

---



---

## Section 39. Oscillator (Part III)

---



---

### HIGHLIGHTS

This section of the manual contains the following topics:

39.1	Introduction .....	39-2
39.2	CPU Clocking.....	39-3
39.3	Oscillator Configuration Registers .....	39-4
39.4	Special Function Registers .....	39-7
39.5	Primary Oscillator.....	39-13
39.6	Internal Fast RC (FRC) Oscillator .....	39-17
39.7	Phase-Locked Loop (PLL) .....	39-18
39.8	Low-Power Secondary Oscillator (SOSC) .....	39-23
39.9	Low-Power RC Oscillator.....	39-24
39.10	Auxiliary Oscillator .....	39-25
39.11	Fail-Safe Clock Monitor (FSCM) .....	39-26
39.12	Clock Switching.....	39-27
39.13	Two-Speed Start-Up.....	39-31
39.14	Register Maps.....	39-32
39.15	Related Application Notes.....	39-33
39.16	Revision History .....	39-34

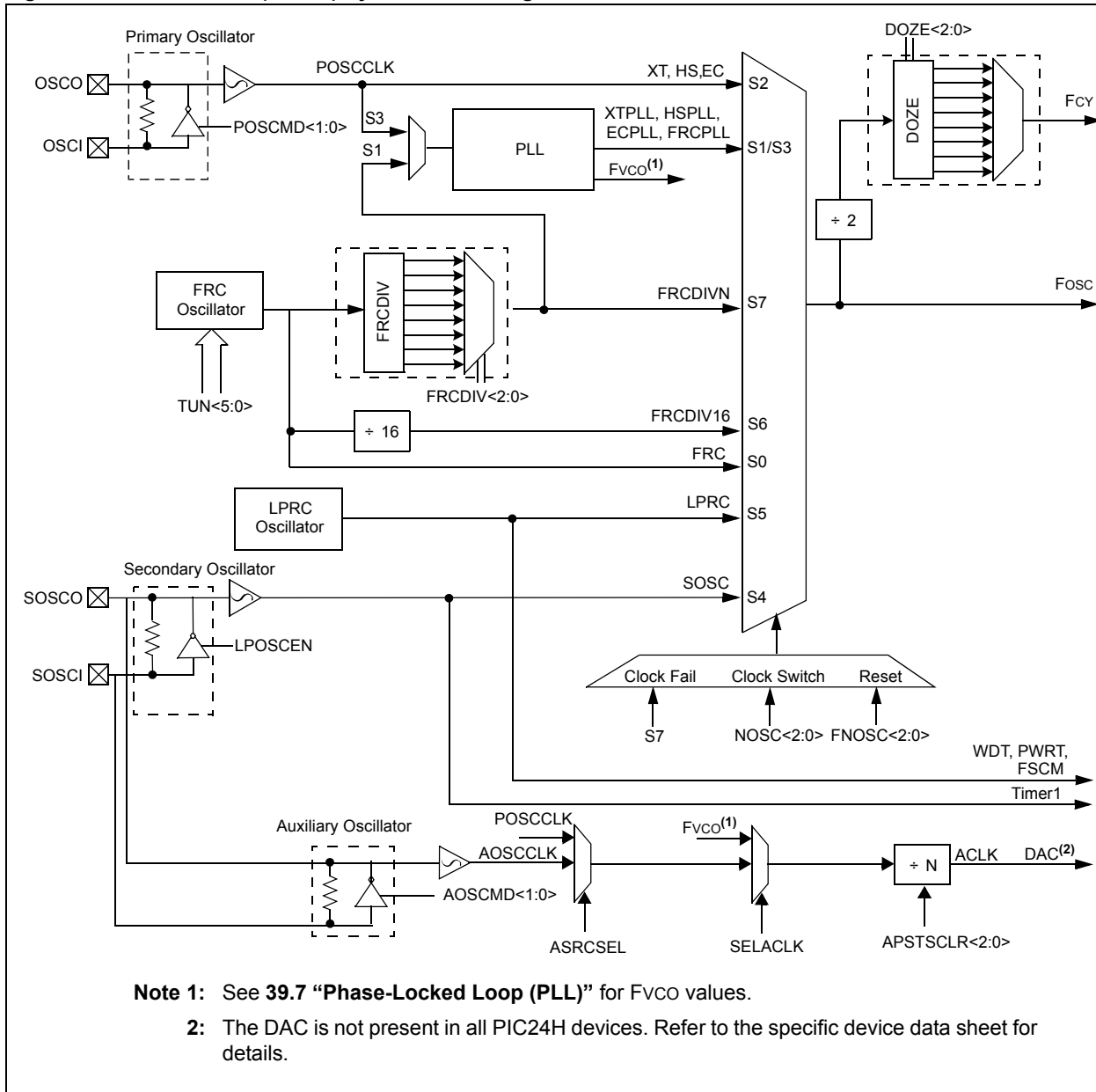
## 39.1 INTRODUCTION

The PIC24H oscillator system includes these characteristics:

- External and Internal Oscillator options
- On-chip PLL to boost internal operating frequency on select internal and external oscillator sources
- On-the-fly clock switching between various clock sources
- Doze mode for system power-saving
- Fail-Safe Clock Monitor (FSCM) that detects clock failure and permits safe application recovery or shutdown
- Nonvolatile Configuration bits for clock source selection
- Auxiliary Crystal Oscillator for Digital-to-Analog Converter (DAC)

A block diagram of the PIC24H oscillator system is shown in Figure 39-1.

**Figure 39-1: Oscillator (Part III) System Block Diagram**



## 39.2 CPU CLOCKING

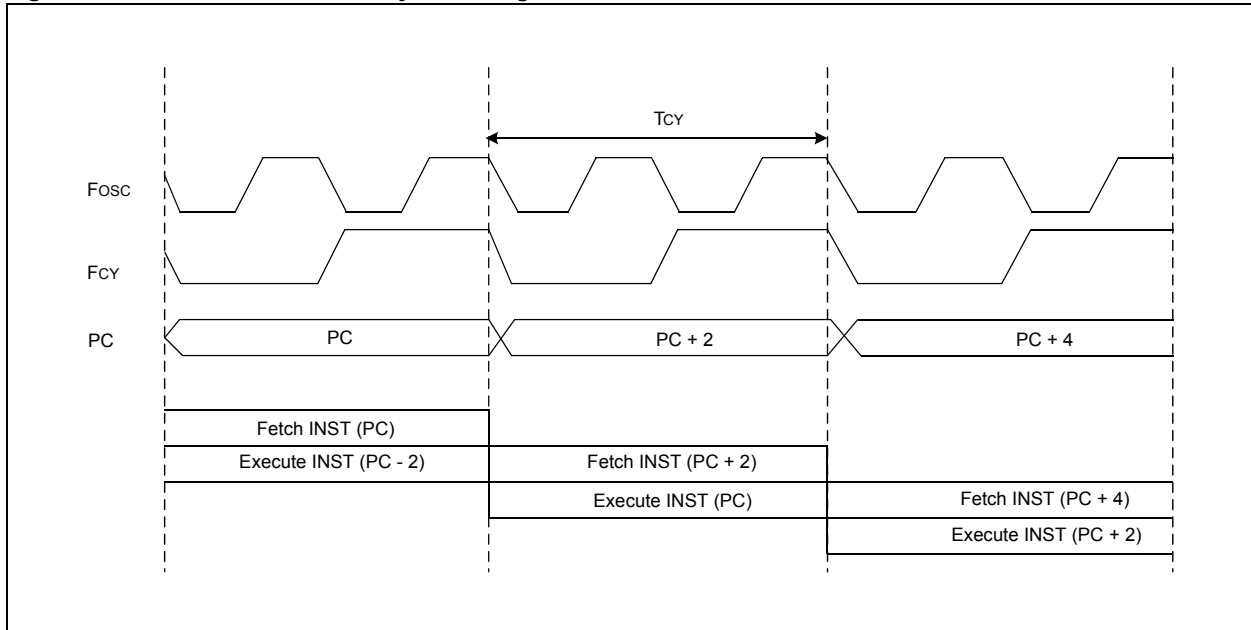
The system clock ( $F_{osc}$ ) source can be provided by one of the following options.

- Primary Oscillator (POSC) on the OSC1 and OSC0 pins
- Secondary Oscillator (SOSC) on the SOSCI and SOSCO pins
- Internal Fast RC Oscillator (FRC) with optional clock divider
- Internal Low-Power RC Oscillator (LPRC)
- Primary Oscillator with PLL
- Internal Fast RC Oscillator with PLL

The system clock source is divided by two to produce the internal instruction cycle clock. The instruction cycle clock is denoted by  $F_{cy}$ . The timing diagram in Figure 39-2 shows the relationship between the system clock ( $F_{osc}$ ), the instruction cycle clock ( $F_{cy}$ ), and the Program Counter (PC).

The internal instruction cycle clock ( $F_{cy}$ ) can be output on the OSC0 I/O pin, if the Primary Oscillator mode or the HS mode is not selected as the clock source (see 39.5 “Primary Oscillator”).

**Figure 39-2: Clock/Instruction Cycle Timing**



## 39.3 OSCILLATOR CONFIGURATION REGISTERS

Table 39-1 lists the configuration settings that select the device oscillator source and operating mode at Power-on Reset (POR). The Configuration bits are contained in these two registers:

- **FOSCSEL: Oscillator Source Selection Register**

The Oscillator Source Selection (FOSCSEL) register selects the initial oscillator source and start-up option.

- **FOSC: Oscillator Configuration Register**

The Oscillator Configuration (FOSC) register configures the Primary Oscillator mode, OSCO pin function, peripheral pin select, and the fail-safe and clock switching modes.

The Configuration registers are located in program memory space. They are not Special Function Registers (SFRs). They are mapped into program memory space and are programmed at the time of device programming.

The Initial Oscillator Source Selection (FNOSC<2:0>) Configuration bits in the Oscillator Source Selection (FOSCSEL<2:0>) register determine the clock source that is used at a Power-on Reset. Thereafter, the clock source can be changed between permissible clock sources with clock switching. The Internal FRC Oscillator with postscaler (FRCDIVN) is the default (unprogrammed) selection.

The Primary Oscillator Mode Selection (POSCMD<1:0>) Configuration bits in the Oscillator Configuration (FOSC<1:0>) register select the Operation mode of the primary oscillator.

The OSCO Pin Function bit (OSCIOFNC) Configuration bit in FOSC<2> selects the OSCO pin function, except in HS or XT mode. When OSCIOFNC is unprogrammed ('1'), the Fcy clock is output on the OSCO pin. When OSCIOFNC is programmed ('0'), the OSCO pin becomes a general purpose I/O pin.

**Table 39-1: Configuration Bit Values for Clock Selection**

Oscillator Source	Oscillator Mode	FNOSC Value	POSCMD Value	Note
S0	Fast RC Oscillator (FRC)	000	xx	<b>1</b>
S1	Fast RC Oscillator with PLL (FRCPLL)	001	xx	<b>1</b>
S2	Primary Oscillator (EC)	010	00	<b>1</b>
S2	Primary Oscillator (XT)	010	01	—
S2	Primary Oscillator (HS)	010	10	—
S3	Primary Oscillator with PLL (ECPLL)	011	00	<b>1</b>
S3	Primary Oscillator with PLL (XTPLL)	011	01	—
S3	Primary Oscillator with PLL (HSPLL)	011	10	—
S4	Secondary Oscillator (SOSC)	100	xx	<b>1</b>
S5	Low-Power RC Oscillator	101	xx	<b>1</b>
S6	Fast RC Oscillator with Divide-by-16 (FRCDIV16)	110	xx	<b>1</b>
S7	Fast RC Oscillator with Divide-by-N (FRCDIVN)	111	xx	<b>1, 2</b>

**Note 1:** The OSCO pin function is determined by the OSCIOFNC Configuration bit.

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

## Section 39. Oscillator (Part III)

**Register 39-1: FOSCSEL: Oscillator Source Selection Register**

U	U	U	U	U	U	U	U
—	—	—	—	—	—	—	—
bit 15						bit 8	
R/P	U	U	U	U	R/P	R/P	R/P
IESO	—	—	—	—	FNOSC<2:0>		
bit 7						bit 0	

<b>Legend:</b>			
R = Readable bit	P = Programmable bit	U = Unused bits, program to Logic '1'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-8 **Reserved:** Reserved bits must be programmed as '1'

bit 7 **IESO:** Internal External Start-up Option bit

- 1 = Start device with Internal FRC, and then auto-switch to the user-selected oscillator source when ready
- 0 = Start device with user-selected oscillator source

bit 6-3 **Reserved:** Reserved bits must be programmed as '1'

bit 2-0 **FNOSC<2:0>:** Initial Oscillator Source Selection bits

- 000 = Fast RC Oscillator (FRC)
- 001 = Fast RC Oscillator with PLL (FRCPLL)
- 010 = Primary Oscillator (XT, HS, EC)
- 011 = Primary Oscillator with PLL (XTPLL, HSPLL, ECPLL)
- 100 = Secondary Oscillator (SOSC)
- 101 = Low-Power RC Oscillator (LPRC)
- 110 = Fast RC Oscillator with Divide-by-16 (FRCDIV16)
- 111 = Fast RC Oscillator with Divide-by-N (FRCDIVN)

# PIC24H Family Reference Manual

**Register 39-2: FOSC: Oscillator Configuration Register**

U	U	U	U	U	U	U	U
—	—	—	—	—	—	—	—
bit 15							bit 8

R/P	R/P	R/P	U	U	R/P	R/P	R/P
FCKSM<1:0>		IOL1WAY	—	—	OSCIOFNC	POSCMD<1:0>	
bit 7							bit 0

**Legend:**

R = Readable bit	P = Programmable bit	U = Unused bits, program to Logic '1'
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared      x = Bit is unknown

- bit 15-8    **Reserved:** Reserved bits must be programmed as '1'
- bit 7-6    **FCKSM<1:0>:** Clock Switching Mode bits
  - 1x = Clock switching is disabled, Fail-Safe Clock Monitor is disabled
  - 01 = Clock switching is enabled, Fail-Safe Clock Monitor is disabled
  - 00 = Clock switching is enabled, Fail-Safe Clock Monitor is enabled
- bit 5      **IOL1WAY:** Peripheral Pin Select Configuration bit
  - 1 = Allow only one reconfiguration
  - 0 = Allow multiple reconfigurations
- bit 4-3    **Reserved:** Reserved bits must be programmed as '1'
- bit 2      **OSCIOFNC:** OSCO Pin Function bit (except in XT and HS modes)
  - 1 = OSCO is clock output and instruction cycle (FCY) clock is output on OSCO pin
  - 0 = OSCO is a general purpose digital I/O pin
- bit 1-0    **POSCMD<1:0>:** Primary Oscillator Mode Selection bits
  - 11 = Primary Oscillator disabled
  - 10 = HS (High-Speed) Crystal Oscillator mode
  - 01 = XT (Crystal) Oscillator mode
  - 00 = EC (External Clock) mode

### 39.4 SPECIAL FUNCTION REGISTERS

These Special Function Registers provide run-time control and status of the oscillator system:

- **OSCCON: Oscillator Control Register**

The Oscillator Control (OSCCON) register controls clock switching and provides status information that allows the current clock source, PLL lock, and clock fail conditions to be monitored.

- **CLKDIV: Clock Divisor Register**

The Clock Divisor (CLKDIV) register controls the Doze mode and selects the PLL prescaler, PLL postscaler and FRC postscaler.

- **PLLFBF: PLL Feedback Divisor Register**

The PLL Feedback Divisor (PLLFBF) register selects the PLL feedback divisor.

- **OSCTUN: FRC Oscillator Tuning Register**

The FRC Oscillator Tuning (OSCTUN) register is used to tune the Internal FRC Oscillator frequency in software. It allows the FRC Oscillator frequency to be adjusted over a range of  $\pm 12\%$ .

- **ACLKCON: Auxiliary Clock Control Register**

The Auxiliary Clock Control (ACLKCON) register controls the Auxiliary Oscillator mode and the auxiliary output clock divider.

<b>Note:</b> The Oscillator Special Function Registers (OSCCON, CLKDIV, PLLFBF, OSCTUN and ACLKCON) are reset only on Power-on Reset.
---

# PIC24H Family Reference Manual

**Register 39-3: OSCCON: Oscillator Control Register**

U-0	R-y	R-y	R-y	U-0	R/W-y	R/W-y	R/W-y
—	COSC<2:0>			—	NOSC<2:0>		
bit 15				bit 8			

R/S-0	R/W-0	R-0	U-0	R/C-0	U-0	R/W-0	R/W-0
CLKLOCK	IOLOCK	LOCK	—	CF	—	LPOSCEN	OSWEN
bit 7							bit 0

<b>Legend:</b>	U= Unimplemented bit, read as '0'	y = Depends on FOSCSEL<FNOSC> bits
R = Readable bit	W = Writable bit	C = Clearable only bit
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared
		S = Settable only bit
		x = Bit is unknown

- bit 15      **Unimplemented:** Read as '0'
- bit 14-12    **COSC<2:0>:** Current Oscillator Selection bits (read-only)
  - 000 = Fast RC Oscillator (FRC)
  - 001 = Fast RC Oscillator with PLL (FRCPLL)
  - 010 = Primary Oscillator (XT, HS, EC)
  - 011 = Primary Oscillator with PLL (XTPLL, HSPLL, ECPLL)
  - 100 = Secondary Oscillator (SOSC)
  - 101 = Low-Power RC Oscillator (LPRC)
  - 110 = Fast RC Oscillator with Divide-by-16 (FRCDIV16)
  - 111 = Fast RC Oscillator with Divide-by-N (FRCDIVN)
- bit 11      **Unimplemented:** Read as '0'
- bit 10-8    **NOSC<2:0>:** New Oscillator Selection bits
  - 000 = Fast RC Oscillator (FRC)
  - 001 = Fast RC Oscillator with PLL (FRCPLL)
  - 010 = Primary Oscillator (XT, HS, EC)
  - 011 = Primary Oscillator with PLL (XTPLL, HSPLL, ECPLL)
  - 100 = Secondary Oscillator (SOSC)
  - 101 = Low-Power RC Oscillator (LPRC)
  - 110 = Fast RC Oscillator with Divide-by-16 (FRCDIV16)
  - 111 = Fast RC Oscillator with Divide-by-N (FRCDIVN)
- bit 7      **CLKLOCK:** Clock Lock Enable bit
  - If clock switching is enabled and FSCM is disabled (FOSC<FCKSM> = 01)
  - 1 = Clock switching is disabled, system clock source is locked
  - 0 = Clock switching is enabled, system clock source may be modified by clock switching
- bit 6      **IOLOCK:** Peripheral Pin Select Lock bit
  - 1 = Peripheral Pin Select is locked. Writes to Peripheral Pin Select registers are not allowed
  - 0 = Peripheral Pin Select is not locked. Writes to Peripheral Pin Select registers are allowed
- bit 5      **LOCK:** PLL Lock Status bit (read-only)
  - 1 = Indicates that PLL is in lock, or PLL start-up timer is satisfied
  - 0 = Indicates that PLL is out of lock, start-up timer is in progress or PLL is disabled
- bit 4      **Unimplemented:** Read as '0'
- bit 3      **CF:** Clock Fail Detect bit (read/clear by application)
  - 1 = FSCM has detected clock failure
  - 0 = FSCM has not detected clock failure
- bit 2      **Unimplemented:** Read as '0'
- bit 1      **LPOSCEN:** Secondary Oscillator (SOSC) Enable bit
  - 1 = Enable Secondary Oscillator
  - 0 = Disable Secondary Oscillator
- bit 0      **OSWEN:** Oscillator Switch Enable bit
  - 1 = Request oscillator switch to selection specified by NOSC<2:0> bits
  - 0 = Oscillator switch is complete



**Register 39-4: CLKDIV: Clock Divisor Register**

R/W-0	R/W-0	R/W-1	R/W-1	R/W-0	R/W-0	R/W-0	R/W-0
ROI	DOZE<2:0>			DOZEN <sup>(1)</sup>	FRCDIV<2:0>		
bit 15							bit 8

R/W-0	R/W-1	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PLLPOST<1:0>		—	PLLPRE<4:0>				
bit 7							bit 0

<b>Legend:</b>	y = Value set from Configuration bits on POR
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 will clear the DOZEN bit and the processor clock/peripheral clock ratio is set to 1:1  
                   0 = Interrupts have no effect on the DOZEN bit
  
- bit 14-12       **DOZE<2:0>:** Processor Clock Reduction Select bits  
                   000 = Fcy/1  
                   001 = Fcy/2  
                   010 = Fcy/4  
                   011 = Fcy/8 (default)  
                   100 = Fcy/16  
                   101 = Fcy/32  
                   110 = Fcy/64  
                   111 = Fcy/128
  
- bit 11           **DOZEN:** DOZE Mode Enable bit<sup>(1)</sup>  
                   1 = DOZE<2:0> field specifies the ratio between the peripheral clocks and the processor clocks  
                   0 = Processor clock/peripheral clock ratio forced to 1:1
  
- bit 10-8        **FRCDIV<2:0>:** Internal Fast RC Oscillator Postscaler bits  
                   000 = FRC/1 (default)  
                   001 = FRC/2  
                   010 = FRC/4  
                   011 = FRC/8  
                   100 = FRC/16  
                   101 = FRC/32  
                   110 = FRC/64  
                   111 = FRC/256
  
- bit 7-6         **PLLPOST<1:0>:** PLL VCO Output Divider Select bits (also denoted as 'N2', PLL postscaler)  
                   00 = Output/2  
                   01 = Output/4 (default)  
                   10 = Reserved  
                   11 = Output/8
  
- bit 5            **Unimplemented:** Read as '0'
  
- bit 4-0         **PLLPRE<4:0>:** PLL Phase Detector Input Divider Select bits (also denoted as 'N1', PLL prescaler)  
                   00000 = Input/2 (default)  
                   00001 = Input/3  
                   •  
                   •  
                   •  
                   11111 = Input/33

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

# PIC24H Family Reference Manual

**Register 39-5: PLLFBD: PLL Feedback Divisor Register**

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	PLLDIV<8>
bit 15							bit 8

R/W-0	R/W-0	R/W-1	R/W-1	R/W-0	R/W-0	R/W-0	R/W-0
PLLDIV<7:0>							
bit 7							bit 0

**Legend:**

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

bit 15-9                      **Unimplemented:** Read as '0'  
 bit 8-0                      **PLLDIV<8:0>:** PLL Feedback Divisor bits (also denoted as 'M', PLL multiplier)  
                                     000000000 = 2  
                                     000000001 = 3  
                                     000000010 = 4  
                                     •  
                                     •  
                                     •  
                                     000110000 = 50 (default)  
                                     •  
                                     •  
                                     •  
                                     111111111 = 513

## Section 39. Oscillator (Part III)

**Register 39-6: 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
—	—	TUN<5:0>					
bit 7							bit 0

**Legend:**

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

bit 15-6      **Unimplemented:** Read as '0'

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

011111 = Center frequency + 11.625% (8.23 MHz)

011110 = Center frequency + 11.25% (8.20 MHz)

•

•

•

000001 = Center frequency + 0.375% (7.40 MHz)

000000 = Center frequency (7.37 MHz nominal)

111111 = Center frequency - 0.375% (7.345 MHz)

•

•

•

100001 = Center frequency - 11.625% (6.52 MHz)

100000 = Center frequency - 12% (6.49 MHz)

# PIC24H Family Reference Manual

**Register 39-7: ACLKCON: Auxiliary Clock Control Register**

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	
—	—	SELACLK	AOSCMD<1:0>	APSTSCLR<2:0>				
bit 15								bit 8

R/W-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0	
ASRCSEL	—	—	—	—	—	—	—	
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 15-14      **Unimplemented:** Read as '0'
- bit 13      **SELACLK:** Select Auxiliary Clock Source for Auxiliary Clock Divider  
             1 = Auxiliary Oscillator provides the source clock for Auxiliary Clock Divider  
             0 = Fast RC (FRC) with PLL provides the source clock for Auxiliary Clock Divider
- bit 12-11      **AOSCMD<1:0>:** Auxiliary Oscillator Mode  
             11 = EC (External Clock) Mode Select  
             10 = XT (Crystal) Oscillator Mode Select  
             01 = HS (High-Speed) Oscillator Mode Select  
             00 = Auxiliary Oscillator Disabled (default)
- bit 10-8      **APSTSCLR<2:0>:** Auxiliary Clock Output Divider  
             111 = Divided by 1  
             110 = Divided by 2  
             101 = Divided by 4  
             100 = Divided by 8  
             011 = Divided by 16  
             010 = Divided by 32  
             001 = Divided by 64  
             000 = Divided by 256 (default)
- bit 7      **ASRCSEL:** Select Reference Clock Source for Auxiliary Clock  
             1 = Primary Oscillator is the Clock Source  
             0 = Auxiliary Oscillator is the Clock Source
- bit 6-0      **Unimplemented:** Read as '0'

## 39.5 PRIMARY OSCILLATOR

The Primary Oscillator is available on the OSCI and OSCO pins of the PIC24H device. This connection enables an external crystal (or ceramic resonator) to provide the clock to the device. Optionally, the internal PLL can be used to boost the system frequency ( $F_{osc}$ ) to 80 MHz for 40 MIPS execution. The Primary Oscillator provides the following modes of operation:

- **Crystal Oscillator (XT Mode)**

The XT mode is a medium-gain, medium-frequency mode used to work with crystal frequencies of 3 MHz to 10 MHz.

- **High-Speed Oscillator (HS Mode)**

The HS mode is a high-gain, high-frequency mode used to work with crystal frequencies of 10 MHz to 40 MHz.

- **External Clock Source Operation (EC Mode)**

If the on-chip oscillator is not used, the EC mode allows the internal oscillator to be bypassed. The device clocks are generated from an external source (0.8 MHz to 64 MHz) and input on the OSCI pin.

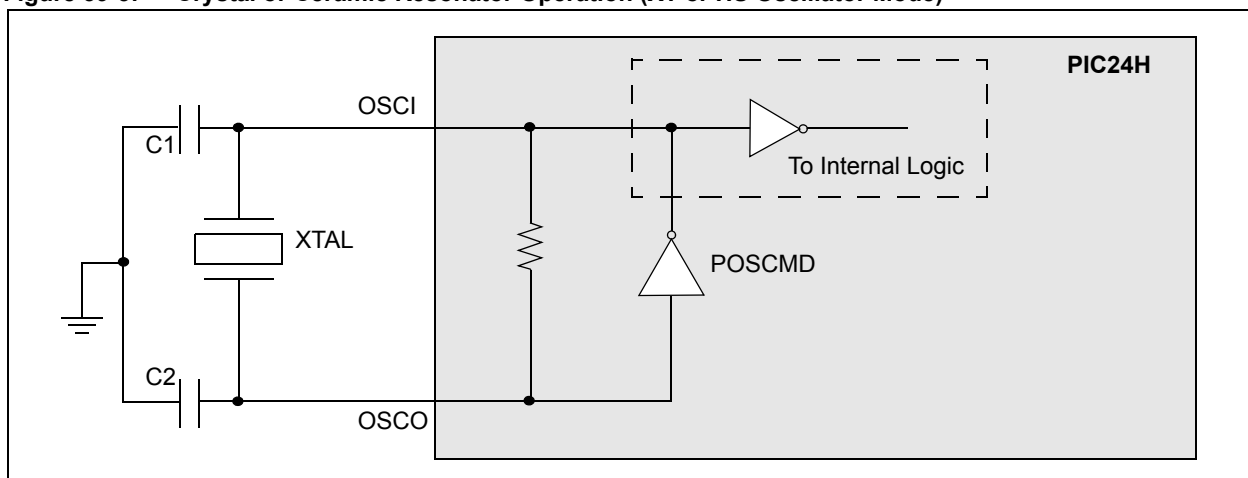
The Initial Oscillator Source Selection (FNOSC<2:0>) Configuration bits in the Oscillator Source Selection (FOSCSEL<2:0>) register specify the primary oscillator clock source at Power-on Reset. The Primary Oscillator Mode Selection (POSCMD<1:0>) Configuration bits in the Oscillator Configuration (FOSC<1:0>) register specify the Primary Oscillator mode. Table 39-2 shows the options selected by specific bit configurations, which are programmed at the time of device programming.

**Table 39-2: Primary Oscillator Clock Source Options**

FNOSC Value	POSCMD	Primary Oscillator Source/Mode
011	00	Primary Oscillator with PLL: External Clock Mode (ECPLL)
011	01	Primary Oscillator with PLL: Crystal Oscillator with PLL Mode (XTPLL)
011	10	Primary Oscillator with PLL: High-Speed Oscillator with PLL Mode (HSPLL)
010	00	Primary Oscillator: External Clock Mode (EC)
010	01	Primary Oscillator: Crystal Oscillator Mode (XT)
010	10	Primary Oscillator: High-Speed Mode (HS)

Figure 39-3 shows a recommended crystal oscillator circuit diagram for PIC24H devices. Capacitors C1 and C2 form the load capacitance for the crystal. The optimum load capacitance ( $C_L$ ) for a given crystal is specified by the crystal manufacturer. Load capacitance can be calculated as shown in Equation 39-1.

**Figure 39-3: Crystal or Ceramic Resonator Operation (XT or HS Oscillator Mode)**



## Equation 39-1: Crystal Load Capacitance

$$C_L = C_S + \frac{C_1 \cdot C_2}{C_1 + C_2}$$

where:

$C_S$  is the stray capacitance

Assuming  $C_1 = C_2$ , Equation 39-2 gives the capacitor value ( $C_1$ ,  $C_2$ ) for a given load and stray capacitance.

## Equation 39-2: External Capacitor for Crystal

$$C_1 = C_2 = 2 \cdot (C_L - C_S)$$

For additional information on crystal oscillators and their operation refer to **39.15 “Related Application Notes”**.

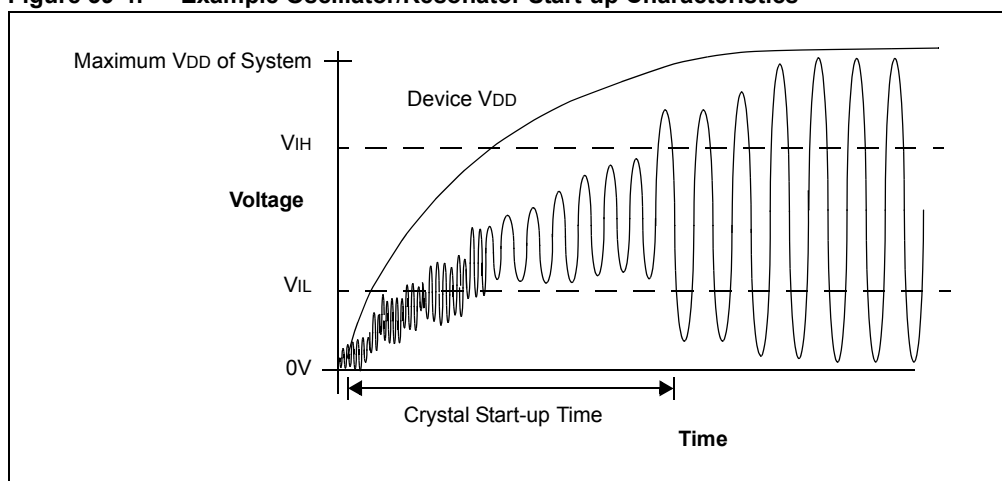
### 39.5.1 Oscillator Start-up Time

The oscillator starts oscillating as the device voltage increases from  $V_{SS}$ . The time required for the oscillator to start oscillating depends on the following factors:

- Crystal/Resonator frequency
- Capacitor values used ( $C_1$  and  $C_2$  in Figure 39-3)
- Device  $V_{DD}$  rise time
- System temperature
- Series resistor value and type if used
- Oscillator mode selection of device (selects the gain of the internal oscillator inverter)
- Crystal quality
- Oscillator circuit layout
- System noise

Figure 39-4 shows a graph of a typical oscillator/resonator start-up.

**Figure 39-4: Example Oscillator/Resonator Start-up Characteristics**



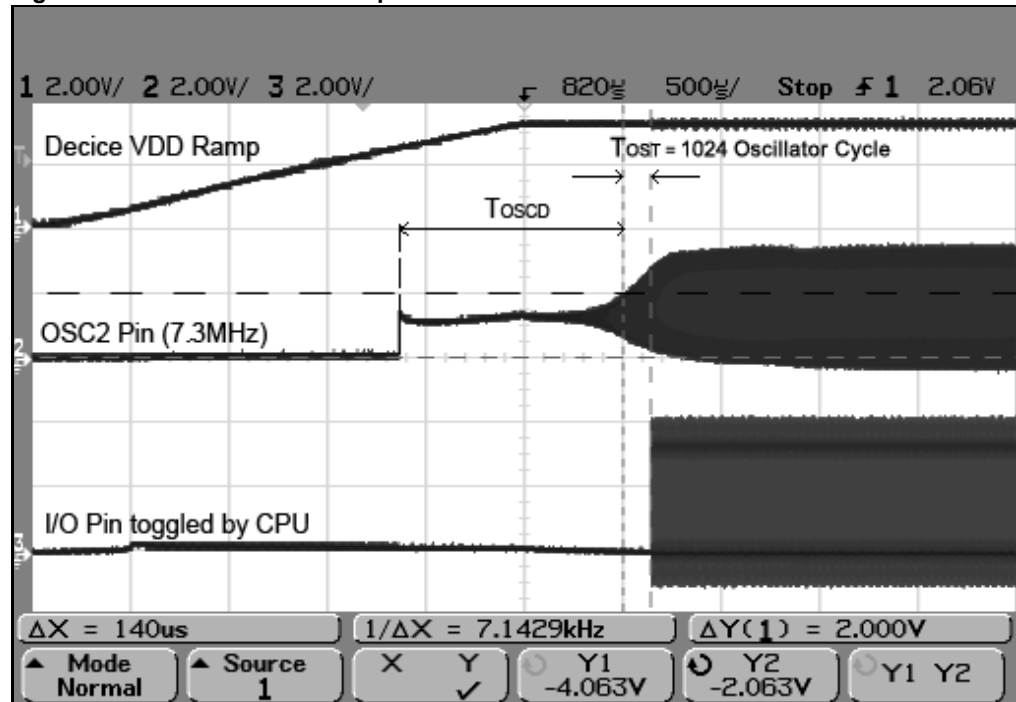
To ensure that a crystal oscillator (or ceramic resonator) has started and stabilized, an Oscillator Start-up Timer (OST) is provided with the Primary Oscillator (POSC) and the Secondary Oscillator (SOSC). The OST is a simple 10-bit counter that counts 1024 cycles before releasing the oscillator clock to the rest of the system. This time-out period is denoted 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 restarts (i.e., on POR, BOR and wake-up from Sleep mode).

Once the Primary Oscillator is enabled, it takes a finite amount of time to start oscillating. This delay is denoted as  $T_{OSCD}$ . After  $T_{OSCD}$ , the OST timer takes 1024 clock cycles ( $T_{OST}$ ) to release the clock. The total delay for the clock to be ready is  $T_{OSCD} + T_{OST}$ . If the PLL is used, an additional delay is required for the PLL to lock (see 39.7 “Phase-Locked Loop (PLL)”).

Primary Oscillator start-up characteristics are illustrated in Figure 39-5, where the CPU starts toggling an I/O pin when it starts execution after the  $T_{OSCD} + T_{OST}$  interval.

**Figure 39-5: Oscillator Start-up Characteristics**



## 39.5.2 Primary Oscillator Pin Functionality

The Primary Oscillator pins (OSCI/OSCO) can be used for other functions when the oscillator is not being used.

The POSCMD Configuration bits in the Oscillator Configuration (FOSC<1:0>) register determine the Oscillator pin function.

The OSCIOFNC bit (FOSC<2>) determines the OSKO pin function. When FOSC<2> is '0', OSKO is a general purpose digital I/O pin (see Figure 39-6). When FOSC<2> is '1', OSKO is a clock output, and the instruction cycle (Fcy) clock is output on the OSKO pin (see Figure 39-7).

The Oscillator pin functions are shown in Table 39-3.

**Table 39-3: Clock Pin Function Selection**

Oscillator Source	OSCIOFNC<2> Value	POSCMD<1:0> Value	OSCI <sup>(1)</sup> Pin Function	OSCO <sup>(2)</sup> Pin Function
Primary OSC Disabled	1	11	Digital I/O	Clock Output (Fcy)
Primary OSC Disabled	0	11	Digital I/O	Digital I/O
HS (High-Speed)	X	10	OSCI	OSCO
XT (Crystal)	X	01	OSCI	OSCO
EC (External Clock)	1	00	OSCI	Clock Output (Fcy)
EC (External Clock)	0	00	OSCI	Digital I/O

**Note 1:** OSCI pin function is determined by Primary Oscillator Mode (POSCMD<1:0>) Configuration bits.

**Note 2:** OSCI pin function is determined by Primary Oscillator Mode (POSCMD<1:0>) and OSKO Pin Function (OSCIOFNC<2>) Configuration bits.

# PIC24H Family Reference Manual

Figure 39-6: OSCO Pin for Digital I/O (in EC Mode), FOSC<2> = 0

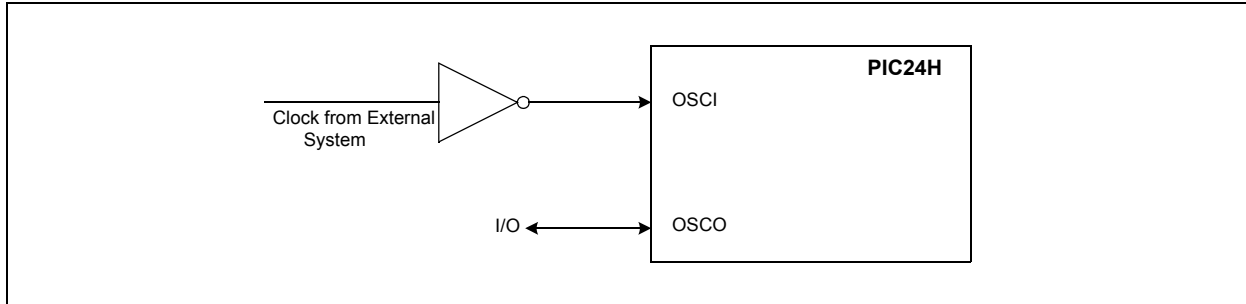
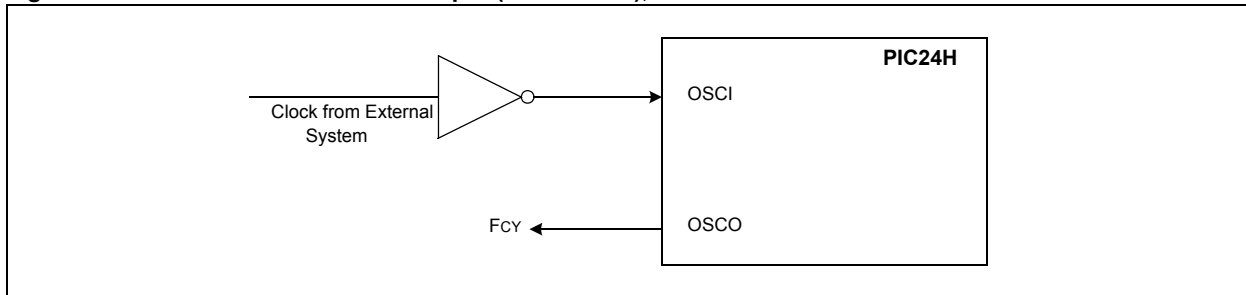


Figure 39-7: OSCO Pin for Clock Output (in EC Mode), FOSC<2> = 1





## 39.6 INTERNAL FAST RC (FRC) OSCILLATOR

The Internal Fast RC (FRC) Oscillator provides a nominal 7.37 MHz clock without requiring an external crystal or ceramic resonator, which results in system cost savings for applications that do not require a precise clock reference.

The application software can tune the frequency of the oscillator from -12% to +11.625% (30 kHz steps) of the nominal frequency value using the FRC Oscillator Tuning (TUN<5:0>) bits in the FRC Oscillator Tuning (OSCTUN<5:0>) register. The nominal or tuned frequency of the FRC Oscillator is expected to remain within  $\pm 2\%$  of the tuned value over temperature and voltage variations of a particular device.

**Note:** Refer to the specific device data sheet for the accuracy of FRC clock frequency over temperature and voltage variations.

The Internal FRC Oscillator starts up instantly. Unlike a crystal oscillator, which can take several milliseconds to begin oscillation, the Internal FRC starts oscillating immediately.

The Initial Oscillator Source Selection (FNOSC<2:0>) Configuration bits in the Oscillator Source Selection (FOSCSEL<2:0>) register select the FRC clock source. The FRC Clock Source options at the time of Power-on Reset are shown in Table 39-4. The Configuration bits are programmed at the time of device programming.

**Table 39-4: FRC Clock Source Options**

FNOSC<2:0> Value	Primary Oscillator Source/Mode
111	FRC Oscillator: Postscaler by N (FRCDIVN)
110	FRC Oscillator: Postscaler by 16 (FRCDIV16)
001	FRC Oscillator with PLL (FRCPLL)
000	FRC Oscillator (FRC)

### 39.6.1 FRC Postscaler Mode (FRCDIVN)

In FRC Postscaler mode, a variable postscaler divides the FRC clock output and allows a lower frequency to be chosen. The postscaler is controlled by the Internal Fast RC Oscillator Postscaler (FRCDIV<2:0>) bits in the Clock Divisor (CLKDIV<10:8>) register. These bits allow for eight settings, from 1:1 to 1:256, as shown in Figure 39-5.

**Table 39-5: Internal Fast RC Oscillator Postscaler Settings**

FRCDIV<2:0> Value	Internal FRC Oscillator Setting
111	FRC divide by 256
110	FRC divide by 64
101	FRC divide by 32
100	FRC divide by 16
011	FRC divide by 8
010	FRC divide by 4
001	FRC divide by 2
000	FRC divide by 1 (default)

Optionally, the FRC postscaler output can be used with the internal PLL to boost the system frequency (Fosc) to 80 MHz for 40 MIPS instruction cycle execution speed.

**Note:** The FRC Divider should not be changed dynamically when operating in Internal FRC with PLL.

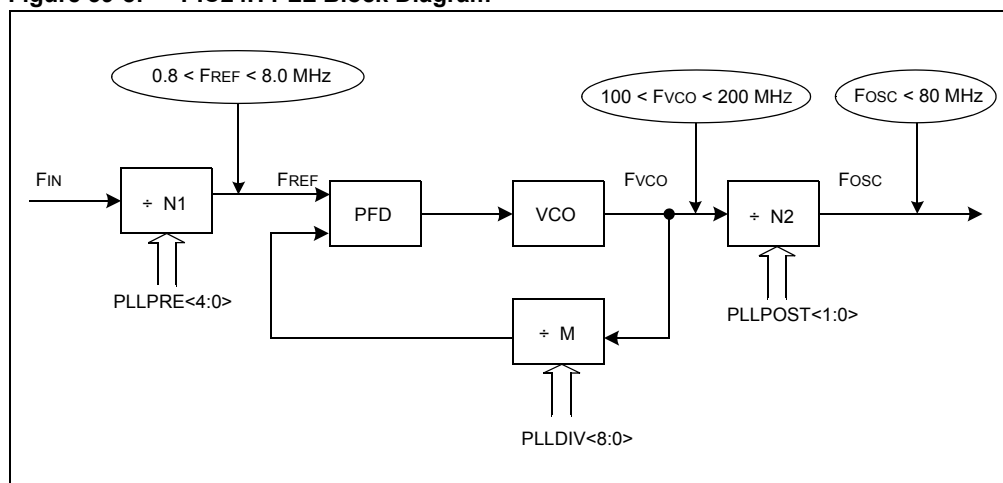
To change the FRC divider:

1. Switch the clock to a non-PLL mode (e.g., Internal FRC).
2. Make the necessary changes.
3. Switch the clock back to the PLL mode.

## 39.7 PHASE-LOCKED LOOP (PLL)

The Primary Oscillator and Internal FRC Oscillator sources can also be used with an on-chip PLL to obtain higher operating speeds. A block diagram of the PLL module is shown in Figure 39-8.

**Figure 39-8: PIC24H PLL Block Diagram**



For proper PLL operation, the Phase Frequency Detector (PFD) input frequency and Voltage Controlled Oscillator (VCO) output frequency must meet the following requirements:

- The PFD input frequency ( $F_{REF}$ ) must be in the range of 0.8 MHz to 8 MHz
- The VCO output frequency ( $F_{VCO}$ ) must be in the range of 100 MHz to 200 MHz

The PLL Phase Detector Input Divider Select ( $PLLPRE<4:0>$ ) bits in the Clock Divisor ( $CLKDIV<4:0>$ ) register specify the input divider ratio ( $N1$ ), which is used to scale down the input clock ( $F_{IN}$ ) to meet the PFD input frequency range of 0.8 MHz to 8 MHz.

The PLL Feedback Divisor ( $PLLDIV<8:0>$ ) bits in the PLL Feedback Divisor ( $PLLFBD<8:0>$ ) register specify the divider ratio ( $M$ ), which scales down the VCO frequency ( $F_{VCO}$ ) for feedback to the PFD. The VCO frequency ( $F_{VCO}$ ) is 'M' times the input reference clock ( $F_{REF}$ ).

The PLL VCO Output Divider Select ( $PLLPOST<1:0>$ ) bits in the Clock Divisor ( $CLKDIV<7:6>$ ) register specify the divider ratio ( $N2$ ) to limit the system clock frequency ( $F_{OSC}$ ) to 80 MHz.

Equation 39-3 gives the relation between the input frequency ( $F_{IN}$ ) and the output frequency ( $F_{OSC}$ ).

### Equation 39-3: Fosc Calculation

$$F_{OSC} = F_{IN} \times \left( \frac{M}{N1 \times N2} \right) = F_{IN} \times \left( \frac{(PLLDIV + 2)}{(PLLPRE + 2) \times 2(PLLPOST + 1)} \right)$$

where:

$$N1 = PLLPRE + 2$$

$$N2 = 2 \times (PLLPOST + 1)$$

$$M = PLLDIV + 2$$

Equation 39-4 gives the relation between the input frequency ( $F_{IN}$ ) and the VCO frequency ( $F_{VCO}$ ).

### Equation 39-4: Fvco Calculation

$$F_{VCO} = F_{IN} \times \left( \frac{M}{N1} \right) = F_{IN} \times \left( \frac{(PLLDIV + 2)}{(PLLPRE + 2)} \right)$$

## 39.7.1 Input Clock Limitation at Start-up for PLL Mode

Table 39-6 gives the default values of the PLL Prescaler, PLL Postscaler and PLL Feedback Divisor Configuration bits at Power-on Reset.

**Table 39-6: PLL Mode Defaults**

Register	Bit Field	Value at POR Reset	PLL Divider Ratio
CLKDIV<4:0>	PLLPRE<4:0>	00	N1 = 2
CLKDIV<7:6>	PLLPOST<1:0>	01	N2 = 4
PLLFBD<8:0>	PLLDIV<8:0>	000110000	M = 50

Given these reset values, the following equations give the relation between the input frequency ( $F_{IN}$ ) and PFD input frequency ( $F_{REF}$ ), and the VCO frequency ( $F_{VCO}$ ) and system clock frequency ( $F_{OSC}$ ) at Power-on Reset.

**Equation 39-5:  $F_{REF}$  at Power-on Reset**

$$F_{REF} = F_{IN} \left( \frac{1}{N1} \right) = 0.5(F_{IN})$$

**Equation 39-6:  $F_{VCO}$  at Power-on Reset**

$$F_{VCO} = F_{IN} \left( \frac{M}{N1} \right) = F_{IN} \left( \frac{50}{2} \right) = 25(F_{IN})$$

**Equation 39-7:  $F_{OSC}$  at Power-on Reset**

$$F_{OSC} = F_{IN} \left( \frac{M}{N1 \cdot N2} \right) = 6.25(F_{IN})$$

Given the above equations at Power-on Reset, the input frequency ( $F_{IN}$ ) to the PLL module must be limited to  $4 \text{ MHz} < F_{IN} < 8 \text{ MHz}$  to comply with the VCO output frequency requirement ( $100\text{M} < F_{VCO} < 200\text{M}$ ), if the default values of PLLPRE, PLLPOST and PLLDIV are used.

The Primary Oscillator can support the following input frequency ranges, which are not within the frequency limit required ( $4 \text{ MHz} < F_{IN} < 8 \text{ MHz}$ ) at Power-on Reset:

- Primary Oscillator in XT mode supports: 3 MHz to 10 MHz crystal
- Primary Oscillator in HS mode supports: 10 MHz to 40 MHz crystal
- Primary Oscillator in EC mode supports: 0.8 MHz to 64 MHz input

To use the PLL when the input frequency is not within the 4 MHz to 8 MHz range, this process must be followed:

1. Power-up the device with Internal FRC or Primary Oscillator without PLL.
2. Change the PLLDIV, PLLPRE and PLLPOST bit values, based on the input frequency, to meet these PLL requirements:
  - The PFD input frequency ( $F_{REF}$ ) must be in the range of 0.8 MHz to 8 MHz
  - The VCO output frequency ( $F_{VCO}$ ) must be in the range of 100 MHz to 200 MHz
3. Switch the clock to the PLL mode in software.

## 39.7.2 PLL Lock Status

Whenever the PLL input frequency, the PLL prescaler or the PLL feedback divisor is changed, the PLL requires a finite amount of time (T<sub>LOCK</sub>) to synchronize to the new settings.

T<sub>LOCK</sub> is applied when PLL is selected as the clock source at Power-on Reset, or during a clock switching operation. The value of T<sub>LOCK</sub> is relative to the time at which the clock is available to the PLL input. For example, with the Primary Oscillator, T<sub>LOCK</sub> starts after the OST delay. Refer to 39.5.1 “Oscillator Start-up Time” for detailed information.

The PLL Lock Status (LOCK) bit in the Oscillator Control (OSCCON<5>) register is a read-only bit that indicates the Lock status of the PLL. The LOCK bit is cleared at Power-on Reset and on a clock-switch operation, when the PLL is selected as the destination clock source. It remains clear when any clock source not using the PLL is selected. It is a good practice to wait for the LOCK bit to be set before executing code after a clock switch event in which the PLL is enabled.

**Note:** The PLL Prescaler (PLLPRE) and PLL Feedback Divisor (PLLDIV) should not be changed when operating in PLL mode. You must clock switch to a non-PLL mode (e.g., Internal FRC), to make the necessary changes, and then clock switch back to the PLL mode.

### 39.7.2.1 SETUP FOR USING PLL WITH PRIMARY OSCILLATOR (POSC)

The following process can be used to set up the PLL to operate the device at 40 MIPS with a 10 MHz external crystal:

1. To execute instructions at 40 MHz, ensure that the required system clock frequency is:  
 $F_{osc} = 2 \times F_{CY} = 80 \text{ MHz}$
2. Ensure that the default reset values of PLLPRE, PLLPOST and PLLDIV meet the PLL and user requirements:
  - $F_{REF} = 0.5 \cdot F_{IN} = 5 \text{ MHz}$
  - $F_{OSC} = 6.25 \cdot F_{IN} = 62.5 \text{ MHz}$
  - $F_{VCO} = 25 \cdot F_{IN} = 250 \text{ MHz}$
  - FVCO is not meeting the PLL requirement
  - FOSC is not meeting the user requirement
3. If the PLL and user requirements are met, directly configure the FNOSC bits (FOSCSEL<2:0>) to select the Primary Oscillator with PLL at Power-on Reset. Otherwise, if the PLL and user requirements are not met, follow these steps:
  - a) Select the PLL postscaler to meet the VCO output frequency requirement ( $100 \text{ MHz} < F_{vco} < 200 \text{ MHz}$ ):
    - Select a PLL postscaler ratio of  $N_2 = 2$
    - Ensure that  $F_{vco} = (F_{osc} \cdot N_2) = 160 \text{ MHz}$
  - b) Select the PLL prescaler to meet the PFD input frequency requirement ( $0.8 \text{ MHz} < F_{REF} < 8 \text{ MHz}$ ):
    - Select a PLL prescaler ratio of  $N_1 = 2$
    - Ensure that  $F_{REF} = (F_{IN}/N_1) = 5 \text{ MHz}$
  - c) Select the PLL feedback divisor to generate the required VCO output frequency based on the PFD input frequency:
    - $F_{vco} = F_{REF} \times M$
    - $M = F_{vco}/F_{REF} = 32$
  - d) Configure the FNOSC bits (FOSCSEL<2:0>) to select a clock source without the PLL (e.g., Internal FRC) at Power-on Reset.
  - e) In the main program, change the PLL prescaler, PLL postscaler and PLL feedback divisor values to those decided in the previous steps, and then perform a clock switch to the PLL mode.

Example 39-1 provides the code for using PLL with the Primary Oscillator. See 39.12 “Clock Switching” for a clock switching code example.

### Example 39-1: Code Example for Using PLL with Primary Oscillator (POSC)

```

// Select Internal FRC at POR
_FOSCSEL(FNOSC_FRC);

// Enable Clock Switching and Configure POSC in XT mode
_FOSC(FCKSM_CSECMD & OSCIOFNC_OFF & POSCMD_XT);

int main()
{

// Configure PLL prescaler, PLL postscaler, PLL divisor
PLLFBFD=30;           // M = 32
CLKDIVbits.PLLPOST=0; // N1 = 2
CLKDIVbits.PLLPRE=0;  // N2 = 2

// Initiate Clock Switch to Primary Oscillator with PLL (NOSC = 0b011)
__builtin_write_OSCCONH(0x03);
__builtin_write_OSCCONL(0x01);

// Wait for Clock switch to occur
while (OSCCONbits.COSC != 0b011);

// Wait for PLL to lock
while(OSCCONbits.LOCK!=1) {};

}

```

#### 39.7.2.2 SETUP FOR USING PLL WITH 7.37 MHz INTERNAL FRC

The following process can be used to set up the PLL to operate the device at 40 MIPS with a 7.37 MHz Internal FRC.

1. To execute instructions at 40 MHz, ensure that the system clock frequency is:  
 $F_{osc} = 2 \times F_{CY} = 80 \text{ MHz}$
2. Ensure that the default reset values of PLLPRE, PLLPOST and PLLDIV meet the PLL and user requirements.
  - $F_{REF} = 0.5 \cdot F_{IN} = 3.68 \text{ MHz}$
  - $F_{OSC} = 6.25 \cdot F_{IN} = 46 \text{ MHz}$
  - $F_{VCO} = 25 \cdot F_{IN} = 184 \text{ MHz}$
  - FOSC is not meeting the user requirement
3. If the PLL and user requirements are met, directly configure the FNOSC bits (FOSCSEL<2:0>) to select the Primary Oscillator with PLL at Power-on Reset. Otherwise, if the PLL and user requirements are not met, follow these steps:
  - a) Select the PLL postscaler to meet the VCO output frequency requirement (100 MHz < Fvco < 200 MHz):
    - Select a PLL postscaler ratio of N2 = 2
    - Ensure that Fvco = (Fosc • N2) = 160 MHz
  - b) Select the PLL prescaler to meet the PFD input frequency requirement (0.8 MHz < FREF < 8 MHz):
    - Select a PLL prescaler ratio of N1 = 2
    - Ensure that FREF = (Fin/N1) = 3.68 MHz
  - c) Select the PLL feedback divisor to generate the required VCO output frequency based on the PFD input frequency:
    - $F_{VCO} = F_{REF} \times M$
    - $M = F_{VCO}/F_{REF} = 43$
  - d) Configure the FNOSC bits (FOSCSEL<2:0>) to select a clock source without PLL (e.g., Internal FRC) at Power-on Reset.
  - e) In the main program, change the PLL prescaler, PLL postscaler and PLL feedback divisor to meet the user and PLL requirement, and then perform a clock switch to the PLL mode.

Example 39-2 provides the code for using PLL with a 7.37 MHz Internal FRC. See **39.12 “Clock Switching”** for a clock switching code example.

**Example 39-2: Code Example for Using PLL with 7.37 MHz Internal FRC**

```
// Select Internal FRC at POR
_FOSCSEL(FNOSC_FRC);

// Enable Clock Switching and Configure
_FOSC(FCKSM_CSECMD & OSCIOFNC_OFF);

int main()
{

// Configure PLL prescaler, PLL postscaler, PLL divisor
PLLFBD = 41;           // M = 43
CLKDIVbits.PLLPOST=0; // N1 = 2
CLKDIVbits.PLLPRE=0;  // N2 = 2

// Initiate Clock Switch to Internal FRC with PLL (NOSC = 0b001)
__builtin_write_OSCCONH(0x01);
__builtin_write_OSCCONL(0x01);

// Wait for Clock switch to occur
while (OSCCONbits.COSC != 0b001);

// Wait for PLL to lock
while(OSCCONbits.LOCK!=1) {};

}
```

### 39.8 LOW-POWER SECONDARY OSCILLATOR (SOSC)

The Low-Power Secondary Oscillator enables a 32.768 kHz crystal to be attached to the PIC24H device as a secondary crystal clock source for low-power operation. This oscillator uses the SOSCI and SOSCO pins. The Low-Power Secondary Oscillator can also drive Timer1 for real-time clock applications.

#### 39.8.1 Secondary Oscillator for System Clock

The Low-Power Secondary Oscillator is enabled as the system clock when:

- The Initial Oscillator Source Selection (FNOSC<2:0>) Configuration bits in the Oscillator Source Selection (FOSCSEL<2:0>) register are appropriately set to select the Secondary Oscillator at a Power-on Reset
- User software initiates a Clock Switch to the Secondary Oscillator for low-power operation

If the Low-Power Secondary Oscillator is not being used to provide the system clock, or if the device enters Sleep mode, it is disabled to save power.

#### 39.8.2 Secondary Oscillator Start-up Delay

When the Low-Power Secondary Oscillator is enabled, it takes a finite amount of time to start oscillating. Refer to **39.5.1 “Oscillator Start-up Time”** for details.

#### 39.8.3 Continuous Secondary Oscillator Operation

Optionally, you can leave the Secondary Oscillator running at all times. The Secondary Oscillator is always enabled if the Secondary Oscillator Enable (LPOSCEN) bit is set in the Oscillator Control (OSCCON<1>) register.

There are two reasons to leave the Low-Power Secondary Oscillator running. First, keeping the oscillator on at all times allows a fast switch to the 32 kHz System Clock for lower power operation. Returning to the faster main oscillator still requires an oscillator start-up time if it is a crystal type source (see **39.5.1 “Oscillator Start-up Time”**). Second, the oscillator should remain on at all times when Timer1 is being used as a real-time clock.

**Note:** In Sleep mode, all clock sources (Primary Oscillator, Internal FRC Oscillator and LPRC Oscillator) are shut down, with the exception of the Low-Power Secondary Oscillator. The Low-Power Secondary Oscillator can be active in Sleep mode if the Secondary Oscillator Enable (LPOSCEN) bit is set in the Oscillator Control (OSCCON<1>) register.

## 39.9 LOW-POWER RC OSCILLATOR

The Low-Power RC (LPRC) Oscillator provides a nominal clock frequency of 32 kHz. The LPRC is the clock source for the Power-up Timer (PWRT), Watchdog Timer (WDT), and Fail-Safe Clock Monitor (FSCM) circuits. It can 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.

**Note:** The clock frequency of the LPRC Oscillator will vary depending on the device voltage and operating temperature. Refer to the “Electrical Characteristics” in the specific device data sheet for details.

### 39.9.1 LPRC Oscillator for System Clock

The LPRC Oscillator is selected as the system clock when:

- The Initial Oscillator Source Selection (FNOSC<2:0>) bits in the Oscillator Source Selection (FOSCSEL<2:0>) register are appropriately set to select the LPRC oscillator at Power-on Reset
- User software initiates a Clock Switch to the LPRC Oscillator for low-power operation

### 39.9.2 Enabling the LPRC Oscillator

The LPRC Oscillator is the clock source for the PWRT, WDT and FSCM. The LPRC Oscillator is enabled at Power-on Reset if the Power-on Reset Timer Value Select (FPWRT) bits in the POR Configuration Fuse (FPOR<2:0>) register.

The LPRC oscillator remains enabled under these conditions:

- The FSCM is enabled
- The WDT is enabled
- The LPRC Oscillator is selected as the system clock

If none of these conditions is true, the LPRC Oscillator shuts off after the PWRT expires. The LPRC Oscillator is shut off in Sleep mode.

**Note:** The LPRC runs in Sleep mode only if the Watchdog Timer is enabled. Under all other conditions, the LPRC is disabled in Sleep mode.

### 39.9.3 LPRC Oscillator Start-up Delay

The LPRC Oscillator starts up instantly, unlike a crystal oscillator, which can take several milliseconds to begin oscillation.



## 39.10 AUXILIARY OSCILLATOR

The Auxiliary Oscillator (AOSC) can be used for peripherals that need to operate at a frequency unrelated to the system clock, such as a Digital-to-Analog Converter (DAC). The Auxiliary Oscillator can use one of the following as its clock source:

- Crystal (XT): Crystal and ceramic resonators in the range of 3 MHz to 10 MHz
- High-Speed Crystal (HS): Crystals in the range of 10 MHz to 40 MHz. The external crystal is connected to the SOSCI and SOSCO pins
- External Clock (EC): External clock signal up to 64 MHz. The external clock signal is directly applied to the SOSCI pin

### 39.10.1 Enabling the Auxiliary Oscillator

To enable the Auxiliary Oscillator mode and External Oscillator mode, the appropriate Auxiliary Oscillator Mode (AOSCMD<1:0>) bits must be selected in the Auxiliary Clock Control (ACLKCON<12:11>) register. These bits allow for four oscillator mode settings, as shown in Table 39-7. Once the mode has been selected, set the Select Auxiliary Clock Source for Auxiliary Clock Divider (SELACLK) bit (ACLKCON<12>) to use the Auxiliary Oscillator as the clock reference.

**Table 39-7: Auxiliary Oscillator and External Oscillator Mode Settings**

AOSCMD<1:0> Bit Value	Oscillator Mode Setting
11	EC (External Clock) Mode Select
10	XT (Crystal) Oscillator Mode Select
01	HS (High-Speed) Oscillator Mode Select
00	Auxiliary Oscillator Disabled (default setting)

**Note:** By default, the DAC module is clocked by the FRC with PLL. To use the Primary Oscillator as the clock source, set the following bits in the ACLKCON register:

- ASRCSEL (selects primary oscillator for the reference clock)
- SELACLK (enables the reference clock)

### 39.10.2 Auxiliary Clock Output Divider

The Auxiliary Clock Output Divider (APSTSCLR<2:0>) bits in the ACLKCON<10:8> register divide the auxiliary clock, which allow a lower frequency to be chosen. These bits allow for eight postscale settings, from 1:1 to 1:256, as shown in Table 39-8.

**Table 39-8: Auxiliary Clock Output Divider Settings**

APSTSCLR<2:0> Bit Value	Auxiliary Oscillator Setting
111	Divide by 1
110	Divide by 2
101	Divide by 4
100	Divide by 8
011	Divide by 16
010	Divide by 32
001	Divide by 64
000	Divide by 256 (default setting)

## 39.11 FAIL-SAFE CLOCK MONITOR (FSCM)

The Fail-Safe Clock Monitor allows the device to continue to operate in the event of an oscillator failure. The FSCM function is enabled by programming the Clock Switching Mode (FCKSM<1:0>) Configuration bits in the Oscillator Configuration (FOSC<7:6>) register at the time of device programming. When FSCM is enabled (FCKSM = 00), the LPRC Internal Oscillator will run at all times except during Sleep mode.

The FSCM monitors the system clock. If it does not detect a system clock within a specific period of time (typically 2 ms, maximum 4 ms), it generates a clock failure trap and switches the system clock to the FRC oscillator. The user application then has the option to either attempt to restart the oscillator or execute a controlled shutdown.

<b>Note:</b> The FSCM does not wake-up the device if the clock fails while the device is in Sleep mode.
---

The FSCM takes the following actions when it switches to the FRC oscillator:

- The Current Oscillator Selection (COSC<2:0>) bits (OSCCON<14:12>) are loaded with '000' (Internal FRC)
- The Clock Fail Detect (CF) bit (OSCCON<3>) is set to indicate the clock failure
- The Oscillator Switch Enable (OSWEN) control bit (OSCCON<0>) is cleared to cancel any pending clock switches

### 39.11.1 FSCM Delay

The FSCM monitors the system clock for activity after the system clock is ready and the nominal delay (TFSCM) has elapsed.

The FSCM delay (TFSCM) is applied when the FSCM is enabled and the primary or secondary oscillator is selected as the system clock.

Refer to **Section 8. "Reset"** (DS70192) for additional information. Check for the most recent documentation on the Microchip web site at [www.microchip.com](http://www.microchip.com).

<b>Note:</b> Please refer to the "Electrical Characteristics" section of the specific device data sheet for TFSCM values.
---

### 39.11.2 FSCM and WDT

The FSCM and the WDT both 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.

## 39.12 CLOCK SWITCHING

Clock switching can be initiated as a result of a hardware event or a software request. Typical scenarios include:

- Two-Speed Start-up sequence upon Power-on Reset, which initially uses the Internal FRC Oscillator for quick start-up, and then automatically switches to the selected clock source when the clock is ready
- Fail-Safe Clock Monitor automatically switches to the Internal FRC Oscillator on a clock failure
- User application software requests clock switching by setting the OSWEN bit (OSCCON<0>), causing the hardware to switch to the clock source selected by the NOSC bits (OSCCON<10:8>) when the clock is ready

In each of these cases, the clock switch event assures that proper make-before-break sequence is executed. That is, the new clock source must be ready before the old clock is deactivated and code must continue to execute as clock switching occurs.

With few limitations, applications are free to switch between any of the four clock sources (POSC, SOSC, FRC and LPRC) under software control at any time. To limit the possible side effects that could result from this flexibility, PIC24H devices have a safeguard lock built into the switch process. That is, the OSCCON register is write-protected during clock switching.

### 39.12.1 Enabling Clock Switching

The Clock Switching Mode (FCKSM<1:0>) Configuration bits in the Oscillator Configuration (FOSC<7:6>) register must be programmed to enable clock switching and the Fail-Safe Clock Monitor (see Table 39-9).

**Table 39-9: Configurable Clock Switching Modes**

FCKSM<1:0> Values	Clock Switching Configuration	FSCM Configuration
1x	Disabled	Disabled
01	Enabled	Disabled
00	Enabled	Enabled

The first bit determines if clock switching is enabled ('0') or disabled ('1'). The second bit determines if the FSCM is enabled ('0') or disabled ('1'). FSCM can only be enabled if clock switching is also enabled. If clock switching is disabled ('1'), the value of the second bit is irrelevant.

### 39.12.2 Clock Switch Sequence

The recommended process for a clock switch is as follows:

1. Read the COSC bits (OSCCON<14:12>) to determine the current oscillator source (if this information is relevant to the application).
2. Execute the unlock sequence to allow a write to the high byte of the OSCCON register.
3. Write the appropriate value to the NOSC control bits (OSCCON<10:8>) for the new oscillator source.
4. Execute the unlock sequence to allow a write to the low byte of the OSCCON register.
5. Set the OSWEN bit (OSCCON<0>) to initiate the oscillator switch.

After the above steps are completed, the clock switch logic performs the following:

1. The clock switching hardware compares the OSCCON<COSC> status bits with the new value of the NOSC control bit (OSCCON<10:8>). If they are the same, the clock switch is a redundant operation. In this case, the OSWEN bit (OSCCON<0>) is cleared automatically and the clock switch is aborted.
2. If a valid clock switch has been initiated, the PLL Lock (OSCCON<5>) and Clock Fail (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 (Primary or Secondary) must be turned on, the hardware waits until the OST expires. If the new source uses the PLL, the hardware waits until a PLL lock is detected (OSCCON<5> = 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 (OSCCON<0>) to indicate a successful clock transition. In addition, the NOSC bit (OSCCON<10:8>) values are transferred to the COSC status bits (OSCCON<14:12>).
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 SOSCCEN remains set). The timing of the transition between clock sources is shown in Figure 39-9.

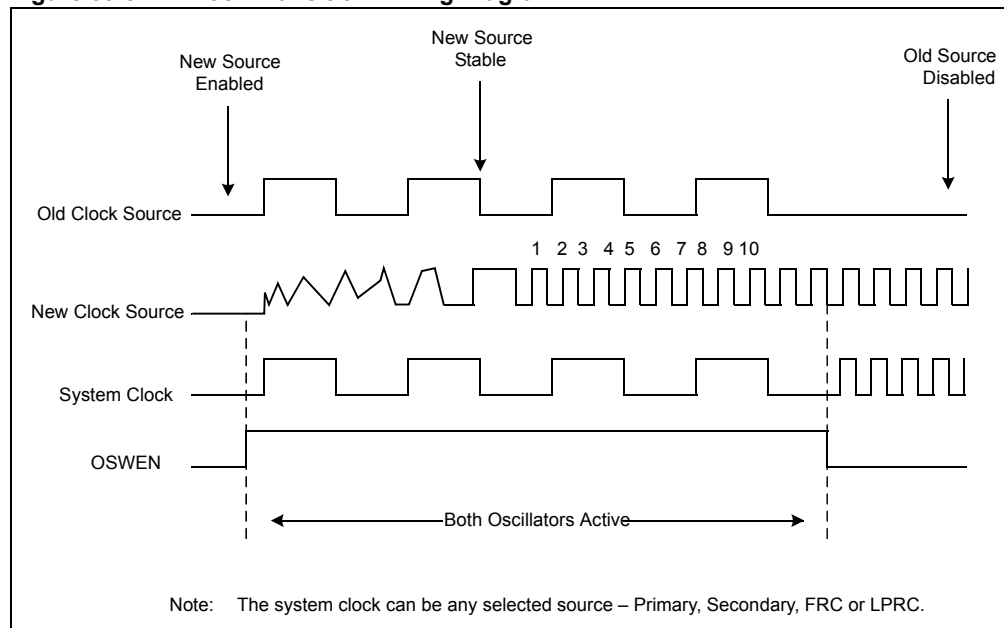
**Note 1:** Clock switching between XT, HS and EC primary oscillator modes is not possible without reprogramming the device.

**2:** Direct clock switching between PLL modes is not possible. For example, clock switching should not occur between Primary Oscillator with PLL, and Internal FRC Oscillator with PLL.

**3:** Setting the CLKLOCK bit (OSCCON<7>) prevents clock switching when clock switching is enabled and fail-safe clock monitoring is disabled by the FCKSM Configuration bits (FOSC<7:6> = 01). The OSCCON<7> bit cannot be cleared once it is set by the software it clears on Power-on Reset.

**4:** The processor continues to execute code throughout the clock switching sequence. Timing sensitive code should not be executed during this time.

**Figure 39-9: 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, in two back-to-back instructions:
  - Write 0x0078 to OSCCON<15:8>
  - Write 0x009A to OSCCON<15:8>
3. In the instruction immediately following the unlock sequence, write the new oscillator source to the NOSC control bits (OSCCON<10:8>).
4. Execute the unlock sequence for the OSCCON low byte.  
In two, back-to-back instructions:
  - Write 0x0046 to OSCCON<7:0>
  - Write 0x0057 to OSCCON<7:0>
5. In the instruction immediately following the unlock sequence, set the OSWEN bit (OSCCON<0>).
6. Continue to execute code that is not clock-sensitive (optional).
7. Check to see if OSCCON<0> is '0'. If it is, the switch was successful.

**Note:** MPLAB® C30 provides built-in C language functions for unlocking the OSCCON register:

```
__builtin_write_OSCCONL(value)  
__builtin_write_OSCCONH(value)
```

See MPLAB IDE Help for more information.

Example 39-3 provides the code sequence for unlocking the OSCCON register, and switching from FRC with PLL clock to the LPRC clock source.

### Example 39-3: Code Example for Clock Switching

```
;Place the New Oscillator Selection (NOSC=0b101) in W0  
MOV #0x15,w0  
  
;OSCCONH (high byte) Unlock Sequence  
MOV #OSCCONH, w1  
MOV #0x78, w2  
MOV #0x9A, w3  
MOV.B w2, [w1] ; Write 0x0078  
MOV.B w3, [w1] ; Write 0x009A  
  
;Set New Oscillator Selection  
MOV.B w0, [w1]  
  
; Place 0x01 in W0 for setting clock switch enabled bit  
MOV #0x01, w0  
  
;OSCCONL (low byte) Unlock Sequence  
MOV #OSCCONL, w1  
MOV #0x46, w2  
MOV #0x57, w3  
MOV.B w2, [w1] ; Write 0x0046  
MOV.B w3, [w1] ; Write 0x0057  
  
; Enable Clock Switch  
MOV.B w0, [w1] ; Request Clock Switching by Setting OSWEN bit  
wait:  
btsc OSCCONL, #OSWEN  
bra wait
```

## 39.12.3 Clock Switching Consideration

When you incorporate clock switching into an application, issues to keep in mind when designing your code include:

- 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.
- 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 continue to run from the current clock source. Your software can detect this situation because the OSWEN bit (OSCCON<0>) remains set indefinitely.
- If the new clock source uses the PLL, a clock switch will not occur until lock has been achieved. Your software can detect a loss of PLL lock because the LOCK bit (OSCCON<5>) is cleared and the OSWEN bit (OSCCON<0>) is set.
- Switching to a low-frequency clock source, such as the Secondary Oscillator, will result in slow device operation.

## 39.12.4 Aborting a Clock Switch

If a clock switch does not complete, the clock switch logic can be reset by clearing the OSWEN bit (OSCCON<0>). When OSWEN is cleared, the clock switch process is aborted, the Oscillator Start Timer (if applicable) is stopped and reset, and the PLL (if applicable) is stopped.

Typical assembly code for aborting a clock switch is shown in Example 39-4. 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 39-4: 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
```

## 39.12.5 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 PWRSAV instruction is then executed normally.

It is particularly useful to perform a clock switch to the internal FRC oscillator before entering Sleep mode, as this will ensure fast wake-up from Sleep.

### 39.13 TWO-SPEED START-UP

The Internal External Start-up Option (IESO) Configuration bit in the Oscillator Source Selection (FOSCSEL<7>) register specifies whether to start the device with a user-selected oscillator source or to initially start with the Internal FRC, and then switch automatically to the user-selected oscillator. If this bit is set to '1', the device will always power up on the Internal FRC Oscillator, regardless of the other oscillator source settings (FOSCSEL<2:0>). The device then automatically switches to the specified oscillator, when it is ready.

Unless FSCM is enabled, the FRC oscillator is automatically turned off immediately after the clock switch is completed. The Two-Speed Start-up option is a faster way to get the device up and running and works independently of the state of the FCKSM Configuration bits (FOSC<7:6>).

Two-Speed Start-up is particularly useful when an external oscillator (either a Primary or Secondary oscillator), which has a longer start-up time, is selected by the FOSCSEL Configuration bits (FOSC<2:0>).

As an Internal RC Oscillator, the FRC clock source is available almost immediately following Power-on Reset. With Two-Speed Start-up, the device starts executing code in its default oscillator configuration (FRC). It continues to operate in this mode until the specified external oscillator source becomes stable, when it automatically switches to that source.

User code can check which clock source is currently providing the device clocking by checking the status of the COSC bits (OSCCON<14:12>) against the NOSC bits (OSCCON<10:8>). If these two sets of bits match, the clock switch has been completed successfully and the device is running from the intended clock source.

**Note:** Two-Speed Start-up is redundant if the selected device clock source is FRC.

## 39.14 REGISTER MAPS

Table 39-10 maps the bit functions for the Oscillator Special Function control registers. Table 39-11 maps the bit functions for the Oscillator Configuration registers.

**Table 39-10: Oscillator Special Function Control Registers**

File Name	Addr	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	0742	—	COSC<2:0>			—	NOSC<2:0>			CLKLOCK	IOLOCK	LOCK	—	CF	—	LPOSCEN	OSWEN	7700 <sup>(1)</sup>
CLKDIV	0744	ROI	DOZE<2:0>			DOZEN	FRCDIV<2:0>			PLLPOST<1:0>		—	PLLPRE<4:0>					3040
PLLFBD	0746	—	—	—	—	—	—	—	—	PLLDIV<8:0>								0030
OSCTUN	0748	—	—	—	—	—	—	—	—	—	—	TUN<5:0>						0000
ACLKCON	074A	—	—	SELACLK	AOSCMD<1:0>		APSTSCLR<2:0>			ASRCSEL	—	—	—	—	—	—	—	0000

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

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

**Table 39-11: Oscillator Configuration Registers**

File Name	Addr	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
FOSCSEL	F80006	—	—	—	—	—	—	—	—	IESO	—	—	—	—	FNOSC<2:0>		
FOSC	F80008	—	—	—	—	—	—	—	—	FCKSM<1:0>		IOL1WAY	—	—	OSCI0FNC	POSCMD<1:0>	

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



### 39.15 RELATED APPLICATION NOTES

This section lists application notes that pertain to this section of the manual. These application notes may not be written specifically for the PIC24H Product Family, but the concepts are pertinent and could be used with modification and possible limitations. The current application notes related to the Oscillator (Part III) module include:

Title	Application Note #
PIC <sup>®</sup> Microcontroller Oscillator Design Guide	AN588
Low-Power Design using PIC <sup>®</sup> Microcontrollers	AN606
Crystal Oscillator Basics and Crystal Selection for rPIC <sup>®</sup> MCU and PIC <sup>®</sup> MCU Devices	AN826

**Note:** Please visit the Microchip web site ([www.microchip.com](http://www.microchip.com)) for additional Application Notes and code examples for the PIC24H family of devices.

## 39.16 REVISION HISTORY

### Revision A (November 2007)

This is the initial release of this document.

### Revision B (September 2008)

This revision incorporates the following updates:

- Figures:
  - Updated the label PLLCLK with Fvco<sup>(1)</sup> in Figure 39-1.
  - The existing DAC<sup>(1)</sup> label in Figure 39-1 is corrected as DAC<sup>(2)</sup>.
- Notes:
  - Added Note 1 to refer the Fvco values (see Figure 39-1).
  - Updated the existing Note 1 as Note 2.
- Registers:
  - The tuned frequencies for bit 5-0 in the OSCTUN: FRC Oscillator Tuning Register have been corrected (see Register 39-6).
- Additional minor corrections such as language and formatting updates are incorporated throughout the document.