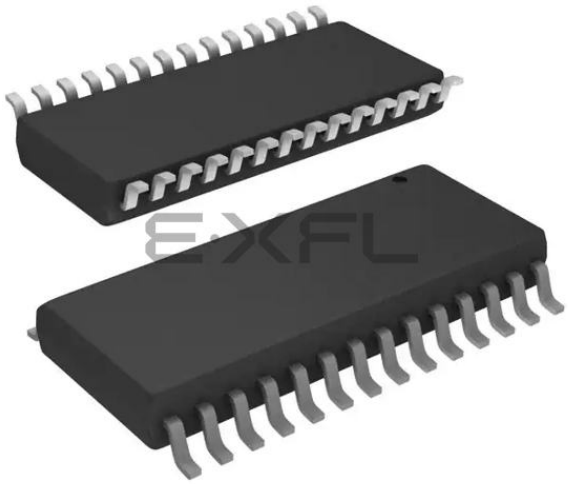


**Welcome to E-XFL.COM**

## What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

## Applications of "Embedded - Microcontrollers"

| Details | |
|---|---|
| Product Status | Active |
| Core Processor | PIC |
| Core Size | 8-Bit |
| Speed | 32MHz |
| Connectivity | I²C, LINbus, SPI, UART/USART |
| Peripherals | Brown-out Detect/Reset, POR, PWM, WDT |
| Number of I/O | 25 |
| Program Memory Size | 28KB (16K x 14) |
| Program Memory Type | FLASH |
| EEPROM Size | 128 x 8 |
| RAM Size | 2K x 8 |
| Voltage - Supply (Vcc/Vdd) | 1.8V ~ 3.6V |
| Data Converters | A/D 17x10b; D/A 3x5b, 3x10b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 28-SOIC (0.295", 7.50mm Width) |
| Supplier Device Package | 28-SOIC |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/pic16lf1778-i-so |

## 1.1 Register and Bit naming conventions

### 1.1.1 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.

### 1.1.2 BIT NAMES

There are two variants for bit names:

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

#### 1.1.2.1 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 *RegisterName*bits.*ShortName*. For example, the enable bit, EN, in the COG1CON0 register can be set in C programs with the instruction `COG1CON0bits.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.

#### 1.1.2.2 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.

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.

#### 1.1.2.3 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 the Push-Pull mode:

**EXAMPLE 1-1:**

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

**EXAMPLE 1-2:**

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

### 1.1.3 REGISTER AND BIT NAMING EXCEPTIONS

#### 1.1.3.1 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.

#### 1.1.3.2 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

**TABLE 3-10: PIC16(L)F1779 MEMORY MAP, BANK 16-23**

| BANK 16 | | BANK 17 | | BANK 18 | | BANK 19 | | BANK 20 | | BANK 21 | | BANK 22 | | BANK 23 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 800h | Core Registers (Table 3-2) | 880h | Core Registers (Table 3-2) | 900h | Core Registers (Table 3-2) | 980h | Core Registers (Table 3-2) | A00h | Core Registers (Table 3-2) | A80h | Core Registers (Table 3-2) | B00h | Core Registers (Table 3-2) | B80h | Core Registers (Table 3-2) |
| 80Bh | | 88Bh | | 90Bh | | 98Bh | | A0Bh | | A8Bh | | B0Bh | | B8Bh | |
| 80Ch | — | 88Ch | — | 90Ch | CM4CON0 | 98Ch | Unimplemented Read as '0' | A0Ch | Unimplemented Read as '0' | A8Ch | Unimplemented Read as '0' | B0Ch | Unimplemented Read as '0' | B8Ch | Unimplemented Read as '0' |
| 80Dh | COG3PHR | | COG4PHR | 90Dh | CM4CON1 | | | | | | | | | | |
| 80Eh | COG3PHF | | COG4PHF | 90Eh | CM4NSEL | | | | | | | | | | |
| 80Fh | COG3BLKR | | COG4BLKR | 90Fh | CM4PSEL | | | | | | | | | | |
| 810h | COG3BLKF | | COG4BLKF | 910h | CM5CON0 | | | | | | | | | | |
| 811h | COG3DBR | | COG4DBR | 911h | CM5CON1 | | | | | | | | | | |
| 812h | COG3DBF | | COG4DBF | 912h | CM5NSEL | | | | | | | | | | |
| 813h | COG3CON0 | | COG4CON0 | 913h | CM5PSEL | | | | | | | | | | |
| 814h | COG3CON1 | | COG4CON1 | 914h | CM6CON0 | | | | | | | | | | |
| 815h | COG3RIS0 | | COG4RIS0 | 915h | CM6CON1 | | | | | | | | | | |
| 816h | COG3RIS1 | | COG4RIS1 | 916h | CM6NSEL | | | | | | | | | | |
| 817h | COG3RSIM0 | | COG4RSIM0 | 917h | CM6PSEL | | | | | | | | | | |
| 818h | COG3RSIM1 | | COG4RSIM1 | 918h | CM7CON0 | | | | | | | | | | |
| 819h | COG3FIS0 | | COG4FIS0 | | CM7CON1 | | | | | | | | | | |
| 81Ah | COG3FIS1 | | COG4FIS1 | | CM7NSEL | | | | | | | | | | |
| 81Bh | COG3FSIM0 | | COG4FSIM0 | | CM7PSEL | | | | | | | | | | |
| 81Ch | COG3FSIM1 | | COG4FSIM1 | | CM8CON0 | | | | | | | | | | |
| 81Dh | COG3ASD0 | | COG4ASD0 | | CM8CON1 | | | | | | | | | | |
| 81Eh | COG3ASD1 | | COG4ASD1 | | CM8NSEL | | | | | | | | | | |
| 81Fh | COG3STR | 89Fh | COG4STR | 91Fh | CM8PSEL | 99Fh | | A1Fh | | A9Fh | | B1Fh | | B9Fh | |
| 820h | General Purpose Register 48 Bytes | 8A0h | General Purpose Register 48 Bytes | 920h | General Purpose Register 48 Bytes | 9A0h | General Purpose Register 48 Bytes | A20H | General Purpose Register 48 Bytes | AA0h | General Purpose Register 48 Bytes | B20h | General Purpose Register 48 Bytes | BA0h | General Purpose Register 48 Bytes |
| 86Fh | | 8EFh | | 96Fh | | 9EFh | | A6Fh | | AEFh | | B6Fh | | BEFh | |
| 870h | Accesses 70h – 7Fh | 8F0h | Accesses 70h – 7Fh | 970h | Accesses 70h – 7Fh | 9F0h | Accesses 70h – 7Fh | A70h | Accesses 70h – 7Fh | AF0h | Accesses 70h – 7Fh | B70h | Accesses 70h – 7Fh | BF0h | Accesses 70h – 7Fh |
| 87Fh | | 8FFh | | 97Fh | | 9FFh | | A7Fh | | AFFh | | B7Fh | | BFFh | |

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

**TABLE 3-18: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)**

| Addr | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR, BOR | Value on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Bank 16** | | | | | | | | | | | |
| 80Ch | — | Unimplemented | | | | | | | | — | — |
| 80Dh | COG3PHR | — | — | COG Rising Edge Phase Delay Count Register | | | | | | --00 0000 | --00 0000 |
| 80Eh | COG3PHF | — | — | COG Falling Edge Phase Delay Count Register | | | | | | --00 0000 | --00 0000 |
| 80Fh | COG3BLKR | — | — | COG Rising Edge Blanking Count Register | | | | | | --00 0000 | --00 0000 |
| 810h | COG3BLKF | — | — | COG Falling Edge Blanking Count Register | | | | | | --00 0000 | --00 0000 |
| 811h | COG3DBR | — | — | COG Rising Edge Dead-band Count Register | | | | | | --00 0000 | --00 0000 |
| 812h | COG3DBF | — | — | COG Falling Edge Dead-band Count Register | | | | | | --00 0000 | --00 0000 |
| 813h | COG3CON0 | EN | LD | — | CS<1:0> | | MD<2:0> | | | 00-0 0000 | 00-0 0000 |
| 814h | COG3CON1 | RDBS | FDBS | — | — | POLD | POLC | POLB | POLA | 00-- 0000 | 00-- 0000 |
| 815h | COG3RIS0 | RIS7 | RIS6 | RIS5 | RIS4 | RIS3 | RIS2 | RIS1 | RIS0 | 0000 0000 | 0000 0000 |
| 816h | COG3RIS1 | RIS15 | RIS14 | RIS13 | RIS12 | RIS11 | RIS10 | RIS9 | RIS8 | 0000 0000 | 0000 0000 |
| 817h | COG3RSIM0 | RSIM7 | RSIM6 | RSIM5 | RSIM4 | RSIM3 | RSIM2 | RSIM1 | RSIM0 | 0000 0000 | 0000 0000 |
| 818h | COG3RSIM1 | RSIM15 | RSIM14 | RSIM13 | RSIM12 | RSIM11 | RSIM10 | RSIM9 | RSIM8 | 0000 0000 | 0000 0000 |
| 819h | COG3FIS0 | FIS7 | FIS6 | FIS5 | FIS4 | FIS3 | FIS2 | FIS1 | FIS0 | 0000 0000 | 0000 0000 |
| 81Ah | COG3FIS1 | FIS15 | FIS14 | FIS13 | FIS12 | FIS11 | FIS10 | FIS9 | FIS8 | 0000 0000 | 0000 0000 |
| 81Bh | COG3FSIM0 | FSIM7 | FSIM6 | FSIM5 | FSIM4 | FSIM3 | FSIM2 | FSIM1 | FSIM0 | 0000 0000 | 0000 0000 |
| 81Ch | COG3FSIM1 | FSIM15 | FSIM14 | FSIM13 | FSIM12 | FSIM11 | FSIM10 | FSIM9 | FSIM8 | 0000 0000 | 0000 0000 |
| 81Dh | COG3ASD0 | ASE | ARSEN | ASDBD<1:0> | | ASDAC<1:0> | | — | — | 0001 01-- | 0001 01-- |
| 81Eh | COG3ASD1 | AS7E | AS6E | AS5E | AS4E | AS3E | AS2E | AS1E | AS0E | 0000 0000 | 0000 0000 |
| 81Fh | COG3STR | SDATD | SDATC | SDATB | SDATA | STRD | STRC | STRB | STRA | 0000 0000 | 0000 0000 |

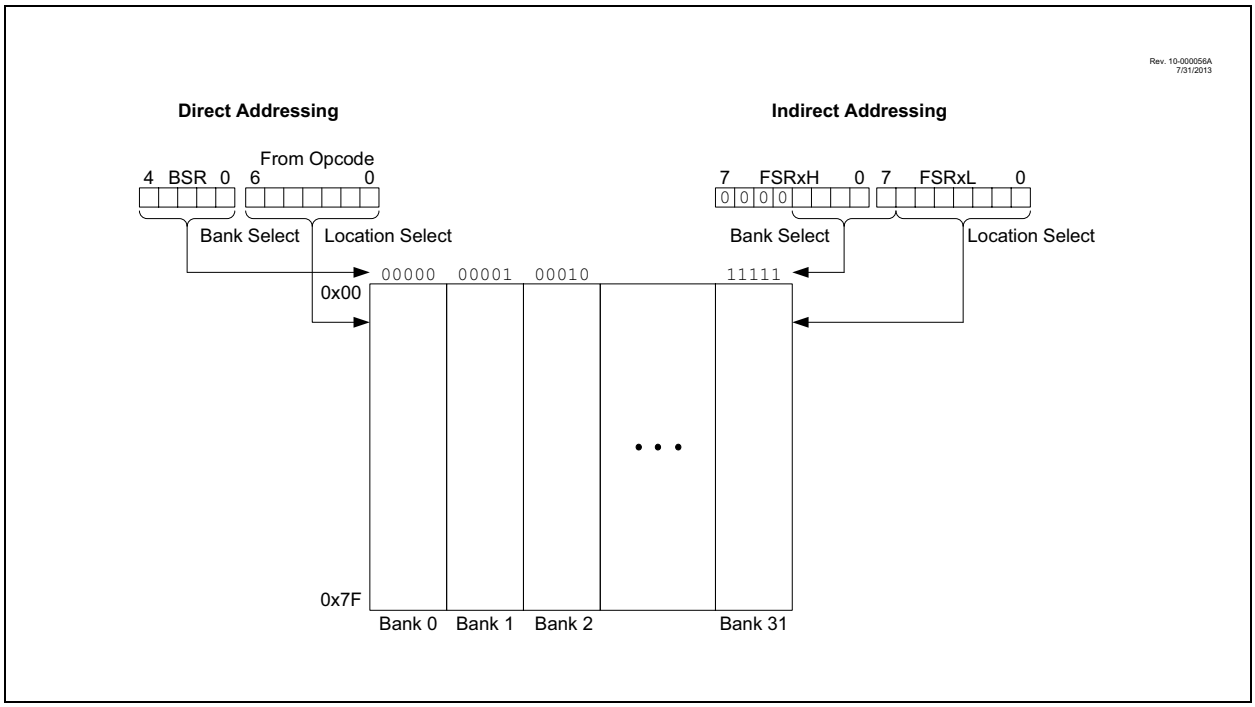
**Legend:** x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, read as '0', r = reserved.
Shaded locations are unimplemented, read as '0'.

**Note 1:** Unimplemented, read as '1'.
**2:** Unimplemented on PIC16LF1777/8/9.
**3:** Unimplemented on PIC16(L)F1778.

### 3.7.1 TRADITIONAL DATA MEMORY

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

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



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

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

| R/P-1 | U-1 | U-1 | U-1 | U-1 | R/P-1 | R/P-1 | R/P-1 |
|---|---|---|---|---|---|---|---|
| $\overline{\text{ZCD}}$ | — | — | — | — | PPS1WAY | WRT<1:0> | |
| bit 7 | | | | | | | bit 0 |

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

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

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

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

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

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

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

bit 7 **$\overline{\text{ZCD}}$:** ZCD Enable bit
- 1 = OFF ZCD disabled. ZCD can be enabled by setting the ZCDSEN bit of ZCDCON
- 0 = ON ZCD always enabled

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

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

bit 1-0 **WRT<1:0>:** Flash Memory Self-Write Protection bits

4 kW Flash memory (PIC16(L)F1764/8)
- 11 = OFF Write protection off
- 10 = BOOT 0000h to 01FFh write protected, 0200h to 0FFFh may be modified by PMCON control
- 01 = HALF 0000h to 07FFh write protected, 0800h to 0FFFh may be modified by PMCON control
- 00 = ALL 0000h to 0FFFh write protected, no addresses may be modified by PMCON control

8 kW Flash memory (PIC16(L)F1765/9)
- 11 = OFF Write protection off
- 10 = BOOT 0000h to 01FFh write protected, 0200h to 1FFFh may be modified by PMCON control
- 01 = HALF 0000h to 0FFFh write protected, 1000h to 1FFFh may be modified by PMCON control
- 00 = ALL 0000h to 1FFFh write protected, no addresses may be modified by PMCON control

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

## 5.2 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 are: oscillator modules (ECH, ECM, ECL mode), quartz crystal resonators or ceramic resonators (LP, XT and HS modes) and Resistor-Capacitor (EXTRC) mode circuits.

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

The system clock can be selected between external or internal clock sources via the System Clock Select (SCS) bits in the OSCCON register. See **Section 5.3 "Clock Switching"** for additional information.

### 5.2.1 EXTERNAL CLOCK SOURCES

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

- Program the FOSC<2:0> bits in the Configuration Words to select an external clock source that will be used as the default system clock upon a device Reset.
- Write the SCS<1:0> bits in the OSCCON register to switch the system clock source to:
  - Secondary oscillator during run-time, or
  - An external clock source determined by the value of the FOSC bits.

See **Section 5.3 "Clock Switching"** for more information.

#### 5.2.1.1 EC Mode

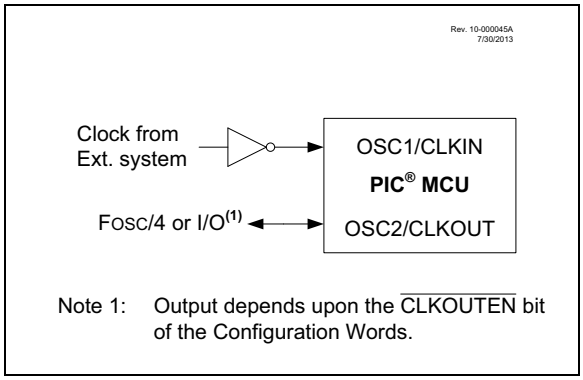
The External Clock (EC) mode allows an externally generated logic level signal to be the system clock source. When operating in this mode, an external clock source is connected to the OSC1 input. OSC2/CLKOUT is available for general purpose I/O or CLKOUT. Figure 5-2 shows the pin connections for EC mode.

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

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

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 5-2: EXTERNAL CLOCK (EC) MODE OPERATION**




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

#### 5.2.1.2 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 5-3). 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).

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

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

Figure 5-3 and Figure 5-4 show typical circuits for quartz crystal and ceramic resonators, respectively.

## 5.3 Clock Switching

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

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

### 5.3.1 SYSTEM CLOCK SELECT (SCS) BITS

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

- When the SCS bits of the OSCCON register = 00, the system clock source is determined by the value of the FOSC<2:0> bits in the Configuration Words.
- When the SCS bits of the OSCCON register = 01, the system clock source is the secondary oscillator.
- When the SCS bits of the OSCCON register = 1x, the system clock source is chosen by the internal oscillator frequency selected by the IRCF<3:0> bits of the OSCCON register. After a Reset, the SCS bits of the OSCCON register are always cleared.

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

When switching between clock sources, a delay is required to allow the new clock to stabilize. These oscillator delays are shown in Table 5-1.

### 5.3.2 OSCILLATOR START-UP TIMER STATUS (OSTS) BIT

The Oscillator Start-up Timer Status (OSTS) bit of the OSCSTAT register indicates whether the system clock is running from the external clock source, as defined by the FOSC<2:0> bits in the Configuration Words, or from the internal clock source. In particular, OSTS indicates that the Oscillator Start-up Timer (OST) has timed out for LP, XT or HS modes. The OST does not reflect the status of the secondary oscillator.

### 5.3.3 SECONDARY OSCILLATOR

The secondary oscillator is a separate crystal oscillator associated with the Timer1 peripheral. It is optimized for timekeeping operations with a 32.768 kHz crystal connected between the SOSCO and SOSCI device pins.

The secondary oscillator is enabled using the OSCEN control bit in the T1CON register. See **Section 22.0 "Timer1/3/5 Module with Gate Control"** for more information about the Timer1 peripheral.

### 5.3.4 SECONDARY OSCILLATOR READY (SOSCR) BIT

The user must ensure that the secondary oscillator is ready to be used before it is selected as a system clock source. The Secondary Oscillator Ready (SOSCR) bit of the OSCSTAT register indicates whether the secondary oscillator is ready to be used. After the SOSCR bit is set, the SCS bits can be configured to select the secondary oscillator.

### 5.3.5 CLOCK SWITCH BEFORE SLEEP

When a clock switch from an old clock to a new clock is requested just prior to entering Sleep mode, it is necessary to confirm that the switch is complete before the sleep instruction is executed. Failure to do so may result in an incomplete switch and consequential loss of the system clock altogether. Clock switching is confirmed by monitoring the clock status bits in the OSCSTAT register. Switch confirmation can be accomplished by sensing that the ready bit for the new clock is set or the ready bit for the old clock is cleared. For example, when switching between the internal oscillator with the PLL and the internal oscillator without the PLL, monitor the PLLR bit. When PLLR is set, the switch to 32 MHz operation is complete. Conversely, when PLLR is cleared the switch from 32 MHz operation to the selected internal clock is complete.

## 19.10 Analog Input Connection Considerations

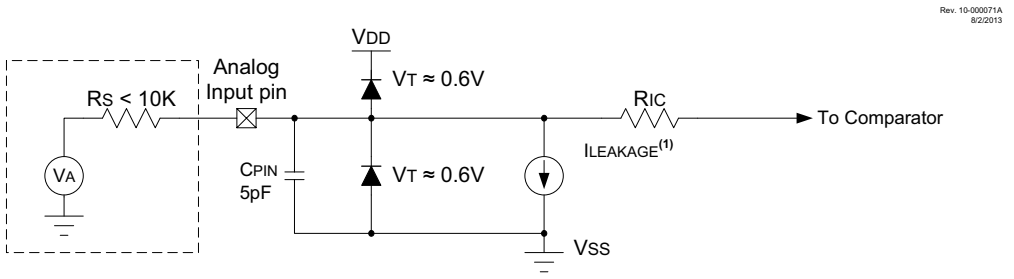
A simplified circuit for an analog input is shown in Figure 19-4. Since the analog input pins share their connection with a digital input, they have reverse biased ESD protection diodes to VDD and VSS. The analog input, therefore, must be between VSS and VDD. If the input voltage deviates from this range by more than 0.6V in either direction, one of the diodes is forward biased and a latch-up may occur.

A maximum source impedance of 10 kΩ is recommended for the analog sources. Also, any external component connected to an analog input pin, such as a capacitor or a Zener diode, should have very little leakage current to minimize inaccuracies introduced.

**Note 1:** When reading a PORT register, all pins configured as analog inputs will read as a '0'. Pins configured as digital inputs will convert as an analog input, according to the input specification.

**2:** Analog levels on any pin defined as a digital input, may cause the input buffer to consume more current than is specified.

**FIGURE 19-4: ANALOG INPUT MODEL**




**Legend:** CPIN = Input Capacitance
ILEAKAGE = Leakage Current at the pin due to various junctions
RIC = Interconnect Resistance
RS = Source Impedance
VA = Analog Voltage
VT = Threshold Voltage

**Note 1:** See I/O Ports in Table 36-4: I/O Ports.

## 21.2 Register Definitions: Option Register

**REGISTER 21-1: OPTION_REG: OPTION REGISTER**

| R/W-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 |
|---|---|---|---|---|---|---|---|
| $\overline{\text{WPUEN}}$ | INTEDG | TMR0CS | TMR0SE | PSA | PS<2:0> | | |
| bit 7 | | | | | | | bit 0 |

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

bit 7 **$\overline{\text{WPUEN}}$:** Weak Pull-Up Enable bit
1 = All weak pull-ups are disabled (except $\overline{\text{MCLR}}$, if it is enabled)
0 = Weak pull-ups are enabled by individual WPUx latch values

bit 6 **INTEDG:** Interrupt Edge Select bit
1 = Interrupt on rising edge of INT pin
0 = Interrupt on falling edge of INT pin

bit 5 **TMR0CS:** Timer0 Clock Source Select bit
1 = Transition on T0CKI pin
0 = Internal instruction cycle clock (FOSC/4)

bit 4 **TMR0SE:** Timer0 Source Edge Select bit
1 = Increment on high-to-low transition on T0CKI pin
0 = Increment on low-to-high transition on T0CKI pin

bit 3 **PSA:** Prescaler Assignment bit
1 = Prescaler is not assigned to the Timer0 module
0 = Prescaler is assigned to the Timer0 module

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

| Bit Value | Timer0 Rate |
|---|---|
| 000 | 1 : 2 |
| 001 | 1 : 4 |
| 010 | 1 : 8 |
| 011 | 1 : 16 |
| 100 | 1 : 32 |
| 101 | 1 : 64 |
| 110 | 1 : 128 |
| 111 | 1 : 256 |

**TABLE 21-1: SUMMARY OF REGISTERS ASSOCIATED WITH TIMER0**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|
| INTCON | GIE | PEIE | TMR0IE | INTE | IOCIE | TMR0IF | INTF | IOCIF | 132 |
| OPTION_REG | $\overline{\text{WPUEN}}$ | INTEDG | TMR0CS | TMR0SE | PSA | PS<2:0> | | | 274 |
| TMR0 | Timer0 Module Register | | | | | | | | 272* |
| TRISA | TRISA7 | TRISA6 | TRISA5 | TRISA4 | TRISA3 | TRISA2 | TRISA1 | TRISA0 | 176 |

**Legend:** — = Unimplemented location, read as '0'. Shaded cells are not used by the Timer0 module.
* Page provides register information.

## 22.0 TIMER1/3/5 MODULE WITH GATE CONTROL

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

- 16-bit timer/counter register pair (TMR1H:TMR1L)
- Programmable internal or external clock source
- 2-bit prescaler
- Dedicated 32 kHz oscillator circuit
- Optionally synchronized comparator out
- Multiple Timer1 gate (count enable) sources
- Interrupt-on-overflow
- Wake-up on overflow (external clock, Asynchronous mode only)
- Time base for the Capture/Compare function
- Auto-conversion Trigger (with CCP)
- Selectable Gate Source Polarity
- Gate Toggle mode
- Gate Single-pulse mode
- Gate Value Status
- Gate Event Interrupt

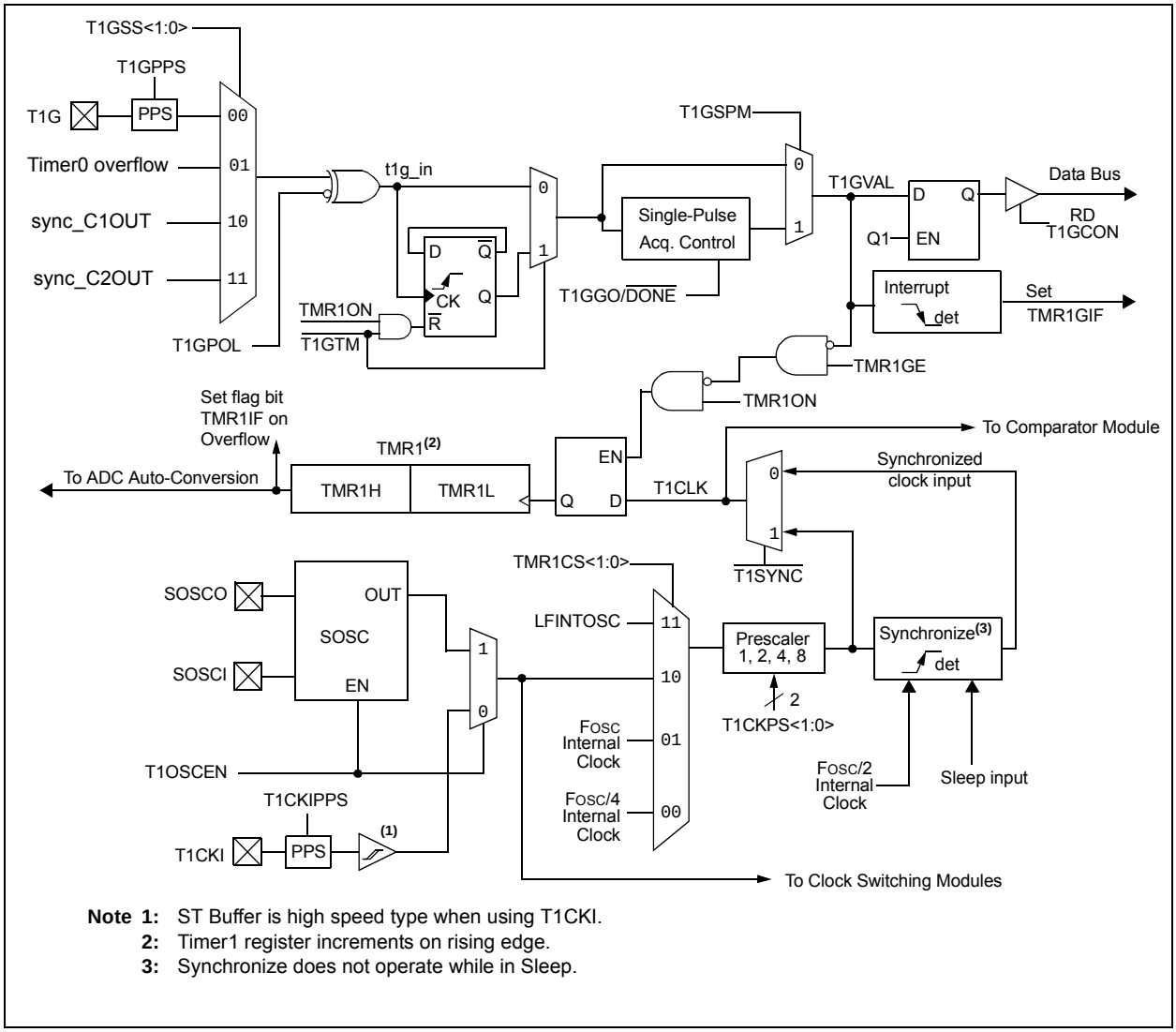
Figure 22-1 is a block diagram of the Timer1 module.

This device has three instances of Timer1 type modules. They include:

- Timer1
- Timer3
- Timer5

| **Note:** | All references to Timer1 and Timer1 Gate apply to Timer3 and Timer5. |
|---|---|

**FIGURE 22-1: TIMER1 BLOCK DIAGRAM**




**Note 1:** ST Buffer is high speed type when using T1CKI.
**2:** Timer1 register increments on rising edge.
**3:** Synchronize does not operate while in Sleep.

## 22.3 Timer1 Prescaler

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

## 22.4 Timer1 (Secondary) Oscillator

A dedicated low-power 32.768 kHz oscillator circuit is built-in between pins SOSCI (input) and SOSCO (amplifier output). This internal circuit is to be used in conjunction with an external 32.768 kHz crystal.

The oscillator circuit is enabled by setting the OSCEN bit of the T1CON register. The oscillator will continue to run during Sleep.

| | |
|---|---|
| **Note:** | The oscillator requires a start-up and stabilization time before use. Thus, OSCEN should be set and a suitable delay observed prior to using Timer1. A suitable delay similar to the OST delay can be implemented in software by clearing the TMR1IF bit then presetting the TMR1H:TMR1L register pair to FC00h. The TMR1IF flag will be set when 1024 clock cycles have elapsed, thereby indicating that the oscillator is running and reasonably stable. |

## 22.5 Timer1 Operation in Asynchronous Counter Mode

If the control bit $\overline{\text{SYNC}}$ of the T1CON register is set, the external clock input is not synchronized. The timer increments asynchronously to the internal phase clocks. If the external clock source is selected then the timer will continue to run during Sleep and can generate an interrupt-on-overflow, which will wake-up the processor. However, special precautions in software are needed to read/write the timer (see **Section 22.5.1 "Reading and Writing Timer1 in Asynchronous Counter Mode"**).

| | |
|---|---|
| **Note:** | When switching from synchronous to asynchronous operation, it is possible to skip an increment. When switching from asynchronous to synchronous operation, it is possible to produce an additional increment. |

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

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

For writes, it is recommended that the user simply stop the timer and write the desired values. A write contention may occur by writing to the timer registers, while the register is incrementing. This may produce an unpredictable value in the TMR1H:TMR1L register pair.

## 22.6 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.

### 22.6.1 TIMER1 GATE ENABLE

The Timer1 Gate Enable mode is enabled by setting the GE bit of the T1GCON register. The polarity of the Timer1 Gate Enable mode is configured using the GPOL bit of the T1GCON register.

When Timer1 Gate Enable mode is enabled, Timer1 will increment on the rising edge of the Timer1 clock source. When Timer1 Gate Enable mode is disabled, no incrementing will occur and Timer1 will hold the current count. See Figure 22-3 for timing details.

**TABLE 22-3: TIMER1 GATE ENABLE SELECTIONS**

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

**REGISTER 23-4: TxRST: TIMERx EXTERNAL RESET SIGNAL SELECTION REGISTER**

| U-0 | U-0 | U-0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| — | — | — | RSEL<4:0> | | | | |
| bit 7 | | | | | | | bit 0 |

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

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

bit 4-0 **RSEL<4:0>:** TimerX External Reset Signal Source Selection bits
See Table 23-4.

**TABLE 23-4: EXTERNAL RESET SOURCES**

| RSEL<4:0> | Timer2 | Timer4 | Timer6 | Timer8 |
|---|---|---|---|---|
| 11111 | Reserved | Reserved | Reserved | Reserved |
| 11110 | Reserved | Reserved | Reserved | Reserved |
| 11101 | LC4_out | LC4_out | LC4_out | LC4_out |
| 11100 | LC3_out | LC3_out | LC3_out | LC3_out |
| 11011 | LC2_out | LC2_out | LC2_out | LC2_out |
| 11010 | LC1_out | LC1_out | LC1_out | LC1_out |
| 11001 | ZCD_out | ZCD_out | ZCD_out | ZCD_out |
| 11000(1) | sync_C8OUT | sync_C8OUT | sync_C8OUT | sync_C8OUT |
| 10111(1) | sync_C7OUT | sync_C7OUT | sync_C7OUT | sync_C7OUT |
| 10110 | sync_C6OUT | sync_C6OUT | sync_C6OUT | sync_C6OUT |
| 10101 | sync_C5OUT | sync_C5OUT | sync_C5OUT | sync_C5OUT |
| 10100 | sync_C4OUT | sync_C4OUT | sync_C4OUT | sync_C4OUT |
| 10011 | sync_C3OUT | sync_C3OUT | sync_C3OUT | sync_C3OUT |
| 10010 | sync_C2OUT | sync_C2OUT | sync_C2OUT | sync_C2OUT |
| 10001 | sync_C1OUT | sync_C1OUT | sync_C1OUT | sync_C1OUT |
| 10000(1) | PWM12_out | PWM12_out | PWM12_out | PWM12_out |
| 01111 | PWM11_out | PWM11_out | PWM11_out | PWM11_out |
| 01110 | PWM6_out | PWM6_out | PWM6_out | PWM6_out |
| 01101 | PWM5_out | PWM5_out | PWM5_out | PWM5_out |
| 01100(1) | PWM10_out | PWM10_out | PWM10_out | PWM10_out |
| 01011 | PWM9_out | PWM9_out | PWM9_out | PWM9_out |
| 01010 | PWM4_out | PWM4_out | PWM4_out | PWM4_out |
| 01001 | PWM3_out | PWM3_out | PWM3_out | PWM3_out |
| 01000(1) | CCP8_out | CCP8_out | CCP8_out | CCP8_out |
| 00111 | CCP7_out | CCP7_out | CCP7_out | CCP7_out |
| 00110 | CCP2_out | CCP2_out | CCP2_out | CCP2_out |
| 00101 | CCP1_out | CCP1_out | CCP1_out | CCP1_out |
| 00100 | TMR8_postscaled | TMR8_postscaled | TMR8_postscaled | Reserved |
| 00011 | TMR6_postscaled | TMR6_postscaled | Reserved | TMR6_postscaled |
| 00010 | TMR4_postscaled | Reserved | TMR4_postscaled | TMR4_postscaled |
| 00001 | Reserved | TMR2_postscaled | TMR2_postscaled | TMR2_postscaled |
| 00000 | Pin selected byT2INPPS | Pin selected by T4INPPS | Pin selected by T6INPPS | Pin selected by T6INPPS |

**Note 1:** PIC16LF1777/9 only.

## 25.1 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.

### 25.1.1 FUNDAMENTAL OPERATION

The PWM module produces a 10-bit resolution output. Timer2 and T2PR set the period of the PWM. The PWMxDCL and PWMxDCH registers configure the duty cycle. The period is common to all PWM modules, whereas the duty cycle is independently controlled.

**Note:** 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 TMR2 is cleared. Each PWMx is cleared when TMR2 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).

**Note:** The PWMxDCH and PWMxDCL registers are double buffered. The buffers are updated when Timer2 matches T2PR. Care should be taken to update both registers before the timer match occurs.

### 25.1.2 PWM OUTPUT POLARITY

The output polarity is inverted by setting the PWMxPOL bit of the PWMxCON register.

### 25.1.3 PWM PERIOD

The PWM period is specified by the T2PR register of Timer2. The PWM period can be calculated using the formula of Equation 25-1.

**EQUATION 25-1: PWM PERIOD**

$$PWM\ Period = [T2PR + 1] \bullet 4 \bullet T_{OSC} \bullet (TMR2\ Prescale\ Value)$$

**Note:** TOSC = 1/FOSC

When TMR2 is equal to T2PR, the following three events occur on the next increment cycle:

- TMR2 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.

**Note:** The Timer2 postscaler has no effect on the PWM operation.

### 25.1.4 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.

Equation 25-2 is used to calculate the PWM pulse width.

Equation 25-3 is used to calculate the PWM duty cycle ratio.

**EQUATION 25-2: PULSE WIDTH**

$$Pulse\ Width = (PWMxDCH:PWMxDCL<7:6>) \bullet T_{OSC} \bullet (TMR2\ Prescale\ Value)$$

**Note:** TOSC = 1/FOSC

**EQUATION 25-3: DUTY CYCLE RATIO**

$$Duty\ Cycle\ Ratio = \frac{(PWMxDCH:PWMxDCL<7:6>)}{4(T2PR + 1)}$$

The 8-bit timer TMR2 register is concatenated with the two Least Significant bits of 1/FOSC, 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.

**REGISTER 27-14: COGxDBR: COG RISING EVENT DEAD-BAND COUNT REGISTER**

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

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

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

bit 5-0 **DBR<5:0>:** Rising Event Dead-Band Count Value bits

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

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

**REGISTER 27-15: COGxDBF: COG FALLING EVENT DEAD-BAND COUNT REGISTER**

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

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

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

bit 5-0 **DBF<5:0>:** Falling Event Dead-Band Count Value bits

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

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

## 28.6 Register Definitions: CLC Control

Long bit name prefixes for the CLC peripherals are shown in Table 28-3. Refer to **Section 1.1 "Register and Bit naming conventions"** for more information

**TABLE 28-3:**

| Peripheral | Bit Name Prefix |
|---|---|
| CLC1 | LC1 |
| CLC2 | LC2 |
| CLC3 | LC3 |
| CLC4 | LC4 |

**REGISTER 28-1: CLCxCON: CONFIGURABLE LOGIC CELL CONTROL REGISTER**

| R/W-0/0 | U-0 | R-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| EN | — | OUT | INTP | INTN | MODE<2:0> | | |
| bit 7 | | | | | | | bit 0 |

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

bit 7 **EN:** Configurable Logic Cell Enable bit
- 1 = Configurable logic cell is enabled and mixing input signals
- 0 = Configurable logic cell is disabled and has logic zero output

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

bit 5 **OUT:** Configurable Logic Cell Data Output bit
Read-only: logic cell output data, after POL; sampled from lcx_out wire.

bit 4 **INTP:** Configurable Logic Cell Positive Edge Going Interrupt Enable bit
- 1 = CLCxIF will be set when a rising edge occurs on lcx_out
- 0 = CLCxIF will not be set

bit 3 **INTN:** Configurable Logic Cell Negative Edge Going Interrupt Enable bit
- 1 = CLCxIF will be set when a falling edge occurs on lcx_out
- 0 = CLCxIF will not be set

bit 2-0 **MODE<2:0>:** Configurable Logic Cell Functional Mode bits
- 111 = Cell is 1-input transparent latch with S and R
- 110 = Cell is J-K flip-flop with R
- 101 = Cell is 2-input D flip-flop with R
- 100 = Cell is 1-input D flip-flop with S and R
- 011 = Cell is S-R latch
- 010 = Cell is 4-input AND
- 001 = Cell is OR-XOR
- 000 = Cell is AND-OR

**REGISTER 31-4: MDxCARH: MODULATION HIGH CARRIER CONTROL REGISTER**

| U-0 | U-0 | U-0 | R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u |
|---|---|---|---|---|---|---|---|
| — | — | — | CH<4:0> | | | | |
| bit 7 | | | | | | | bit 0 |

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

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

bit 4-0 **CH<4:0>** Modulator Data High Carrier Selection bits(1)
See Table 31-6.

**Note 1:** Narrowed carrier pulse widths or spurs may occur in the signal stream if the carrier is not synchronized.

### 32.2.3 SPI MASTER MODE

The master can initiate the data transfer at any time because it controls the SCK line. The master determines when the slave (Processor 2, Figure 32-5) is to broadcast data by the software protocol.

In Master mode, the data is transmitted/received as soon as the SSPxBUF register is written to. If the SPI is only going to receive, the SDO output could be disabled (programmed as an input). The SSPSR register will continue to shift in the signal present on the SDI pin at the programmed clock rate. As each byte is received, it will be loaded into the SSPxBUF register as if a normal received byte (interrupts and Status bits appropriately set).

The clock polarity is selected by appropriately programming the CKP bit of the SSPxCON1 register and the CKE bit of the SSPxSTAT register. This then, would give waveforms for SPI communication as shown in Figure 32-6, Figure 32-8, Figure 32-9 and Figure 32-10, where the MSB is transmitted first. In Master mode, the SPI clock rate (bit rate) is user programmable to be one of the following:

- FOSC/4 (or TCY)
- FOSC/16 (or 4 * TCY)
- FOSC/64 (or 16 * TCY)
- Timer2 output/2
- FOSC/(4 * (SSPxADD + 1))

Figure 32-6 shows the waveforms for Master mode.

When the CKE bit is set, the SDO data is valid before there is a clock edge on SCK. The change of the input sample is shown based on the state of the SMP bit. The time when the SSPxBUF is loaded with the received data is shown.

| **Note:** | In Master mode the clock signal output to the SCK pin is also the clock signal input to the peripheral. The pin selected for output with the RxyPPS register must also be selected as the peripheral input with the SSPCLKPPS register. |
|---|---|

**TABLE 35-3: INSTRUCTION SET (CONTINUED)**

| Mnemonic, Operands | | Description | Cycles | 14-Bit Opcode MSb LSb | Status Affected | Notes |
|---|---|---|---|---|---|---|
| **CONTROL OPERATIONS** | | | | | | |
| BRA | k | Relative Branch | 2 | 11 001k kkkk kkkk | | |
| BRW | – | Relative Branch with W | 2 | 00 0000 0000 1011 | | |
| CALL | k | Call Subroutine | 2 | 10 0kkk kkkk kkkk | | |
| CALLW | – | Call Subroutine with W | 2 | 00 0000 0000 1010 | | |
| GOTO | k | Go to address | 2 | 10 1kkk kkkk kkkk | | |
| RETFIE | k | Return from interrupt | 2 | 00 0000 0000 1001 | | |
| RETLW | k | Return with literal in W | 2 | 11 0100 kkkk kkkk | | |
| RETURN | – | Return from Subroutine | 2 | 00 0000 0000 1000 | | |
| **INHERENT OPERATIONS** | | | | | | |
| CLRWDT | – | Clear Watchdog Timer | 1 | 00 0000 0110 0100 | $\overline{TO}$, $\overline{PD}$ | |
| NOP | – | No Operation | 1 | 00 0000 0000 0000 | | |
| OPTION | – | Load OPTION_REG register with W | 1 | 00 0000 0110 0010 | | |
| RESET | – | Software device Reset | 1 | 00 0000 0000 0001 | | |
| SLEEP | – | Go into Standby mode | 1 | 00 0000 0110 0011 | $\overline{TO}$, $\overline{PD}$ | |
| TRIS | f | Load TRIS register with W | 1 | 00 0000 0110 0fff | | |
| **C-COMPILER OPTIMIZED** | | | | | | |
| ADDFSR | n, k | Add Literal k to FSRn | 1 | 11 0001 0nkk kkkk | | |
| MOVIW | n mm | Move Indirect FSRn to W with pre/post inc/dec modifier, mm | 1 | 00 0000 0001 0nmm | Z | **2, 3** |
| | k[n] | Move INDFn to W, Indexed Indirect. | 1 | 11 1111 0nkk kkkk | Z | **2** |
| MOVWI | n mm | Move W to Indirect FSRn with pre/post inc/dec modifier, mm | 1 | 00 0000 0001 1nmm | | **2, 3** |
| | k[n] | Move W to INDFn, Indexed Indirect. | 1 | 11 1111 1nkk kkkk | | **2** |

**Note 1:** If the Program Counter (PC) is modified, or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a NOP.

**2:** If this instruction addresses an INDF register and the MSb of the corresponding FSR is set, this instruction will require one additional instruction cycle.

**3:** See Table in the MOVIW and MOVWI instruction descriptions.

**TABLE 36-6: THERMAL CHARACTERISTICS**

| Standard Operating Conditions (unless otherwise stated) | | | | | |
|---|---|---|---|---|---|
| **Param No.** | **Sym.** | **Characteristic** | **Typ.** | **Units** | **Conditions** |
| TH01 | $\theta_{JA}$ | Thermal Resistance Junction to Ambient | 60.0 | °C/W | 28-pin SPDIP package |
| | | | 80.0 | °C/W | 28-pin SOIC package |
| | | | 90.0 | °C/W | 28-pin SSOP package |
| | | | 48 | °C/W | 28-pin UQFN 4x4mm package |
| | | | 47.2 | °C/W | 40-pin PDIP package |
| | | | 46.0 | °C/W | 44-pin TQFP package |
| | | | 41.0 | °C/W | 40-pin UQFN 5x5mm package |
| TH02 | $\theta_{JC}$ | Thermal Resistance Junction to Case | 31.4 | °C/W | 28-pin SPDIP package |
| | | | 24 | °C/W | 28-pin SOIC package |
| | | | 24 | °C/W | 28-pin SSOP package |
| | | | 12 | °C/W | 28-pin UQFN 4x4mm package |
| | | | 24.70 | °C/W | 40-pin PDIP package |
| | | | 14.5 | °C/W | 44-pin TQFP package |
| | | | 5.5 | °C/W | 40-pin UQFN 5x5mm package |
| TH03 | $T_{JMAX}$ | Maximum Junction Temperature | 150 | °C | |
| TH04 | PD | Power Dissipation | — | W | $PD = P_{INTERNAL} + P_{I/O}$ |
| TH05 | $P_{INTERNAL}$ | Internal Power Dissipation | — | W | $P_{INTERNAL} = I_{DD} \times V_{DD}$ (1) |
| TH06 | $P_{I/O}$ | I/O Power Dissipation | — | W | $P_{I/O} = \Sigma (I_{OL} * V_{OL}) + \Sigma (I_{OH} * (V_{DD} - V_{OH}))$ |
| TH07 | $P_{DER}$ | Derated Power | — | W | $P_{DER} = P_{DMAX} (T_J - T_A)/\theta_{JA}$ (2) |

**Note 1:** $I_{DD}$ is current to run the chip alone without driving any load on the output pins.

**2:** $T_A$ = Ambient Temperature, $T_J$ = Junction Temperature

**FIGURE 36-13: ADC CONVERSION TIMING (ADC CLOCK Fosc-BASED)**

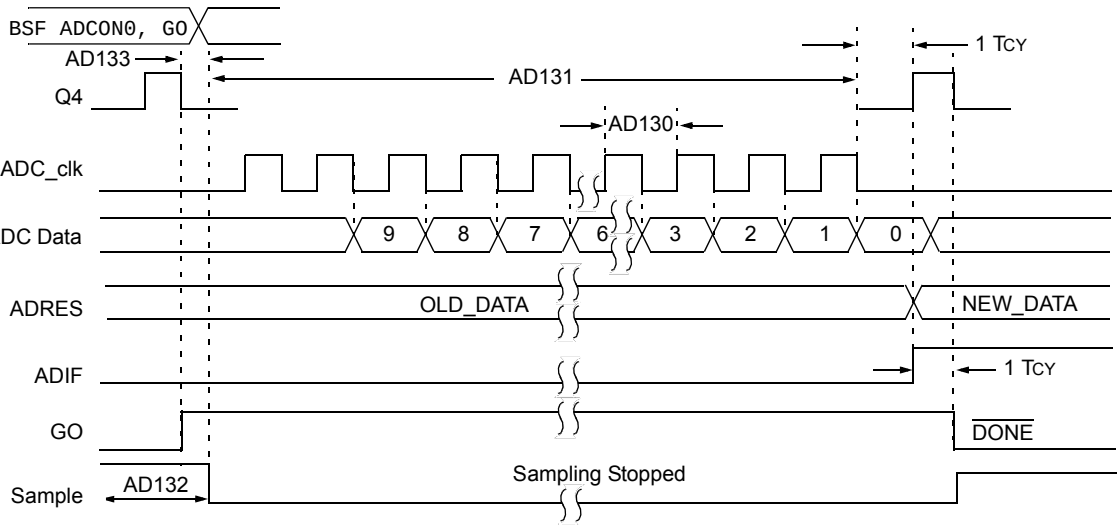
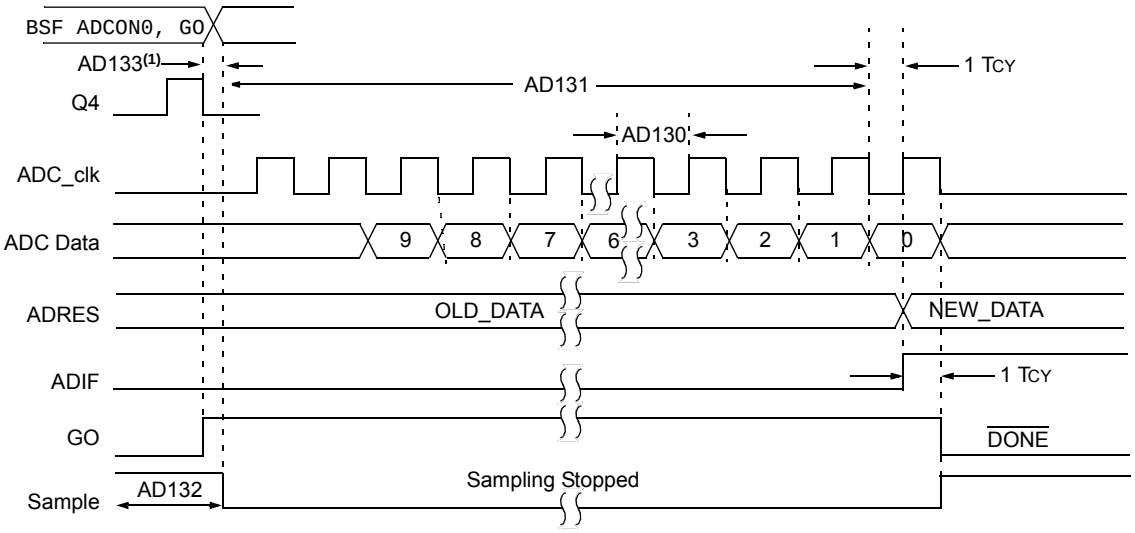



**FIGURE 36-14: ADC CONVERSION TIMING (ADC CLOCK FROM FRC)**




**Note 1:** If the ADC clock source is selected as FRC, a time of TCY is added before the ADC clock starts. This allows the SLEEP instruction to be executed.