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| | ACC[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | x | x | x | x | x | x | x | x |
| | | | | | | | | |
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | ACC[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | x | x | x | x | x | x | x | x |

Bits 17:0 – ACC[17:0] ADC Accumulator - Signed 2's complement

Note:

1. This register can only be written when GO = 0.
2. The individual bytes in this multi-byte register can be accessed with the following register names:
 - ADACCU: Accesses the upper byte ADACC[17:16]
 - ADACCH: Accesses the high byte ADACC[15:8]
 - ADACCL: Accesses the low byte ADACC[7:0]

32.8.7 ADCNT

Name: ADCNT

Address: 0x099

ADC Repeat Counter Register

| | | | | | | | | |
|--------|----------|-----|-----|-----|-----|-----|-----|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | CNT[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | x | x | x | x | x | x | x | x |

Bits 7:0 – CNT[7:0] ADC Repeat Count bits

Counts the number of times that the ADC is triggered before the threshold is checked. When this value reaches ADPRT then the threshold is checked. Used when the computation mode is Low-pass Filter, Burst Average, or Average. See [Computation Modes](#) for more details.

Reset States: POR/BOR = xxxxxxxx

All Other Resets = uuuuuuuu

32.8.8 ADRPT

Name: ADRPT
Address: 0x09A

ADC Repeat Setting Register

| | | | | | | | | |
|--------|----------|-----|-----|-----|-----|-----|-----|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | RPT[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bits 7:0 – RPT[7:0] ADC Repeat Threshold bits

Determines the number of times that the ADC is triggered for a threshold check. When ADCNT reaches this value the error threshold is checked. Used when the computation mode is Low-pass Filter, Burst Average, or Average. See [Computation Modes](#) for more details.

32.8.9 ADPREV

Name: ADPREV

Address: 0x09B

ADC Previous Result Register

| | | | | | | | | |
|--------|------------|----|----|----|----|----|----|----|
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| | PREV[15:8] | | | | | | | |
| Access | RO | RO | RO | RO | RO | RO | RO | RO |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | PREV[7:0] | | | | | | | |
| Access | RO | RO | RO | RO | RO | RO | RO | RO |
| Reset | x | x | x | x | x | x | x | x |

Bits 15:0 – PREV[15:0] Previous ADC Result

| Value | Condition | Description |
|-------|-----------|---|
| n | PSIS = 1 | n = ADFLTR value at the start of current ADC conversion |
| n | PSIS = 0 | n = ADRES at the start of current ADC conversion ⁽¹⁾ |

Note:

1. If PSIS = 0, PREVH and PREVL are formatted the same way as ADRES is, depending on the FRM bit.
2. The individual bytes in this multi-byte register can be accessed with the following register names:
 - ADPREVH: Accesses ADPREV[15:8]
 - ADPREVL: Accesses ADPREV[7:0]

32.8.10 ADRES

Name: ADRES
Address: 0x09D

ADC Result Register

| | | | | | | | | |
|--------|-----------|-----|-----|-----|-----|-----|-----|-----|
| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| | RES[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | x | x | x | x | x | x | x | x |
| | | | | | | | | |
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | RES[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | x | x | x | x | x | x | x | x |

Bits 15:0 – RES[15:0] ADC Sample Result

Reset States: POR/BOR = xxxxxxxxxxxxxxxx

All Other Resets = uuuuuuuuuuuuuuuuu

| Value | Condition | Description |
|------------------------|----------------|----------------------|
| 0x0FFF to 0x0000 | FRM = 1 | Right-aligned result |
| 0xFFFx to 0x000x | FRM = 0 | Left-aligned result |

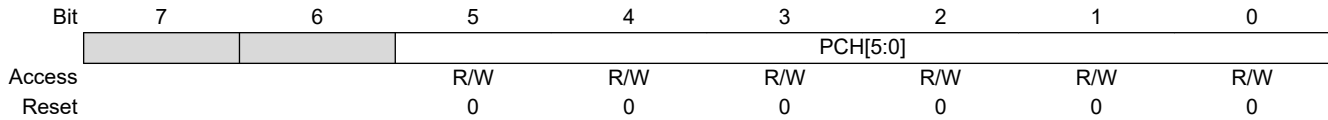
Note: The individual bytes in this multi-byte register can be accessed with the following register names:

- ADRESH: Accesses the high byte ADRES[15:8]
- ADRESL: Accesses the low byte ADRES[7:0]

32.8.11 ADPCH

Name: ADPCH
Address: 0x09F

ADC Positive Channel Selection Register



Bits 5:0 – PCH[5:0] ADC Positive Input Channel Selection bits

Table 32-4. ADC Positive Input Channel Selections

| PCH | ADC Positive Channel Input | Observations |
|---------------|----------------------------------|-------------------------|
| 111111 | Fixed Voltage Reference (FVR) 2 | |
| 111110 | Fixed Voltage Reference (FVR) 1 | |
| 111101 | DAC1 output | |
| 111100 | Temperature Indicator | |
| 111011 | AV _{SS} (Analog Ground) | |
| 111010-011000 | Reserved. No channel connected. | |
| 010111 | RC7/ANC7 | Only for 20-pin devices |
| 010110 | RC6/ANC6 | Only for 20-pin devices |
| 010101 | RC5/ ANC5 | |
| 010100 | RC4/ ANC4 | |
| 010011 | RC3/ANC3 | |
| 010010 | RC2/ANC2 | |
| 010001 | RC1/ ANC1 | |
| 010000 | RC0/ANC0 | |
| 001111 | RB7/ANB7 | Only for 20-pin devices |
| 001110 | RB6/ANB6 | Only for 20-pin devices |
| 001101 | RB5/ANB5 | Only for 20-pin devices |
| 001100 | RB4/ ANB4 | Only for 20-pin devices |
| 001011-000110 | Reserved. No channel connected. | |
| 000101 | RA5/ANA5 | |
| 000100 | RA4/ANA4 | |
| 000011 | Reserved. No channel connected. | |
| 000010 | RA2/ ANA2 | |
| 000001 | RA1/ ANA1 | |
| 000000 | RA0/ANA0 | |

32.8.12 ADACQ

Name: ADACQ
Address: 0x10C

ADC Acquisition Time Control Register

| | | | | | | | | | |
|--------|-----|-----------|-----|-----|-----|-----|-----|-----|-----|
| | Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| | | ACQ[12:8] | | | | | | | |
| Access | | | | | R/W | R/W | R/W | R/W | R/W |
| Reset | | | | | 0 | 0 | 0 | 0 | 0 |
| | Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | ACQ[7:0] | | | | | | | |
| Access | | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bits 12:0 – ACQ[12:0] Acquisition (charge share time) Select

| Value | Description |
|----------|--|
| 11111111 | Number of ADC clock periods in the acquisition time |
| 11111-00 | |
| 00000000 | |
| 001 | Acquisition time is not included in the data conversion cycle ⁽¹⁾ |
| 00000000 | |
| 00000 | |

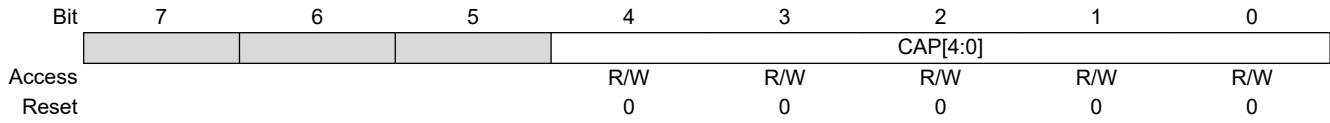
Note:

1. If ADPRE is not equal to '0', then ADACQ = 0b0_0000_0000_0000 means Acquisition time is 8192 clocks of the selected ADC clock.
2. The individual bytes in this multi-byte register can be accessed with the following register names:
 - ADACQH: Accesses the high byte ADACQ[12:8]
 - ADACQL: Accesses the low byte ADACQ[7:0]

32.8.13 ADCAP

Name: ADCAP
Address: 0x10E

ADC Additional Sample Capacitor Selection Register



Bits 4:0 – CAP[4:0] ADC Additional Sample Capacitor Selection bits

| Value | Description |
|----------|--|
| 11111-00 | Number of pF in the additional capacitance |
| 001 | |
| 0 | No additional capacitance |

32.8.14 ADPRE

Name: ADPRE
Address: 0x10F

ADC Precharge Time Control Register

The individual bytes in this multi-byte register can be accessed with the following register names:

- ADPREH: Accesses the high byte PRE[12:8]
- ADPREL: Accesses the low byte PRE[7:0]

| | | | | | | | | |
|--------|-----------|-----|-----|-----|-----|-----|-----|-----|
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| | PRE[12:8] | | | | | | | |
| Access | | | | R/W | R/W | R/W | R/W | R/W |
| Reset | | | | 0 | 0 | 0 | 0 | 0 |
| | | | | | | | | |
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | PRE[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bits 12:0 – PRE[12:0] Precharge Time Select

| Value | Description |
|-------------------|---|
| 0x1FFF-0 x0001 | Number of ADC clocks in the precharge time |
| 0 | Precharge time is not included in the data conversion cycle |

Note: If ADPRE is not equal to '0', then ADACQ = b' 00000000 means Acquisition time is 256 clocks of the selected ADC clock.

32.8.15 ADCON0

Name: ADCON0
Address: 0x111

ADC Control Register 0

| | | | | | | | | |
|--------|-----|------|---|-----|---|-----|---|--------|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | ON | CONT | | CS | | FRM | | GO |
| Access | R/W | R/W | | R/W | | R/W | | R/W/HC |
| Reset | 0 | 0 | | 0 | | 0 | | 0 |

Bit 7 – ON ADC Enable bit

| Value | Description |
|-------|-----------------|
| 1 | ADC is enabled |
| 0 | ADC is disabled |

Bit 6 – CONT ADC Continuous Operation Enable bit

| Value | Description |
|-------|---|
| 1 | GO is retriggered upon completion of each conversion trigger until TIF is set (if SOI is set) or until GO is cleared (regardless of the value of SOI) |
| 0 | GO is cleared upon completion of each conversion trigger |

Bit 4 – CS ADC Clock Selection bit

| Value | Description |
|-------|--|
| 1 | Clock supplied from FRC dedicated oscillator |
| 0 | Clock supplied by F _{OSC} , divided according to ADCLK register |

Bit 2 – FRM ADC Results Format/Alignment Selection

| Value | Description |
|-------|---|
| 1 | ADRES and ADPREV data are right-justified |
| 0 | ADRES and ADPREV data are left-justified, zero-filled |

Bit 0 – GO ADC Conversion Status bit^(1,2)

| Value | Description |
|-------|---|
| 1 | ADC conversion cycle in progress. Setting this bit starts an ADC conversion cycle. The bit is cleared by hardware as determined by the CONT bit |
| 0 | ADC conversion completed/not in progress |

Note:

1. This bit requires ON bit to be set.
2. If cleared by software while a conversion is in progress, the results of the conversion up to this point will be transferred to ADRES and the state machine will be reset, but the ADIF interrupt flag bit will not be set; filter and threshold operations will not be performed.

32.8.16 ADCON1

Name: ADCON1
Address: 0x112

ADC Control Register 1

| | | | | | | | | |
|--------|------|------|------|---|---|---|---|------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | PPOL | IPEN | GPOL | | | | | DSEN |
| Access | R/W | R/W | R/W | | | | | R/W |
| Reset | 0 | 0 | 0 | | | | | 0 |

Bit 7 – PPOL Precharge Polarity bit
 Action During 1st Precharge Stage

| Value | Condition | Description |
|-------|-------------------------------|---|
| x | PRE=0 | Bit has no effect |
| 1 | PRE>0 & ADC input is I/O pin | Pin shorted to AV _{DD} |
| 0 | PRE>0 & ADC input is I/O pin | Pin shorted to V _{SS} |
| 1 | PRE>0 & ADC input is internal | C _{HOLD} Shorted to AV _{DD} |
| 0 | PRE>0 & ADC input is internal | C _{HOLD} Shorted to V _{SS} |

Bit 6 – IPEN A/D Inverted Precharge Enable bit

| Value | Condition | Description |
|-------|-----------|---|
| x | DSEN = 0 | Bit has no effect |
| 1 | DSEN = 1 | The precharge and guard signals in the second conversion cycle are the opposite polarity of the first cycle |
| 0 | DSEN = 1 | Both Conversion cycles use the precharge and guards specified by PPOL and GPOL |

Bit 5 – GPOL Guard Ring Polarity Selection bit

| Value | Description |
|-------|---|
| 1 | ADC guard Ring outputs start as digital high during Precharge stage |
| 0 | ADC guard Ring outputs start as digital low during Precharge stage |

Bit 0 – DSEN Double-Sample Enable bit

| Value | Description |
|-------|---|
| 1 | Two conversions are performed on each trigger. Data from the first conversion appears in PREV |
| 0 | One conversion is performed for each trigger |

32.8.17 ADCON2

Name: ADCON2
Address: 0x113

ADC Control Register 2

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|------|----------|-----|-----|--------|---------|-----|-----|
| | PSIS | CRS[2:0] | | | ACLR | MD[2:0] | | |
| Access | R/W | R/W | R/W | R/W | R/W/HC | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bit 7 – PSIS ADC Previous Sample Input Select bits

| Value | Description |
|-------|---|
| 1 | FLTR is transferred to PREV at start-of-conversion |
| 0 | ADRES is transferred to PREV at start-of-conversion |

Bits 6:4 – CRS[2:0] ADC Accumulated Calculation Right Shift Select bits

| Value | Condition | Description |
|---------|-----------------------|--|
| 111-000 | MD = b'100' | Low-pass filter time constant is 2^{CRS} , filter gain is 1:1 |
| 111-000 | MD = b'011' to b'001' | The accumulated value is right-shifted by CRS (divided by 2^{CRS})(1,2) |
| xxx | MD = b'000' to b'001' | These bits are ignored |

Bit 3 – ACLR A/D Accumulator Clear Command bit⁽³⁾

| Value | Description |
|-------|--|
| 1 | ACC, OV and CNT bits are cleared |
| 0 | Clearing action is complete (or not started) |

Bits 2:0 – MD[2:0] ADC Operating Mode Selection bits⁽⁴⁾

| Value | Description |
|---------|----------------------|
| 111-101 | Reserved |
| 100 | Low-pass Filter mode |
| 011 | Burst Average mode |
| 010 | Average mode |
| 001 | Accumulate mode |
| 000 | Basic (Legacy) mode |

Note:

- To correctly calculate an average, the number of samples (set in RPT) must be 2^{CRS} .
- CRS = 3'b111 is a reserved option.
- This bit is cleared by hardware when the accumulator operation is complete; depending on oscillator selections, the delay may be many instructions.
- See [Computation Modes](#) for full mode descriptions.

32.8.18 ADCON3

Name: ADCON3
Address: 0x114

ADC Control Register 3

| | | | | | | | | |
|--------|-----------|-----|-----|-----|----------|-----|-----|-----|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | CALC[2:0] | | SOI | | TMD[2:0] | | | |
| Access | | R/W | R/W | R/W | R/W/HC | R/W | R/W | R/W |
| Reset | | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bits 6:4 – CALC[2:0] ADC Error Calculation Mode Select bits
See table below for selection details.

Table 32-5. ADC Error Calculation Mode

| CALC | Action During 1st Precharge Stage | | Application |
|------|-----------------------------------|--|---|
| | DSEN = 0 Single-Sample Mode | DSEN = 1 CVD Double-Sample Mode ⁽¹⁾ | |
| 000 | ADRES-ADPREV | ADRES-ADPREV | First derivative of single measurement ⁽²⁾ Actual CVD result in CVD mode ⁽²⁾ |
| 001 | ADRES-ADSTPT | (ADRES-ADPREV)-ADSTPT | Actual result vs. setpoint |
| 010 | ADRES-ADFLTR | (ADRES-ADPREV)-ADFLTR | Actual result vs. averaged/filtered value |
| 011 | Reserved | Reserved | Reserved |
| 100 | ADPREV-ADFLTR | ADPREV-ADFLTR | First derivative of filtered value ⁽³⁾ (negative) |
| 101 | ADFLTR-ADSTPT | ADFLTR-ADSTPT | Average/filtered value vs. setpoint |
| 110 | Reserved | Reserved | Reserved |
| 111 | Reserved | Reserved | Reserved |

Note:

1. When **PSIS** = 0, the value of ADRES-ADPREV is the value of (S2-S1) from [Computation Modes](#).
2. When **PSIS** = 0
3. When **PSIS** = 1.

Bit 3 – SOI ADC Stop-on-Interrupt bit

| Value | Condition | Description |
|-------|-----------------|--|
| 1 | CONT = 1 | GO is cleared when the threshold conditions are met, otherwise the conversion is retrIGGERED |
| 0 | CONT = 1 | GO is not cleared by hardware, must be cleared by software to stop retriggers |
| x | CONT = 0 | This bit is not used |

Bits 2:0 – TMD[2:0] Threshold Interrupt Mode Select bits

| Value | Description |
|-------|--|
| 111 | Interrupt regardless of threshold test results |
| 110 | Interrupt if ERR>UTH |
| 101 | Interrupt if ERR≤UTH |
| 100 | Interrupt if ERR<LTH or ERR>UTH |
| 011 | Interrupt if ERR>LTH and ERR<UTH |
| 010 | Interrupt if ERR≥LTH |

PIC16(L)F18426/46

ADC2 - Analog-to-Digital Converter

| Value | Description |
|-------|----------------------|
| 001 | Interrupt if ERR<LTH |
| 000 | Never interrupt |

32.8.19 ADSTAT

Name: ADSTAT
Address: 0x115

ADC Status Register

| | | | | | | | | |
|--------|----|------|------|---------|---|----|-----------|----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | OV | UTHR | LTHR | MATH | | | STAT[2:0] | |
| Access | RO | RO | RO | R/HS/HC | | RO | RO | RO |
| Reset | 0 | 0 | 0 | 0 | | 0 | 0 | 0 |

Bit 7 – OV ADC Accumulator Overflow bit

| Value | Description |
|-------|---|
| 1 | ADC accumulator or ERR calculation have overflowed |
| 0 | ADC accumulator and ERR calculation have not overflowed |

Bit 6 – UTHR ADC Module Greater-than Upper Threshold Flag bit

| Value | Description |
|-------|-------------|
| 1 | ERR > UTH |
| 0 | ERR ≤ UTH |

Bit 5 – LTHR ADC Module Less-than Lower Threshold Flag bit

| Value | Description |
|-------|-------------|
| 1 | ERR < LTH |
| 0 | ERR ≥ LTH |

Bit 4 – MATH ADC Module Computation Status bit

| Value | Description |
|-------|---|
| 1 | Registers ADACC, ADFLTR, ADUTH, ADLTH and the OV bit are updating or have already updated |
| 0 | Associated registers/bits have not changed since this bit was last cleared |

Bits 2:0 – STAT[2:0] ADC Module Cycle Multistage Status bits⁽¹⁾

| Value | Description |
|-------|--|
| 111 | ADC module is in 2 nd conversion stage |
| 110 | ADC module is in 2 nd acquisition stage |
| 101 | ADC module is in 2 nd precharge stage |
| 100 | Not used |
| 011 | ADC module is in 1 st conversion stage |
| 010 | ADC module is in 1 st acquisition stage |
| 001 | ADC module is in 1 st precharge stage |
| 000 | ADC module is not converting |

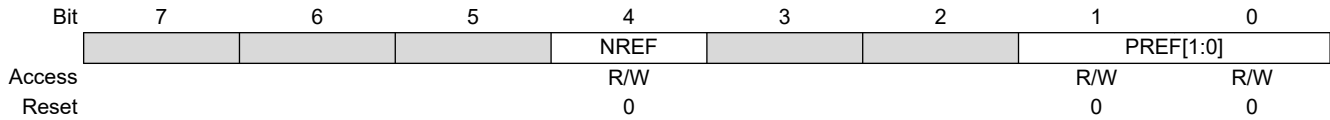
Note:

1. If CS = 1, and F_{OSC} < FRC, these bits may be invalid.

32.8.20 ADREF

Name: ADREF
Address: 0x116

ADC Reference Selection Register



Bit 4 – NREF ADC Negative Voltage Reference Selection bit

| Value | Description |
|-------|--|
| 1 | V _{REF-} is connected to external V _{REF-} |
| 0 | V _{REF-} is connected to AV _{SS} |

Bits 1:0 – PREF[1:0] ADC Positive Voltage Reference Selection bits

| Value | Description |
|-------|---|
| 11 | V _{REF+} is connected to internal Fixed Voltage Reference (FVR) module |
| 10 | V _{REF+} is connected to external V _{REF+} |
| 01 | Reserved |
| 00 | V _{REF+} is connected to V _{DD} |

32.8.21 ADACT

Name: ADACT
Address: 0x117

ADC AUTO Conversion Trigger Source Selection Register

| | | | | | | | | |
|--------|---|---|---|-----|----------|-----|-----|-----|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | | | | ACT[4:0] | | | |
| Access | | | | R/W | R/W | R/W | R/W | R/W |
| Reset | | | | 0 | 0 | 0 | 0 | 0 |

Bits 4:0 – ACT[4:0] Auto-Conversion Trigger Select Bits

Table 32-6. ADC Auto-Conversion Trigger Sources

| ACT | Auto-conversion Trigger Source |
|----------------|---|
| 11111 | Software write to ADPCH |
| 11110 | Reserved, do not use |
| 11101 | Software read of ADRESH |
| 11100 | Software read of ADERRH |
| 11011 to 11000 | Reserved, do not use |
| 10111 | CLC4_out |
| 10110 | CLC3_out |
| 10101 | CLC2_out |
| 10100 | CLC1_out |
| 10011 | Logical OR of all Interrupt-on-change Interrupt Flags |
| 10010 | C2_out |
| 10001 | C1_out |
| 10000 | NCO1_out |
| 01111 | PWM7_out |
| 01110 | PWM6_out |
| 01101 | CCP4_trigger |
| 01100 | CCP3_trigger |
| 01011 | CCP2_trigger |
| 01010 | CCP1_trigger |
| 01001 | SMT1_trigger |
| 01000 | TMR6_postscaled |
| 00111 | TMR5_overflow |
| 00110 | TMR4_postscaled |
| 00101 | TMR3_overflow |
| 00100 | TMR2_postscaled |
| 00011 | TMR1_overflow |

PIC16(L)F18426/46
ADC2 - Analog-to-Digital Converter

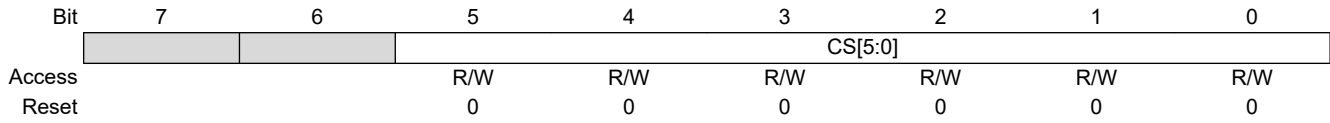
.....continued

| ACT | Auto-conversion Trigger Source |
|------------|---------------------------------------|
| 00010 | TMR0_overflow |
| 00001 | Pin selected by ADACTPPS |
| 00000 | External Trigger Disabled |

32.8.22 ADCLK

Name: ADCLK
Address: 0x118

ADC Clock Selection Register



Bits 5:0 – CS[5:0] ADC Conversion Clock Select bits

| Value | Description |
|---------|--|
| 11 1111 | ADC Clock frequency = $F_{OSC}/(2*(CS+1))$ |
| - 00 | |
| 0000 | |

32.8.23 ADCPCON0

Name: ADCPCON0

Address: 0x29F

ADC Charge Pump Control Register 0

| | | | | | | | | |
|--------|------|---|---|---|---|---|---|-------|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | CPON | | | | | | | CPRDY |
| Access | R/W | | | | | | | RO |
| Reset | 0 | | | | | | | 0 |

Bit 7 – CPON Charge Pump On Control bit

| Value | Description |
|-------|--|
| 1 | Charge Pump On when requested by the ADC |
| 0 | Charge Pump Off |

Bit 0 – CPRDY Charge Pump Ready Status bit

| Value | Description |
|-------|---|
| 1 | Charge Pump is ready |
| 0 | Charge Pump is not ready (or never started) |

33. DAC - 5-Bit Digital-to-Analog Converter

The Digital-to-Analog Converter supplies a variable voltage reference, ratiometric with the input source, with 32 selectable output levels.

The positive input source ($V_{SOURCE+}$) of the DAC can be connected to:

- FVR Buffer
- External V_{REF+} pin
- V_{DD} supply voltage

The negative input source ($V_{SOURCE-}$) of the DAC can be connected to:

- External V_{REF-} pin
- V_{SS}

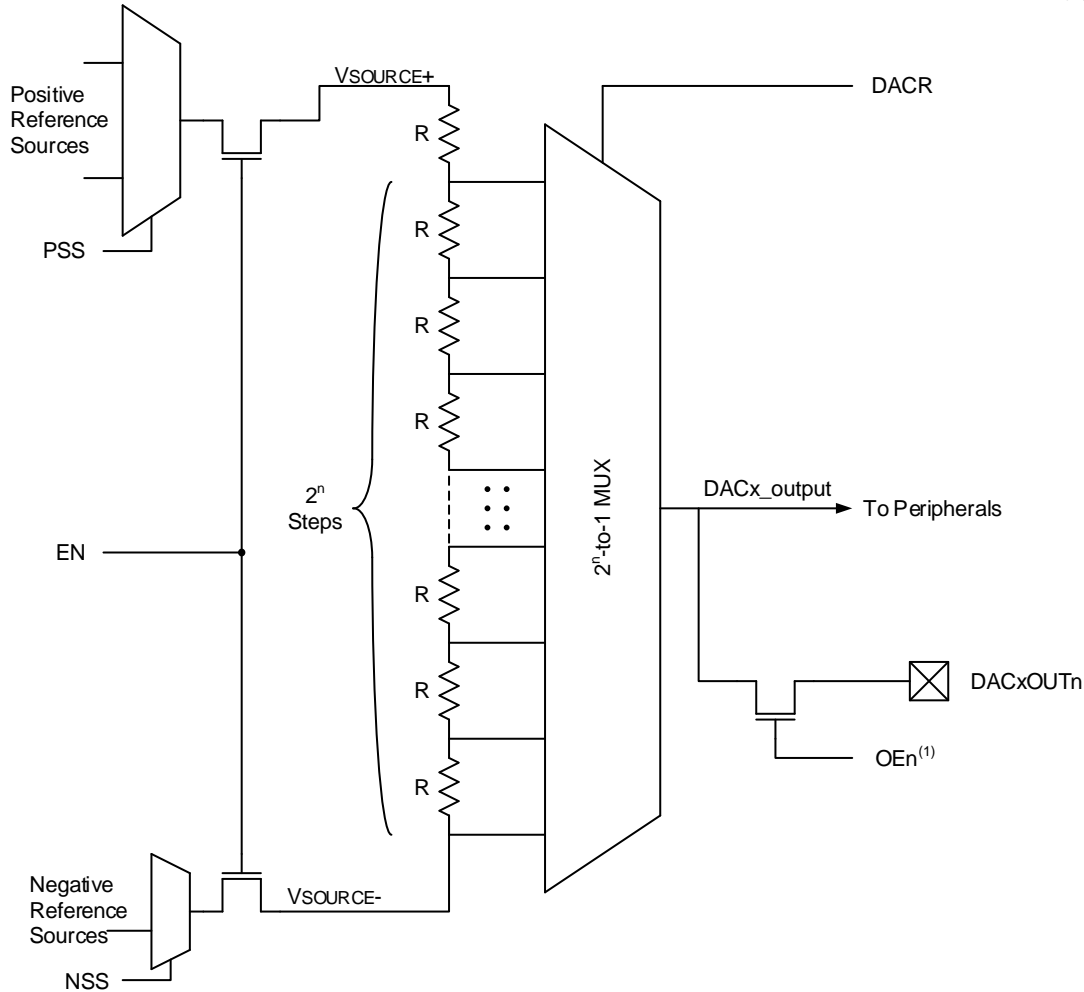
The output of the DAC (DACx_output) can be selected as a reference voltage to the following:

- Comparator positive input
- ADC input channel
- DACxOUT1 pin

The Digital-to-Analog Converter (DAC) can be enabled by setting the [EN](#) bit.

Figure 33-1. Digital-to-Analog Converter Block Diagram

Rev. 10-000 026J
2/26/2019



Note 1: The output enable bits are configured so that they act as a “one-hot” system, meaning only one DAC output can be enabled at a time.

33.1 Output Voltage Selection

The DAC has 32 voltage level ranges. The 32 levels are set with the [DAC1R](#) bits.

The DAC output voltage can be determined by using the following equation.

Equation 33-1. DAC Output Voltage

$$DACx_output = \left(\left(V_{SOURCE+} - V_{SOURCE-} \right) \times \frac{DACR[4:0]}{32} \right) + V_{SOURCE-}$$

Note: See the [DAC1CON0](#) register for the available $V_{SOURCE+}$ and $V_{SOURCE-}$ selections.

33.2 Ratiometric Output Level

The DAC output value is derived using a resistor ladder with each end of the ladder tied to a positive and negative voltage reference input source. If the voltage of either input source fluctuates, a similar fluctuation will result in the DAC output value.

The value of the individual resistors within the ladder can be found in the “5-Bit DAC Specifications” table from the “Electrical Specifications” chapter.

Related Links

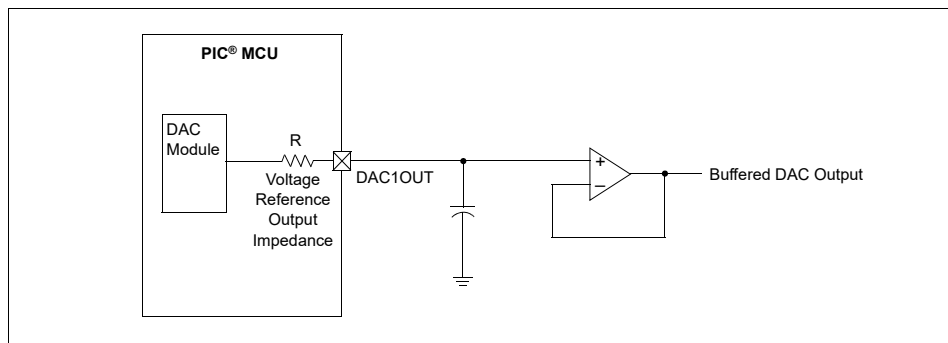
[40.4.10 5-Bit DAC Specifications](#)

33.3 DAC Voltage Reference Output

The DAC voltage can be output to the DAC1OUT pin by setting the DAC1OE bit of the DAC1CON0 register. Selecting the DAC reference voltage for output on the DAC1OUT pin automatically overrides the digital output buffer and digital input threshold detector functions, it disables the weak pull-up and the constant-current drive function of that pin. Reading the DAC1OUT pin when it has been configured for DAC reference voltage output will always return a ‘0’.

Due to the limited current drive capability, a buffer must be used on the DAC voltage reference output for external connections to the DAC1OUT pin. [Figure 33-2](#) below shows an example buffering technique.

Figure 33-2. Voltage Reference Output Buffer Example



33.4 Operation During Sleep

The DAC continues to function during Sleep. When the device wakes up from Sleep through an interrupt or a Watchdog Timer time-out, the contents of the DAC1CON0 register are not affected.

33.5 Effects of a Reset

A device Reset affects the following:

- DACx is disabled.
- DACx output voltage is removed from the DACxOUTn pin(s).
- The [DAC1R](#) range select bits are cleared.

33.6 Register Summary - DAC Control

| Address | Name | Bit Pos. | | | | | | | |
|-------------|----------|----------|----|--|-----|--|------------|--|-----|
| 0x00 ... | Reserved | | | | | | | | |
| 0x090D | | | | | | | | | |
| 0x090E | DAC1CON0 | 7:0 | EN | | OE1 | | PSS[1:0] | | NSS |
| 0x090F | DAC1CON1 | 7:0 | | | | | DAC1R[4:0] | | |

33.7 Register Definitions: DAC Control

33.7.1 DAC1CON0

Name: DAC1CON0

Address: 0x90E

DAC Control Register

| | | | | | | | | | |
|--------|-----|-----|---|-----|---|----------|-----|---|-----|
| | Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | EN | | OE1 | | PSS[1:0] | | | NSS |
| Access | | R/W | | R/W | | R/W | R/W | | R/W |
| Reset | | 0 | | 0 | | 0 | 0 | | 0 |

Bit 7 – EN DAC Enable bit

| Value | Description |
|-------|-----------------|
| 1 | DAC is enabled |
| 0 | DAC is disabled |

Bit 5 – OE1 DAC Voltage Output Enable bit

| Value | Description |
|-------|---|
| 1 | DAC voltage level is output on the DAC1OUT1 pin |
| 0 | DAC voltage level is disconnected from the DAC1OUT1 pin |

Bits 3:2 – PSS[1:0] DAC Positive Source Select bit

| Value | Description |
|-------|-------------------|
| 11 | Reserved |
| 10 | FVR buffer |
| 01 | V _{REF+} |
| 00 | AV _{DD} |

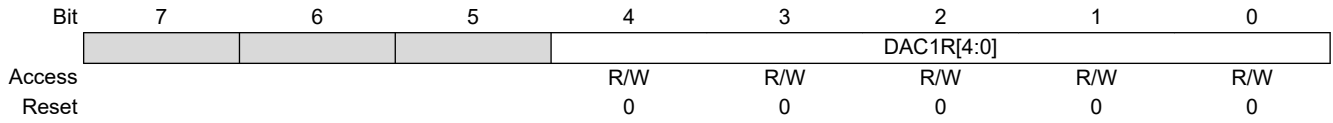
Bit 0 – NSS DAC Negative Source Select bit

| Value | Description |
|-------|-------------------|
| 1 | V _{REF-} |
| 0 | AV _{SS} |

33.7.2 DAC1CON1

Name: DAC1CON1
Address: 0x90F

DAC Data Register



Bits 4:0 – DAC1R[4:0] Data Input Register for DAC bits

34. CMP - Comparator Module

Comparators are used to interface analog circuits to a digital circuit by comparing two analog voltages and providing a digital indication of their relative magnitudes. Comparators are very useful mixed-signal building blocks because they provide analog functionality independent of program execution. The PIC16(L)F18426/46 devices have 2 comparators (C1/C2).

The analog comparator module includes the following features:

- Programmable Input Selection
- Programmable Output Polarity
- Rising/Falling Output Edge Interrupts
- Wake-up from Sleep
- CWG Auto-Shutdown Source
- Selectable Voltage Reference
- ADC Auto-Trigger
- Odd Numbered Timers (Timer1, Timer3, etc.) Gate
- Even Numbered Timers (Timer2, Timer4, etc.) Reset
- CCP Capture Mode Input
- DSM Modulator Source
- Input and Window Signal-to-Signal Measurement Timer

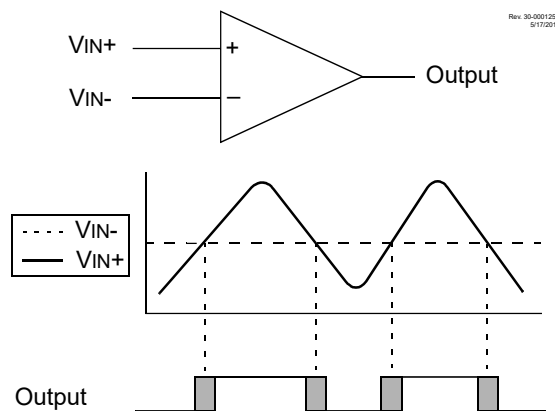
34.1 Comparator Overview

A single comparator is shown in Figure 34-1 along with the relationship between the analog input levels and the digital output. When the analog voltage at V_{IN+} is less than the analog voltage at V_{IN-} , the output of the comparator is a digital low level. When the analog voltage at V_{IN+} is greater than the analog voltage at V_{IN-} , the output of the comparator is a digital high level.

Table 34-1. Available Comparators

| Device | C1 | C2 |
|----------------|----|----|
| PIC16(L)F18426 | • | • |
| PIC16(L)F18446 | • | • |

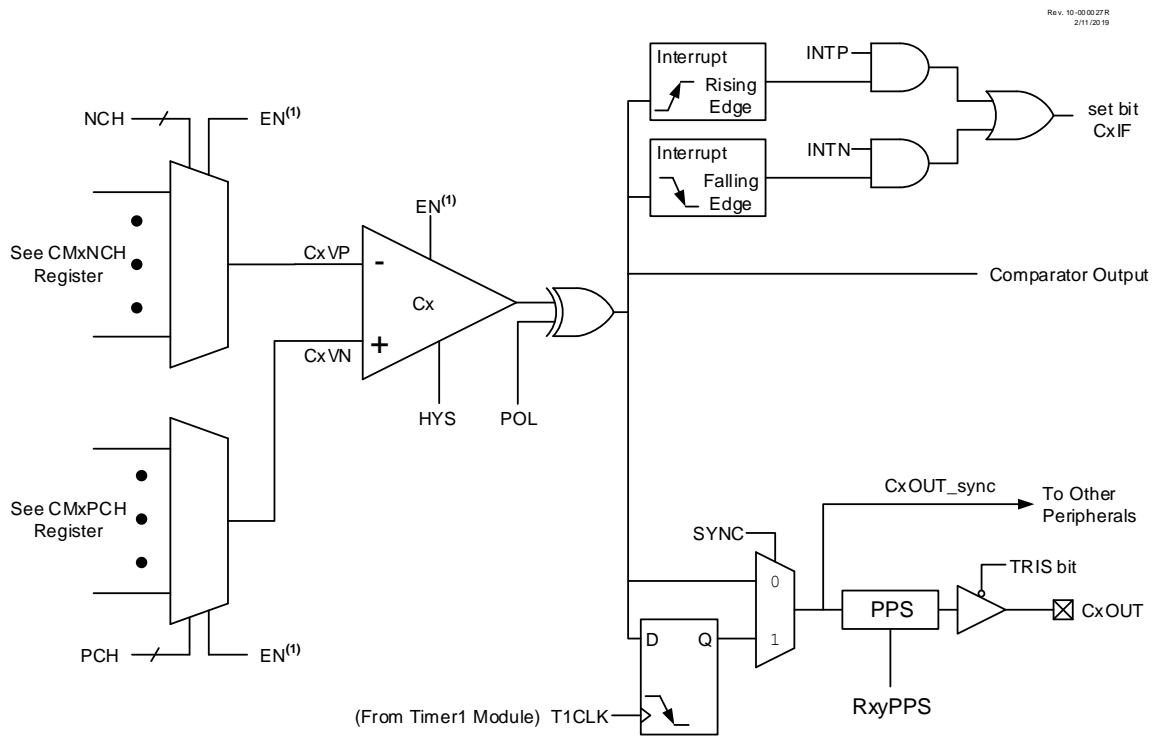
Figure 34-1. Single Comparator



Note:

1. The black areas of the output of the comparator represent the uncertainty due to input offsets and response time.

Figure 34-2. Comparator Module Simplified Block Diagram



Note 1: When EN = 0, all multiplexer inputs are disconnected and the Comparator will produce a '0' at the output.

Related Links

[34.15.4 CMxNCH](#)

[34.15.5 CMxPCH](#)

34.2 Comparator Control

Each comparator has two control registers: CMxCON0 and CMxCON1.

The [CMxCON0](#) register contains Control and Status bits for the following:

- Enable
- Output
- Output Polarity
- Hysteresis Enable
- Timer1 Output Synchronization

The [CMxCON1](#) register contains Control bits for the following:

- Interrupt on Positive/Negative Edge Enables
- Positive Input Channel Selection
- Negative Input Channel Selection

34.2.1 Comparator Enable

Setting the **EN** bit enables the comparator for operation. Clearing the CxEN bit disables the comparator, resulting in minimum current consumption.

34.2.2 Comparator Output

The output of the comparator can be monitored by reading either the [CxOUT](#) bit or the [MCxOUT](#) bit.

The comparator output can also be routed to an external pin through the RxyPPS register. The corresponding TRIS bit must be clear to enable the pin as an output.

Note:

1. The internal output of the comparator is latched with each instruction cycle. Unless otherwise specified, external outputs are not latched.

Related Links

[16.9.3 RxyPPS](#)

34.2.3 Comparator Output Polarity

Inverting the output of the comparator is functionally equivalent to swapping the comparator inputs. The polarity of the comparator output can be inverted by setting the [CxPOL](#) bit. Clearing the CxPOL bit results in a non-inverted output.

[Table 34-2](#) shows the output state versus input conditions, including polarity control.

Table 34-2. Comparator Output State vs. Input Conditions

| Input Condition | CxPOL | CxOUT |
|-----------------|-------|-------|
| $CxVn > CxVp$ | 0 | 0 |
| $CxVn < CxVp$ | 0 | 1 |
| $CxVn > CxVp$ | 1 | 1 |
| $CxVn < CxVp$ | 1 | 0 |

34.3 Comparator Hysteresis

A selectable amount of separation voltage can be added to the input pins of each comparator to provide a hysteresis function to the overall operation. Hysteresis is enabled by setting the [CxHYS](#) bit.

See Comparator Specifications for more information.

Related Links

[40.4.9 Comparator Specifications](#)

34.4 Operation With Timer1 Gate

The output resulting from a comparator operation can be used as a source for gate control of the odd numbered timers (Timer1, Timer3, etc.). See [19.7 Timer1 Gate](#) section for more information. This feature is useful for timing the duration or interval of an analog event.

It is recommended that the comparator output be synchronized to the timer by setting the [SYNC](#) bit in the CMxCON0 register. This ensures that the timer does not increment while a change in the comparator is occurring. However, synchronization is only possible with the Timer1 clock source. Synchronization with the other odd numbered timers is only possible when they use the same clock source as Timer1.

Related Links

[19.7 Timer1 Gate](#)

34.4.1 Comparator Output Synchronization

The output from a comparator can be synchronized with Timer1 by setting the [SYNC](#) bit.

Once enabled, the comparator output is latched on the falling edge of the Timer1 source clock. If a prescaler is used with Timer1, the comparator output is latched after the prescaling function. To prevent a race condition, the comparator output is latched on the falling edge of the Timer1 clock source and Timer1 increments on the rising edge

of its clock source. See the [Figure 34-2](#) Comparator Block Diagram and the Timer1 Block Diagram for more information.

Related Links

[19. TMR1 - Timer1 Module with Gate Control](#)

34.5 Comparator Interrupt

An interrupt can be generated when either the rising edge or falling edge detector detects a change in the output value of each comparator.

When either edge detector is triggered and its associated enable bit is set ([CxINTP](#) and/or [CxINTN](#) bits), the Corresponding Interrupt Flag bit (CxIF bit of the PIR2 register) will be set.

To enable the interrupt, the following bits must be set:

- [EN](#) and [POL](#) bits
- CxIE bit of the PIE2 register
- [INTP](#) bit (for a rising edge detection)
- [INTN](#) bit (for a falling edge detection)
- PEIE and GIE bits of the INTCON register

The associated Interrupt Flag bit, CxIF bit of the PIR2 register, must be cleared in software. If another edge is detected while this flag is being cleared, the flag will still be set at the end of the sequence.



Important: Although a comparator is disabled, an interrupt can be generated by changing the output polarity with the [CxPOL](#) bit, or by switching the comparator on or off with the [CxEN](#) bit.

34.6 Comparator Positive Input Selection

Configuring the [PCH](#) bits in the CMxPCH register direct an internal voltage reference or an analog pin to the non-inverting input of the comparator:

- CxINy+ analog pin
- DAC output
- Fixed Voltage Reference (FVR)



Important: To use CxINy+ pins as analog input, the appropriate bits must be set in the ANSEL register and the corresponding TRIS bits must also be set to disable the output drivers.

Any time the comparator is disabled ($CxEN = 0$), all comparator inputs are disabled.

Related Links

[30. FVR - Fixed Voltage Reference](#)

[33. DAC - 5-Bit Digital-to-Analog Converter](#)

34.7 Comparator Negative Input Selection

The [NCH](#) bits in the CMxNCH register direct an analog input pin and internal reference voltage or analog ground to the inverting input of the comparator:

- CxINy- analog pin
- Fixed Voltage Reference (FVR)



Important: To use CxINy- pins as analog input, the appropriate bits must be set in the ANSEL register and the corresponding TRIS bits must also be set to disable the output drivers.

34.8 Comparator Response Time

The comparator output is indeterminate for a period of time after the change of an input source or the selection of a new reference voltage. This period is referred to as the response time. The response time of the comparator differs from the settling time of the voltage reference. Therefore, both of these times must be considered when determining the total response time to a comparator input change. See the Comparator and Voltage Reference Specifications in Comparator Specifications and Fixed Voltage Reference (FVR) Specifications for more details.

Related Links

[40.4.9 Comparator Specifications](#)

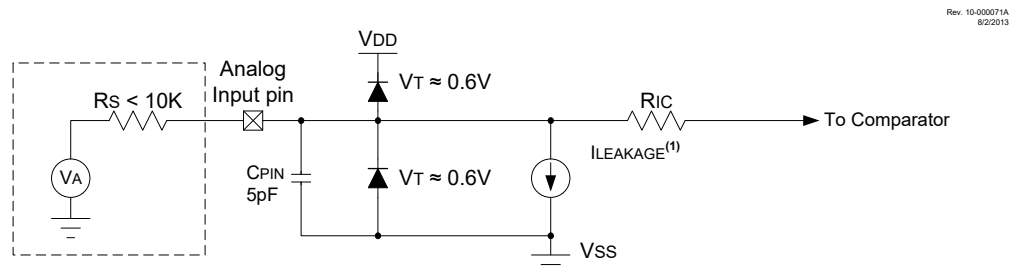
[40.4.11 Fixed Voltage Reference \(FVR\) Specifications](#)

34.9 Analog Input Connection Considerations

A simplified circuit for an analog input is shown in [Figure 34-3](#). Since the analog input pins share their connection with a digital input, they have reverse biased ESD protection diodes to V_{DD} and V_{SS} . The analog input, therefore, must be between V_{SS} and V_{DD} . 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 k Ω 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.

Figure 34-3. 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

Related Links

[40. Electrical Specifications](#)

34.10 CWG1 Auto-Shutdown Source

The output of the comparator module can be used as an auto-shutdown source for the CWG1 module. When the output of the comparator is active and the corresponding WGASxE is enabled, the CWG operation will be suspended immediately.

Related Links

[25.11.1.2 External Input Source](#)

34.11 ADC Auto-Trigger Source

The output of the comparator module can be used to trigger an ADC conversion. When the ADOACT register is set to trigger on a comparator output, an ADC conversion will trigger when the comparator output goes high.

34.12 Even Numbered Timers Reset

The output of the comparator module can be used to reset the even numbered timers (Timer2, Timer4, etc.). When the TxERS register is appropriately set, the timer will reset when the comparator output goes high.

34.13 Operation in Sleep Mode

The comparator module can operate during Sleep. The comparator clock source is based on the Timer1 clock source. If the Timer1 clock source is either the system clock (F_{OSC}) or the instruction clock ($F_{OSC}/4$), Timer1 will not operate during Sleep, and synchronized comparator outputs will not operate.

A comparator interrupt will wake the device from Sleep. The CxIE bits of the PIE2 register must be set to enable comparator interrupts.

34.14 Register Summary - Comparator

| Address | Name | Bit Pos. | | | | | | | |
|-------------|----------|----------|----|-----|--|-----|--|----------|--------|
| 0x00 ... | Reserved | | | | | | | | |
| 0x098E | | | | | | | | | |
| 0x098F | CMOUT | 7:0 | | | | | | MC2OUT | MC1OUT |
| 0x0990 | CM1CON0 | 7:0 | EN | OUT | | POL | | HYS | SYNC |
| 0x0991 | CM1CON1 | 7:0 | | | | | | INTP | INTN |
| 0x0992 | CM1NCH | 7:0 | | | | | | NCH[2:0] | |
| 0x0993 | CM1PCH | 7:0 | | | | | | PCH[2:0] | |
| 0x0994 | CM2CON0 | 7:0 | EN | OUT | | POL | | HYS | SYNC |
| 0x0995 | CM2CON1 | 7:0 | | | | | | INTP | INTN |
| 0x0996 | CM2NCH | 7:0 | | | | | | NCH[2:0] | |
| 0x0997 | CM2PCH | 7:0 | | | | | | PCH[2:0] | |

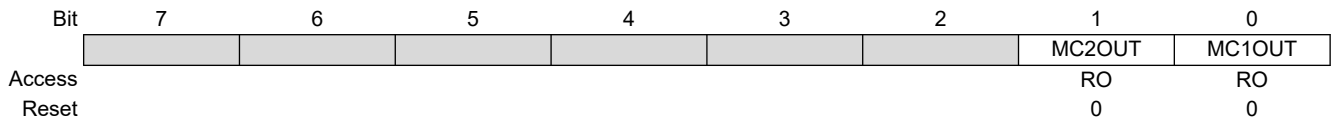
34.15 Register Definitions: Comparator Control

34.15.1 CMOUT

Name: CMOUT

Address: 0x98F

Comparator Output Register



Bits 0, 1 – MCxOUT Mirror copy of CxOUT bit

34.15.2 CMxCON0

Name: CMxCON0
Address: 0x990,0x994

Comparator x Control Register 0

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|-----|-----|---|-----|---|---|-----|------|
| | EN | OUT | | POL | | | HYS | SYNC |
| Access | R/W | RO | | R/W | | | R/W | R/W |
| Reset | 0 | 0 | | 0 | | | 0 | 0 |

Bit 7 – EN Comparator Enable bit

| Value | Description |
|-------|---|
| 1 | Comparator is enabled |
| 0 | Comparator is disabled and consumes no active power |

Bit 6 – OUT Comparator Output bit

| Value | Condition | Description |
|-------|--|-------------|
| 1 | If POL = 0 (non-inverted polarity): | CxVP > CxVN |
| 0 | If POL = 0 (non-inverted polarity): | CxVP < CxVN |
| 1 | If POL = 1 (inverted polarity): | CxVP < CxVN |
| 0 | If POL = 1 (inverted polarity): | CxVP > CxVN |

Bit 4 – POL Comparator Output Polarity Select bit

| Value | Description |
|-------|-----------------------------------|
| 1 | Comparator output is inverted |
| 0 | Comparator output is not inverted |

Bit 1 – HYS Comparator Hysteresis Enable bit

| Value | Description |
|-------|--------------------------------|
| 1 | Comparator hysteresis enabled |
| 0 | Comparator hysteresis disabled |

Bit 0 – SYNC Comparator Output Synchronous Mode bit

Output updated on the falling edge of prescaled Timer1 clock.

| Value | Description |
|-------|--|
| 1 | Comparator output to Timer1 and I/O pin is synchronous to changes on the prescaled Timer1 clock. |
| 0 | Comparator output to Timer1 and I/O pin is asynchronous |

34.15.3 CMxCON1

Name: CMxCON1
Address: 0x991,0x995

Comparator x Control Register 1

| | | | | | | | | |
|--------|---|---|---|---|---|---|------|------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | | | | | | INTP | INTN |
| Access | | | | | | | R/W | R/W |
| Reset | | | | | | | 0 | 0 |

Bit 1 – INTP Comparator Interrupt on Positive-Going Edge Enable bit

| Value | Description |
|-------|---|
| 1 | The CxIF interrupt flag will be set upon a positive-going edge of the CxOUT bit |
| 0 | No interrupt flag will be set on a positive-going edge of the CxOUT bit |

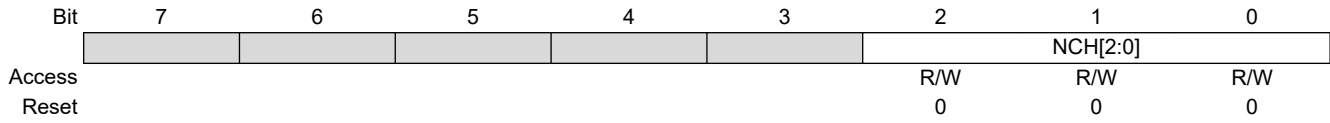
Bit 0 – INTN Comparator Interrupt on Negative-Going Edge Enable bit

| Value | Description |
|-------|---|
| 1 | The CxIF interrupt flag will be set upon a negative-going edge of the CxOUT bit |
| 0 | No interrupt flag will be set on a negative-going edge of the CxOUT bit |

34.15.4 CMxNCH

Name: CMxNCH
Address: 0x992,0x996

Comparator x Inverting Channel Select Register



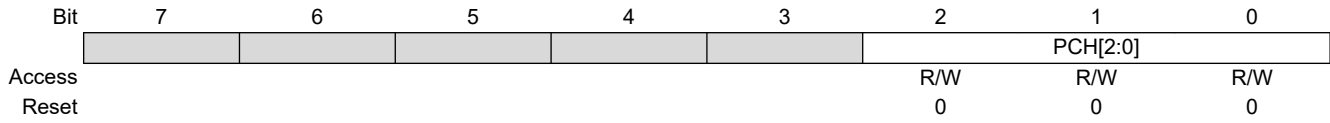
Bits 2:0 – NCH[2:0] Comparator Inverting Input Channel Select bits

| NCH | Negative Input Sources |
|-----|---|
| 111 | CxV _N connects to AV _{SS} |
| 110 | CxV _N connects to FVR Buffer 2 |
| 101 | CxV _N not connected |
| 100 | CxV _N not connected |
| 011 | CxV _N connects to CxIN3- pin |
| 010 | CxV _N connects to CxIN2- pin |
| 001 | CxV _N connects to CxIN1- pin |
| 000 | CxV _N connects to CxIN0- pin |

34.15.5 CMxPCH

Name: CMxPCH
Address: 0x993,0x997

Comparator x Non-Inverting Channel Select Register



Bits 2:0 – PCH[2:0] Comparator Non-Inverting Input Channel Select bits

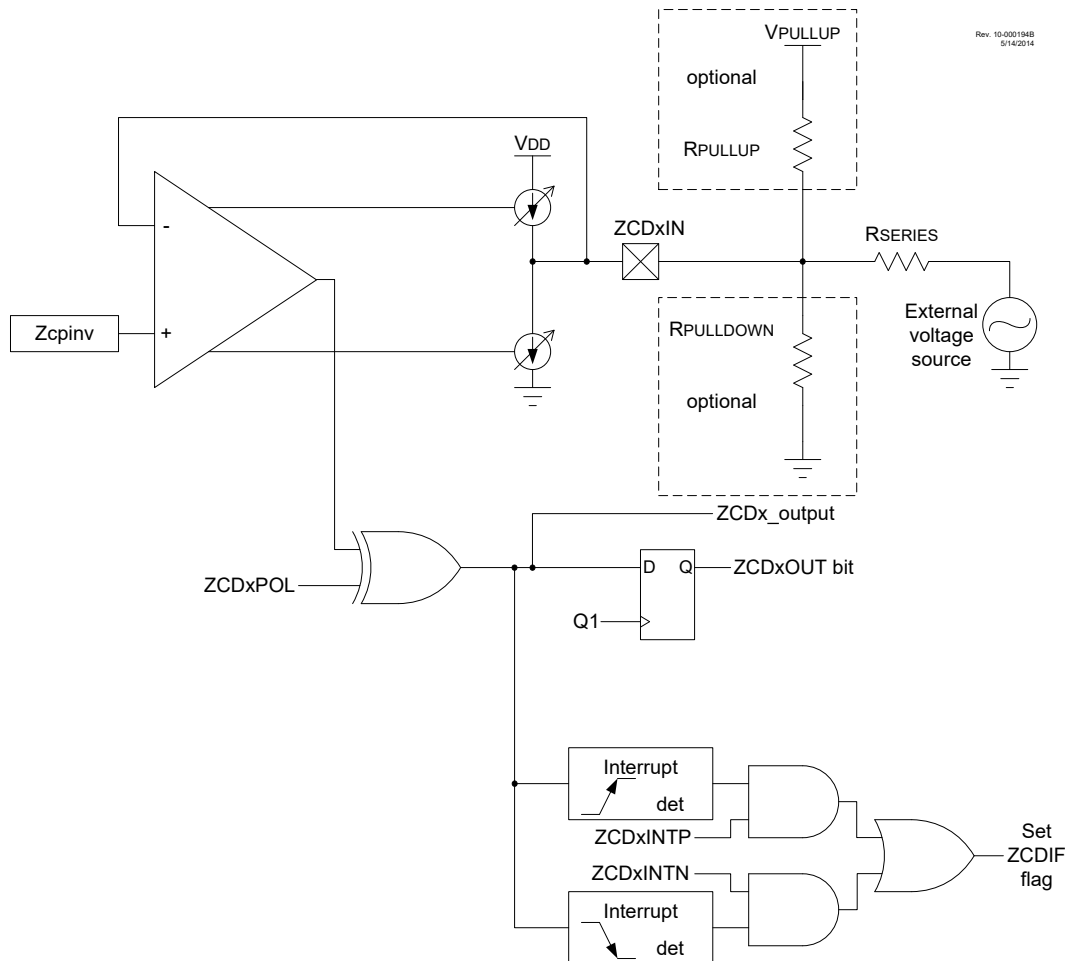
| PCH | Positive Input Source |
|-----|--|
| 111 | CxV _P connects to V _{SS} |
| 110 | CxV _P connects to FVR Buffer 2 |
| 101 | CxV _P connects to DAC1 output |
| 100 | CxV _P not connected |
| 011 | CxV _P not connected |
| 010 | CxV _P not connected |
| 001 | CxV _P not connected |
| 000 | CxV _P connects to CxIN0+ pin |

35. ZCD - Zero-Cross Detection Module

The Zero-Cross Detection (ZCD) module detects when an A/C signal crosses through the ground potential. The actual zero crossing threshold is the zero crossing reference voltage, Z_{CPINV} , which is typically 0.75V above ground.

The connection to the signal to be detected is through a series current-limiting resistor. The module applies a current source or sink to the ZCD pin to maintain a constant voltage on the pin, thereby preventing the pin voltage from forward biasing the ESD protection diodes. When the applied voltage is greater than the reference voltage, the module sinks current. When the applied voltage is less than the reference voltage, the module sources current. The current source and sink action keeps the pin voltage constant over the full range of the applied voltage. The ZCD module is shown in the following simplified block diagram.

Figure 35-1. Simplified ZCD Block Diagram



The ZCD module is useful when monitoring an A/C waveform for, but not limited to, the following purposes:

- A/C period measurement
- Accurate long term time measurement
- Dimmer phase delayed drive
- Low EMI cycle switching

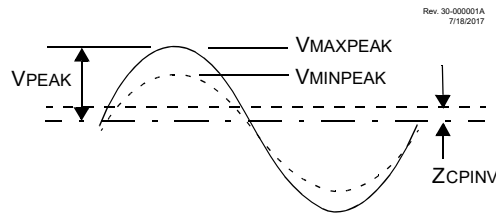
35.1 External Resistor Selection

The ZCD module requires a current-limiting resistor in series with the external voltage source. The impedance and rating of this resistor depends on the external source peak voltage. Select a resistor value that will drop all of the peak voltage when the current through the resistor is nominally 300 μ A. Make sure that the ZCD I/O pin internal weak pull-up is disabled so it does not interfere with the current source and sink.

Equation 35-1. External Resistor

$$R_{SERIES} = \frac{V_{PEAK}}{3 \times 10^{-4}}$$

Figure 35-2. External Voltage Source



35.2 ZCD Logic Output

The ZCD module includes a Status bit, which can be read to determine whether the current source or sink is active. The **OUT** bit is set when the current sink is active, and cleared when the current source is active. The OUT bit is affected by the polarity bit.

The OUT signal can also be used as input to other modules. This is controlled by the registers of the corresponding module. OUT can be used as follows:

- Gate source for TMR1/3/5
- Clock source for TMR2/4/6
- Reset source for TMR2/4/6

35.3 ZCD Logic Polarity

The **POL** bit inverts the OUT bit relative to the current source and sink output. When the POL bit is set, a OUT high indicates that the current source is active, and a low output indicates that the current sink is active.

The POL bit affects the ZCD interrupts.

35.4 ZCD Interrupts

An interrupt will be generated upon a change in the ZCD logic output when the appropriate interrupt enables are set. A rising edge detector and a falling edge detector are present in the ZCD for this purpose.

The ZCDIF bit of the PIR2 register will be set when either edge detector is triggered and its associated enable bit is set. The **INTP** bit in the ZCDxCON register enables rising edge interrupts and the **INTN** bit in the ZCDxCON register enables falling edge interrupts.

To fully enable the interrupt, the following bits must be set:

- ZCDIE bit of the PIE2 register
- INTP bit for rising edge detection
- INTN bit for falling edge detection

- PEIE and GIE bits of the INTCON register

Changing the POL bit will cause an interrupt, regardless of the level of the SEN bit.

The ZCDIF bit of the PIR2 register must be cleared in software as part of the interrupt service. If another edge is detected while this flag is being cleared, the flag will still be set at the end of the sequence.

Related Links

[5.8.10 INTCON](#)

[7.7.13 PIR2](#)

35.5 Correction for Z_{CPINV} Offset

The actual voltage at which the ZCD switches is the reference voltage at the non-inverting input of the ZCD op amp. For external voltage source waveforms other than square waves, this voltage offset from zero causes the zero-cross event to occur either too early or too late.

35.5.1 Correction by AC Coupling

When the external voltage source is sinusoidal, the effects of the Z_{CPINV} offset can be eliminated by isolating the external voltage source from the ZCD pin with a capacitor, in addition to the voltage reducing resistor. The capacitor will cause a phase shift resulting in the ZCD output switch in advance of the actual zero crossing event. The phase shift will be the same for both rising and falling zero crossings, which can be compensated for by either delaying the CPU response to the ZCD switch by a timer or other means, or selecting a capacitor value large enough that the phase shift is negligible.

To determine the series resistor and capacitor values for this configuration, start by computing the impedance, Z, to obtain a peak current of 300 μA. Next, arbitrarily select a suitably large non-polar capacitor and compute its reactance, X_C, at the external voltage source frequency. Finally, compute the series resistor, capacitor peak voltage, and phase shift by the formulas shown below.

When this technique is used and the input signal is not present, the ZCD will tend to oscillate. To avoid this oscillation, connect the ZCD pin to V_{DD} or GND with a high-impedance resistor such as 200K.

Equation 35-2. R-C Equations

V_{PEAK} = external voltage source peak voltage

f = external voltage source frequency

C = series capacitor

R = series resistor

V_C = Peak capacitor voltage

Φ = Capacitor induced zero crossing phase advance in radians

T_Φ = Time ZC event occurs before actual zero crossing

$$Z = \frac{V_{PEAK}}{3 \times 10^{-4}}$$

$$X_C = \frac{1}{2\pi fC}$$

$$R = \sqrt{Z^2 - X_C^2}$$

$$V_C = X_C(3 \times 10^{-4})$$

$$\Phi = \tan^{-1}\theta\left(\frac{X_C}{R}\right)$$

$$T_\Phi = \frac{\Phi}{2\pi f}$$

Equation 35-3. R-C Calculation Example

$$V_{rms} = 120$$

$$V_{PEAK} = V_{rms} \times \sqrt{2} = 169.7$$

$$f = 60 \text{ Hz}$$

$$C = 0.1 \mu F$$

$$Z = \frac{V_{PEAK}}{3 \times 10^{-4}} = \frac{169.7}{3 \times 10^{-4}} = 565.7 \text{ k}\Omega$$

$$X_C = \frac{1}{2\pi f C} = \frac{1}{2\pi \times 60 \times 10^{-7}} = 26.53 \text{ k}\Omega$$

$$R = \sqrt{Z^2 - X_C^2} = 565.1 \text{ k}\Omega \text{ (computed)}$$

$$R_a = 560 \text{ k}\Omega \text{ (used)}$$

$$Z_R = \sqrt{R_a^2 + X_C^2} = 560.6 \text{ k}\Omega$$

$$I_{PEAK} = \frac{V_{PEAK}}{Z_R} = 302.7 \times 10^{-6} \text{ A}$$

$$V_C = X_C \times I_{PEAK} = 8.0 \text{ V}$$

$$\Phi = \tan^{-1}\theta\left(\frac{X_C}{R}\right) = 0.047 \text{ radians}$$

$$T_\Phi = \frac{\Phi}{2\pi f} = 125.6 \mu s$$

35.5.2 Correction By Offset Current

When the waveform is varying relative to V_{SS} , then the zero cross is detected too early as the waveform falls and too late as the waveform rises. When the waveform is varying relative to V_{DD} , then the zero cross is detected too late as the waveform rises and too early as the waveform falls. The actual offset time can be determined for sinusoidal waveforms with the corresponding equations shown below.

Equation 35-4. ZCD Event Offset

When External Voltage source is relative to V_{SS}

$$T_{offset} = \frac{\sin^{-1}\left(\frac{Z_{CPINV}}{V_{PEAK}}\right)}{2\pi f}$$

When External Voltage source is relative to V_{DD}

$$T_{offset} = \frac{\sin^{-1}\left(\frac{V_{DD} - Z_{CPINV}}{V_{PEAK}}\right)}{2\pi f}$$

This offset time can be compensated for by adding a pull-up or pull-down biasing resistor to the ZCD pin. A pull-up resistor is used when the external voltage source is varying relative to V_{SS} . A pull-down resistor is used when the voltage is varying relative to V_{DD} . The resistor adds a bias to the ZCD pin so that the target external voltage source must go to zero to pull the pin voltage to the Z_{CPINV} switching voltage. The pull-up or pull-down value can be determined with the equations shown below.

Equation 35-5. ZCD Pull-up/Pull-down Resistor

When External Voltage source is relative to V_{SS}

$$R_{pullup} = \frac{R_{SERIES}(V_{pullup} - Z_{CPINV})}{Z_{CPINV}}$$

When External Voltage source is relative to V_{DD}

$$R_{pulldown} = \frac{R_{SERIES}(Z_{CPINV})}{(V_{DD} - Z_{CPINV})}$$

35.6 Handling V_{PEAK} Variations

If the peak amplitude of the external voltage is expected to vary, the series resistor must be selected to keep the ZCD current source and sink below the design maximum range of $\pm 600 \mu A$ and above a reasonable minimum range. A general rule of thumb is that the maximum peak voltage can be no more than six times the minimum peak voltage. To ensure that the maximum current does not exceed $\pm 600 \mu A$ and the minimum is at least $\pm 100 \mu A$, compute the series resistance as shown in [Equation 35-6](#). The compensating pull-up for this series resistance can be determined with the equations shown in [Equation 35-5](#) because the pull-up value is independent from the peak voltage.

Equation 35-6. Series R for V range

$$R_{SERIES} = \frac{V_{MAX_PEAK} + V_{MIN_PEAK}}{7 \times 10^{-4}}$$

35.7 Operation During Sleep

The ZCD current sources and interrupts are unaffected by Sleep.

35.8 Effects of a Reset

The ZCD circuit can be configured to default to the Active or Inactive state on Power-on Reset (POR). When the \overline{ZCD} Configuration bit is cleared, the ZCD circuit will be active at POR. When the \overline{ZCD} Configuration bit is set, the [SEN](#) bit must be set to enable the ZCD module.

35.9 Disabling the ZCD Module

The ZCD module can be disabled in two ways:

1. The \overline{ZCD} Configuration bit disables the ZCD module when set. When this is the case then the ZCD module will be enabled by setting the [SEN](#) bit. When the \overline{ZCD} bit is cleared, the ZCD is always enabled and the [SEN](#) bit has no effect.
2. The ZCD can also be disabled using the ZCDMD bit of the PMD3 register. This is subject to the status of the \overline{ZCD} bit.

35.10 Register Summary: ZCD Control

| Address | Name | Bit Pos. | | | | | | | | |
|-------------|----------|----------|-----|--|-----|-----|--|--|------|------|
| 0x00 ... | Reserved | | | | | | | | | |
| 0x091E | | | | | | | | | | |
| 0x091F | ZCDCON | 7:0 | SEN | | OUT | POL | | | INTP | INTN |

35.11 Register Definitions: ZCD Control

35.11.1 ZCDCON

Name: ZCDCON
Address: 0x91F

Zero-Cross Detect Control Register

| | | | | | | | | |
|--------|-----|---|-----|-----|---|---|------|------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | SEN | | OUT | POL | | | INTP | INTN |
| Access | R/W | | RO | R/W | | | R/W | R/W |
| Reset | 0 | | x | 0 | | | 0 | 0 |

Bit 7 – SEN Zero-Cross Detect Software Enable bit

This bit is ignored when ZCD fuse is cleared.

| Value | Condition | Description |
|-------|---------------------|---|
| X | ZCD Config fuse = 0 | Zero-cross detect is always enabled. This bit is ignored. |
| 1 | ZCD Config fuse = 1 | Zero-cross detect is enabled. ZCD pin is forced to output to source and sink current. |
| 0 | ZCD Config fuse = 1 | Zero-cross detect is disabled. ZCD pin operates according to PPS and TRIS controls. |

Bit 5 – OUT Zero-Cross Detect Data Output bit

| Value | Condition | Description |
|-------|-----------|-----------------------------|
| 1 | POL = 0 | ZCD pin is sinking current |
| 0 | POL = 0 | ZCD pin is sourcing current |
| 1 | POL = 1 | ZCD pin is sourcing current |
| 0 | POL = 1 | ZCD pin is sinking current |

Bit 4 – POL Zero-Cross Detect Polarity bit

| Value | Description |
|-------|----------------------------------|
| 1 | ZCD logic output is inverted |
| 0 | ZCD logic output is not inverted |

Bit 1 – INTP Zero-Cross Detect Positive-Going Edge Interrupt Enable bit

| Value | Description |
|-------|--|
| 1 | ZCDIF bit is set on low-to-high ZCD_output transition |
| 0 | ZCDIF bit is unaffected by low-to-high ZCD_output transition |

Bit 0 – INTN Zero-Cross Detect Negative-Going Edge Interrupt Enable bit

| Value | Description |
|-------|--|
| 1 | ZCDIF bit is set on high-to-low ZCD_output transition |
| 0 | ZCDIF bit is unaffected by high-to-low ZCD_output transition |

36. Register Summary

| Address | Name | Bit Pos. | | | | | | | | |
|---------|----------|----------|-------------|--------|--------|--------|--------|--------|--------|--------|
| 0x00 | INDF0 | 7:0 | INDF0[7:0] | | | | | | | |
| 0x01 | INDF1 | 7:0 | INDF1[7:0] | | | | | | | |
| 0x02 | PCL | 7:0 | PCL[7:0] | | | | | | | |
| 0x03 | STATUS | 7:0 | | | | TO | PD | Z | DC | C |
| 0x04 | FSR0 | 7:0 | FSR0[7:0] | | | | | | | |
| | | 15:8 | FSR0[15:8] | | | | | | | |
| 0x06 | FSR1 | 7:0 | FSR1[7:0] | | | | | | | |
| | | 15:8 | FSR1[15:8] | | | | | | | |
| 0x08 | BSR | 7:0 | BSR[5:0] | | | | | | | |
| 0x09 | WREG | 7:0 | WREG[7:0] | | | | | | | |
| 0x0A | PCLATH | 7:0 | PCLATH[6:0] | | | | | | | |
| 0x0B | INTCON | 7:0 | GIE | PEIE | | | | | | INTEDG |
| 0x0C | PORTA | 7:0 | | | RA5 | RA4 | RA3 | RA2 | RA1 | RA0 |
| 0x0D | PORTB | 7:0 | RB7 | RB6 | RB5 | RB4 | | | | |
| 0x0E | PORTC | 7:0 | RC7 | RC6 | RC5 | RC4 | RC3 | RC2 | RC1 | RC0 |
| 0x0F | Reserved | | | | | | | | | |
| 0x11 | | | | | | | | | | |
| 0x12 | TRISA | 7:0 | | | TRISA5 | TRISA4 | | TRISA2 | TRISA1 | TRISA0 |
| 0x13 | TRISB | 7:0 | TRISB7 | TRISB6 | TRISB5 | TRISB4 | | | | |
| 0x14 | TRISC | 7:0 | TRISC7 | TRISC6 | TRISC5 | TRISC4 | TRISC3 | TRISC2 | TRISC1 | TRISC0 |
| 0x15 | Reserved | | | | | | | | | |
| 0x17 | | | | | | | | | | |
| 0x18 | LATA | 7:0 | | | LATA5 | LATA4 | | LATA2 | LATA1 | LATA0 |
| 0x19 | LATB | 7:0 | LATB7 | LATB6 | LATB5 | LATB4 | | | | |
| 0x1A | LATC | 7:0 | LATC7 | LATC6 | LATC5 | LATC4 | LATC3 | LATC2 | LATC1 | LATC0 |
| 0x1B | Reserved | | | | | | | | | |
| 0x7F | | | | | | | | | | |
| 0x80 | INDF0 | 7:0 | INDF0[7:0] | | | | | | | |
| 0x81 | INDF1 | 7:0 | INDF1[7:0] | | | | | | | |
| 0x82 | PCL | 7:0 | PCL[7:0] | | | | | | | |
| 0x83 | STATUS | 7:0 | | | | TO | PD | Z | DC | C |
| 0x84 | FSR0 | 7:0 | FSR0[7:0] | | | | | | | |
| | | 15:8 | FSR0[15:8] | | | | | | | |
| 0x86 | FSR1 | 7:0 | FSR1[7:0] | | | | | | | |
| | | 15:8 | FSR1[15:8] | | | | | | | |
| 0x88 | BSR | 7:0 | BSR[5:0] | | | | | | | |
| 0x89 | WREG | 7:0 | WREG[7:0] | | | | | | | |
| 0x8A | PCLATH | 7:0 | PCLATH[6:0] | | | | | | | |
| 0x8B | INTCON | 7:0 | GIE | PEIE | | | | | | INTEDG |
| 0x8C | ADLTH | 7:0 | LTH[7:0] | | | | | | | |
| | | 15:8 | LTH[15:8] | | | | | | | |
| 0x8E | ADUTH | 7:0 | UTH[7:0] | | | | | | | |
| | | 15:8 | UTH[15:8] | | | | | | | |
| 0x90 | ADERR | 7:0 | ERR[7:0] | | | | | | | |
| | | 15:8 | ERR[15:8] | | | | | | | |
| 0x92 | ADSTPT | 7:0 | STPT[7:0] | | | | | | | |
| | | 15:8 | STPT[15:8] | | | | | | | |
| 0x94 | ADFLTR | 7:0 | FLTR[7:0] | | | | | | | |
| | | 15:8 | FLTR[15:8] | | | | | | | |
| 0x96 | ADACC | 7:0 | ACC[7:0] | | | | | | | |
| | | 15:8 | ACC[15:8] | | | | | | | |
| | | 23:16 | ACC[17:16] | | | | | | | |
| 0x99 | ADCNT | 7:0 | CNT[7:0] | | | | | | | |

PIC16(L)F18426/46

Register Summary

|continued | | | | | | | | | | |
|----------------|----------|----------|--------|-------|-----------|------|-------|-------|------|---------------|
| Address | Name | Bit Pos. | | | | | | | | |
| 0x9A | ADRPT | 7:0 | | | | | | | | RPT[7:0] |
| 0x9B | ADPREV | 7:0 | | | | | | | | PREV[7:0] |
| | | 15:8 | | | | | | | | PREV[15:8] |
| 0x9D | ADRES | 7:0 | | | | | | | | RES[7:0] |
| | | 15:8 | | | | | | | | RES[15:8] |
| 0x9F | ADPCH | 7:0 | | | | | | | | PCH[5:0] |
| 0xA0 | Reserved | | | | | | | | | |
| ... | | | | | | | | | | |
| 0xFF | | | | | | | | | | |
| 0x0100 | INDF0 | 7:0 | | | | | | | | INDF0[7:0] |
| 0x0101 | INDF1 | 7:0 | | | | | | | | INDF1[7:0] |
| 0x0102 | PCL | 7:0 | | | | | | | | PCL[7:0] |
| 0x0103 | STATUS | 7:0 | | | | | | | | TO PD Z DC C |
| 0x0104 | FSR0 | 7:0 | | | | | | | | FSR0[7:0] |
| | | 15:8 | | | | | | | | FSR0[15:8] |
| 0x0106 | FSR1 | 7:0 | | | | | | | | FSR1[7:0] |
| | | 15:8 | | | | | | | | FSR1[15:8] |
| 0x0108 | BSR | 7:0 | | | | | | | | BSR[5:0] |
| 0x0109 | WREG | 7:0 | | | | | | | | WREG[7:0] |
| 0x010A | PCLATH | 7:0 | | | | | | | | PCLATH[6:0] |
| 0x010B | INTCON | 7:0 | GIE | PEIE | | | | | | INTEDG |
| 0x010C | ADACQ | 7:0 | | | | | | | | ACQ[7:0] |
| | | 15:8 | | | | | | | | ACQ[12:8] |
| 0x010E | ADCAP | 7:0 | | | | | | | | CAP[4:0] |
| 0x010F | ADPRE | 7:0 | | | | | | | | PRE[7:0] |
| | | 15:8 | | | | | | | | PRE[12:8] |
| 0x0111 | ADCON0 | 7:0 | ON | CONT | | | CS | | FRM | GO |
| 0x0112 | ADCON1 | 7:0 | PPOL | IPEN | GPOL | | | | | DSEN |
| 0x0113 | ADCON2 | 7:0 | PSIS | | CRS[2:0] | | ACLR | | | MD[2:0] |
| 0x0114 | ADCON3 | 7:0 | | | CALC[2:0] | | SOI | | | TMD[2:0] |
| 0x0115 | ADSTAT | 7:0 | OV | UTHR | LTHR | | MATH | | | STAT[2:0] |
| 0x0116 | ADREF | 7:0 | | | | | NREF | | | PREF[1:0] |
| 0x0117 | ADACT | 7:0 | | | | | | | | ACT[4:0] |
| 0x0118 | ADCLK | 7:0 | | | | | | | | CS[5:0] |
| 0x0119 | RC1REG | 7:0 | | | | | | | | RCREG[7:0] |
| 0x011A | TX1REG | 7:0 | | | | | | | | TXREG[7:0] |
| 0x011B | SP1BRG | 7:0 | | | | | | | | SPBRG[7:0] |
| | | 15:8 | | | | | | | | SPBRG[15:8] |
| 0x011D | RC1STA | 7:0 | SPEN | RX9 | SREN | CREN | ADDEN | FERR | OERR | RX9D |
| 0x011E | TX1STA | 7:0 | CSRC | TX9 | TXEN | SYNC | SENDB | BRGH | TRMT | TX9D |
| 0x011F | BAUD1CON | 7:0 | ABDOVF | RCIDL | | | SCKP | BRG16 | | WUE ABDEN |
| 0x0120 | Reserved | | | | | | | | | |
| ... | | | | | | | | | | |
| 0x017F | | | | | | | | | | |
| 0x0180 | INDF0 | 7:0 | | | | | | | | INDF0[7:0] |
| 0x0181 | INDF1 | 7:0 | | | | | | | | INDF1[7:0] |
| 0x0182 | PCL | 7:0 | | | | | | | | PCL[7:0] |
| 0x0183 | STATUS | 7:0 | | | | | | | | TO PD Z DC C |
| 0x0184 | FSR0 | 7:0 | | | | | | | | FSR0[7:0] |
| | | 15:8 | | | | | | | | FSR0[15:8] |
| 0x0186 | FSR1 | 7:0 | | | | | | | | FSR1[7:0] |
| | | 15:8 | | | | | | | | FSR1[15:8] |
| 0x0188 | BSR | 7:0 | | | | | | | | BSR[5:0] |
| 0x0189 | WREG | 7:0 | | | | | | | | WREG[7:0] |
| 0x018A | PCLATH | 7:0 | | | | | | | | PCLATH[6:0] |
| 0x018B | INTCON | 7:0 | GIE | PEIE | | | | | | INTEDG |
| 0x018C | SSP1BUF | 7:0 | | | | | | | | BUF[7:0] |
| 0x018D | SSP1ADD | 7:0 | | | | | | | | ADD[7:0] |
| 0x018E | SSP1MSK | 7:0 | | | | | | | | MSK[6:0] MSK0 |

PIC16(L)F18426/46

Register Summary

.....continued

| Address | Name | Bit Pos. | | | | | | | | | |
|---------|----------|----------|-------------|---------|-------------|-------|-------------|-------|-------------|--------|--|
| 0x018F | SSP1STAT | 7:0 | SMP | CKE | D/Ā | P | S | R/W | UA | BF | |
| 0x0190 | SSP1CON1 | 7:0 | WCOL | SSPOV | SSPEN | CKP | SSPM[3:0] | | | | |
| 0x0191 | SSP1CON2 | 7:0 | GCEN | ACKSTAT | ACKDT | ACKEN | RCEN | PEN | RSEN | SEN | |
| 0x0192 | SSP1CON3 | 7:0 | ACKTIM | PCIE | SCIE | BOEN | SDAHT | SBCDE | AHEN | DHEN | |
| 0x0193 | Reserved | | | | | | | | | | |
| ... | | | | | | | | | | | |
| 0x0195 | | | | | | | | | | | |
| 0x0196 | SSP2BUF | 7:0 | BUF[7:0] | | | | | | | | |
| 0x0197 | SSP2ADD | 7:0 | ADD[7:0] | | | | | | | | |
| 0x0198 | SSP2MSK | 7:0 | MSK[6:0] | | | | | | | | |
| 0x0199 | SSP2STAT | 7:0 | SMP | CKE | D/Ā | P | S | R/W | UA | BF | |
| 0x019A | SSP2CON1 | 7:0 | WCOL | SSPOV | SSPEN | CKP | SSPM[3:0] | | | | |
| 0x019B | SSP2CON2 | 7:0 | GCEN | ACKSTAT | ACKDT | ACKEN | RCEN | PEN | RSEN | SEN | |
| 0x019C | SSP2CON3 | 7:0 | ACKTIM | PCIE | SCIE | BOEN | SDAHT | SBCDE | AHEN | DHEN | |
| 0x019D | Reserved | | | | | | | | | | |
| ... | | | | | | | | | | | |
| 0x01FF | | | | | | | | | | | |
| 0x0200 | INDF0 | 7:0 | INDF0[7:0] | | | | | | | | |
| 0x0201 | INDF1 | 7:0 | INDF1[7:0] | | | | | | | | |
| 0x0202 | PCL | 7:0 | PCL[7:0] | | | | | | | | |
| 0x0203 | STATUS | 7:0 | | | | TO | PD | Z | DC | C | |
| 0x0204 | FSR0 | 7:0 | FSR0[7:0] | | | | | | | | |
| | | 15:8 | FSR0[15:8] | | | | | | | | |
| 0x0206 | FSR1 | 7:0 | FSR1[7:0] | | | | | | | | |
| | | 15:8 | FSR1[15:8] | | | | | | | | |
| 0x0208 | BSR | 7:0 | BSR[5:0] | | | | | | | | |
| 0x0209 | WREG | 7:0 | WREG[7:0] | | | | | | | | |
| 0x020A | PCLATH | 7:0 | PCLATH[6:0] | | | | | | | | |
| 0x020B | INTCON | 7:0 | GIE | PEIE | | | | | | INTEDG | |
| 0x020C | TMR1 | 7:0 | TMRx[7:0] | | | | | | | | |
| | | 15:8 | TMRx[15:8] | | | | | | | | |
| 0x020E | T1CON | 7:0 | | | CKPS[1:0] | | | SYNC | RD16 | ON | |
| 0x020F | T1GCON | 7:0 | GE | GPOL | GTM | GSPM | GGO/DONE | GVAL | | | |
| 0x0210 | TMR1GATE | 7:0 | GSS[4:0] | | | | | | | | |
| 0x0211 | TMR1CLK | 7:0 | CS[4:0] | | | | | | | | |
| 0x0212 | TMR3 | 7:0 | TMRx[7:0] | | | | | | | | |
| | | 15:8 | TMRx[15:8] | | | | | | | | |
| 0x0214 | T3CON | 7:0 | | | CKPS[1:0] | | | SYNC | RD16 | ON | |
| 0x0215 | T3GCON | 7:0 | GE | GPOL | GTM | GSPM | GGO/DONE | GVAL | | | |
| 0x0216 | TMR3GATE | 7:0 | GSS[4:0] | | | | | | | | |
| 0x0217 | TMR3CLK | 7:0 | CS[4:0] | | | | | | | | |
| 0x0218 | TMR5 | 7:0 | TMRx[7:0] | | | | | | | | |
| | | 15:8 | TMRx[15:8] | | | | | | | | |
| 0x021A | T5CON | 7:0 | | | CKPS[1:0] | | | SYNC | RD16 | ON | |
| 0x021B | T5GCON | 7:0 | GE | GPOL | GTM | GSPM | GGO/DONE | GVAL | | | |
| 0x021C | TMR5GATE | 7:0 | GSS[4:0] | | | | | | | | |
| 0x021D | TMR5CLK | 7:0 | CS[4:0] | | | | | | | | |
| 0x021E | CCPTMRS0 | 7:0 | C4TSEL[1:0] | | C3TSEL[1:0] | | C2TSEL[1:0] | | C1TSEL[1:0] | | |
| 0x021F | CCPTMRS1 | 7:0 | | | P7TSEL[1:0] | | P6TSEL[1:0] | | | | |
| 0x0220 | Reserved | | | | | | | | | | |
| ... | | | | | | | | | | | |
| 0x027F | | | | | | | | | | | |
| 0x0280 | INDF0 | 7:0 | INDF0[7:0] | | | | | | | | |
| 0x0281 | INDF1 | 7:0 | INDF1[7:0] | | | | | | | | |
| 0x0282 | PCL | 7:0 | PCL[7:0] | | | | | | | | |
| 0x0283 | STATUS | 7:0 | | | | TO | PD | Z | DC | C | |
| 0x0284 | FSR0 | 7:0 | FSR0[7:0] | | | | | | | | |
| | | 15:8 | FSR0[15:8] | | | | | | | | |

PIC16(L)F18426/46

Register Summary

.....continued

| Address | Name | Bit Pos. | | | | | | | | | |
|---------------|----------|----------|-------------|-----------|-------|-----------|------------|---|----|----------|-------|
| 0x0286 | FSR1 | 7:0 | FSR1[7:0] | | | | | | | | |
| | | 15:8 | FSR1[15:8] | | | | | | | | |
| 0x0288 | BSR | 7:0 | BSR[5:0] | | | | | | | | |
| 0x0289 | WREG | 7:0 | WREG[7:0] | | | | | | | | |
| 0x028A | PCLATH | 7:0 | PCLATH[6:0] | | | | | | | | |
| 0x028B | INTCON | 7:0 | GIE | PEIE | | | | | | INTEDG | |
| 0x028C | T2TMR | 7:0 | TxTMR[7:0] | | | | | | | | |
| 0x028D | T2PR | 7:0 | TxPR[7:0] | | | | | | | | |
| 0x028E | T2CON | 7:0 | ON | CKPS[2:0] | | | OUTPS[3:0] | | | | |
| 0x028F | T2HLT | 7:0 | PSYNC | CPOL | CSYNC | MODE[4:0] | | | | | |
| 0x0290 | T2CLKCON | 7:0 | CS[3:0] | | | | | | | | |
| 0x0291 | T2RST | 7:0 | RSEL[3:0] | | | | | | | | |
| 0x0292 | T4TMR | 7:0 | TxTMR[7:0] | | | | | | | | |
| 0x0293 | T4PR | 7:0 | TxPR[7:0] | | | | | | | | |
| 0x0294 | T4CON | 7:0 | ON | CKPS[2:0] | | | OUTPS[3:0] | | | | |
| 0x0295 | T4HLT | 7:0 | PSYNC | CPOL | CSYNC | MODE[4:0] | | | | | |
| 0x0296 | T4CLKCON | 7:0 | CS[3:0] | | | | | | | | |
| 0x0297 | T4RST | 7:0 | RSEL[3:0] | | | | | | | | |
| 0x0298 | T6TMR | 7:0 | TxTMR[7:0] | | | | | | | | |
| 0x0299 | T6PR | 7:0 | TxPR[7:0] | | | | | | | | |
| 0x029A | T6CON | 7:0 | ON | CKPS[2:0] | | | OUTPS[3:0] | | | | |
| 0x029B | T6HLT | 7:0 | PSYNC | CPOL | CSYNC | MODE[4:0] | | | | | |
| 0x029C | T6CLKCON | 7:0 | CS[3:0] | | | | | | | | |
| 0x029D | T6RST | 7:0 | RSEL[3:0] | | | | | | | | |
| 0x029E | Reserved | | | | | | | | | | |
| 0x029F | ADCPON0 | 7:0 | CPON | | | | | | | | CPRDY |
| 0x02A0 | Reserved | | | | | | | | | | |
| ... 0x02FF | | | | | | | | | | | |
| 0x0300 | INDF0 | 7:0 | INDF0[7:0] | | | | | | | | |
| 0x0301 | INDF1 | 7:0 | INDF1[7:0] | | | | | | | | |
| 0x0302 | PCL | 7:0 | PCL[7:0] | | | | | | | | |
| 0x0303 | STATUS | 7:0 | | | | TO | PD | Z | DC | C | |
| 0x0304 | FSR0 | 7:0 | FSR0[7:0] | | | | | | | | |
| | | 15:8 | FSR0[15:8] | | | | | | | | |
| 0x0306 | FSR1 | 7:0 | FSR1[7:0] | | | | | | | | |
| | | 15:8 | FSR1[15:8] | | | | | | | | |
| 0x0308 | BSR | 7:0 | BSR[5:0] | | | | | | | | |
| 0x0309 | WREG | 7:0 | WREG[7:0] | | | | | | | | |
| 0x030A | PCLATH | 7:0 | PCLATH[6:0] | | | | | | | | |
| 0x030B | INTCON | 7:0 | GIE | PEIE | | | | | | INTEDG | |
| 0x030C | CCPR1 | 7:0 | CCPR[7:0] | | | | | | | | |
| | | 15:8 | CCPR[15:8] | | | | | | | | |
| 0x030E | CCP1CON | 7:0 | EN | | OUT | FMT | MODE[3:0] | | | | |
| 0x030F | CCP1CAP | 7:0 | | | | | | | | CTS[2:0] | |
| 0x0310 | CCPR2 | 7:0 | CCPR[7:0] | | | | | | | | |
| | | 15:8 | CCPR[15:8] | | | | | | | | |
| 0x0312 | CCP2CON | 7:0 | EN | | OUT | FMT | MODE[3:0] | | | | |
| 0x0313 | CCP2CAP | 7:0 | | | | | | | | CTS[2:0] | |
| 0x0314 | CCPR3 | 7:0 | CCPR[7:0] | | | | | | | | |
| | | 15:8 | CCPR[15:8] | | | | | | | | |
| 0x0316 | CCP3CON | 7:0 | EN | | OUT | FMT | MODE[3:0] | | | | |
| 0x0317 | CCP3CAP | 7:0 | | | | | | | | CTS[2:0] | |
| 0x0318 | CCPR4 | 7:0 | CCPR[7:0] | | | | | | | | |
| | | 15:8 | CCPR[15:8] | | | | | | | | |
| 0x031A | CCP4CON | 7:0 | EN | | OUT | FMT | MODE[3:0] | | | | |
| 0x031B | CCP4CAP | 7:0 | | | | | | | | CTS[2:0] | |

PIC16(L)F18426/46

Register Summary

.....continued

| Address | Name | Bit Pos. | | | | | | | | | |
|-------------------------|----------|----------|----------|------|-----|-----|----|----|---|------------|-------------|
| 0x031C ... 0x037F | Reserved | | | | | | | | | | |
| 0x0380 | INDF0 | 7:0 | | | | | | | | INDF0[7:0] | |
| 0x0381 | INDF1 | 7:0 | | | | | | | | INDF1[7:0] | |
| 0x0382 | PCL | 7:0 | | | | | | | | PCL[7:0] | |
| 0x0383 | STATUS | 7:0 | | | | | TO | PD | Z | DC | C |
| 0x0384 | FSR0 | 7:0 | | | | | | | | | FSR0[7:0] |
| | | 15:8 | | | | | | | | | FSR0[15:8] |
| 0x0386 | FSR1 | 7:0 | | | | | | | | | FSR1[7:0] |
| | | 15:8 | | | | | | | | | FSR1[15:8] |
| 0x0388 | BSR | 7:0 | | | | | | | | | BSR[5:0] |
| 0x0389 | WREG | 7:0 | | | | | | | | | WREG[7:0] |
| 0x038A | PCLATH | 7:0 | | | | | | | | | PCLATH[6:0] |
| 0x038B | INTCON | 7:0 | GIE | PEIE | | | | | | | INTEDG |
| 0x038C | PWM6DC | 7:0 | DCL[1:0] | | | | | | | | |
| | | 15:8 | | | | | | | | | DCH[7:0] |
| 0x038E | PWM6CON | 7:0 | EN | | OUT | POL | | | | | |
| 0x038F | Reserved | | | | | | | | | | |
| 0x0390 | PWM7DC | 7:0 | DCL[1:0] | | | | | | | | |
| | | 15:8 | | | | | | | | | DCH[7:0] |
| 0x0392 | PWM7CON | 7:0 | EN | | OUT | POL | | | | | |
| 0x0393 ... 0x03FF | Reserved | | | | | | | | | | |
| 0x0400 | INDF0 | 7:0 | | | | | | | | | INDF0[7:0] |
| 0x0401 | INDF1 | 7:0 | | | | | | | | | INDF1[7:0] |
| 0x0402 | PCL | 7:0 | | | | | | | | | PCL[7:0] |
| 0x0403 | STATUS | 7:0 | | | | | TO | PD | Z | DC | C |
| 0x0404 | FSR0 | 7:0 | | | | | | | | | FSR0[7:0] |
| | | 15:8 | | | | | | | | | FSR0[15:8] |
| 0x0406 | FSR1 | 7:0 | | | | | | | | | FSR1[7:0] |
| | | 15:8 | | | | | | | | | FSR1[15:8] |
| 0x0408 | BSR | 7:0 | | | | | | | | | BSR[5:0] |
| 0x0409 | WREG | 7:0 | | | | | | | | | WREG[7:0] |
| 0x040A | PCLATH | 7:0 | | | | | | | | | PCLATH[6:0] |
| 0x040B | INTCON | 7:0 | GIE | PEIE | | | | | | | INTEDG |
| 0x040C ... 0x047F | Reserved | | | | | | | | | | |
| 0x0480 | INDF0 | 7:0 | | | | | | | | | INDF0[7:0] |
| 0x0481 | INDF1 | 7:0 | | | | | | | | | INDF1[7:0] |
| 0x0482 | PCL | 7:0 | | | | | | | | | PCL[7:0] |
| 0x0483 | STATUS | 7:0 | | | | | TO | PD | Z | DC | C |
| 0x0484 | FSR0 | 7:0 | | | | | | | | | FSR0[7:0] |
| | | 15:8 | | | | | | | | | FSR0[15:8] |
| 0x0486 | FSR1 | 7:0 | | | | | | | | | FSR1[7:0] |
| | | 15:8 | | | | | | | | | FSR1[15:8] |
| 0x0488 | BSR | 7:0 | | | | | | | | | BSR[5:0] |
| 0x0489 | WREG | 7:0 | | | | | | | | | WREG[7:0] |
| 0x048A | PCLATH | 7:0 | | | | | | | | | PCLATH[6:0] |
| 0x048B | INTCON | 7:0 | GIE | PEIE | | | | | | | INTEDG |
| 0x048C | SMT1TMR | 7:0 | | | | | | | | | TMR[7:0] |
| | | 15:8 | | | | | | | | | TMR[15:8] |
| | | 23:16 | | | | | | | | | TMR[23:16] |
| 0x048F | SMT1CPR | 7:0 | | | | | | | | | CPR[7:0] |
| | | 15:8 | | | | | | | | | CPR[15:8] |
| | | 23:16 | | | | | | | | | CPR[23:16] |

PIC16(L)F18426/46

Register Summary

.....continued

| Address | Name | Bit Pos. | | | | | | | | | | | | | | | | | | | | | | |
|---------|----------|----------|------------|--------|-------------|---------|------|-----------|---------|----|--|--|--|---------|-------------|--|--|--|--|--------|--------|--------------|------------|--|
| 0x0492 | SMT1CPW | 7:0 | CPW[7:0] | | | | | | | | | | | | | | | | | | | | | |
| | | 15:8 | CPW[15:8] | | | | | | | | | | | | | | | | | | | | | |
| | | 23:16 | CPW[23:16] | | | | | | | | | | | | | | | | | | | | | |
| 0x0495 | SMT1PR | 7:0 | PR[7:0] | | | | | | | | | | | | | | | | | | | | | |
| | | 15:8 | PR[15:8] | | | | | | | | | | | | | | | | | | | | | |
| | | 23:16 | PR[23:16] | | | | | | | | | | | | | | | | | | | | | |
| 0x0498 | SMT1CON0 | 7:0 | EN | | STP | WPOL | SPOL | CPOL | PS[1:0] | | | | | | | | | | | | | | | |
| 0x0499 | SMT1CON1 | 7:0 | GO | REPEAT | | | | MODE[3:0] | | | | | | | | | | | | | | | | |
| 0x049A | SMT1STAT | 7:0 | CPRUP | CPWUP | | RST | | TS | WS | AS | | | | | | | | | | | | | | |
| 0x049B | SMT1CLK | 7:0 | | | | | | CSEL[2:0] | | | | | | | | | | | | | | | | |
| 0x049C | SMT1SIG | 7:0 | | | | | | SSEL[4:0] | | | | | | | | | | | | | | | | |
| 0x049D | SMT1WIN | 7:0 | | | | | | WSEL[4:0] | | | | | | | | | | | | | | | | |
| 0x049E | Reserved | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | |
| 0x04FF | | | | | | | | | | | | | | | | | | | | | | | | |
| 0x0500 | INDF0 | 7:0 | INDF0[7:0] | | | | | | | | | | | | | | | | | | | | | |
| 0x0501 | INDF1 | 7:0 | INDF1[7:0] | | | | | | | | | | | | | | | | | | | | | |
| 0x0502 | PCL | 7:0 | PCL[7:0] | | | | | | | | | | | | | | | | | | | | | |
| 0x0503 | STATUS | 7:0 | | | | TO | PD | Z | DC | C | | | | | | | | | | | | | | |
| 0x0504 | FSR0 | 7:0 | FSR0[7:0] | | | | | | | | | | | | | | | | | | | | | |
| | | 15:8 | FSR0[15:8] | | | | | | | | | | | | | | | | | | | | | |
| 0x0506 | FSR1 | 7:0 | FSR1[7:0] | | | | | | | | | | | | | | | | | | | | | |
| | | 15:8 | FSR1[15:8] | | | | | | | | | | | | | | | | | | | | | |
| 0x0508 | BSR | 7:0 | | | | | | BSR[5:0] | | | | | | | | | | | | | | | | |
| 0x0509 | WREG | 7:0 | WREG[7:0] | | | | | | | | | | | | | | | | | | | | | |
| 0x050A | PCLATH | 7:0 | | | PCLATH[6:0] | | | | | | | | | | | | | | | | | | | |
| 0x050B | INTCON | 7:0 | GIE | PEIE | | | | | | | | | | | | | | | | INTEDG | | | | |
| 0x050C | Reserved | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | |
| 0x057F | | | | | | | | | | | | | | | | | | | | | | | | |
| 0x0580 | INDF0 | 7:0 | INDF0[7:0] | | | | | | | | | | | | | | | | | | | | | |
| 0x0581 | INDF1 | 7:0 | INDF1[7:0] | | | | | | | | | | | | | | | | | | | | | |
| 0x0582 | PCL | 7:0 | PCL[7:0] | | | | | | | | | | | | | | | | | | | | | |
| 0x0583 | STATUS | 7:0 | | | | TO | PD | Z | DC | C | | | | | | | | | | | | | | |
| 0x0584 | FSR0 | 7:0 | FSR0[7:0] | | | | | | | | | | | | | | | | | | | | | |
| | | 15:8 | FSR0[15:8] | | | | | | | | | | | | | | | | | | | | | |
| 0x0586 | FSR1 | 7:0 | FSR1[7:0] | | | | | | | | | | | | | | | | | | | | | |
| | | 15:8 | FSR1[15:8] | | | | | | | | | | | | | | | | | | | | | |
| 0x0588 | BSR | 7:0 | | | | | | BSR[5:0] | | | | | | | | | | | | | | | | |
| 0x0589 | WREG | 7:0 | WREG[7:0] | | | | | | | | | | | | | | | | | | | | | |
| 0x058A | PCLATH | 7:0 | | | PCLATH[6:0] | | | | | | | | | | | | | | | | | | | |
| 0x058B | INTCON | 7:0 | GIE | PEIE | | | | | | | | | | | | | | | | | INTEDG | | | |
| 0x058C | NCO1ACC | 7:0 | ACC[7:0] | | | | | | | | | | | | | | | | | | | | | |
| | | 15:8 | ACC[15:8] | | | | | | | | | | | | | | | | | | | | | |
| | | 23:16 | ACC[23:16] | | | | | | | | | | | | | | | | | | | | | |
| 0x058F | NCO1INC | 7:0 | INC[7:0] | | | | | | | | | | | | | | | | | | | | | |
| | | 15:8 | INC[15:8] | | | | | | | | | | | | | | | | | | | | | |
| | | 23:16 | | | | | | | | | | | | | | | | | | | | | INC[19:16] | |
| 0x0592 | NCO1CON | 7:0 | EN | | OUT | POL | | | | | | | | | | | | | | | | PFM | | |
| 0x0593 | NCO1CLK | 7:0 | PWS[2:0] | | | | | | | | | | | | CKS[3:0] | | | | | | | | | |
| 0x0594 | Reserved | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | |
| 0x059B | | | | | | | | | | | | | | | | | | | | | | | | |
| 0x059C | TMR0L | 7:0 | TMR0L[7:0] | | | | | | | | | | | | | | | | | | | | | |
| 0x059D | TMR0H | 7:0 | TMR0H[7:0] | | | | | | | | | | | | | | | | | | | | | |
| 0x059E | T0CON0 | 7:0 | T0EN | | T0OUT | T016BIT | | | | | | | | | | | | | | | | T0OUTPS[3:0] | | |
| 0x059F | T0CON1 | 7:0 | T0CS[2:0] | | | | | | | | | | | T0ASYNC | T0CKPS[3:0] | | | | | | | | | |

PIC16(L)F18426/46

Register Summary

.....continued

| Address | Name | Bit Pos. | | | | | | | | | |
|-------------------------|----------|----------|----------|------|------|------|-----------|------|-----------|------|-------------|
| 0x05A0 ... 0x05FF | Reserved | | | | | | | | | | |
| 0x0600 | INDF0 | 7:0 | | | | | | | | | INDF0[7:0] |
| 0x0601 | INDF1 | 7:0 | | | | | | | | | INDF1[7:0] |
| 0x0602 | PCL | 7:0 | | | | | | | | | PCL[7:0] |
| 0x0603 | STATUS | 7:0 | | | | | TO | PD | Z | DC | C |
| 0x0604 | FSR0 | 7:0 | | | | | | | | | FSR0[7:0] |
| | | 15:8 | | | | | | | | | FSR0[15:8] |
| 0x0606 | FSR1 | 7:0 | | | | | | | | | FSR1[7:0] |
| | | 15:8 | | | | | | | | | FSR1[15:8] |
| 0x0608 | BSR | 7:0 | | | | | | | | | BSR[5:0] |
| 0x0609 | WREG | 7:0 | | | | | | | | | WREG[7:0] |
| 0x060A | PCLATH | 7:0 | | | | | | | | | PCLATH[6:0] |
| 0x060B | INTCON | 7:0 | GIE | PEIE | | | | | | | INTEDG |
| 0x060C | CWG1CLK | 7:0 | | | | | | | | | CS |
| 0x060D | CWG1ISM | 7:0 | | | | | | | | | ISM[3:0] |
| 0x060E | CWG1DBR | 7:0 | | | | | | | | | DBR[5:0] |
| 0x060F | CWG1DBF | 7:0 | | | | | | | | | DBF[5:0] |
| 0x0610 | CWG1CON0 | 7:0 | EN | LD | | | | | | | MODE[2:0] |
| 0x0611 | CWG1CON1 | 7:0 | | | IN | | POLD | POLC | POLB | POLA | |
| 0x0612 | CWG1AS0 | 7:0 | SHUTDOWN | REN | | | LSBD[1:0] | | LSAC[1:0] | | |
| 0x0613 | CWG1AS1 | 7:0 | | | AS5E | AS4E | AS3E | AS2E | AS1E | AS0E | |
| 0x0614 | CWG1STR | 7:0 | OVRD | OVRC | OVRB | OVRA | STRD | STRC | STRB | STRA | |
| 0x0615 | Reserved | | | | | | | | | | |
| 0x0616 | CWG2CLK | 7:0 | | | | | | | | | CS |
| 0x0617 | CWG2ISM | 7:0 | | | | | | | | | ISM[3:0] |
| 0x0618 | CWG2DBR | 7:0 | | | | | | | | | DBR[5:0] |
| 0x0619 | CWG2DBF | 7:0 | | | | | | | | | DBF[5:0] |
| 0x061A | CWG2CON0 | 7:0 | EN | LD | | | | | | | MODE[2:0] |
| 0x061B | CWG2CON1 | 7:0 | | | IN | | POLD | POLC | POLB | POLA | |
| 0x061C | CWG2AS0 | 7:0 | SHUTDOWN | REN | | | LSBD[1:0] | | LSAC[1:0] | | |
| 0x061D | CWG2AS1 | 7:0 | | | AS5E | AS4E | AS3E | AS2E | AS1E | AS0E | |
| 0x061E | CWG2STR | 7:0 | OVRD | OVRC | OVRB | OVRA | STRD | STRC | STRB | STRA | |
| 0x061F ... 0x067F | Reserved | | | | | | | | | | |
| 0x0680 | INDF0 | 7:0 | | | | | | | | | INDF0[7:0] |
| 0x0681 | INDF1 | 7:0 | | | | | | | | | INDF1[7:0] |
| 0x0682 | PCL | 7:0 | | | | | | | | | PCL[7:0] |
| 0x0683 | STATUS | 7:0 | | | | | TO | PD | Z | DC | C |
| 0x0684 | FSR0 | 7:0 | | | | | | | | | FSR0[7:0] |
| | | 15:8 | | | | | | | | | FSR0[15:8] |
| 0x0686 | FSR1 | 7:0 | | | | | | | | | FSR1[7:0] |
| | | 15:8 | | | | | | | | | FSR1[15:8] |
| 0x0688 | BSR | 7:0 | | | | | | | | | BSR[5:0] |
| 0x0689 | WREG | 7:0 | | | | | | | | | WREG[7:0] |
| 0x068A | PCLATH | 7:0 | | | | | | | | | PCLATH[6:0] |
| 0x068B | INTCON | 7:0 | GIE | PEIE | | | | | | | INTEDG |
| 0x068C ... 0x06FF | Reserved | | | | | | | | | | |
| 0x0700 | INDF0 | 7:0 | | | | | | | | | INDF0[7:0] |
| 0x0701 | INDF1 | 7:0 | | | | | | | | | INDF1[7:0] |
| 0x0702 | PCL | 7:0 | | | | | | | | | PCL[7:0] |
| 0x0703 | STATUS | 7:0 | | | | | TO | PD | Z | DC | C |
| 0x0704 | FSR0 | 7:0 | | | | | | | | | FSR0[7:0] |
| | | 15:8 | | | | | | | | | FSR0[15:8] |

PIC16(L)F18426/46

Register Summary

|continued | | | | | | | | | | |
|----------------|----------|----------|-------------|--------|--------|---------|--------|-----------|-----------|---------|
| Address | Name | Bit Pos. | | | | | | | | |
| 0x0706 | FSR1 | 7:0 | FSR1[7:0] | | | | | | | |
| | | 15:8 | FSR1[15:8] | | | | | | | |
| 0x0708 | BSR | 7:0 | BSR[5:0] | | | | | | | |
| 0x0709 | WREG | 7:0 | WREG[7:0] | | | | | | | |
| 0x070A | PCLATH | 7:0 | PCLATH[6:0] | | | | | | | |
| 0x070B | INTCON | 7:0 | GIE | PEIE | | | | | | INTEDG |
| 0x070C | PIR0 | 7:0 | | | TMR0IF | IOCIF | | | | INTF |
| 0x070D | PIR1 | 7:0 | OSFIF | CSWIF | | | | | ADTIF | ADIF |
| 0x070E | PIR2 | 7:0 | | ZCDIF | | | | | C2IF | C1IF |
| 0x070F | PIR3 | 7:0 | | | RC1IF | TX1IF | BCL2IF | SSP2IF | BCL1IF | SSP1IF |
| 0x0710 | PIR4 | 7:0 | | | TMR6IF | TMR5IF | TMR4IF | TMR3IF | TMR2IF | TMR1IF |
| 0x0711 | PIR5 | 7:0 | CLC4IF | CLC3IF | CL24IF | CLC1IF | | TMR5GIF | TMR3GIF | TMR1GIF |
| 0x0712 | PIR6 | 7:0 | | | | | CCP4IF | CCP3IF | CCP2IF | CCP1IF |
| 0x0713 | PIR7 | 7:0 | | | NVMIF | NCO1IF | | | CWG2IF | CWG1IF |
| 0x0714 | PIR8 | 7:0 | | | | | | SMT1PWAIF | SMT1PRAIF | SMT1IF |
| 0x0715 | Reserved | | | | | | | | | |
| 0x0716 | PIE0 | 7:0 | | | TMR0IE | IOCIE | | | | INTE |
| 0x0717 | PIE1 | 7:0 | OSFIE | CSWIE | | | | | ADTIE | ADIE |
| 0x0718 | PIE2 | 7:0 | | ZCDIE | | | | | C2IE | C1IE |
| 0x0719 | PIE3 | 7:0 | | | RC1IE | TX1IE | BCL2IE | SSP2IE | BCL1IE | SSP1IE |
| 0x071A | PIE4 | 7:0 | | | TMR6IE | TMR5IE | TMR4IE | TMR3IE | TMR2IE | TMR1IE |
| 0x071B | PIE5 | 7:0 | CLC4IE | CLC3IE | CLC2IE | CLC1IE | | TMR5GIE | TMR3GIE | TMR1GIE |
| 0x071C | PIE6 | 7:0 | | | | | CCP4IE | CCP3IE | CCP2IE | CCP1IE |
| 0x071D | PIE7 | 7:0 | | | NVMIE | NCO1IE | | | CWG2IE | CWG1IE |
| 0x071E | PIE8 | 7:0 | | | | | | SMT1PWAIE | SMT1PRAIE | SMT1IE |
| 0x071F | Reserved | | | | | | | | | |
| ... | | | | | | | | | | |
| 0x077F | | | | | | | | | | |
| 0x0780 | INDF0 | 7:0 | INDF0[7:0] | | | | | | | |
| 0x0781 | INDF1 | 7:0 | INDF1[7:0] | | | | | | | |
| 0x0782 | PCL | 7:0 | PCL[7:0] | | | | | | | |
| 0x0783 | STATUS | 7:0 | | | TO | PD | | Z | DC | C |
| 0x0784 | FSR0 | 7:0 | FSR0[7:0] | | | | | | | |
| | | 15:8 | FSR0[15:8] | | | | | | | |
| 0x0786 | FSR1 | 7:0 | FSR1[7:0] | | | | | | | |
| | | 15:8 | FSR1[15:8] | | | | | | | |
| 0x0788 | BSR | 7:0 | BSR[5:0] | | | | | | | |
| 0x0789 | WREG | 7:0 | WREG[7:0] | | | | | | | |
| 0x078A | PCLATH | 7:0 | PCLATH[6:0] | | | | | | | |
| 0x078B | INTCON | 7:0 | GIE | PEIE | | | | | | INTEDG |
| 0x078C | Reserved | | | | | | | | | |
| ... | | | | | | | | | | |
| 0x0795 | | | | | | | | | | |
| 0x0796 | PMD0 | 7:0 | SYSCMD | FVRMD | | | | NVMMD | CLKRMD | IOCMD |
| 0x0797 | PMD1 | 7:0 | | TMR6MD | TMR5MD | TMR4MD | TMR3MD | TMR2MD | TMR1MD | TMR0MD |
| 0x0798 | PMD2 | 7:0 | NCO1MD | | | | | | | |
| 0x0799 | PMD3 | 7:0 | | DAC1MD | ADCMD | | | C2MD | C1MD | ZCDMD |
| 0x079A | PMD4 | 7:0 | | PWM7MD | PWM6MD | | CCP4MD | CCP3MD | CCP2MD | CCP1MD |
| 0x079B | PMD5 | 7:0 | | CWG2MD | CWG1MD | | | | | |
| 0x079C | PMD6 | 7:0 | | | | UART1MD | | | MSSP2MD | MSSP1MD |
| 0x079D | PMD7 | 7:0 | | | SMT1MD | CLC4MD | CLC3MD | CLC2MD | CLC1MD | DSM1MD |
| 0x079E | Reserved | | | | | | | | | |
| ... | | | | | | | | | | |
| 0x07FF | | | | | | | | | | |
| 0x0800 | INDF0 | 7:0 | INDF0[7:0] | | | | | | | |
| 0x0801 | INDF1 | 7:0 | INDF1[7:0] | | | | | | | |
| 0x0802 | PCL | 7:0 | PCL[7:0] | | | | | | | |
| 0x0803 | STATUS | 7:0 | | | TO | PD | | Z | DC | C |

PIC16(L)F18426/46

Register Summary

.....continued

| Address | Name | Bit Pos. | | | | | | | | | | | | | | | | | | |
|---------|----------|----------|--------------|---------|-------|-------|--------|-------|--|--|--|--|--|--|--|--|-----------|------------|-----------|-----|
| 0x0804 | FSR0 | 7:0 | FSR0[7:0] | | | | | | | | | | | | | | | | | |
| | | 15:8 | FSR0[15:8] | | | | | | | | | | | | | | | | | |
| 0x0806 | FSR1 | 7:0 | FSR1[7:0] | | | | | | | | | | | | | | | | | |
| | | 15:8 | FSR1[15:8] | | | | | | | | | | | | | | | | | |
| 0x0808 | BSR | 7:0 | BSR[5:0] | | | | | | | | | | | | | | | | | |
| 0x0809 | WREG | 7:0 | WREG[7:0] | | | | | | | | | | | | | | | | | |
| 0x080A | PCLATH | 7:0 | PCLATH[6:0] | | | | | | | | | | | | | | | | | |
| 0x080B | INTCON | 7:0 | GIE | PEIE | | | | | | | | | | | | | | INTEDG | | |
| 0x080C | WDTCON0 | 7:0 | WDTPS[4:0] | | | | | | | | | | | | | | | | | |
| 0x080D | WDTCON1 | 7:0 | WDTCS[2:0] | | | | | | | | | | | | | | | | | |
| 0x080E | WDTPSL | 7:0 | PSCNTL[7:0] | | | | | | | | | | | | | | | | | |
| 0x080F | WDTPSH | 7:0 | PSCNTH[7:0] | | | | | | | | | | | | | | | | | |
| 0x0810 | WDTTMR | 7:0 | WDTTMR[4:0] | | | | | | | | | | | | | | | | | |
| 0x0811 | BORCON | 7:0 | SBOREN | | | | | | | | | | | | | | STATE | PSCNT[1:0] | BORRDY | |
| 0x0812 | VREGCON | 7:0 | VREGPM | | | | | | | | | | | | | | | | | |
| 0x0813 | PCON0 | 7:0 | STKOVF | STKUNF | WDTWV | RWDT | RMCLR | RI | | | | | | | | | | | POR | BOR |
| 0x0814 | PCON1 | 7:0 | MEMV | | | | | | | | | | | | | | | | | |
| 0x0815 | Reserved | | | | | | | | | | | | | | | | | | | |
| 0x0819 | | | | | | | | | | | | | | | | | | | | |
| 0x081A | NVMADR | 7:0 | NVMADR[7:0] | | | | | | | | | | | | | | | | | |
| | | 15:8 | NVMADR[14:8] | | | | | | | | | | | | | | | | | |
| 0x081C | NVMDAT | 7:0 | NVMDAT[7:0] | | | | | | | | | | | | | | | | | |
| | | 15:8 | NVMDAT[13:8] | | | | | | | | | | | | | | | | | |
| 0x081E | NVMCON1 | 7:0 | NVMREGS | | | | | | | | | | | | | | | | | |
| 0x081F | NVMCON2 | 7:0 | NVMCON2[7:0] | | | | | | | | | | | | | | | | | |
| 0x0820 | Reserved | | | | | | | | | | | | | | | | | | | |
| 0x087F | | | | | | | | | | | | | | | | | | | | |
| 0x0880 | INDF0 | 7:0 | INDF0[7:0] | | | | | | | | | | | | | | | | | |
| 0x0881 | INDF1 | 7:0 | INDF1[7:0] | | | | | | | | | | | | | | | | | |
| 0x0882 | PCL | 7:0 | PCL[7:0] | | | | | | | | | | | | | | | | | |
| 0x0883 | STATUS | 7:0 | TO PD Z DC C | | | | | | | | | | | | | | | | | |
| 0x0884 | FSR0 | 7:0 | FSR0[7:0] | | | | | | | | | | | | | | | | | |
| | | 15:8 | FSR0[15:8] | | | | | | | | | | | | | | | | | |
| 0x0886 | FSR1 | 7:0 | FSR1[7:0] | | | | | | | | | | | | | | | | | |
| | | 15:8 | FSR1[15:8] | | | | | | | | | | | | | | | | | |
| 0x0888 | BSR | 7:0 | BSR[5:0] | | | | | | | | | | | | | | | | | |
| 0x0889 | WREG | 7:0 | WREG[7:0] | | | | | | | | | | | | | | | | | |
| 0x088A | PCLATH | 7:0 | PCLATH[6:0] | | | | | | | | | | | | | | | | | |
| 0x088B | INTCON | 7:0 | GIE | PEIE | | | | | | | | | | | | | | INTEDG | | |
| 0x088C | CPUDOZE | 7:0 | IDLEN | DOZEN | ROI | DOE | | | | | | | | | | | DOZE[2:0] | | | |
| 0x088D | OSCCON1 | 7:0 | NOSC[2:0] | | | | | | | | | | | | | | | | | |
| 0x088E | OSCCON2 | 7:0 | COSC[2:0] | | | | | | | | | | | | | | | | | |
| 0x088F | OSCCON3 | 7:0 | CSWHOLD | SOSCPWR | | | ORDY | NOSCR | | | | | | | | | | | NDIV[3:0] | |
| 0x0890 | OSCSTAT | 7:0 | EXTOR | HFOR | MFOR | LFOR | SOR | ADOR | | | | | | | | | | | PLL | |
| 0x0891 | OSCEN | 7:0 | EXTOEN | HFOEN | MFOEN | LFOEN | SOSCEN | ADOEN | | | | | | | | | | | CDIV[3:0] | |
| 0x0892 | OSCTUNE | 7:0 | HFTUN[5:0] | | | | | | | | | | | | | | | | | |
| 0x0893 | OSCFRQ | 7:0 | HFFRQ[2:0] | | | | | | | | | | | | | | | | | |
| 0x0894 | Reserved | | | | | | | | | | | | | | | | | | | |
| 0x0895 | CLKRCON | 7:0 | EN | | | | | | | | | | | | | | DC[1:0] | DIV[2:0] | | |
| 0x0896 | CLKRCLK | 7:0 | CLK[3:0] | | | | | | | | | | | | | | | | | |
| 0x0897 | MD1CON0 | 7:0 | EN | | | | | | | | | | | | | | OUT | OPOL | BIT | |
| 0x0898 | MD1CON1 | 7:0 | CHPOL | | | | | | | | | | | | | | | | | |
| 0x0899 | MD1SRC | 7:0 | CHSYNC | | | | | | | | | | | | | | | | | |
| 0x089A | MD1CARL | 7:0 | SRCS[4:0] | | | | | | | | | | | | | | | | | |
| 0x089B | MD1CARH | 7:0 | CLS[3:0] | | | | | | | | | | | | | | | | | |
| | | | CHS[3:0] | | | | | | | | | | | | | | | | | |

PIC16(L)F18426/46

Register Summary

.....continued

| Address | Name | Bit Pos. | | | | | | | | | |
|-------------------------|----------|----------|-------|--------|------|-------|----|----|-------------|-------------|------------|
| 0x089C ... 0x08FF | Reserved | | | | | | | | | | |
| 0x0900 | INDF0 | 7:0 | | | | | | | | INDF0[7:0] | |
| 0x0901 | INDF1 | 7:0 | | | | | | | | INDF1[7:0] | |
| 0x0902 | PCL | 7:0 | | | | | | | | PCL[7:0] | |
| 0x0903 | STATUS | 7:0 | | | | | TO | PD | Z | DC | C |
| 0x0904 | FSR0 | 7:0 | | | | | | | | | FSR0[7:0] |
| | | 15:8 | | | | | | | | | FSR0[15:8] |
| 0x0906 | FSR1 | 7:0 | | | | | | | | | FSR1[7:0] |
| | | 15:8 | | | | | | | | | FSR1[15:8] |
| 0x0908 | BSR | 7:0 | | | | | | | | BSR[5:0] | |
| 0x0909 | WREG | 7:0 | | | | | | | | WREG[7:0] | |
| 0x090A | PCLATH | 7:0 | | | | | | | | PCLATH[6:0] | |
| 0x090B | INTCON | 7:0 | GIE | PEIE | | | | | | | INTEDG |
| 0x090C | FVRCON | 7:0 | FVREN | FVRRDY | TSEN | TSRNG | | | CDAFVR[1:0] | | ADFVR[1:0] |
| 0x090D | Reserved | | | | | | | | | | |
| 0x090E | DAC1CON0 | 7:0 | EN | | OE1 | | | | PSS[1:0] | | NSS |
| 0x090F | DAC1CON1 | 7:0 | | | | | | | DAC1R[4:0] | | |
| 0x0910 ... 0x091E | Reserved | | | | | | | | | | |
| 0x091F | ZCDCON | 7:0 | SEN | | OUT | POL | | | | INTP | INTN |
| 0x0920 ... 0x097F | Reserved | | | | | | | | | | |
| 0x0980 | INDF0 | 7:0 | | | | | | | | INDF0[7:0] | |
| 0x0981 | INDF1 | 7:0 | | | | | | | | INDF1[7:0] | |
| 0x0982 | PCL | 7:0 | | | | | | | | PCL[7:0] | |
| 0x0983 | STATUS | 7:0 | | | | | TO | PD | Z | DC | C |
| 0x0984 | FSR0 | 7:0 | | | | | | | | | FSR0[7:0] |
| | | 15:8 | | | | | | | | | FSR0[15:8] |
| 0x0986 | FSR1 | 7:0 | | | | | | | | | FSR1[7:0] |
| | | 15:8 | | | | | | | | | FSR1[15:8] |
| 0x0988 | BSR | 7:0 | | | | | | | | BSR[5:0] | |
| 0x0989 | WREG | 7:0 | | | | | | | | WREG[7:0] | |
| 0x098A | PCLATH | 7:0 | | | | | | | | PCLATH[6:0] | |
| 0x098B | INTCON | 7:0 | GIE | PEIE | | | | | | | INTEDG |
| 0x098C ... 0x098E | Reserved | | | | | | | | | | |
| 0x098F | CMOUT | 7:0 | | | | | | | | MC2OUT | MC1OUT |
| 0x0990 | CM1CON0 | 7:0 | EN | OUT | | POL | | | | HYS | SYNC |
| 0x0991 | CM1CON1 | 7:0 | | | | | | | | INTP | INTN |
| 0x0992 | CM1NCH | 7:0 | | | | | | | | NCH[2:0] | |
| 0x0993 | CM1PCH | 7:0 | | | | | | | | PCH[2:0] | |
| 0x0994 | CM2CON0 | 7:0 | EN | OUT | | POL | | | | HYS | SYNC |
| 0x0995 | CM2CON1 | 7:0 | | | | | | | | INTP | INTN |
| 0x0996 | CM2NCH | 7:0 | | | | | | | | NCH[2:0] | |
| 0x0997 | CM2PCH | 7:0 | | | | | | | | PCH[2:0] | |
| 0x0998 ... 0x09FF | Reserved | | | | | | | | | | |
| 0x0A00 | INDF0 | 7:0 | | | | | | | | INDF0[7:0] | |
| 0x0A01 | INDF1 | 7:0 | | | | | | | | INDF1[7:0] | |
| 0x0A02 | PCL | 7:0 | | | | | | | | PCL[7:0] | |
| 0x0A03 | STATUS | 7:0 | | | | | TO | PD | Z | DC | C |
| 0x0A04 | FSR0 | 7:0 | | | | | | | | | FSR0[7:0] |
| | | 15:8 | | | | | | | | | FSR0[15:8] |

PIC16(L)F18426/46

Register Summary

.....continued

| Address | Name | Bit Pos. | | | | | | | | |
|-------------------------|----------|----------|-------------|------|----|----|---|----|---|--------|
| 0x0A06 | FSR1 | 7:0 | FSR1[7:0] | | | | | | | |
| | | 15:8 | FSR1[15:8] | | | | | | | |
| 0x0A08 | BSR | 7:0 | BSR[5:0] | | | | | | | |
| 0x0A09 | WREG | 7:0 | WREG[7:0] | | | | | | | |
| 0x0A0A | PCLATH | 7:0 | PCLATH[6:0] | | | | | | | |
| 0x0A0B | INTCON | 7:0 | GIE | PEIE | | | | | | INTEDG |
| 0x0A0C ... 0x0A7F | Reserved | | | | | | | | | |
| 0x0A80 | INDF0 | 7:0 | INDF0[7:0] | | | | | | | |
| 0x0A81 | INDF1 | 7:0 | INDF1[7:0] | | | | | | | |
| 0x0A82 | PCL | 7:0 | PCL[7:0] | | | | | | | |
| 0x0A83 | STATUS | 7:0 | | | TO | PD | Z | DC | C | |
| 0x0A84 | FSR0 | 7:0 | FSR0[7:0] | | | | | | | |
| | | 15:8 | FSR0[15:8] | | | | | | | |
| 0x0A86 | FSR1 | 7:0 | FSR1[7:0] | | | | | | | |
| | | 15:8 | FSR1[15:8] | | | | | | | |
| 0x0A88 | BSR | 7:0 | BSR[5:0] | | | | | | | |
| 0x0A89 | WREG | 7:0 | WREG[7:0] | | | | | | | |
| 0x0A8A | PCLATH | 7:0 | PCLATH[6:0] | | | | | | | |
| 0x0A8B | INTCON | 7:0 | GIE | PEIE | | | | | | INTEDG |
| 0x0A8C ... 0x0AFF | Reserved | | | | | | | | | |
| 0x0B00 | INDF0 | 7:0 | INDF0[7:0] | | | | | | | |
| 0x0B01 | INDF1 | 7:0 | INDF1[7:0] | | | | | | | |
| 0x0B02 | PCL | 7:0 | PCL[7:0] | | | | | | | |
| 0x0B03 | STATUS | 7:0 | | | TO | PD | Z | DC | C | |
| 0x0B04 | FSR0 | 7:0 | FSR0[7:0] | | | | | | | |
| | | 15:8 | FSR0[15:8] | | | | | | | |
| 0x0B06 | FSR1 | 7:0 | FSR1[7:0] | | | | | | | |
| | | 15:8 | FSR1[15:8] | | | | | | | |
| 0x0B08 | BSR | 7:0 | BSR[5:0] | | | | | | | |
| 0x0B09 | WREG | 7:0 | WREG[7:0] | | | | | | | |
| 0x0B0A | PCLATH | 7:0 | PCLATH[6:0] | | | | | | | |
| 0x0B0B | INTCON | 7:0 | GIE | PEIE | | | | | | INTEDG |
| 0x0B0C ... 0x0B7F | Reserved | | | | | | | | | |
| 0x0B80 | INDF0 | 7:0 | INDF0[7:0] | | | | | | | |
| 0x0B81 | INDF1 | 7:0 | INDF1[7:0] | | | | | | | |
| 0x0B82 | PCL | 7:0 | PCL[7:0] | | | | | | | |
| 0x0B83 | STATUS | 7:0 | | | TO | PD | Z | DC | C | |
| 0x0B84 | FSR0 | 7:0 | FSR0[7:0] | | | | | | | |
| | | 15:8 | FSR0[15:8] | | | | | | | |
| 0x0B86 | FSR1 | 7:0 | FSR1[7:0] | | | | | | | |
| | | 15:8 | FSR1[15:8] | | | | | | | |
| 0x0B88 | BSR | 7:0 | BSR[5:0] | | | | | | | |
| 0x0B89 | WREG | 7:0 | WREG[7:0] | | | | | | | |
| 0x0B8A | PCLATH | 7:0 | PCLATH[6:0] | | | | | | | |
| 0x0B8B | INTCON | 7:0 | GIE | PEIE | | | | | | INTEDG |
| 0x0B8C ... 0x0BFF | Reserved | | | | | | | | | |
| 0x0C00 | INDF0 | 7:0 | INDF0[7:0] | | | | | | | |
| 0x0C01 | INDF1 | 7:0 | INDF1[7:0] | | | | | | | |
| 0x0C02 | PCL | 7:0 | PCL[7:0] | | | | | | | |
| 0x0C03 | STATUS | 7:0 | | | TO | PD | Z | DC | C | |

PIC16(L)F18426/46

Register Summary

.....continued

| Address | Name | Bit Pos. | | | | | | | | |
|---------|----------|----------|-------------|------|--|----|----|---|----|--------|
| 0x0C04 | FSR0 | 7:0 | FSR0[7:0] | | | | | | | |
| | | 15:8 | FSR0[15:8] | | | | | | | |
| 0x0C06 | FSR1 | 7:0 | FSR1[7:0] | | | | | | | |
| | | 15:8 | FSR1[15:8] | | | | | | | |
| 0x0C08 | BSR | 7:0 | BSR[5:0] | | | | | | | |
| 0x0C09 | WREG | 7:0 | WREG[7:0] | | | | | | | |
| 0x0C0A | PCLATH | 7:0 | PCLATH[6:0] | | | | | | | |
| 0x0C0B | INTCON | 7:0 | GIE | PEIE | | | | | | INTEDG |
| 0x0C0C | Reserved | | | | | | | | | |
| 0x0C7F | | | | | | | | | | |
| 0x0C80 | INDF0 | 7:0 | INDF0[7:0] | | | | | | | |
| 0x0C81 | INDF1 | 7:0 | INDF1[7:0] | | | | | | | |
| 0x0C82 | PCL | 7:0 | PCL[7:0] | | | | | | | |
| 0x0C83 | STATUS | 7:0 | | | | TO | PD | Z | DC | C |
| 0x0C84 | FSR0 | 7:0 | FSR0[7:0] | | | | | | | |
| | | 15:8 | FSR0[15:8] | | | | | | | |
| 0x0C86 | FSR1 | 7:0 | FSR1[7:0] | | | | | | | |
| | | 15:8 | FSR1[15:8] | | | | | | | |
| 0x0C88 | BSR | 7:0 | BSR[5:0] | | | | | | | |
| 0x0C89 | WREG | 7:0 | WREG[7:0] | | | | | | | |
| 0x0C8A | PCLATH | 7:0 | PCLATH[6:0] | | | | | | | |
| 0x0C8B | INTCON | 7:0 | GIE | PEIE | | | | | | INTEDG |
| 0x0C8C | Reserved | | | | | | | | | |
| 0x0CFF | | | | | | | | | | |
| 0x0D00 | INDF0 | 7:0 | INDF0[7:0] | | | | | | | |
| 0x0D01 | INDF1 | 7:0 | INDF1[7:0] | | | | | | | |
| 0x0D02 | PCL | 7:0 | PCL[7:0] | | | | | | | |
| 0x0D03 | STATUS | 7:0 | | | | TO | PD | Z | DC | C |
| 0x0D04 | FSR0 | 7:0 | FSR0[7:0] | | | | | | | |
| | | 15:8 | FSR0[15:8] | | | | | | | |
| 0x0D06 | FSR1 | 7:0 | FSR1[7:0] | | | | | | | |
| | | 15:8 | FSR1[15:8] | | | | | | | |
| 0x0D08 | BSR | 7:0 | BSR[5:0] | | | | | | | |
| 0x0D09 | WREG | 7:0 | WREG[7:0] | | | | | | | |
| 0x0D0A | PCLATH | 7:0 | PCLATH[6:0] | | | | | | | |
| 0x0D0B | INTCON | 7:0 | GIE | PEIE | | | | | | INTEDG |
| 0x0D0C | Reserved | | | | | | | | | |
| 0x0D7F | | | | | | | | | | |
| 0x0D80 | INDF0 | 7:0 | INDF0[7:0] | | | | | | | |
| 0x0D81 | INDF1 | 7:0 | INDF1[7:0] | | | | | | | |
| 0x0D82 | PCL | 7:0 | PCL[7:0] | | | | | | | |
| 0x0D83 | STATUS | 7:0 | | | | TO | PD | Z | DC | C |
| 0x0D84 | FSR0 | 7:0 | FSR0[7:0] | | | | | | | |
| | | 15:8 | FSR0[15:8] | | | | | | | |
| 0x0D86 | FSR1 | 7:0 | FSR1[7:0] | | | | | | | |
| | | 15:8 | FSR1[15:8] | | | | | | | |
| 0x0D88 | BSR | 7:0 | BSR[5:0] | | | | | | | |
| 0x0D89 | WREG | 7:0 | WREG[7:0] | | | | | | | |
| 0x0D8A | PCLATH | 7:0 | PCLATH[6:0] | | | | | | | |
| 0x0D8B | INTCON | 7:0 | GIE | PEIE | | | | | | INTEDG |
| 0x0D8C | Reserved | | | | | | | | | |
| 0x0DFF | | | | | | | | | | |
| 0x0E00 | INDF0 | 7:0 | INDF0[7:0] | | | | | | | |
| 0x0E01 | INDF1 | 7:0 | INDF1[7:0] | | | | | | | |
| 0x0E02 | PCL | 7:0 | PCL[7:0] | | | | | | | |

PIC16(L)F18426/46

Register Summary

.....continued

| Address | Name | Bit Pos. | | | | | | | | |
|---------|----------|----------|-------------|------|--|----|----|---|----------|--------|
| 0x0E03 | STATUS | 7:0 | | | | T0 | PD | Z | DC | C |
| 0x0E04 | FSR0 | 7:0 | FSR0[7:0] | | | | | | | |
| | | 15:8 | FSR0[15:8] | | | | | | | |
| 0x0E06 | FSR1 | 7:0 | FSR1[7:0] | | | | | | | |
| | | 15:8 | FSR1[15:8] | | | | | | | |
| 0x0E08 | BSR | 7:0 | | | | | | | BSR[5:0] | |
| 0x0E09 | WREG | 7:0 | WREG[7:0] | | | | | | | |
| 0x0E0A | PCLATH | 7:0 | PCLATH[6:0] | | | | | | | |
| 0x0E0B | INTCON | 7:0 | GIE | PEIE | | | | | | INTEDG |
| 0x0E0C | Reserved | | | | | | | | | |
| 0x0E7F | | | | | | | | | | |
| 0x0E80 | INDF0 | 7:0 | INDF0[7:0] | | | | | | | |
| 0x0E81 | INDF1 | 7:0 | INDF1[7:0] | | | | | | | |
| 0x0E82 | PCL | 7:0 | PCL[7:0] | | | | | | | |
| 0x0E83 | STATUS | 7:0 | | | | T0 | PD | Z | DC | C |
| 0x0E84 | FSR0 | 7:0 | FSR0[7:0] | | | | | | | |
| | | 15:8 | FSR0[15:8] | | | | | | | |
| 0x0E86 | FSR1 | 7:0 | FSR1[7:0] | | | | | | | |
| | | 15:8 | FSR1[15:8] | | | | | | | |
| 0x0E88 | BSR | 7:0 | | | | | | | BSR[5:0] | |
| 0x0E89 | WREG | 7:0 | WREG[7:0] | | | | | | | |
| 0x0E8A | PCLATH | 7:0 | PCLATH[6:0] | | | | | | | |
| 0x0E8B | INTCON | 7:0 | GIE | PEIE | | | | | | INTEDG |
| 0x0E8C | Reserved | | | | | | | | | |
| 0x0EFF | | | | | | | | | | |
| 0x0F00 | INDF0 | 7:0 | INDF0[7:0] | | | | | | | |
| 0x0F01 | INDF1 | 7:0 | INDF1[7:0] | | | | | | | |
| 0x0F02 | PCL | 7:0 | PCL[7:0] | | | | | | | |
| 0x0F03 | STATUS | 7:0 | | | | T0 | PD | Z | DC | C |
| 0x0F04 | FSR0 | 7:0 | FSR0[7:0] | | | | | | | |
| | | 15:8 | FSR0[15:8] | | | | | | | |
| 0x0F06 | FSR1 | 7:0 | FSR1[7:0] | | | | | | | |
| | | 15:8 | FSR1[15:8] | | | | | | | |
| 0x0F08 | BSR | 7:0 | | | | | | | BSR[5:0] | |
| 0x0F09 | WREG | 7:0 | WREG[7:0] | | | | | | | |
| 0x0F0A | PCLATH | 7:0 | PCLATH[6:0] | | | | | | | |
| 0x0F0B | INTCON | 7:0 | GIE | PEIE | | | | | | INTEDG |
| 0x0F0C | Reserved | | | | | | | | | |
| 0x0F7F | | | | | | | | | | |
| 0x0F80 | INDF0 | 7:0 | INDF0[7:0] | | | | | | | |
| 0x0F81 | INDF1 | 7:0 | INDF1[7:0] | | | | | | | |
| 0x0F82 | PCL | 7:0 | PCL[7:0] | | | | | | | |
| 0x0F83 | STATUS | 7:0 | | | | T0 | PD | Z | DC | C |
| 0x0F84 | FSR0 | 7:0 | FSR0[7:0] | | | | | | | |
| | | 15:8 | FSR0[15:8] | | | | | | | |
| 0x0F86 | FSR1 | 7:0 | FSR1[7:0] | | | | | | | |
| | | 15:8 | FSR1[15:8] | | | | | | | |
| 0x0F88 | BSR | 7:0 | | | | | | | BSR[5:0] | |
| 0x0F89 | WREG | 7:0 | WREG[7:0] | | | | | | | |
| 0x0F8A | PCLATH | 7:0 | PCLATH[6:0] | | | | | | | |
| 0x0F8B | INTCON | 7:0 | GIE | PEIE | | | | | | INTEDG |
| 0x0F8C | Reserved | | | | | | | | | |
| 0x0FEC | | | | | | | | | | |
| 0x0FED | STKPTR | 7:0 | STKPTR[4:0] | | | | | | | |

PIC16(L)F18426/46

Register Summary

|continued | | | | | | | | | |
|----------------|----------|----------|-------------|------|----|----|----------|----|--------|
| Address | Name | Bit Pos. | | | | | | | |
| 0x0FEE | TOS | 7:0 | TOS[7:0] | | | | | | |
| | | 15:8 | TOS[15:8] | | | | | | |
| 0x0FF0 | Reserved | | | | | | | | |
| 0x0FFF | | | | | | | | | |
| 0x1000 | INDF0 | 7:0 | INDF0[7:0] | | | | | | |
| 0x1001 | INDF1 | 7:0 | INDF1[7:0] | | | | | | |
| 0x1002 | PCL | 7:0 | PCL[7:0] | | | | | | |
| 0x1003 | STATUS | 7:0 | | | TO | PD | Z | DC | C |
| 0x1004 | FSR0 | 7:0 | FSR0[7:0] | | | | | | |
| | | 15:8 | FSR0[15:8] | | | | | | |
| 0x1006 | FSR1 | 7:0 | FSR1[7:0] | | | | | | |
| | | 15:8 | FSR1[15:8] | | | | | | |
| 0x1008 | BSR | 7:0 | | | | | BSR[5:0] | | |
| 0x1009 | WREG | 7:0 | WREG[7:0] | | | | | | |
| 0x100A | PCLATH | 7:0 | PCLATH[6:0] | | | | | | |
| 0x100B | INTCON | 7:0 | GIE | PEIE | | | | | INTEDG |
| 0x100C | Reserved | | | | | | | | |
| 0x107F | | | | | | | | | |
| 0x1080 | INDF0 | 7:0 | INDF0[7:0] | | | | | | |
| 0x1081 | INDF1 | 7:0 | INDF1[7:0] | | | | | | |
| 0x1082 | PCL | 7:0 | PCL[7:0] | | | | | | |
| 0x1083 | STATUS | 7:0 | | | TO | PD | Z | DC | C |
| 0x1084 | FSR0 | 7:0 | FSR0[7:0] | | | | | | |
| | | 15:8 | FSR0[15:8] | | | | | | |
| 0x1086 | FSR1 | 7:0 | FSR1[7:0] | | | | | | |
| | | 15:8 | FSR1[15:8] | | | | | | |
| 0x1088 | BSR | 7:0 | | | | | BSR[5:0] | | |
| 0x1089 | WREG | 7:0 | WREG[7:0] | | | | | | |
| 0x108A | PCLATH | 7:0 | PCLATH[6:0] | | | | | | |
| 0x108B | INTCON | 7:0 | GIE | PEIE | | | | | INTEDG |
| 0x108C | Reserved | | | | | | | | |
| 0x10FF | | | | | | | | | |
| 0x1100 | INDF0 | 7:0 | INDF0[7:0] | | | | | | |
| 0x1101 | INDF1 | 7:0 | INDF1[7:0] | | | | | | |
| 0x1102 | PCL | 7:0 | PCL[7:0] | | | | | | |
| 0x1103 | STATUS | 7:0 | | | TO | PD | Z | DC | C |
| 0x1104 | FSR0 | 7:0 | FSR0[7:0] | | | | | | |
| | | 15:8 | FSR0[15:8] | | | | | | |
| 0x1106 | FSR1 | 7:0 | FSR1[7:0] | | | | | | |
| | | 15:8 | FSR1[15:8] | | | | | | |
| 0x1108 | BSR | 7:0 | | | | | BSR[5:0] | | |
| 0x1109 | WREG | 7:0 | WREG[7:0] | | | | | | |
| 0x110A | PCLATH | 7:0 | PCLATH[6:0] | | | | | | |
| 0x110B | INTCON | 7:0 | GIE | PEIE | | | | | INTEDG |
| 0x110C | Reserved | | | | | | | | |
| 0x117F | | | | | | | | | |
| 0x1180 | INDF0 | 7:0 | INDF0[7:0] | | | | | | |
| 0x1181 | INDF1 | 7:0 | INDF1[7:0] | | | | | | |
| 0x1182 | PCL | 7:0 | PCL[7:0] | | | | | | |
| 0x1183 | STATUS | 7:0 | | | TO | PD | Z | DC | C |
| 0x1184 | FSR0 | 7:0 | FSR0[7:0] | | | | | | |
| | | 15:8 | FSR0[15:8] | | | | | | |
| 0x1186 | FSR1 | 7:0 | FSR1[7:0] | | | | | | |
| | | 15:8 | FSR1[15:8] | | | | | | |
| 0x1188 | BSR | 7:0 | | | | | BSR[5:0] | | |

PIC16(L)F18426/46

Register Summary

|continued | | | | | | | | | | |
|----------------|----------|----------|-------------|------|--|----|----|---|--------|--|
| Address | Name | Bit Pos. | | | | | | | | |
| 0x1189 | WREG | 7:0 | WREG[7:0] | | | | | | | |
| 0x118A | PCLATH | 7:0 | PCLATH[6:0] | | | | | | | |
| 0x118B | INTCON | 7:0 | GIE | PEIE | | | | | INTEDG | |
| 0x118C | Reserved | | | | | | | | | |
| ... 0x11FF | | | | | | | | | | |
| 0x1200 | INDF0 | 7:0 | INDF0[7:0] | | | | | | | |
| 0x1201 | INDF1 | 7:0 | INDF1[7:0] | | | | | | | |
| 0x1202 | PCL | 7:0 | PCL[7:0] | | | | | | | |
| 0x1203 | STATUS | 7:0 | | | | TO | PD | Z | DC C | |
| 0x1204 | FSR0 | 7:0 | FSR0[7:0] | | | | | | | |
| | | 15:8 | FSR0[15:8] | | | | | | | |
| 0x1206 | FSR1 | 7:0 | FSR1[7:0] | | | | | | | |
| | | 15:8 | FSR1[15:8] | | | | | | | |
| 0x1208 | BSR | 7:0 | BSR[5:0] | | | | | | | |
| 0x1209 | WREG | 7:0 | WREG[7:0] | | | | | | | |
| 0x120A | PCLATH | 7:0 | PCLATH[6:0] | | | | | | | |
| 0x120B | INTCON | 7:0 | GIE | PEIE | | | | | INTEDG | |
| 0x120C | Reserved | | | | | | | | | |
| ... 0x127F | | | | | | | | | | |
| 0x1280 | INDF0 | 7:0 | INDF0[7:0] | | | | | | | |
| 0x1281 | INDF1 | 7:0 | INDF1[7:0] | | | | | | | |
| 0x1282 | PCL | 7:0 | PCL[7:0] | | | | | | | |
| 0x1283 | STATUS | 7:0 | | | | TO | PD | Z | DC C | |
| 0x1284 | FSR0 | 7:0 | FSR0[7:0] | | | | | | | |
| | | 15:8 | FSR0[15:8] | | | | | | | |
| 0x1286 | FSR1 | 7:0 | FSR1[7:0] | | | | | | | |
| | | 15:8 | FSR1[15:8] | | | | | | | |
| 0x1288 | BSR | 7:0 | BSR[5:0] | | | | | | | |
| 0x1289 | WREG | 7:0 | WREG[7:0] | | | | | | | |
| 0x128A | PCLATH | 7:0 | PCLATH[6:0] | | | | | | | |
| 0x128B | INTCON | 7:0 | GIE | PEIE | | | | | INTEDG | |
| 0x128C | Reserved | | | | | | | | | |
| ... 0x12FF | | | | | | | | | | |
| 0x1300 | INDF0 | 7:0 | INDF0[7:0] | | | | | | | |
| 0x1301 | INDF1 | 7:0 | INDF1[7:0] | | | | | | | |
| 0x1302 | PCL | 7:0 | PCL[7:0] | | | | | | | |
| 0x1303 | STATUS | 7:0 | | | | TO | PD | Z | DC C | |
| 0x1304 | FSR0 | 7:0 | FSR0[7:0] | | | | | | | |
| | | 15:8 | FSR0[15:8] | | | | | | | |
| 0x1306 | FSR1 | 7:0 | FSR1[7:0] | | | | | | | |
| | | 15:8 | FSR1[15:8] | | | | | | | |
| 0x1308 | BSR | 7:0 | BSR[5:0] | | | | | | | |
| 0x1309 | WREG | 7:0 | WREG[7:0] | | | | | | | |
| 0x130A | PCLATH | 7:0 | PCLATH[6:0] | | | | | | | |
| 0x130B | INTCON | 7:0 | GIE | PEIE | | | | | INTEDG | |
| 0x130C | Reserved | | | | | | | | | |
| ... 0x137F | | | | | | | | | | |
| 0x1380 | INDF0 | 7:0 | INDF0[7:0] | | | | | | | |
| 0x1381 | INDF1 | 7:0 | INDF1[7:0] | | | | | | | |
| 0x1382 | PCL | 7:0 | PCL[7:0] | | | | | | | |
| 0x1383 | STATUS | 7:0 | | | | TO | PD | Z | DC C | |
| 0x1384 | FSR0 | 7:0 | FSR0[7:0] | | | | | | | |
| | | 15:8 | FSR0[15:8] | | | | | | | |
| 0x1386 | FSR1 | 7:0 | FSR1[7:0] | | | | | | | |
| | | 15:8 | FSR1[15:8] | | | | | | | |

PIC16(L)F18426/46

Register Summary

.....continued

| Address | Name | Bit Pos. | | | | | | | | |
|---------|----------|----------|-----|------|--|----|----|---|----|-------------|
| 0x1388 | BSR | 7:0 | | | | | | | | BSR[5:0] |
| 0x1389 | WREG | 7:0 | | | | | | | | WREG[7:0] |
| 0x138A | PCLATH | 7:0 | | | | | | | | PCLATH[6:0] |
| 0x138B | INTCON | 7:0 | GIE | PEIE | | | | | | INTEDG |
| 0x138C | Reserved | | | | | | | | | |
| ... | | | | | | | | | | |
| 0x13FF | | | | | | | | | | |
| 0x1400 | INDF0 | 7:0 | | | | | | | | INDF0[7:0] |
| 0x1401 | INDF1 | 7:0 | | | | | | | | INDF1[7:0] |
| 0x1402 | PCL | 7:0 | | | | | | | | PCL[7:0] |
| 0x1403 | STATUS | 7:0 | | | | TO | PD | Z | DC | C |
| 0x1404 | FSR0 | 7:0 | | | | | | | | FSR0[7:0] |
| | | 15:8 | | | | | | | | FSR0[15:8] |
| 0x1406 | FSR1 | 7:0 | | | | | | | | FSR1[7:0] |
| | | 15:8 | | | | | | | | FSR1[15:8] |
| 0x1408 | BSR | 7:0 | | | | | | | | BSR[5:0] |
| 0x1409 | WREG | 7:0 | | | | | | | | WREG[7:0] |
| 0x140A | PCLATH | 7:0 | | | | | | | | PCLATH[6:0] |
| 0x140B | INTCON | 7:0 | GIE | PEIE | | | | | | INTEDG |
| 0x140C | Reserved | | | | | | | | | |
| ... | | | | | | | | | | |
| 0x147F | | | | | | | | | | |
| 0x1480 | INDF0 | 7:0 | | | | | | | | INDF0[7:0] |
| 0x1481 | INDF1 | 7:0 | | | | | | | | INDF1[7:0] |
| 0x1482 | PCL | 7:0 | | | | | | | | PCL[7:0] |
| 0x1483 | STATUS | 7:0 | | | | TO | PD | Z | DC | C |
| 0x1484 | FSR0 | 7:0 | | | | | | | | FSR0[7:0] |
| | | 15:8 | | | | | | | | FSR0[15:8] |
| 0x1486 | FSR1 | 7:0 | | | | | | | | FSR1[7:0] |
| | | 15:8 | | | | | | | | FSR1[15:8] |
| 0x1488 | BSR | 7:0 | | | | | | | | BSR[5:0] |
| 0x1489 | WREG | 7:0 | | | | | | | | WREG[7:0] |
| 0x148A | PCLATH | 7:0 | | | | | | | | PCLATH[6:0] |
| 0x148B | INTCON | 7:0 | GIE | PEIE | | | | | | INTEDG |
| 0x148C | Reserved | | | | | | | | | |
| ... | | | | | | | | | | |
| 0x14FF | | | | | | | | | | |
| 0x1500 | INDF0 | 7:0 | | | | | | | | INDF0[7:0] |
| 0x1501 | INDF1 | 7:0 | | | | | | | | INDF1[7:0] |
| 0x1502 | PCL | 7:0 | | | | | | | | PCL[7:0] |
| 0x1503 | STATUS | 7:0 | | | | TO | PD | Z | DC | C |
| 0x1504 | FSR0 | 7:0 | | | | | | | | FSR0[7:0] |
| | | 15:8 | | | | | | | | FSR0[15:8] |
| 0x1506 | FSR1 | 7:0 | | | | | | | | FSR1[7:0] |
| | | 15:8 | | | | | | | | FSR1[15:8] |
| 0x1508 | BSR | 7:0 | | | | | | | | BSR[5:0] |
| 0x1509 | WREG | 7:0 | | | | | | | | WREG[7:0] |
| 0x150A | PCLATH | 7:0 | | | | | | | | PCLATH[6:0] |
| 0x150B | INTCON | 7:0 | GIE | PEIE | | | | | | INTEDG |
| 0x150C | Reserved | | | | | | | | | |
| ... | | | | | | | | | | |
| 0x157F | | | | | | | | | | |
| 0x1580 | INDF0 | 7:0 | | | | | | | | INDF0[7:0] |
| 0x1581 | INDF1 | 7:0 | | | | | | | | INDF1[7:0] |
| 0x1582 | PCL | 7:0 | | | | | | | | PCL[7:0] |
| 0x1583 | STATUS | 7:0 | | | | TO | PD | Z | DC | C |
| 0x1584 | FSR0 | 7:0 | | | | | | | | FSR0[7:0] |
| | | 15:8 | | | | | | | | FSR0[15:8] |

PIC16(L)F18426/46

Register Summary

.....continued

| Address | Name | Bit Pos. | | | | | | | | |
|---------|----------|----------|-------------|------|----|----|---|----|---|--------|
| 0x1586 | FSR1 | 7:0 | FSR1[7:0] | | | | | | | |
| | | 15:8 | FSR1[15:8] | | | | | | | |
| 0x1588 | BSR | 7:0 | BSR[5:0] | | | | | | | |
| 0x1589 | WREG | 7:0 | WREG[7:0] | | | | | | | |
| 0x158A | PCLATH | 7:0 | PCLATH[6:0] | | | | | | | |
| 0x158B | INTCON | 7:0 | GIE | PEIE | | | | | | INTEDG |
| 0x158C | Reserved | | | | | | | | | |
| ... | | | | | | | | | | |
| 0x15FF | | | | | | | | | | |
| 0x1600 | INDF0 | 7:0 | INDF0[7:0] | | | | | | | |
| 0x1601 | INDF1 | 7:0 | INDF1[7:0] | | | | | | | |
| 0x1602 | PCL | 7:0 | PCL[7:0] | | | | | | | |
| 0x1603 | STATUS | 7:0 | | | TO | PD | Z | DC | C | |
| 0x1604 | FSR0 | 7:0 | FSR0[7:0] | | | | | | | |
| | | 15:8 | FSR0[15:8] | | | | | | | |
| 0x1606 | FSR1 | 7:0 | FSR1[7:0] | | | | | | | |
| | | 15:8 | FSR1[15:8] | | | | | | | |
| 0x1608 | BSR | 7:0 | BSR[5:0] | | | | | | | |
| 0x1609 | WREG | 7:0 | WREG[7:0] | | | | | | | |
| 0x160A | PCLATH | 7:0 | PCLATH[6:0] | | | | | | | |
| 0x160B | INTCON | 7:0 | GIE | PEIE | | | | | | INTEDG |
| 0x160C | Reserved | | | | | | | | | |
| ... | | | | | | | | | | |
| 0x167F | | | | | | | | | | |
| 0x1680 | INDF0 | 7:0 | INDF0[7:0] | | | | | | | |
| 0x1681 | INDF1 | 7:0 | INDF1[7:0] | | | | | | | |
| 0x1682 | PCL | 7:0 | PCL[7:0] | | | | | | | |
| 0x1683 | STATUS | 7:0 | | | TO | PD | Z | DC | C | |
| 0x1684 | FSR0 | 7:0 | FSR0[7:0] | | | | | | | |
| | | 15:8 | FSR0[15:8] | | | | | | | |
| 0x1686 | FSR1 | 7:0 | FSR1[7:0] | | | | | | | |
| | | 15:8 | FSR1[15:8] | | | | | | | |
| 0x1688 | BSR | 7:0 | BSR[5:0] | | | | | | | |
| 0x1689 | WREG | 7:0 | WREG[7:0] | | | | | | | |
| 0x168A | PCLATH | 7:0 | PCLATH[6:0] | | | | | | | |
| 0x168B | INTCON | 7:0 | GIE | PEIE | | | | | | INTEDG |
| 0x168C | Reserved | | | | | | | | | |
| ... | | | | | | | | | | |
| 0x16FF | | | | | | | | | | |
| 0x1700 | INDF0 | 7:0 | INDF0[7:0] | | | | | | | |
| 0x1701 | INDF1 | 7:0 | INDF1[7:0] | | | | | | | |
| 0x1702 | PCL | 7:0 | PCL[7:0] | | | | | | | |
| 0x1703 | STATUS | 7:0 | | | TO | PD | Z | DC | C | |
| 0x1704 | FSR0 | 7:0 | FSR0[7:0] | | | | | | | |
| | | 15:8 | FSR0[15:8] | | | | | | | |
| 0x1706 | FSR1 | 7:0 | FSR1[7:0] | | | | | | | |
| | | 15:8 | FSR1[15:8] | | | | | | | |
| 0x1708 | BSR | 7:0 | BSR[5:0] | | | | | | | |
| 0x1709 | WREG | 7:0 | WREG[7:0] | | | | | | | |
| 0x170A | PCLATH | 7:0 | PCLATH[6:0] | | | | | | | |
| 0x170B | INTCON | 7:0 | GIE | PEIE | | | | | | INTEDG |
| 0x170C | Reserved | | | | | | | | | |
| ... | | | | | | | | | | |
| 0x177F | | | | | | | | | | |
| 0x1780 | INDF0 | 7:0 | INDF0[7:0] | | | | | | | |
| 0x1781 | INDF1 | 7:0 | INDF1[7:0] | | | | | | | |
| 0x1782 | PCL | 7:0 | PCL[7:0] | | | | | | | |
| 0x1783 | STATUS | 7:0 | | | TO | PD | Z | DC | C | |

PIC16(L)F18426/46

Register Summary

.....continued

| Address | Name | Bit Pos. | | | | | | | | | |
|---------------|----------|----------|-------------|------|--|----|----|---|----|---|--------|
| 0x1784 | FSR0 | 7:0 | FSR0[7:0] | | | | | | | | |
| | | 15:8 | FSR0[15:8] | | | | | | | | |
| 0x1786 | FSR1 | 7:0 | FSR1[7:0] | | | | | | | | |
| | | 15:8 | FSR1[15:8] | | | | | | | | |
| 0x1788 | BSR | 7:0 | BSR[5:0] | | | | | | | | |
| 0x1789 | WREG | 7:0 | WREG[7:0] | | | | | | | | |
| 0x178A | PCLATH | 7:0 | PCLATH[6:0] | | | | | | | | |
| 0x178B | INTCON | 7:0 | GIE | PEIE | | | | | | | INTEDG |
| 0x178C | Reserved | | | | | | | | | | |
| ... 0x17FF | | | | | | | | | | | |
| 0x1800 | INDF0 | 7:0 | INDF0[7:0] | | | | | | | | |
| 0x1801 | INDF1 | 7:0 | INDF1[7:0] | | | | | | | | |
| 0x1802 | PCL | 7:0 | PCL[7:0] | | | | | | | | |
| 0x1803 | STATUS | 7:0 | | | | TO | PD | Z | DC | C | |
| 0x1804 | FSR0 | 7:0 | FSR0[7:0] | | | | | | | | |
| | | 15:8 | FSR0[15:8] | | | | | | | | |
| 0x1806 | FSR1 | 7:0 | FSR1[7:0] | | | | | | | | |
| | | 15:8 | FSR1[15:8] | | | | | | | | |
| 0x1808 | BSR | 7:0 | BSR[5:0] | | | | | | | | |
| 0x1809 | WREG | 7:0 | WREG[7:0] | | | | | | | | |
| 0x180A | PCLATH | 7:0 | PCLATH[6:0] | | | | | | | | |
| 0x180B | INTCON | 7:0 | GIE | PEIE | | | | | | | INTEDG |
| 0x180C | Reserved | | | | | | | | | | |
| ... 0x187F | | | | | | | | | | | |
| 0x1880 | INDF0 | 7:0 | INDF0[7:0] | | | | | | | | |
| 0x1881 | INDF1 | 7:0 | INDF1[7:0] | | | | | | | | |
| 0x1882 | PCL | 7:0 | PCL[7:0] | | | | | | | | |
| 0x1883 | STATUS | 7:0 | | | | TO | PD | Z | DC | C | |
| 0x1884 | FSR0 | 7:0 | FSR0[7:0] | | | | | | | | |
| | | 15:8 | FSR0[15:8] | | | | | | | | |
| 0x1886 | FSR1 | 7:0 | FSR1[7:0] | | | | | | | | |
| | | 15:8 | FSR1[15:8] | | | | | | | | |
| 0x1888 | BSR | 7:0 | BSR[5:0] | | | | | | | | |
| 0x1889 | WREG | 7:0 | WREG[7:0] | | | | | | | | |
| 0x188A | PCLATH | 7:0 | PCLATH[6:0] | | | | | | | | |
| 0x188B | INTCON | 7:0 | GIE | PEIE | | | | | | | INTEDG |
| 0x188C | Reserved | | | | | | | | | | |
| ... 0x18FF | | | | | | | | | | | |
| 0x1900 | INDF0 | 7:0 | INDF0[7:0] | | | | | | | | |
| 0x1901 | INDF1 | 7:0 | INDF1[7:0] | | | | | | | | |
| 0x1902 | PCL | 7:0 | PCL[7:0] | | | | | | | | |
| 0x1903 | STATUS | 7:0 | | | | TO | PD | Z | DC | C | |
| 0x1904 | FSR0 | 7:0 | FSR0[7:0] | | | | | | | | |
| | | 15:8 | FSR0[15:8] | | | | | | | | |
| 0x1906 | FSR1 | 7:0 | FSR1[7:0] | | | | | | | | |
| | | 15:8 | FSR1[15:8] | | | | | | | | |
| 0x1908 | BSR | 7:0 | BSR[5:0] | | | | | | | | |
| 0x1909 | WREG | 7:0 | WREG[7:0] | | | | | | | | |
| 0x190A | PCLATH | 7:0 | PCLATH[6:0] | | | | | | | | |
| 0x190B | INTCON | 7:0 | GIE | PEIE | | | | | | | INTEDG |
| 0x190C | Reserved | | | | | | | | | | |
| ... 0x197F | | | | | | | | | | | |
| 0x1980 | INDF0 | 7:0 | INDF0[7:0] | | | | | | | | |
| 0x1981 | INDF1 | 7:0 | INDF1[7:0] | | | | | | | | |
| 0x1982 | PCL | 7:0 | PCL[7:0] | | | | | | | | |

PIC16(L)F18426/46

Register Summary

|continued | | | | | | | | | | |
|----------------|----------|----------|-------------|------|--|----|----|---|----------|--------|
| Address | Name | Bit Pos. | | | | | | | | |
| 0x1983 | STATUS | 7:0 | | | | T0 | PD | Z | DC | C |
| 0x1984 | FSR0 | 7:0 | FSR0[7:0] | | | | | | | |
| | | 15:8 | FSR0[15:8] | | | | | | | |
| 0x1986 | FSR1 | 7:0 | FSR1[7:0] | | | | | | | |
| | | 15:8 | FSR1[15:8] | | | | | | | |
| 0x1988 | BSR | 7:0 | | | | | | | BSR[5:0] | |
| 0x1989 | WREG | 7:0 | WREG[7:0] | | | | | | | |
| 0x198A | PCLATH | 7:0 | PCLATH[6:0] | | | | | | | |
| 0x198B | INTCON | 7:0 | GIE | PEIE | | | | | | INTEDG |
| 0x198C | Reserved | | | | | | | | | |
| 0x19FF | | | | | | | | | | |
| 0x1A00 | INDF0 | 7:0 | INDF0[7:0] | | | | | | | |
| 0x1A01 | INDF1 | 7:0 | INDF1[7:0] | | | | | | | |
| 0x1A02 | PCL | 7:0 | PCL[7:0] | | | | | | | |
| 0x1A03 | STATUS | 7:0 | | | | T0 | PD | Z | DC | C |
| 0x1A04 | FSR0 | 7:0 | FSR0[7:0] | | | | | | | |
| | | 15:8 | FSR0[15:8] | | | | | | | |
| 0x1A06 | FSR1 | 7:0 | FSR1[7:0] | | | | | | | |
| | | 15:8 | FSR1[15:8] | | | | | | | |
| 0x1A08 | BSR | 7:0 | | | | | | | BSR[5:0] | |
| 0x1A09 | WREG | 7:0 | WREG[7:0] | | | | | | | |
| 0x1A0A | PCLATH | 7:0 | PCLATH[6:0] | | | | | | | |
| 0x1A0B | INTCON | 7:0 | GIE | PEIE | | | | | | INTEDG |
| 0x1A0C | Reserved | | | | | | | | | |
| 0x1A7F | | | | | | | | | | |
| 0x1A80 | INDF0 | 7:0 | INDF0[7:0] | | | | | | | |
| 0x1A81 | INDF1 | 7:0 | INDF1[7:0] | | | | | | | |
| 0x1A82 | PCL | 7:0 | PCL[7:0] | | | | | | | |
| 0x1A83 | STATUS | 7:0 | | | | T0 | PD | Z | DC | C |
| 0x1A84 | FSR0 | 7:0 | FSR0[7:0] | | | | | | | |
| | | 15:8 | FSR0[15:8] | | | | | | | |
| 0x1A86 | FSR1 | 7:0 | FSR1[7:0] | | | | | | | |
| | | 15:8 | FSR1[15:8] | | | | | | | |
| 0x1A88 | BSR | 7:0 | | | | | | | BSR[5:0] | |
| 0x1A89 | WREG | 7:0 | WREG[7:0] | | | | | | | |
| 0x1A8A | PCLATH | 7:0 | PCLATH[6:0] | | | | | | | |
| 0x1A8B | INTCON | 7:0 | GIE | PEIE | | | | | | INTEDG |
| 0x1A8C | Reserved | | | | | | | | | |
| 0x1AFF | | | | | | | | | | |
| 0x1B00 | INDF0 | 7:0 | INDF0[7:0] | | | | | | | |
| 0x1B01 | INDF1 | 7:0 | INDF1[7:0] | | | | | | | |
| 0x1B02 | PCL | 7:0 | PCL[7:0] | | | | | | | |
| 0x1B03 | STATUS | 7:0 | | | | T0 | PD | Z | DC | C |
| 0x1B04 | FSR0 | 7:0 | FSR0[7:0] | | | | | | | |
| | | 15:8 | FSR0[15:8] | | | | | | | |
| 0x1B06 | FSR1 | 7:0 | FSR1[7:0] | | | | | | | |
| | | 15:8 | FSR1[15:8] | | | | | | | |
| 0x1B08 | BSR | 7:0 | | | | | | | BSR[5:0] | |
| 0x1B09 | WREG | 7:0 | WREG[7:0] | | | | | | | |
| 0x1B0A | PCLATH | 7:0 | PCLATH[6:0] | | | | | | | |
| 0x1B0B | INTCON | 7:0 | GIE | PEIE | | | | | | INTEDG |
| 0x1B0C | Reserved | | | | | | | | | |
| 0x1B7F | | | | | | | | | | |
| 0x1B80 | INDF0 | 7:0 | INDF0[7:0] | | | | | | | |
| 0x1B81 | INDF1 | 7:0 | INDF1[7:0] | | | | | | | |

PIC16(L)F18426/46

Register Summary

.....continued

| Address | Name | Bit Pos. | | | | | | | | |
|---------|----------|----------|-------------|------|--|----|----|---|----|--------|
| 0x1B82 | PCL | 7:0 | PCL[7:0] | | | | | | | |
| 0x1B83 | STATUS | 7:0 | | | | TO | PD | Z | DC | C |
| 0x1B84 | FSR0 | 7:0 | FSR0[7:0] | | | | | | | |
| | | 15:8 | FSR0[15:8] | | | | | | | |
| 0x1B86 | FSR1 | 7:0 | FSR1[7:0] | | | | | | | |
| | | 15:8 | FSR1[15:8] | | | | | | | |
| 0x1B88 | BSR | 7:0 | BSR[5:0] | | | | | | | |
| 0x1B89 | WREG | 7:0 | WREG[7:0] | | | | | | | |
| 0x1B8A | PCLATH | 7:0 | PCLATH[6:0] | | | | | | | |
| 0x1B8B | INTCON | 7:0 | GIE | PEIE | | | | | | INTEDG |
| 0x1B8C | Reserved | | | | | | | | | |
| 0x1BFF | | | | | | | | | | |
| 0x1C00 | INDF0 | 7:0 | INDF0[7:0] | | | | | | | |
| 0x1C01 | INDF1 | 7:0 | INDF1[7:0] | | | | | | | |
| 0x1C02 | PCL | 7:0 | PCL[7:0] | | | | | | | |
| 0x1C03 | STATUS | 7:0 | | | | TO | PD | Z | DC | C |
| 0x1C04 | FSR0 | 7:0 | FSR0[7:0] | | | | | | | |
| | | 15:8 | FSR0[15:8] | | | | | | | |
| 0x1C06 | FSR1 | 7:0 | FSR1[7:0] | | | | | | | |
| | | 15:8 | FSR1[15:8] | | | | | | | |
| 0x1C08 | BSR | 7:0 | BSR[5:0] | | | | | | | |
| 0x1C09 | WREG | 7:0 | WREG[7:0] | | | | | | | |
| 0x1C0A | PCLATH | 7:0 | PCLATH[6:0] | | | | | | | |
| 0x1C0B | INTCON | 7:0 | GIE | PEIE | | | | | | INTEDG |
| 0x1C0C | Reserved | | | | | | | | | |
| 0x1C7F | | | | | | | | | | |
| 0x1C80 | INDF0 | 7:0 | INDF0[7:0] | | | | | | | |
| 0x1C81 | INDF1 | 7:0 | INDF1[7:0] | | | | | | | |
| 0x1C82 | PCL | 7:0 | PCL[7:0] | | | | | | | |
| 0x1C83 | STATUS | 7:0 | | | | TO | PD | Z | DC | C |
| 0x1C84 | FSR0 | 7:0 | FSR0[7:0] | | | | | | | |
| | | 15:8 | FSR0[15:8] | | | | | | | |
| 0x1C86 | FSR1 | 7:0 | FSR1[7:0] | | | | | | | |
| | | 15:8 | FSR1[15:8] | | | | | | | |
| 0x1C88 | BSR | 7:0 | BSR[5:0] | | | | | | | |
| 0x1C89 | WREG | 7:0 | WREG[7:0] | | | | | | | |
| 0x1C8A | PCLATH | 7:0 | PCLATH[6:0] | | | | | | | |
| 0x1C8B | INTCON | 7:0 | GIE | PEIE | | | | | | INTEDG |
| 0x1C8C | Reserved | | | | | | | | | |
| 0x1CFF | | | | | | | | | | |
| 0x1D00 | INDF0 | 7:0 | INDF0[7:0] | | | | | | | |
| 0x1D01 | INDF1 | 7:0 | INDF1[7:0] | | | | | | | |
| 0x1D02 | PCL | 7:0 | PCL[7:0] | | | | | | | |
| 0x1D03 | STATUS | 7:0 | | | | TO | PD | Z | DC | C |
| 0x1D04 | FSR0 | 7:0 | FSR0[7:0] | | | | | | | |
| | | 15:8 | FSR0[15:8] | | | | | | | |
| 0x1D06 | FSR1 | 7:0 | FSR1[7:0] | | | | | | | |
| | | 15:8 | FSR1[15:8] | | | | | | | |
| 0x1D08 | BSR | 7:0 | BSR[5:0] | | | | | | | |
| 0x1D09 | WREG | 7:0 | WREG[7:0] | | | | | | | |
| 0x1D0A | PCLATH | 7:0 | PCLATH[6:0] | | | | | | | |
| 0x1D0B | INTCON | 7:0 | GIE | PEIE | | | | | | INTEDG |
| 0x1D0C | Reserved | | | | | | | | | |
| 0x1D7F | | | | | | | | | | |
| 0x1D80 | INDF0 | 7:0 | INDF0[7:0] | | | | | | | |

PIC16(L)F18426/46

Register Summary

.....continued

| Address | Name | Bit Pos. | | | | | | | | | | | |
|---------|----------|----------|-------|-------|-------|-------|-------|---------|---------|---------|--|-----------|-------------|
| 0x1D81 | INDF1 | 7:0 | | | | | | | | | | | INDF1[7:0] |
| 0x1D82 | PCL | 7:0 | | | | | | | | | | | PCL[7:0] |
| 0x1D83 | STATUS | 7:0 | | | | | TO | PD | Z | DC | | | C |
| 0x1D84 | FSR0 | 7:0 | | | | | | | | | | | FSR0[7:0] |
| | | 15:8 | | | | | | | | | | | FSR0[15:8] |
| 0x1D86 | FSR1 | 7:0 | | | | | | | | | | | FSR1[7:0] |
| | | 15:8 | | | | | | | | | | | FSR1[15:8] |
| 0x1D88 | BSR | 7:0 | | | | | | | | | | | BSR[5:0] |
| 0x1D89 | WREG | 7:0 | | | | | | | | | | | WREG[7:0] |
| 0x1D8A | PCLATH | 7:0 | | | | | | | | | | | PCLATH[6:0] |
| 0x1D8B | INTCON | 7:0 | GIE | PEIE | | | | | | | | | INTEDG |
| 0x1D8C | Reserved | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | |
| 0x1DFF | | | | | | | | | | | | | |
| 0x1E00 | INDF0 | 7:0 | | | | | | | | | | | INDF0[7:0] |
| 0x1E01 | INDF1 | 7:0 | | | | | | | | | | | INDF1[7:0] |
| 0x1E02 | PCL | 7:0 | | | | | | | | | | | PCL[7:0] |
| 0x1E03 | STATUS | 7:0 | | | | | TO | PD | Z | DC | | | C |
| 0x1E04 | FSR0 | 7:0 | | | | | | | | | | | FSR0[7:0] |
| | | 15:8 | | | | | | | | | | | FSR0[15:8] |
| 0x1E06 | FSR1 | 7:0 | | | | | | | | | | | FSR1[7:0] |
| | | 15:8 | | | | | | | | | | | FSR1[15:8] |
| 0x1E08 | BSR | 7:0 | | | | | | | | | | | BSR[5:0] |
| 0x1E09 | WREG | 7:0 | | | | | | | | | | | WREG[7:0] |
| 0x1E0A | PCLATH | 7:0 | | | | | | | | | | | PCLATH[6:0] |
| 0x1E0B | INTCON | 7:0 | GIE | PEIE | | | | | | | | | INTEDG |
| 0x1E0C | Reserved | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | |
| 0x1E0E | | | | | | | | | | | | | |
| 0x1E0F | CLCDATA | 7:0 | | | | | | MLC4OUT | MLC3OUT | MLC2OUT | | | MLC1OUT |
| 0x1E10 | CLC1CON | 7:0 | EN | | OUT | INTP | INTN | | | | | MODE[2:0] | |
| 0x1E11 | CLC1POL | 7:0 | POL | | | | | G4POL | G3POL | G2POL | | | G1POL |
| 0x1E12 | CLC1SEL0 | 7:0 | | | | | | | | | | | D1S[5:0] |
| 0x1E13 | CLC1SEL1 | 7:0 | | | | | | | | | | | D2S[5:0] |
| 0x1E14 | CLC1SEL2 | 7:0 | | | | | | | | | | | D3S[5:0] |
| 0x1E15 | CLC1SEL3 | 7:0 | | | | | | | | | | | D4S[5:0] |
| 0x1E16 | CLC1GLS0 | 7:0 | G1D4T | G1D4N | G1D3T | G1D3N | G1D2T | G1D2N | G1D1T | G1D1N | | | |
| 0x1E17 | CLC1GLS1 | 7:0 | G2D4T | G2D4N | G2D3T | G2D3N | G2D2T | G2D2N | G2D1T | G2D1N | | | |
| 0x1E18 | CLC1GLS2 | 7:0 | G3D4T | G3D4N | G3D3T | G3D3N | G3D2T | G3D2N | G3D1T | G3D1N | | | |
| 0x1E19 | CLC1GLS3 | 7:0 | G4D4T | G4D4N | G4D3T | G4D3N | G4D2T | G4D2N | G4D1T | G4D1N | | | |
| 0x1E1A | CLC2CON | 7:0 | EN | | OUT | INTP | INTN | | | | | MODE[2:0] | |
| 0x1E1B | CLC2POL | 7:0 | POL | | | | | G4POL | G3POL | G2POL | | | G1POL |
| 0x1E1C | CLC2SEL0 | 7:0 | | | | | | | | | | | D1S[5:0] |
| 0x1E1D | CLC2SEL1 | 7:0 | | | | | | | | | | | D2S[5:0] |
| 0x1E1E | CLC2SEL2 | 7:0 | | | | | | | | | | | D3S[5:0] |
| 0x1E1F | CLC2SEL3 | 7:0 | | | | | | | | | | | D4S[5:0] |
| 0x1E20 | CLC2GLS0 | 7:0 | G1D4T | G1D4N | G1D3T | G1D3N | G1D2T | G1D2N | G1D1T | G1D1N | | | |
| 0x1E21 | CLC2GLS1 | 7:0 | G2D4T | G2D4N | G2D3T | G2D3N | G2D2T | G2D2N | G2D1T | G2D1N | | | |
| 0x1E22 | CLC2GLS2 | 7:0 | G3D4T | G3D4N | G3D3T | G3D3N | G3D2T | G3D2N | G3D1T | G3D1N | | | |
| 0x1E23 | CLC2GLS3 | 7:0 | G4D4T | G4D4N | G4D3T | G4D3N | G4D2T | G4D2N | G4D1T | G4D1N | | | |
| 0x1E24 | CLC3CON | 7:0 | EN | | OUT | INTP | INTN | | | | | MODE[2:0] | |
| 0x1E25 | CLC3POL | 7:0 | POL | | | | | G4POL | G3POL | G2POL | | | G1POL |
| 0x1E26 | CLC3SEL0 | 7:0 | | | | | | | | | | | D1S[5:0] |
| 0x1E27 | CLC3SEL1 | 7:0 | | | | | | | | | | | D2S[5:0] |
| 0x1E28 | CLC3SEL2 | 7:0 | | | | | | | | | | | D3S[5:0] |
| 0x1E29 | CLC3SEL3 | 7:0 | | | | | | | | | | | D4S[5:0] |
| 0x1E2A | CLC3GLS0 | 7:0 | G1D4T | G1D4N | G1D3T | G1D3N | G1D2T | G1D2N | G1D1T | G1D1N | | | |
| 0x1E2B | CLC3GLS1 | 7:0 | G2D4T | G2D4N | G2D3T | G2D3N | G2D2T | G2D2N | G2D1T | G2D1N | | | |
| 0x1E2C | CLC3GLS2 | 7:0 | G3D4T | G3D4N | G3D3T | G3D3N | G3D2T | G3D2N | G3D1T | G3D1N | | | |

PIC16(L)F18426/46

Register Summary

.....continued

| Address | Name | Bit Pos. | | | | | | | | | |
|---------|------------|----------|-------------|-------|-------|-----------|----------|-----------|----------|-------|-----------|
| 0x1E2D | CLC3GLS3 | 7:0 | G4D4T | G4D4N | G4D3T | G4D3N | G4D2T | G4D2N | G4D1T | G4D1N | |
| 0x1E2E | CLC4CON | 7:0 | EN | | OUT | INTP | INTN | MODE[2:0] | | | |
| 0x1E2F | CLC4POL | 7:0 | POL | | | | G4POL | G3POL | G2POL | G1POL | |
| 0x1E30 | CLC4SEL0 | 7:0 | | | | | D1S[5:0] | | | | |
| 0x1E31 | CLC4SEL1 | 7:0 | | | | | D2S[5:0] | | | | |
| 0x1E32 | CLC4SEL2 | 7:0 | | | | | D3S[5:0] | | | | |
| 0x1E33 | CLC4SEL3 | 7:0 | | | | | D4S[5:0] | | | | |
| 0x1E34 | CLC4GLS0 | 7:0 | G1D4T | G1D4N | G1D3T | G1D3N | G1D2T | G1D2N | G1D1T | G1D1N | |
| 0x1E35 | CLC4GLS1 | 7:0 | G2D4T | G2D4N | G2D3T | G2D3N | G2D2T | G2D2N | G2D1T | G2D1N | |
| 0x1E36 | CLC4GLS2 | 7:0 | G3D4T | G3D4N | G3D3T | G3D3N | G3D2T | G3D2N | G3D1T | G3D1N | |
| 0x1E37 | CLC4GLS3 | 7:0 | G4D4T | G4D4N | G4D3T | G4D3N | G4D2T | G4D2N | G4D1T | G4D1N | |
| 0x1E38 | Reserved | | | | | | | | | | |
| ... | | | | | | | | | | | |
| 0x1E7F | | | | | | | | | | | |
| 0x1E80 | INDF0 | 7:0 | INDF0[7:0] | | | | | | | | |
| 0x1E81 | INDF1 | 7:0 | INDF1[7:0] | | | | | | | | |
| 0x1E82 | PCL | 7:0 | PCL[7:0] | | | | | | | | |
| 0x1E83 | STATUS | 7:0 | | | | TO | PD | Z | DC | C | |
| 0x1E84 | FSR0 | 7:0 | FSR0[7:0] | | | | | | | | |
| | | 15:8 | FSR0[15:8] | | | | | | | | |
| 0x1E86 | FSR1 | 7:0 | FSR1[7:0] | | | | | | | | |
| | | 15:8 | FSR1[15:8] | | | | | | | | |
| 0x1E88 | BSR | 7:0 | BSR[5:0] | | | | | | | | |
| 0x1E89 | WREG | 7:0 | WREG[7:0] | | | | | | | | |
| 0x1E8A | PCLATH | 7:0 | PCLATH[6:0] | | | | | | | | |
| 0x1E8B | INTCON | 7:0 | GIE | PEIE | | | | | | | INTEDG |
| 0x1E8C | Reserved | | | | | | | | | | |
| ... | | | | | | | | | | | |
| 0x1E8E | | | | | | | | | | | |
| 0x1E8F | PPSLOCK | 7:0 | | | | | | | | | PPSLOCKED |
| 0x1E90 | INTPPS | 7:0 | | | | PORT[1:0] | | | PIN[2:0] | | |
| 0x1E91 | T0CKIPPS | 7:0 | | | | PORT[1:0] | | | PIN[2:0] | | |
| 0x1E92 | T1CKIPPS | 7:0 | | | | PORT[1:0] | | | PIN[2:0] | | |
| 0x1E93 | T1GPPS | 7:0 | | | | PORT[1:0] | | | PIN[2:0] | | |
| 0x1E94 | T3CKIPPS | 7:0 | | | | PORT[1:0] | | | PIN[2:0] | | |
| 0x1E95 | T3GPPS | 7:0 | | | | PORT[1:0] | | | PIN[2:0] | | |
| 0x1E96 | T5CKIPPS | 7:0 | | | | PORT[1:0] | | | PIN[2:0] | | |
| 0x1E97 | T5GPPS | 7:0 | | | | PORT[1:0] | | | PIN[2:0] | | |
| 0x1E98 | Reserved | | | | | | | | | | |
| ... | | | | | | | | | | | |
| 0x1E9B | | | | | | | | | | | |
| 0x1E9C | T2INPPS | 7:0 | | | | PORT[1:0] | | | PIN[2:0] | | |
| 0x1E9D | T4INPPS | 7:0 | | | | PORT[1:0] | | | PIN[2:0] | | |
| 0x1E9E | T6INPPS | 7:0 | | | | PORT[1:0] | | | PIN[2:0] | | |
| 0x1E9F | Reserved | | | | | | | | | | |
| ... | | | | | | | | | | | |
| 0x1EA0 | | | | | | | | | | | |
| 0x1EA1 | CCP1PPS | 7:0 | | | | PORT[1:0] | | | PIN[2:0] | | |
| 0x1EA2 | CCP2PPS | 7:0 | | | | PORT[1:0] | | | PIN[2:0] | | |
| 0x1EA3 | CCP3PPS | 7:0 | | | | PORT[1:0] | | | PIN[2:0] | | |
| 0x1EA4 | CCP4PPS | 7:0 | | | | PORT[1:0] | | | PIN[2:0] | | |
| 0x1EA5 | Reserved | | | | | | | | | | |
| ... | | | | | | | | | | | |
| 0x1EA8 | | | | | | | | | | | |
| 0x1EA9 | SMT1WINPPS | 7:0 | | | | PORT[1:0] | | | PIN[2:0] | | |
| 0x1EAA | SMT1SIGPPS | 7:0 | | | | PORT[1:0] | | | PIN[2:0] | | |
| 0x1EAB | Reserved | | | | | | | | | | |
| ... | | | | | | | | | | | |
| 0x1EB0 | | | | | | | | | | | |

PIC16(L)F18426/46

Register Summary

.....continued

| Address | Name | Bit Pos. | | | | | | | | |
|---------|------------|-----------|-----|------|--|-------------|-----------|----------|----------|----------|
| 0x1EB1 | CWG1PPS | 7:0 | | | | PORT[1:0] | | | PIN[2:0] | |
| 0x1EB2 | CWG2PPS | 7:0 | | | | PORT[1:0] | | | PIN[2:0] | |
| 0x1EB3 | Reserved | | | | | | | | | |
| ... | | | | | | | | | | |
| 0x1EB7 | | | | | | | | | | |
| 0x1EB8 | | MDCARLPPS | 7:0 | | | | PORT[1:0] | | | PIN[2:0] |
| 0x1EB9 | MDCARHPPS | 7:0 | | | | PORT[1:0] | | | PIN[2:0] | |
| 0x1EBA | MDSRCPPS | 7:0 | | | | PORT[1:0] | | | PIN[2:0] | |
| 0x1EBB | CLCIN1PPS | 7:0 | | | | PORT[1:0] | | | PIN[2:0] | |
| 0x1EBC | CLCIN2PPS | 7:0 | | | | PORT[1:0] | | | PIN[2:0] | |
| 0x1EBD | CLCIN3PPS | 7:0 | | | | PORT[1:0] | | | PIN[2:0] | |
| 0x1EBE | CLCIN4PPS | 7:0 | | | | PORT[1:0] | | | PIN[2:0] | |
| 0x1EBF | Reserved | | | | | | | | | |
| ... | | | | | | | | | | |
| 0x1EC2 | | | | | | | | | | |
| 0x1EC3 | ADACTPPS | 7:0 | | | | PORT[1:0] | | | PIN[2:0] | |
| 0x1EC4 | Reserved | | | | | | | | | |
| 0x1EC5 | SSP1CLKPPS | 7:0 | | | | PORT[1:0] | | | PIN[2:0] | |
| 0x1EC6 | SSP1DATPPS | 7:0 | | | | PORT[1:0] | | | PIN[2:0] | |
| 0x1EC7 | SSP1SSPPS | 7:0 | | | | PORT[1:0] | | | PIN[2:0] | |
| 0x1EC8 | SSP2CLKPPS | 7:0 | | | | PORT[1:0] | | | PIN[2:0] | |
| 0x1EC9 | SSP2DATPPS | 7:0 | | | | PORT[1:0] | | | PIN[2:0] | |
| 0x1ECA | SSP2SSPPS | 7:0 | | | | PORT[1:0] | | | PIN[2:0] | |
| 0x1ECB | RX1PPS | 7:0 | | | | PORT[1:0] | | | PIN[2:0] | |
| 0x1ECC | CK1PPS | 7:0 | | | | PORT[1:0] | | | PIN[2:0] | |
| 0x1ECD | Reserved | | | | | | | | | |
| ... | | | | | | | | | | |
| 0x1EFF | | | | | | | | | | |
| 0x1F00 | INDF0 | 7:0 | | | | INDF0[7:0] | | | | |
| 0x1F01 | INDF1 | 7:0 | | | | INDF1[7:0] | | | | |
| 0x1F02 | PCL | 7:0 | | | | PCL[7:0] | | | | |
| 0x1F03 | STATUS | 7:0 | | | | TO | PD | Z | DC | C |
| 0x1F04 | FSR0 | 7:0 | | | | FSR0[7:0] | | | | |
| | | 15:8 | | | | FSR0[15:8] | | | | |
| 0x1F06 | FSR1 | 7:0 | | | | FSR1[7:0] | | | | |
| | | 15:8 | | | | FSR1[15:8] | | | | |
| 0x1F08 | BSR | 7:0 | | | | | | BSR[5:0] | | |
| 0x1F09 | WREG | 7:0 | | | | WREG[7:0] | | | | |
| 0x1F0A | PCLATH | 7:0 | | | | PCLATH[6:0] | | | | |
| 0x1F0B | INTCON | 7:0 | GIE | PEIE | | | | | | INTEDG |
| 0x1F0C | Reserved | | | | | | | | | |
| ... | | | | | | | | | | |
| 0x1F0F | | | | | | | | | | |
| 0x1F10 | RA0PPS | 7:0 | | | | PPS[5:0] | | | | |
| 0x1F11 | RA1PPS | 7:0 | | | | PPS[5:0] | | | | |
| 0x1F12 | RA2PPS | 7:0 | | | | PPS[5:0] | | | | |
| 0x1F13 | Reserved | | | | | | | | | |
| 0x1F14 | RA4PPS | 7:0 | | | | PPS[5:0] | | | | |
| 0x1F15 | RA5PPS | 7:0 | | | | PPS[5:0] | | | | |
| 0x1F16 | Reserved | | | | | | | | | |
| ... | | | | | | | | | | |
| 0x1F1B | | | | | | | | | | |
| 0x1F1C | RB4PPS | 7:0 | | | | PPS[5:0] | | | | |
| 0x1F1D | RB5PPS | 7:0 | | | | PPS[5:0] | | | | |
| 0x1F1E | RB6PPS | 7:0 | | | | PPS[5:0] | | | | |
| 0x1F1F | RB7PPS | 7:0 | | | | PPS[5:0] | | | | |
| 0x1F20 | RC0PPS | 7:0 | | | | PPS[5:0] | | | | |
| 0x1F21 | RC1PPS | 7:0 | | | | PPS[5:0] | | | | |
| 0x1F22 | RC2PPS | 7:0 | | | | PPS[5:0] | | | | |

PIC16(L)F18426/46

Register Summary

|continued | | | | | | | | | | |
|----------------|-------------|----------|---------|---------|---------|---------|---------|---------|---------|-------------|
| Address | Name | Bit Pos. | | | | | | | | |
| 0x1F23 | RC3PPS | 7:0 | | | | | | | | PPS[5:0] |
| 0x1F24 | RC4PPS | 7:0 | | | | | | | | PPS[5:0] |
| 0x1F25 | RC5PPS | 7:0 | | | | | | | | PPS[5:0] |
| 0x1F26 | RC6PPS | 7:0 | | | | | | | | PPS[5:0] |
| 0x1F27 | RC7PPS | 7:0 | | | | | | | | PPS[5:0] |
| 0x1F28 | Reserved | | | | | | | | | |
| ... | | | | | | | | | | |
| 0x1F37 | | | | | | | | | | |
| 0x1F38 | ANSELA | 7:0 | | | ANSELA5 | ANSELA4 | | ANSELA2 | ANSELA1 | ANSELA0 |
| 0x1F39 | WPUA | 7:0 | | | WPUA5 | WPUA4 | WPUA3 | WPUA2 | WPUA1 | WPUA0 |
| 0x1F3A | ODCONA | 7:0 | | | ODCA5 | ODCA4 | | ODCA2 | ODCA1 | ODCA0 |
| 0x1F3B | SLRCONA | 7:0 | | | SLRA5 | SLRA4 | | SLRA2 | SLRA1 | SLRA0 |
| 0x1F3C | INLVLA | 7:0 | | | INLVLA5 | INLVLA4 | INLVLA3 | INLVLA2 | INLVLA1 | INLVLA0 |
| 0x1F3D | IOCAP | 7:0 | | | IOCAP5 | IOCAP4 | IOCAP3 | IOCAP2 | IOCAP1 | IOCAP0 |
| 0x1F3E | IOCAN | 7:0 | | | IOCAN5 | IOCAN4 | IOCAN3 | IOCAN2 | IOCAN1 | IOCAN0 |
| 0x1F3F | IOCAF | 7:0 | | | IOCAF5 | IOCAF4 | IOCAF3 | IOCAF2 | IOCAF1 | IOCAF0 |
| 0x1F40 | Reserved | | | | | | | | | |
| ... | | | | | | | | | | |
| 0x1F42 | | | | | | | | | | |
| 0x1F43 | ANSELB | 7:0 | ANSELB7 | ANSELB6 | ANSELB5 | ANSELB4 | | | | |
| 0x1F44 | WPUB | 7:0 | WPUB7 | WPUB6 | WPUB5 | WPUB4 | | | | |
| 0x1F45 | ODCONB | 7:0 | ODCB7 | ODCB6 | ODCB5 | ODCB4 | | | | |
| 0x1F46 | SLRCONB | 7:0 | SLRB7 | SLRB6 | SLRB5 | SLRB4 | | | | |
| 0x1F47 | INVLVB | 7:0 | INVLVB7 | INVLVB6 | INVLVB5 | INVLVB4 | | | | |
| 0x1F48 | IOCBP | 7:0 | IOCBP7 | IOCBP6 | IOCBP5 | IOCBP4 | | | | |
| 0x1F49 | IOCBN | 7:0 | IOCBN7 | IOCBN6 | IOCBN5 | IOCBN4 | | | | |
| 0x1F4A | IOCBF | 7:0 | IOCBF7 | IOCBF6 | IOCBF5 | IOCBF4 | | | | |
| 0x1F4B | Reserved | | | | | | | | | |
| ... | | | | | | | | | | |
| 0x1F4D | | | | | | | | | | |
| 0x1F4E | ANSELC | 7:0 | ANSELC7 | ANSELC6 | ANSELC5 | ANSELC4 | ANSELC3 | ANSELC2 | ANSELC1 | ANSELC0 |
| 0x1F4F | WPUC | 7:0 | WPUC7 | WPUC6 | WPUC5 | WPUC4 | WPUC3 | WPUC2 | WPUC1 | WPUC0 |
| 0x1F50 | ODCONC | 7:0 | ODCC7 | ODCC6 | ODCC5 | ODCC4 | ODCC3 | ODCC2 | ODCC1 | ODCC0 |
| 0x1F51 | SLRCONC | 7:0 | SLRC7 | SLRC6 | SLRC5 | SLRC4 | SLRC3 | SLRC2 | SLRC1 | SLRC0 |
| 0x1F52 | INLVLC | 7:0 | INLVLC7 | INLVLC6 | INLVLC5 | INLVLC4 | INLVLC3 | INLVLC2 | INLVLC1 | INLVLC0 |
| 0x1F53 | IOCCP | 7:0 | IOCCP7 | IOCCP6 | IOCCP5 | IOCCP4 | IOCCP3 | IOCCP2 | IOCCP1 | IOCCP0 |
| 0x1F54 | IOCCN | 7:0 | IOCCN7 | IOCCN6 | IOCCN5 | IOCCN4 | IOCCN3 | IOCCN2 | IOCCN1 | IOCCN0 |
| 0x1F55 | IOCCF | 7:0 | IOCCF7 | IOCCF6 | IOCCF5 | IOCCF4 | IOCCF3 | IOCCF2 | IOCCF1 | IOCCF0 |
| 0x1F56 | Reserved | | | | | | | | | |
| ... | | | | | | | | | | |
| 0x1F7F | | | | | | | | | | |
| 0x1F80 | INDF0 | 7:0 | | | | | | | | INDF0[7:0] |
| 0x1F81 | INDF1 | 7:0 | | | | | | | | INDF1[7:0] |
| 0x1F82 | PCL | 7:0 | | | | | | | | PCL[7:0] |
| 0x1F83 | STATUS | 7:0 | | | | TO | PD | Z | DC | C |
| 0x1F84 | FSR0 | 7:0 | | | | | | | | FSR0[7:0] |
| | | 15:8 | | | | | | | | FSR0[15:8] |
| 0x1F86 | FSR1 | 7:0 | | | | | | | | FSR1[7:0] |
| | | 15:8 | | | | | | | | FSR1[15:8] |
| 0x1F88 | BSR | 7:0 | | | | | | | | BSR[5:0] |
| 0x1F89 | WREG | 7:0 | | | | | | | | WREG[7:0] |
| 0x1F8A | PCLATH | 7:0 | | | | | | | | PCLATH[6:0] |
| 0x1F8B | INTCON | 7:0 | GIE | PEIE | | | | | | INTEDG |
| 0x1F8C | Reserved | | | | | | | | | |
| ... | | | | | | | | | | |
| 0x1FE3 | | | | | | | | | | |
| 0x1FE4 | STATUS_SHAD | 7:0 | | | | TO | PD | Z | DC | C |
| 0x1FE5 | WREG_SHAD | 7:0 | | | | | | | | WREG[7:0] |
| 0x1FE6 | BSR_SHAD | 7:0 | | | | | | | | BSR[5:0] |

PIC16(L)F18426/46

Register Summary

.....continued

| Address | Name | Bit Pos. | | | | | | | | |
|---------|-------------|----------|--|--|--|--|--|--|--|----------------|
| 0x1FE7 | PCLATH_SHAD | 7:0 | | | | | | | | PCLATH[6:0] |
| 0x1FE8 | FSR0_SHAD | 7:0 | | | | | | | | FSR0_SHAD[7:0] |
| | | 15:8 | | | | | | | | |
| 0x1FEA | FSR1_SHAD | 7:0 | | | | | | | | FSR1_SHAD[7:0] |
| | | 15:8 | | | | | | | | |

37. Instruction Set Summary

PIC16(L)F18426/46 devices incorporate the standard set of 50 PIC16 core instructions. Each instruction is a 14-bit word containing the operation code (opcode) and all required operands. The opcodes are broken into three broad categories:

- Byte Oriented
- Bit Oriented
- Literal and Control

The literal and control category contains the most varied instruction word format.

The [Instruction Set](#) table lists the instructions recognized by the MPASM™ assembler.

All instructions are executed within a single instruction cycle, with the following exceptions, which may take two or three cycles:

- Subroutine entry takes two cycles (CALL, CALLW)
- Returns from interrupts or subroutines take two cycles (RETURN, RETLW, RETFIE)
- Program branching takes two cycles (GOTO, BRA, BRW, BTFSS, BTFSC, DECFSZ, INCSFZ)
- One additional instruction cycle will be used when any instruction references an indirect file register and the file select register is pointing to program memory.

One instruction cycle consists of 4 oscillator cycles; for an oscillator frequency of 4 MHz, this gives a nominal instruction execution rate of 1 MHz.

All instruction examples use the format '0xhh' to represent a hexadecimal number, where 'h' signifies a hexadecimal digit.

37.1 Read-Modify-Write Operations

Any WRITE instruction that specifies a file register as part of the instruction performs a Read-Modify-Write (R-M-W) operation. The register is read, the data is modified, and the result is stored according to either the working (W) register, or the originating file register, depending on the state of the destination designator 'd' (see the table below for more information). A read operation is performed on a register even if the instruction writes to that register.

Table 37-1. 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). The assembler will generate code with x = 0. It is the recommended form of use for compatibility with all Microchip software tools. |
| d | Destination select; d = 0: store result in W, d = 1: store result in file register f. |
| n | FSR or INDF number. (0-1) |
| mm | Prepost increment-decrement mode selection |

Table 37-2. Abbreviation Descriptions

| Field | Description |
|-------|-----------------|
| PC | Program Counter |

PIC16(L)F18426/46

Instruction Set Summary

|continued | |
|-----------------|-----------------|
| Field | Description |
| \overline{TO} | Time-Out bit |
| C | Carry bit |
| DC | Digit Carry bit |
| Z | Zero bit |
| \overline{PD} | Power-Down bit |

37.2 Standard Instruction Set

Table 37-3. Instruction Set

| Mnemonic, Operands | Description | Cycles | 14-Bit Opcode | | | | Status Affected | Notes | |
|---------------------------------|-------------|------------------------------|---------------|----|------|------|--------------------|----------|---|
| | | | MSb | | | LSb | | | |
| BYTE-ORIENTED OPERATIONS | | | | | | | | | |
| ADDWF | f, d | Add WREG and f | 1 | 00 | 0111 | dfff | ffff | C, DC, Z | 2 |
| ADDWFC | f, d | Add WREG and CARRY bit to f | 1 | 11 | 1101 | dfff | ffff | C, DC, Z | 2 |
| ANDWF | f, d | AND WREG with f | 1 | 00 | 0101 | dfff | ffff | Z | 2 |
| ASRF | f, d | Arithmetic Right Shift | 1 | 11 | 0111 | dfff | ffff | C, Z | 2 |
| LSLF | f, d | Logical Left Shift | 1 | 11 | 0101 | dfff | ffff | C, Z | 2 |
| LSRF | f, d | Logical Right Shift | 1 | 11 | 0110 | dfff | ffff | C, Z | 2 |
| CLRF | f | Clear f | 1 | 00 | 0001 | 1fff | ffff | Z | 2 |
| CLRW | – | Clear WREG | 1 | 00 | 0001 | 0000 | 00xx | Z | |
| COMF | f, d | Complement f | 1 | 00 | 1001 | dfff | ffff | Z | 2 |
| DECf | f, d | Decrement f | 1 | 00 | 0011 | dfff | ffff | Z | 2 |
| INCF | f, d | Increment f | 1 | 00 | 1010 | dfff | ffff | Z | 2 |
| IORWF | f, d | Inclusive OR WREG with f | 1 | 00 | 0100 | dfff | ffff | Z | 2 |
| MOVF | f, d | Move f | 1 | 00 | 1000 | dfff | ffff | Z | 2 |
| MOVWF | f | Move WREG to f | 1 | 00 | 0000 | 1fff | ffff | None | 2 |
| RLF | f, d | Rotate Left f through Carry | 1 | 00 | 1101 | dfff | ffff | C | 2 |
| RRF | f, d | Rotate Right f through Carry | 1 | 00 | 1100 | dfff | ffff | C | 2 |

PIC16(L)F18426/46

Instruction Set Summary

|continued | | | | | | | | | |
|--|-------------|----------------------------------|---------------|------|------|------|--------------------|----------|------|
| Mnemonic, Operands | Description | Cycles | 14-Bit Opcode | | | | Status Affected | Notes | |
| | | | MSb | | | LSb | | | |
| SUBWF | f, d | Subtract WREG from f | 1 | 00 | 0010 | dfff | ffff | C, DC, Z | 2 |
| SUBWFB | f, d | Subtract WREG from f with borrow | 1 | 11 | 1011 | dfff | ffff | C, DC, Z | 2 |
| SWAPF | f, d | Swap nibbles in f | 1 | 00 | 1110 | dfff | ffff | None | 2 |
| XORWF | f, d | Exclusive OR WREG with f | 1 | 00 | 0110 | dfff | ffff | Z | 2 |
| BYTE ORIENTED SKIP OPERATIONS | | | | | | | | | |
| DECFSZ | f, d | Decrement f, Skip if 0 | 1(2) | 00 | 1011 | dfff | ffff | None | 1, 2 |
| INCFSZ | f, d | Increment f, Skip if 0 | 1(2) | 00 | 1111 | dfff | ffff | None | 1, 2 |
| BIT-ORIENTED FILE REGISTER OPERATIONS | | | | | | | | | |
| BCF | f, b | Bit Clear f | 1 | 01 | 00bb | bfff | ffff | None | 2 |
| BSF | f, b | Bit Set f | 1 | 01 | 01bb | bfff | ffff | None | 2 |
| BIT-ORIENTED SKIP OPERATIONS | | | | | | | | | |
| BTFSC | f, b | Bit Test f, Skip if Clear | 1(2) | 01 | 10bb | bfff | ffff | None | 1, 2 |
| BTFSS | f, b | Bit Test f, Skip if Set | 1(2) | 1010 | 11bb | bfff | ffff | None | 1, 2 |
| LITERAL OPERATIONS | | | | | | | | | |
| ADDLW | k | Add literal and WREG | 1 | 11 | 1110 | kkkk | kkkk | C, DC, Z | |
| ANDLW | k | AND literal with WREG | 1 | 11 | 1001 | kkkk | kkkk | Z | |
| IORLW | k | Inclusive OR literal with WREG | 1 | 11 | 1000 | kkkk | kkkk | Z | |
| MOVLB | k | Move literal to BSR | 1 | 00 | 000 | 0k | kkkk | None | |
| MOVLP | k | Move literal to PCLATH | 1 | 11 | 0001 | 1kkk | kkkk | None | |
| MOVLW | k | Move literal to W | 1 | 11 | 0000 | kkkk | kkkk | None | |
| SUBLW | k | Subtract W from literal | 1 | 11 | 1100 | kkkk | kkkk | C, DC, Z | |
| XORLW | k | Exclusive OR literal with W | 1 | 11 | 1010 | kkkk | kkkk | Z | |
| CONTROL OPERATIONS | | | | | | | | | |

PIC16(L)F18426/46

Instruction Set Summary

.....continued

| Mnemonic, Operands | | Description | Cycles | 14-Bit Opcode | | | | Status Affected | Notes |
|-----------------------------|-------|---|--------|---------------|------|------|------|-----------------------------------|-------|
| | | | | MSb | | | LSb | | |
| BRA | k | Relative Branch | 2 | 11 | 001k | kkkk | kkkk | None | |
| BRW | — | Relative Branch with WREG | 2 | 00 | 0000 | 0000 | 1011 | None | |
| CALL | k | Call Subroutine | 2 | 10 | 0kkk | kkkk | kkkk | None | |
| CALLW | — | Call Subroutine with WREG | 2 | 00 | 0000 | 0000 | 1010 | None | |
| GOTO | k | Go to address | 2 | 10 | 1kkk | kkkk | kkkk | None | |
| RETFIE | k | Return from interrupt | 2 | 00 | 0000 | 0000 | 1001 | None | |
| RETLW | k | Return with literal in WREG | 2 | 11 | 0100 | kkkk | kkkk | None | |
| RETURN | — | Return from Subroutine | 2 | 00 | 0000 | 0000 | 1000 | None | |
| INHERENT OPERATIONS | | | | | | | | | |
| CLRWDT | — | Clear Watchdog Timer | 1 | 00 | 0000 | 0110 | 0100 | \overline{TO} , \overline{PD} | |
| NOP | — | No Operation | 1 | 00 | 0000 | 0000 | 0000 | None | |
| RESET | — | Software device Reset | 1 | 00 | 0000 | 0000 | 0001 | None | |
| SLEEP | — | Go into Standby or Idle mode | 1 | 00 | 0000 | 0110 | 0011 | \overline{TO} , \overline{PD} | |
| TRIS | f | Load TRIS register with WREG | 1 | 00 | 0000 | 0110 | 0fff | None | |
| C-COMPILER OPTIMIZED | | | | | | | | | |
| ADDFSR | n, k | Add Literal k to FSRn | 1 | 11 | 0001 | 0nkk | kkkk | None | |
| MOVIW | n, mm | Move Indirect FSRn to WREG with pre/post inc/dec modifier, mm | 1 | 00 | 0000 | 0001 | 0nmm | Z | 2, 3 |
| | k[n] | Move INDFn to WREG, Indexed Indirect. | 1 | 11 | 1111 | 0nkk | kkkk | Z | 2 |
| MOVWI | n, mm | Move WREG to Indirect FSRn with pre/post inc/dec modifier, mm | 1 | 00 | 0000 | 0001 | 1nmm | None | 2, 3 |
| | k[n] | Move WREG to INDFn, Indexed Indirect. | 1 | 11 | 1111 | 1nkk | kkkk | None | 2 |

Note:

1. If the Program Counter (PC) is modified, or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a NOP.
2. If this instruction addresses an INDF register and the MSb of the corresponding FSR is set, this instruction will require one additional instruction cycle.
3. Details on MOVIW and MOVWI instruction descriptions are available in the next section.

37.2.1 Standard Instruction Set

| ADDFSR | Add Literal to FSRn |
|------------------|---|
| Syntax: | [<i>label</i>] ADDFSR FSRn, k |
| Operands: | $-32 \leq k \leq 31$; $n \in [0, 1]$ |
| Operation: | $FSR(n) + k \rightarrow FSR(n)$ |
| Status Affected: | None |
| Description: | The signed 6-bit literal 'k' is added to the contents of the FSRnH:FSRnL register pair. FSRn is limited to the range 0000h-FFFFh. Moving beyond these bounds will cause the FSR to wrap-around. |

| ADDLW | ADD literal to W |
|------------------|---|
| Syntax: | [<i>label</i>] ADDLW k |
| Operands: | $0 \leq k \leq 255$ |
| Operation: | $(W) + k \rightarrow (W)$ |
| Status Affected: | C, DC, Z |
| Description: | The contents of W are added to the 8-bit literal 'k' and the result is placed in W. |

| ADDWF | ADD W to f |
|------------------|--|
| Syntax: | [<i>label</i>] ADDWF f, d |
| Operands: | $0 \leq f \leq 127$ $d \in [0,1]$ |
| Operation: | $(W) + (f) \rightarrow \text{dest}$ |
| Status Affected: | C, DC, Z |
| Description: | Add the contents of the W register with register 'f'. If 'd' is '0', the result is stored in the W register. If 'd' is '1', the result is stored back in register 'f'. |

| ADDWFC | ADD W and CARRY bit to f |
|------------------|--|
| Syntax: | [<i>label</i>] ADDWFC f {,d} |
| Operands: | $0 \leq f \leq 127$ $d \in [0,1]$ |
| Operation: | $(W) + (f) + (C) \rightarrow \text{dest}$ |
| Status Affected: | C, DC, Z |
| Description: | Add W, the Carry flag and data memory location 'f'. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed in data memory location 'f'. |

| ANDLW | AND literal with W |
|------------|--|
| Syntax: | [<i>label</i>] ANDLW k |
| Operands: | $0 \leq k \leq 255$ |
| Operation: | $(W) \text{ .AND. } k \rightarrow (W)$ |

|continued | |
|------------------|--|
| ANDLW | AND literal with W |
| Status Affected: | Z |
| Description: | The contents of W are AND'ed with the 8-bit literal 'k'. The result is placed in W. |

| ANDWF | AND W with f |
|------------------|---|
| Syntax: | [<i>label</i>] ANDWF f, d |
| Operands: | $0 \leq f \leq 127$ $d \in [0,1]$ |
| Operation: | (W) .AND. (f) → dest |
| Status Affected: | Z |
| Description: | AND the W register with register 'f'. If 'd' is '0', the result is stored in the W register. If 'd' is '1', the result is stored back in register 'f'. |

| ASRF | Arithmetic Right Shift |
|------------------|--|
| Syntax: | [<i>label</i>] ASRF f, d |
| Operands: | $0 \leq f \leq 127$ $d \in [0,1]$ |
| Operation: | (f[7]) → dest[7] (f[7:1]) → dest[6:0] (f[0]) → C |
| Status Affected: | C, Z |
| Description: | The contents of register 'f' are shifted one bit to the right through the Carry flag. The MSb remains unchanged. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is stored back in register 'f'. Register f → C |

| BCF | Bit Clear f |
|------------------|--|
| Syntax: | [<i>label</i>] BCF f, b |
| Operands: | $0 \leq f \leq 127$ $0 \leq b \leq 7$ |
| Operation: | $0 \rightarrow f[b]$ |
| Status Affected: | None |
| Description: | Bit 'b' in register 'f' is cleared. |

| BRA | Relative Branch |
|-----------|---|
| Syntax: | [<i>label</i>] BRA label [<i>label</i>] BRA \$+k |
| Operands: | $-256 \leq \text{label} - \text{PC} + \leq 255$ $-256 \leq k \leq 255$ |

PIC16(L)F18426/46

Instruction Set Summary

|continued | |
|------------------|--|
| BRA | Relative Branch |
| Operation: | $(PC) + 1 + k \rightarrow PC$ |
| Status Affected: | None |
| Description: | Add the signed 9-bit literal 'k' to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be $PC + 1 + k$. This instruction is a 2-cycle instruction. This branch has a limited range. |

| BRW | Relative Branch with W |
|------------------|---|
| Syntax: | <code>[label] BRW</code> |
| Operands: | None |
| Operation: | $(PC) + (W) \rightarrow PC$ |
| Status Affected: | None |
| Description: | Add the contents of W (unsigned) to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be $PC + 1 + (W)$. This instruction is a 2-cycle instruction. |

| BSF | Bit Set f |
|------------------|--|
| Syntax: | <code>[label] BSF f, b</code> |
| Operands: | $0 \leq f \leq 127$ $0 \leq b \leq 7$ |
| Operation: | $1 \rightarrow (f[b])$ |
| Status Affected: | None |
| Description: | Bit 'b' in register 'f' is set. |

| BTFSC | Bit Test File, Skip if Clear |
|------------------|---|
| Syntax: | <code>[label] BTFSC f, b</code> |
| Operands: | $0 \leq f \leq 127$ $0 \leq b \leq 7$ |
| Operation: | skip if $(f[b]) = 0$ |
| Status Affected: | None |
| Description: | If bit 'b' in register 'f' is '1', the next instruction is executed. If bit 'b', in register 'f', is '0', the next instruction is discarded, and a NOP is executed instead, making this a 2-cycle instruction. |

| BTFSS | Bit Test File, Skip if Set |
|------------|---------------------------------------|
| Syntax: | <code>[label] BTFSS f, b</code> |
| Operands: | $0 \leq f \leq 127$ $0 \leq b < 7$ |
| Operation: | skip if $(f[b]) = 1$ |

PIC16(L)F18426/46

Instruction Set Summary

|continued | |
|------------------|--|
| BTFSS | Bit Test File, Skip if Set |
| Status Affected: | None |
| Description: | If bit 'b' in register 'f' is '0', the next instruction is executed. If bit 'b' is '1', then the next instruction is discarded, and a NOP is executed instead, making this a 2-cycle instruction. |

| CALL | Subroutine Call |
|------------------|---|
| Syntax: | [<i>label</i>] CALL k |
| Operands: | $0 \leq k \leq 2047$ |
| Operation: | (PC) + 1 → TOS, k → PC[10:0], (PCLATH[6:3]) → PC[14:11] |
| Status Affected: | None |
| Description: | Call Subroutine. First, return address (PC + 1) is pushed onto the stack. The 11-bit immediate address is loaded into PC bits [10:0]. The upper bits of the PC are loaded from PCLATH. CALL is a 2-cycle instruction. |

| CALLW | Subroutine Call with W |
|------------------|---|
| Syntax: | [<i>label</i>] CALLW |
| Operands: | None |
| Operation: | (PC) + 1 → TOS, (W) → PC[7:0], (PCLATH[6:0]) → PC[14:8] |
| Status Affected: | None |
| Description: | Subroutine call with W. First, the return address (PC + 1) is pushed onto the return stack. Then, the contents of W is loaded into PC[7:0], and the contents of PCLATH into PC[14:8]. CALLW is a 2-cycle instruction. |

| CLRF | Clear f |
|------------------|--|
| Syntax: | [<i>label</i>] CLRF f |
| Operands: | $0 \leq f \leq 127$ |
| Operation: | 000h → f 1 → Z |
| Status Affected: | Z |
| Description: | The contents of register 'f' are cleared and the Z bit is set. |

| CLRW | Clear W |
|-----------|-----------------------|
| Syntax: | [<i>label</i>] CLRW |
| Operands: | None |

|continued | |
|------------------|---|
| CLRW | Clear W |
| Operation: | 00h → (W) 1 → Z |
| Status Affected: | Z |
| Description: | W register is cleared. Zero bit (Z) is set. |

| CLRWDT | Clear Watchdog Timer |
|------------------|--|
| Syntax: | [<i>label</i>] CLRWDT |
| Operands: | None |
| Operation: | 00h → WDT, 00h → WDT prescaler, 1 → \overline{TO} , 1 → \overline{PD} |
| Status Affected: | \overline{TO} , \overline{PD} |
| Description: | CLRWDT instruction resets the Watchdog Timer. It also resets the prescaler of the WDT. Status bits, \overline{TO} and \overline{PD} , are set. |

| COMF | Complement f |
|------------------|--|
| Syntax: | [<i>label</i>] COMF f, d |
| Operands: | $0 \leq f \leq 127$ $d \in [0,1]$ |
| Operation: | $(\bar{f}) \rightarrow \text{dest}$ |
| 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: | [<i>label</i>] DECF f, d |
| Operands: | $0 \leq f \leq 127$ $d \in [0,1]$ |
| Operation: | $(f) - 1 \rightarrow \text{dest}$ |
| Status Affected: | Z |
| Description: | Decrement register 'f'. If 'd' is '0', the result is stored in the W register. If 'd' is '1', the result is stored back in register 'f'. |

| DECFSZ | Decrement f, skip if 0 |
|---------|------------------------------|
| Syntax: | [<i>label</i>] DECFSZ f, d |

|continued | |
|----------------|---|
| DECFSZ | Decrement f, skip if 0 |
| Operands: | $0 \leq f \leq 127$ $d \in [0,1]$ |
| Operation: | $(f) - 1 \rightarrow \text{dest}$, skip if result = 0 |
| Description: | The contents of register 'f' are decremented. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'. If the result is '1', the next instruction is executed. If the result is '0', then a NOP is executed instead, making it a 2-cycle instruction. |

| GOTO | Unconditional Branch |
|------------------|--|
| Syntax: | [<i>label</i>] GOTO k |
| Operands: | $0 \leq k \leq 2047$ |
| Operation: | $k \rightarrow \text{PC}[10:0]$ $\text{PCLATH}[6:3] \rightarrow \text{PC}[14:11]$ |
| Status Affected: | None |
| Description: | GOTO is an unconditional branch. The 11-bit immediate value is loaded into PC bits [10:0]. The upper bits of PC are loaded from PCLATH[4:3]. GOTO is a 2-cycle instruction. |

| INCF | Increment f |
|------------------|--|
| Syntax: | [<i>label</i>] INCF f, d |
| Operands: | $0 \leq f \leq 127$ $d \in [0,1]$ |
| Operation: | $(f) + 1 \rightarrow \text{dest}$ |
| 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: | [<i>label</i>] INCFSZ f, d |
| Operands: | $0 \leq f \leq 127$ $d \in [0,1]$ |
| Operation: | $(f) + 1 \rightarrow \text{dest}$, skip if result = 0 |
| Status Affected: | None |

|continued | |
|----------------|--|
| INCFSZ | Increment f, skip if 0 |
| Description: | The contents of register 'f' are incremented. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'. If the result is '1', the next instruction is executed. If the result is '0', a NOP is executed instead, making it a 2-cycle instruction. |

| IORLW | Inclusive OR literal with W |
|------------------|--|
| Syntax: | [<i>label</i>] IORLW k |
| Operands: | $0 \leq k \leq 255$ |
| Operation: | (W) .OR. k \rightarrow (W) |
| Status Affected: | Z |
| Description: | The contents of W are ORed with the 8-bit literal 'k'. The result is placed in W. |

| IORWF | Inclusive OR W with f |
|------------------|---|
| Syntax: | IORWF f, d |
| Operands: | $0 \leq f \leq 127$ $d \in [0,1]$ |
| Operation: | (W) .OR. (f) \rightarrow dest |
| 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'. |

| LSLF | Logical Left Shift |
|------------------|---|
| Syntax: | [<i>label</i>] LSLF f {,d} |
| Operands: | $0 \leq f \leq 127$ $d \in [0,1]$ |
| Operation: | (f[7]) \rightarrow C (f[6:0]) \rightarrow dest[7:1] $0 \rightarrow$ dest[0] |
| Status Affected: | C, Z |
| Description: | The contents of register 'f' are shifted one bit to the left through the Carry flag. A '0' is shifted into the LSb. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is stored back in register 'f'. C \leftarrow Register f \leftarrow 0 |

| LSRF | Logical Right Shift |
|---------|------------------------------|
| Syntax: | [<i>label</i>] LSRF f {,d} |

|continued | |
|------------------|---|
| LSRF | Logical Right Shift |
| Operands: | $0 \leq f \leq 127$ $d \in [0,1]$ |
| Operation: | $0 \rightarrow \text{dest}[7]$ $(f[7:1]) \rightarrow \text{dest}[6:0],$ $(f[0]) \rightarrow C$ |
| Status Affected: | C, Z |
| Description: | The contents of register 'f' are shifted one bit to the right through the Carry flag. A '0' is shifted into the MSb. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is stored back in register 'f'. 0 → register f → C |

| MOVF | Move f |
|------------------|--|
| Syntax: | [<i>label</i>] MOVF f, d |
| Operands: | $0 \leq f \leq 127$ $d \in [0,1]$ |
| Operation: | $f \rightarrow \text{dest}$ |
| Status Affected: | Z |
| Description: | The contents of register f is moved to a destination dependent upon the status of d. If $d = 0$, destination is W register. If $d = 1$, the destination is file register f itself. $d = 1$ is useful to test a file register since status flag Z is affected. |
| Words: | 1 |
| Cycles: | 1 |

| | |
|---|-------------|
| Example: | MOVF FSR, 0 |
| After Instruction W = value in FSR register Z = 1 | |

| MOVIW | Move INDFn to W |
|-----------|--|
| Syntax: | [<i>label</i>] MOVIW ++FSRn [<i>label</i>] MOVIW --FSRn [<i>label</i>] MOVIW FSRn++ [<i>label</i>] MOVIW FSRn-- [<i>label</i>] MOVIW k[FSRn] |
| Operands: | $n \in [0,1]$ $mm \in [00,01,10,11]$ $-32 \leq k \leq 31$ |

|continued | | | |
|------------------|--|-------------|---------------|
| MOVIW | Move INDFn to W | | |
| Operation: | INDFn → (W) Effective address is determined by <ul style="list-style-type: none"> • FSR + 1 (preincrement) • FSR - 1 (predecrement) • FSR + k (relative offset) After the Move, the FSR value will be either: <ul style="list-style-type: none"> • FSR + 1 (all increments) • FSR - 1 (all decrements) • Unchanged | | |
| Status Affected: | Z | | |
| | | MODE | SYNTAX |
| | | | mm |
| | Preincrement | | ++FSRn |
| | Predecrement | | --FSRn |
| | Postincrement | | FSRn++ |
| | Postdecrement | | FSRn-- |
| Description: | This instruction is used to move data between W and one of the indirect registers (INDFn). Before/after this move, the pointer (FSRn) is updated by pre/post incrementing/decrementing it. The INDFn registers are not physical registers. Any instruction that accesses an INDFn register actually accesses the register at the address specified by the FSRn. FSRn is limited to the range 0000h - FFFFh. Incrementing/decrementing it beyond these bounds will cause it to wrap-around. | | |

| MOVLB | Move literal to BSR |
|------------------|--|
| Syntax: | [<i>label</i>] MOVLB k |
| Operands: | 0 ≤ k ≤ 127 |
| Operation: | k → BSR |
| Status Affected: | None |
| Description: | The 6-bit literal 'k' is loaded into the Bank Select Register (BSR). |

| MOVLP | Move literal to PCLATH |
|------------------|---|
| Syntax: | [<i>label</i>] MOVLP k |
| Operands: | 0 ≤ k ≤ 127 |
| Operation: | k → PCLATH |
| Status Affected: | None |
| Description: | The 7-bit literal 'k' is loaded into the PCLATH register. |

| MOVLW | Move literal to W |
|-----------|--------------------------|
| Syntax: | [<i>label</i>] MOVLW k |
| Operands: | 0 ≤ k ≤ 255 |

|continued | | | | |
|------------------|--|--|--|--|
| MOVLW | Move literal to W | | | |
| Operation: | k → (W) | | | |
| Status Affected: | None | | | |
| Description: | The 8-bit literal 'k' is loaded into W register. The "don't cares" will assemble as '0's. | | | |
| Words: | 1 | | | |
| Cycles: | 1 | | | |

| | | |
|------------------------------|-------|-----|
| Example: | MOVLW | 5Ah |
| After Instruction W = 5Ah | | |

| MOVWF | Move W to f | | | |
|------------------|-----------------------------------|--|--|--|
| Syntax: | [<i>label</i>] MOVWF f | | | |
| Operands: | 0 ≤ f ≤ 127 | | | |
| Operation: | (W) → f | | | |
| Status Affected: | None | | | |
| Description: | Move data from W to register 'f'. | | | |
| Words: | 1 | | | |
| Cycles: | 1 | | | |

| | | |
|---|-------|------|
| Example: | MOVWF | LATA |
| Before Instruction LATA = FFh W = 4Fh After Instruction LATA = 4Fh W = 4Fh | | |

| MOVWI | Move W to INDFn | | | |
|-----------|--|--|--|--|
| Syntax: | [<i>label</i>] MOVWI ++FSRn [<i>label</i>] MOVWI --FSRn [<i>label</i>] MOVWI FSRn++ [<i>label</i>] MOVWI FSRn-- [<i>label</i>] MOVWI k[FSRn] | | | |
| Operands: | n ∈ [0,1] mm ∈ [00,01,10,11] -32 ≤ k ≤ 31 | | | |

|continued | | | | |
|------------------|--|--------------|---------------|----|
| MOVWI | Move W to INDFn | | | |
| Operation: | <p>(W) → INDFn</p> <p>Effective address is determined by</p> <ul style="list-style-type: none"> • FSR + 1 (preincrement) • FSR - 1 (predecrement) • FSR + k (relative offset) <p>After the Move, the FSR value will be either:</p> <ul style="list-style-type: none"> • FSR + 1 (all increments) • FSR - 1 (all decrements) • Unchanged | | | |
| Status Affected: | None | | | |
| | | MODE | SYNTAX | |
| | | | mm | |
| | | Preincrement | ++FSRn | 00 |
| | | Predecrement | --FSRn | 01 |
| | Postincrement | FSRn++ | 10 | |
| | Postdecrement | FSRn-- | 11 | |
| Description: | <p>This instruction is used to move data between W and one of the indirect registers (INDFn). Before/after this move, the pointer (FSRn) is updated by pre/post incrementing/decrementing it. The INDFn registers are not physical registers. Any instruction that accesses an INDFn register actually accesses the register at the address specified by the FSRn. FSRn is limited to the range 0000h-FFFFh. Incrementing/decrementing it beyond these bounds will cause it to wrap-around. The increment/decrement operation on FSRn WILL NOT affect any Status bits.</p> | | | |

| NOP | No Operation | | | |
|------------------|----------------------|--|--|--|
| Syntax: | [<i>label</i>] NOP | | | |
| Operands: | None | | | |
| Operation: | No operation | | | |
| Status Affected: | None | | | |
| Description: | No operation. | | | |
| Words: | 1 | | | |
| Cycles: | 1 | | | |

| | |
|----------|-----|
| Example: | NOP |
| None. | |

| RESET | Software Reset |
|------------|--|
| Syntax: | [<i>label</i>] RESET |
| Operands: | None |
| Operation: | Execute a device Reset. Resets the $\bar{R}I$ flag of the PCON register. |

PIC16(L)F18426/46

Instruction Set Summary

.....continued

| RESET | Software Reset |
|------------------|--|
| Status Affected: | None |
| Description: | This instruction provides a way to execute a hardware Reset by software. |

| RETFIE | Return from Interrupt |
|------------------|---|
| Syntax: | [<i>label</i>] RETFIE k |
| Operands: | None |
| Operation: | (TOS) → PC, 1 → GIE |
| Status Affected: | None |
| Description: | Return from Interrupt. Stack is POPed and Top-of-Stack (TOS) is loaded in the PC. Interrupts are enabled by setting Global Interrupt Enable bit, GIE (INTCON[7]). This is a 2-cycle instruction. |
| Words: | 1 |
| Cycles: | 2 |

| | |
|--|--------|
| Example: | RETFIE |
| After Interrupt PC = TOS GIE = 1 | |

| RETLW | Return literal to W |
|------------------|---|
| Syntax: | [<i>label</i>] RETLW k |
| Operands: | 0 ≤ k ≤ 255 |
| Operation: | k → (W), (TOS) → PC, |
| Status Affected: | None |
| Description: | The W register is loaded with the 8-bit literal 'k'. The program counter is loaded from the top of the stack (the return address). This is a 2-cycle instruction. |
| Words: | 1 |
| Cycles: | 2 |

Example:

```

CALL TABLE          ; W contains table
; offset value
; W now has
; table value
:
TABLE
ADDWF PC             ; W = offset
RETLW k1             ; Begin table
RETLW k2             ;
:
:
RETLW kn             ; End of table

```

Before Instruction

W = 07h

After Instruction

W = value of k8

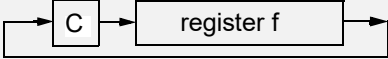
| RETURN | Return from Subroutine |
|------------------|---|
| Syntax: | [<i>label</i>] RETURN |
| Operands: | None |
| Operation: | (TOS) → PC, |
| Status Affected: | None |
| Encoding: | 0000 0000 0001 001s |
| Description: | Return from subroutine. The stack is POPped and the top of the stack (TOS) is loaded into the Program Counter. This is a 2-cycle instruction. |

| RLF | Rotate Left f through Carry |
|------------------|---|
| Syntax: | [<i>label</i>] RLF f, d |
| Operands: | 0 ≤ f ≤ 127 d ∈ [0,1] |
| Operation: | (f[n]) → dest[n + 1], (f[7]) → C, (C) → dest[0] |
| Status Affected: | C |
| Encoding: | 0011 01da ffff ffff |
| 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 W. If 'd' is '1', the result is stored back in register 'f' (default). <div style="text-align: center;"> </div> |
| Words: | 1 |
| Cycles: | 1 |

PIC16(L)F18426/46

Instruction Set Summary

| | | |
|---|-----|---------|
| Example: | RLF | REG1, 0 |
| <p>Before Instruction REG1 = 1110 0110 C = 0</p> <p>After Instruction REG = 1110 0110 W = 1100 1100 C = 1</p> | | |

| RRF | Rotate Right f through Carry |
|------------------|---|
| Syntax: | [<i>label</i>] RRF f, d |
| Operands: | $0 \leq f \leq 127$ $d \in [0,1]$ |
| Operation: | $(f[n]) \rightarrow \text{dest}[n - 1]$, $(f[0]) \rightarrow C$, $(C) \rightarrow \text{dest}[7]$ |
| Status Affected: | C |
| Description: | <p>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 W. If 'd' is '1', the result is placed back in register 'f' (default).</p>  |

| SLEEP | Enter Sleep mode |
|------------------|--|
| Syntax: | [<i>label</i>] SLEEP |
| Operands: | None |
| Operation: | $00h \rightarrow \text{WDT}$, $0 \rightarrow \text{WDT prescaler}$, $1 \rightarrow \overline{\text{TO}}$, $0 \rightarrow \overline{\text{PD}}$ |
| Status Affected: | $\overline{\text{TO}}$, $\overline{\text{PD}}$ |
| Description: | <p>The Power-down Status bit ($\overline{\text{PD}}$) is cleared. The Time-out Status bit ($\overline{\text{TO}}$) is set. Watchdog Timer and its prescaler are cleared.</p> |

| SUBLW | Subtract W from literal |
|------------------|---------------------------|
| Syntax: | [<i>label</i>] SUBLW k |
| Operands: | $0 \leq k \leq 255$ |
| Operation: | $k - (W) \rightarrow (W)$ |
| Status Affected: | C, DC, Z |

|continued | |
|----------------|---|
| SUBLW | Subtract W from literal |
| Description | <p>The W register is subtracted (2's complement method) from the 8-bit literal 'k'. The result is placed in the W register. C = 0, W > k C = 1, W ≤ k DC = 0, W[3:0] > k[3:0] DC = 1, W[3:0] ≤ k[3:0]</p> |

| SUBWF | Subtract W from f |
|------------------|--|
| Syntax: | [<i>label</i>] SUBWF f, d |
| Operands: | 0 ≤ f ≤ 127 d ∈ [0,1] |
| Operation: | (f) - (W) → (dest) |
| Status Affected: | C, DC, Z |
| Description | <p>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]</p> |

| SUBFWB | Subtract W from f with Borrow |
|------------------|--|
| Syntax: | [<i>label</i>] SUBFWB f {,d} |
| Operands: | 0 ≤ f ≤ 127 d ∈ [0,1] |
| Operation: | (W) - (f) - (\bar{B}) → dest |
| Status Affected: | C, DC, Z |
| Description: | <p>Subtract W and the BORROW flag (CARRY) from register 'f' (2's complement method). If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in register 'f'.</p> |

| SWAPF | Swap Nibbles in f |
|------------------|---|
| Syntax: | [<i>label</i>] SWAPF f, d |
| Operands: | 0 ≤ f ≤ 127 d ∈ [0,1] |
| Operation: | (f[3:0]) → dest[7:4], (f[7:4]) → dest[3:0] |
| Status Affected: | None |

.....continued

| SWAPF | Swap Nibbles in f |
|--------------|---|
| Description: | The upper and lower nibbles of register 'f' are exchanged. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed in register 'f' (default). |

| TRIS | Load TRIS Register with W |
|------------------|---|
| Syntax: | [<i>label</i>] TRIS f |
| Operands: | $5 \leq f \leq 7$ |
| Operation: | (W) → TRIS register 'f' |
| Status Affected: | None |
| Description: | Move data from W register to TRIS register. When 'f' = 5, TRISA is loaded. When 'f' = 6, TRISB is loaded. When 'f' = 7, TRISC is loaded. |

| XORLW | Exclusive OR literal with W |
|------------------|---|
| Syntax: | [<i>label</i>] XORLW k |
| Operands: | $0 \leq k \leq 255$ |
| Operation: | (W) .XOR. k → (W) |
| Status Affected: | Z |
| Description: | The contents of W are XORed with the 8-bit literal 'k'. The result is placed in W. |

| XORWF | Exclusive OR W with f |
|------------------|---|
| Syntax: | [<i>label</i>] XORWF f, d |
| Operands: | $0 \leq f \leq 127$ $d \in [0,1]$ |
| Operation: | (W) .XOR. (f) → dest |
| 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'. |

38. ICSP™ - In-Circuit Serial Programming™

ICSP™ programming allows customers to manufacture circuit boards with unprogrammed devices. Programming can be done after the assembly process, allowing the device to be programmed with the most recent firmware or a custom firmware. Five pins are needed for ICSP™ programming:

- ICSPCLK
- ICSPDAT
- $\overline{\text{MCLR}}/V_{\text{PP}}$
- V_{DD}
- V_{SS}

In Program/Verify mode the program memory, User IDs and the Configuration Words are programmed through serial communications. The ICSPDAT pin is a bidirectional I/O used for transferring the serial data and the ICSPCLK pin is the clock input. For more information on ICSP™ refer to “*Memory Programming Specification*” (DS40001970).

38.1 High-Voltage Programming Entry Mode

The device is placed into High-Voltage Programming Entry mode by holding the ICSPCLK and ICSPDAT pins low then raising the voltage on $\overline{\text{MCLR}}/V_{\text{PP}}$ to V_{IH} .

38.2 Low-Voltage Programming Entry Mode

The Low-Voltage Programming Entry mode allows the PIC® Flash MCUs to be programmed using V_{DD} only, without high voltage. When the LVP bit of [Configuration Word 4](#) is set to ‘1’, the low-voltage ICSP programming entry is enabled. To disable the Low-Voltage ICSP mode, the LVP bit must be programmed to ‘0’.

Entry into the Low-Voltage Programming Entry mode requires the following steps:

1. $\overline{\text{MCLR}}$ is brought to V_{IL} .
2. A 32-bit key sequence is presented on ICSPDAT, while clocking ICSPCLK.

Once the key sequence is complete, $\overline{\text{MCLR}}$ must be held at V_{IL} for as long as Program/Verify mode is to be maintained.

If low-voltage programming is enabled ($\text{LVP} = 1$), the $\overline{\text{MCLR}}$ Reset function is automatically enabled and cannot be disabled. See the $\overline{\text{MCLR}}$ Section for more information.

The LVP bit can only be reprogrammed to ‘0’ by using the High-Voltage Programming mode.

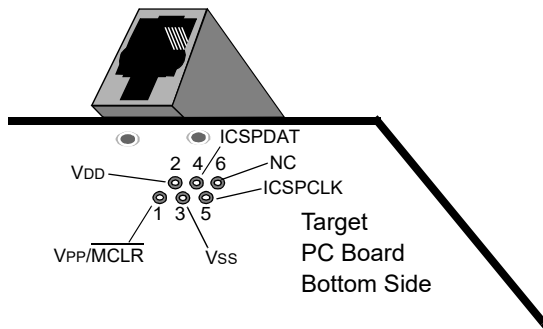
Related Links

[10.4 MCLR Reset](#)

38.3 Common Programming Interfaces

Connection to a target device is typically done through an ICSP™ header. A commonly found connector on development tools is the RJ-11 in the 6P6C (6-pin, 6-connector) configuration. See [Figure 38-1](#).

Figure 38-1. ICD RJ-11 Style Connector Interface



Pin Description*

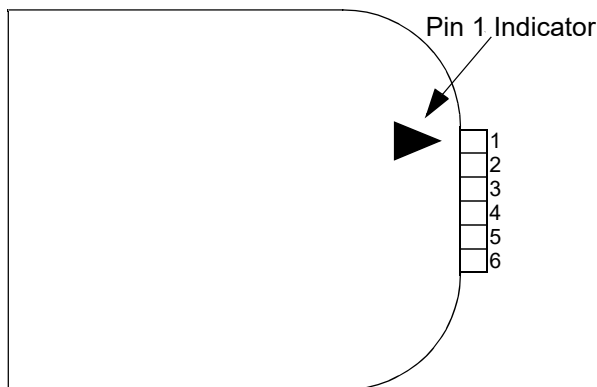
- 1 = V_{PP}/\overline{MCLR}
- 2 = V_{DD} Target
- 3 = V_{SS} (ground)
- 4 = ICSPDAT
- 5 = ICSPCLK
- 6 = No Connect

Another connector often found in use with the PICkit™ programmers is a standard 6-pin header with 0.1 inch spacing. Refer to [Figure 38-2](#).

For additional interface recommendations, refer to the specific device programmer manual prior to PCB design.

It is recommended that isolation devices be used to separate the programming pins from other circuitry. The type of isolation is highly dependent on the specific application and may include devices such as resistors, diodes, or even jumpers. See [Figure 38-3](#) for more information.

Figure 38-2. PICkit™ Programmer Style Connector Interface



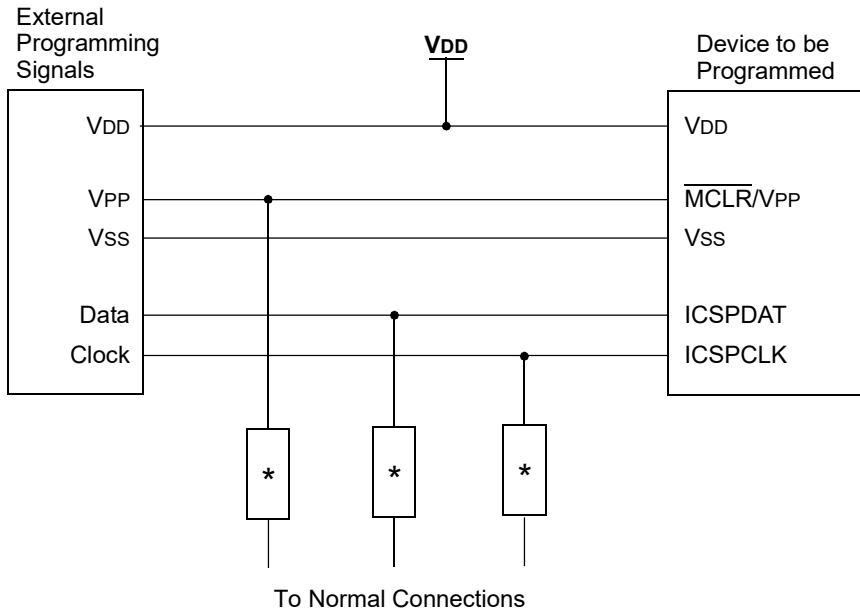
Pin Description⁽¹⁾

- 1 = V_{PP}/\overline{MCLR}
- 2 = V_{DD} Target
- 3 = V_{SS} (ground)
- 4 = ICSPDAT
- 5 = ICSPCLK
- 6 = No Connect

Note:

1. **Note:** The 6-pin header (0.100" spacing) accepts 0.025" square pins.

Figure 38-3. Typical Connection for ICSP™ Programming



* Isolation devices (as required).

39. Development Support

The PIC[®] microcontrollers (MCU) and dsPIC[®] digital signal controllers (DSC) are supported with a full range of software and hardware development tools:

- Integrated Development Environment
 - MPLAB[®] X IDE Software
- Compilers/Assemblers/Linkers
 - MPLAB XC Compiler
 - MPASM[™] Assembler
 - MPLINK[™] Object Linker/
MPLIB[™] Object Librarian
 - MPLAB Assembler/Linker/Librarian for
Various Device Families
- Simulators
 - MPLAB X SIM Software Simulator
- Emulators
 - MPLAB REAL ICE[™] In-Circuit Emulator
- In-Circuit Debuggers/Programmers
 - MPLAB ICD 3
 - PICkit[™] 3 In-Circuit Debugger
- Device Programmers
 - MPLAB PM3 Device Programmer
- Low-Cost Demonstration/Development Boards, Evaluation Kits and Starter Kits
- Third-party development tools

39.1 MPLAB X Integrated Development Environment Software

The MPLAB X IDE is a single, unified graphical user interface for Microchip and third-party software, and hardware development tool that runs on Windows[®], Linux and Mac OS[®] X. Based on the NetBeans IDE, MPLAB X IDE is an entirely new IDE with a host of free software components and plug-ins for high-performance application development and debugging. Moving between tools and upgrading from software simulators to hardware debugging and programming tools is simple with the seamless user interface.

With complete project management, visual call graphs, a configurable watch window and a feature-rich editor that includes code completion and context menus, MPLAB X IDE is flexible and friendly enough for new users. With the ability to support multiple tools on multiple projects with simultaneous debugging, MPLAB X IDE is also suitable for the needs of experienced users.

Feature-Rich Editor:

- Color syntax highlighting
- Smart code completion makes suggestions and provides hints as you type
- Automatic code formatting based on user-defined rules
- Live parsing

User-Friendly, Customizable Interface:

- Fully customizable interface: toolbars, toolbar buttons, windows, window placement, etc.
- Call graph window

Project-Based Workspaces:

- Multiple projects
- Multiple tools
- Multiple configurations

- Simultaneous debugging sessions

File History and Bug Tracking:

- Local file history feature
- Built-in support for Bugzilla issue tracker

39.2 MPLAB XC Compilers

The MPLAB XC Compilers are complete ANSI C compilers for all of Microchip's 8, 16, and 32-bit MCU and DSC devices. These compilers provide powerful integration capabilities, superior code optimization and ease of use. MPLAB XC Compilers run on Windows, Linux or MAC OS X.

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

The free MPLAB XC Compiler editions support all devices and commands, with no time or memory restrictions, and offer sufficient code optimization for most applications.

MPLAB XC Compilers include an assembler, linker and utilities. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. MPLAB XC Compiler uses the assembler to produce its object file. Notable features of the assembler include:

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

39.3 MPASM Assembler

The MPASM Assembler is a full-featured, universal macro assembler for PIC10/12/16/18 MCUs.

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

The MPASM Assembler features include:

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

39.4 MPLINK Object Linker/MPLIB Object Librarian

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

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

The object linker/library features include:

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

39.5 MPLAB Assembler, Linker and Librarian for Various Device Families

MPLAB Assembler produces relocatable machine code from symbolic assembly language for PIC24, PIC32 and dsPIC DSC devices. MPLAB XC Compiler uses the assembler to produce its object file. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. Notable features of the assembler include:

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

39.6 MPLAB X SIM Software Simulator

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

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

39.7 MPLAB REAL ICE In-Circuit Emulator System

The MPLAB REAL ICE In-Circuit Emulator System is Microchip's next generation high-speed emulator for Microchip Flash DSC and MCU devices. It debugs and programs all 8, 16 and 32-bit MCU, and DSC devices with the easy-to-use, powerful graphical user interface of the MPLAB X IDE.

The emulator is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with either a connector compatible with in-circuit debugger systems (RJ-11) or with the new high-speed, noise tolerant, Low-Voltage Differential Signal (LVDS) interconnection (CAT5).

The emulator is field upgradable through future firmware downloads in MPLAB X IDE. MPLAB REAL ICE offers significant advantages over competitive emulators including full-speed emulation, run-time variable watches, trace analysis, complex breakpoints, logic probes, a ruggedized probe interface and long (up to three meters) interconnection cables.

39.8 MPLAB ICD 3 In-Circuit Debugger System

The MPLAB ICD 3 In-Circuit Debugger System is Microchip's most cost-effective, high-speed hardware debugger/programmer for Microchip Flash DSC and MCU devices. It debugs and programs PIC Flash microcontrollers and dsPIC DSCs with the powerful, yet easy-to-use graphical user interface of the MPLAB IDE.

The MPLAB ICD 3 In-Circuit Debugger probe is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with a connector compatible with the MPLAB ICD 2 or MPLAB REAL ICE systems (RJ-11). MPLAB ICD 3 supports all MPLAB ICD 2 headers.

39.9 PICkit 3 In-Circuit Debugger/Programmer

The MPLAB PICkit 3 allows debugging and programming of PIC and dsPIC Flash microcontrollers at a most affordable price point using the powerful graphical user interface of the MPLAB IDE. The MPLAB PICkit 3 is connected to the design engineer's PC using a full-speed USB interface and can be connected to the target via a

Microchip debug (RJ-11) connector (compatible with MPLAB ICD 3 and MPLAB REAL ICE). The connector uses two device I/O pins and the Reset line to implement in-circuit debugging and In-Circuit Serial Programming™ (ICSP™).

39.10 MPLAB PM3 Device Programmer

The MPLAB PM3 Device Programmer is a universal, CE compliant device programmer with programmable voltage verification at V_{DDMIN} and V_{DDMAX} for maximum reliability. It features a large LCD display (128 x 64) for menus and error messages, and a modular, detachable socket assembly to support various package types. The ICSP cable assembly is included as a standard item. In Stand-Alone mode, the MPLAB PM3 Device Programmer can read, verify and program PIC devices without a PC connection. It can also set code protection in this mode. The MPLAB PM3 connects to the host PC via an RS-232 or USB cable. The MPLAB PM3 has high-speed communications and optimized algorithms for quick programming of large memory devices, and incorporates an MMC card for file storage and data applications.

39.11 Demonstration/Development Boards, Evaluation Kits, and Starter Kits

A wide variety of demonstration, development and evaluation boards for various PIC MCUs and dsPIC DSCs allows quick application development on fully functional systems. Most boards include prototyping areas for adding custom circuitry and provide application firmware and source code for examination and modification.

The boards support a variety of features, including LEDs, temperature sensors, switches, speakers, RS-232 interfaces, LCD displays, potentiometers and additional EEPROM memory.

The demonstration and development boards can be used in teaching environments, for prototyping custom circuits and for learning about various microcontroller applications.

In addition to the PICDEM™ and dsPICDEM™ demonstration/development board series of circuits, Microchip has a line of evaluation kits and demonstration software for analog filter design, KeeLoq® security ICs, CAN, IrDA®, PowerSmart battery management, SEEVAL® evaluation system, Sigma-Delta ADC, flow rate sensing, plus many more.

Also available are starter kits that contain everything needed to experience the specified device. This usually includes a single application and debug capability, all on one board.

Check the Microchip webpage (www.microchip.com) for the complete list of demonstration, development and evaluation kits.

39.12 Third-Party Development Tools

Microchip also offers a great collection of tools from third-party vendors. These tools are carefully selected to offer good value and unique functionality.

- Device Programmers and Gang Programmers from companies, such as SoftLog and CCS
- Software Tools from companies, such as Gimpel and Trace Systems
- Protocol Analyzers from companies, such as Saleae and Total Phase
- Demonstration Boards from companies, such as MikroElektronika, Digilent® and Olimex
- Embedded Ethernet Solutions from companies, such as EZ Web Lynx, WIZnet and IPLogika®

40. Electrical Specifications

40.1 Absolute Maximum Ratings^(†)

| Parameter | Rating |
|---|--|
| Ambient temperature under bias | -40°C to +125°C |
| Storage temperature | -65°C to +150°C |
| Voltage on pins with respect to V_{SS} | |
| • on V _{DD} pin: | |
| | PIC16LF18426/46 PIC16F18426/46 |
| | -0.3V to +4.0V -0.3V to +6.5V |
| • on MCLR pin: | -0.3V to +9.0V |
| • on all other pins: | -0.3V to (V _{DD} + 0.3V) |
| Maximum current | |
| • on V _{SS} pin ⁽¹⁾ | -40°C ≤ T _A ≤ +85°C 85°C < T _A ≤ +125°C |
| | 250 mA 120 mA |
| • on V _{DD} pin ⁽¹⁾ | -40°C ≤ T _A ≤ +85°C 85°C < T _A ≤ +125°C |
| | 250 mA 85 mA |
| • on any standard I/O pin | ±50 mA |
| Clamp current, I _K (V _{PIN} < 0 or V _{PIN} > V _{DD}) | ±20 mA |
| Total power dissipation ⁽²⁾ | 800 mW |



Important:

- Maximum current rating requires even load distribution across I/O pins. Maximum current rating may be limited by the device package power dissipation characterizations, see [Thermal Characteristics](#) to calculate device specifications.
- Power dissipation is calculated as follows:

$$P_{DIS} = V_{DD} \times \{I_{DD} - \sum I_{OH}\} + \sum \{(V_{DD} - V_{OH}) \times I_{OH}\} + \sum (V_{OI} \times I_{OL})$$
- Internal Power Dissipation is calculated as follows: P_{INTERNAL} = I_{DD} × V_{DD} where I_{DD} is current to run the chip alone without driving any load on the output pins.
- I/O Power Dissipation is calculated as follows: P_{I/O} = Σ(I_{OL} × V_{OL}) + Σ(I_{OH} × (V_{DD} - V_{OH}))
- Derated Power is calculated as follows: P_{DER} = P_{MAX}(T_J - T_A) / θ_{JA} where T_A = Ambient Temperature, T_J = Junction Temperature.

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.

40.2 Standard Operating Conditions

The standard operating conditions for any device are defined as:

Operating Voltage:

$$V_{DDMIN} \leq V_{DD} \leq V_{DDMAX}$$

PIC16(L)F18426/46

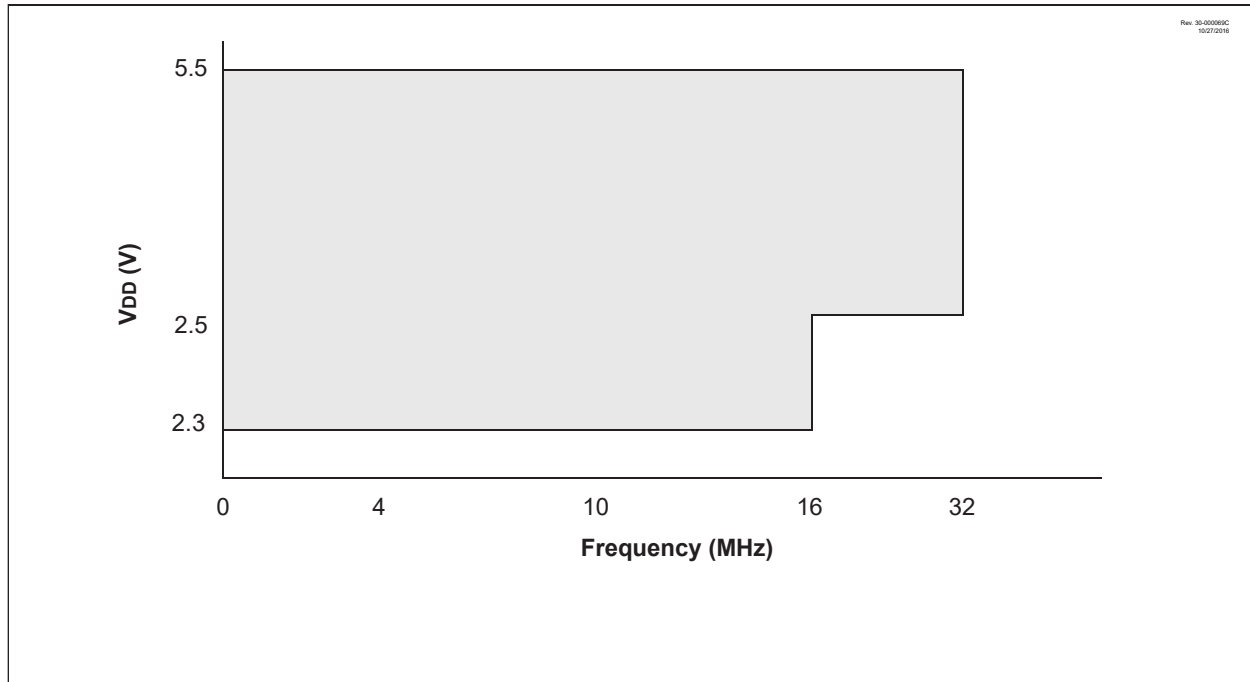
Electrical Specifications

$$T_{A_MIN} \leq T_A \leq T_{A_MAX}$$

Operating Temperature:

| Parameter | | Ratings |
|--|--|---------|
| V_{DD} — Operating Supply Voltage⁽¹⁾ | | |
| PIC16LF18426/46 | V _{DDMIN} (F _{OSC} ≤ 16 MHz) | +1.8V |
| | V _{DDMIN} (F _{OSC} ≤ 32 MHz) | +2.5V |
| | V _{DDMAX} | +3.6V |
| PIC16F18426/46 | V _{DDMIN} (F _{OSC} ≤ 16 MHz) | +2.3V |
| | V _{DDMIN} (F _{OSC} ≤ 32 MHz) | +2.5V |
| | V _{DDMAX} | +5.5V |
| T_A — Operating Ambient Temperature Range | | |
| Industrial Temperature | T _{A_MIN} | -40°C |
| | T _{A_MAX} | +85°C |
| Extended Temperature | T _{A_MIN} | -40°C |
| | T _{A_MAX} | +125°C |
| Note: | | |
| 1. See Parameter D002, DC Characteristics: Supply Voltage. | | |

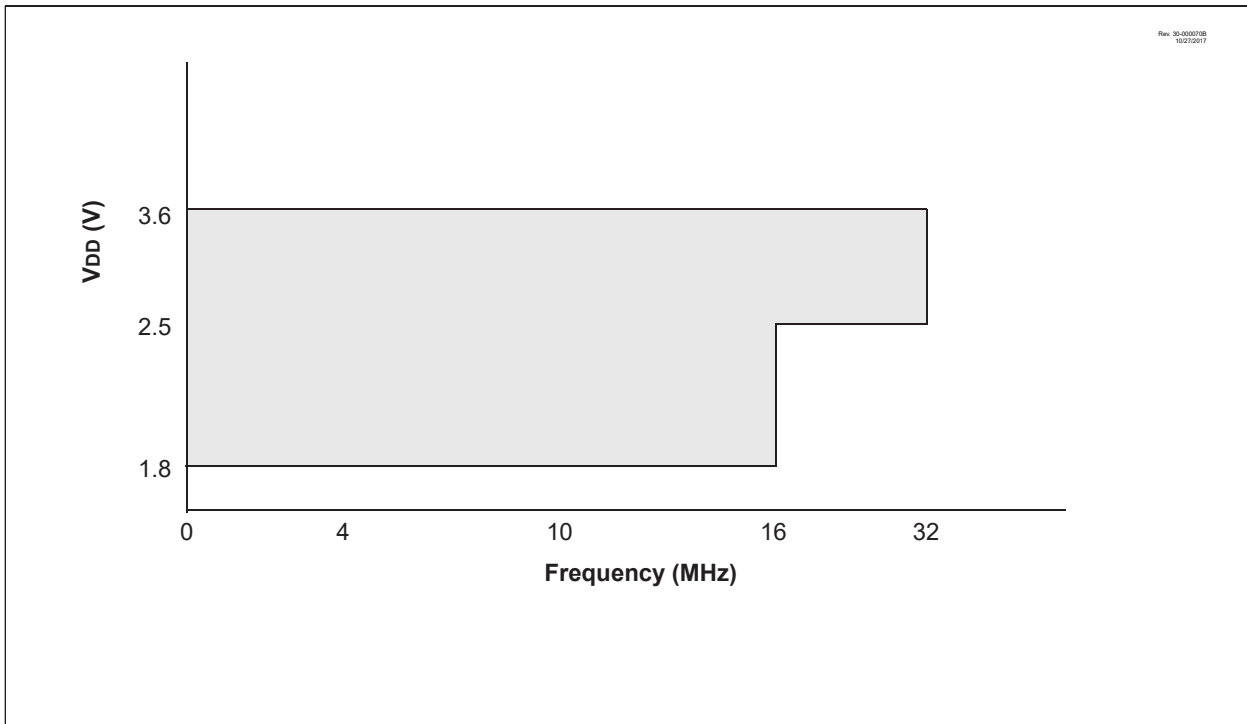
Figure 40-1. Voltage Frequency Graph, -40°C ≤ T_A ≤ +125°C, for PIC16F18426/46 only



Note:

1. The shaded region indicates the permissible combinations of voltage and frequency.
2. Refer to [40.4.1 External Clock/Oscillator Timing Requirements](#) for each Oscillator mode's supported frequencies.

Figure 40-2. Voltage Frequency Graph, $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$, for PIC16LF18426/46 Devices only



Note:

1. The shaded region indicates the permissible combinations of voltage and frequency.
2. Refer to [40.4.1 External Clock/Oscillator Timing Requirements](#) for each Oscillator mode's supported frequencies.

Related Links

[40.3.1 Supply Voltage](#)

40.3 DC Characteristics

40.3.1 Supply Voltage

Table 40-1.

| PIC16LF18426/46 only | | | | | | | |
|---|-----------------|----------------|------|-------|------|-------|---------------------------|
| Standard Operating Conditions (unless otherwise stated) | | | | | | | |
| Param. No. | Sym. | Characteristic | Min. | Typ.† | Max. | Units | Conditions |
| Supply Voltage | | | | | | | |
| D002 | V _{DD} | | 1.8 | — | 3.6 | V | F _{OSC} ≤ 16 MHz |
| | | | 2.5 | — | 3.6 | V | F _{OSC} > 16 MHz |
| RAM Data Retention⁽¹⁾ | | | | | | | |
| D003 | V _{DR} | | 1.5 | — | — | V | Device in Sleep mode |
| Power-on Reset Release Voltage⁽²⁾ | | | | | | | |

PIC16(L)F18426/46

Electrical Specifications

|continued | | | | | | | |
|--|-------------------|----------------|------|-------|------|-------|--------------------------------------|
| PIC16LF18426/46 only | | | | | | | |
| Standard Operating Conditions (unless otherwise stated) | | | | | | | |
| Param. No. | Sym. | Characteristic | Min. | Typ.† | Max. | Units | Conditions |
| D004 | V _{POR} | | — | 1.6 | — | V | BOR or LPBOR disabled ⁽³⁾ |
| Power-on Reset Rearm Voltage⁽²⁾ | | | | | | | |
| D005 | V _{PORR} | | — | 0.8 | — | V | BOR or LPBOR disabled ⁽³⁾ |
| V_{DD} Rise Rate to ensure internal Power-on Reset signal⁽²⁾ | | | | | | | |
| D006 | S _{VDD} | | 0.05 | — | — | V/ms | BOR or LPBOR disabled ⁽³⁾ |
| † - Data in "Typ." column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested. | | | | | | | |
| Note: | | | | | | | |
| 1. This is the limit to which V _{DD} can be lowered in Sleep mode without losing RAM data. | | | | | | | |
| 2. See the following figure, POR and POR REARM with Slow Rising V _{DD} . | | | | | | | |
| 3. Please see 40.4.5 Reset, WDT, Oscillator Start-up Timer, Power-up Timer, Brown-Out Reset and Low-Power Brown-Out Reset Specifications for BOR and LPBOR trip point information. | | | | | | | |

| PIC16F18426/46 only | | | | | | | |
|--|-------------------|----------------|------|-------|------|-------|--------------------------------------|
| Standard Operating Conditions (unless otherwise stated) | | | | | | | |
| Param. No. | Sym. | Characteristic | Min. | Typ.† | Max. | Units | Conditions |
| Supply Voltage | | | | | | | |
| D002 | V _{DD} | | 2.3 | — | 5.5 | V | F _{OSC} ≤ 16 MHz |
| | | | 2.5 | — | 5.5 | V | F _{OSC} > 16 MHz |
| RAM Data Retention⁽¹⁾ | | | | | | | |
| D003 | V _{DR} | | 1.7 | — | — | V | Device in Sleep mode |
| Power-on Reset Release Voltage⁽²⁾ | | | | | | | |
| D004 | V _{POR} | | — | 1.6 | — | V | BOR or LPBOR disabled ⁽³⁾ |
| Power-on Reset Rearm Voltage⁽²⁾ | | | | | | | |
| D005 | V _{PORR} | | — | 1.5 | — | V | BOR or LPBOR disabled ⁽³⁾ |
| V_{DD} Rise Rate to ensure internal Power-on Reset signal⁽²⁾ | | | | | | | |
| D006 | S _{VDD} | | 0.05 | — | — | V/ms | BOR or LPBOR disabled ⁽³⁾ |

.....continued

PIC16F18426/46 only

Standard Operating Conditions (unless otherwise stated)

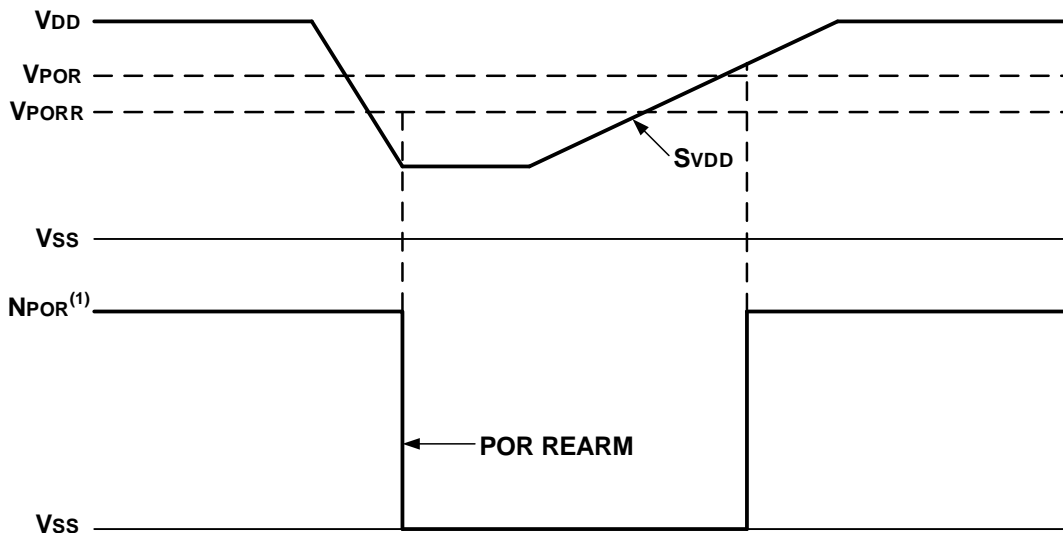
| Param. No. | Sym. | Characteristic | Min. | Typ.† | Max. | Units | Conditions |
|------------|------|----------------|------|-------|------|-------|------------|
|------------|------|----------------|------|-------|------|-------|------------|

† - Data in "Typ." column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note:

1. This is the limit to which V_{DD} can be lowered in Sleep mode without losing RAM data.
2. See the following figure, POR and POR REARM with Slow Rising V_{DD} .
3. Please see [40.4.5 Reset, WDT, Oscillator Start-up Timer, Power-up Timer, Brown-Out Reset and Low-Power Brown-Out Reset Specifications](#) for BOR and LPBOR trip point information.

Figure 40-3. POR and POR Rearm with Slow Rising V_{DD}



Note:

1. When $NPOR$ is low, the device is held in Reset.

40.3.2 Supply Current (I_{DD})^(1,2,4)

Table 40-2.

| PIC16LF18426/46 only | | | | | | | | |
|---|------------------|------------------------|------|-------|------|---------|------------|------|
| Standard Operating Conditions (unless otherwise stated) | | | | | | | | |
| Param. No. | Sym. | Device Characteristics | Min. | Typ.† | Max. | Units | Conditions | |
| | | | | | | | V_{DD} | Note |
| D100 | $I_{DD_{XT4}}$ | XT = 4 MHz | — | 465 | 670 | μA | 3.0V | |
| D101 | $I_{DD_{HFO16}}$ | HFINTOSC = 16 MHz | — | 2.1 | 2.6 | mA | 3.0V | |

PIC16(L)F18426/46

Electrical Specifications

.....continued

PIC16LF18426/46 only

Standard Operating Conditions (unless otherwise stated)

| Param. No. | Sym. | Device Characteristics | Min. | Typ.† | Max. | Units | Conditions | |
|------------|-------------------------------------|---|------|-------|------|-------|-----------------|------|
| | | | | | | | V _{DD} | Note |
| D102 | I _{DD_HFOPLL} | HFINTOSC = 32 MHz | — | 3.9 | 4.8 | mA | 3.0V | |
| D103 | I _{DD_HSPLL32} | HS+PLL = 32 MHz | — | 3.6 | 4.4 | mA | 3.0V | |
| D104 | I _{DD_IDLE} | IDLE mode, HFINTOSC = 16 MHz | — | 1.5 | 1.9 | mA | 3.0V | |
| D105 | I _{DD_DOZE} ⁽³⁾ | DOZE mode, HFINTOSC = 16 MHz, Doze Ratio = 16 | — | 1.5 | — | mA | 3.0V | |

† - Data in "Typ." column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note:

1. The test conditions for all I_{DD} measurements in active operation mode are: OSC1 = external square wave, from rail-to-rail; all I/O pins are outputs driven low; $\overline{MCLR} = V_{DD}$; WDT disabled.
2. The supply current is mainly a function of the operating voltage and frequency. Other factors, such as I/O pin loading and switching rate, oscillator type, internal code execution pattern and temperature, also have an impact on the current consumption.
3. $I_{DD_{DOZE}} = [I_{DD_{IDLE}} * (N-1)/N] + I_{DD_{HFO}} * 16/N$ where N = DOZE Ratio (see *CPUDOZE* register).
4. PMD bits are all in the default state, no modules are disabled.

PIC16F18426/46 only

Standard Operating Conditions (unless otherwise stated)

| Param. No. | Sym. | Device Characteristics | Min. | Typ.† | Max. | Units | Conditions | |
|------------|-------------------------------------|---|------|-------|------|-------|-----------------|------|
| | | | | | | | V _{DD} | Note |
| D100 | I _{DD_XT4} | XT = 4 MHz | — | 515 | 710 | µA | 3.0V | |
| D101 | I _{DD_HFO16} | HFINTOSC = 16 MHz | — | 2.2 | 2.7 | mA | 3.0V | |
| D102 | I _{DD_HFOPLL} | HFINTOSC = 32 MHz | — | 4.0 | 4.9 | mA | 3.0V | |
| D103 | I _{DD_HSPLL32} | HS+PLL = 32 MHz | — | 3.7 | 4.5 | mA | 3.0V | |
| D104 | I _{DD_IDLE} | IDLE mode, HFINTOSC = 16 MHz | — | 1.6 | 2.0 | mA | 3.0V | |
| D105 | I _{DD_DOZE} ⁽³⁾ | DOZE mode, HFINTOSC = 16 MHz, Doze Ratio = 16 | — | 1.6 | — | mA | 3.0V | |

.....continued

PIC16F18426/46 only

Standard Operating Conditions (unless otherwise stated)

| Param. No. | Sym. | Device Characteristics | Min. | Typ.† | Max. | Units | Conditions | |
|------------|------|------------------------|------|-------|------|-------|-----------------|------|
| | | | | | | | V _{DD} | Note |

† - Data in "Typ." column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note:

- The test conditions for all I_{DD} measurements in active operation mode are: OSC1 = external square wave, from rail-to-rail; all I/O pins are outputs driven low; $\overline{MCLR} = V_{DD}$; WDT disabled.
- 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.
- $I_{DD_{DOZE}} = [I_{DD_{IDLE}} * (N-1)/N] + I_{DD_{HFO}} 16/N$ where N = DOZE Ratio (see CPUDOZE register).
- PMD bits are all in the default state, no modules are disabled.

Related Links

[12.5.2 CPUDOZE](#)

40.3.3 Power-Down Current (I_{PD})^(1,2)

Table 40-3.

PIC16LF18426/46 only

Standard Operating Conditions (unless otherwise stated)

| Param. No. | Sym. | Device Characteristics | Min. | Typ.† | Max. +85°C | Max. +125°C | Units | Conditions | |
|------------|-----------------------|--|------|-------|------------|-------------|-------|-----------------|-----------------------------------|
| | | | | | | | | V _{DD} | Note |
| D200 | I _{PD} | I _{PD} Base | — | 0.08 | 2.0 | 7 | µA | 3.0V | |
| D201 | I _{PD_WDT} | Low-Frequency Internal Oscillator/WDT | — | 0.8 | 2.8 | 8 | µA | 3.0V | |
| D202 | I _{PD_SOSC} | Secondary Oscillator (S _O SC) | — | 1.0 | 3.8 | 9 | µA | 3.0V | |
| D203 | I _{PD_FVR} | FVR | — | 46 | 76 | 77 | µA | 3.0V | |
| D204 | I _{PD_BOR} | Brown-out Reset (BOR) | — | 10 | 15 | 18 | µA | 3.0V | |
| D205 | I _{PD_LPBOR} | Low-Power Brown-out Reset (LPBOR) | — | 0.13 | 2.2 | 8 | µA | 3.0V | |
| D207 | I _{PD_ADCA} | ADC - Non-converting | — | 0.08 | 2.0 | 7.0 | µA | 3.0V | ADC not converting ⁽⁴⁾ |
| D208 | I _{PD_CMP} | Comparator | — | 30 | 57 | 58 | µA | 3.0V | |

PIC16(L)F18426/46

Electrical Specifications

.....continued

PIC16LF18426/46 only

Standard Operating Conditions (unless otherwise stated)

| Param. No. | Sym. | Device Characteristics | Min. | Typ.† | Max. +85°C | Max. +125°C | Units | Conditions | |
|------------|------|------------------------|------|-------|------------|-------------|-------|-----------------|------|
| | | | | | | | | V _{DD} | Note |

† - Data in "Typ." column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note:

1. The peripheral current is the sum of the base I_{DD} and the additional current consumed when this peripheral is enabled. The peripheral Δ current can be determined by subtracting the base I_{DD} or I_{PD} current from this limit. Max. values should be used when calculating total current consumption.
2. The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode with all I/O pins in high-impedance state and tied to V_{SS}.
3. All peripheral currents listed are on a per-peripheral basis if more than one instance of a peripheral is available.
4. ADC clock source is FRC.

PIC16F18426/46 only

Standard Operating Conditions (unless otherwise stated), VREGPM = 1

| Param. No. | Sym. | Device Characteristics | Min. | Typ.† | Max. +85°C | Max. +125°C | Units | Conditions | |
|------------|----------------------|--|------|-------|------------|-------------|-------|-----------------|-----------------------------------|
| | | | | | | | | V _{DD} | Note |
| D200 | I _{PD} | I _{PD} Base | — | 0.40 | 2.5 | 8 | μA | 3.0V | |
| D200A | | | — | 18 | 25 | 30 | μA | 3.0V | VREGPM = 0 |
| D201 | I _{PD_WDT} | Low-Frequency Internal Oscillator/WDT | — | 1.0 | 2.9 | 9 | μA | 3.0V | |
| D202 | I _{PD_SOSC} | Secondary Oscillator (S _O SC) | — | 1.2 | 4.3 | 9.2 | μA | 3.0V | |
| D203 | I _{PD_FVR} | FVR | — | 40 | 69 | 70 | μA | 3.0V | |
| D204 | I _{PD_BOR} | Brown-out Reset (BOR) | — | 11 | 16 | 19 | μA | 3.0V | |
| D207 | I _{PD_ADCA} | ADC - Non-converting | — | 0.38 | 2.5 | 8.0 | μA | 3.0V | ADC not converting ⁽⁴⁾ |
| D208 | I _{PD_CMP} | Comparator | — | 31 | 58 | 59 | μA | 3.0V | |

† - Data in "Typ." column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note:

1. The peripheral current is the sum of the base I_{DD} and the additional current consumed when this peripheral is enabled. The peripheral Δ current can be determined by subtracting the base I_{DD} or I_{PD} current from this limit. Max. values should be used when calculating total current consumption.
2. The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode with all I/O pins in high-impedance state and tied to V_{SS}.
3. All peripheral currents listed are on a per-peripheral basis if more than one instance of a peripheral is available.
4. ADC clock source is FRC.

PIC16(L)F18426/46

Electrical Specifications

40.3.4 I/O Ports

Table 40-4.

| Standard Operating Conditions (unless otherwise stated) | | | | | | | |
|---|------------------|--------------------------------|----------------------------|-------|----------------------|-------|---|
| Param. No. | Sym. | Device Characteristics | Min. | Typ.† | Max. | Units | Conditions |
| Input Low Voltage | | | | | | | |
| | V _{IL} | I/O PORT: | | | | | |
| D300 | | • with TTL buffer | — | — | 0.8 | V | 4.5V ≤ V _{DD} ≤ 5.5V |
| D301 | | | — | — | 0.15 V _{DD} | V | 1.8V ≤ V _{DD} ≤ 4.5V |
| D302 | | • with Schmitt Trigger buffer | — | — | 0.2 V _{DD} | V | 2.0V ≤ V _{DD} ≤ 5.5V |
| D303 | | • with I ² C levels | — | — | 0.3 V _{DD} | V | |
| D304 | | • with SMBus levels | — | — | 0.8 | V | 2.7V ≤ V _{DD} ≤ 5.5V |
| D305 | | MCLR | — | — | 0.2 V _{DD} | V | |
| Input High Voltage | | | | | | | |
| | V _{IH} | I/O PORT: | | | | | |
| D320 | | • with TTL buffer | 2.0 | — | — | V | 4.5V ≤ V _{DD} ≤ 5.5V |
| D321 | | | 0.25 V _{DD} + 0.8 | — | — | V | 1.8V ≤ V _{DD} ≤ 4.5V |
| D322 | | • with Schmitt Trigger buffer | 0.8V _{DD} | — | — | V | 2.0V ≤ V _{DD} ≤ 5.5V |
| D323 | | • with I ² C levels | 0.7 V _{DD} | — | — | V | |
| D324 | | • with SMBus levels | 2.1 | — | — | V | 2.7V ≤ V _{DD} ≤ 5.5V |
| D325 | | MCLR | 0.7 V _{DD} | — | — | V | |
| Input Leakage Current⁽¹⁾ | | | | | | | |
| D340 | I _{IL} | I/O PORTS | — | ±5 | ±125 | nA | V _{SS} ≤ V _{PIN} ≤ V _{DD} , Pin at high-impedance, 85°C |
| D341 | | | — | ±5 | ±1000 | nA | V _{SS} ≤ V _{PIN} ≤ V _{DD} , Pin at high-impedance, 125°C |
| D342 | | MCLR ⁽²⁾ | — | ±50 | ±200 | nA | V _{SS} ≤ V _{PIN} ≤ V _{DD} , Pin at high-impedance, 85°C |
| Weak Pull-up Current | | | | | | | |
| D350 | I _{PUR} | | 25 | 120 | 200 | μA | V _{DD} = 3.0V, V _{PIN} = V _{SS} |
| Output Low Voltage | | | | | | | |
| D360 | V _{OL} | Standard I/O PORTS | — | — | 0.6 | V | I _{OL} = 10 mA, V _{DD} = 3.0V |
| Output High Voltage | | | | | | | |
| D370 | V _{OH} | Standard I/O PORTS | V _{DD} - 0.7 | — | — | V | I _{OH} = 6 mA, V _{DD} = 3.0V |
| All I/O Pins | | | | | | | |
| D380 | C _{IO} | | — | 5 | 50 | pF | |

PIC16(L)F18426/46

Electrical Specifications

.....continued

Standard Operating Conditions (unless otherwise stated)

| Param. No. | Sym. | Device Characteristics | Min. | Typ.† | Max. | Units | Conditions |
|------------|------|------------------------|------|-------|------|-------|------------|
|------------|------|------------------------|------|-------|------|-------|------------|

† - Data in "Typ" column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note:

1. Negative current is defined as current sourced by the pin.
2. The leakage current on the $\overline{\text{MCLR}}$ pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.

40.3.5 Memory Programming Specifications

Table 40-5.

| Standard Operating Conditions (unless otherwise stated) | | | | | | | |
|---|---------------------|---|--------------------|------|--------------------|-------|--|
| Param No. | Sym. | Device Characteristics | Min. | Typ† | Max. | Units | Conditions |
| High Voltage Entry Programming Mode Specifications | | | | | | | |
| MEM01 | V_{IHH} | Voltage on $\overline{\text{MCLR}}/V_{\text{PP}}$ pin to enter programming mode | 8 | — | 9 | V | (Note 2, Note 3) |
| MEM02 | I_{PPGM} | Current on $\overline{\text{MCLR}}/V_{\text{PP}}$ pin during programming mode | — | 1 | — | mA | (Note 2) |
| Programming Mode Specifications | | | | | | | |
| MEM10 | V_{BE} | V_{DD} for Bulk Erase | — | — | — | V | (Note 4) |
| MEM11 | I_{DDPGM} | Supply Current during Programming operation | — | — | 10 | mA | |
| Data EEPROM Memory Specifications | | | | | | | |
| MEM20 | E_{D} | DataEE Byte Endurance | 100k | — | — | E/W | $-40^{\circ}\text{C} \leq T_{\text{A}} \leq +85^{\circ}\text{C}$ |
| MEM21 | $T_{\text{D_RET}}$ | Characteristic Retention | — | 40 | — | Year | Provided no other specifications are violated |
| MEM22 | $N_{\text{D_REF}}$ | Total Erase/Write Cycles before Refresh | — | — | 100k | E/W | |
| MEM23 | $V_{\text{D_RW}}$ | V_{DD} for Read or Erase/Write operation | V_{DDMIN} | — | V_{DDMAX} | V | |
| MEM24 | $T_{\text{D_BEW}}$ | Byte Erase and Write Cycle Time | — | 4.0 | 5.0 | ms | |
| Program Flash Memory Specifications | | | | | | | |
| MEM30 | E_{P} | Flash Memory Cell Endurance | 10k | — | — | E/W | $-40^{\circ}\text{C} \leq T_{\text{A}} \leq +85^{\circ}\text{C}$ (Note 1) |

.....continued

| Standard Operating Conditions (unless otherwise stated) | | | | | | | |
|---|--------------------|--|--------------------|------|--------------------|-------|---|
| Param No. | Sym. | Device Characteristics | Min. | Typ† | Max. | Units | Conditions |
| MEM32 | T _{P_RET} | Characteristic Retention | — | 40 | — | Year | Provided no other specifications are violated |
| MEM33 | V _{P_RD} | V _{DD} for Read operation | V _{DDMIN} | — | V _{DDMAX} | V | |
| MEM34 | V _{P_REW} | V _{DD} for Row Erase or Write operation | V _{DDMIN} | — | V _{DDMAX} | V | |
| MEM35 | T _{P_REW} | Self-Timed Row Erase or Self-Timed Write | — | 2.0 | 2.8 | ms | |

† - Data in “Typ” column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note:

- Flash Memory Cell Endurance for the Flash memory is defined as: One Row Erase operation and one Self-Timed Write.
- Required only if CONFIG4, bit LVP is disabled.
- The MPLAB® ICD2 does not support variable V_{PP} output. Circuitry to limit the ICD2 V_{PP} voltage must be placed between the ICD2 and target system when programming or debugging with the ICD2.
- Refer to the “PIC16(L)F184XX Memory Programming Specification” document for description.

Related Links

[4.7.4 CONFIG4](#)

40.3.6 Thermal Characteristics

Table 40-6.

| Standard Operating Conditions (unless otherwise stated) | | | | | |
|---|-----------------|--|-------|-------|---------------------------|
| Operating temperature -40°C ≤ TA ≤ +125°C | | | | | |
| Param No. | Sym. | Characteristic | Typ. | Units | Conditions |
| TH01 | θ _{JA} | Thermal Resistance Junction to Ambient | 70.0 | °C/W | 14-pin PDIP package |
| | | | 95.3 | °C/W | 14-pin SOIC package |
| | | | 100.0 | °C/W | 14-pin TSSOP package |
| | | | 51.5 | °C/W | 16-pin UQFN 4x4mm package |
| | | | 62.2 | °C/W | 20-pin PDIP package |
| | | | 87.3 | °C/W | 20-pin SSOP package |
| | | | 77.7 | °C/W | 20-pin SOIC package |
| | | | 43.0 | °C/W | 20-pin UQFN 4x4mm package |

.....continued

Standard Operating Conditions (unless otherwise stated)
 Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$

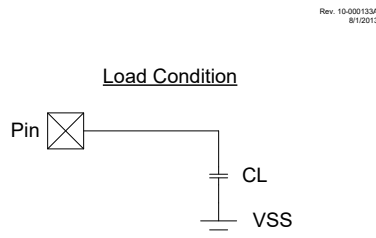
| Param No. | Sym. | Characteristic | Typ. | Units | Conditions |
|-----------|----------------|-------------------------------------|-------|-----------------------------|--|
| TH02 | θ_{JC} | Thermal Resistance Junction to Case | 32.75 | $^{\circ}\text{C}/\text{W}$ | 14-pin PDIP package |
| | | | 31.0 | $^{\circ}\text{C}/\text{W}$ | 14-pin SOIC package |
| | | | 24.4 | $^{\circ}\text{C}/\text{W}$ | 14-pin TSSOP package |
| | | | 5.4 | $^{\circ}\text{C}/\text{W}$ | 16-pin UQFN 4x4mm package |
| | | | 27.5 | $^{\circ}\text{C}/\text{W}$ | 20-pin PDIP package |
| | | | 31.1 | $^{\circ}\text{C}/\text{W}$ | 20-pin SSOP package |
| | | | 23.1 | $^{\circ}\text{C}/\text{W}$ | 20-pin SOIC package |
| | | | 5.3 | $^{\circ}\text{C}/\text{W}$ | 20-pin UQFN 4x4mm package |
| TH03 | T_{JMAX} | Maximum Junction Temperature | 150 | $^{\circ}\text{C}$ | |
| TH04 | PD | Power Dissipation | — | W | $PD = P_{INTERNAL} + P_{I/O}$ |
| TH05 | $P_{INTERNAL}$ | Internal Power Dissipation | — | W | $P_{INTERNAL} = I_{DD} \times V_{DD}^{(1)}$ |
| TH06 | $P_{I/O}$ | I/O Power Dissipation | — | W | $P_{I/O} = \Sigma(I_{OL} \times V_{OL}) + \Sigma(I_{OH} \times (V_{DD} - V_{OH}))$ |
| TH07 | P_{DER} | Derated Power | — | W | $P_{DER} = PD_{MAX} (T_J - T_A) / \theta_{JA}^{(2)}$ |

Note:

- I_{DD} is current to run the chip alone without driving any load on the output pins.
- T_A = Ambient Temperature, T_J = Junction Temperature.

40.4 AC Characteristics

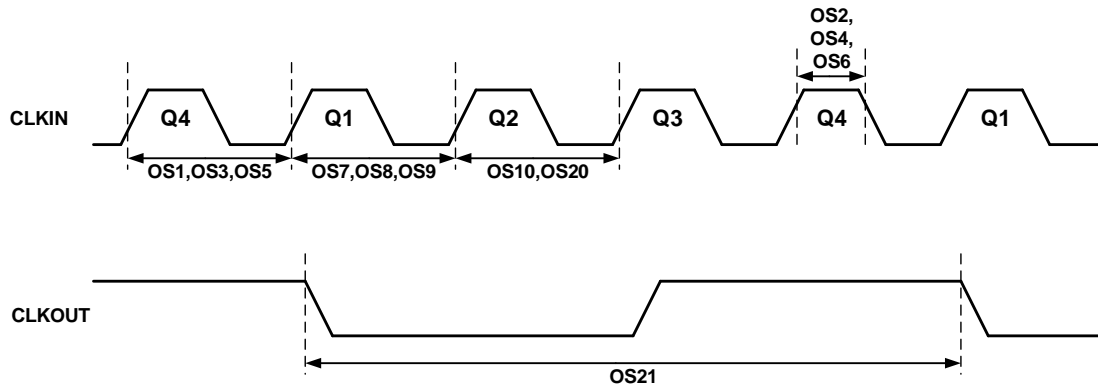
Figure 40-4. Load Conditions



Legend: CL=50 pF for all pins

40.4.1 External Clock/Oscillator Timing Requirements

Figure 40-5. Clock Timing



Note: See table below.

Table 40-7.

| Standard Operating Conditions (unless otherwise stated) | | | | | | | |
|---|---------------|------------------------|------|--------|------|-------|------------------|
| Param No. | Sym. | Characteristic | Min. | Typ. † | Max. | Units | Conditions |
| ECL Oscillator | | | | | | | |
| OS1 | F_{ECL} | Clock Frequency | — | — | 500 | kHz | |
| OS2 | T_{ECL_DC} | Clock Duty Cycle | 40 | — | 60 | % | |
| ECM Oscillator | | | | | | | |
| OS3 | F_{ECM} | Clock Frequency | — | — | 4 | MHz | |
| OS4 | T_{ECM_DC} | Clock Duty Cycle | 40 | — | 60 | % | |
| ECH Oscillator | | | | | | | |
| OS5 | F_{ECH} | Clock Frequency | — | — | 32 | MHz | |
| OS6 | T_{ECH_DC} | Clock Duty Cycle | 40 | — | 60 | % | |
| LP Oscillator | | | | | | | |
| OS7 | F_{LP} | Clock Frequency | — | — | 100 | kHz | Note 4 |
| XT Oscillator | | | | | | | |
| OS8 | F_{XT} | Clock Frequency | — | — | 4 | MHz | Note 4 |
| HS Oscillator | | | | | | | |
| OS9 | F_{HS} | Clock Frequency | — | — | 20 | MHz | Note 4 |
| Secondary Oscillator | | | | | | | |
| OS10 | F_{SEC} | Clock Frequency | 32.4 | 32.768 | 33.1 | kHz | Note 4 |
| System Oscillator | | | | | | | |
| OS20 | F_{OSC} | System Clock Frequency | — | — | 32 | MHz | (Note 2, Note 3) |

.....continued

| Standard Operating Conditions (unless otherwise stated) | | | | | | | |
|---|-----------------|-----------------------|------|---------------------|------|-------|------------|
| Param No. | Sym. | Characteristic | Min. | Typ. † | Max. | Units | Conditions |
| OS21 | F _{CY} | Instruction Frequency | — | F _{OSC} /4 | — | MHz | |
| OS22 | T _{CY} | Instruction Period | 125 | 1/F _{CY} | — | ns | |

* These parameters are characterized but not tested.
† - Data in "Typ" column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note:

1. Instruction cycle period (T_{CY}) 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.
2. The system clock frequency (F_{OSC}) is selected by the "main clock switch controls" as described in the "Oscillator Module (with Fail-Safe Clock Monitor)" section.
3. The system clock frequency (F_{OSC}) must meet the voltage requirements defined in the "Standard Operating Conditions" section.
4. LP, XT and HS oscillator modes require an appropriate crystal or resonator to be connected to the device. For clocking the device with the external square wave, one of the EC mode selections must be used.

Related Links

- [8. OSC - Oscillator Module](#)
- [40.2 Standard Operating Conditions](#)

40.4.2 Internal Oscillator Parameters⁽¹⁾

Table 40-8.

| Standard Operating Conditions (unless otherwise stated) | | | | | | | |
|---|----------------------|---|------------------------------|--------------------------|------------------------------|--------------------------|---|
| Param No. | Sym. | Characteristic | Min. | Typ. † | Max. | Units | Conditions |
| OS50 | F _{HFOSC} | Precision Calibrated HFINTOSC Frequency | — | 4 8 12 16 32 | — | MHz | (Note 2) |
| OS51 | F _{HFOSCLP} | Low-Power Optimized HFINTOSC Frequency | 0.92 1.84 0.88 1.76 | 1 2 1 2 | 1.08 2.16 1.12 2.24 | MHz MHz MHz MHz | -40°C to 85°C -40°C to 85°C -40°C to 125 °C -40°C to 125°C |
| OS52 | F _{MFOSC} | Internal Calibrated MFINTOSC Frequency | — | 500 | — | kHz | |
| OS53 | F _{LFOSC} | Internal LFINTOSC Frequency | — | 31 | — | kHz | |

.....continued

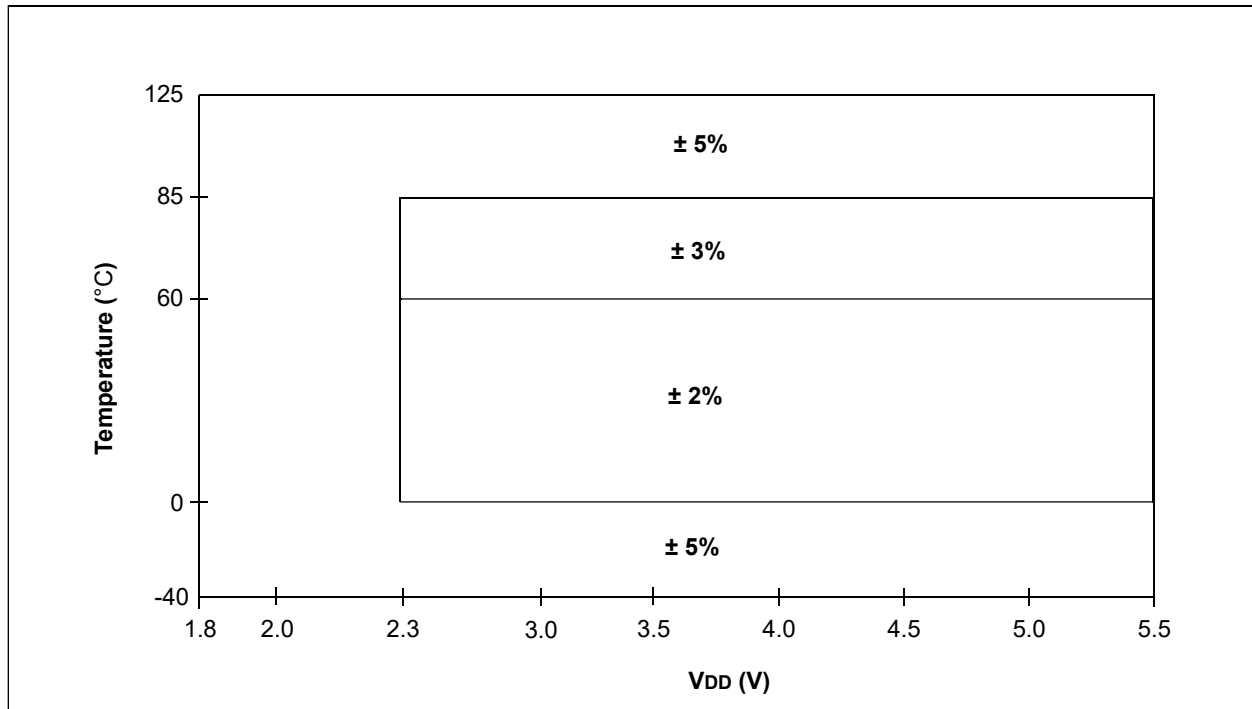
| Standard Operating Conditions (unless otherwise stated) | | | | | | | |
|---|----------------------|---|------|--------|------|-------|------------|
| Param No. | Sym. | Characteristic | Min. | Typ. † | Max. | Units | Conditions |
| OS54 | T _{HFOSCST} | HFINTOSC Wake-up from Sleep Start-up Time | — | 11 | 20 | μs | VREGPM=0 |
| | | | — | 85 | — | μs | VREGPM=1 |
| OS56 | T _{LFOSCST} | LFINTOSC Wake-up from Sleep Start-up Time | — | 0.2 | — | ms | |

† - Data in "Typ" column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note:

- To ensure these oscillator frequency tolerances, V_{DD} and V_{SS} must be capacitively decoupled as close to the device as possible. 0.1 μF and 0.01 μF values in parallel are recommended.
- See the figure below.

Figure 40-6. Precision Calibrated HFINTOSC Frequency Accuracy Over Device V_{DD} and Temperature



40.4.3 PLL Specifications

Table 40-9.

| Standard Operating Conditions (unless otherwise stated) | | | | | | | |
|---|---------------------|----------------------------|------|--------|------|-------|------------|
| Param No. | Sym. | Characteristic | Min. | Typ. † | Max. | Units | Conditions |
| PLL01 | F _{PLLIN} | PLL Input Frequency Range | 4 | — | 16 | MHz | |
| PLL02 | F _{PLLOUT} | PLL Output Frequency Range | 16 | — | 32 | MHz | (Note 1) |

.....continued

| Standard Operating Conditions (unless otherwise stated) | | | | | | | |
|---|---------------------|---|-------|--------|------|-------|------------|
| Param No. | Sym. | Characteristic | Min. | Typ. † | Max. | Units | Conditions |
| PLL03 | F _{PLLST} | PLL Lock Time from Start-up | — | 200 | — | μs | |
| PLL04 | F _{PLLJIT} | PLL Output Frequency Stability (Jitter) | -0.25 | — | 0.25 | % | |

* - 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:

- The output frequency of the PLL must meet the F_{OSC} requirements listed in Parameter D002.

40.4.4 I/O and CLKOUT Timing Specifications

Figure 40-7. CLKOUT and I/O Timing

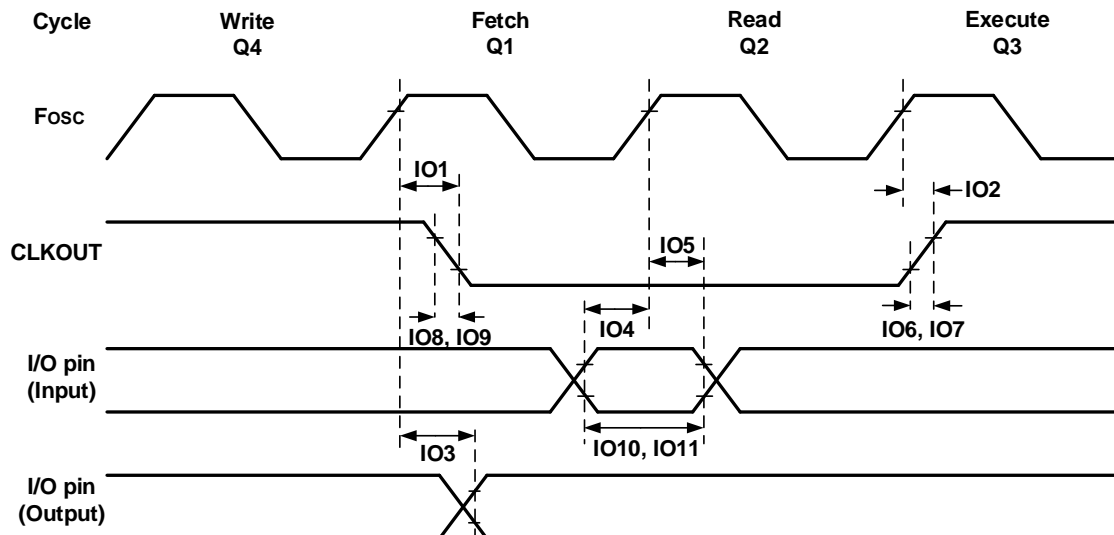


Table 40-10. I/O and CLKOUT Timing Specifications

| Standard Operating Conditions (unless otherwise stated) | | | | | | | |
|---|-----------------------|---|------|--------|------|-------|------------|
| Param No. | Sym. | Characteristic | Min. | Typ. † | Max. | Units | Conditions |
| IO1* | T _{CLKOUTH} | CLKOUT rising edge delay (rising edge F _{OSC} (Q1 cycle) to falling edge CLKOUT) | — | — | 70 | ns | |
| IO2* | T _{CLKOUTL} | CLKOUT falling edge delay (rising edge F _{OSC} (Q3 cycle) to rising edge CLKOUT) | — | — | 72 | ns | |
| IO3* | T _{IO_VALID} | Port output valid time (rising edge F _{OSC} (Q1 cycle) to port valid) | — | 50 | 70 | ns | |
| IO4* | T _{IO_SETUP} | Port input setup time (Setup time before rising edge F _{OSC} – Q2 cycle) | 20 | — | — | ns | |

.....continued

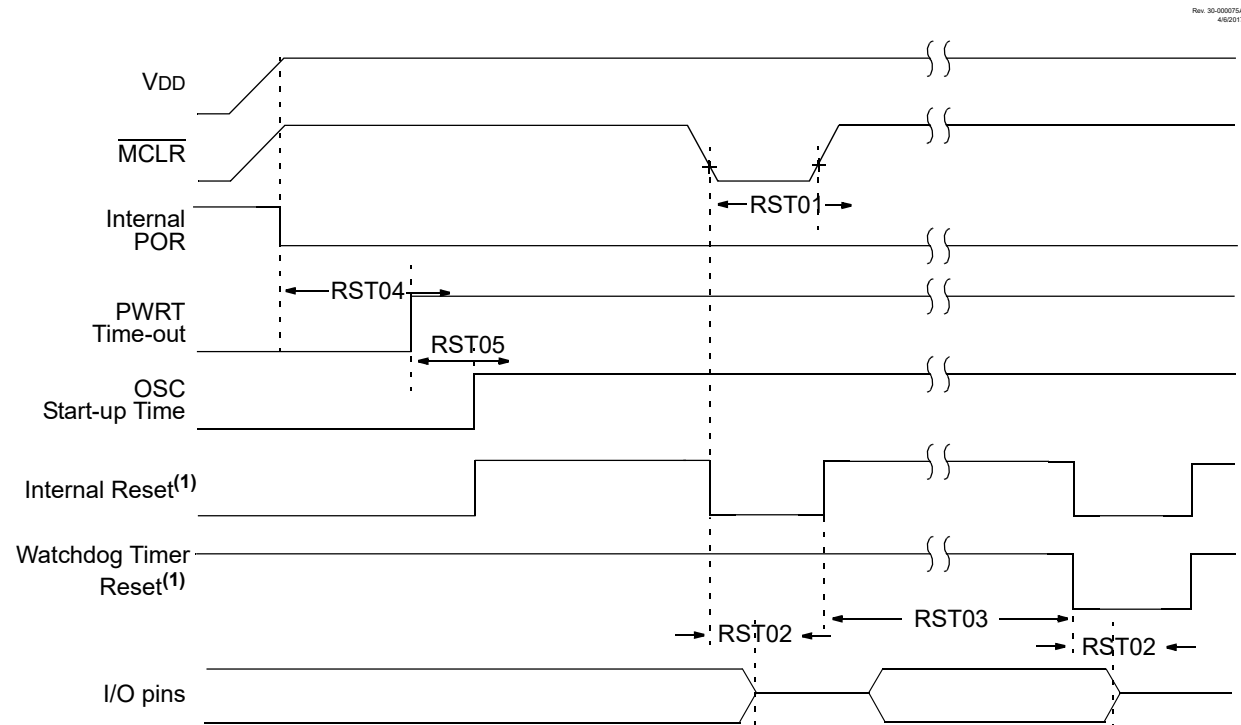
Standard Operating Conditions (unless otherwise stated)

| Param No. | Sym. | Characteristic | Min. | Typ. † | Max. | Units | Conditions |
|-----------|-------------------------|--|------|--------|------|-------|-----------------------|
| IO5* | T _{IO_HOLD} | Port input hold time (Hold time after rising edge F _{OSC} – Q2 cycle) | 50 | — | — | ns | |
| IO6* | T _{IOR_SLREN} | Port I/O rise time, slew rate enabled | — | 25 | — | ns | V _{DD} =3.0V |
| IO7* | T _{IOR_SLRDIS} | Port I/O rise time, slew rate disabled | — | 5 | — | ns | V _{DD} =3.0V |
| IO8* | T _{IOF_SLREN} | Port I/O fall time, slew rate enabled | — | 25 | — | ns | V _{DD} =3.0V |
| IO9* | T _{IOF_SLRDIS} | Port I/O fall time, slew rate disabled | — | 5 | — | ns | V _{DD} =3.0V |
| IO10* | T _{INT} | INT pin high or low time to trigger an interrupt | 25 | — | — | ns | |
| IO11* | T _{IOC} | Interrupt-on-Change minimum high or low time to trigger interrupt | 25 | — | — | ns | |

* - These parameters are characterized but not tested.

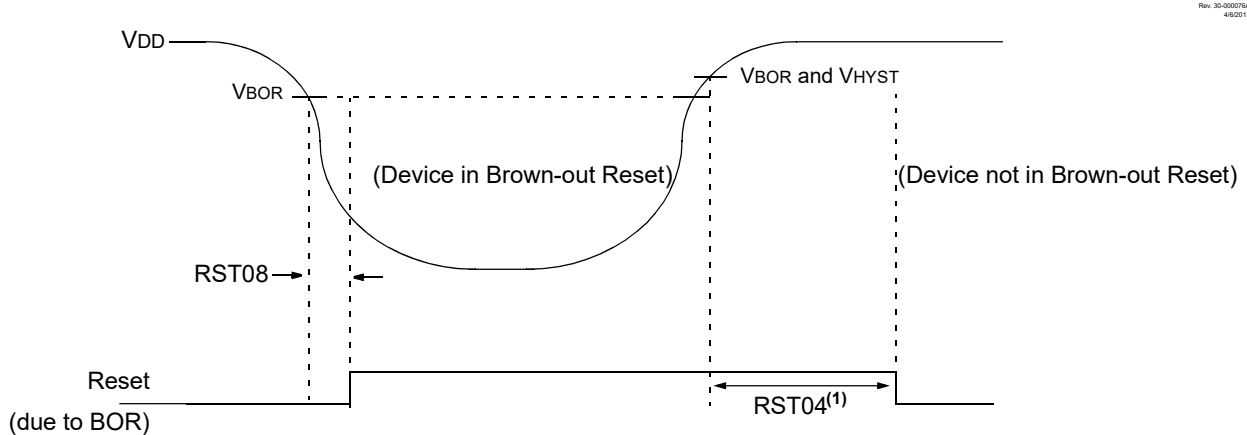
40.4.5 Reset, WDT, Oscillator Start-up Timer, Power-up Timer, Brown-Out Reset and Low-Power Brown-Out Reset Specifications

Figure 40-8. Reset, Watchdog Timer, Oscillator Start-up Timer and Power-up Timer Timing



Note:
1. Asserted low.

Figure 40-9. Brown-out Reset Timing and Characteristics



Note:

1. Only if $\overline{\text{PWRTE}}$ bit in the Configuration Word register is programmed to '1'; 2 ms delay if $\overline{\text{PWRTE}} = 0$.

Table 40-11.

| Standard Operating Conditions (unless otherwise stated) | | | | | | | |
|---|---------------------|--|------|--------|---------------------|------------------|-------------------------|
| Param No. | Sym. | Characteristic | Min. | Typ. † | Max. | Units | Conditions |
| RST01* | T_{MCLR} | MCLR Pulse Width Low to ensure Reset | 2 | — | — | μs | |
| RST02* | T_{IOZ} | I/O high-impedance from Reset detection | — | — | 2 | μs | |
| RST03 | T_{WDT} | Watchdog Timer Time-out Period | — | 16 | — | ms | 1:512 Prescaler |
| RST04* | T_{PWRT} | Power-up Timer Period | — | 65 | — | ms | |
| RST05 | T_{OST} | Oscillator Start-up Timer Period ^(1, 2) | — | 1024 | — | T_{OSC} | |
| RST06 | V_{BOR} | Brown-out Reset Voltage | 2.55 | 2.7 | 2.85 | V | BORV=0 |
| | | | 2.30 | 2.45 | 2.60 ⁽³⁾ | V | BORV=1(F devices only) |
| | | | 1.80 | 1.90 | 2.05 | V | BORV=1(LF Devices only) |
| RST07 | V_{BORHYS} | Brown-out Reset Hysteresis | — | 40 | — | mV | |
| RST08 | T_{BORDC} | Brown-out Reset Response Time | — | 3 | — | μs | |
| RST09 | V_{LPBOR} | Low-Power Brown-out Reset Voltage | 1.8 | 1.9 | 2.2 | V | LF Devices only |

* - These parameters are characterized but not tested.

† Data in "Typ" column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note:

1. By design, the Oscillator Start-up Timer (OST) counts the first 1024 cycles, independent of frequency.
2. To ensure these voltage tolerances, V_{DD} and V_{SS} must be capacitively decoupled as close to the device as possible. 0.1 μF and 0.01 μF values in parallel are recommended.
3. This value corresponds to V_{BORMAX}

40.4.6 Temperature Indicator Requirements

Table 40-12.

| Standard Operating Conditions (unless otherwise stated) | | | | | | | | |
|---|--------------|------------------------------------|------------|------|--------|------|---------|------------|
| Param. No. | Sym. | Characteristic | | Min. | | Max. | Units | Conditions |
| TS01 | T_{ACQMIN} | Minimum ADC Acquisition Time Delay | | — | 25 | — | μs | |
| TS02 | Mv | Voltage Sensitivity | High Range | — | -3.684 | — | mV/°C | TSRNG = 1 |
| | | | Low Range | — | -3.456 | — | mV/°C | TSRNG = 0 |

* - These parameters are characterized but not tested.
† Data in "Typ" column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

40.4.7 Analog-to-Digital Converter (ADC) Accuracy Specifications^(1,2)

Table 40-13.

| Standard Operating Conditions (unless otherwise stated) | | | | | | | | |
|---|-------------|---|--|-------------|-----------|-------------|------------|--------------------------------------|
| $V_{DD} = 3.0V, T_A = 25^\circ C, T_{AD} = 1\mu s$ | | | | | | | | |
| Param No. | Sym. | Characteristic | | Min. | Typ. † | Max. | Units | Conditions |
| AD01 | N_R | Resolution | | — | — | 12 | bit | |
| AD02 | E_{IL} | Integral Error | | — | ± 1.0 | ± 2.0 | LSb | $ADC_{REF+} = 3.0V, ADC_{REF-} = 0V$ |
| AD03 | E_{DL} | Differential Error | | — | ± 1.0 | ± 1.0 | LSb | $ADC_{REF+} = 3.0V, ADC_{REF-} = 0V$ |
| AD04 | E_{OFF} | Offset Error | | — | 0.5 | ± 6.0 | LSb | $ADC_{REF+} = 3.0V, ADC_{REF-} = 0V$ |
| AD05 | E_{GN} | Gain Error | | — | ± 0.2 | ± 6.0 | LSb | $ADC_{REF+} = 3.0V, ADC_{REF-} = 0V$ |
| AD06 | V_{ADREF} | ADC Reference Voltage ($AD_{REF+} - AD_{REF-}$) | | 1.8 | — | V_{DD} | V | |
| AD07 | V_{AIN} | Full-Scale Range | | AD_{REF-} | — | AD_{REF+} | V | |
| AD08 | Z_{AIN} | Recommended Impedance of Analog Voltage Source | | — | 10 | — | k Ω | |
| AD09 | R_{VREF} | ADC Voltage Reference Ladder Impedance | | — | 50 | — | k Ω | |

* - These parameters are characterized but not tested.
† Data in "Typ" column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note:

- Total Absolute Error is the sum of the offset, gain and integral non-linearity (INL) errors.
- The ADC conversion result never decreases with an increase in the input and has no missing codes.

40.4.8 Analog-to-Digital Converter (ADC) Conversion Timing Specifications

Table 40-14.

| Standard Operating Conditions (unless otherwise stated) | | | | | | | |
|---|------------------|---|------|-------------------------------------|------|-------|---|
| Param No. | Sym. | Characteristic | Min. | Typ. † | Max. | Units | Conditions |
| AD20 | T _{AD} | ADC Clock Period | 0.5 | — | 9 | μs | Using F _{OSC} as the ADC clock source ADCS = 1 |
| AD21 | | | — | 2 | — | μs | Using F _{RC} as the ADC clock source ADCS = 0 |
| AD22 | T _{CNV} | Conversion Time ⁽¹⁾ | — | 14T _{AD} +2T _{CY} | — | — | Using F _{OSC} as the ADC clock source ADCS = 1 |
| | | | — | 16T _{AD} +2T _{CY} | — | — | Using F _{RC} as the ADC clock source ADCS = 0 |
| AD23 | T _{ACQ} | Acquisition Time | — | 2 | — | μs | |
| AD24 | T _{HCD} | Sample and Hold Capacitor Disconnect Time | — | 2T _{AD} +1T _{CY} | — | — | Using F _{OSC} as the ADC clock source ADCS = 1 |
| | | | — | 3T _{AD} +2T _{CY} | — | — | Using F _{RC} as the ADC clock source ADCS = 0 |

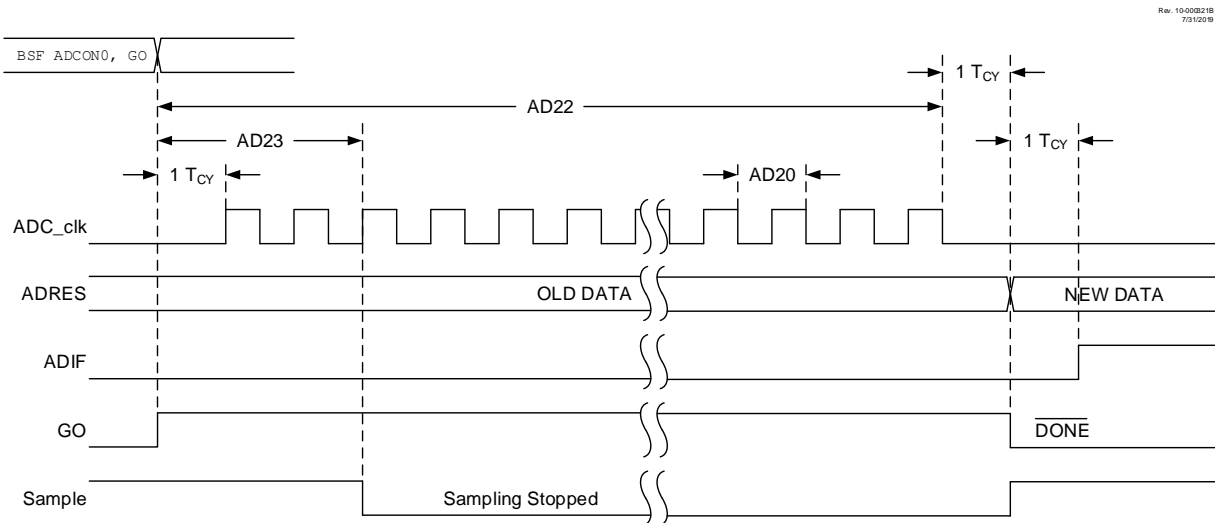
* - These parameters are characterized but not tested.

† Data in "Typ" column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note:

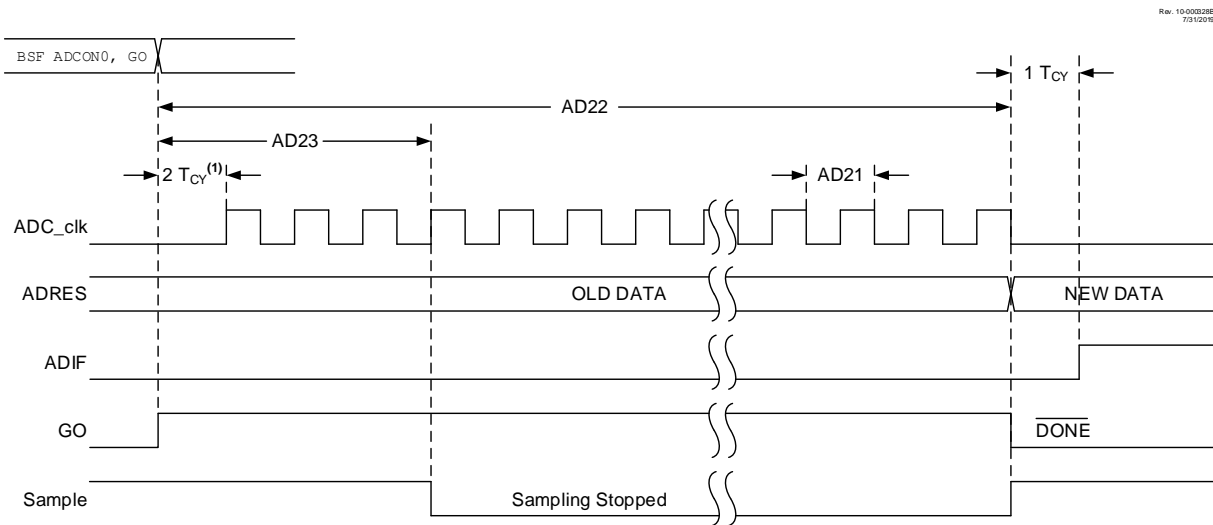
- Does not apply for the F_{RC} oscillator.

Figure 40-10. ADC Conversion Timing (ADC Clock F_{OSC}-Based)



Rev. 10-00021B
7/31/2019

Figure 40-11. ADC Conversion Timing (ADC Clock from F_{RC})



Note:

1. If the ADC clock source is selected as F_{RC}, a time of T_{CY} is added before the ADC clock starts. This allows the SLEEP instruction to be executed.

40.4.9 Comparator Specifications

Table 40-15.

| Standard Operating Conditions (unless otherwise stated) | | | | | | | |
|---|-------------------------------------|-----------------------------------|------|--------|-----------------|-------|---------------------------------------|
| V _{DD} = 3.0V, T _A = 25°C | | | | | | | |
| Param No. | Sym. | Characteristic | Min. | Typ. † | Max. | Units | Conditions |
| CM01 | V _I OFF | Input Offset Voltage | — | — | ±60 | mV | V _{ICM} = V _{DD} /2 |
| CM02 | V _{ICM} | Input Common Mode Range | GND | — | V _{DD} | V | |
| CM03 | CMRR | Common Mode Input Rejection Ratio | — | 50 | — | dB | |
| CM04 | V _{HYST} | Comparator Hysteresis | 15 | 25 | 35 | mV | |
| CM05 | T _{RESP} ⁽¹⁾ | Response Time, Rising Edge | — | 300 | 600 | ns | |
| | | Response Time, Falling Edge | — | 220 | 500 | ns | |
| CM06* | T _M CV2VO ⁽²⁾ | Mode Change to Valid Output | — | — | 10 | ns | |

* - These parameters are characterized but not tested.

† Data in "Typ" column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note:

1. Response time measured with one comparator input at V_{DD}/2, while the other input transitions from V_{SS} to V_{DD}.
2. A mode change includes changing any of the control register values, including module enable.

40.4.10 5-Bit DAC Specifications

Table 40-16.

| Standard Operating Conditions (unless otherwise stated) | | | | | | | |
|---|------------|------------------------------|------|----------------------------------|-----------|----------|------------|
| $V_{DD} = 3.0V, T_A = 25^\circ C$ | | | | | | | |
| Param No. | Sym. | Characteristic | Min. | Typ. † | Max. | Units | Conditions |
| DSB01 | V_{LSB} | Step Size | — | $(V_{DACREF+} - V_{DACREF-})/32$ | — | V | |
| DSB02 | V_{ACC} | Absolute Accuracy | — | — | ± 0.5 | LSb | |
| DSB03* | R_{UNIT} | Unit Resistor Value | — | 5000 | — | Ω | |
| DSB04* | T_{ST} | Settling Time ⁽¹⁾ | — | — | 10 | μs | |

* - These parameters are characterized but not tested.

† Data in "Typ" column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note:

- Settling time measured while DACR[4:0] transitions from '00000' to '01111'.

40.4.11 Fixed Voltage Reference (FVR) Specifications

Table 40-17.

| Standard Operating Conditions (unless otherwise stated) | | | | | | | |
|---|-------------|-------------------|------|--------|------|---------|--|
| Param No. | Sym. | Characteristic | Min. | Typ. † | Max. | Units | Conditions |
| FVR01 | V_{FVR1} | 1x Gain (1.024V) | -4 | — | +4 | % | $V_{DD} \geq 2.5V, -40^\circ C$ to $85^\circ C$ |
| FVR02 | V_{FVR2} | 2x Gain (2.048V) | -4 | — | +4 | % | $V_{DD} \geq 2.5V, -40^\circ C$ to $85^\circ C$ |
| FVR03 | V_{FVR4} | 4x Gain (4.096V) | -5 | — | +5 | % | $V_{DD} \geq 4.75V, -40^\circ C$ to $85^\circ C$ |
| FVR04 | T_{FVRST} | FVR Start-up Time | — | 60 | — | μs | |

40.4.12 Zero-Cross Detect (ZCD) Specifications

Table 40-18.

| Standard Operating Conditions (unless otherwise stated) | | | | | | | |
|---|----------------|--------------------------------|------|--------|------|---------|------------|
| $V_{DD} = 3.0V, T_A = 25^\circ C$ | | | | | | | |
| Param No. | Sym. | Characteristic | Min. | Typ. † | Max. | Units | Conditions |
| ZC01 | V_{PINZC} | Voltage on Zero Cross Pin | — | 0.75 | — | V | |
| ZC02 | I_{ZCD_MAX} | Maximum source or sink current | — | — | 600 | μA | |
| ZC03 | T_{RESPH} | Response Time, Rising Edge | — | 1 | — | μs | |
| | T_{RESPL} | Response Time, Falling Edge | — | 1 | — | μs | |

† Data in "Typ" column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

40.4.13 Timer0 and Timer1 External Clock Requirements

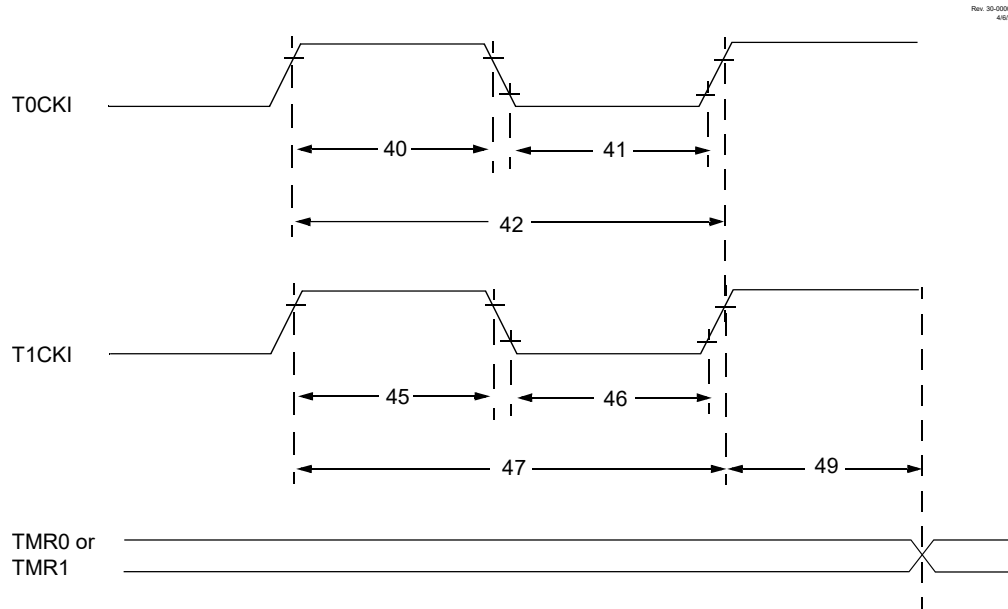
Table 40-19.

| Standard Operating Conditions (unless otherwise stated) | | | | | | | | |
|---|-----------------------|---|-----------------------------|--|--------|-------------|-------|---------------------|
| Operating Temperature: $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ | | | | | | | | |
| Param No. | Sym. | Characteristic | | Min. | Typ. † | Max. | Units | Conditions |
| 40* | T _{T0H} | T0CKI High Pulse Width | No Prescaler | $0.5T_{CY}+20$ | — | — | ns | |
| | | | With Prescaler | 10 | — | — | ns | |
| 41* | T _{T0L} | T0CKI Low Pulse Width | No Prescaler | $0.5T_{CY}+20$ | — | — | ns | |
| | | | With Prescaler | 10 | — | — | ns | |
| 42* | T _{T0P} | T0CKI Period | | Greater of: 20 or $(T_{CY} + 40)/N$ | — | — | ns | N = Prescale value |
| 45* | T _{T1H} | T1CKI High Time | Synchronous, No Prescaler | $0.5T_{CY}+20$ | — | — | ns | |
| | | | Synchronous, with Prescaler | 15 | — | — | ns | |
| | | | Asynchronous | 30 | — | — | ns | |
| 46* | T _{T1L} | T1CKI Low Time | Synchronous, No Prescaler | $0.5T_{CY}+20$ | — | — | ns | |
| | | | Synchronous, with Prescaler | 15 | — | — | ns | |
| | | | Asynchronous | 30 | — | — | ns | |
| 47* | T _{T1P} | T1CKI Input Period | Synchronous | Greater of: 30 or $(T_{CY} + 40)/N$ | — | — | ns | N = Prescale value |
| | | | Asynchronous | 60 | — | — | ns | |
| 49* | TCKEZ _{TMR1} | Delay from External Clock Edge to Timer Increment | | $2 T_{OSC}$ | — | $7 T_{OSC}$ | — | Timers in Sync mode |

* - These parameters are characterized but not tested.

† Data in "Typ" column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Figure 40-12. Timer0 and Timing1 External Clock Timings



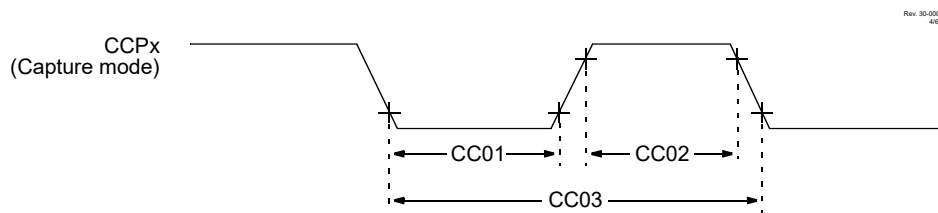
40.4.14 Capture/Compare/PWM Requirements (CCP)

Table 40-20.

| Standard Operating Conditions (unless otherwise stated) | | | | | | | | |
|---|-----------|----------------------|----------------|------------------|--------|------|-------|--------------------|
| Operating Temperature: $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ | | | | | | | | |
| Param No. | Sym. | Characteristic | | Min. | Typ. † | Max. | Units | Conditions |
| CC01* | T_{CCL} | CCPx Input Low Time | No Prescaler | $0.5T_{CY}+20$ | — | — | ns | |
| | | | With Prescaler | 20 | — | — | ns | |
| CC02* | T_{CCH} | CCPx Input High Time | No Prescaler | $0.5T_{CY}+20$ | — | — | ns | |
| | | | With Prescaler | 20 | — | — | ns | |
| CC03* | T_{CCP} | CCPx Input Period | | $(3T_{CY}+40)/N$ | — | — | ns | N = Prescale value |

* - These parameters are characterized but not tested.
 † Data in "Typ" column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Figure 40-13. Capture/Compare/PWM Timings (CCP)



Note: Refer to Figure 40-4 for load conditions.

40.4.15 Configurable Logic Cell (CLC) Characteristics

Table 40-21.

| Standard Operating Conditions (unless otherwise stated) | | | | | | | | |
|---|---------------------|---|-----------|------|--------|------------------|-------|-------------------------------|
| Operating Temperature: $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ | | | | | | | | |
| Param No. | Sym. | Characteristic | | Min. | Typ. † | Max. | Units | Conditions |
| CLC01* | T_{CLCIN} | CLC input time | | — | 7 | OS5 | ns | (Note1) |
| CLC02* | T_{CLC} | CLC module input to output propagation time | | — | 24 | — | ns | $V_{\text{DD}} = 1.8\text{V}$ |
| | | | | — | 12 | — | ns | $V_{\text{DD}} > 3.6\text{V}$ |
| CLC03* | T_{CLCOUT} | CLC output time | Rise Time | — | OS7 | — | — | (Note1) |
| | | | Fall Time | — | OS8 | — | — | (Note1) |
| CLC04* | F_{CLCMAX} | CLC maximum switching frequency | | — | 32 | F_{OSC} | MHz | |

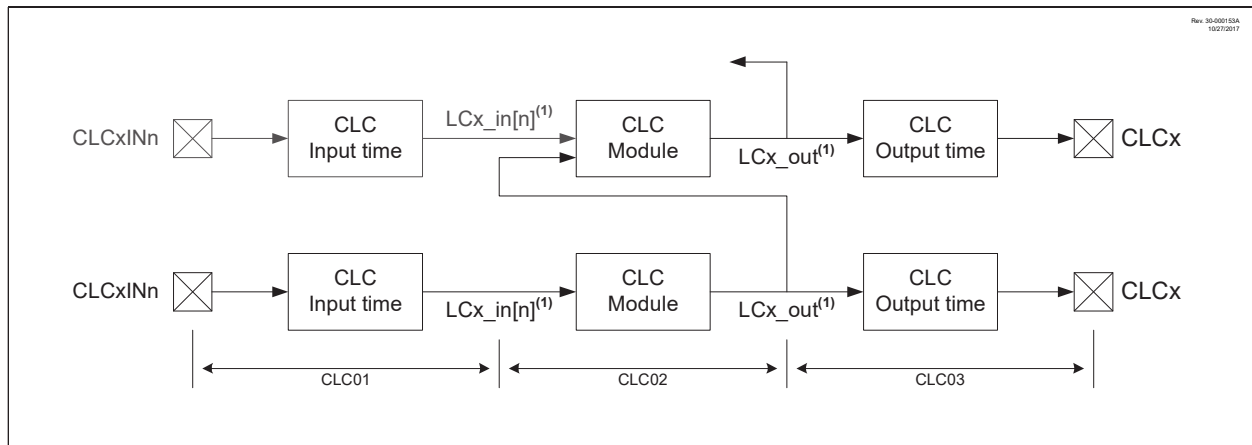
* - These parameters are characterized but not tested.

† - Data in "Typ" column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note:

- See "I/O and CLKOUT Timing Specifications" for OS5, OS7 and OS8 rise and fall times.

Figure 40-14. CLC Propagation Timing



Related Links

[40.4.4 I/O and CLKOUT Timing Specifications](#)

40.4.16 EUSART Synchronous Transmission Requirements

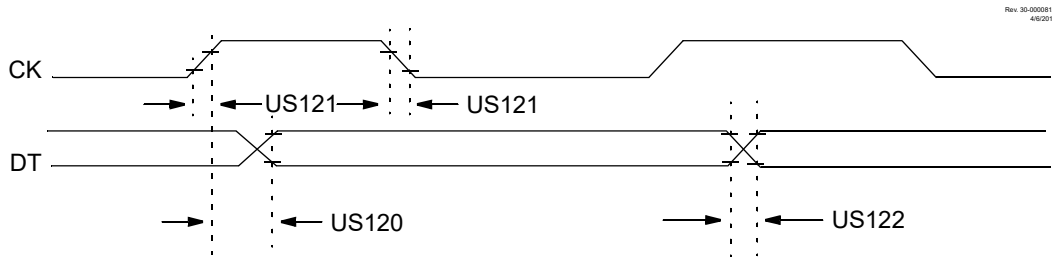
Table 40-22.

| Standard Operating Conditions (unless otherwise stated) | | | | | | | |
|---|----------------------|------------------------------|--|------|------|-------|---|
| Param No. | Sym. | Characteristic | | Min. | Max. | Units | Conditions |
| US120 | T_{CKH2DTV} | SYNC XMIT (Master and Slave) | | — | 80 | ns | $3.0\text{V} \leq V_{\text{DD}} \leq 5.5\text{V}$ |
| | | Clock high to data-out valid | | — | 100 | ns | $1.8\text{V} \leq V_{\text{DD}} \leq 5.5\text{V}$ |

.....continued

| Standard Operating Conditions (unless otherwise stated) | | | | | | |
|---|------------|--|------|------|-------|------------------------------|
| Param No. | Sym. | Characteristic | Min. | Max. | Units | Conditions |
| US121 | T_{CKRF} | Clock out rise time and fall time (Master mode) | — | 45 | ns | $3.0V \leq V_{DD} \leq 5.5V$ |
| | | | — | 50 | ns | $1.8V \leq V_{DD} \leq 5.5V$ |
| US122 | T_{DTRF} | Data-out rise time and fall time | — | 45 | ns | $3.0V \leq V_{DD} \leq 5.5V$ |
| | | | — | 50 | ns | $1.8V \leq V_{DD} \leq 5.5V$ |

Figure 40-15. EUSART Synchronous Transmission (Master/Slave) Timing



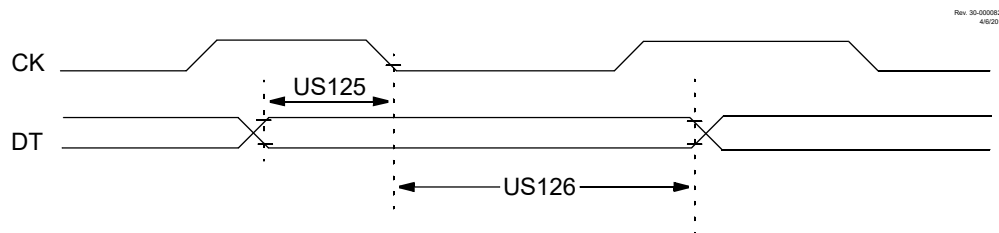
Note: Refer to Figure 40-4 for load conditions.

40.4.17 EUSART Synchronous Receive Requirements

Table 40-23.

| Standard Operating Conditions (unless otherwise stated) | | | | | | |
|---|------------------|---------------------------------------|------|------|-------|------------|
| Param No. | Sym. | Characteristic | Min. | Max. | Units | Conditions |
| US125 | $T_{DT}V2_{CKL}$ | SYNC RCV (Master and Slave) | 10 | — | ns | |
| | | Data-setup before CK ↓ (DT hold time) | | | | |
| US126 | $T_{CKL2_{DTL}}$ | Data-hold after CK ↓ (DT hold time) | 15 | — | ns | |

Figure 40-16. EUSART Synchronous Receive (Master/Slave) Timing



Note: Refer to Figure 40-4 for load conditions.

40.4.18 SPI Mode Requirements

Table 40-24.

| Standard Operating Conditions (unless otherwise stated) | | | | | | | |
|---|--|---|---------------------|--------|------|-------|------------|
| Param No. | Sym. | Characteristic | Min. | Typ. † | Max. | Units | Conditions |
| SP70* | $T_{SSL2_{SCH}}$, $T_{SSL2_{SCL}}$ | $\overline{SS} \downarrow$ to SCK ↓ or SCK ↑ input | $2.25 \cdot T_{CY}$ | — | — | ns | |

PIC16(L)F18426/46

Electrical Specifications

.....continued

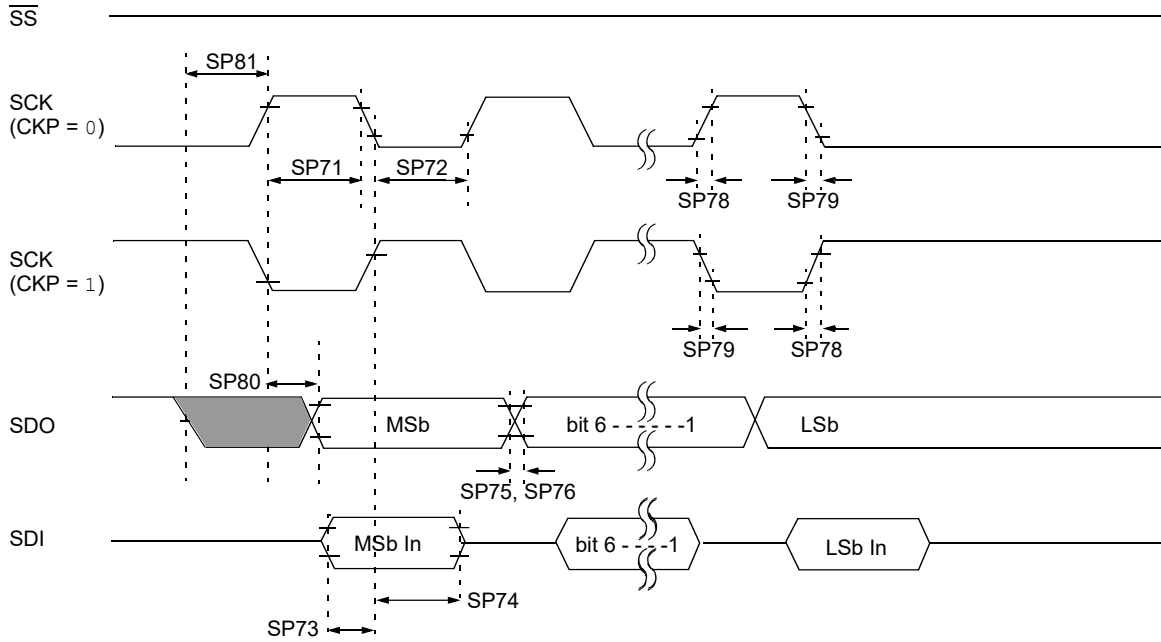
| Standard Operating Conditions (unless otherwise stated) | | | | | | | |
|---|---|--|--------------------------|--------|------|-------|-------------------------------|
| Param No. | Sym. | Characteristic | Min. | Typ. † | Max. | Units | Conditions |
| SP71* | T _{SC} H | SCK input high time (Slave mode) | T _{CY} + 20 | — | — | ns | |
| SP72* | T _{SC} L | SCK input low time (Slave mode) | T _{CY} + 20 | — | — | ns | |
| SP73* | T _{DI} V2 _{SC} H, T _{DI} V2 _{SC} L | Setup time of SDI data input to SCK edge | 100 | — | — | ns | |
| SP74* | T _{SC} H2 _{DI} L, T _{SC} L2 _{DI} L | Hold time of SDI data input to SCK edge | 100 | — | — | ns | |
| SP75* | T _{DO} R | SDO data output rise time | — | 10 | 25 | ns | 3.0V ≤ V _{DD} ≤ 5.5V |
| | | | — | 25 | 50 | ns | 1.8V ≤ V _{DD} ≤ 5.5V |
| SP76* | T _{DO} F | SDO data output fall time | — | 10 | 25 | ns | |
| SP77* | T _{SS} H2 _{DO} Z | \overline{SS} ↑ to SDO output high-impedance | 10 | — | 50 | ns | |
| SP78* | T _{SC} R | SCK output rise time (Master mode) | — | 10 | 25 | ns | 3.0V ≤ V _{DD} ≤ 5.5V |
| | | | — | 25 | 50 | ns | 1.8V ≤ V _{DD} ≤ 5.5V |
| SP79* | T _{SC} F | SCK output fall time (Master mode) | — | 10 | 25 | ns | |
| SP80* | T _{SC} H2 _{DO} V, T _{SC} L2 _{DO} V | SDO data output valid after SCK edge | — | — | 50 | ns | 3.0V ≤ V _{DD} ≤ 5.5V |
| | | | — | — | 145 | ns | 1.8V ≤ V _{DD} ≤ 5.5V |
| SP81* | T _{DO} V2 _{SC} H, T _{DO} V2 _{SC} L | SDO data output setup to SCK edge | 1 T _{CY} | — | — | ns | |
| SP82* | T _{SS} L2 _{DO} V | SDO data output valid after \overline{SS} ↓ edge | — | — | 50 | ns | |
| SP83* | T _{SC} H2 _{SS} H, T _{SC} L2 _{SS} H | \overline{SS} ↑ □ after SCK edge | 1.5 T _{CY} + 40 | — | — | ns | |

* - These parameters are characterized but not tested.

† Data in "Typ" column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Figure 40-17. SPI Master Mode Timing (CKE = 0, SMP = 0)

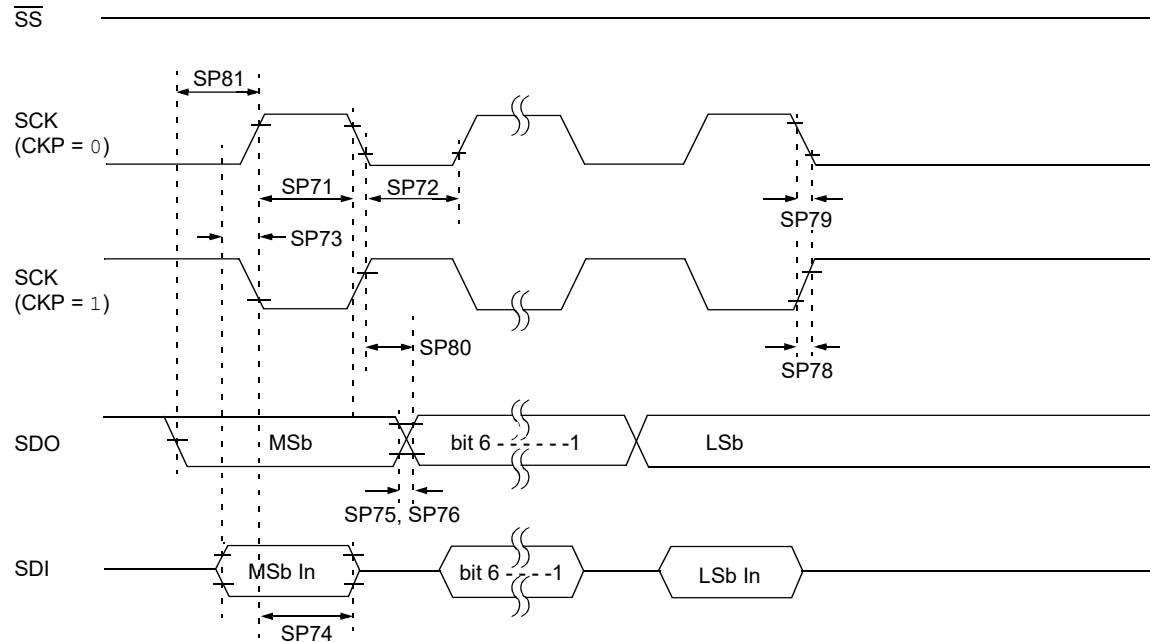
Rev. 30-000083A
4/6/2017



Note: Refer to [Figure 40-4](#) for load conditions.

Figure 40-18. SPI Master Mode Timing (CKE = 1, SMP = 1)

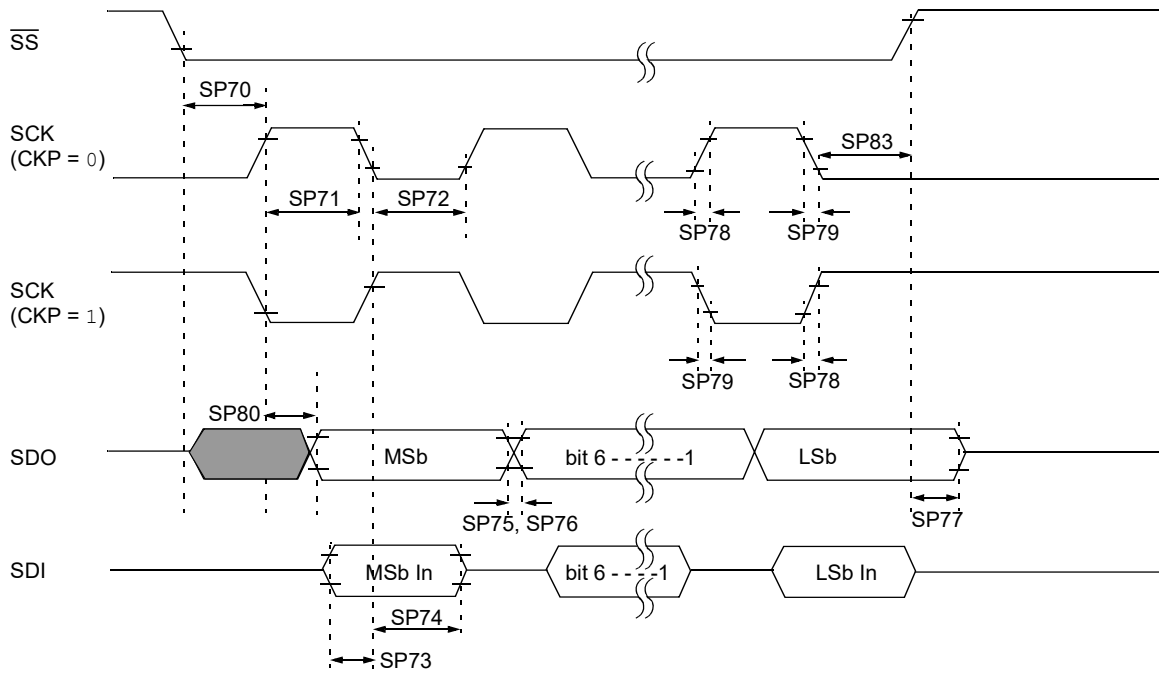
Rev. 30-000084A
4/6/2017



Note: Refer to [Figure 40-4](#) for load conditions.

Figure 40-19. SPI Slave Mode Timing (CKE = 0)

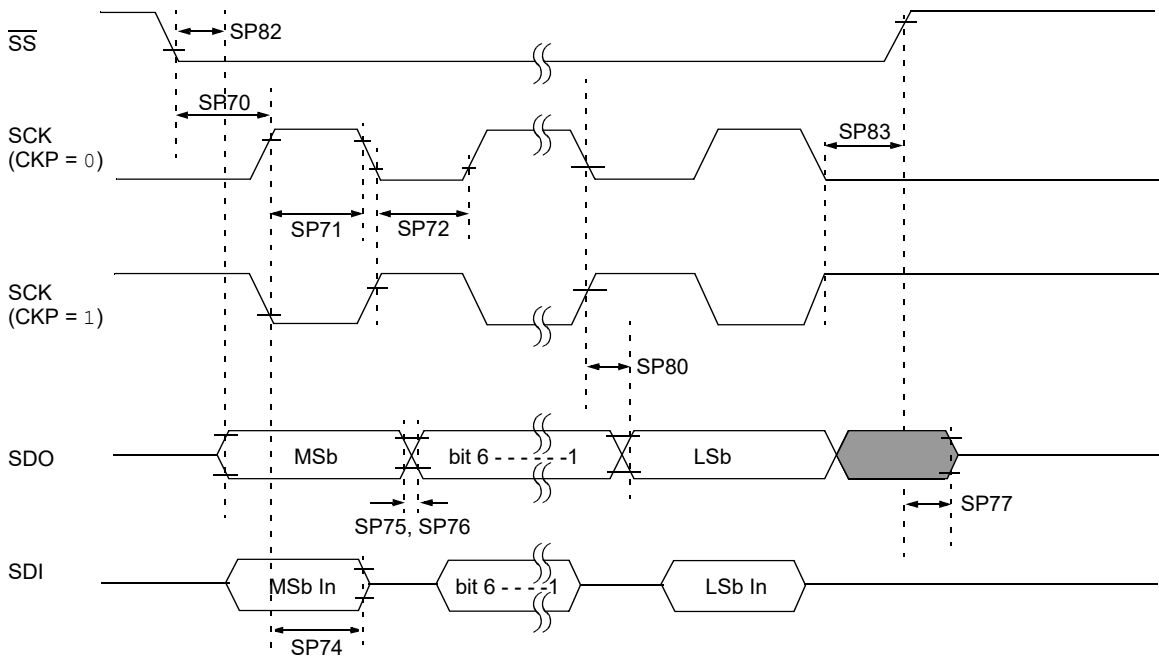
Rev. 35-000985A
4/5/2017



Note: Refer to Figure 40-4 for load conditions.

Figure 40-20. SPI Slave Mode Timing (CKE = 1)

Rev. 35-000985A
4/5/2017



Note: Refer to Figure 40-4 for load conditions.

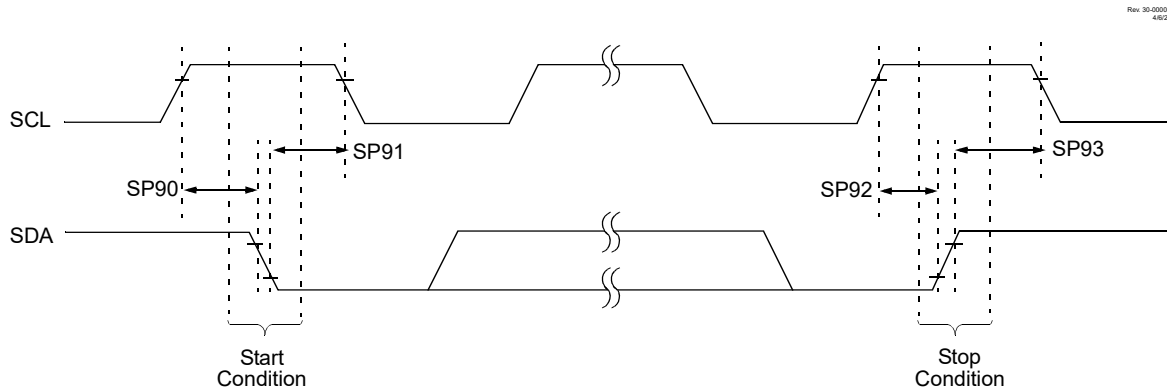
40.4.19 I²C Bus Start/Stop Bits Requirements

Table 40-25.

| Standard Operating Conditions (unless otherwise stated) | | | | | | | | |
|---|---------------------|-----------------|--------------|------|--------|------|-------|--|
| Param. No. | Sym. | Characteristic | | Min. | Typ. † | Max. | Units | Conditions |
| SP90* | T _{SU:STA} | Start condition | 100 kHz mode | 4700 | — | — | ns | Only relevant for Repeated Start Setup time 400 kHz mode 600 condition |
| | | Setup time | 400 kHz mode | 600 | — | — | | |
| SP91* | T _{HD:STA} | Start condition | 100 kHz mode | 4000 | — | — | ns | After this period, the first clock Hold time 400 kHz mode 600 — — pulse is generated |
| | | Hold time | 400 kHz mode | 600 | — | — | | |
| SP92* | T _{SU:STO} | Stop condition | 100 kHz mode | 4700 | — | — | ns | |
| | | Setup time | 400 kHz mode | 600 | — | — | | |
| SP93* | T _{HD:STO} | Stop condition | 100 kHz mode | 4000 | — | — | ns | |
| | | Hold time | 400 kHz mode | 600 | — | — | | |

* - These parameters are characterized but not tested.

Figure 40-21. I²C Bus Start/Stop Bits Timing



Note: Refer to Figure 40-4 for load conditions.

40.4.20 I²C Bus Data Requirements

Table 40-26.

| Standard Operating Conditions (unless otherwise stated) | | | | | | | |
|---|-------------------|-----------------|--------------|--------------------|------|-------|---|
| Param. No. | Sym. | Characteristic | | Min. | Max. | Units | Conditions |
| SP100* | T _{HIGH} | 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 | 1.5T _{CY} | — | | |

PIC16(L)F18426/46

Electrical Specifications

.....continued

Standard Operating Conditions (unless otherwise stated)

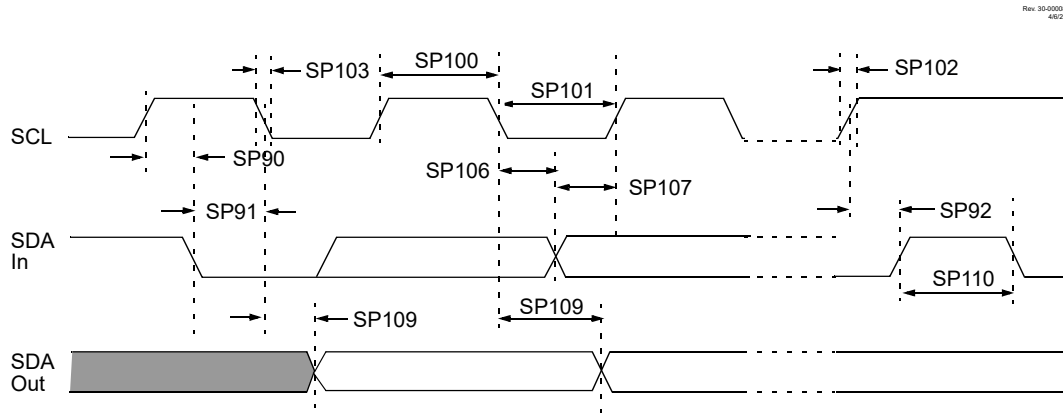
| Param. No. | Sym. | Characteristic | Min. | Max. | Units | Conditions | |
|------------|---------------------|-------------------------|--------------|------------------------|-------|------------|---|
| SP101* | T _{LOW} | Clock low time | 100 kHz mode | 4.7 | — | μs | Device must operate at a minimum of 1.5 MHz |
| | | | 400 kHz mode | 1.3 | — | μs | Device must operate at a minimum of 10 MHz |
| | | | SSP module | 1.5T _{CY} | — | | |
| SP102* | T _R | SDA and SCL rise time | 100 kHz mode | — | 1000 | ns | |
| | | | 400 kHz mode | 20 + 0.1C _B | 300 | ns | C _B is specified to be from 10-400 pF |
| SP103* | T _F | SDA and SCL fall time | 100 kHz mode | — | 250 | ns | |
| | | | 400 kHz mode | 20 + 0.1C _B | 250 | ns | C _B is specified to be from 10-400 pF |
| SP106* | T _{HD:DAT} | Data input hold time | 100 kHz mode | 0 | — | ns | |
| | | | 400 kHz mode | 0 | 0.9 | μs | |
| SP107* | T _{SU:DAT} | Data input setup time | 100 kHz mode | 250 | — | ns | (Note 2) |
| | | | 400 kHz mode | 100 | — | ns | |
| SP109* | T _{AA} | Output valid from clock | 100 kHz mode | — | 3500 | ns | (Note 1) |
| | | | 400 kHz mode | — | — | ns | |
| SP110* | T _{BUF} | Bus free time | 100 kHz mode | 4.7 | — | μs | Time the bus must be free before a new transmission can start |
| | | | 400 kHz mode | 1.3 | — | μs | |
| SP111 | C _B | Bus capacitive loading | — | 400 | pF | | |

* - These parameters are characterized but not tested.

Note:

- 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.
- A Fast mode (400 kHz) I²C bus device can be used in a Standard mode (100 kHz) I²C bus system, but the requirement T_{SU: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. + T_{SU:DAT} = 1000 + 250 = 1250 ns (according to the Standard mode I²C bus specification), before the SCL line is released.

Figure 40-22. I²C Bus Data Timing



Note: Refer to [Figure 40-4](#) for load conditions.

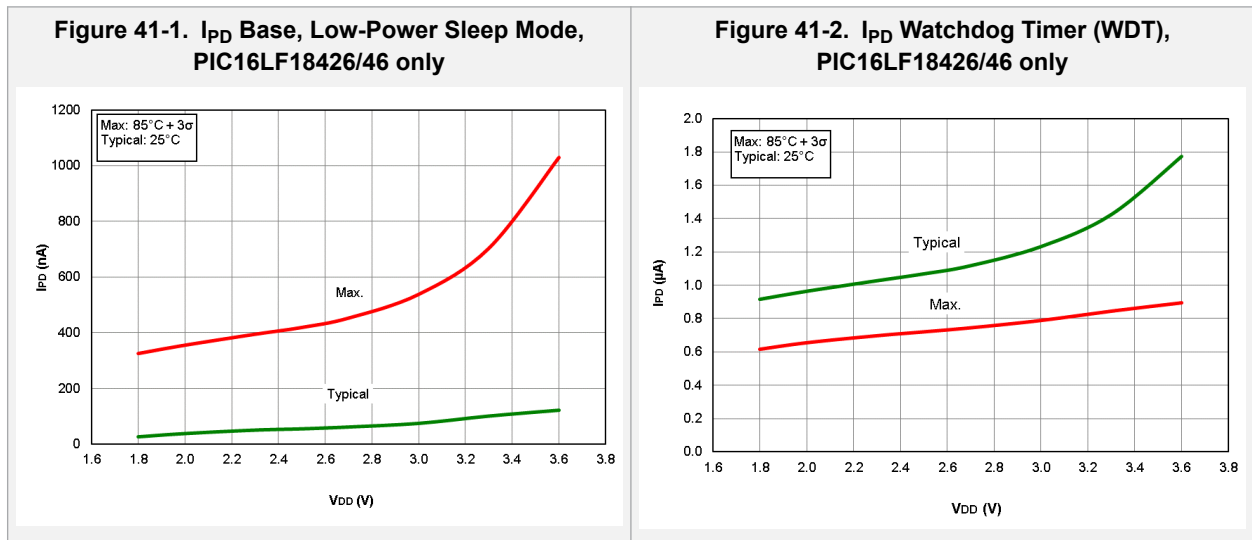
41. DC and AC Characteristics Graphs and Tables

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. Unless otherwise noted, all graphs apply to both the L and LF devices.

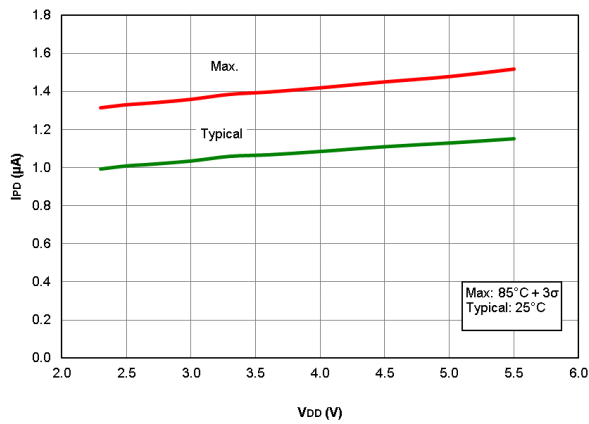
Note: The graphs and tables provided following this note are a statistical summary based on a limited number of samples and are provided for informational purposes only. The performance characteristics listed herein are not tested or guaranteed. In some graphs or tables, the data presented may be outside the specified operating range (e.g., outside specified power supply range) and therefore, outside the warranted range.

Note: “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.

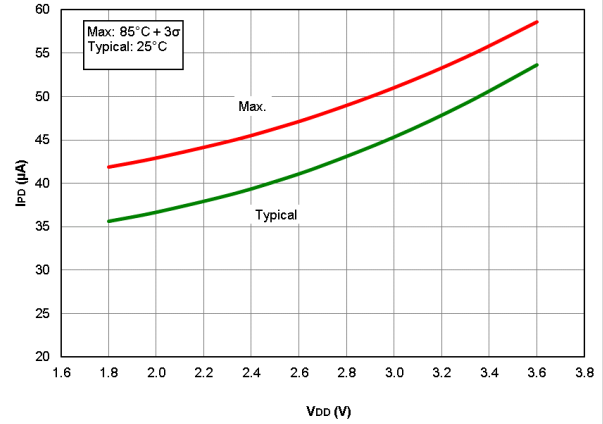
41.1 Graphs



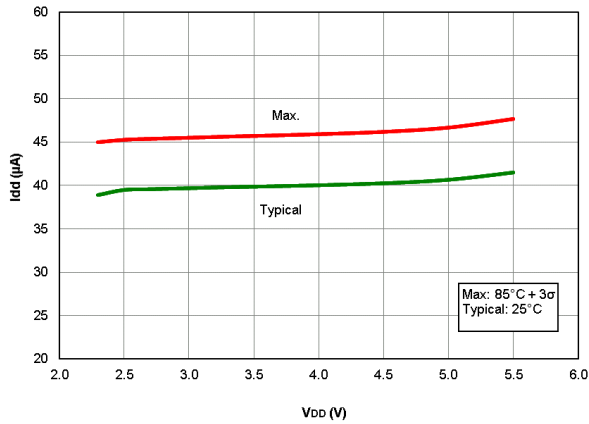
**Figure 41-3. I_{PD} Watchdog Timer (WDT),
PIC16F18426/46 only**



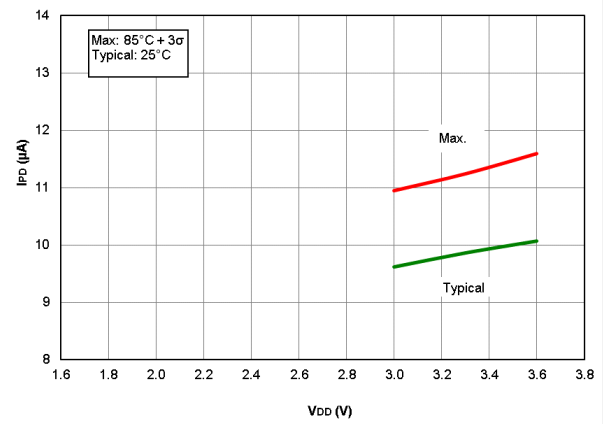
**Figure 41-4. I_{PD} Fixed Voltage Reference (FVR),
PIC16LF18426/46 only**



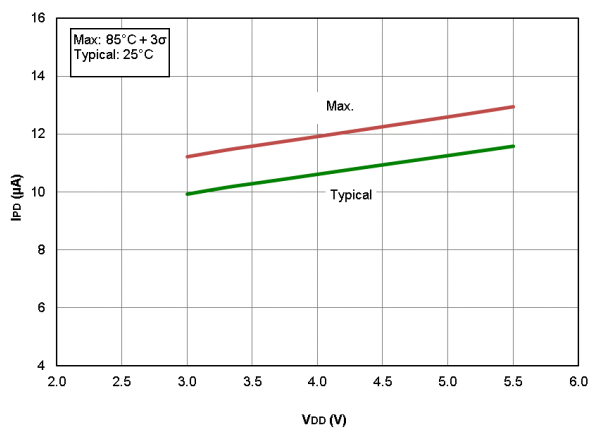
**Figure 41-5. I_{DD} Fixed Voltage Reference (FVR),
PIC16F18426/46 only**



**Figure 41-6. I_{PD} Brown-Out Reset (BOR), BORV = 1,
PIC16LF18426/46 only**



**Figure 41-7. I_{PD} Brown-Out Reset (BOR), BORV = 1,
PIC16F18426/46 only**



**Figure 41-8. I_{PD} Low-Power Brown-Out Reset,
LPBOR = 0, PIC16LF18426/46 only**

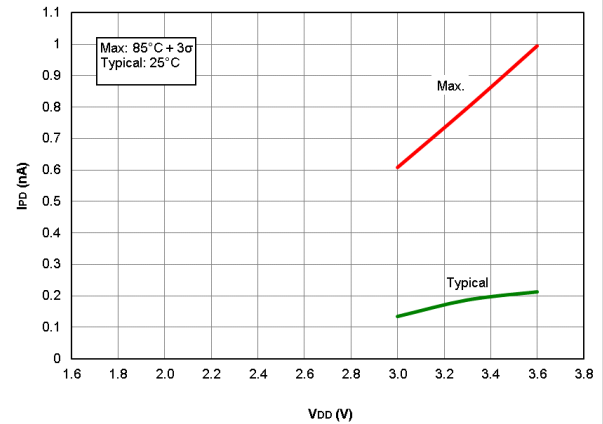


Figure 41-9. I_{PD} Comparator, PIC16LF18426/46 only

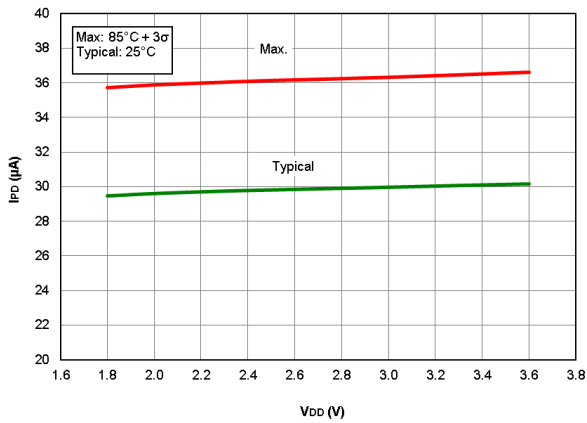


Figure 41-10. I_{PD} Comparator, PIC16F18426/46 only

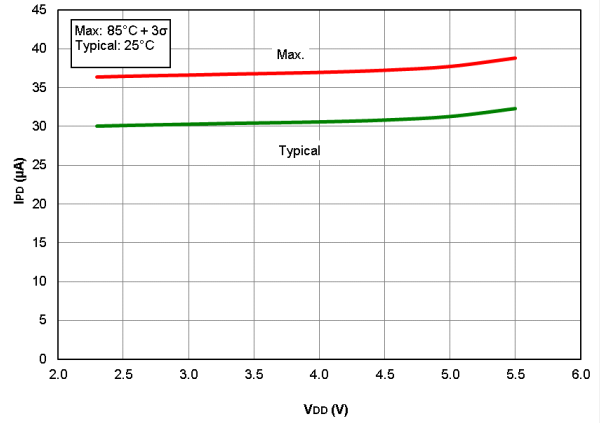


Figure 41-11. I_{PD} Base, 01, PIC16F18426/46 only

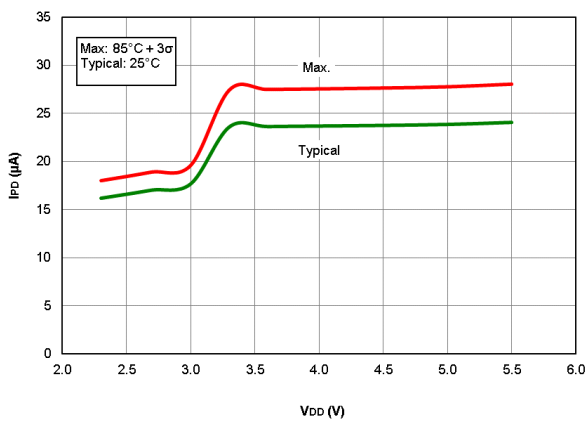


Figure 41-12. I_{PD} Base, 11, PIC16F18426/46 only

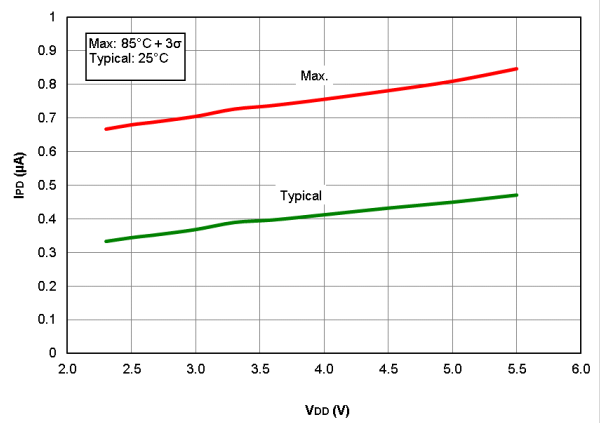


Figure 41-13. I_{DD} XT Oscillator 4 MHz, PIC16LF18426/46 only

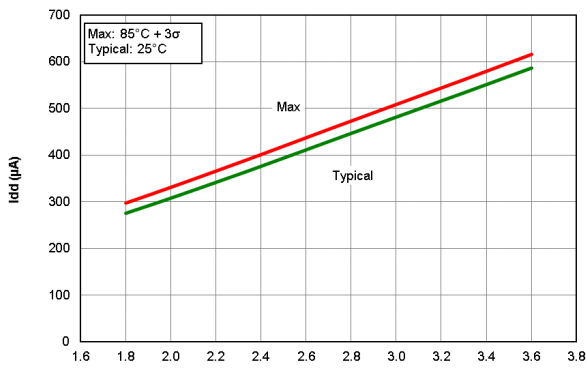


Figure 41-14. I_{DD} XT Oscillator 4 MHz, PIC16F18426/46 only

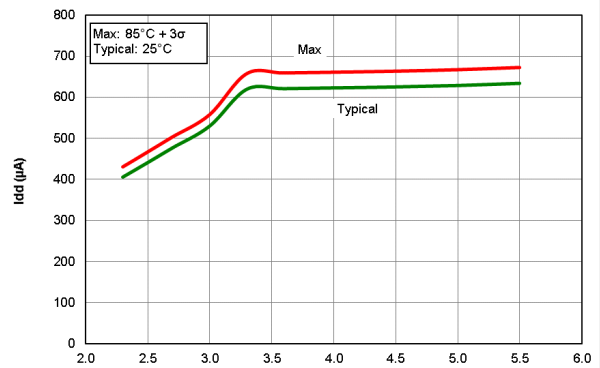


Figure 41-15. I_{DD} HS Oscillator 32 MHz, PIC16LF18426/46 only

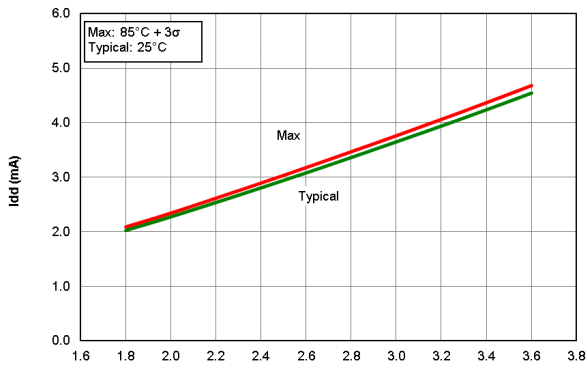


Figure 41-16. I_{DD} HS Oscillator 32 MHz, PIC16F18426/46 only

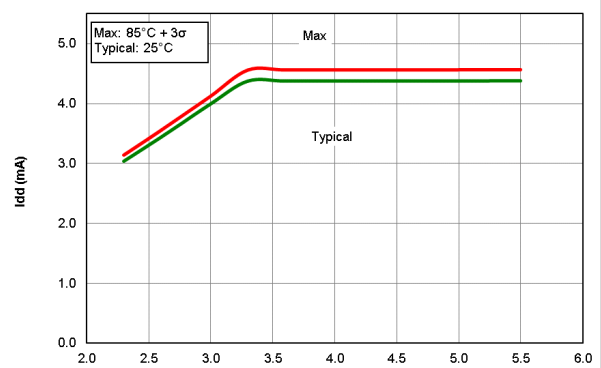


Figure 41-17. I_{DD} HFINTOSC Mode, F_{OSC} = 32 MHz, PIC16LF18426/46 only

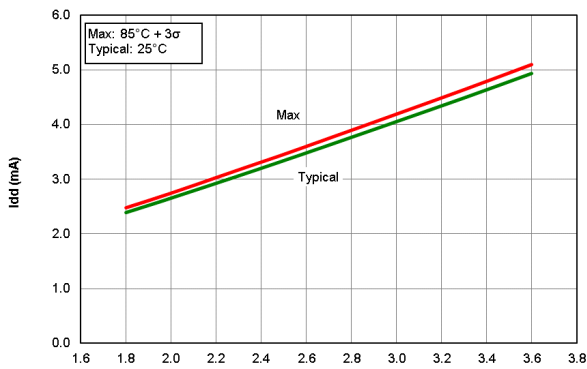


Figure 41-18. I_{DD} HFINTOSC Mode, F_{OSC} = 32 MHz, PIC16F18426/46 only

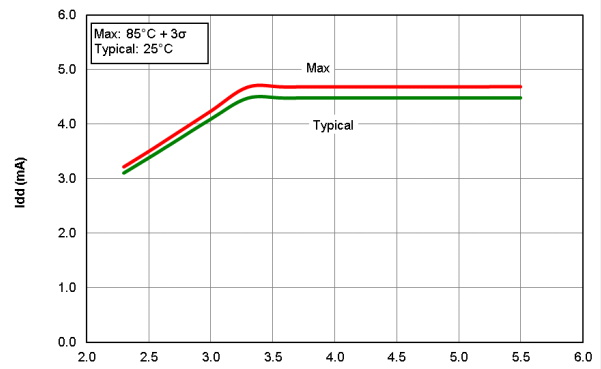


Figure 41-19. I_{DD} HFINTOSC Mode, F_{OSC} = 16 MHz, PIC16LF18426/46 only

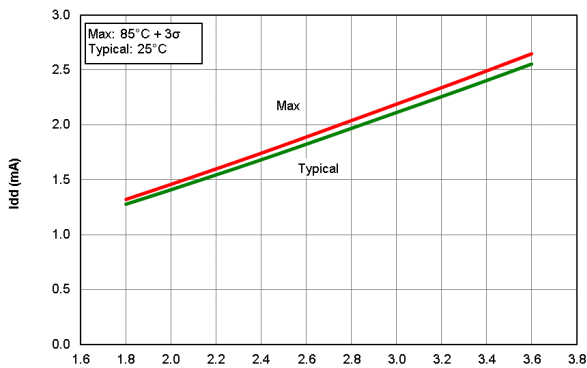


Figure 41-20. I_{DD} HFINTOSC Mode, F_{OSC} = 16 MHz, PIC16F18426/46 only

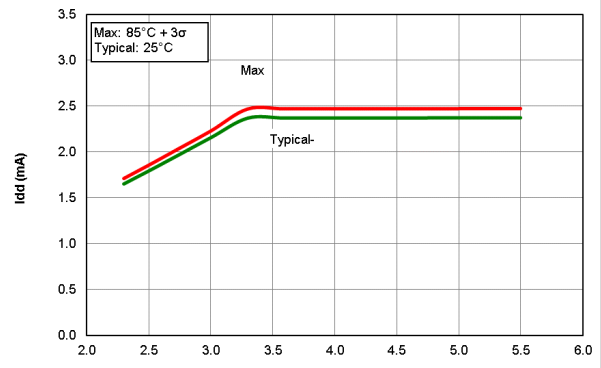


Figure 41-21. I_{DD} HFINTOSC Idle Mode, $F_{OSC}=16$ MHz, PIC16LF18426/46 only

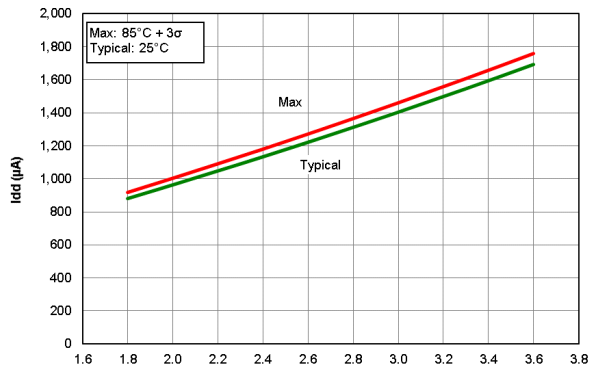


Figure 41-22. I_{DD} HFINTOSC Idle Mode, $F_{OSC}=16$ MHz, PIC16F18426/46 only

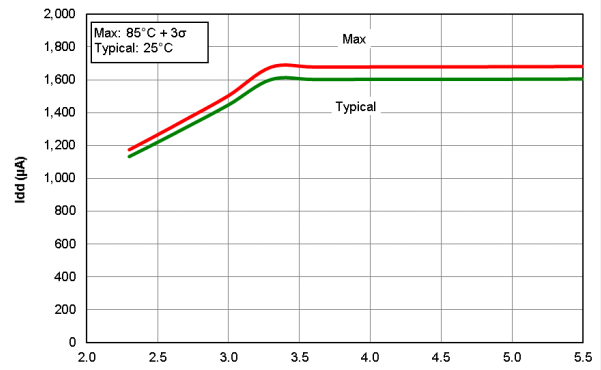


Figure 41-23. I_{DD} HFINTOSC Doze Mode, $F_{OSC}=16$ MHz, PIC16LF18426/46 only

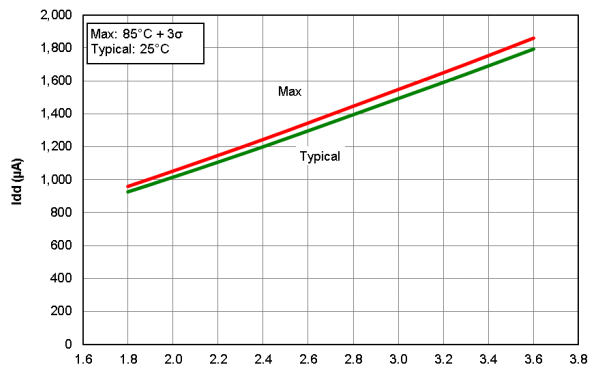


Figure 41-24. I_{DD} HFINTOSC Doze Mode, $F_{OSC}=16$ MHz, PIC16F18426/46 only

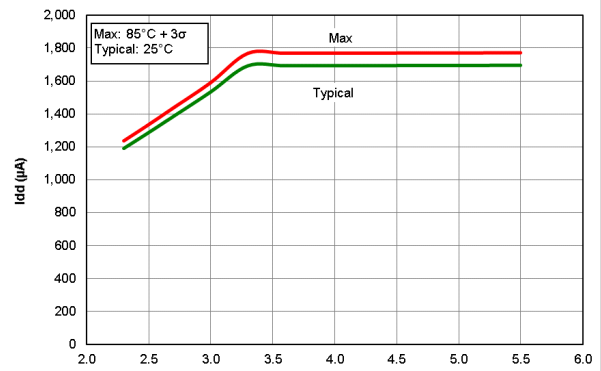


Figure 41-25. LFINTOSC Frequency, PIC16LF18426/46 only

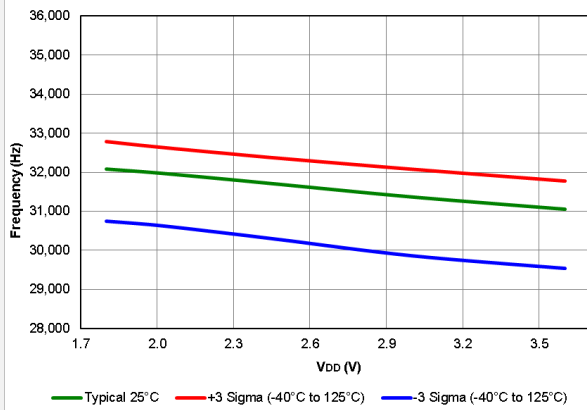


Figure 41-26. LFINTOSC Frequency, PIC16F18426/46 only

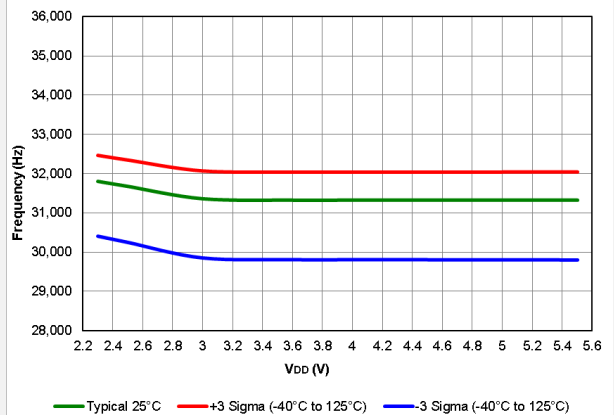


Figure 41-27. HFINTOSC Typical Frequency Error, PIC16LF18426/46 only

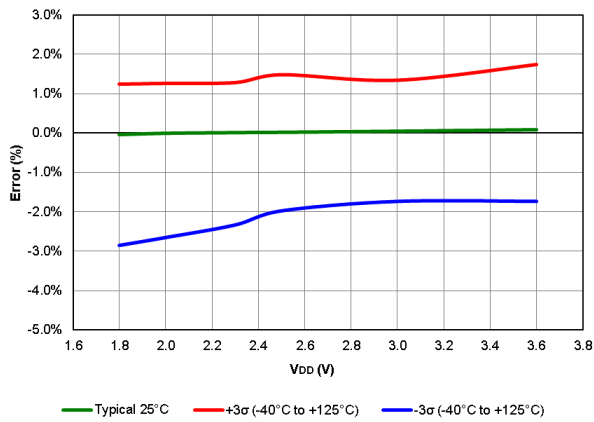


Figure 41-28. HFINTOSC Typical Frequency Error, PIC16F18426/46 only

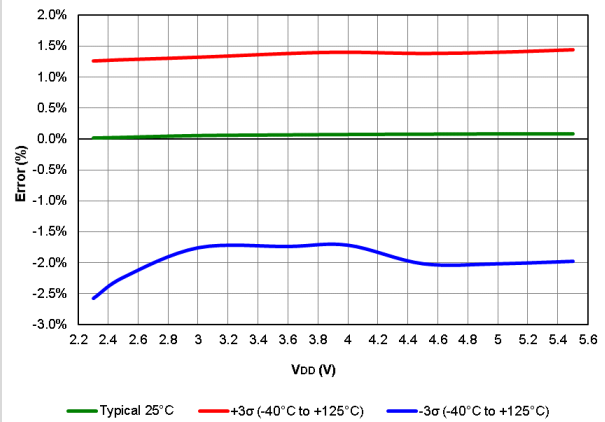


Figure 41-29. HFINTOSC Frequency Error $V_{DD} = 3V$, All devices

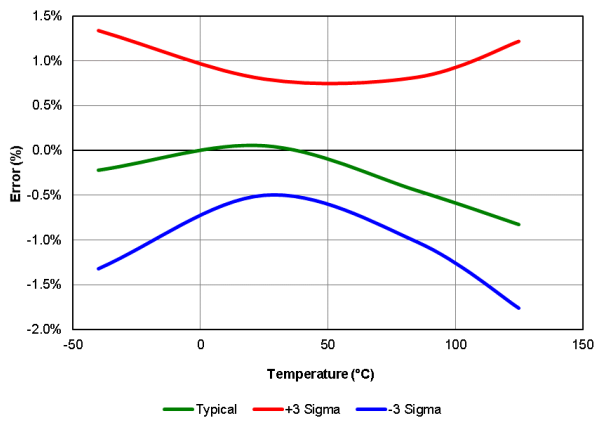


Figure 41-30. Weak Pull-Up Current, PIC16F18426/46 only

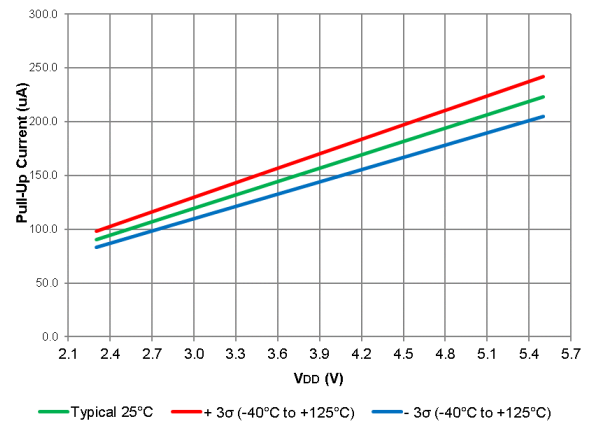


Figure 41-31. Weak Pull-Up Current, PIC16LF18426/46 only

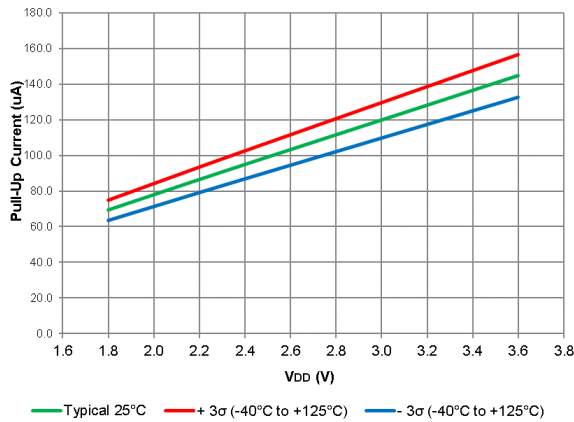


Figure 41-32. V_{OH} vs. I_{OH} Overtemperature, $V_{DD} = 5.5V$, PIC16F18426/46 only

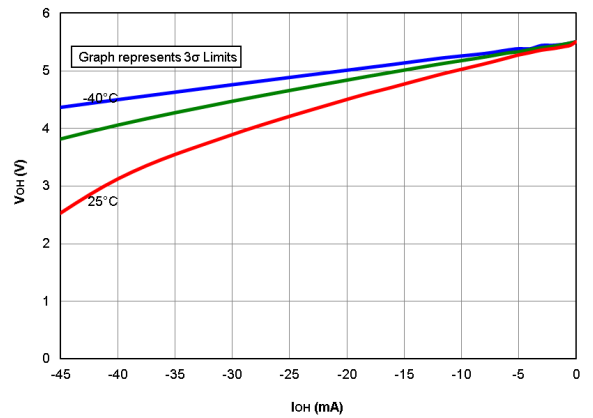


Figure 41-33. V_{OL} vs. I_{OL} Overtemperature, $V_{DD} = 5.5V$, PIC16F18426/46 only

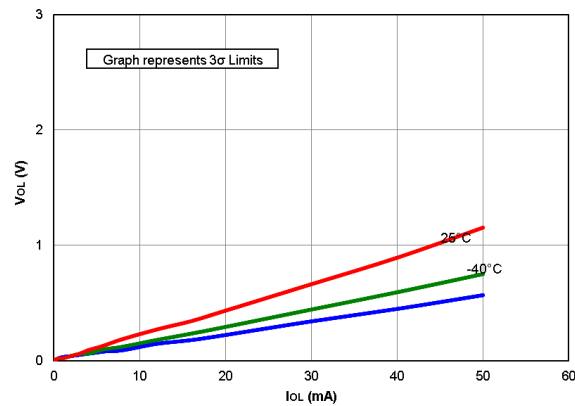


Figure 41-34. V_{OH} vs. I_{OH} Overtemperature, $V_{DD} = 3.0V$, All devices

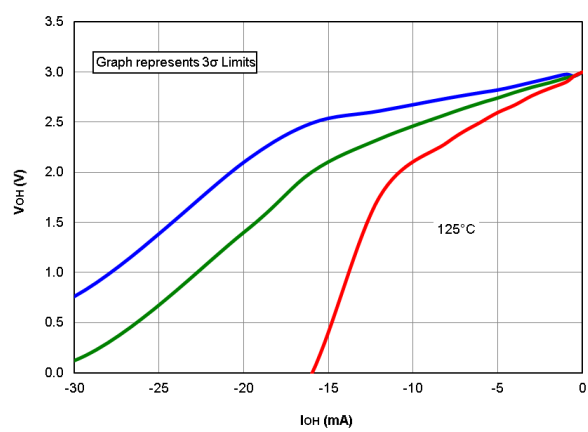


Figure 41-35. V_{OL} vs. I_{OL} Overtemperature, $V_{DD} = 3.0V$, All devices

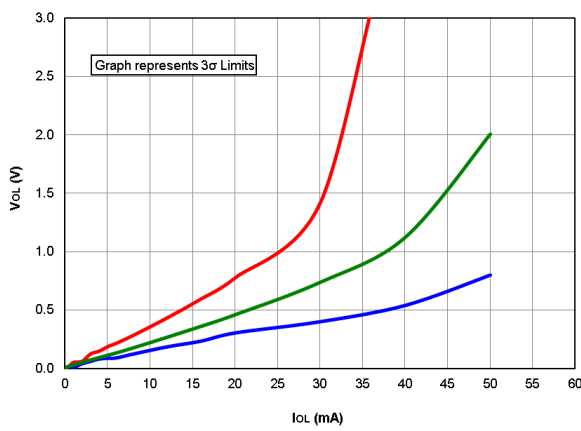


Figure 41-36. V_{OH} vs. I_{OH} Overtemperature, $V_{DD} = 1.8V$, PIC16LF18426/46 only

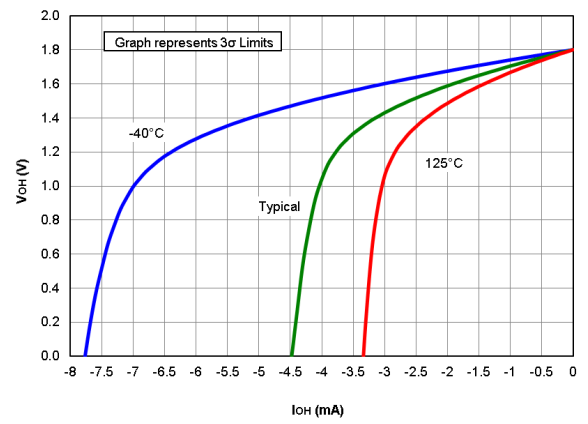


Figure 41-37. V_{OL} vs. I_{OL} Overtemperature, $V_{DD} = 1.8V$, PIC16LF18426/46 only

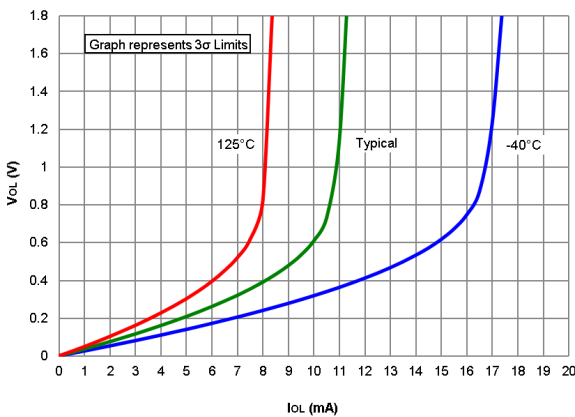


Figure 41-38. Brown-Out Reset Voltage, Trip Point BORV = 00, All devices

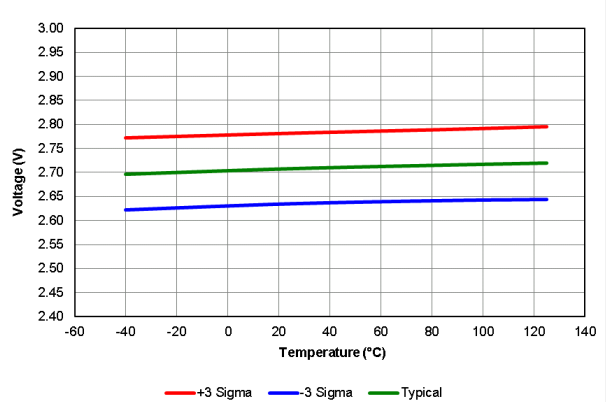


Figure 41-39. Brown-Out Reset Hysteresis, Low Trip Point BORV = 00, All devices

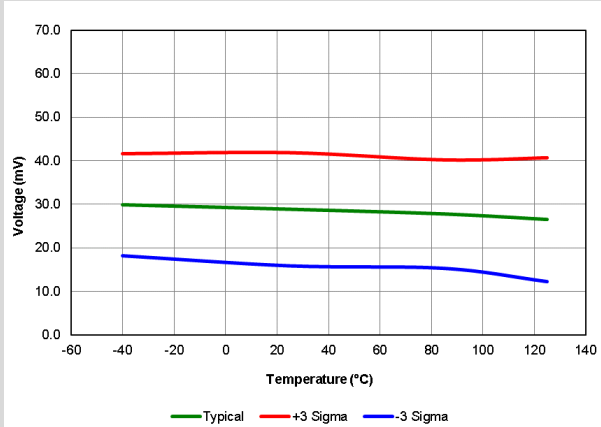


Figure 41-40. Brown-Out Reset Voltage, Trip Point BORV = 01, PIC16LF18426/46 only

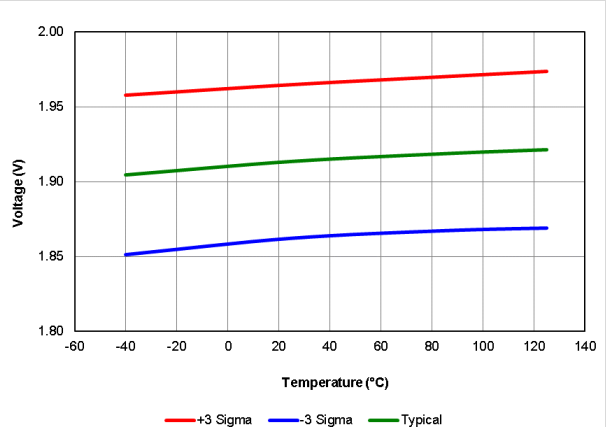


Figure 41-41. Brown-Out Reset Hysteresis, Trip Point BORV = 01, PIC16LF18426/46 only

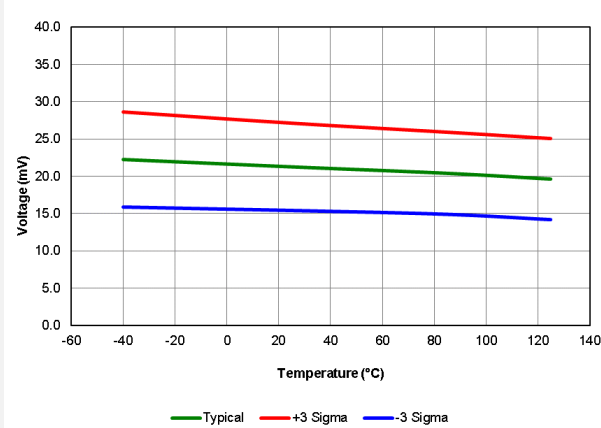


Figure 41-42. Brown-Out Reset Voltage, Trip Point BORV = 01, PIC16F18426/46 only

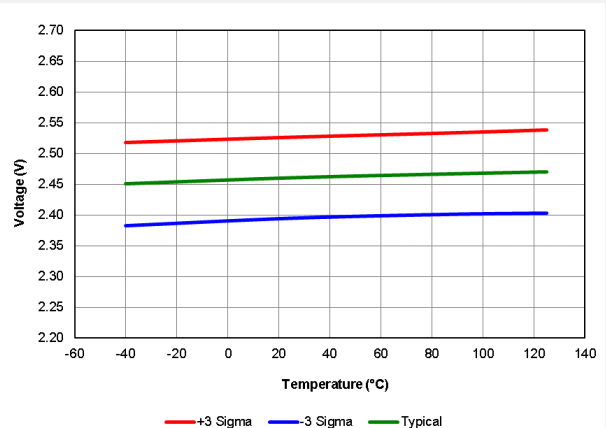


Figure 41-43. Brown-Out Reset Hysteresis, Trip Point BORV = 01, PIC16F18426/46 only

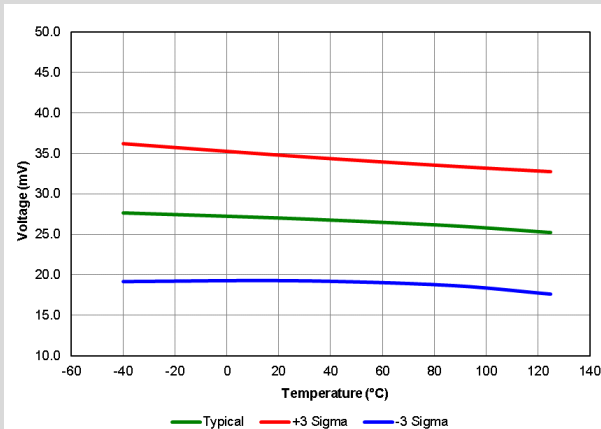


Figure 41-44. LPBOR Reset Voltage, All devices

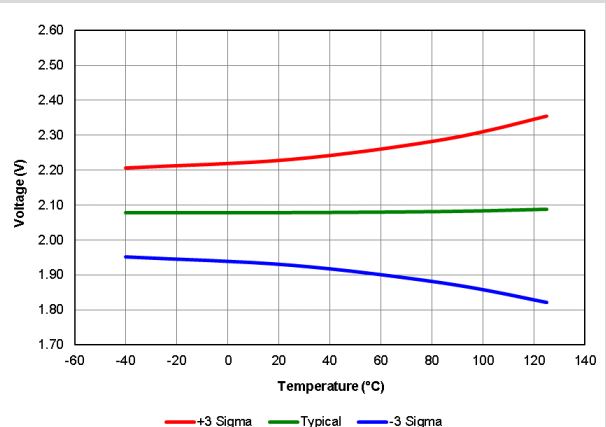


Figure 41-45. LPBOR Reset Hysteresis, All devices

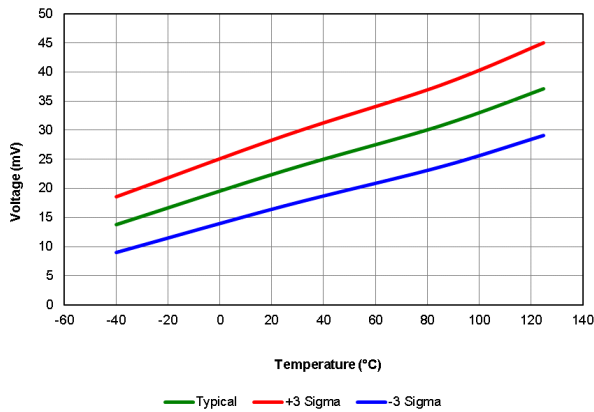


Figure 41-46. BOR Response Time, PIC16LF18426/46 only

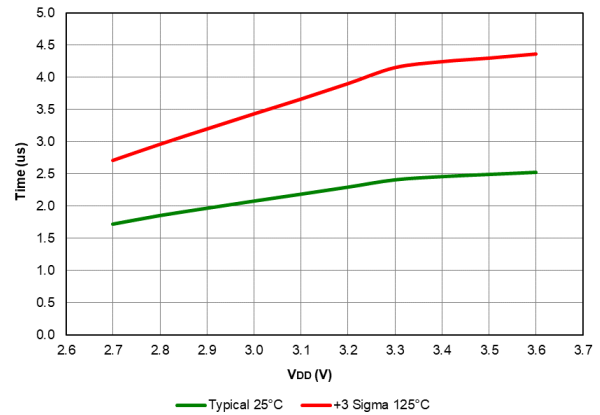


Figure 41-47. BOR Response Time, PIC16F18426/46 only

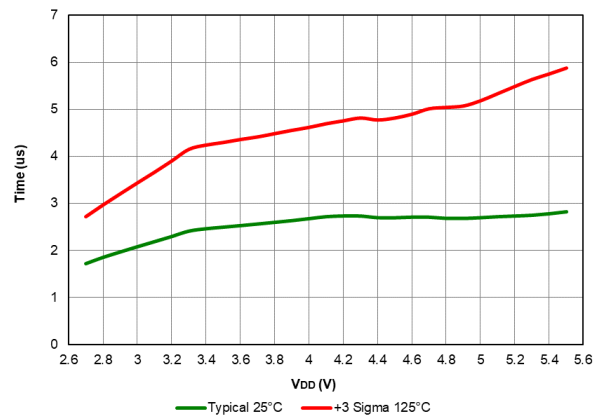


Figure 41-48. ADC 12-Bit Mode, Single-Ended Typical DNL, $V_{DD} = 3.0V$, $V_{REF} = 3.0V$, $T_{AD} = 0.5 \mu s$, 25°C, All devices

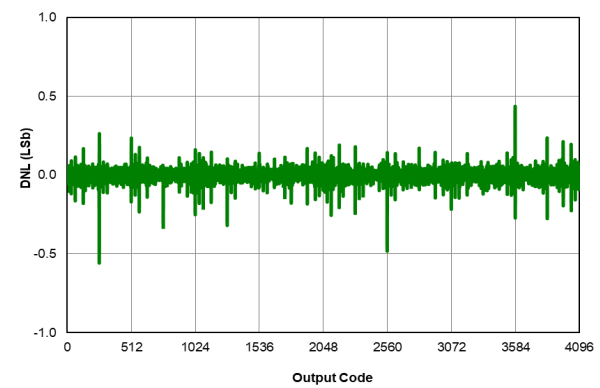


Figure 41-49. ADC 12-Bit Mode, Single-Ended DNL, $V_{DD} = 3.0V$, $V_{REF} = 3.0V$, $T_{AD} = 1 \mu s$, 25°C, All devices

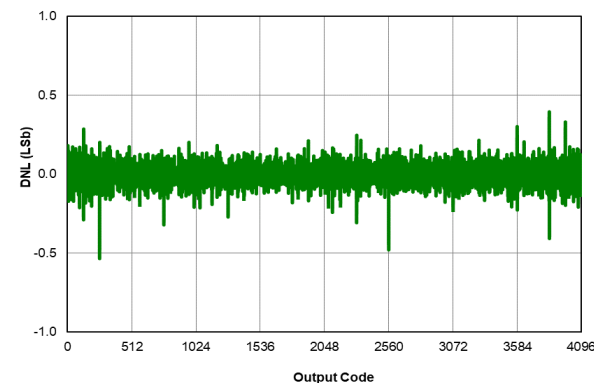


Figure 41-50. ADC 12-Bit Mode, Single-Ended DNL, $V_{DD} = 3.0V$, $V_{REF} = 3.0V$, $T_{AD} = 4 \mu s$, 25°C, All devices

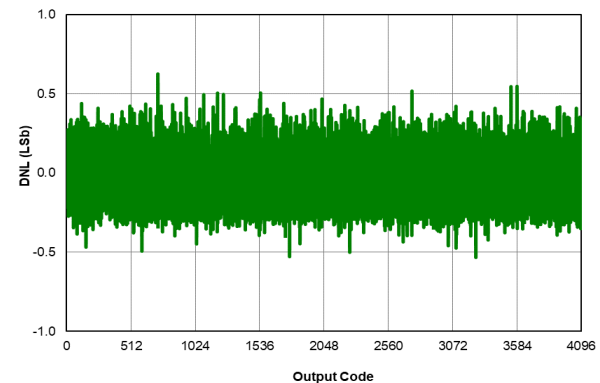


Figure 41-51. ADC 12-Bit Mode, Single-Ended Typical INL, $V_{DD} = 3.0V$, $V_{REF} = 3.0V$, $T_{AD} = 0.5 \mu s$, $25^{\circ}C$, All devices

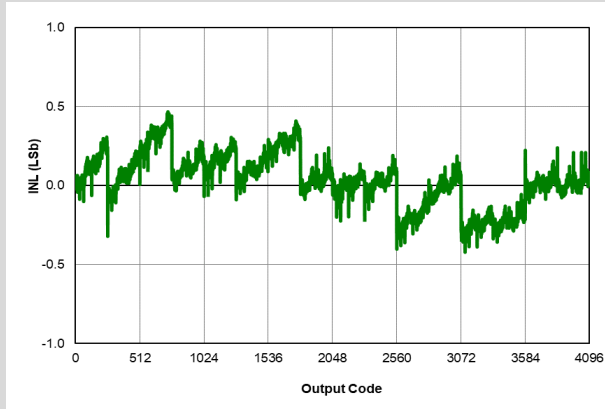


Figure 41-52. ADC 12-Bit Mode, Single-Ended INL, $V_{DD} = 3.0V$, $V_{REF} = 3.0V$, $T_{AD} = 1 \mu s$, $25^{\circ}C$, All devices

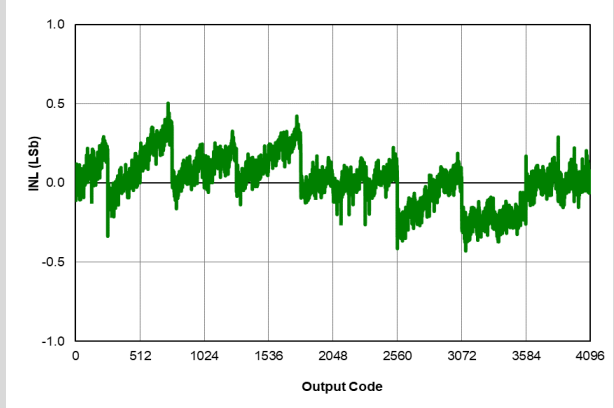


Figure 41-53. ADC 12-Bit Mode, Single-Ended INL, $V_{DD} = 3.0V$, $V_{REF} = 3.0V$, $T_{AD} = 4 \mu s$, $25^{\circ}C$, All devices

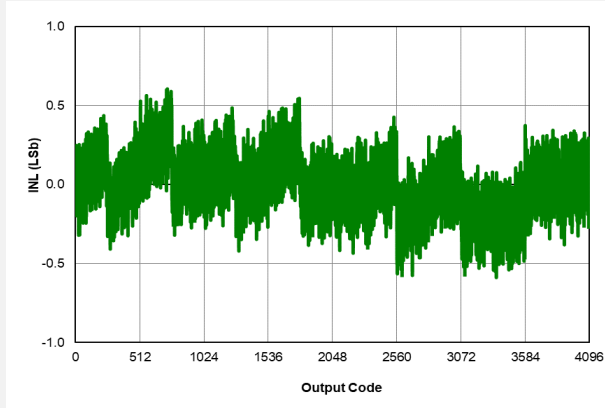


Figure 41-54. ADC 12-Bit Mode, Single-Ended DNL, $V_{DD} = 3.0V$, $V_{REF} = 3.0V$, All devices

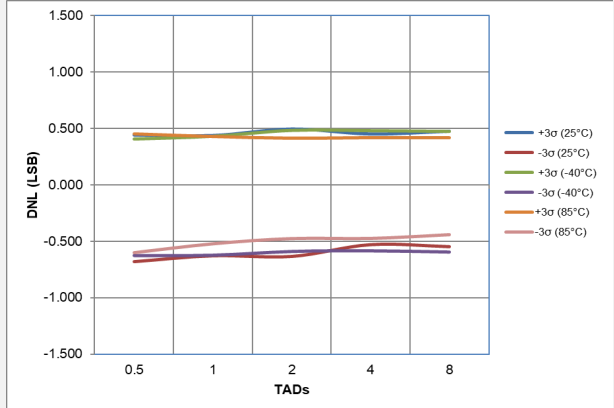


Figure 41-55. ADC 12-Bit Mode, Single-Ended INL, $V_{DD} = 3.0V$, $V_{REF} = 3.0V$, All devices

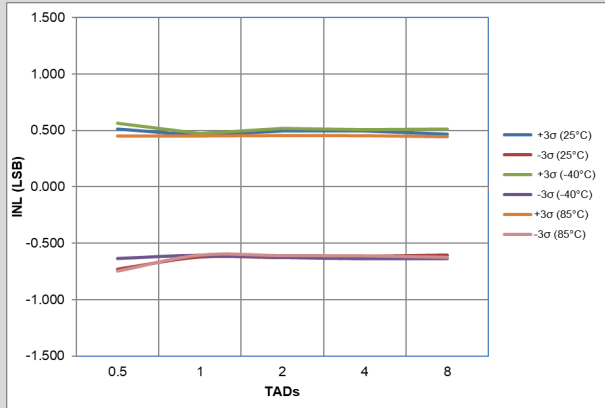


Figure 41-56. ADC 12-Bit Mode, Single-Ended DNL, $V_{DD} = 3.0V$, $T_{AD} = 1 \mu s$, All devices

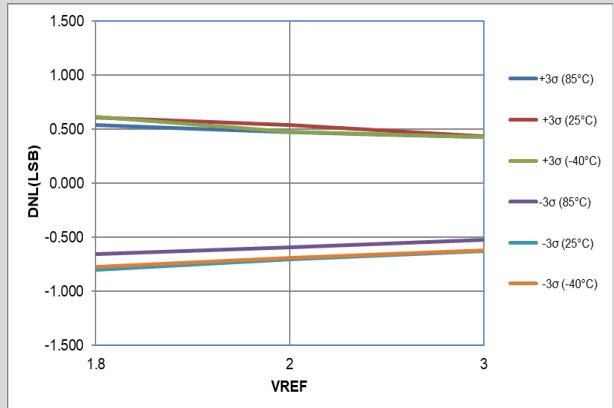


Figure 41-57. ADC 12-Bit Mode, Single-Ended INL, $V_{DD} = 3.0V$, $T_{AD} = 1 \mu s$, All devices

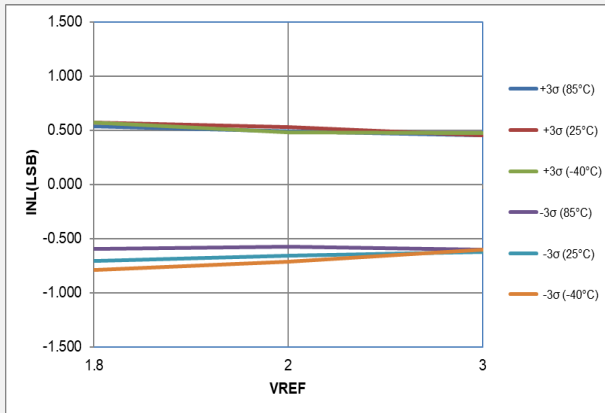


Figure 41-58. ADC 12-Bit Mode, Single-Ended Gain Error, $V_{DD} = 3.0V$, $T_{AD} = 1 \mu s$, All devices

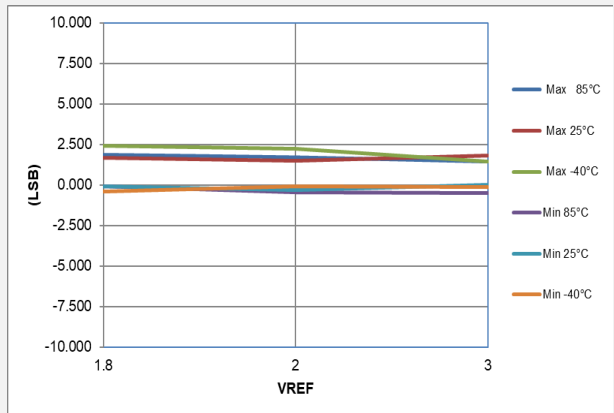


Figure 41-59. ADC 12-Bit Mode, Single-Ended Offset Error, $V_{DD} = 3.0V$, $T_{AD} = 1 \mu s$, All devices

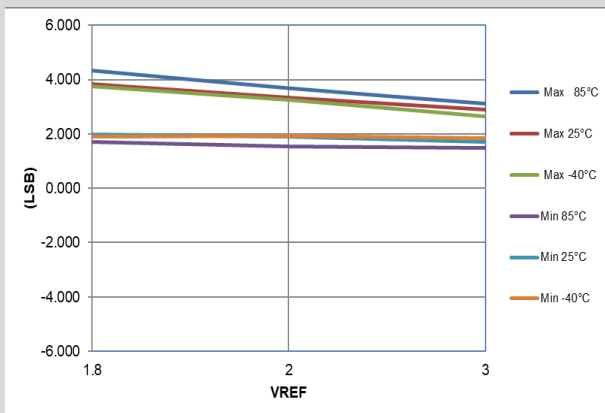


Figure 41-60. ADC 12-Bit Mode, Single-Ended DNL, $V_{DD} = 3.0V$, $T_{AD} = 4 \mu s$, All devices

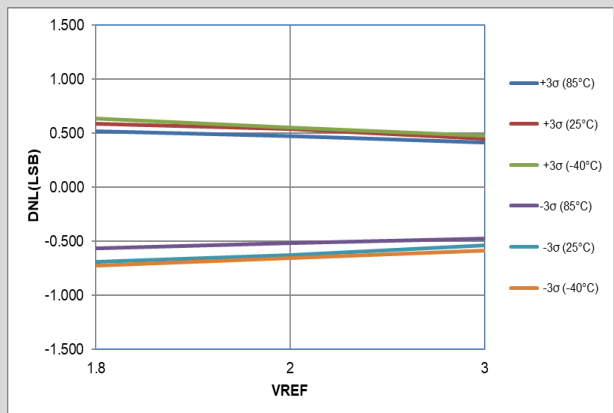


Figure 41-61. ADC 12-Bit Mode, Single-Ended INL, $V_{DD} = 3.0V$, $T_{AD} = 4 \mu s$, All devices

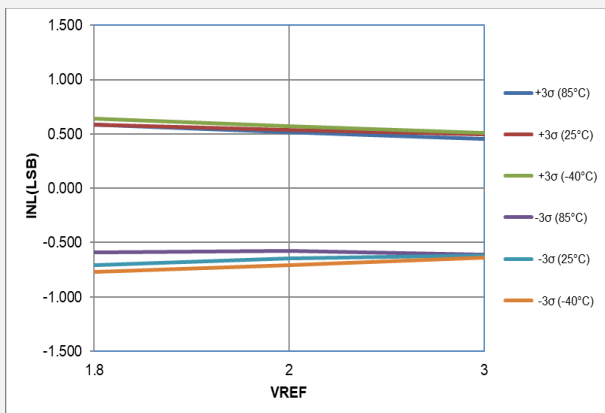


Figure 41-62. ADC 12-Bit Mode, Single-Ended Gain Error, $V_{DD} = 3.0V$, $T_{AD} = 4 \mu s$, All devices

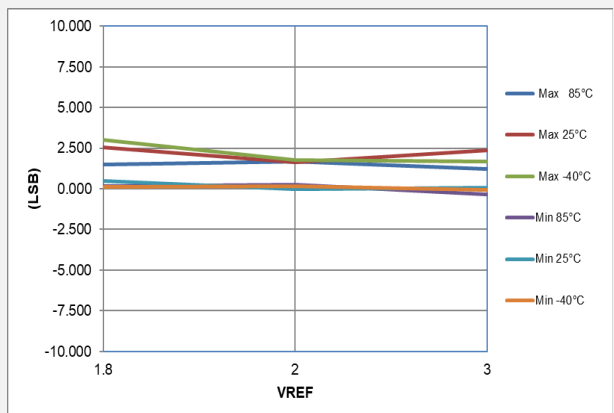


Figure 41-63. ADC 12-Bit Mode, Single-Ended Offset Error, $V_{DD} = 3.0V$, $T_{AD} = 4 \mu s$, All devices

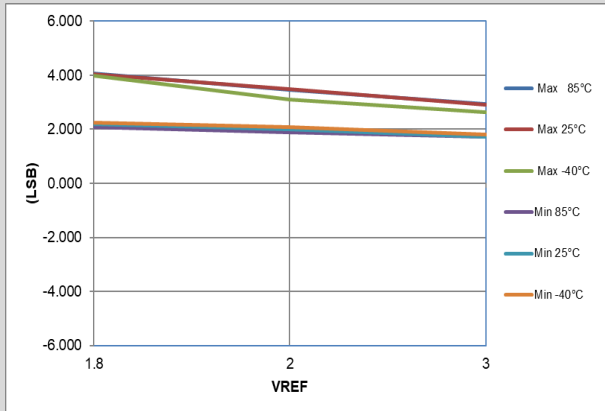


Figure 41-64. ADC 12-Bit Mode, Single-Ended Gain Error, $V_{DD} = 3.0V$, $V_{REF} = 3.0V$, $-40^{\circ}C$ to $85^{\circ}C$, All devices

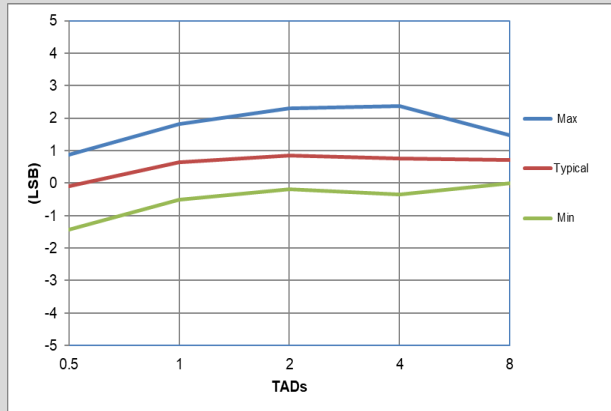


Figure 41-65. ADC 12-Bit Mode, Single-Ended Offset Error, $V_{DD} = 3.0V$, $V_{REF} = 3.0V$, $-40^{\circ}C$ to $85^{\circ}C$, All devices

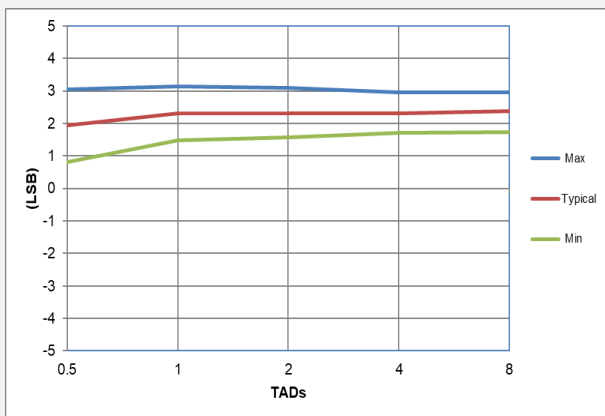


Figure 41-66. ADC RC Oscillator Period, PIC16LF18426/46 only

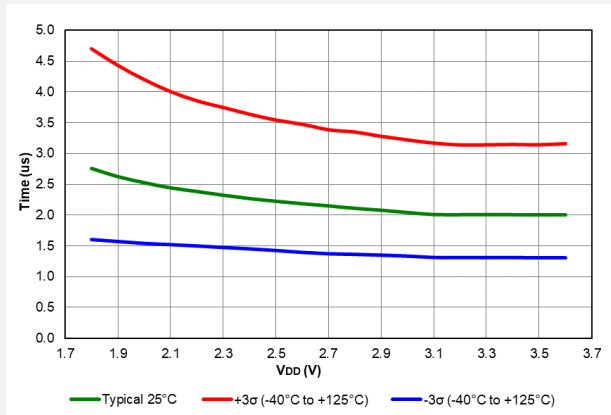


Figure 41-67. ADC RC Oscillator Period, PIC16F18426/46 only

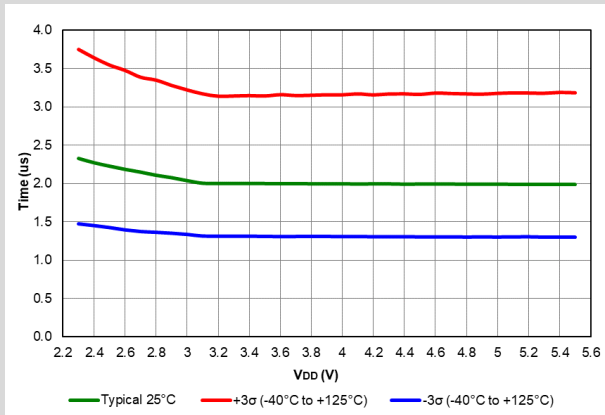


Figure 41-68. Typical DAC DNL Error, $V_{DD} = 3.0V$, $V_{REF} = \text{External } 3V$, All devices

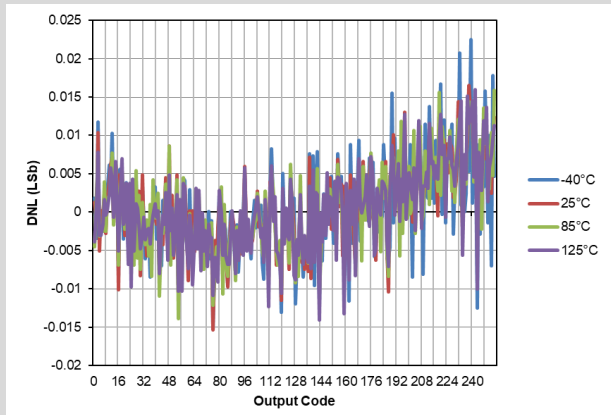


Figure 41-69. Typical DAC INL Error, $V_{DD} = 3.0V$, $V_{REF} = \text{External } 3V$, All devices

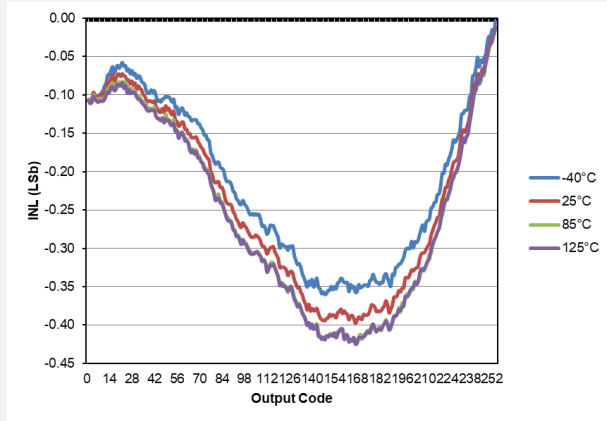


Figure 41-70. Typical DAC DNL Error, $V_{DD} = 5.0V$, $V_{REF} = \text{External } 5V$, PIC16F18426/46 only

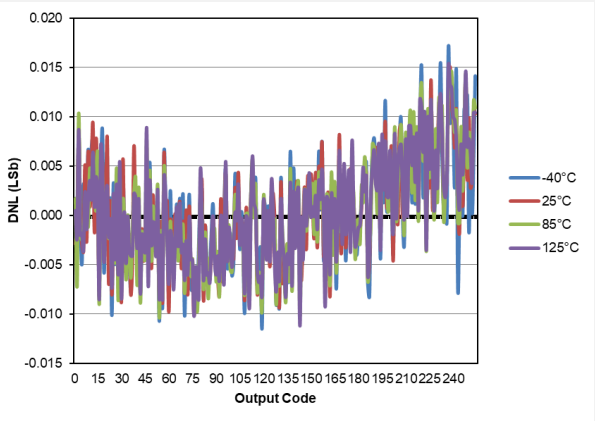


Figure 41-71. Typical DAC INL Error, $V_{DD} = 5.0V$, $V_{REF} = \text{External } 5V$, PIC16F18426/46 only

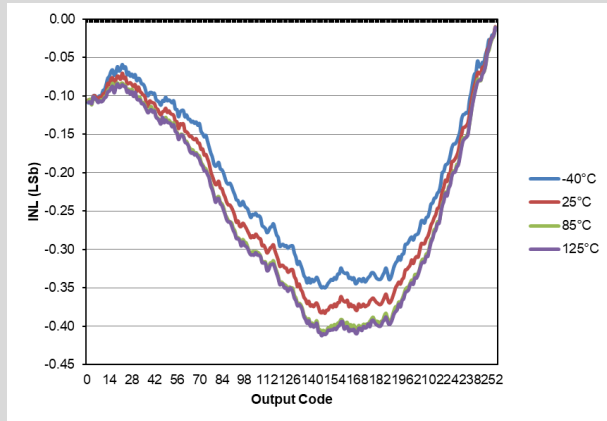


Figure 41-72. DAC DNL Error, $V_{DD} = 3.0V$, PIC16LF18426/46 only

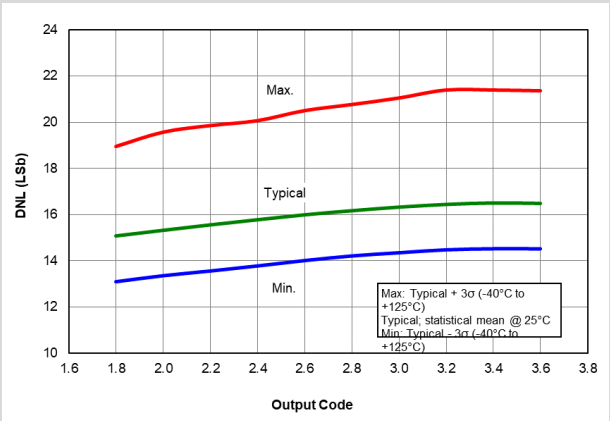


Figure 41-73. Absolute Value of DAC DNL Error, $V_{DD} = 3.0V$, $V_{REF} = V_{DD}$, All devices

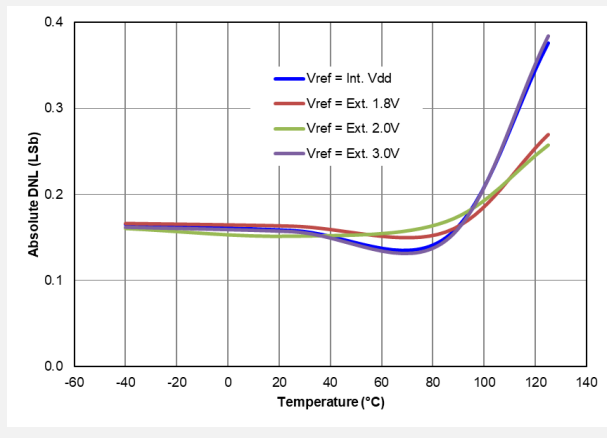


Figure 41-74. Absolute Value of DAC INL Error, $V_{DD} = 3.0V$, $V_{REF} = V_{DD}$, All devices

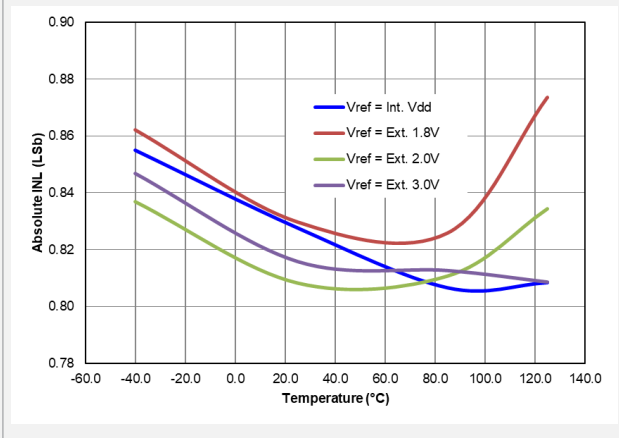


Figure 41-75. Absolute Value of DAC DNL Error, $V_{DD} = 5.0V$, $V_{REF} = V_{DD}$, PIC16F18426/46 only

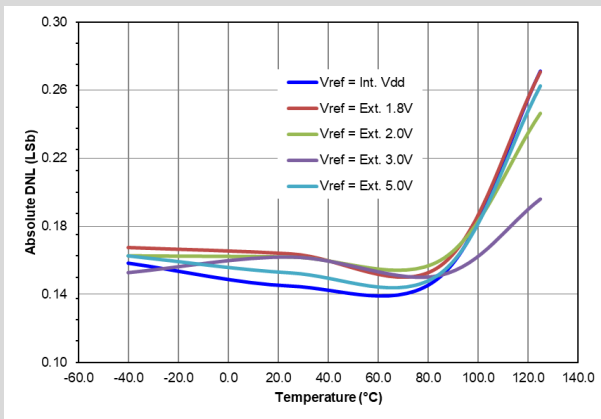


Figure 41-76. Absolute Value of DAC INL Error, $V_{DD} = 5.0V$, $V_{REF} = V_{DD}$, PIC16F18426/46 only

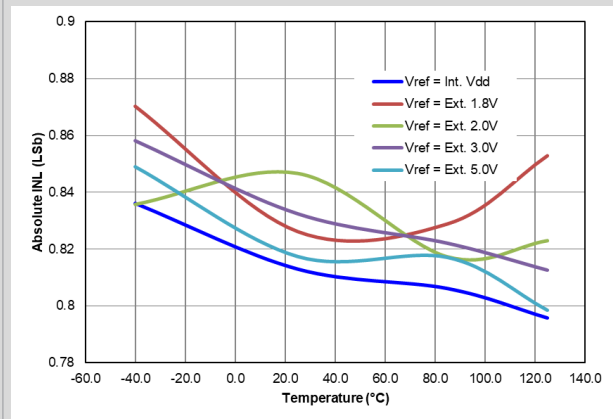


Figure 41-77. Comparator Hysteresis, Normal Power Mode ($CxSP = 1$), $V_{DD} = 3.0V$, Typical Measured Values, All devices

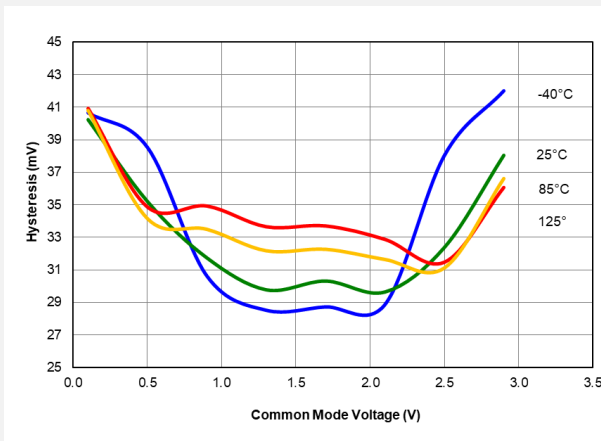


Figure 41-78. Comparator Offset, Normal Power Mode ($CxSP = 1$), $V_{DD} = 3.0V$, Typical Measured Values at 25°C, All devices

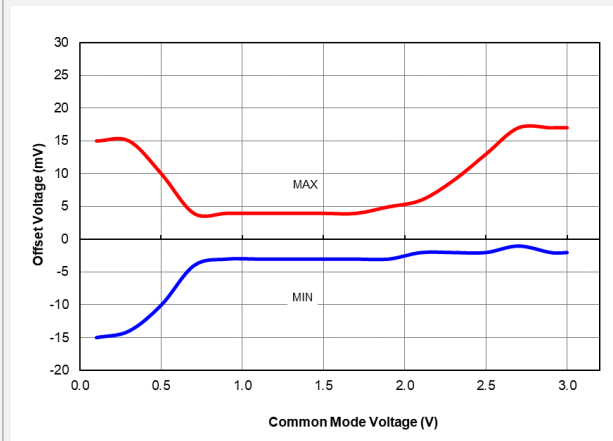


Figure 41-79. Comparator Offset, Normal Power Mode ($CxSP = 1$), $V_{DD} = 3.0V$, Typical Measured Values from -40°C to 125°C, All devices

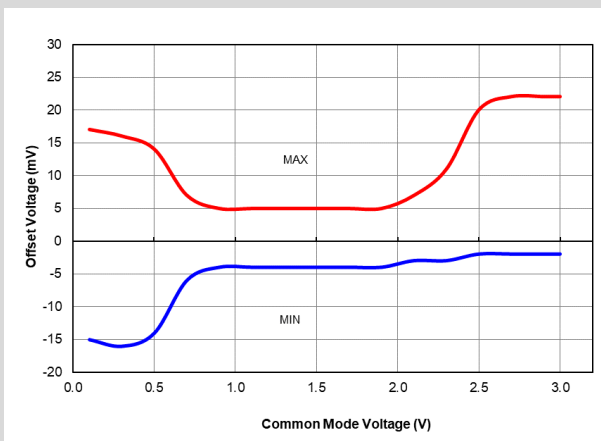


Figure 41-80. Comparator Hysteresis, Normal Power Mode ($CxSP = 1$), $V_{DD} = 5.5V$, Typical Measured Values, PIC16F18426/46 only

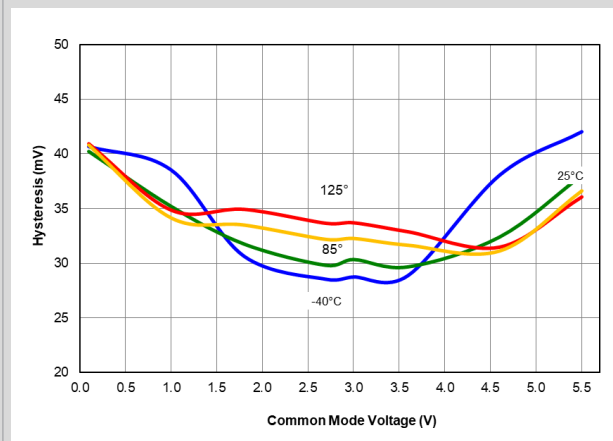


Figure 41-81. Comparator Offset, Normal Power Mode (CxSP = 1), V_{DD} = 5.0V, Typical Measured Values at 25°C, PIC16F18426/46 only

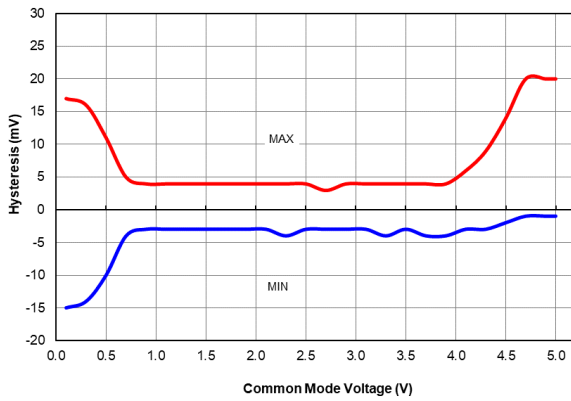


Figure 41-82. Comparator Offset, Normal Power Mode (CxSP = 1), V_{DD} = 5.5V, Typical Measured Values from -40°C to 125°C, PIC16F18426/46 only

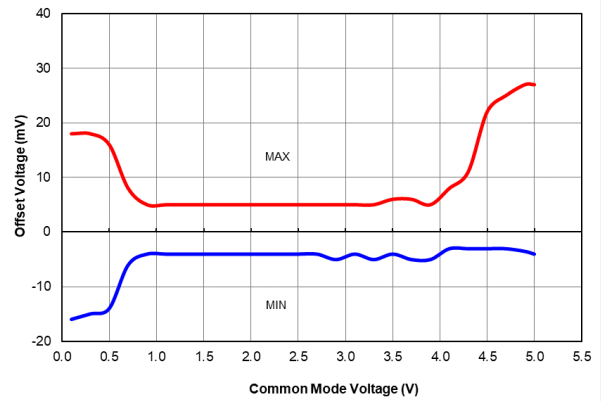


Figure 41-83. Comparator Response Time Overvoltage, Normal Power Mode (CxSP = 1), Typical Measured Values, PIC16F18426/46 only

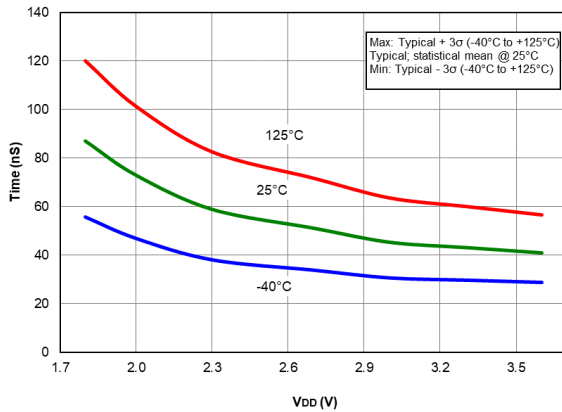


Figure 41-84. Comparator Response Time Overvoltage, Normal Power Mode (CxSP = 1), Typical Measured Values, PIC16F18426/46 only

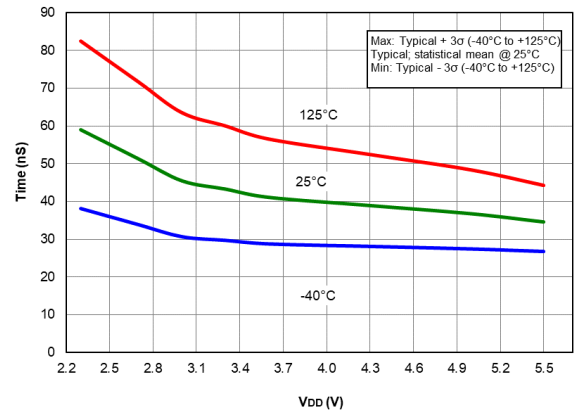


Figure 41-85. Comparator Response Time Falling Edge, PIC16F18426/46 only

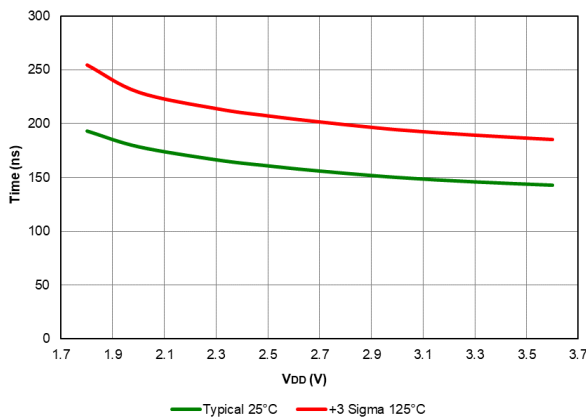


Figure 41-86. Comparator Response Time Falling Edge, PIC16F18426/46 only

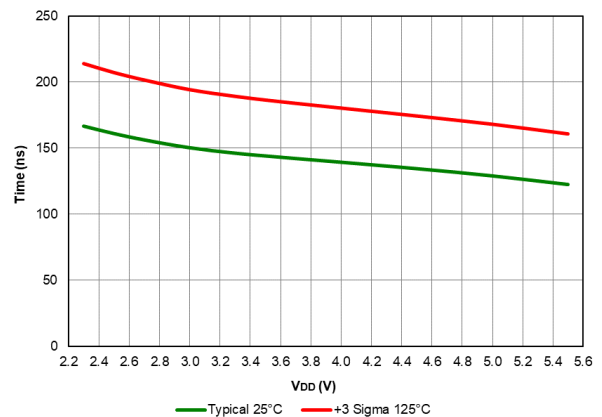


Figure 41-87. Comparator Response Time Rising Edge, PIC16LF18426/46 only

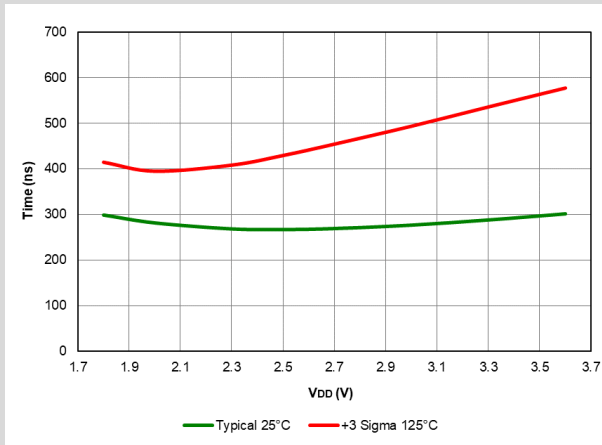


Figure 41-88. Comparator Response Time Rising Edge, PIC16F18426/46 only

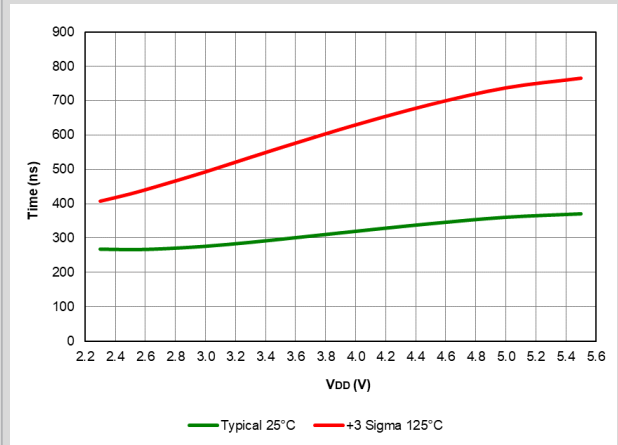


Figure 41-89. Band Gap Ready Time, PIC16LF18426/46 only

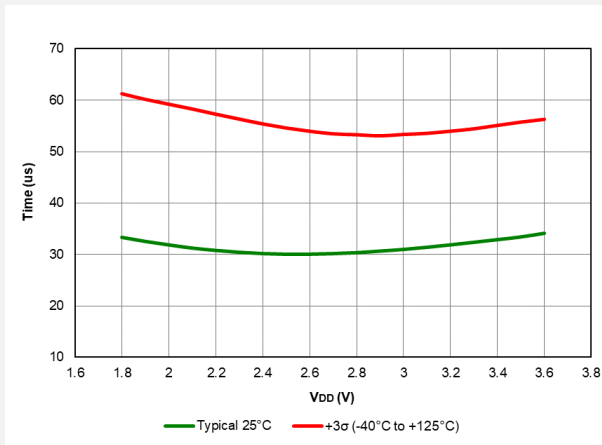


Figure 41-90. FVR Stabilization Period, PIC16LF18426/46 only

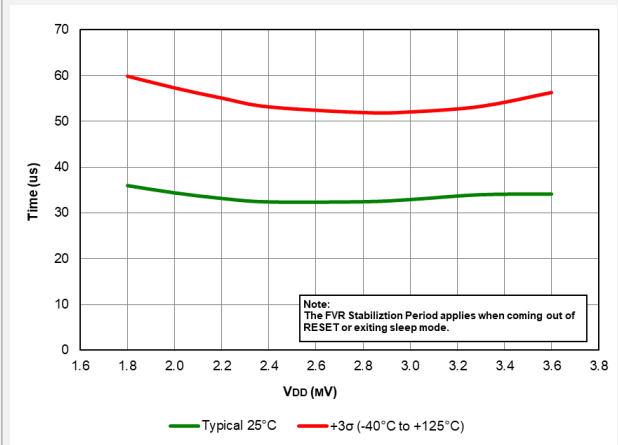


Figure 41-91. Typical FVR Voltage 1x, PIC16LF18426/46 only

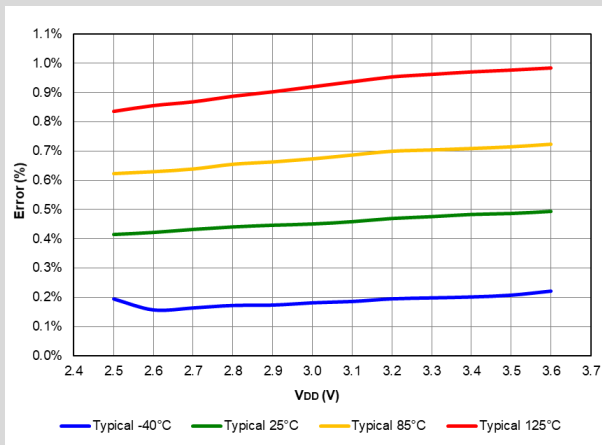


Figure 41-92. FVR Voltage Error 1x, PIC16F18426/46 only

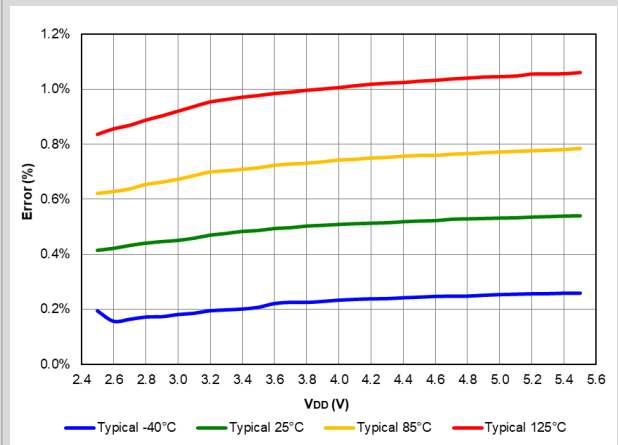


Figure 41-93. FVR Voltage Error 2x, PIC16LF18426/46 only

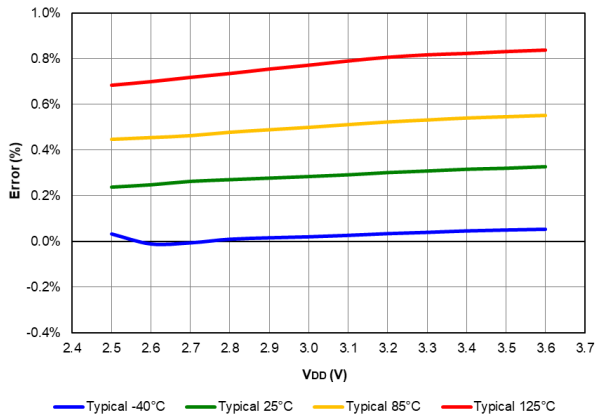


Figure 41-94. FVR Voltage Error 2x, PIC16F18426/46 only

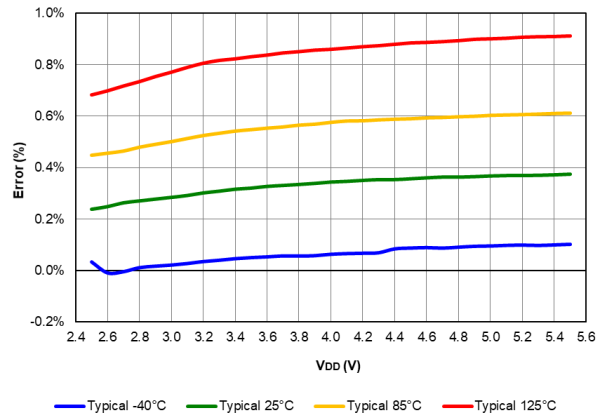


Figure 41-95. FVR Voltage Error 4x, PIC16F18426/46 only

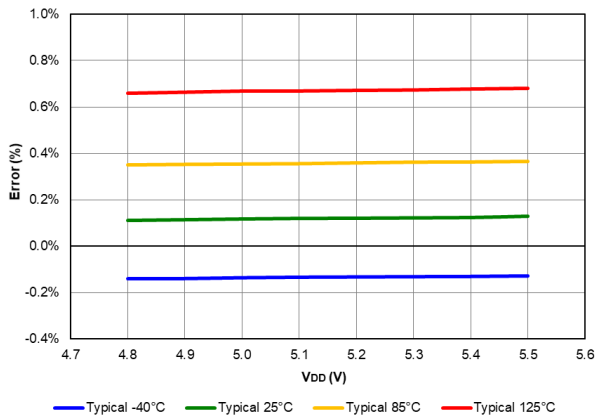


Figure 41-96. Schmitt Trigger High Values, All devices

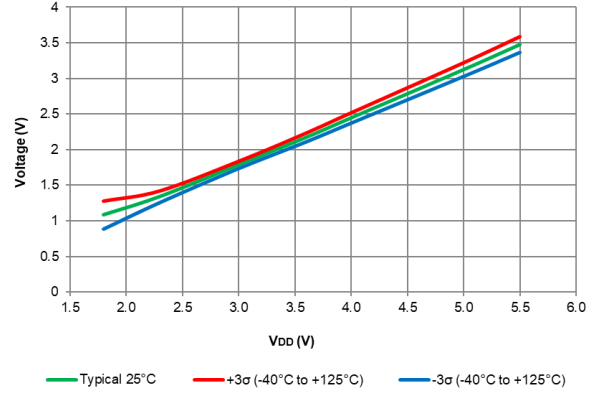


Figure 41-97. Schmitt Trigger Low Values, All devices

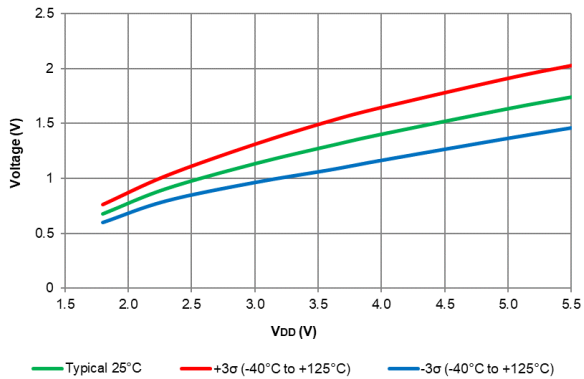


Figure 41-98. Input Level TTL, All devices

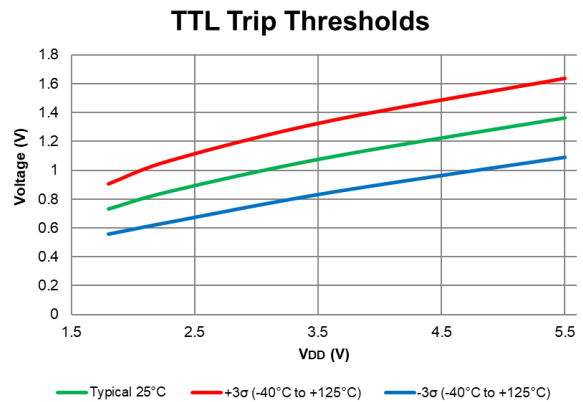


Figure 41-99. Rise Time, Slew Rate Control Enabled, All devices

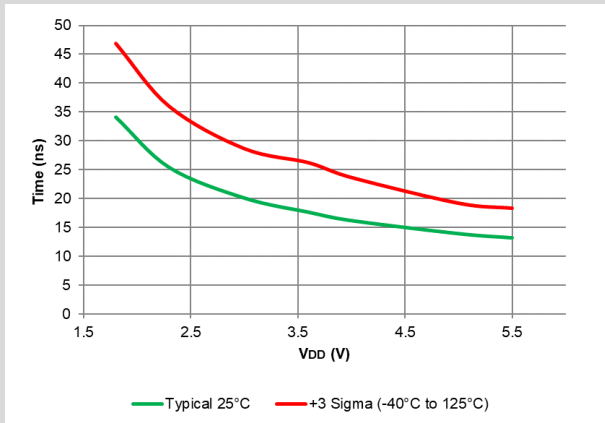


Figure 41-100. Fall Time, Slew Rate Control Enabled, All devices

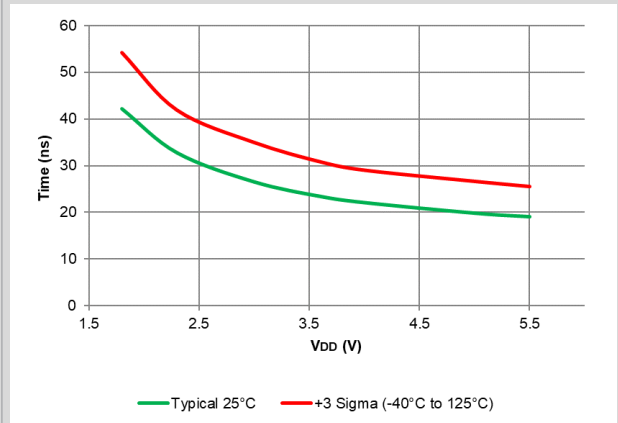


Figure 41-101. Rise Time, Slew Rate Control Disabled, All devices

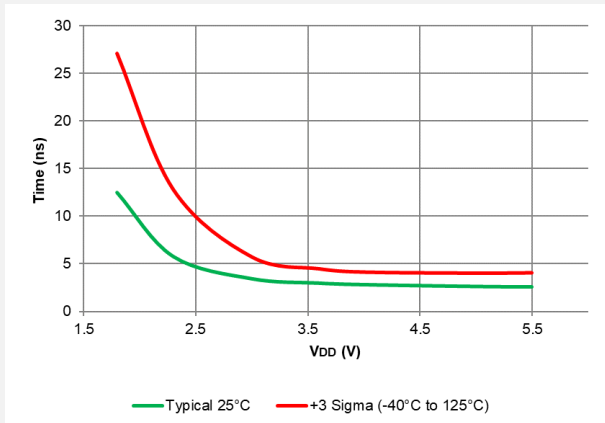


Figure 41-102. Fall Time, Slew Rate Control Disabled, All devices

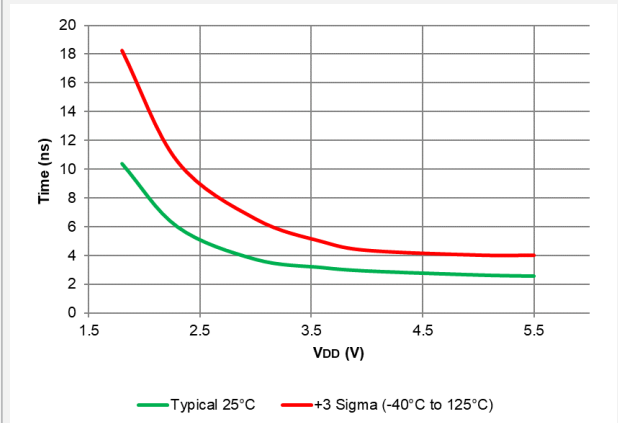


Figure 41-103. OSCTUNE Center Frequency, PIC16LF18426/46 only

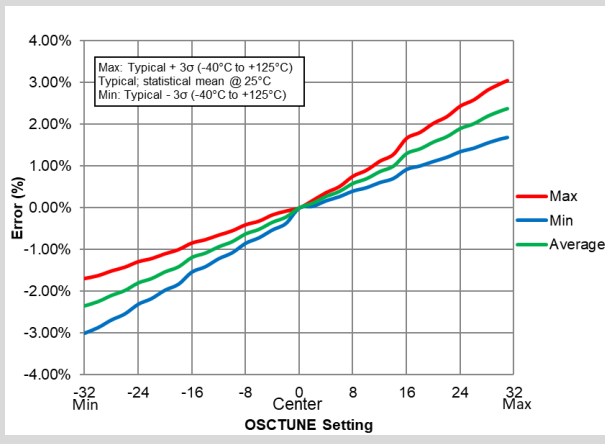


Figure 41-104. POR Release Voltage, All devices

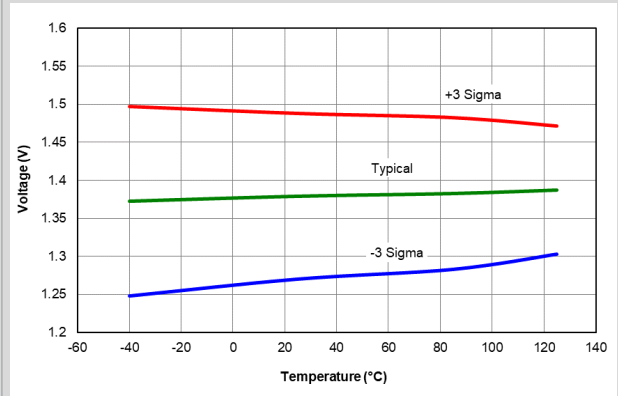


Figure 41-105. POR REARM Voltage, Normal Power Mode, PIC16F18426/46 only

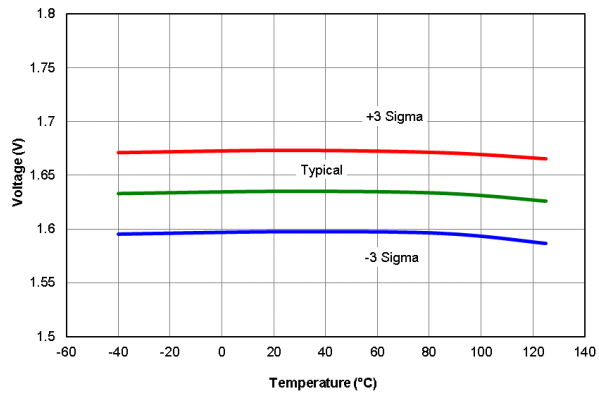


Figure 41-106. PWRT Period, PIC16F18426/46 only

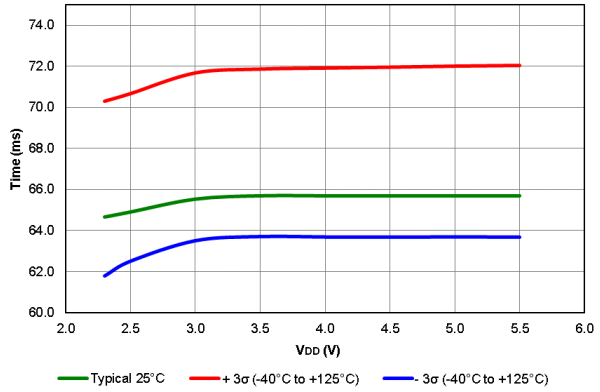


Figure 41-107. PWRT Period, PIC16LF18426/46 only

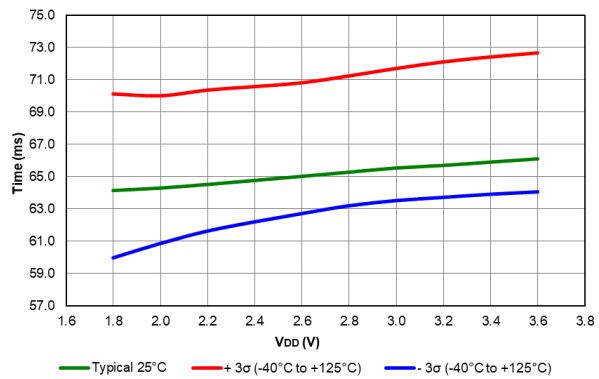


Figure 41-108. Wake from Sleep, VREGPM = 0, HFINTOSC = 4 MHz, PIC16F18426/46 only

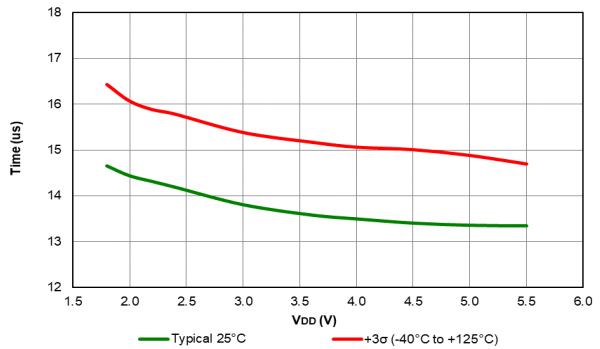


Figure 41-109. Wake from Sleep, VREGPM = 1, HFINTOSC = 4 MHz, PIC16F18426/46 only

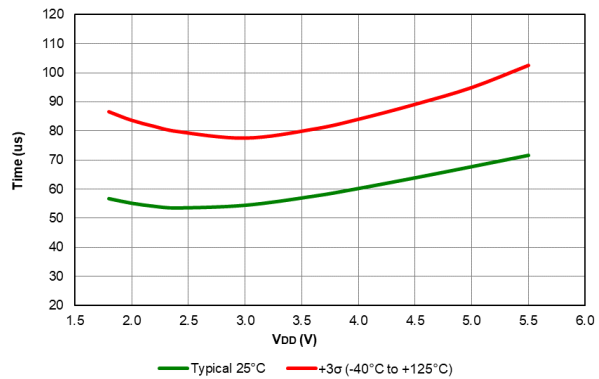


Figure 41-110. Wake from Sleep, VREGPM = 0, HFINTOSC = 16 MHz, PIC16F18426/46 only

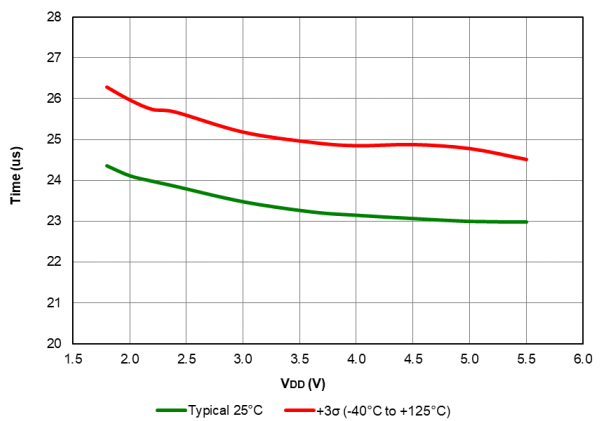


Figure 41-111. Wake from Sleep, VREGPM = 1, HFINTOSC = 16 MHz, PIC16F18426/46 only

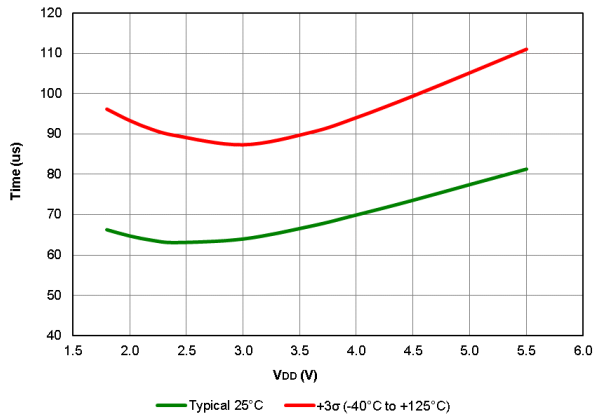


Figure 41-112. Wake from Sleep, VREGPM = 1, PIC16F18426/46 only

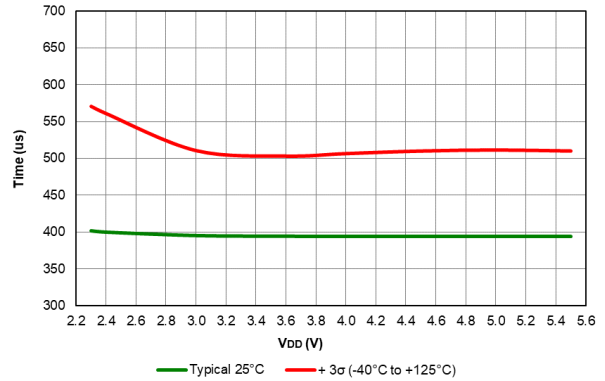


Figure 41-113. Wake from Sleep, PIC16LF18426/46 only

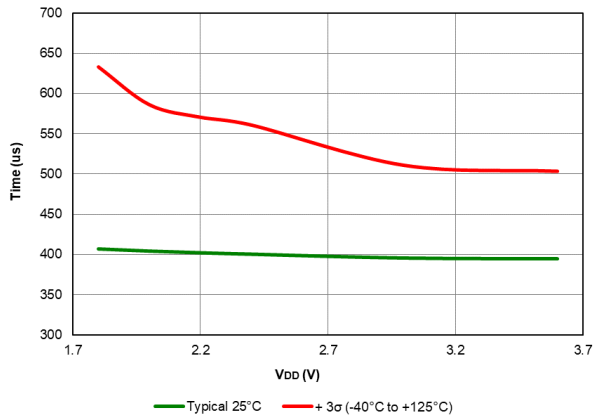


Figure 41-114. WDT Time-Out Period, PIC16F18426/46 only

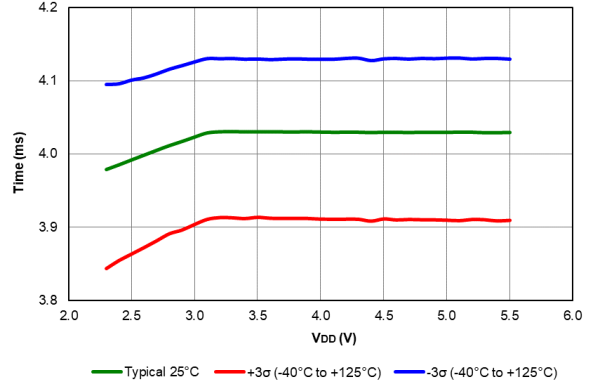
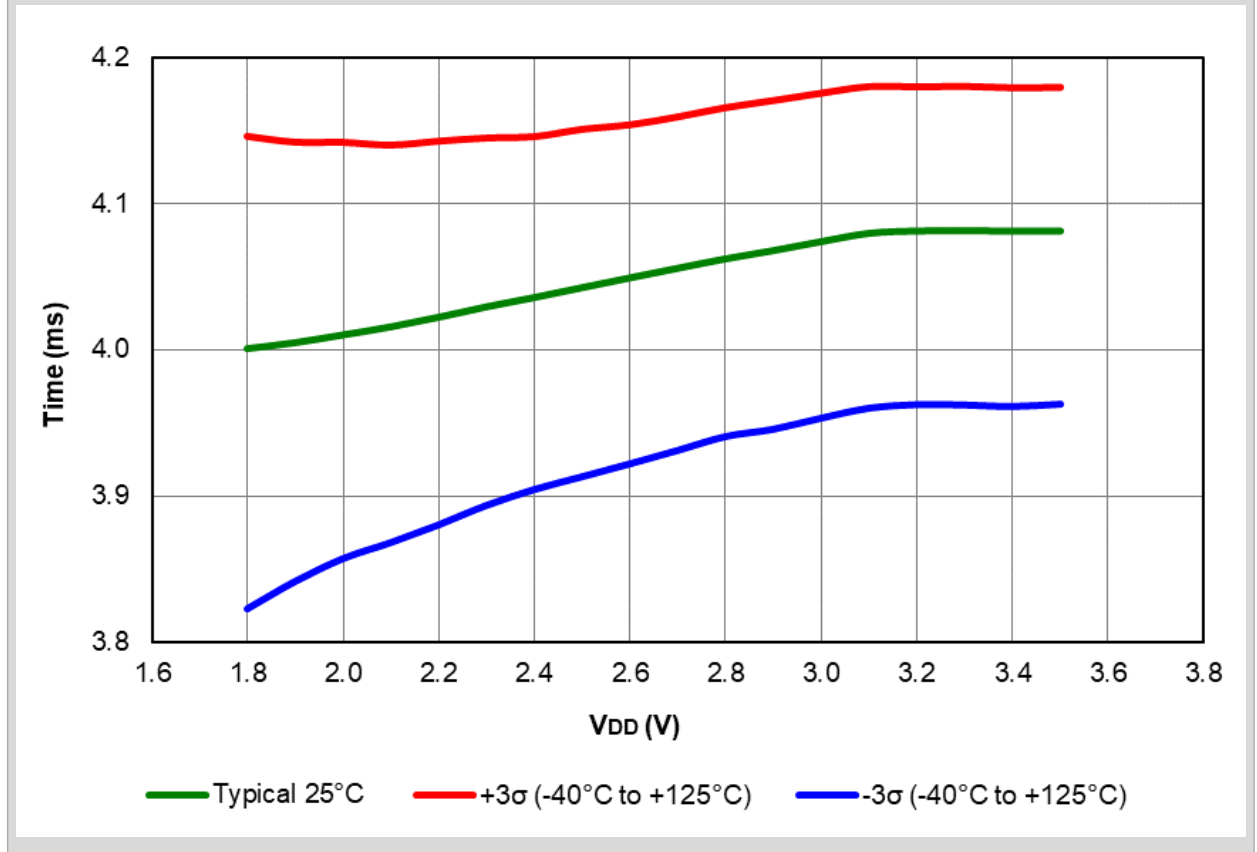


Figure 41-115. WDT Time-Out Period, PIC16LF18426/46 only



42. Packaging Information

Package Marking Information

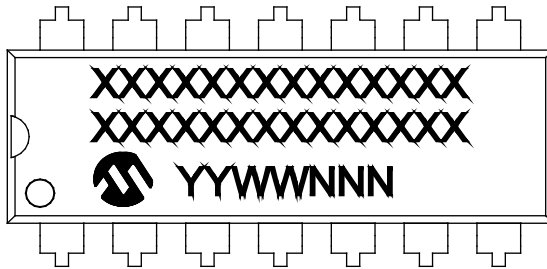
Rev. 30-095000A
5/17/2017

| | | |
|----------------|--------|--|
| Legend: | XX...X | Customer-specific information or Microchip part number |
| | 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) | b-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. |

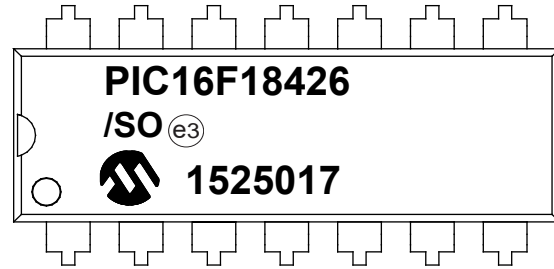
Note: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line, thus limiting the number of available characters for customer-specific information.

Rev. 30-095000A
09/21/2017

14-Lead PDIP (300 mil)

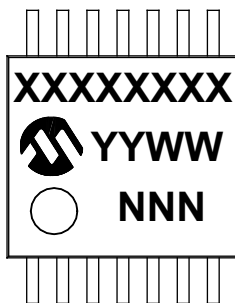


Example

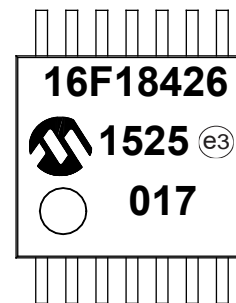


Rev. 30-095014B
09/21/2017

14-Lead TSSOP (4.4 mm)



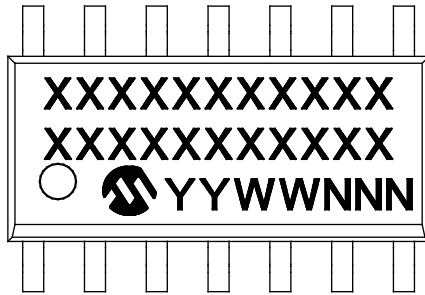
Example



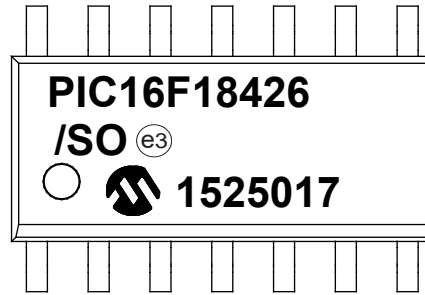
PIC16(L)F18426/46

Packaging Information

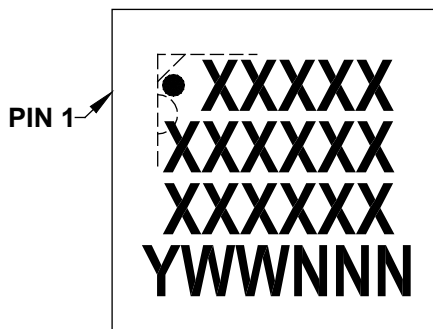
14-Lead SOIC (3.90 mm)



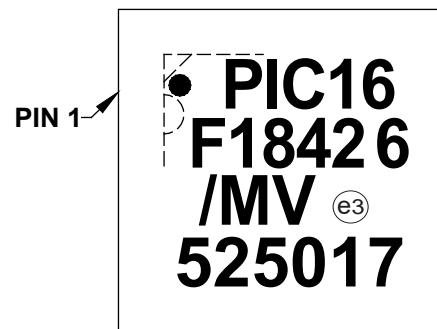
Example



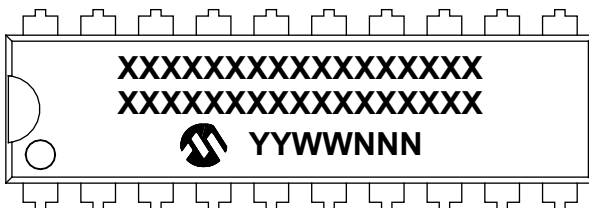
16-Lead UQFN (4x4x0.5 mm)



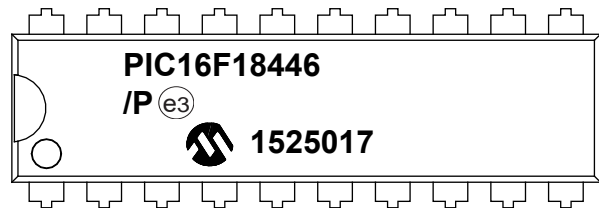
Example



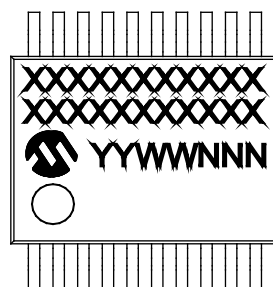
20-Lead PDIP (300 mil)



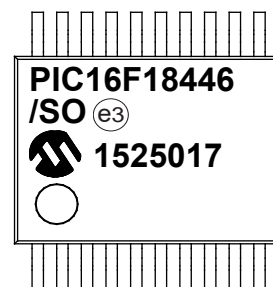
Example



20-Lead SSOP (5.30 mm)



Example

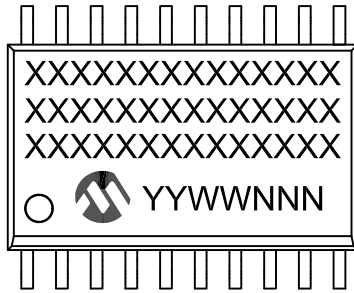


PIC16(L)F18426/46

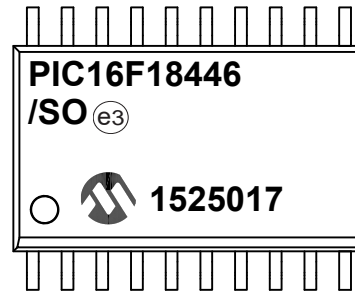
Packaging Information

Rev. 30-00000C
09/21/2017

20-Lead SOIC (7.50 mm)

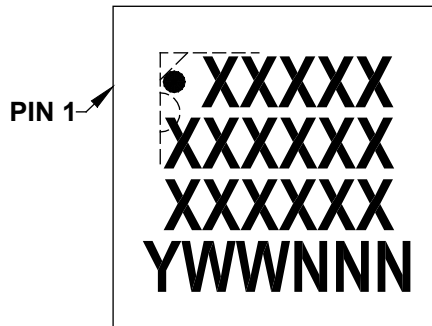


Example

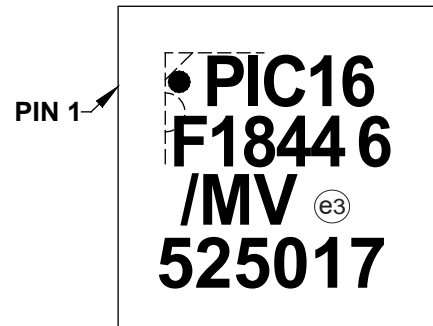


Rev. 30-00000D
09/21/2017

20-Lead UQFN (4x4x0.5 mm)



Example



42.1 Package Details

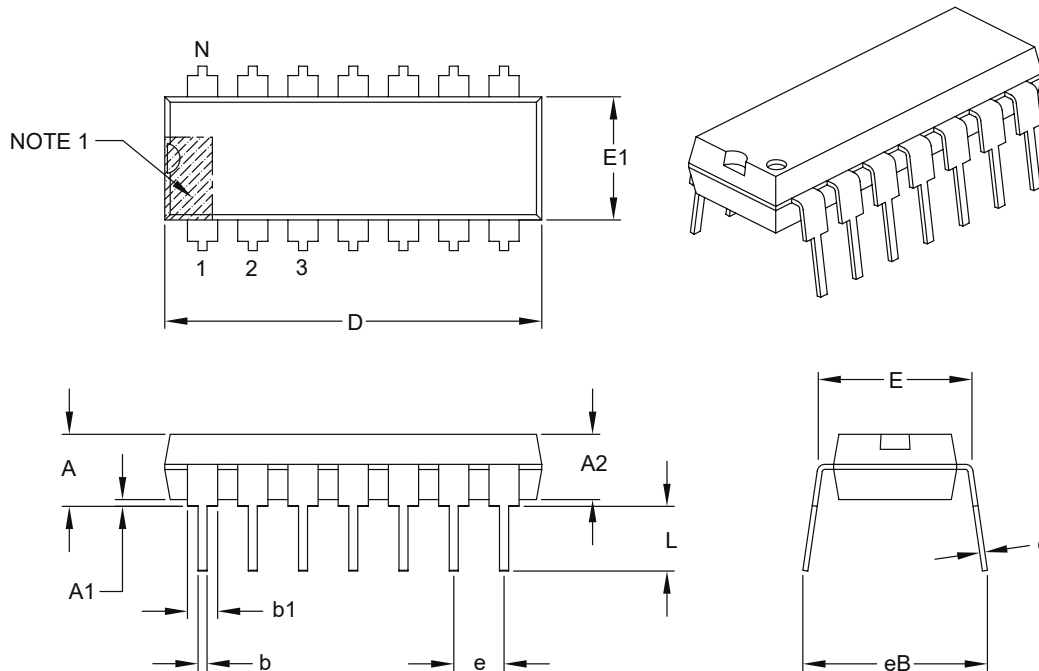
The following sections give the technical details of the packages.

PIC16(L)F18426/46

Packaging Information

14-Lead Plastic Dual In-Line (P) – 300 mil Body [PDIP]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



| Dimension Limits | Units | INCHES | | |
|----------------------------|-------|----------|------|------|
| | | MIN | NOM | MAX |
| Number of Pins | N | 14 | | |
| Pitch | e | .100 BSC | | |
| Top to Seating Plane | A | – | – | .210 |
| Molded Package Thickness | A2 | .115 | .130 | .195 |
| Base to Seating Plane | A1 | .015 | – | – |
| Shoulder to Shoulder Width | E | .290 | .310 | .325 |
| Molded Package Width | E1 | .240 | .250 | .280 |
| Overall Length | D | .735 | .750 | .775 |
| Tip to Seating Plane | L | .115 | .130 | .150 |
| Lead Thickness | c | .008 | .010 | .015 |
| Upper Lead Width | b1 | .045 | .060 | .070 |
| Lower Lead Width | b | .014 | .018 | .022 |
| Overall Row Spacing § | eB | – | – | .430 |

Notes:

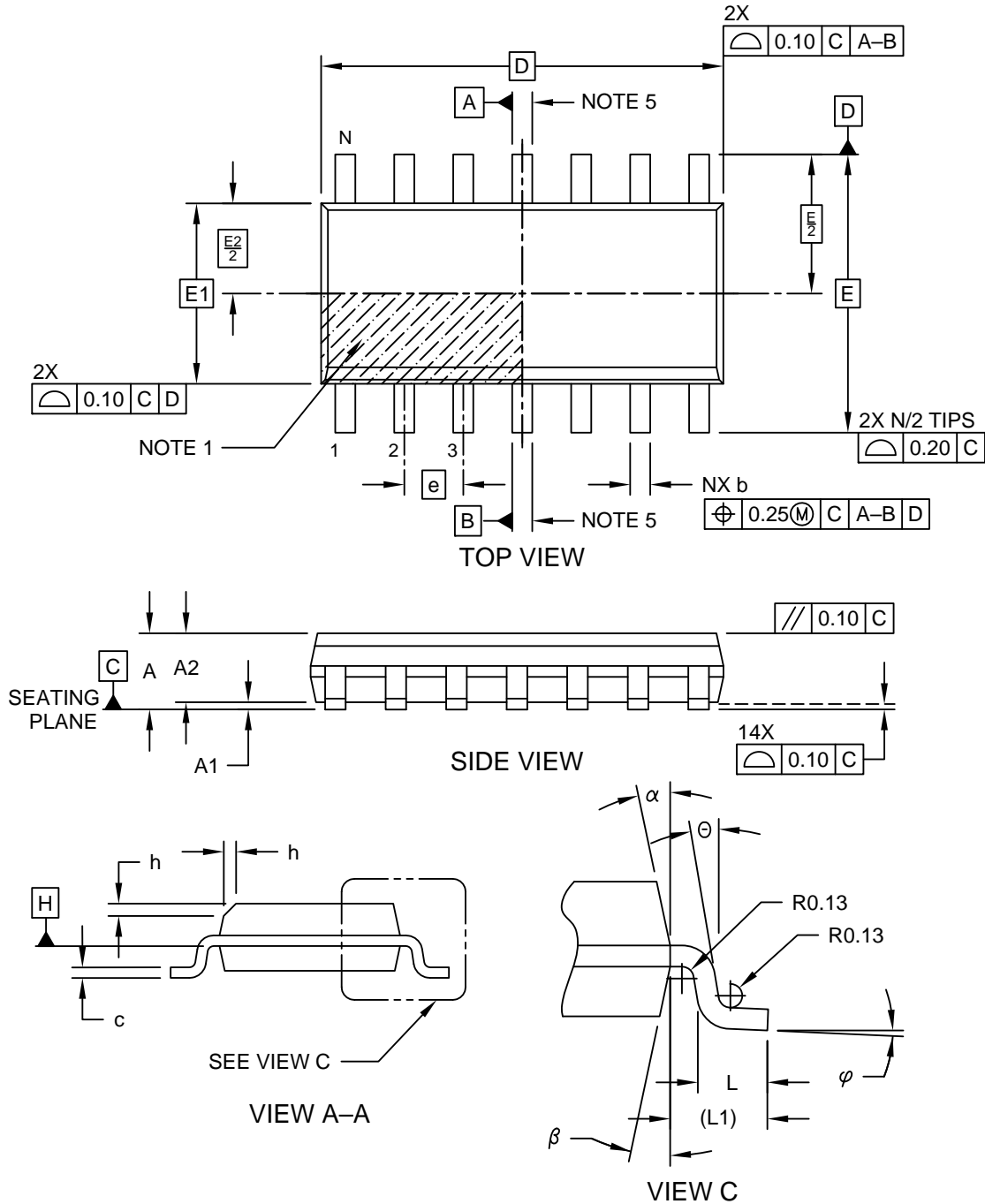
- Pin 1 visual index feature may vary, but must be located with the hatched area.
- § Significant Characteristic.
- Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.
- Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing C04-005B

14-Lead Plastic Small Outline (SL) - Narrow, 3.90 mm Body [SOIC]

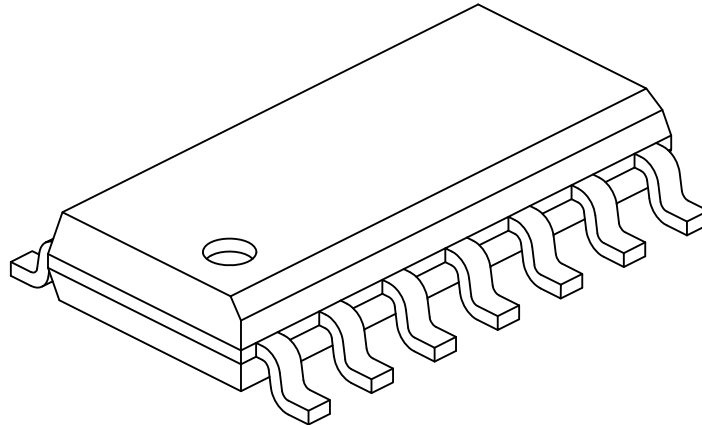
Note: For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



Microchip Technology Drawing No. C04-065-SL Rev D Sheet 1 of 2

14-Lead Plastic Small Outline (SL) - Narrow, 3.90 mm Body [SOIC]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



| Dimension Limits | Units | MILLIMETERS | | |
|--------------------------|-------|-------------|-----|------|
| | | MIN | NOM | MAX |
| Number of Pins | N | 14 | | |
| Pitch | e | 1.27 BSC | | |
| Overall Height | A | - | - | 1.75 |
| Molded Package Thickness | A2 | 1.25 | - | - |
| Standoff § | A1 | 0.10 | - | 0.25 |
| Overall Width | E | 6.00 BSC | | |
| Molded Package Width | E1 | 3.90 BSC | | |
| Overall Length | D | 8.65 BSC | | |
| Chamfer (Optional) | h | 0.25 | - | 0.50 |
| Foot Length | L | 0.40 | - | 1.27 |
| Footprint | L1 | 1.04 REF | | |
| Lead Angle | Θ | 0° | - | - |
| Foot Angle | φ | 0° | - | 8° |
| Lead Thickness | c | 0.10 | - | 0.25 |
| Lead Width | b | 0.31 | - | 0.51 |
| Mold Draft Angle Top | α | 5° | - | 15° |
| Mold Draft Angle Bottom | β | 5° | - | 15° |

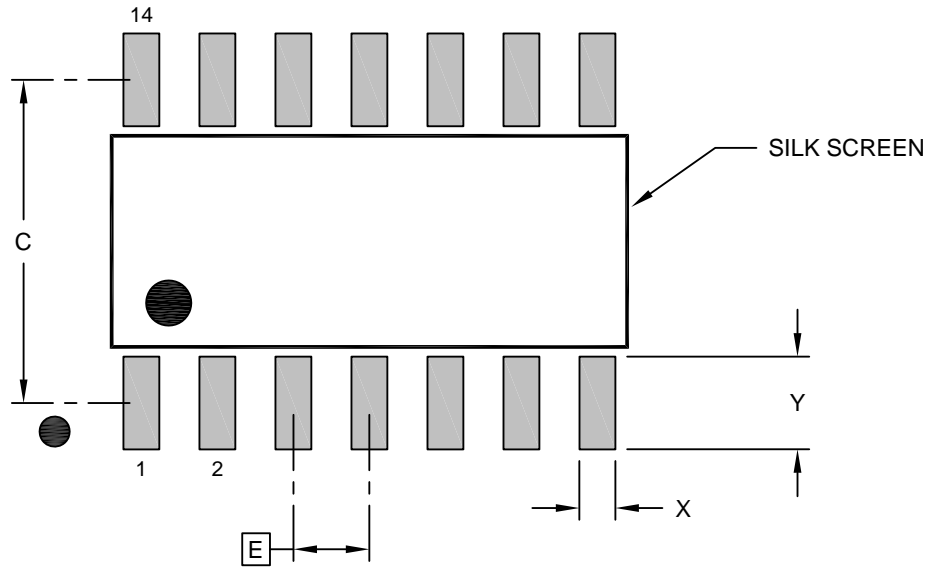
Notes:

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- § Significant Characteristic
- 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.
- 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.
- Datums A & B to be determined at Datum H.

Microchip Technology Drawing No. C04-065-SL Rev D Sheet 2 of 2

14-Lead Plastic Small Outline (SL) - Narrow, 3.90 mm Body [SOIC]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



RECOMMENDED LAND PATTERN

| Dimension Limits | Units | MILLIMETERS | | |
|--------------------------|-------|-------------|----------|------|
| | | MIN | NOM | MAX |
| Contact Pitch | E | | 1.27 BSC | |
| Contact Pad Spacing | C | | 5.40 | |
| Contact Pad Width (X14) | X | | | 0.60 |
| Contact Pad Length (X14) | Y | | | 1.55 |

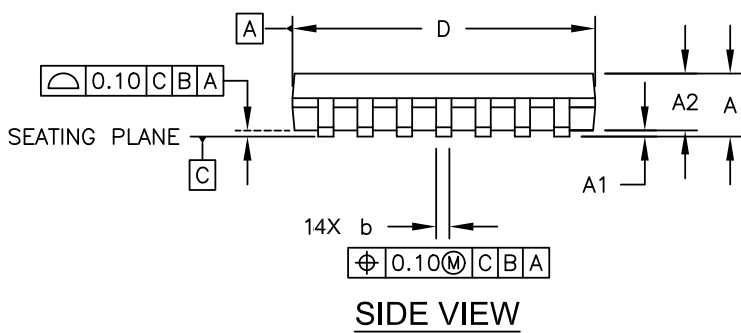
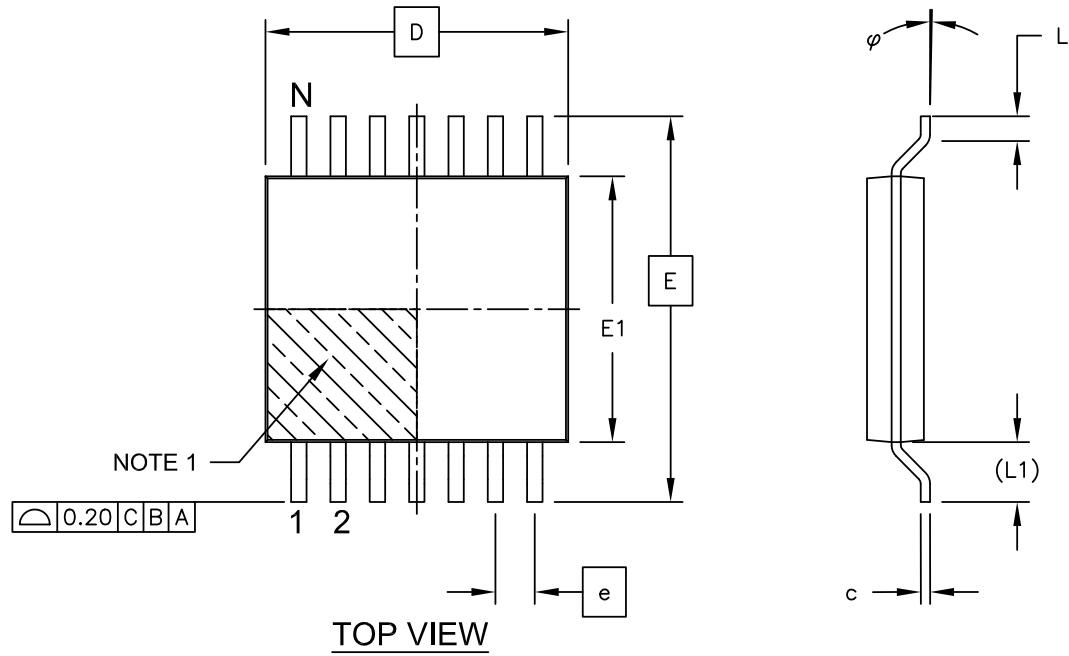
Notes:

- Dimensioning and tolerancing per ASME Y14.5M
 BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2065-SL Rev D

14-Lead Plastic Thin Shrink Small Outline (ST) - 4.4 mm Body [TSSOP]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



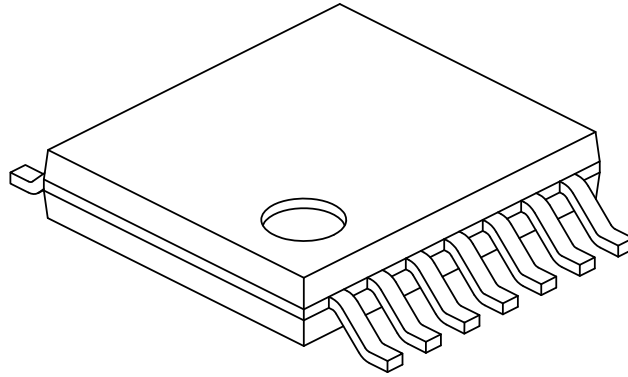
Microchip Technology Drawing C04-087C Sheet 1 of 2

PIC16(L)F18426/46

Packaging Information

14-Lead Plastic Thin Shrink Small Outline (ST) - 4.4 mm Body [TSSOP]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



| Units | | MILLIMETERS | | |
|--------------------------|--------|-------------|------|------|
| Dimension Limits | | MIN | NOM | MAX |
| Number of Pins | N | 14 | | |
| Pitch | e | 0.65 BSC | | |
| Overall Height | A | - | - | 1.20 |
| Molded Package Thickness | A2 | 0.80 | 1.00 | 1.05 |
| Standoff | A1 | 0.05 | - | 0.15 |
| Overall Width | E | 6.40 BSC | | |
| Molded Package Width | E1 | 4.30 | 4.40 | 4.50 |
| Molded Package Length | D | 4.90 | 5.00 | 5.10 |
| Foot Length | L | 0.45 | 0.60 | 0.75 |
| Footprint | (L1) | 1.00 REF | | |
| Foot Angle | ϕ | 0° | - | 8° |
| Lead Thickness | c | 0.09 | - | 0.20 |
| Lead Width | b | 0.19 | - | 0.30 |

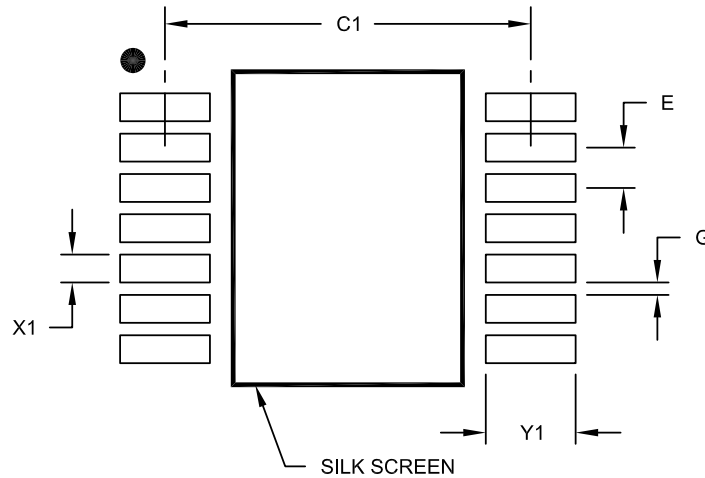
Notes:

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.15mm per side.
- Dimensioning and tolerancing per ASME Y14.5M
 BSC: Basic Dimension. Theoretically exact value shown without tolerances.
 REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing No. C04-087C Sheet 2 of 2

14-Lead Plastic Thin Shrink Small Outline (ST) - 4.4 mm Body [TSSOP]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



RECOMMENDED LAND PATTERN

| Dimension Limits | Units | MILLIMETERS | | |
|--------------------------|-------|-------------|------|------|
| | | MIN | NOM | MAX |
| Contact Pitch | E | 0.65 BSC | | |
| Contact Pad Spacing | C1 | | 5.90 | |
| Contact Pad Width (X14) | X1 | | | 0.45 |
| Contact Pad Length (X14) | Y1 | | | 1.45 |
| Distance Between Pads | G | 0.20 | | |

Notes:

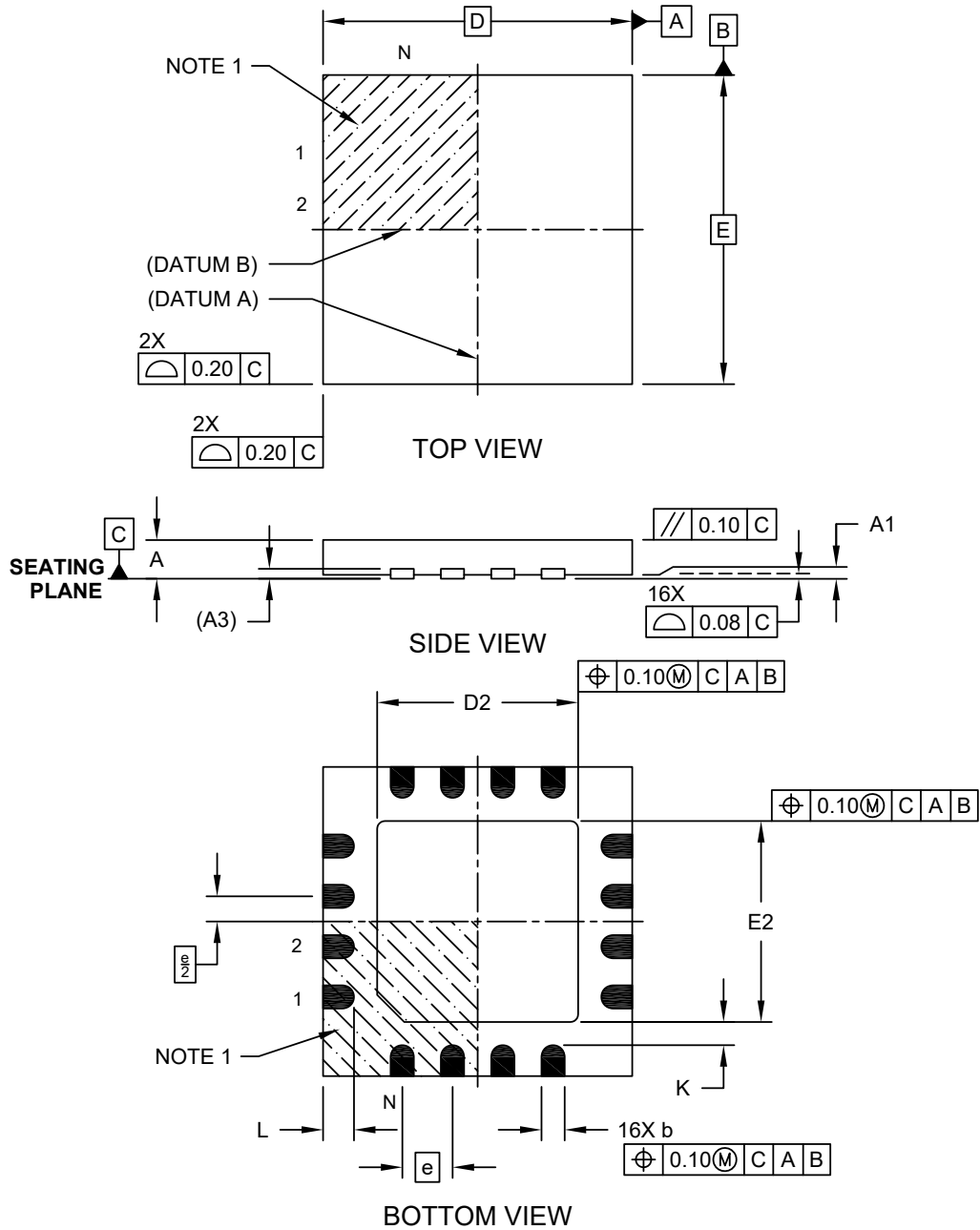
1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2087A

16-Lead Ultra Thin Plastic Quad Flat, No Lead Package (JQ) - 4x4x0.5 mm Body [UQFN]

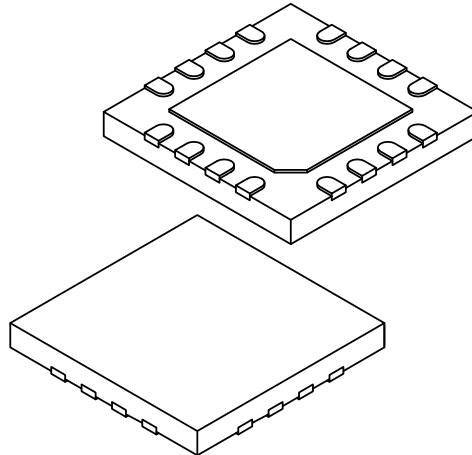
Note: For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



Microchip Technology Drawing C04-257A Sheet 1 of 2

16-Lead Ultra Thin Plastic Quad Flat, No Lead Package (JQ) - 4x4x0.5 mm Body [UQFN]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



| | | Units | MILLIMETERS | | |
|-------------------------|----|-----------|-------------|------|-----|
| Dimension Limits | | | MIN | NOM | MAX |
| Number of Pins | N | | 16 | | |
| Pitch | e | | 0.65 BSC | | |
| Overall Height | A | 0.45 | 0.50 | 0.55 | |
| Standoff | A1 | 0.00 | 0.02 | 0.05 | |
| Terminal Thickness | A3 | 0.127 REF | | | |
| Overall Width | E | 4.00 BSC | | | |
| Exposed Pad Width | E2 | 2.50 | 2.60 | 2.70 | |
| Overall Length | D | 4.00 BSC | | | |
| Exposed Pad Length | D2 | 2.50 | 2.60 | 2.70 | |
| Terminal Width | b | 0.25 | 0.30 | 0.35 | |
| Terminal Length | L | 0.30 | 0.40 | 0.50 | |
| Terminal-to-Exposed-Pad | K | 0.20 | - | - | |

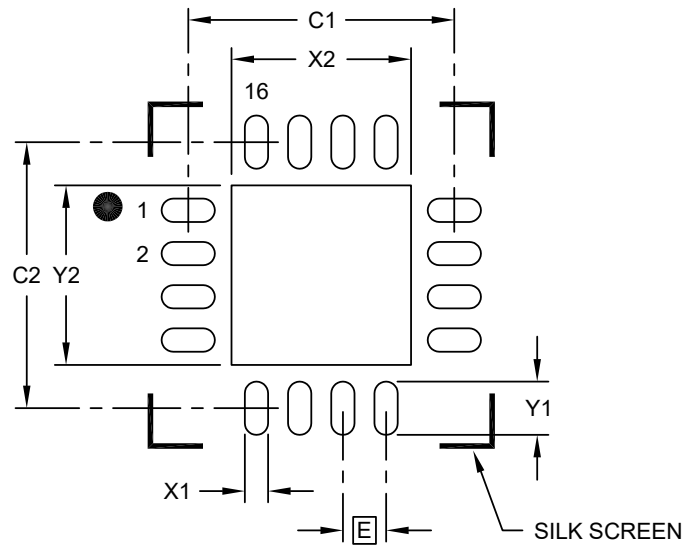
Notes:

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 C04-257A Sheet 2 of 2

16-Lead Ultra Thin Plastic Quad Flat, No Lead Package (JQ) - 4x4x0.5 mm Body [UQFN]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



RECOMMENDED LAND PATTERN

| Dimension Limits | Units | MILLIMETERS | | |
|----------------------------|-------|-------------|------|------|
| | | MIN | NOM | MAX |
| Contact Pitch | E | 0.65 BSC | | |
| Optional Center Pad Width | X2 | | | 2.70 |
| Optional Center Pad Length | Y2 | | | 2.70 |
| Contact Pad Spacing | C1 | | 4.00 | |
| Contact Pad Spacing | C2 | | 4.00 | |
| Contact Pad Width (X16) | X1 | | | 0.35 |
| Contact Pad Length (X16) | Y1 | | | 0.80 |

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

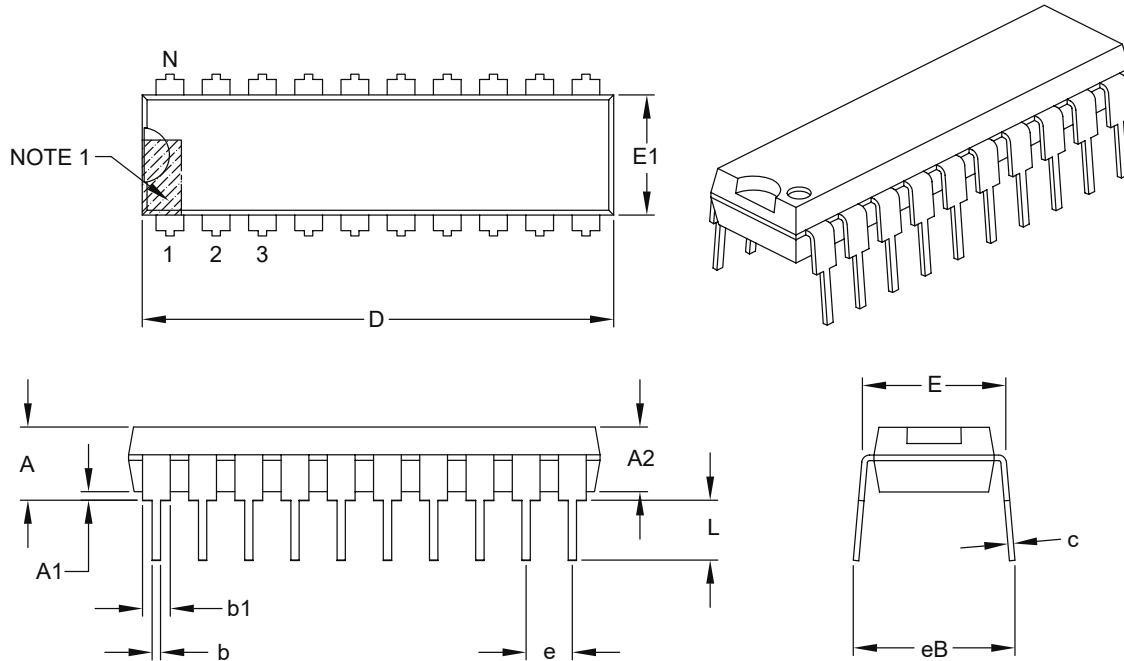
Microchip Technology Drawing C04-2257A

PIC16(L)F18426/46

Packaging Information

20-Lead Plastic Dual In-Line (P) – 300 mil Body [PDIP]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



| Dimension Limits | Units | INCHES | | |
|----------------------------|-------|----------|-------|-------|
| | | MIN | NOM | MAX |
| Number of Pins | N | 20 | | |
| Pitch | e | .100 BSC | | |
| Top to Seating Plane | A | – | – | .210 |
| Molded Package Thickness | A2 | .115 | .130 | .195 |
| Base to Seating Plane | A1 | .015 | – | – |
| Shoulder to Shoulder Width | E | .300 | .310 | .325 |
| Molded Package Width | E1 | .240 | .250 | .280 |
| Overall Length | D | .980 | 1.030 | 1.060 |
| Tip to Seating Plane | L | .115 | .130 | .150 |
| Lead Thickness | c | .008 | .010 | .015 |
| Upper Lead Width | b1 | .045 | .060 | .070 |
| Lower Lead Width | b | .014 | .018 | .022 |
| Overall Row Spacing § | eB | – | – | .430 |

Notes:

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- § Significant Characteristic.
- Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.
- Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

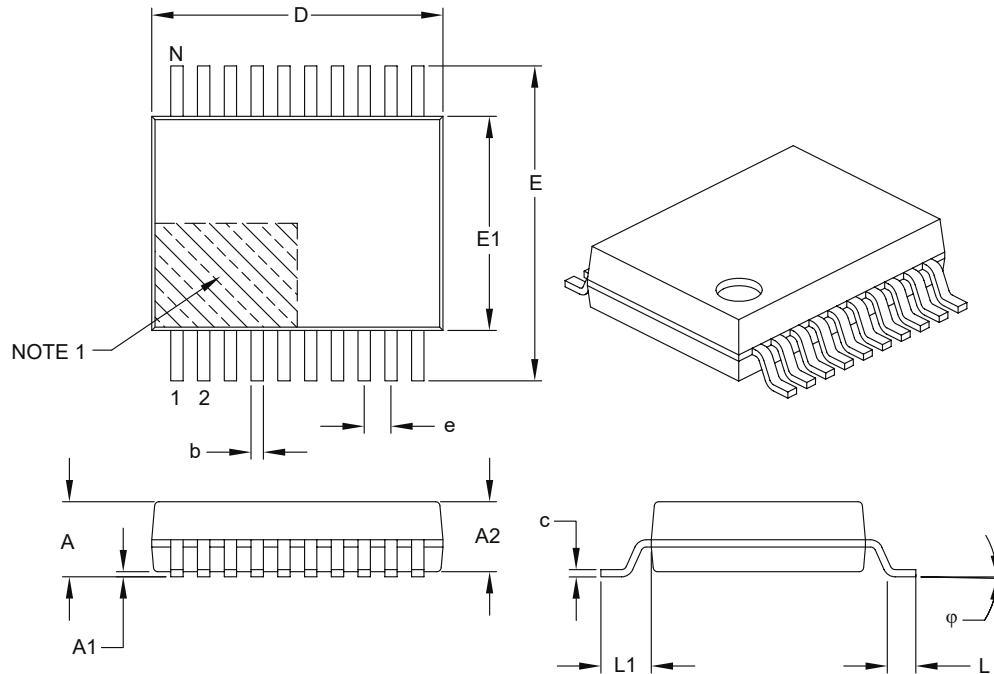
Microchip Technology Drawing C04-019B

PIC16(L)F18426/46

Packaging Information

20-Lead Plastic Shrink Small Outline (SS) – 5.30 mm Body [SSOP]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



| Dimension Limits | Units | MILLIMETERS | | |
|--------------------------|--------|-------------|------|------|
| | | MIN | NOM | MAX |
| Number of Pins | N | 20 | | |
| Pitch | e | 0.65 BSC | | |
| Overall Height | A | – | – | 2.00 |
| Molded Package Thickness | A2 | 1.65 | 1.75 | 1.85 |
| Standoff | A1 | 0.05 | – | – |
| Overall Width | E | 7.40 | 7.80 | 8.20 |
| Molded Package Width | E1 | 5.00 | 5.30 | 5.60 |
| Overall Length | D | 6.90 | 7.20 | 7.50 |
| Foot Length | L | 0.55 | 0.75 | 0.95 |
| Footprint | L1 | 1.25 REF | | |
| Lead Thickness | c | 0.09 | – | 0.25 |
| Foot Angle | ϕ | 0° | 4° | 8° |
| Lead Width | b | 0.22 | – | 0.38 |

Notes:

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.20 mm per side.
- Dimensioning and tolerancing per ASME Y14.5M.

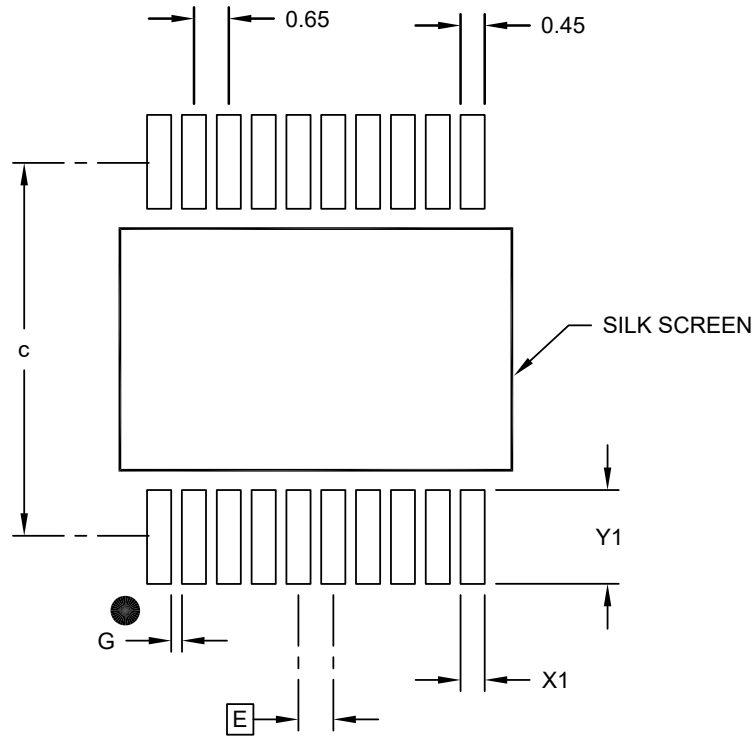
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-072B

20-Lead Plastic Shrink Small Outline (SS) - 5.30 mm Body [SSOP]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



RECOMMENDED LAND PATTERN

| | | Units | MILLIMETERS | | |
|--------------------------|----|-------|-------------|------|------|
| Dimension Limits | | | MIN | NOM | MAX |
| Contact Pitch | E | | 0.65 BSC | | |
| Contact Pad Spacing | C | | | 7.20 | |
| Contact Pad Width (X20) | X1 | | | | 0.45 |
| Contact Pad Length (X20) | Y1 | | | | 1.75 |
| Distance Between Pads | G | 0.20 | | | |

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M
 BSC: Basic Dimension. Theoretically exact value shown without tolerances.

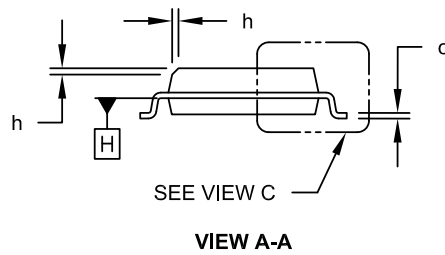
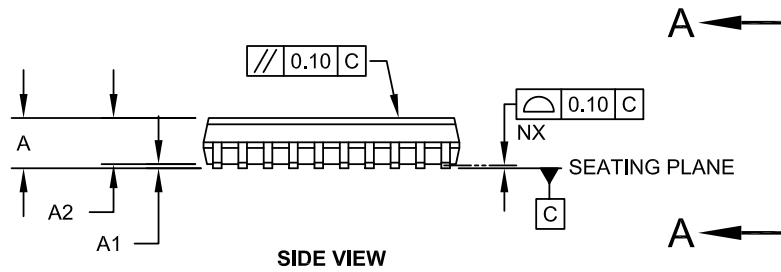
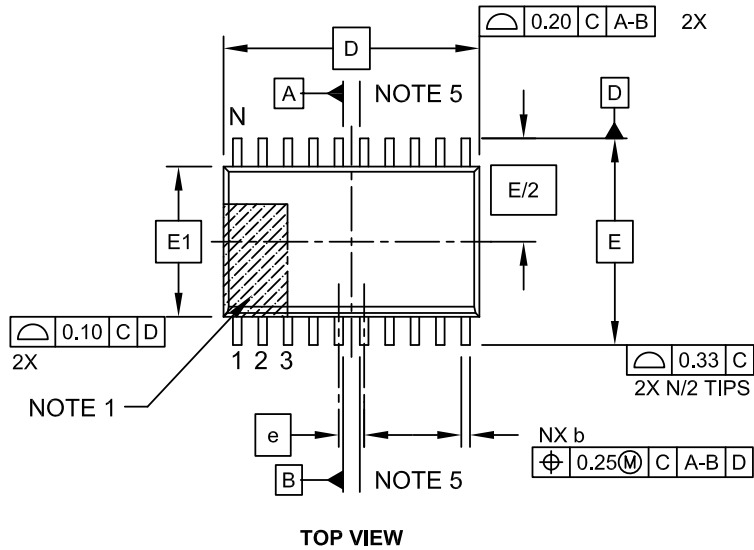
Microchip Technology Drawing No. C04-2072B

PIC16(L)F18426/46

Packaging Information

20-Lead Plastic Small Outline (SO) - Wide, 7.50 mm Body [SOIC]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



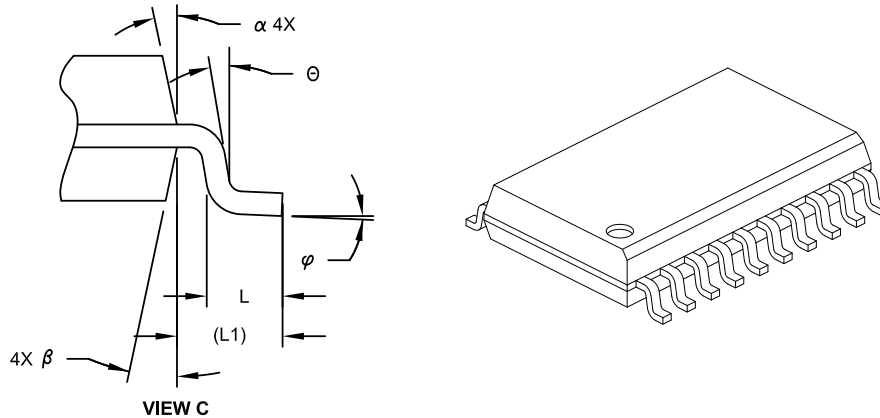
Microchip Technology Drawing C04-094C Sheet 1 of 2

PIC16(L)F18426/46

Packaging Information

20-Lead Plastic Small Outline (SO) - Wide, 7.50 mm Body [SOIC]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



| Dimension Limits | Units | MILLIMETERS | | |
|--------------------------|-----------|-------------|-----|------|
| | | MIN | NOM | MAX |
| Number of Pins | N | 20 | | |
| Pitch | e | 1.27 BSC | | |
| Overall Height | A | - | - | 2.65 |
| Molded Package Thickness | A2 | 2.05 | - | - |
| Standoff § | A1 | 0.10 | - | 0.30 |
| Overall Width | E | 10.30 BSC | | |
| Molded Package Width | E1 | 7.50 BSC | | |
| Overall Length | D | 12.80 BSC | | |
| Chamfer (Optional) | h | 0.25 | - | 0.75 |
| Foot Length | L | 0.40 | - | 1.27 |
| Footprint | L1 | 1.40 REF | | |
| Lead Angle | θ | 0° | - | - |
| Foot Angle | φ | 0° | - | 8° |
| Lead Thickness | c | 0.20 | - | 0.33 |
| Lead Width | b | 0.31 | - | 0.51 |
| Mold Draft Angle Top | α | 5° | - | 15° |
| Mold Draft Angle Bottom | β | 5° | - | 15° |

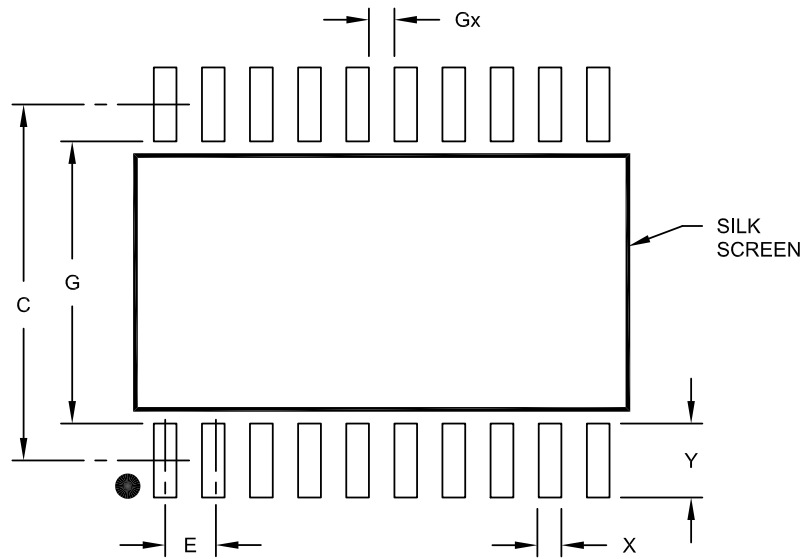
Notes:

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- § Significant Characteristic
- 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.
- 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.
- Datums A & B to be determined at Datum H.

Microchip Technology Drawing No. C04-094C Sheet 2 of 2

20-Lead Plastic Small Outline (SO) - Wide, 7.50 mm Body [SOIC]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



RECOMMENDED LAND PATTERN

| Dimension Limits | Units | MILLIMETERS | | |
|--------------------------|-------|-------------|------|------|
| | | MIN | NOM | MAX |
| Contact Pitch | E | 1.27 BSC | | |
| Contact Pad Spacing | C | | 9.40 | |
| Contact Pad Width (X20) | X | | | 0.60 |
| Contact Pad Length (X20) | Y | | | 1.95 |
| Distance Between Pads | Gx | 0.67 | | |
| Distance Between Pads | G | 7.45 | | |

Notes:

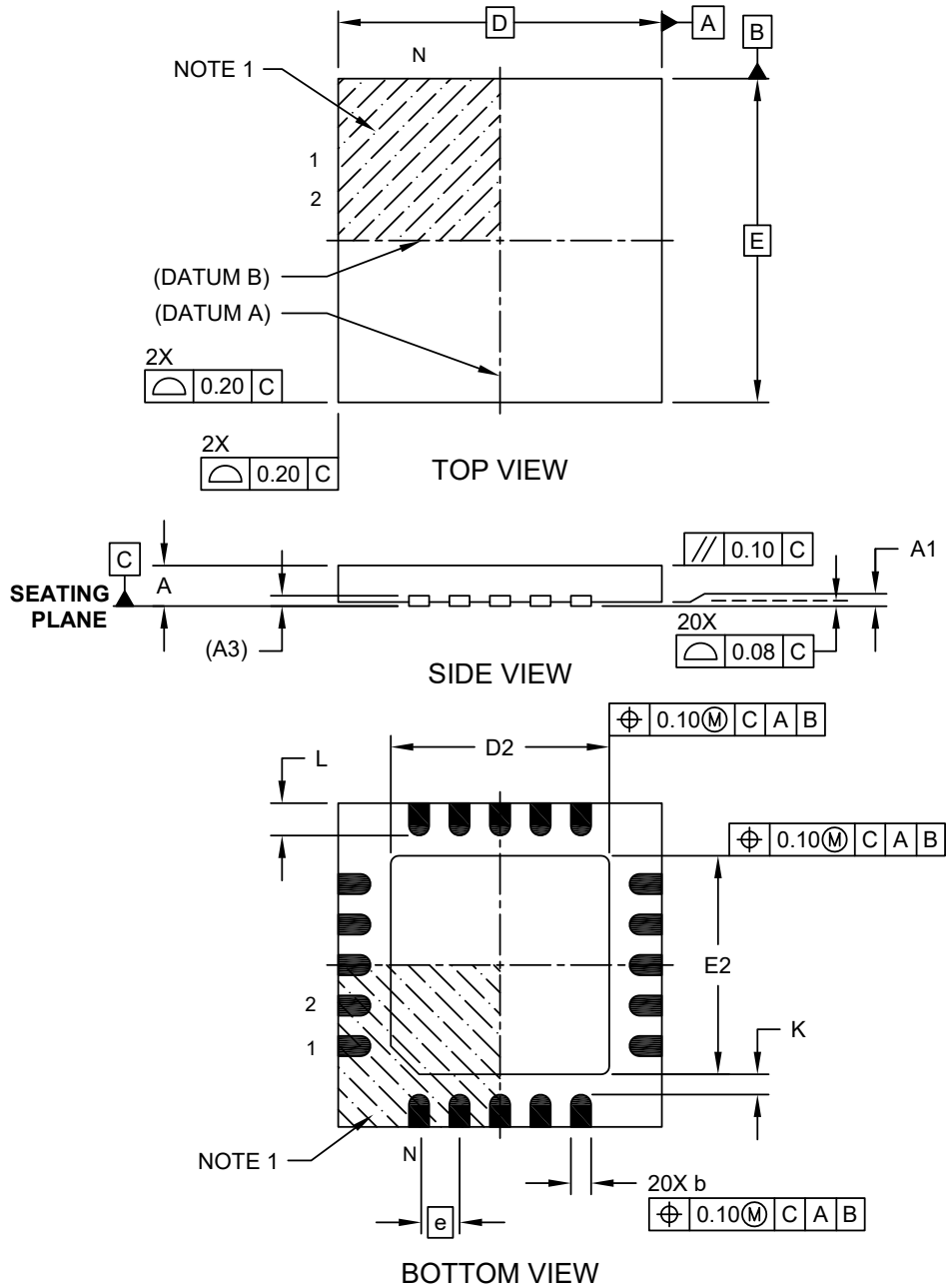
1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2094A

20-Lead Ultra Thin Plastic Quad Flat, No Lead Package (GZ) - 4x4x0.5 mm Body [UQFN]

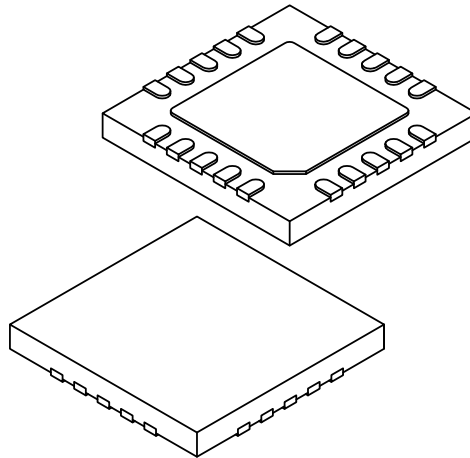
Note: For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



Microchip Technology Drawing C04-255A Sheet 1 of 2

20-Lead Ultra Thin Plastic Quad Flat, No Lead Package (GZ) - 4x4x0.5 mm Body [UQFN]

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 Terminals | N | | 20 | | |
| Pitch | e | | 0.50 BSC | | |
| Overall Height | A | 0.45 | 0.50 | 0.55 | |
| Standoff | A1 | 0.00 | 0.02 | 0.05 | |
| Terminal Thickness | A3 | 0.127 REF | | | |
| Overall Width | E | 4.00 BSC | | | |
| Exposed Pad Width | E2 | 2.60 | 2.70 | 2.80 | |
| Overall Length | D | 4.00 BSC | | | |
| Exposed Pad Length | D2 | 2.60 | 2.70 | 2.80 | |
| Terminal Width | b | 0.20 | 0.25 | 0.30 | |
| Terminal Length | L | 0.30 | 0.40 | 0.50 | |
| Terminal-to-Exposed-Pad | K | 0.20 | - | - | |

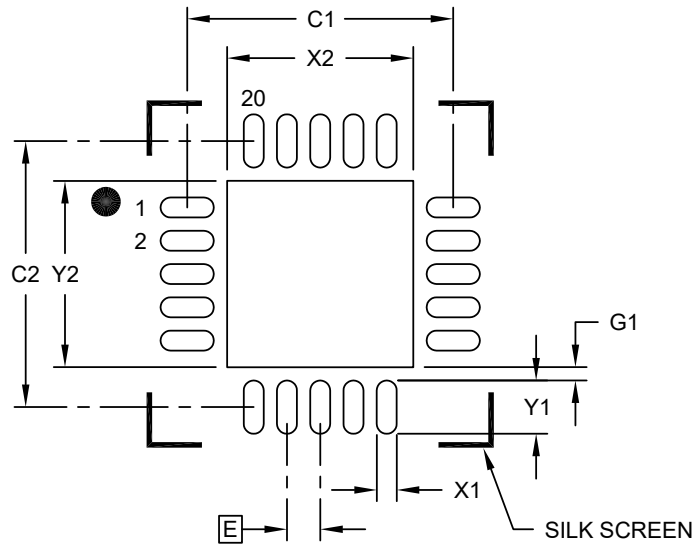
Notes:

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 C04-255A Sheet 2 of 2

20-Lead Ultra Thin Plastic Quad Flat, No Lead Package (GZ) - 4x4x0.5 mm Body [UQFN]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



RECOMMENDED LAND PATTERN

| Dimension Limits | Units | MILLIMETERS | | |
|---------------------------------|-------|-------------|------|------|
| | | MIN | NOM | MAX |
| Contact Pitch | E | 0.50 BSC | | |
| Optional Center Pad Width | X2 | | | 2.80 |
| Optional Center Pad Length | Y2 | | | 2.80 |
| Contact Pad Spacing | C1 | | 4.00 | |
| Contact Pad Spacing | C2 | | 4.00 | |
| Contact Pad Width (X20) | X1 | | | 0.30 |
| Contact Pad Length (X20) | Y1 | | | 0.80 |
| Contact Pad to Center Pad (X20) | G1 | 0.20 | | |

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M
 BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing C04-2255A

43. Revision History

| Doc Rev. | Date | Comments |
|----------|---------|--|
| B | 11/2019 | Adding Electrical Specifications and DC and AC Characteristics Graphs and Tables. Other minor corrections. |
| A | 02/2018 | Initial release of this document. |

The Microchip Website

Microchip provides online support via our website at <http://www.microchip.com/>. This website is used to make files and information easily available to customers. Some of the content available includes:

- **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 (FAQs), technical support requests, online discussion groups, Microchip design partner 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

Product Change Notification Service

Microchip's product change notification service helps keep customers current on Microchip products. Subscribers will receive email notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, go to <http://www.microchip.com/pcn> and follow the registration instructions.

Customer Support

Users of Microchip products can receive assistance through several channels:

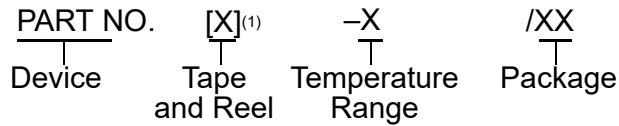
- Distributor or Representative
- Local Sales Office
- Embedded Solutions Engineer (ESE)
- Technical Support

Customers should contact their distributor, representative or ESE for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in this document.

Technical support is available through the website at: <http://www.microchip.com/support>

Product Identification System

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.



| | | |
|---------------------|---|-------------------------------|
| Device: | PIC16F18426, PIC16LF18426, PIC16F18446, PIC16LF18446 | |
| Tape & Reel Option: | Blank | = Tube |
| | T | = Tape & Reel |
| Temperature Range: | I | = -40°C to +85°C (Industrial) |
| | E | = -40°C to +125°C (Extended) |
| Package: | JQ | = 16-lead UQFN 4x4x0.5mm |
| | P | = 14-lead, 20-lead PDIP |
| | SL | = 14-lead SOIC |
| | SO | = 20-lead SOIC |
| | SS | = 20-lead SSOP |
| | ST | = 14-lead TSSOP |
| | GZ | = 20-lead UQFN 4x4x0.5mm |
| Pattern: | QTP, SQTP, Code or Special Requirements (blank otherwise) | |

Examples:

- PIC16F18426- E/P Extended temperature PDIP package

Note:

1. Tape and Reel identifier only appears in the catalog part number description. This identifier is used for ordering purposes and is not printed on the device package. Check with your Microchip Sales Office for package availability with the Tape and Reel option.
2. Small form-factor packaging options may be available. Please check <http://www.microchip.com/packaging> for small-form factor package availability, or contact your local Sales Office.

Microchip Devices Code Protection Feature

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.

Legal Notice

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

Trademarks

The Microchip name and logo, the Microchip logo, Adaptec, AnyRate, AVR, AVR logo, AVR Freaks, BesTime, BitCloud, chipKIT, chipKIT logo, CryptoMemory, CryptoRF, dsPIC, FlashFlex, flexPWR, HELDO, IGLOO, JukeBlox, KeeLoq, Klear, LANCheck, LinkMD, maXStylus, maXTouch, MediaLB, megaAVR, Microsemi, Microsemi logo, MOST, MOST logo, MPLAB, OptoLyzer, PackeTime, PIC, picoPower, PICSTART, PIC32 logo, PolarFire, Prochip Designer, QTouch, SAM-BA, SenGenuity, SpyNIC, SST, SST Logo, SuperFlash, Symmetricom, SyncServer, Tachyon, TempTrackr, TimeSource, tinyAVR, UNI/O, Vectron, and XMEGA are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

APT, ClockWorks, The Embedded Control Solutions Company, EtherSynch, FlashTec, Hyper Speed Control, HyperLight Load, IntelliMOS, Libero, motorBench, mTouch, Powermite 3, Precision Edge, ProASIC, ProASIC Plus, ProASIC Plus logo, Quiet-Wire, SmartFusion, SyncWorld, Temux, TimeCesium, TimeHub, TimePictra, TimeProvider, Vite, WinPath, and ZL are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Adjacent Key Suppression, AKS, Analog-for-the-Digital Age, Any Capacitor, AnyIn, AnyOut, BlueSky, BodyCom, CodeGuard, CryptoAuthentication, CryptoAutomotive, CryptoCompanion, CryptoController, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, EtherGREEN, In-Circuit Serial Programming, ICSP, INICnet, Inter-Chip Connectivity, JitterBlocker, KlearNet, KlearNet logo, memBrain, Mindi, MiWi, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, PowerSmart, PureSilicon, QMatrix, REAL ICE, Ripple Blocker, SAM-ICE, Serial Quad I/O, SMART-I.S., SQI, SuperSwitcher, SuperSwitcher II, Total Endurance, TSHARC, USBCheck, VariSense, ViewSpan, WiperLock, Wireless DNA, and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

The Adaptec logo, Frequency on Demand, Silicon Storage Technology, and Symmcom are registered trademarks of Microchip Technology Inc. in other countries.

GestIC is a registered trademark of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

© 2019, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

ISBN: 978-1-5224-5234-8

Quality Management System

For information regarding Microchip's Quality Management Systems, please visit <http://www.microchip.com/quality>.

Worldwide Sales and Service

| AMERICAS | ASIA/PACIFIC | ASIA/PACIFIC | EUROPE |
|---|---|--|---|
| <p>Corporate Office 2355 West Chandler Blvd. Chandler, AZ 85224-6199 Tel: 480-792-7200 Tel: 480-792-7277 Technical Support: http://www.microchip.com/support Web Address: http://www.microchip.com</p> <p>Atlanta Duluth, GA Tel: 678-957-9614 Fax: 678-957-1455</p> <p>Austin, TX Tel: 512-257-3370</p> <p>Boston Westborough, MA Tel: 774-760-0087 Fax: 774-760-0088</p> <p>Chicago Itasca, IL Tel: 630-285-0071 Fax: 630-285-0075</p> <p>Dallas Addison, TX Tel: 972-818-7423 Fax: 972-818-2924</p> <p>Detroit Novi, MI Tel: 248-848-4000</p> <p>Houston, TX Tel: 281-894-5983</p> <p>Indianapolis Noblesville, IN Tel: 317-773-8323 Fax: 317-773-5453 Tel: 317-536-2380</p> <p>Los Angeles Mission Viejo, CA Tel: 949-462-9523 Fax: 949-462-9608 Tel: 951-273-7800</p> <p>Raleigh, NC Tel: 919-844-7510</p> <p>New York, NY Tel: 631-435-6000</p> <p>San Jose, CA Tel: 408-735-9110 Tel: 408-436-4270</p> <p>Canada - Toronto Tel: 905-695-1980 Fax: 905-695-2078</p> | <p>Australia - Sydney Tel: 61-2-9868-6733</p> <p>China - Beijing Tel: 86-10-8569-7000</p> <p>China - Chengdu Tel: 86-28-8665-5511</p> <p>China - Chongqing Tel: 86-23-8980-9588</p> <p>China - Dongguan Tel: 86-769-8702-9880</p> <p>China - Guangzhou Tel: 86-20-8755-8029</p> <p>China - Hangzhou Tel: 86-571-8792-8115</p> <p>China - Hong Kong SAR Tel: 852-2943-5100</p> <p>China - Nanjing Tel: 86-25-8473-2460</p> <p>China - Qingdao Tel: 86-532-8502-7355</p> <p>China - Shanghai Tel: 86-21-3326-8000</p> <p>China - Shenyang Tel: 86-24-2334-2829</p> <p>China - Shenzhen Tel: 86-755-8864-2200</p> <p>China - Suzhou Tel: 86-186-6233-1526</p> <p>China - Wuhan Tel: 86-27-5980-5300</p> <p>China - Xian Tel: 86-29-8833-7252</p> <p>China - Xiamen Tel: 86-592-2388138</p> <p>China - Zhuhai Tel: 86-756-3210040</p> | <p>India - Bangalore Tel: 91-80-3090-4444</p> <p>India - New Delhi Tel: 91-11-4160-8631</p> <p>India - Pune Tel: 91-20-4121-0141</p> <p>Japan - Osaka Tel: 81-6-6152-7160</p> <p>Japan - Tokyo Tel: 81-3-6880-3770</p> <p>Korea - Daegu Tel: 82-53-744-4301</p> <p>Korea - Seoul Tel: 82-2-554-7200</p> <p>Malaysia - Kuala Lumpur Tel: 60-3-7651-7906</p> <p>Malaysia - Penang Tel: 60-4-227-8870</p> <p>Philippines - Manila Tel: 63-2-634-9065</p> <p>Singapore Tel: 65-6334-8870</p> <p>Taiwan - Hsin Chu Tel: 886-3-577-8366</p> <p>Taiwan - Kaohsiung Tel: 886-7-213-7830</p> <p>Taiwan - Taipei Tel: 886-2-2508-8600</p> <p>Thailand - Bangkok Tel: 66-2-694-1351</p> <p>Vietnam - Ho Chi Minh Tel: 84-28-5448-2100</p> | <p>Austria - Wels Tel: 43-7242-2244-39 Fax: 43-7242-2244-393</p> <p>Denmark - Copenhagen Tel: 45-4450-2828 Fax: 45-4485-2829</p> <p>Finland - Espoo Tel: 358-9-4520-820</p> <p>France - Paris Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79</p> <p>Germany - Garching Tel: 49-8931-9700</p> <p>Germany - Haan Tel: 49-2129-3766400</p> <p>Germany - Heilbronn Tel: 49-7131-72400</p> <p>Germany - Karlsruhe Tel: 49-721-625370</p> <p>Germany - Munich Tel: 49-89-627-144-0 Fax: 49-89-627-144-44</p> <p>Germany - Rosenheim Tel: 49-8031-354-560</p> <p>Israel - Ra'anana Tel: 972-9-744-7705</p> <p>Italy - Milan Tel: 39-0331-742611 Fax: 39-0331-466781</p> <p>Italy - Padova Tel: 39-049-7625286</p> <p>Netherlands - Drunen Tel: 31-416-690399 Fax: 31-416-690340</p> <p>Norway - Trondheim Tel: 47-72884388</p> <p>Poland - Warsaw Tel: 48-22-3325737</p> <p>Romania - Bucharest Tel: 40-21-407-87-50</p> <p>Spain - Madrid Tel: 34-91-708-08-90 Fax: 34-91-708-08-91</p> <p>Sweden - Gothenberg Tel: 46-31-704-60-40</p> <p>Sweden - Stockholm Tel: 46-8-5090-4654</p> <p>UK - Wokingham Tel: 44-118-921-5800 Fax: 44-118-921-5820</p> |

X-ON Electronics

Largest Supplier of Electrical and Electronic Components

Click to view similar products for [8-bit Microcontrollers - MCU category](#):

Click to view products by [Microchip manufacturer](#):

Other Similar products are found below :

[CY8C20524-12PVXIT](#) [MB95F013KPMC-G-SNE2](#) [MB95F263KPF-G-SNE2](#) [MB95F264KPFT-G-SNE2](#) [MB95F398KPMC-G-SNE2](#)
[MB95F478KPMC2-G-SNE2](#) [MB95F564KPF-G-SNE2](#) [MB95F636KWQN-G-SNE1](#) [MB95F696KPMC-G-SNE2](#) [MB95F698KPMC2-G-SNE2](#)
[MB95F698KPMC-G-SNE2](#) [MB95F818KPMC1-G-SNE2](#) [901015X](#) [CY8C3MFIDOCK-125](#) [403708R](#) [MB95F354EPF-G-SNE2](#)
[MB95F564KWQN-G-SNE1](#) [MB95F636KP-G-SH-SNE2](#) [MB95F694KPMC-G-SNE2](#) [MB95F778JPMC1-G-SNE2](#) [MB95F818KPMC-G-SNE2](#)
[LC87F0G08AUJA-AH](#) [CP8361BT](#) [CG8421AF](#) [MB95F202KPF-G-SNE2](#) [DF36014FPV](#) [5962-8768407MUA](#) [MB95F318EPMC-G-SNE2](#)
[MB94F601APMC1-GSE1](#) [MB95F656EPF-G-SNE2](#) [LC78615E-01US-H](#) [LC87F5WC8AVU-QIP-H](#) [MB95F108AJSPMC-G-JNE1](#) [73S1210F-](#)
[68M/F/PJ](#) [MB89F538-101PMC-GE1](#) [LC87F7DC8AVU-QIP-H](#) [MB95F876KPMC-G-SNE2](#) [MB88386PMC-GS-BNDE1](#) [LC87FBK08AU-](#)
[SSOP-H](#) [LC87F2C64AU-QFP-H](#) [MB95F636KNWQN-G-118-SNE1](#) [MB95F136NBSTPFV-GS-N2E1](#) [LC87F5NC8AVU-QIP-E](#)
[LC87F76C8AU-TQFP-E](#) [LC87F2G08AU-SSOP-E](#) [CP8085AT](#) [MB95F564KPF-G-UNE2](#) [MC9S08PA4VWJ](#) [MC9S08QG8CDTE](#)
[MC9S08SH4CWJR](#)