

---

## Section 38. Oscillator with 500 kHz Low-Power FRC

---

### HIGHLIGHTS

This section of the manual contains the following major topics:

|       |                                               |       |
|-------|-----------------------------------------------|-------|
| 38.1  | Introduction .....                            | 38-2  |
| 38.2  | CPU Clocking Scheme .....                     | 38-3  |
| 38.3  | Oscillator Configuration.....                 | 38-4  |
| 38.4  | Control Registers .....                       | 38-6  |
| 38.5  | Primary Oscillator (POSC) .....               | 38-12 |
| 38.6  | Phase Locked Loop (PLL) Branch .....          | 38-17 |
| 38.7  | Secondary Oscillator (SOSC) .....             | 38-18 |
| 38.8  | Internal Fast RC Oscillator (FRC) .....       | 38-18 |
| 38.9  | Internal Low-Power RC Oscillator (LPRC) ..... | 38-19 |
| 38.10 | Fail-Safe Clock Monitor (FSCM) .....          | 38-19 |
| 38.11 | Clock Switching Operation .....               | 38-20 |
| 38.12 | Two-Speed Start-up .....                      | 38-24 |
| 38.13 | Reference Clock Output Generator .....        | 38-24 |
| 38.14 | AC Electrical Specifications .....            | 38-25 |
| 38.15 | Design Tips .....                             | 38-26 |
| 38.16 | Register Maps .....                           | 38-27 |
| 38.17 | Related Application Notes.....                | 38-28 |
| 38.18 | Revision History .....                        | 38-29 |

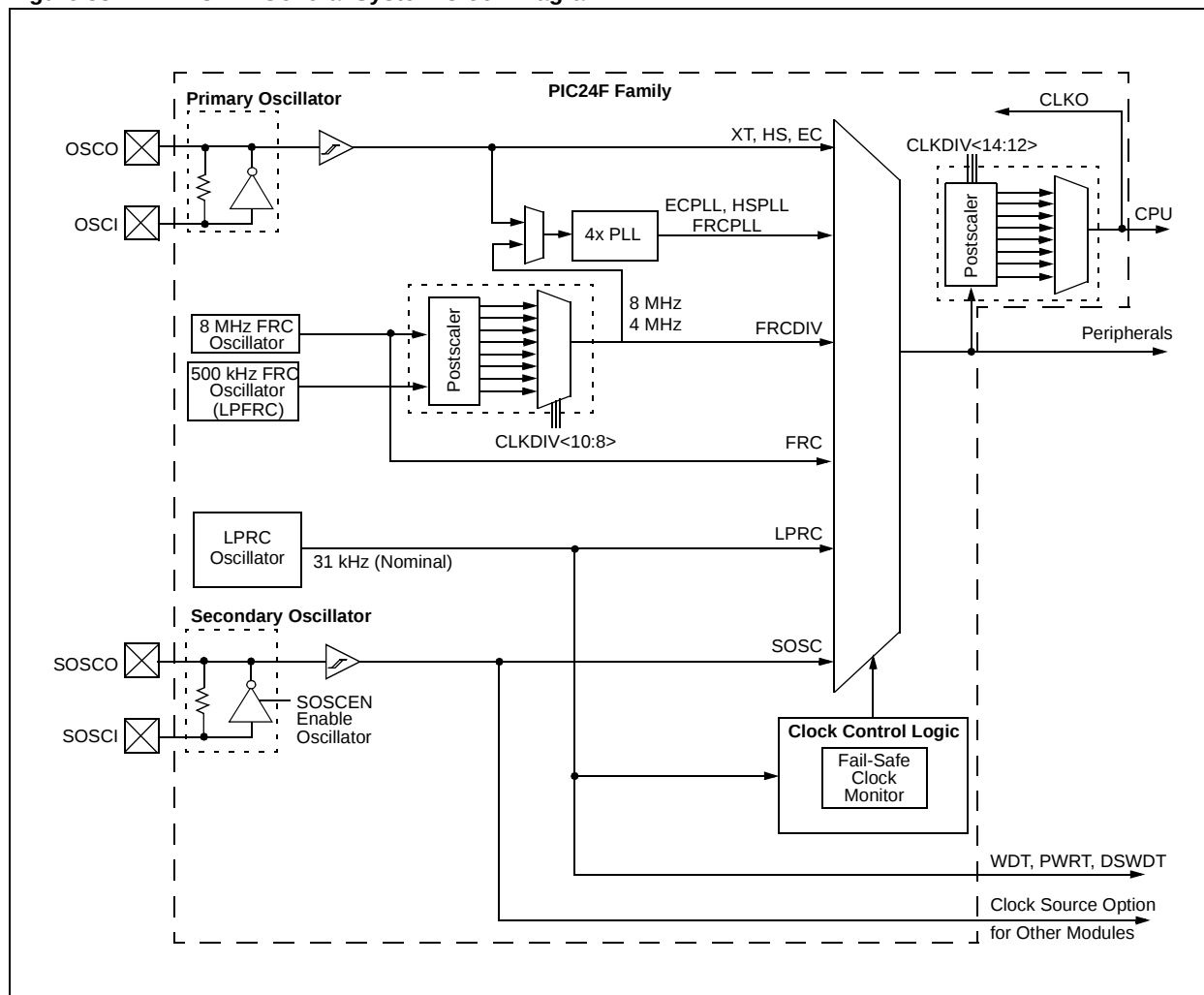
## 38.1 INTRODUCTION

This section describes the PIC24F oscillator system and its operation. The PIC24F oscillator system has the following modules and features:

- A total of five external and internal oscillator options as clock sources, providing up to 12 different clock modes.
- An on-chip Phase Locked Loop (PLL) block to boost internal operating frequency on select internal and external oscillator sources or to provide a precise clock source for special peripheral features (in select devices only).
- Software-controllable switching between various clock sources.
- Software-controllable postscaler for selective clocking of CPU for system power saving.
- A Fail-Safe Clock Monitor (FSCM) that detects clock failure and permits safe application recovery or shutdown.
- A programmable reference clock generator to provide a clock source with a wide range of frequencies for synchronizing external devices (in select devices only).

Figure 38-1 displays a simplified diagram of the PIC24F oscillator system.

**Figure 38-1: PIC24F General System Clock Diagram**



## 38.2 CPU CLOCKING SCHEME

The system clock source can be provided by one of five sources:

- Primary Oscillator (POSC) on the OSC1 and OSCO pins
- Secondary Oscillator (SOSC) on the SOSCI and SOSCO pins

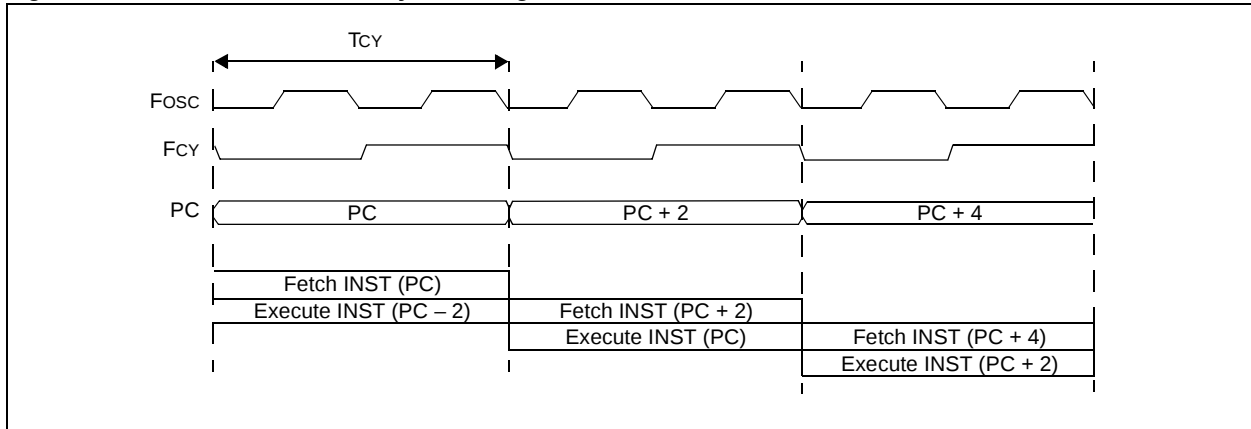
See **Section 38.7 “Secondary Oscillator (SOSC)”** for more details.

- Fast Internal RC (FRC) Oscillator
  - 8 MHz FRC oscillator
  - 500 kHz Low-Power FRC (LPFRC) oscillator
- Low-Power Internal RC (LPRC) Oscillator

The primary oscillator and 8 MHz FRC sources have the option of using the internal 4x PLL. The frequency of the FRC clock source can optionally be reduced by the programmable clock divider. The selected clock source generates the processor and peripheral clock sources.

The processor clock source is divided by two to produce the internal instruction cycle clock, Fcy. In this document, the instruction cycle clock is also denoted by Fosc/2. The timing diagram in Figure 38-2 illustrates the relationship between the processor clock source and instruction execution. The internal instruction cycle clock, Fosc/2, can be provided on the OSC2 I/O pin for some operating modes of the primary oscillator.

**Figure 38-2: Clock/Instruction Cycle Timing**



## 38.3 OSCILLATOR CONFIGURATION

The oscillator source (and operating mode) that is used at a device Power-on Reset (POR) event is selected using Configuration bit settings. The oscillator Configuration bit settings are in the Configuration registers (refer to the specific product data sheet for further details). These bits are mapped starting at program memory location F80000h. The Primary Oscillator Configuration bits, POSCMD<1:0> (Configuration Word FOSC<1:0>), and Oscillator Configuration bits, FNOSC<2:0> (Configuration Word FOSCSEL<2:0>), select the oscillator source that is used at POR. The FRC oscillator with postscaler (FRCDIV) is the default (unprogrammed) selection. The secondary oscillator, or one of the internal oscillators, may be chosen by programming these bit locations.

The Configuration bits allow users to choose among 11 different clock modes, listed in Table 38-1.

**Table 38-1: Configuration Bit Values for Clock Selection**

| Oscillator Mode                                 | Oscillator Source | POSCMD<1:0> | FNOSC<2:0> | Note        |
|-------------------------------------------------|-------------------|-------------|------------|-------------|
| 8 MHz FRC Oscillator with Postscaler (FRCDIV)   | Internal          | 11          | 111        | <b>1, 2</b> |
| 500 KHz FRC Oscillator with Postscaler (FRCDIV) | Internal          | 11          | 110        | <b>1</b>    |
| Low-Power RC Oscillator (LPRC)                  | Internal          | 11          | 101        | <b>1</b>    |
| Secondary (Timer1) Oscillator (SOSC)            | Secondary         | 11          | 100        | <b>1</b>    |
| Primary Oscillator (HS) with PLL Module (HSPLL) | Primary           | 10          | 011        | —           |
| Primary Oscillator (XT) with PLL Module (XTPLL) | Primary           | 01          | 011        | —           |
| Primary Oscillator (EC) with PLL Module (ECPLL) | Primary           | 00          | 011        | —           |
| Primary Oscillator (HS)                         | Primary           | 10          | 010        | —           |
| Primary Oscillator (XT)                         | Primary           | 01          | 010        | —           |
| Primary Oscillator (EC)                         | Primary           | 00          | 010        | —           |
| 8 MHz FRCPLL                                    | Internal          | 11          | 001        | <b>1</b>    |
| 8 MHz FRC                                       | Internal          | 11          | 000        | <b>1</b>    |

**Note 1:** OSC2 pin function is determined by the OSCIOFCN Configuration bit.

**2:** Default oscillator mode for an unprogrammed (erased) device.

### 38.3.1 Oscillator Frequency Configuration Bits

In order for the device to apply the most efficient power mode when using the primary oscillator (EC mode), a frequency range of EC is configurable using the POSCFREQ<1:0> (Configuration Word FOSC<4:3>) bits.

### 38.3.2 Clock Switching Mode Configuration Bits

The FCKSM<1:0> Configuration bits are used to jointly configure device clock switching and the Fail-Safe Clock Monitor (FSCM). Clock switching is enabled only when FCKSM1 is programmed ('0'). The FSCM is enabled only when the FCKSM<1:0> (Configuration Word FOSC<7:6>) bits are both programmed ('00').

### 38.3.3 OSC1 and OSC2 Pin Functions in Non-Crystal Modes

When the primary oscillator on OSC1 and OSC2 is not configured as the clock source (POSCMD<1:0> = 11), the OSC1 pin is automatically reconfigured as a digital I/O. In this configuration, and also when the primary oscillator is configured for EC mode (POSCMD<1:0> = 00), the OSC2 pin can also be configured as a digital I/O by programming the OSCIOFCN (Configuration Word FOSC<2>) bit.

When OSCIOFCN is unprogrammed ('1'), a Fosc/2 clock output is available on OSC2 for testing or synchronization purposes. With OSCIOFCN programmed ('0'), the OSC2 pin becomes a general purpose I/O pin. In both of these configurations, the feedback device between OSC1 and OSC2 is turned off to save current.

## 38.4 CONTROL REGISTERS

The operation of the oscillator is controlled by three (four for some devices) Special Function Registers (SFRs):

- OSCCON
- CLKDIV
- OSCTUN
- REFOCON

### 38.4.1 Oscillator Control Register (OSCCON)

The OSCCON register (Register 38-1) is the main control register for the oscillator. It controls clock source switching and allows the monitoring of clock sources.

The COSC status bits are read-only bits that indicate the current oscillator source from which the device is operating. The COSC bits default to the internal FRC oscillator with postscaler (FRCDIV) and are configured for 4 MHz on a POR and Master Clear (MCLR) Reset. A clock switch will be performed automatically to the new oscillator source selected by the FNOSC<1:0> Configuration bits. The COSC bits will change to indicate the new oscillator source at the end of a clock switch operation.

The NOSC status bits select the clock source for the next clock switch operation. On POR and MCLR, these bits automatically select the oscillator source defined by the FNOSC Configuration bits. These bits can be modified by the software.

|                                                                                                                                                                   |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Note:</b> An unlock sequence must be performed before writing to OSCCON. Refer to <b>Section 38.11.2 “Oscillator Switching Sequence”</b> for more information. |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Setting the CLKLOCK bit (OSCCON<7>) prevents clock switching if the FCKSM1 Configuration bit is set. If the FCKSM1 bit is clear, the CLKLOCK bit state is ignored and clock switching can occur.

The LOCK status bit (OSCCON<5>) is read-only and indicates the status of the PLL circuit. It is set when the PLL achieves a frequency lock and is reset when a valid clock switching sequence is initiated. It reads as '0' whenever the PLL is not used as part of the current clock source.

The CF status bit (OSCCON<3>) is a readable/clearable bit that indicates a clock failure; it is reset whenever a valid clock switch occurs.

The SOSSEN Control bit (OSCCON<1>) is used to enable or disable the 32 kHz SOSC crystal oscillator.

The OSWEN Control bit (OSCCON<0>) is used to initiate a clock switch operation. OSWEN is cleared automatically after a successful clock switch, any redundant clock switch and by the FSCM module after the switch to the FRC has completed.

## Section 38. Oscillator with 500 kHz Low-Power FRC

**Register 38-1: OSCCON: Oscillator Control Register**

| U-0    | R-0   | R-0   | R-0   | U-0   | R/W-x <sup>(1)</sup> | R/W-x <sup>(1)</sup> | R/W-x <sup>(1)</sup> |
|--------|-------|-------|-------|-------|----------------------|----------------------|----------------------|
| —      | COSC2 | COSC1 | COSC0 | —     | NOSC2                | NOSC1                | NOSC0                |
| bit 15 |       |       |       | bit 8 |                      |                      |                      |

| R/SO-0  | U-0 | R-0 <sup>(2)</sup> | U-0 | R/CO-0 | U-0 | R/W-0  | R/W-0 |
|---------|-----|--------------------|-----|--------|-----|--------|-------|
| CLKLOCK | —   | LOCK               | —   | CF     | —   | SOSCEN | OSWEN |
| bit 7   |     |                    |     | bit 0  |     |        |       |

|                   |                         |                                              |
|-------------------|-------------------------|----------------------------------------------|
| <b>Legend:</b>    | CO = Clearable Only bit | SO = Settable Only bit                       |
| R = Readable bit  | W = Writable bit        | U = Unimplemented bit, read as '0'           |
| -n = Value at POR | '1' = Bit is set        | '0' = Bit is cleared      x = Bit is unknown |

bit 15 **Unimplemented:** Read as '0'

bit 14-12 **COSC<2:0>:** Current Oscillator Selection bits

111 = 8 MHz Fast RC Oscillator with Postscaler (FRCDIV)  
 110 = 500 kHz Low-Power FRC Oscillator (LPFRC) with Postscaler (FRCDIV)  
 101 = Low-Power RC Oscillator (LPRC)  
 100 = Secondary Oscillator (SOSC)  
 011 = Primary Oscillator with PLL module (XTPLL, HSPLL and ECPLL)  
 010 = Primary Oscillator (XT, HS and EC)  
 001 = 8 MHz Fast RC Oscillator with Postscaler and PLL module (FRCPLL)  
 000 = 8 MHz Fast RC Oscillator (FRC)

bit 11 **Unimplemented:** Read as '0'

bit 10-8 **NOSC<2:0>:** New Oscillator Selection bits<sup>(1)</sup>

111 = 8 MHz Fast RC Oscillator with Postscaler (FRCDIV)  
 110 = 500 kHz Low-Power Fast RC Oscillator (FRC) with Postscaler (FRCDIV)  
 101 = Low-Power RC Oscillator (LPRC)  
 100 = Secondary Oscillator (SOSC)  
 011 = Primary Oscillator with PLL module (XTPLL, HSPLL and ECPLL)  
 010 = Primary Oscillator (XT, HS and EC)  
 001 = 8 MHz Fast RC Oscillator with Postscaler and PLL module (FRCPLL)  
 000 = 8 MHz Fast RC Oscillator (FRC)

bit 7 **CLKLOCK:** Clock Selection Lock Enabled bit

If FSCM is enabled (FCKSM1 = 1):

1 = Clock and PLL selections are locked

0 = Clock and PLL selections are not locked and may be modified by setting the OSWEN bit

If FSCM is disabled (FCKSM1 = 0):

Clock and PLL selections are never locked and may be modified by setting the OSWEN bit.

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

bit 5 **LOCK:** PLL Lock Status bit<sup>(2)</sup>

1 = PLL module is in lock or PLL module start-up timer is satisfied

0 = PLL module is out of lock, PLL start-up timer is running or PLL is disabled

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

bit 3 **CF:** Clock Fail Detect bit

1 = FSCM has detected a clock failure

0 = No clock failure has been detected

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

**Note 1:** Reset values for these bits are determined by the FNOSC Configuration bits.

**2:** Also resets to '0' during any valid clock switch or whenever a non-PLL Clock mode is selected.

## Register 38-1: OSCCON: Oscillator Control Register (Continued)

- bit 1      **SOSCEN:** 32 kHz Secondary Oscillator (SOSC) Enable bit  
1 = Enable secondary oscillator  
0 = Disable secondary oscillator
- bit 0      **OSWEN:** Oscillator Switch Enable bit  
1 = Initiate an oscillator switch to clock source specified by NOSC<2:0> bits  
0 = Oscillator switch is complete

- Note 1:** Reset values for these bits are determined by the FNOSC Configuration bits.  
**2:** Also resets to '0' during any valid clock switch or whenever a non-PLL Clock mode is selected.

### 38.4.2 Clock Divider Register (CLKDIV)

The Clock Divider register (Register 38-2) controls the features associated with Doze mode, as well as the postscaler for the FRC Oscillator.

The ROI bit (CLKDIV<15>) allows an interrupt to exit Doze mode and automatically selects a 1:1 ratio for the processor and peripheral clocks. The DOZEN bit (CLKDIV<11>) is cleared after the exit from Doze mode. Clearing the ROI bit prevents interrupts from affecting Doze mode.

The DOZE bits (CLKDIV<14:12>) select the ratio of processor clocks to peripheral clocks. The range is software selectable between 1:1 to 1:128. MCLR and PORs default to the 1:1 ratio. This feature allows the CPU to consume less power without disrupting the peripheral's operations.

Setting the DOZEN bit places the device into Doze mode and engages the processor clock postscaler. This bit is cleared when the ROI bit is set and an interrupt occurs.

The RCDIV bits (CLKDIV<10:8>) select the postscaler option for the FRC oscillator output, allowing users to choose a lower clock frequency than the nominal 8 MHz. This option is described more in detail in **Section 38.8.3 "FRC Postscaler Mode (FRCDIV)"** and **Section 38.8.4 "FRC Oscillator with PLL Mode (FRCPLL)"**.

## Section 38. Oscillator with 500 kHz Low-Power FRC

**Register 38-2: CLKDIV: Clock Divider Register**

| R/W-0  | R/W-0 | R/W-0 | R/W-0 | R/W-0                | R/W-0                 | R/W-0                 | R/W-1                 |
|--------|-------|-------|-------|----------------------|-----------------------|-----------------------|-----------------------|
| ROI    | DOZE2 | DOZE1 | DOZE0 | DOZEN <sup>(1)</sup> | RCDIV2 <sup>(2)</sup> | RCDIV1 <sup>(2)</sup> | RCDIV0 <sup>(2)</sup> |
| bit 15 |       |       |       |                      |                       |                       | bit 8                 |
|        |       |       |       |                      |                       |                       |                       |
| U-0    | U-0   | U-0   | U-0   | U-0                  | U-0                   | U-0                   | U-0                   |
| —      | —     | —     | —     | —                    | —                     | —                     | —                     |
| bit 7  |       |       |       |                      |                       |                       | bit 0                 |

**Legend:**

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

- bit 15      **ROI:** Recover on Interrupt bit  
1 = Interrupts clear the DOZEN bit and reset the CPU peripheral clock ratio to 1:1  
0 = Interrupts have no effect on the DOZEN bit
- bit 14-12    **DOZE<2:0>:** CPU Peripheral Clock Ratio Select bits  
111 = 1:128  
110 = 1:64  
101 = 1:32  
100 = 1:16  
011 = 1:8  
010 = 1:4  
001 = 1:2  
000 = 1:1
- bit 11      **DOZEN:** DOZE Enable bit<sup>(1)</sup>  
1 = DOZE<2:0> bits specify the CPU peripheral clock ratio  
0 = CPU peripheral clock ratio set to 1:1
- bit 10-8    **RCDIV<2:0>:** FRC Postscaler Select bits<sup>(2)</sup>  
111 = 1:256  
110 = 1:64  
101 = 1:32  
100 = 1:16  
011 = 1:8  
010 = 1:4  
001 = 1:2 (default)  
000 = 1:1
- bit 7-0      **Unimplemented:** Read as '0'

**Note 1:** This bit is automatically cleared when the ROI bit is set and an interrupt occurs.

**2:** 8 MHz FRC is postscaled when OSCCON (COSC<2:0>) = 111, and 500 kHz Low-Power FRC (LPFRC) is postscaled when OSCCON (COSC<2:0>) = 110.

## 38.4.3 Oscillator Tuning Register (OSCTUN)

The FRC Oscillator Tuning register (Register 38-3) allows the user to fine tune the FRC and LPFRC oscillators over a range of approximately  $\pm 12\%$ .

The tuning response of the FRC oscillator may not be monotonic or linear; the next closest frequency may be offset by a number of steps. It is recommended that users try multiple values of OSCTUN to find the closest value to the desired frequency.

**Register 38-3: OSCTUN: FRC Oscillator Tuning Register**

|        |     |     |     |     |     |     |       |
|--------|-----|-----|-----|-----|-----|-----|-------|
| U-0    | U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | U-0   |
| —      | —   | —   | —   | —   | —   | —   | —     |
| bit 15 |     |     |     |     |     |     | bit 8 |

|       |     |       |       |       |       |       |       |
|-------|-----|-------|-------|-------|-------|-------|-------|
| U-0   | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| —     | —   | TUN5  | TUN4  | TUN3  | TUN2  | TUN1  | TUN0  |
| bit 7 |     |       |       |       |       |       | bit 0 |

### Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at all Resets

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

bit 15-6

**Unimplemented:** Read as '0'

bit 5-0

**TUN<5:0>:** FRC Oscillator Tuning bits

011111 = Maximum frequency deviation

011110 =

.

.

.

000001 =

000000 = Center frequency, oscillator is running at factory calibrated frequency

111111 =

.

.

.

100001 =

100000 = Minimum frequency deviation

## Section 38. Oscillator with 500 kHz Low-Power FRC

### 38.4.4 Reference Clock Output Control Register (REFOCON)

For PIC24F devices that include a reference clock output generator, the REFOCON register (Register 38-4) controls the operations of this feature.

Setting the ROEN bit (REFOCON<15>) makes the clock signal available on the REFO pin. The RODIV bits (REFOCON<11:8>) select one of 16 different clock divider options.

The ROSSLP and ROSEL bits (REFOCON<13:12>) control the clock source for the reference output. The ROSEL bit determines if the primary oscillator on OSC1 and OSC2, or the current system clock source, provides the reference clock output. If the primary oscillator is selected, the ROSSLP bit determines its availability in Sleep mode.

Additional information is provided in **Section 38.13 “Reference Clock Output Generator”**.

**Register 38-4: REFOCON: Reference Oscillator Control Register**

|        |     |        |       |        |        |        |        |
|--------|-----|--------|-------|--------|--------|--------|--------|
| R/W-0  | U-0 | R/W-0  | R/W-0 | R/W-0  | R/W-0  | R/W-0  | R/W-0  |
| ROEN   | —   | ROSSLP | ROSEL | RODIV3 | RODIV2 | RODIV1 | RODIV0 |
| bit 15 |     |        |       |        |        |        | bit 8  |
| U-0    | U-0 | U-0    | U-0   | U-0    | U-0    | U-0    | U-0    |
| —      | —   | —      | —     | —      | —      | —      | —      |
| bit 7  |     |        |       |        |        |        | bit 0  |

**Legend:**

R = Readable bit  
-n = Value at POR

W = Writable bit  
'1' = Bit is set

U = Unimplemented bit, read as '0'  
'0' = Bit is cleared  
x = Bit is unknown

- bit 15     **ROEN:** Reference Oscillator Output Enable bit  
          1 = Reference oscillator enabled on REFO pin  
          0 = Reference oscillator disabled
- bit 14     **Unimplemented:** Read as '0'
- bit 13     **ROSSLP:** Reference Oscillator Output Stop in Sleep bit  
          1 = Reference oscillator continues to run in Sleep  
          0 = Reference oscillator is disabled in Sleep
- bit 12     **ROSEL:** Reference Oscillator Source Select bit  
          1 = Primary oscillator used as the base clock. Note that the crystal oscillator must be enabled using the FOSC<2:0> bits; crystal maintains the operation in Sleep mode  
          0 = System clock used as the base clock; base clock reflects any clock switching of the device
- bit 11-8   **RODIV<3:0>:** Reference Oscillator Divisor Select bits  
          1111 = Base clock value divided by 32,768  
          1110 = Base clock value divided by 16,384  
          1101 = Base clock value divided by 8,192  
          1100 = Base clock value divided by 4,096  
          1011 = Base clock value divided by 2,048  
          1010 = Base clock value divided by 1,024  
          1001 = Base clock value divided by 512  
          1000 = Base clock value divided by 256  
          0111 = Base clock value divided by 128  
          0110 = Base clock value divided by 64  
          0101 = Base clock value divided by 32  
          0100 = Base clock value divided by 16  
          0011 = Base clock value divided by 8  
          0010 = Base clock value divided by 4  
          0001 = Base clock value divided by 2  
          0000 = Base clock value
- bit 7-0    **Unimplemented:** Read as '0'

**38**

Oscillator with 500 kHz  
Low-Power FRC

## 38.5 PRIMARY OSCILLATOR (POSC)

The primary oscillator is available on the OSC1 and OSC2 pins of the PIC24F family. In general, the primary oscillator can be configured for an external clock input or an external crystal. Further details of the primary oscillator operating modes are described in subsequent sections. The primary oscillator has up to six operating modes, listed in Table 38-2.

**Table 38-2: Primary Oscillator Operating Modes**

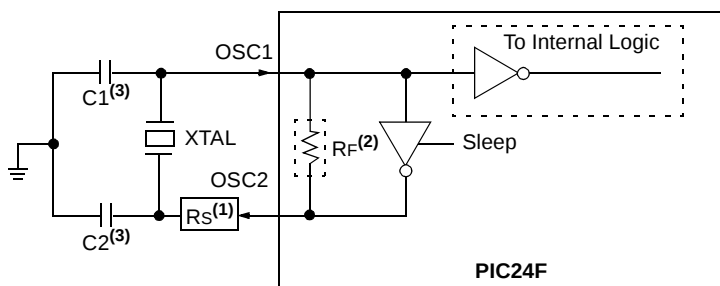
| Oscillator Mode          | Description                                                | OSC2 Pin Function |
|--------------------------|------------------------------------------------------------|-------------------|
| EC                       | External clock input (0 MHz-32 MHz)                        | Fosc/2            |
| ECPLL                    | External clock input (4 MHz-8 MHz), PLL enabled            | Fosc/2            |
| HS <sup>(1,3)</sup>      | 4 MHz-25 MHz crystal                                       | —                 |
| HSPLL <sup>(1,2)</sup>   | 4 MHz-8 MHz crystal (the PLL output will be 16 MHz-32 MHz) | —                 |
| XT <sup>(1,3)</sup>      | 200 kHz-4 MHz crystal                                      | —                 |
| XTPLL <sup>(1,2,3)</sup> | 3.5 MHz-4 MHz crystal, PLL enabled                         | —                 |

- Note 1:** External crystal connected to OSC1 and OSC2 in these modes.
- 2:** Available only in devices with specialty PLL blocks (such as the USB PLL); the basic 4x PLL block generates clock frequencies beyond the device's operating range.
- 3:** Frequency range depends on the device. Refer to the product data sheet for available range.

POSCMD<1:0> and FNOSC<2:0> Configuration bits select the operating mode of the primary oscillator. The POSCMD<1:0> bits select the particular submode to be used (XT, HS or EC) while the FNOSC<2:0> bits determine if the oscillator will be used by itself or with the internal PLL. The PIC24F operates from the primary oscillator whenever the COSC bits (OSCCON<14:12>) are set to '010' or '011'.

Refer to the “**Electrical Characteristics**” section in the specific device data sheet for further information regarding frequency range for each crystal mode.

**Figure 38-3: Crystal or Ceramic Resonator Operation (XT or HS Oscillator Mode)**



- Note 1:** A series resistor  $R_S$  may be required for AT strip cut crystals.
- 2:** The internal feedback resistor,  $R_F$ , is typically in the range of 2 to 10 M $\Omega$ .
- 3:** See **Section 38.5.2.4 “Determining the Best Values for Oscillator Components”**.

### 38.5.1 Selecting a Primary Oscillator Mode

The main difference between the XT and HS modes is the gain of the internal inverter of the oscillator circuit, which allows the different frequency ranges. The XT mode is a medium power and medium frequency mode. HS mode provides the highest oscillator frequencies with a crystal. OSC2 provides crystal feedback in both HS and XT Oscillator modes.

The EC and HS modes that use the PLL circuit provide the highest device operating frequencies. The oscillator circuit will consume the most current in these modes because the PLL is enabled to multiply the frequency of the oscillator by 4.

In general, users should select the oscillator option with the lowest possible gain that still meets their specifications. This will result in lower dynamic currents ( $I_{DD}$ ). The frequency range of each oscillator mode is the recommended frequency cutoff, but the selection of a different gain mode is acceptable as long as thorough validation has been performed (voltage, temperature and component variations, such as resistor, capacitor and internal oscillator circuitry).

The oscillator feedback circuit is disabled in all EC modes. The OSC1 pin is a high-impedance input and can be driven by a CMOS driver.

If the primary oscillator is configured for an external clock input, the OSC2 pin is not required to support the oscillator function. For these modes, the OSC2 pin can be used as an additional device I/O pin or a clock output pin. When the OSC2 pin is used as a clock output pin, the output frequency is  $F_{osc}/2$ .

### 38.5.2 Crystal Oscillators and Ceramic Resonators

In XT and HS modes, a crystal or ceramic resonator is connected to the OSC1 and OSC2 pins to establish oscillation (Figure 38-3). The PIC24F oscillator design requires the use of a parallel cut crystal. Using a series cut crystal may give a frequency out of the crystal manufacturer's specifications.

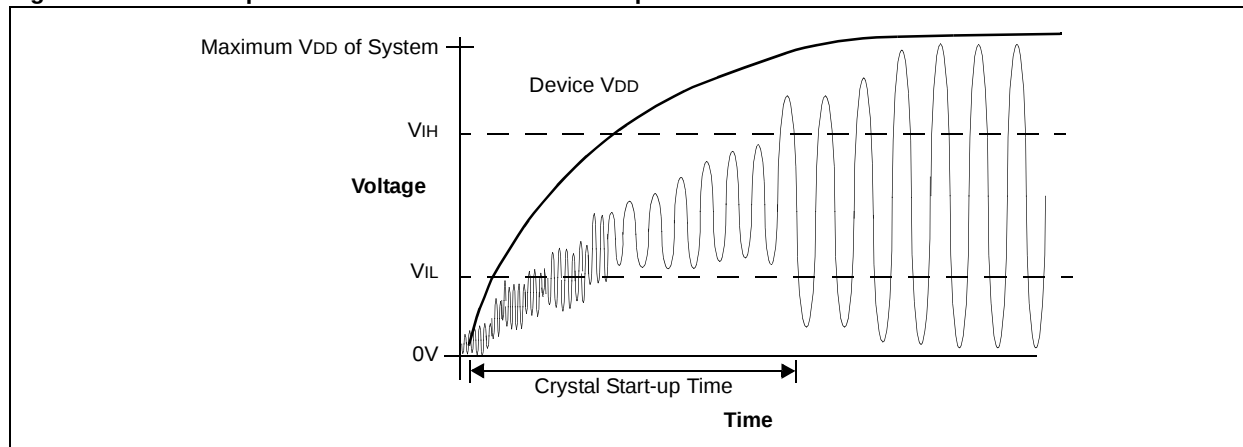
#### 38.5.2.1 OSCILLATOR/RESONATOR START-UP

As the device voltage increases from  $V_{SS}$ , the oscillator will start its oscillations. The time required for the oscillator to start oscillating depends on many factors, including:

- Crystal/resonator frequency
- Capacitor values used
- Series resistor, if used, and its value and type
- Device  $V_{DD}$  rise time
- System temperature
- Oscillator mode selection of device (selects the gain of the internal oscillator inverter)
- Crystal quality
- Oscillator circuit layout
- System noise

The course of a typical crystal or resonator start-up is displayed in Figure 38-4. Notice that the time to achieve stable oscillation is not instantaneous.

**Figure 38-4: Example of Oscillator/Resonator Start-up Characteristics**



#### 38.5.2.1.1 Primary Oscillator Start-up from Sleep Mode

The most difficult time for the oscillator to start-up is when waking up from Sleep mode. This is because the load capacitors have both partially charged to some quiescent value and phase differential at wake-up is minimal. Thus, more time is required to achieve stable oscillation. Also remember that low voltage, high temperatures and the lower frequency clock modes also impose limitations on loop gain, which in turn, affects the start-up.

Each of the following factors increases the start-up time:

- Low-frequency design (with a Low Gain Clock mode)
- Quiet environment (such as a battery-operated device)
- Operating in a shielded box (away from the noisy RF area)
- Low voltage
- High temperature
- Wake-up from Sleep mode

Circuit noise, on the other hand, may actually help to “kick start” the oscillator and help to lower the oscillator start-up time.

#### 38.5.2.2 OSCILLATOR START-UP TIMER

In order to ensure that a crystal oscillator (or ceramic resonator) has started and stabilized, an Oscillator Start-up Timer (OST) is provided. The OST is a simple, 10-bit counter that counts 1024  $T_{osc}$  cycles before releasing the oscillator clock to the rest of the system. This time-out period is designated as  $T_{OST}$ . The amplitude of the oscillator signal must reach the  $V_{IL}$  and  $V_{IH}$  thresholds for the oscillator pins before the OST can begin to count cycles.

The  $T_{OST}$  interval is required every time the oscillator has to restart (i.e., on POR, Brown-out Reset (BOR) and wake-up from Sleep mode). The OST is applied to the XT and HS modes for the primary oscillator, and the Secondary Oscillator (SOSC). See **Section 38.7 “Secondary Oscillator (SOSC)”** for more information.

### 38.5.2.3 TUNING THE OSCILLATOR CIRCUIT

Since Microchip devices have wide operating ranges (frequency, voltage and temperature; depending on the part and version ordered) and external components (crystals, capacitors, etc.) of varying quality and manufacture, validation of operation needs to be performed to ensure that the component selection will comply with the requirements of the application. There are many factors that go into the selection and arrangement of these external components. Depending on the application, these may include any of the following:

- Amplifier gain
- Desired frequency
- Resonant frequency(s) of the crystal
- Temperature of operation
- Supply voltage range
- Start-up time
- Stability
- Crystal life
- Power consumption
- Simplification of the circuit
- Use of standard components
- Component count

### 38.5.2.4 DETERMINING THE BEST VALUES FOR OSCILLATOR COMPONENTS

The best method for selecting components is to apply a little knowledge and a lot of trial measurement and testing. Crystals are usually selected by their parallel resonant frequency only. However, other parameters may be important to your design, such as temperature or frequency tolerance. Microchip application note AN588, “PICmicro® Microcontroller Oscillator Design Guide” is an excellent reference to learn more about crystal operation and ordering information.

The PIC24F internal oscillator circuit is a parallel oscillator circuit, which requires that a parallel resonant crystal be selected. The load capacitance is usually specified in the 22 pF to 33 pF range. The crystal will oscillate closest to the desired frequency with a load capacitance in this range. It may be necessary to alter these values, as described later, in order to achieve other benefits.

The clock mode is primarily chosen based on the desired frequency of the crystal oscillator. The main difference between the XT and HS Oscillator modes is the gain of the internal inverter of the oscillator circuit which allows the different frequency ranges. In general, use the oscillator option with the lowest possible gain that still meets the specifications; this will result in lower dynamic currents ( $I_{DD}$ ). The frequency range of each oscillator mode is the recommended frequency cutoff, but the selection of a different gain mode is acceptable as long as thorough validation is performed (voltage, temperature and component variations, such as resistor, capacitor and internal oscillator circuitry). C1 and C2 should also be initially selected based on the load capacitance, as recommended by the crystal manufacturer and the tables provided in the device data sheet. The values given in the device data sheet can only be used as a starting point since the crystal manufacturer, supply voltage and other factors already mentioned may cause your circuit to differ from the one used in the factory characterization process.

Ideally, the capacitance is chosen so that it will oscillate at the highest temperature and the lowest  $V_{DD}$  that the circuit will be expected to perform under. High temperature and low  $V_{DD}$  both have a limiting effect on the loop gain, such that if the circuit functions at these extremes, the designer can be more assured of proper operation at other temperatures and supply voltage combinations. The output sine wave should not be clipped in the highest gain environment (highest  $V_{DD}$  and lowest temperature) and the sine output amplitude should be large enough in the lowest gain environment (lowest  $V_{DD}$  and highest temperature) to cover the logic input requirements of the clock as listed in the device data sheet. OSC1 may have specified  $V_{IL}$  and  $V_{IH}$  Levels (refer to the specific product data sheet for more information).

A method for improving start-up is to use a value of C2 greater than C1. This causes a greater phase shift across the crystal at power-up, which speeds the oscillator start-up. Besides loading the crystal for proper frequency response, these capacitors can have the effect of lowering loop gain if their value is increased. C2 can be selected to affect the overall gain of the circuit. A higher C2 can lower the gain if the crystal is being overdriven (also, see discussion on Rs). Capacitance values that are too high can store and dump too much current through the crystal, hence, C1 and C2 should not become excessively large. Unfortunately, measuring the wattage through a crystal is difficult, but if you do not stray too far from the suggested values, you should not be concerned with this.

After all the other external components are selected to satisfaction, a series resistor, Rs, is added to the circuit if the crystal is still being overdriven. This can be determined by looking at the OSC2 pin, which is the driven pin with an oscilloscope. Connecting the probe to the OSC1 pin will load the pin too much and negatively affect performance. Remember that a scope probe adds its own capacitance to the circuit, hence, this may have to be accounted for in your design (i.e., if the circuit worked best with a C2 of 22 pF and the scope probe was 10 pF, a 33 pF capacitor may actually be called for). The output signal should not be clipping or flattened. Overdriving the crystal can also lead to the circuit jumping to a higher harmonic level, or even, crystal damage.

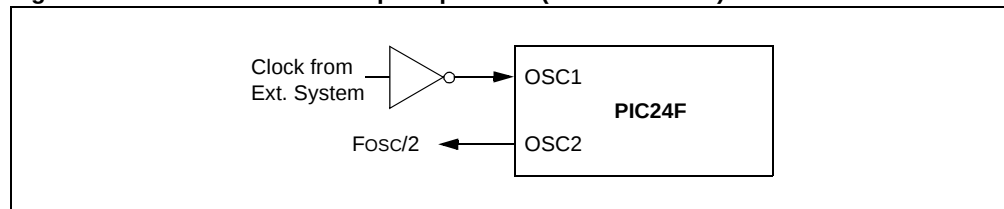
The OSC2 signal should be a clean sine wave that easily spans the input minimum and maximum of the clock input pin. An easy way to set this is to again test the circuit at the minimum temperature and maximum VDD that the design will be expected to perform in, then look at the output. This should be the maximum amplitude of the clock output. If there is clipping, or the sine wave is distorted near VDD and VSS, increasing load capacitors may cause too much current to flow through the crystal or push the value too far from the manufacturer's load specification. To adjust the crystal current, add a trimmer potentiometer between the crystal inverter output pin and C2, and adjust it until the sine wave is clean. The crystal will experience the highest drive currents at the low temperature and high VDD extremes.

The trimmer potentiometer should be adjusted at these limits to prevent overdriving. A series resistor, Rs, of the closest standard value can now be inserted in place of the trimmer. If Rs is too high (more than 20 k $\Omega$ ), the input will be too isolated from the output making the clock more susceptible to noise. If you find a value this high is required to prevent overdriving the crystal, try increasing C2 to compensate or changing the oscillator operating mode. Try to get a combination where Rs is around 10 k $\Omega$  or less, and the load capacitance is not too far from the manufacturer's specification.

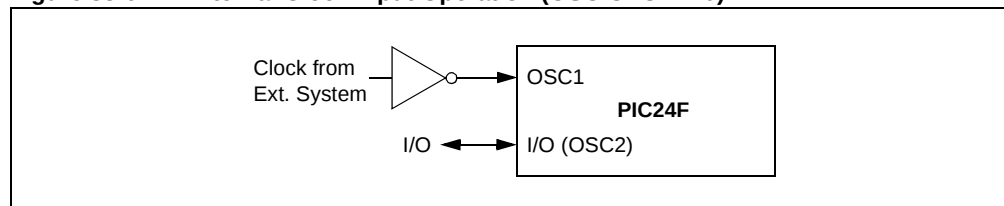
### 38.5.3 External Clock Input

In the EC mode, the OSC1 pin is in a high-impedance state and can be driven by CMOS drivers. The OSC2 pin can be configured as either an I/O or the clock output (Fosc/2) by selecting the OSCIOFCN bit (Configuration Word FOSC<2>). With OSCIOFCN set (see Figure 38-5), the clock output is available for testing or synchronization purposes. With OSCIOFCN clear (see Figure 38-6), the OSC2 pin becomes a general purpose I/O pin. The feedback device between OSC1 and OSC2 is turned off to save current.

**Figure 38-5: External Clock Input Operation (OSCIOFCN = 1)**



**Figure 38-6: External Clock Input Operation (OSCIOFCN = 0)**



## 38.6 PHASE LOCKED LOOP (PLL) BRANCH

The system clock for all the PIC24F devices includes a frequency multiplier branch built around a Phase Locked Loop (PLL). This branch allows the user to obtain a higher clock speed using a low-speed primary oscillator or external clock source, eliminating the need for an expensive high-speed crystal or resonator. It also allows the use of the internal FRC oscillator (FRC) to clock the device at its maximum operating speed without the use of an external oscillator.

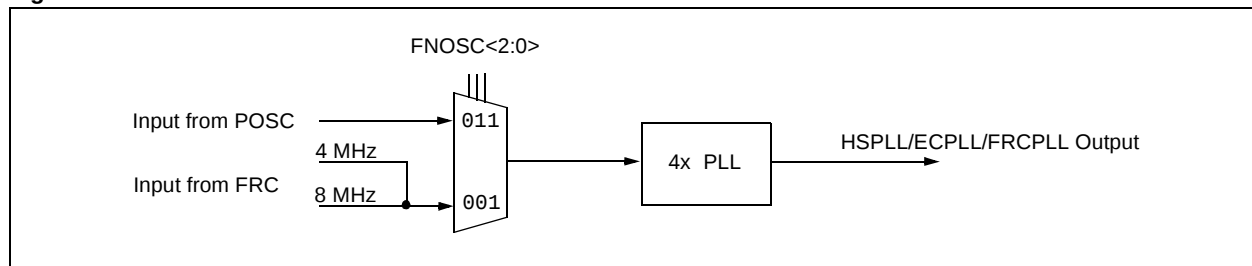
There are several different versions of the PLL block implemented on PIC24F devices; the version implemented depends on the particular device family. Some PLL blocks provide a single branch, frequency multiplied output. Others provide multiple branches with clocks at different frequencies, including a special clock for a particular peripheral. Refer to the particular device data sheet to see which PLL block is implemented.

### 38.6.1 Basic 4x PLL Block

In most PIC24F devices, the implemented PLL block is the basic PLL (see Figure 38-7). This provides a fixed 4x multiplier, which can be used with XT and EC primary oscillators and the FRC oscillator. The PLL accepts any frequency input from approximately 3.5 MHz to 8 MHz.

Whenever the clock source of the PLL is changed, the PLL ready timer must be reset to allow the PLL to synchronize to the new clock source. After the ready timer has counted the required time, the PLL output is ready for use.

Figure 38-7: Basic 4x PLL Block



### 38.6.2 Considerations for Using the PLL Block

All PLL blocks use the LOCK bit (OSCCON<5>) as a read-only status bit to indicate the lock status of the PLL. It is automatically set after the typical time delay for the PLL to achieve lock, designated as T<sub>LOCK</sub>. It is cleared at a POR and on clock switches when the PLL is selected as a destination clock source. It remains clear when any clock source not using the PLL is selected.

If the PLL does not stabilize properly during start-up, LOCK may not reflect the actual status of the PLL lock, and it does not detect when the PLL loses lock during normal operation. Refer to the “**Electrical Characteristics**” section in the specific device data sheet for further information on the PLL lock interval.

Using any PLL block with the FRC oscillator provides a stable system clock for microcontroller operations. In specific devices or families, this combination may not meet the frequency accuracy requirements for use in synchronous communications. This is particularly true with serial communications using the UARTs. Refer to the “**Electrical Characteristics**” section of the particular device data sheet for specific information.

If an application is being migrated between two PIC24F platforms with different PLL blocks (e.g., from a GA0 family device to a GB1 family device), the differences in PLL and clock options may require the reconfiguration of peripherals that use the system clock. This is particularly true with serial communications’ peripherals, such as the UARTs.

## 38.7 SECONDARY OSCILLATOR (SOSC)

The low-power Secondary Oscillator (SOSC) is designed specifically for low-power operation with a 32.768 kHz crystal. The oscillator is located on the SOSCO and SOSCI device pins and serves as a secondary crystal clock source for low-power operation. It can also drive Timer1 or the Real-Time Clock and Calendar (RTCC) module for Real-Time Clock (RTC) applications.

There are two modes of secondary oscillators:

1. Secondary oscillator, which takes more power but is easier to start-up and less susceptible to noise.
2. Low-power secondary oscillator, which takes less power but is more susceptible to noise and requires careful consideration for circuit layout and component selection.

Secondary and low-power secondary oscillators can be selected by using the SOSCSEL bit.

### 38.7.1 Enabling the Secondary Oscillator

The operation of SOSC is selected by the FNOSC Configuration bits and is further controlled by the SOSCEN bit (OSCCON<1>). Setting SOSCEN enables the oscillator; the SOSCO and SOSCI pins are controlled by the oscillator and cannot be used for port I/O or other functions.

|                                                                                                                                                                       |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Note:</b> An unlock sequence is required before a write to OSCCON can occur. Refer to <b>Section 38.11.2 “Oscillator Switching Sequence”</b> for more information. |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|

### 38.7.2 Secondary Oscillator Continuous Operation

The SOSC is always enabled when SOSCEN is set. Leaving the oscillator running always allows a fast switch to the 32 kHz system clock for low-power operation. Returning to the faster main oscillator will still require an oscillator start-up time if it is a crystal type source (see **Section 38.5.2.2 “Oscillator Start-up Timer”**).

Additionally, the oscillator will always need to remain running for the Real-Time Clock application using Timer1 or the RTCC module. Refer to the *“PIC24F Family Reference Manual”*, **Section 14. “Timers”** and **Section 29. “Real-Time Clock and Calendar (RTCC)”** for further details.

### 38.7.3 Secondary Oscillator Intermittent Operation

When SOSCEN is cleared, the oscillator will only operate when it is selected as the current device clock source (COSC<2:0> = 100). It will be disabled automatically if it is the current device clock source and the device enters Sleep mode.

## 38.8 INTERNAL FAST RC OSCILLATOR (FRC)

The FRC oscillator is a fast (8 MHz nominal), internal RC oscillator. This oscillator is intended to provide reasonable device operating speed without the use of an external crystal or ceramic resonator. The PIC24F operates from the FRC oscillator whenever the COSC bits are '111', '001' or '000'.

|                                                                                                                                                                                                                              |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Note:</b> Due to specified FRC accuracy, the FRC Oscillator modes may not meet the minimum frequency accuracy requirements for serial communications (such as UART and USB). Refer to the product data sheet for details. |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

### 38.8.1 Enabling the FRC Oscillator

Since it serves as the reference clock during device initialization, the FRC oscillator is always enabled at a POR. After the device is configured and PWRT expires, FRC remains active only if it is selected as the device clock source.

### 38.8.2 500 kHz Low-Power FRC Mode

LPFRC mode is selected whenever the COSC bits are '110'. In this mode, the system clock is running at 500 kHz, provided the FRCDIV selection is 1:1.

### 38.8.3 FRC Postscaler Mode (FRCDIV)

Users are not limited to the nominal 8 MHz/500 kHz FRC output if they wish to use the fast internal oscillator as a clock source. An additional FRC mode, FRCDIV, implements a selectable postscaler that allows the choice of a lower clock frequency from eight different options, plus the direct 8 MHz/500 kHz output. The postscaler is configured using bits, CLKDIV<10:8>. Assuming a nominal 8 MHz/500 kHz output, available lower frequency options range from 8 MHz direct to 31 kHz (divide-by-256). The range of frequencies allows users the ability to save power at any time in an application by simply changing the RCDIV bits.

The FRCDIV mode is selected whenever the COSC bits are '111'.

### 38.8.4 FRC Oscillator with PLL Mode (FRCPLL)

The FRCPLL mode is selected whenever the COSC bits are '001'. This mode only functions when the direct or divide-by-2 FRC postscaler options are selected (RCDIV<2:0> = 000 or 001).

For devices with the basic 4x PLL block, the output of the FRC postscaler block may also be combined with the PLL to produce a nominal system clock of either 16 MHz or 32 MHz. Although somewhat less precise in frequency than using the primary oscillator with a crystal or resonator, it still allows high-speed operation of the device without the use of external oscillator components.

## 38.9 INTERNAL LOW-POWER RC OSCILLATOR (LPRC)

The LPRC oscillator is separate from the FRC and oscillates at a nominal frequency of 31 kHz. LPRC is the clock source for the Power-up Timer (PWRT), WDT and FSCM circuits. It may also be used to provide a low-frequency clock source option for the device in those applications where power consumption is critical, and timing accuracy is not required.

### 38.9.1 Enabling the LPRC Oscillator

Since it serves the PWRT clock source, the LPRC oscillator is enabled at POR whenever the on-board voltage regulator is disabled. After the PWRT expires, the LPRC oscillator will remain on if any one of the following is true:

- The FSCM is enabled
- The WDT is enabled
- The LPRC Oscillator is selected as the system clock (COSC<2:0> = 101)

If none of the above is true, the LPRC will shut off after the PWRT expires.

## 38.10 FAIL-SAFE CLOCK MONITOR (FSCM)

The Fail-Safe Clock Monitor (FSCM) allows the device to continue to operate even in the event of an oscillator failure. The FSCM function is enabled by programming the Clock Switch and Monitor (FCKSM) bits in the Configuration Word. FSCM is only enabled when FCKSM<1:0> are programmed ('00'). When FSCM is enabled, the LPRC internal oscillator will run at all times (except during Sleep mode).

In the event of an oscillator failure, the FSCM will generate a clock failure trap and will switch the system clock to the FRC oscillator. The user will then have the option to either attempt to restart the oscillator or execute a controlled shutdown. FSCM will monitor the system clock source regardless of its source or oscillator mode. This includes the primary oscillator for all oscillator modes and the Secondary Oscillator (SOSC) when configured as the system clock.

The FSCM module performs the following actions when switching to the FRC oscillator:

1. The COSC bits are loaded with '000'.
2. The CF status bit is set to indicate the clock failure.
3. The OSWEN control bit is cleared to cancel any pending clock switches.

**Note:** For more information about the oscillator failure trap, refer to the "PIC24F Family Reference Manual", Section 8. "Interrupts".

## 38.10.1 FSCM Delay

On a POR, BOR or wake from Sleep mode event, a nominal delay (T<sub>FSCM</sub>) may be inserted before the FSCM begins to monitor the system clock source. The purpose of the FSCM delay is to provide time for the oscillator and/or PLL to stabilize when the Power-up Timer (PWRT) is not utilized. The FSCM delay will be generated after the internal System Reset signal, SYSRST, has been released.

The T<sub>FSCM</sub> interval is applied whenever the FSCM is enabled and the EC, HS or SOSC Oscillator modes are selected as the system clock.

**Note:** Please refer to the “**Electrical Characteristics**” section of the specific device data sheet for T<sub>FSCM</sub> specification values.

## 38.10.2 FSCM and Slow Oscillator Start-up

If the chosen device oscillator has a slow start-up time coming out of POR, BOR or Sleep mode, it is possible that the FSCM delay will expire before the oscillator has started. In this case, the FSCM will initiate a clock failure trap. As this happens, the COSC bits are loaded with the FRC oscillator selection. This will effectively shut off the original oscillator that was trying to start. The user can detect this situation and initiate a clock switch back to the desired oscillator in the Trap Service Routine (TSR).

## 38.10.3 FSCM and WDT

Both the FSCM and WDT use the LPRC oscillator as their time base. In the event of a clock failure, the WDT is unaffected and continues to run on the LPRC.

## 38.11 CLOCK SWITCHING OPERATION

With few limitations, applications are free to switch between any of the five clock sources (Primary, SOSC, FRC, LPFRC and LPRC) under software control and at any time. To limit the possible side effects that could result from this flexibility, PIC24F devices have a safeguard lock built into the switch process.

**Note:** The Primary Oscillator mode has three different submodes (XT, HS and EC), which are determined by the POSCMD Configuration bits. While an application can switch to and from Primary Oscillator mode in software, it cannot switch between the different primary submodes without reprogramming the device.

### 38.11.1 Enabling Clock Switching

To enable clock switching, the FCKSM1 Configuration bit must be programmed to '0'. (Refer to the specific device data sheet for further details.) If the FCKSM1 Configuration bit is unprogrammed ('1'), the clock switching function and FSCM function are disabled. This is the default setting.

The NOSC control bits (OSCCON<10:8>) do not control the clock selection when clock switching is disabled. However, the COSC bits (OSCCON<14:12>) will reflect the clock source selected by the FNOSC Configuration bits.

The OSWEN Control bit (OSCCON<0>) has no effect when clock switching is disabled; it is always held at '0'.

## 38.11.2 Oscillator Switching Sequence

At a minimum, performing a clock switch requires this basic sequence:

1. If required, read the COSC bits (OSCCON<14:12>) to determine the current oscillator source.
2. Perform the unlock sequence to allow a write to the OSCCON register high byte.
3. Write the appropriate value to the NOSC control bits (OSCCON<10:8>) for the new oscillator source.
4. Perform the unlock sequence to allow a write to the OSCCON register low byte.
5. Set the OSWEN bit to initiate the oscillator switch.

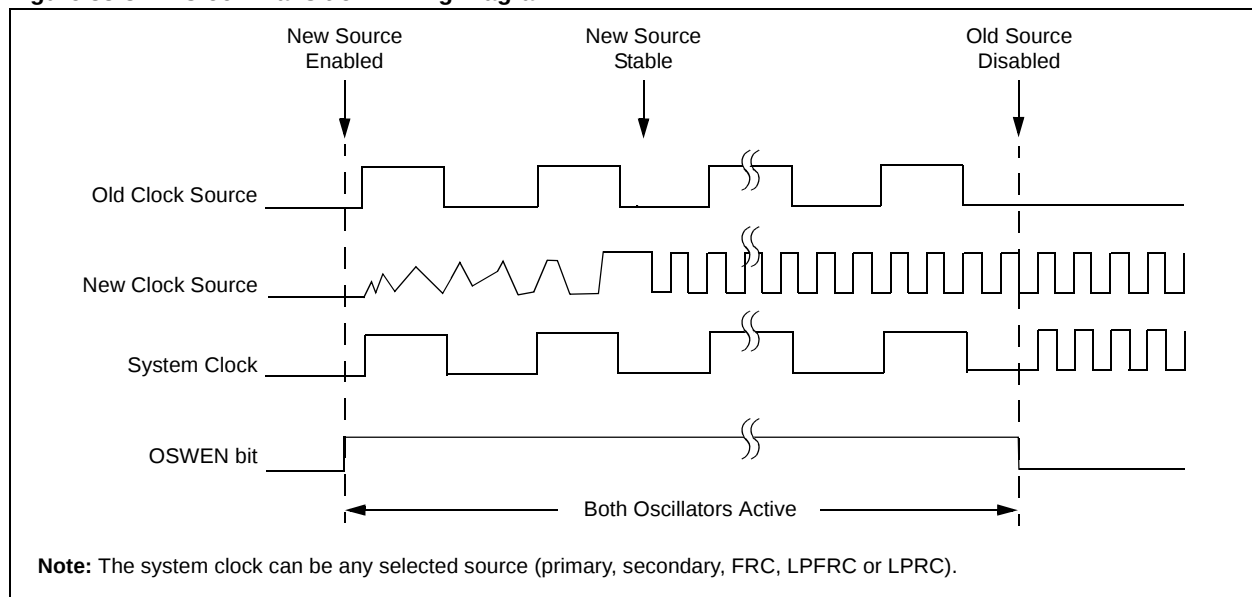
Once the basic sequence is completed, the system clock hardware responds automatically as follows:

1. The clock switching hardware compares the COSC status bits with the new value of the NOSC control bits. If they are the same, then the clock switch is a redundant operation. In this case, the OSWEN bit is cleared automatically and the clock switch is aborted.
2. If a valid clock switch has been initiated, the LOCK (OSCCON<5>) and the CF (OSCCON<3>) status bits are cleared.
3. The new oscillator is turned on by the hardware if it is not currently running. If a crystal oscillator must be turned on, the hardware will wait until the OST expires. If the new source is using the PLL, then the hardware waits until a PLL lock is detected (LOCK = 1).
4. The hardware waits for the new clock source to stabilize and then performs the clock switch.
5. The hardware clears the OSWEN bit to indicate a successful clock transition. Additionally, the NOSC bit values are transferred to the COSC status bits.
6. The old clock source is turned off at this time, with the exception of LPRC (if WDT or FSCM are enabled) or SOSC (if SOSSEN remains set).

Figure 38-8 displays the timing of the transition between clock sources.

**Note:** The processor will continue to execute code throughout the clock switching sequence. Timing-sensitive code should not be executed during this time.

**Figure 38-8: Clock Transition Timing Diagram**



A recommended code sequence for a clock switch includes the following:

1. Disable interrupts during the OSCCON register unlock and write sequence.
2. Execute the unlock sequence for the OSCCON high byte by writing 78h and 9Ah to OSCCON<15:8> in two back-to-back instructions.
3. Write new oscillator source to NOSC control bits in the instruction immediately following the unlock sequence.
4. Execute the unlock sequence for the OSCCON low byte by writing 46h and 57h to OSCCON<7:0> in two back-to-back instructions.
5. Set the OSWEN bit in the instruction immediately following the unlock sequence.
6. Continue to execute code that is not clock-sensitive (optional).
7. Invoke an appropriate amount of software delay (cycle counting) to allow the selected oscillator and/or PLL to start and stabilize.
8. Check to see if OSWEN is '0'; If it is, the switch was successful.

The core sequence for unlocking the OSCCON register and initiating a clock switch is illustrated in Example 38-1.

#### Example 38-1: Basic Code Sequence for Clock Switching in Assembly Language

```
;Place the new oscillator selection in w0
;OSCCONH (high byte) Unlock Sequence
MOV      #OSCCONH,w1
MOV      #0x78, w2
MOV      #0x9A, w3
MOV.b    w2, [w1]
MOV.b    w3, [w1]
;Set new oscillator selection
MOV.b    WREG, OSCCONH
;OSCCONL (low byte) unlock sequence
MOV      #OSCCONL,w1
MOV.b    #0x01, w0
MOV      #0x46, w2
MOV      #0x57, w3
MOV.b    w2, [w1]
MOV.b    w3, [w1]
;Start oscillator switch operation
MOV.b    w0, [w1]
```

#### Example 38-2: Basic Code Sequence for Clock Switching in C Language

```
__builtin_write_OSCCONH(NewOscillatorValue);
__builtin_write_OSCCONL(NewOscillatorValue);
OSCCONbits.OSWEN =1;
```

### 38.11.2.1 CLOCK SWITCHING CONSIDERATIONS

When incorporating clock switching into an application, users should pay attention to the following points when designing their code:

- The OSCCON unlock sequence is extremely timing critical. The OSCCON register byte is only writable for one instruction cycle following the sequence. Some high-level languages, such as C, may not preserve the timing-sensitive sequence of instructions when compiled. When clock switching is required for an application written in a high-level language, it is best to create the routine in assembler and link it to the application, calling it as a function when it is required or use a built-in function if available.
- If the destination clock source is a crystal oscillator, the clock switch time will be dominated by the oscillator start-up time.
- If the new clock source does not start, or is not present, the clock switching hardware will wait indefinitely for the new clock source. The user can detect this situation because the OSWEN bit remains set indefinitely.
- If the new clock source uses the PLL, a clock switch will not occur until the lock has been achieved. The user can detect a loss of PLL lock because the LOCK bit will be cleared and the OSWEN bit is set.
- Switching to a low-frequency clock source, such as the secondary oscillator, will result in very slow device operation.

**Note:** The application should not attempt to switch to a clock with a frequency lower than 100 kHz when the Fail-Safe Clock Monitor is enabled. Clock switching in these instances may generate a false oscillator fail trap and result in a switch to the 8 MHz FRC oscillator.

### 38.11.3 Aborting a Clock Switch

In the event the clock switch did not complete, the clock switch logic can be reset by clearing the OSWEN bit. This will abandon the clock switch process, stop and reset the OST (if applicable) and stop the PLL (if applicable). Example 38-3 displays a typical assembly code for aborting a clock switch.

A clock switch procedure can be aborted at any time. A clock switch that is already in progress can also be aborted by performing a second clock switch.

#### Example 38-3: Aborting a Clock Switch

```
MOV      #OSCCON,W1      ; pointer to OSCCON
MOV.b    #0x46,W2         ; first unlock code
MOV.b    #0x57,W3         ; second unlock code
MOV.b    W2, [W1]         ; write first unlock code
MOV.b    W3, [W1]         ; write second unlock code
BCLR     OSCCON,#OSWEN    ; ABORT the switch
```

### 38.11.4 Entering Sleep Mode During a Clock Switch

If the device enters Sleep mode during a clock switch operation, the clock switch operation is aborted. The processor keeps the old clock selection and the OSWEN bit is cleared. The PWSAV instruction is then executed normally.

## 38.12 TWO-SPEED START-UP

Two-Speed Start-up is an automatic clock switching feature, which is independent of the manually controlled clock switching, described in the preceding section. It helps to minimize the latency period from oscillator start-up to code execution by allowing the microcontroller to use the FRC oscillator as a clock source until the primary clock source is available. This feature is controlled by the IESO Configuration bit (FOSCSEL) and operates independently of the state of the FCKSM Configuration bits.

Two-Speed Start-up is particularly useful when an external oscillator, selected by the FNOSC Configuration bits and a crystal-based oscillator (either a primary or secondary oscillator), may have a longer start-up time. As an internal RC oscillator, the FRC clock source is available almost immediately following POR or device wake-up.

With Two-Speed Start-up, the device starts executing code on POR in its default oscillator configuration (FRC). It continues to operate in this mode until the external oscillator source, specified by the FNOSC Configuration bits, becomes stable. When it automatically switches to that source.

Two-Speed Start-up is used on wake-up from the power-saving Sleep mode. The device uses the FRC clock source until the selected primary clock is ready. It is not used in Idle mode as the device will be clocked by the currently selected clock source until the primary clock source becomes available.

### 38.12.1 Special Considerations for Using Two-Speed Start-up

While using the FRC oscillator in Two-Speed Start-up, the device still obeys the normal command sequences for entering power-saving modes including serial PWRSAV instructions. In practice, this means that the user code can change the NOSC<2:0> bit settings or issue PWRSAV#SLEEP instructions before the OST times out. This would allow an application to briefly wake-up, perform routine “housekeeping” tasks and return to Sleep before the device starts to operate from the external oscillator.

The user code can also check which clock source is currently providing the device clocking by checking the status of the COSC<2:0> bits against the NOSC<2:0> bits. If these two sets of bits match, the clock switch has been completed successfully and the device is running from the intended clock source; the primary oscillator is providing the clock. Otherwise, FRC is providing the clock during wake-up from Reset or Sleep mode.

## 38.13 REFERENCE CLOCK OUTPUT GENERATOR

The CLKO output available on OSC2 in select oscillator modes can be useful for synchronizing external logic to the microcontroller. However, it is limited in that it is only available in a few modes, and at only one frequency ( $F_{osc}/2$ ). For select PIC24F families, the reference clock output generator provides another option: a separate synchronous and programmable clock source to the REFO port pin.

The reference clock output generator receives inputs from both the primary oscillator and the currently selected system clock. This allows the user to select an output clock signal with a constant frequency, or an output clock that changes as the system clock changes during run time (e.g., on a change to or from a low-power mode). A 16-step divider allows for the selection of a wide range of system clock sub-multiples. The reference clock changes as the system clock frequency changes, which occurs in a glitchless fashion.

This reference clock output is controlled by the REFOCON register (see Register 38-4). Setting the ROEN bit (REFOCON<15>) makes the clock signal available on the REFO pin. The RODIV bits (REFOCON<11:8>) select a range of output frequencies from divide-by-1 through divide-by-32,768. This postscaler can be changed as needed at run time. It is recommended that the clock generator be disabled prior to changing the RODIV bits to ensure the smoothest speed transition.

The ROSEL bit (REFOCON<12>) determines if the primary oscillator, or the current system clock source designated by COSC<2:0>, provides the reference clock output. Clearing the ROSEL bit allows the reference output frequency to change as the system clock changes during any clock switches.

If one of the primary oscillator modes (EC, HS or XT) is used for the system clock, the ROSEL and ROSSLP bits determine if the reference clock output is available when the device is in Sleep mode. Both the ROSSLP and ROSEL bits must be set; otherwise, the oscillator on OSC1 and OSC2 will be powered down when the device enters Sleep mode.

## 38.14 AC ELECTRICAL SPECIFICATIONS

Figure 38-9: External Clock Timing

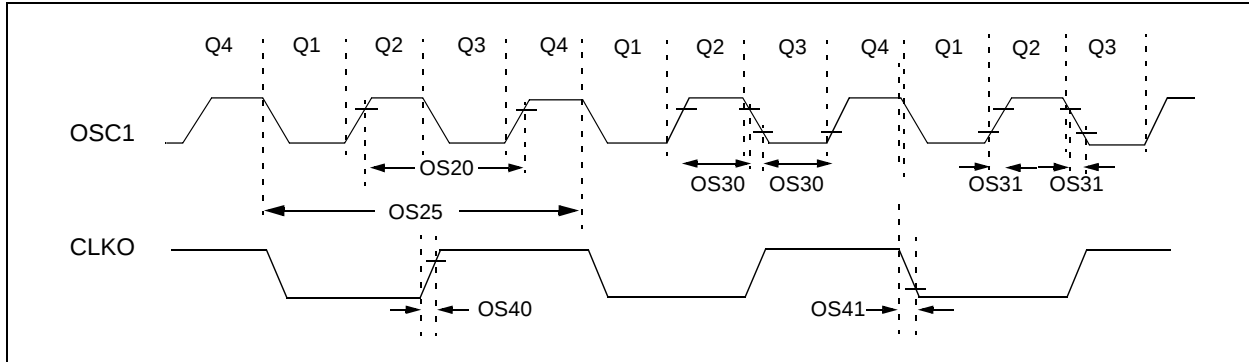


Table 38-3: External Clock Timing Requirements

| AC CHARACTERISTICS |               |                                               | Standard Operating Conditions: 2.0V to 3.6V (unless otherwise stated)                        |                    |     |       |            |
|--------------------|---------------|-----------------------------------------------|----------------------------------------------------------------------------------------------|--------------------|-----|-------|------------|
|                    |               |                                               | Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for Industrial |                    |     |       |            |
| Param No.          | Sym           | Characteristic                                | Min                                                                                          | Typ <sup>(1)</sup> | Max | Units | Conditions |
| OS20               | Tosc          | $T_{osc} = 1/F_{osc}$                         | —                                                                                            | —                  | —   | —     |            |
| OS25               | Tcy           | Instruction Cycle Time <sup>(2)</sup>         | 62.5                                                                                         | —                  | DC  | ns    |            |
| OS30               | TosL,<br>TosH | External Clock in (OSC1)<br>High or Low Time  | $0.45 \times T_{osc}$                                                                        | —                  | —   | ns    | EC         |
| OS31               | TosR,<br>TosF | External Clock in (OSC1)<br>Rise or Fall Time | —                                                                                            | —                  | 20  | ns    | EC         |
| OS40               | TckR          | CLKO Rise Time <sup>(3)</sup>                 | —                                                                                            | 6                  | 10  | ns    |            |
| OS41               | TckF          | CLKO Fall Time <sup>(3)</sup>                 | —                                                                                            | 6                  | 10  | ns    |            |

**Note 1:** Data in "Typ" column is at 3.3V, 25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

**2:** Instruction cycle period (Tcy) equals two times the input oscillator time base period. All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at "Min." values with an external clock applied to the OSC1/CLKI pin. When an external clock input is used, the "Max." cycle time limit is "DC" (no clock) for all devices.

**3:** Measurements are taken in EC mode. The CLKO signal is measured on the OSC2 pin. CLKO is low for the Q1-Q2 period (1/2 Tcy) and high for the Q3-Q4 period (1/2 Tcy).

## 38.15 DESIGN TIPS

**Question 1:** *When looking at the OSC2 pin after power-up with an oscilloscope, there is no clock. What can cause this?*

**Answer:** There are several possible causes:

1. Entering Sleep mode with no source for wake-up (such as WDT,  $\overline{\text{MCLR}}$  or an interrupt). Ensure that the code does not put the device to Sleep without providing for wake-up. If it is possible, try waking it up with a low pulse on MCLR. Powering up with MCLR held low will also give the crystal oscillator more time to start-up, but the program counter will not advance until the  $\overline{\text{MCLR}}$  pin is high.
2. The wrong clock mode is selected for the desired frequency. For a blank device, the default oscillator is FRCDIV. Most parts come with the clock selected in the Default mode, which will not start oscillation with a crystal or resonator. Verify if the clock mode has been programmed correctly.
3. The proper power-up sequence has not been followed. If a CMOS part is powered through an I/O pin prior to power-up, bad things can happen (latch-up, improper start-up, etc.). It is also possible for brown-out conditions, noisy power lines at start-up and slow VDD rise times to cause problems. Try powering up the device with nothing connected to the I/O, and power-up with a known, good, fast rise power supply. Refer to the power-up information in the specific device data sheet for considerations on brown-out and power-up sequences.
4. The C1 and C2 capacitors attached to the crystal have not been connected properly or are not the correct values. Ensure that all the connections are correct. The device data sheet values for these components will usually get the oscillator running. However, they might just not be the optimal values for your design.
5. For the non-crystal oscillator section, the OSC2 pin may be configured as an I/O. Ensure that the OSCIOFNC bit is set (Configuration Word FOSC<2>).

**Question 2:** *Why does my device run at a frequency much higher than the resonant frequency of the crystal?*

**Answer:** The gain is too high for this oscillator circuit. Refer to **Section 38.5.2.4 “Determining the Best Values for Oscillator Components”** to aid in the selection of C2 (it may need to be higher), Rs (may be needed) and clock mode (wrong mode may be selected). This is especially possible for low-frequency crystals, such as the common 32.768 kHz.

**Question 3:** *The design runs fine, but the frequency is slightly off. What can be done to adjust this?*

**Answer:** Changing the value of C1 has some effect on the oscillator frequency. If a series resonant crystal is used, it will resonate at a different frequency than a parallel resonant crystal of the same frequency call-out. Ensure that you are using a parallel resonant crystal.

**Question 4:** *What would cause my application to work fine, but then suddenly quit or lose time?*

**Answer:** Other than the obvious software checks that should be done to investigate losing time, it is possible that the amplitude of the oscillator output is not high enough to reliably trigger the oscillator input. Also, look at the C1 and C2 values and ensure that the device Configuration bits are correct for the desired oscillator mode.

**Question 5:** *If I put an oscilloscope probe on an oscillator pin, I do not see what I expect. Why?*

**Answer:** Remember that an oscilloscope probe has capacitance. Connecting the probe to the oscillator circuitry will modify the oscillator characteristics. Consider using a low capacitance (active) probe.

## 38.16 REGISTER MAPS

A summary of the registers associated with the PIC24F Oscillator with 500 kHz Low-Power FRC module is provided in Table 38-4.

**Table 38-4: Oscillator Register Map**

| File Name | Bit 15 | Bit 14 | Bit 13 | Bit 12 | Bit 11 | Bit 10 | Bit 9  | Bit 8  | Bit 7   | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1  | Bit 0 | All Resets          |
|-----------|--------|--------|--------|--------|--------|--------|--------|--------|---------|-------|-------|-------|-------|-------|--------|-------|---------------------|
| OSCCON    | —      | COSC2  | COSC1  | COSC0  | —      | NOSC2  | NOSC1  | NOSC0  | CLKLOCK | —     | LOCK  | —     | CF    | —     | SOSCEN | OSWEN | xxxx <sup>(1)</sup> |
| CLKDIV    | ROI    | DOZE2  | DOZE1  | DOZE0  | DOZEN  | RCDIV2 | RCDIV1 | RCDIV0 | —       | —     | —     | —     | —     | —     | —      | —     | 0100                |
| OSCTUN    | —      | —      | —      | —      | —      | —      | —      | —      | —       | —     | TUN5  | TUN4  | TUN3  | TUN2  | TUN1   | TUN0  | 0000                |
| REFOCON   | ROEN   | —      | ROSSLP | ROSEL  | RODIV3 | RODIV2 | RODIV1 | RODIV0 | —       | —     | —     | —     | —     | —     | —      | —     | 0000                |

**Legend:** x = unknown value on Reset, — = unimplemented, read as '0'. Reset values are shown in hexadecimal for the fully implemented registers.

**Note 1:** OSCCON register Reset values dependent on the FOSC Configuration bits and by type of Reset.

## 38.17 RELATED APPLICATION NOTES

This section lists application notes that are related to this section of the manual. These application notes may not be written specifically for the PIC24F device family, but the concepts are pertinent and could be used with modification and possible limitations. The current application notes related to the Oscillator with 500 kHz Low-Power FRC of PIC24F Devices are:

| Title                                                                            | Application Note # |
|----------------------------------------------------------------------------------|--------------------|
| Crystal Oscillator Basics and Crystal Selection for rfPIC® and PICmicro® Devices | AN826              |
| Basic PICmicro® Oscillator Design                                                | AN849              |
| Practical PICmicro® Oscillator Analysis and Design                               | AN943              |
| Making Your Oscillator Work                                                      | AN949              |

|                                                                                                                                                                                                    |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Note:</b> Please visit the Microchip web site ( <a href="http://www.microchip.com">www.microchip.com</a> ) for additional application notes and code examples for the PIC24F family of devices. |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

### 38.18 REVISION HISTORY

#### Revision A (January 2009)

This is the initial released revision of this document.

NOTES: