Section 48. Oscillator (Part V)

HIGHLIGHTS

This section of the manual contains the following major topics:

48.1 INTRODUCTION

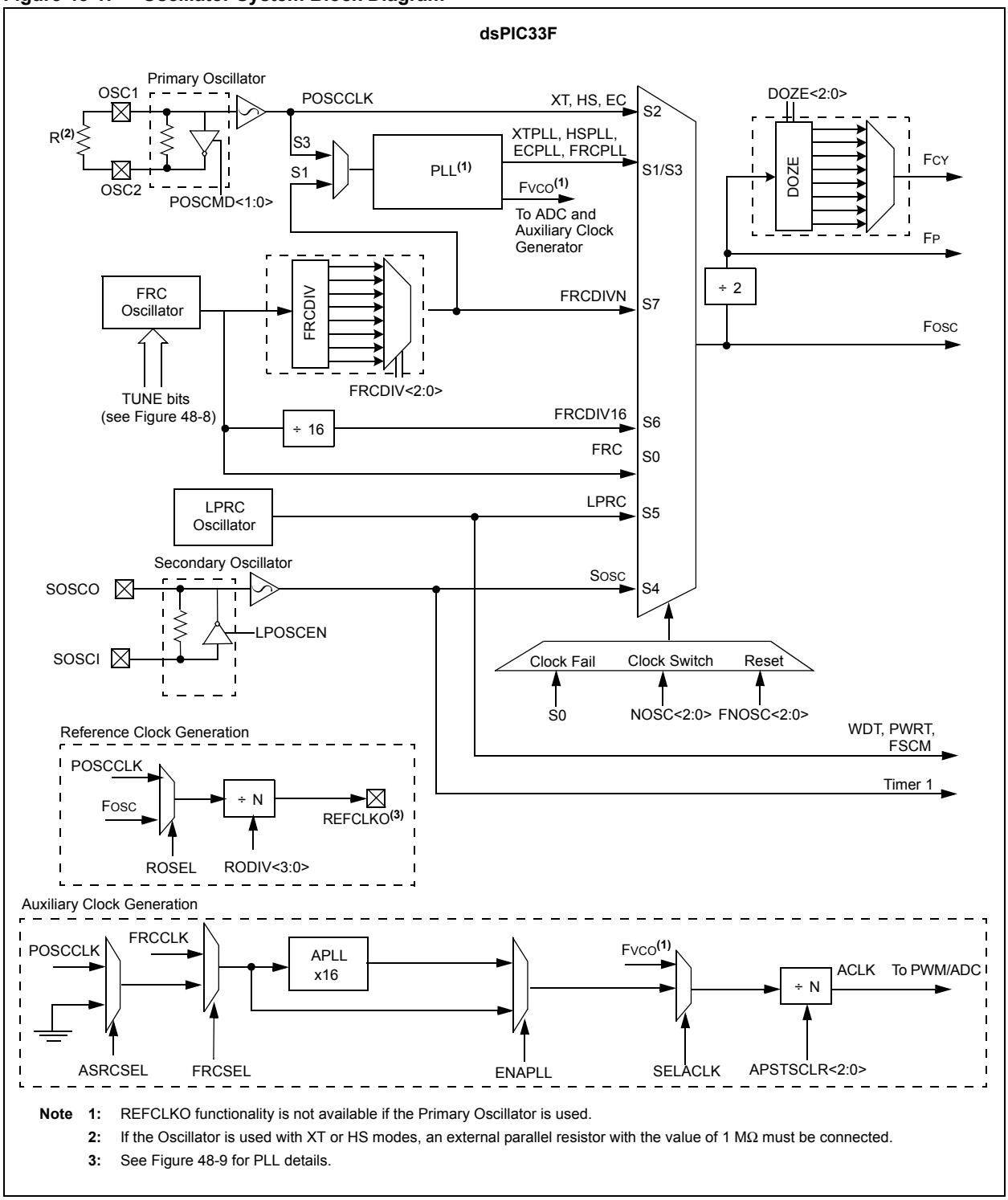
The dsPIC33F oscillator system includes the following characteristics:

- External and internal oscillator options
- On-chip Phase-Locked Loop (PLL) to boost internal operating frequency on select internal and external oscillator sources
- On-chip auxiliary PLL to boost internal operating frequency of the PWM and ADC
- On-the-fly clock switching between various clock sources
- Doze mode for system power savings
- Fail-Safe Clock Monitor (FSCM) that detects clock failure and permits safe application recovery or shutdown
- Nonvolatile Configuration bits for clock source selection

Figure 48-1 shows a block diagram of the dsPIC33F oscillator system.

 © 2009 Microchip Technology Inc.

Figure 48-1: Oscillator System Block Diagram



Note 1: REFCLKO functionality is not available if the Primary Oscillator is used.

2: If the Oscillator is used with XT or HS modes, an external parallel resistor with the value of 1 MΩ must be connected.

3: See Figure 48-9 for PLL details.

48.2 CPU CLOCKING

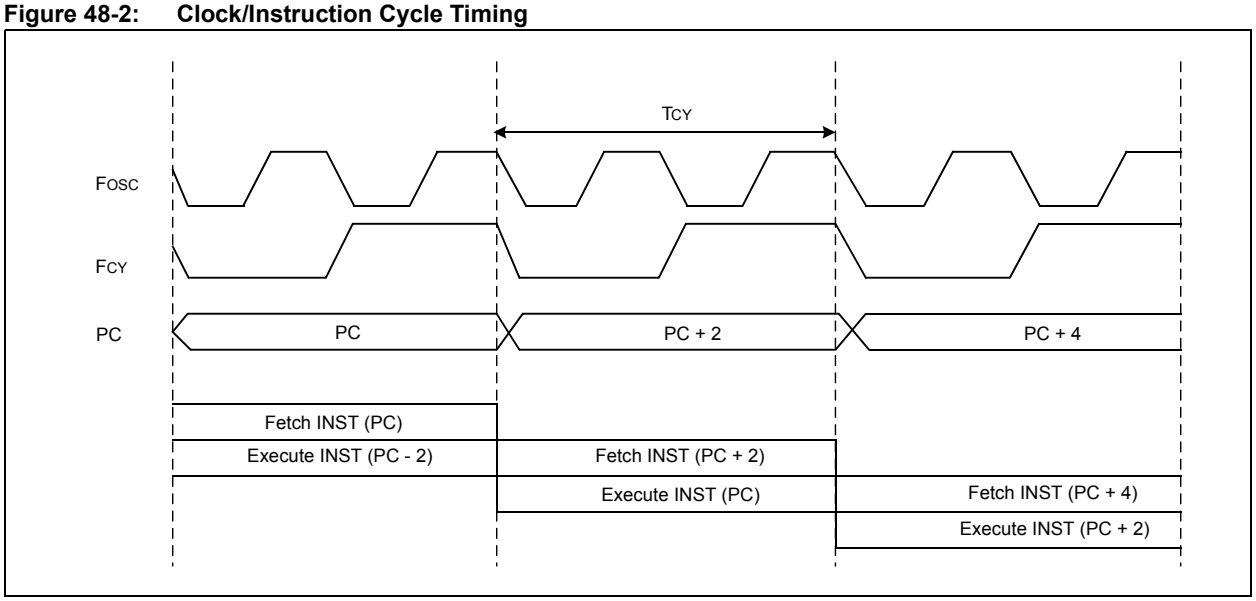
The system clock (FOSC) source can be provided by one of the following options:

- Primary Oscillator (POSC) on the OSC1 and OSC2 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 FCY. The timing diagram in Figure 48-2 shows the relationship between the system clock (FOSC), the instruction cycle clock (FCY) and the Program Counter (PC).

The internal instruction cycle clock (FCY) can be output on the OSC2 I/O pin, if the Primary Oscillator mode or the HS mode is not selected as the clock source (see **48.5 "Primary Oscillator (Posc)"**).

Figure 48-2: Clock/Instruction Cycle Timing



 © 2009 Microchip Technology Inc.

48.3 OSCILLATOR CONFIGURATION REGISTERS

Table 48-1 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 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, OSC2 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 POR. 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 OSC2 Pin Function (OSCIOFNC) Configuration bit in the Oscillator Configuration (FOSC<2>) register selects the OSC2 pin function, except in HS or XT mode. When the OSCIOFNC bit is unprogrammed ('1'), the FCY clock is output on the OSC2 pin. When the OSCIOFNC bit is programmed to ('0'), the OSC2 pin becomes a general purpose I/O pin.

Table 48-1: Configuration Bit Values for Clock Selection

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

Note 1: The OSC2 pin function is determined by the OSCIOFNC Configuration bit.

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

Register 48-1: FOSCSEL: Oscillator Source Selection Register

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

R/P	U	U-0	U	U	R/P	R/P	R/P
IESO	—	—	—	—	FNOSC<2:0>		
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 **IESO:** Internal External Start-up Option bit
1 = Start device with internal FRC, and then automatically switch to the user-selected oscillator source when ready
0 = Start device with user-selected oscillator source

bit 6 **Reserved:** Reserved bits must be programmed as ‘1’

bit 5 **Unimplemented:** Read as ‘0’

bit 4-3 **Reserved:** Reserved bits must be programmed as ‘1’

bit 2-0 **FNOSC<2:0>:** Initial Oscillator Source Selection bits
111 = Fast RC oscillator with Divide-by-N (FRCDIVN)
110 = Fast RC oscillator with Divide-by-16 (FRCDIV16)
101 = Low-Power RC oscillator (LPRC)
100 = Secondary oscillator (SOSC)
011 = Primary oscillator with PLL (XTPLL, HSPLL, ECPLL)
010 = Primary oscillator (XT, HS, EC)
001 = Fast RC oscillator with PLL (FRCPLL)
000 = Fast RC oscillator (FRC)

 © 2009 Microchip Technology Inc.

Register 48-2: 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)
1 = Allow only one reconfiguration
0 = Allow multiple reconfigurations

bit 4-3 **Reserved:** Reserved bits must be programmed as ‘1’

bit 2 **OSCIOFNC:** OSC2 Pin Function bit (except in XT and HS modes)
1 = OSC2 is clock output and instruction cycle (FCY) clock is output on OSC2 pin
0 = OSC2 is a general purpose digital I/O pin

bit 1-0 **POSCMD<1:0>:** Primary Oscillator Mode Selection bits
11 = Primary oscillator is disabled
10 = HS (High-Speed) Crystal oscillator mode
01 = XT (Crystal) oscillator mode
00 = EC (External Clock) mode

Note 1: The IOL1WAY bit is not available on all dsPIC33F devices. Refer to the specific device data sheet for more information.

48.4 SPECIAL FUNCTION REGISTERS (SFRs)

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 Doze mode and selects the PLL prescaler, PLL postscaler and FRC postscaler.

- **PLLFBD: PLL Feedback Divisor Register**

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

- **OSCTUN: Oscillator Tuning Register**

  The FRC Oscillator Tuning (OSCTUN) register is used to tune the frequency of the internal FRC oscillator from -12% to +11.625% (30 kHz steps) of the nominal frequency value. The nominal or tuned frequency of the FRC oscillator is expected to remain within ± 2% of the tuned value over the temperature and voltage variations of a particular device.

- **OSCTUN2: Oscillator Tuning Register 2**

  The OSCTUN and the OSCTUN2 registers enable the PWM to operate in Frequency Sequencing mode. These registers allow the user application to select a sequence of eight different FRC TUN values to vary the system frequency with each rollover of the primary PWM time base.

- **LFSR: Linear Feedback Shift Register**

  The Linear Feedback Shift Register is used to implement the Pseudo Random Clock Dither (PRCD) logic.

- **ACLKCON: Auxiliary Clock Control Register**

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

- **REFOCON: Reference Oscillator Control Register**

  The reference clock output provides a clock signal to any remappable pin (RPx). The reference clock can be either the external oscillator or the system clock.

Note:	The Oscillator Special Function Registers (OSCCON, CLKDIV, PLLFBD, OSCTUN, and ACLKCON) are reset only on Power-on Reset.

 © 2009 Microchip Technology Inc.

Register 48-3: OSCCON: Oscillator Control Register

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

R/W-0	U-0	R-0	R/W-0	R/C-0	R/W-0	R/W-0	R/W-0
CLKLOCK	—	LOCK	PRCDEN	CF	TSEQEN	LPOSCEN	OSWEN
bit 7							bit 0

Legend:	y = Value set from Configuration bits on POR		C = Clearable 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 (read-only)
111 = Fast RC oscillator (FRC) with Divide-by-N
110 = Fast RC oscillator (FRC) with Divide-by-16
101 = Low-Power RC oscillator (LPRC)
100 = Secondary oscillator (SOSC)
011 = Primary oscillator (XT, HS, EC) with PLL
010 = Primary oscillator (XT, HS, EC)
001 = Fast RC oscillator (FRC) with PLL
000 = Fast RC oscillator (FRC)

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

bit 10-8 **NOSC<2:0>:** New Oscillator Selection bits
111 = Fast RC oscillator (FRC) with Divide-by-N
110 = Fast RC oscillator (FRC) with Divide-by-16
101 = Low-Power RC oscillator (LPRC)
100 = Secondary oscillator (SOSC)
011 = Primary oscillator (XT, HS, EC) with PLL
010 = Primary oscillator (XT, HS, EC)
001 = Fast RC oscillator (FRC) with PLL
000 = Fast RC oscillator (FRC)

bit 7 **CLKLOCK:** Clock Lock Enable bit
If clock switching is enabled and FSCM is disabled, (FOSC<FCKSM> = 0b01):
1 = Clock switching is disabled, system clock source is locked
0 = Clock switching is enabled, system clock source can be modified by clock switching

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

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 **PRCDEN:** Pseudo Random Clock Dither Enable bit
1 = Pseudo random clock dither is enabled
0 = Pseudo random clock dither is disabled

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 **TSEQEN:** FRC Tune Sequencer Enable bit
1 = The TUN<3:0>, TSEQ1<3:0>, … , TSEQ7<3:0> bits in the OSCTUN and the OSCTUN2 registers sequentially tune the FRC oscillator. Each field being sequentially selected via the ROLL<2:0> signals from the PWM module.
0 = The TUN<3:0> bits in the OSCTUN register tune the FRC oscillator

Register 48-3: OSCCON: Oscillator Control Register (Continued)

bit 1 **LPOSCEN:** Secondary (LP) Oscillator 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

Note: Writes to this register requires an unlock sequence. For details and examples refer to **48.12 "Clock Switching"**.

 © 2009 Microchip Technology Inc.

Register 48-4: 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(1)	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

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 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
111 = FCY divided by 128
110 = FCY divided by 64
101 = FCY divided by 32
100 = FCY divided by 16
011 = FCY divided by 8 (default)
010 = FCY divided by 4
001 = FCY divided by 2
000 = FCY divided by 1

bit 11 **DOZEN:** DOZE Mode Enable bit(1)
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
111 = FRC divided by 256
110 = FRC divided by 64
101 = FRC divided by 32
100 = FRC divided by 16
011 = FRC divided by 8
010 = FRC divided by 4
001 = FRC divided by 2
000 = FRC divided by 1 (default)

bit 7-6 **PLLPOST<1:0>:** PLL VCO Output Divider Select bits (also denoted as 'N2', PLL postscaler)
11 = Output divided by 8
10 = Reserved
01 = Output divided by 4 (default)
00 = Output divided by 2

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)
11111 = Input divided by 33
•
•
•
00000 = Input divided by 2 (default)
00001 = Input divided by 3

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

2: Refer to **Section 9. "Watchdog Timer and Power-Saving modes"** (DS70196) in the "*dsPIC33F Family Reference Manual*" for more information on Doze mode.

Register 48-5: 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)

111111111 = 513
•
•
•
000110000 = 50 (default)
•
•
•
000000010 = 4
000000001 = 3
000000000 = 2

 © 2009 Microchip Technology Inc.

Register 48-6: OSCTUN: Oscillator Tuning Register

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TSEQ3<3:0>				TSEQ2<3:0>			
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TSEQ1<3:0>				TUN<3:0>			
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 15-12 **TSEQ3<3:0>:** Tune Sequence Value 3 bits
When PWM ROLL<2:0> = 011, this field is used to tune the FRC instead of TUN<3:0>

bit 11-8 **TSEQ2<3:0>:** Tune Sequence Value 2 bits
When PWM ROLL<2:0> = 010, this field is used to tune the FRC instead of TUN<3:0>

bit 7-4 **TSEQ1<3:0>:** Tune Sequence Value 1 bits
When PWM ROLL<2:0> = 001, this field is used to tune the FRC instead of TUN<3:0>

bit 3-0 **TUN<3:0>:** Specifies the user tuning capability for the internal fast RC oscillator. If the TSEQEN bit in the OSCCON register is set, this field, along with bits TSEQ1-TSEQ7, will sequentially tune the FRC oscillator.
0111 = Maximum frequency
0110 =
0101 =
0100 =
0011 =
0010 =
0001 =
0000 = Center frequency, oscillator is running at calibrated frequency
1111 =
1110 =
1101 =
1100 =
1011 =
1010 =
1001 =
1000 = Minimum frequency

Register 48-7: OSCTUN2: Oscillator Tuning Register 2

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TSEQ7<3:0>				TSEQ6<3:0>			
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TSEQ5<3:0>				TSEQ4<3:0>			
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 15-12 **TSEQ7<3:0>:** Tune Sequence Value 7 bits
When PWM ROLL<2:0> = 111, this field is used to tune the FRC instead of TUN<3:0>

bit 11-8 **TSEQ6<3:0>:** Tune Sequence Value 6 bits
When PWM ROLL<2:0> = 110, this field is used to tune the FRC instead of TUN<3:0>

bit 7-4 **TSEQ5<3:0>:** Tune Sequence Value 5 bits
When PWM ROLL<2:0> = 101, this field is used to tune the FRC instead of TUN<3:0>

bit 3-0 **TSEQ4<3:0>:** Tune Sequence Value 4 bits
When PWM ROLL<2:0> = 100, this field is used to tune the FRC instead of TUN<3:0>

Register 48-8: LFSR: Linear Feedback Shift Register

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	LFSR<14:8>						
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
LFSR<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 **Unimplemented:** Read as '0'

bit 14-8 **LFSR<14:8>:** Most Significant 7 bits of the pseudo random FRC trim value bits

bit 7-0 **LFSR<7:0>:** Least Significant 8 bits of the pseudo random FRC trim value bits

 © 2009 Microchip Technology Inc.

Register 48-9: ACLKCON: Auxiliary Clock Control Register

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

R/W-0	R/W-0	U-0	U-0	U-0	U-0	U-0	U-0
ASRCSEL	FRCSEL	—	—	—	—	—	—
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 **ENAPLL:** Auxiliary PLL Enable bit
1 = Auxiliary PLL is enabled
0 = Auxiliary PLL is disabled

bit 14 **APLLCK:** Auxiliary PLL Locked Status bit (read-only)
1 = Indicates that auxiliary PLL is in lock
0 = Indicates that auxiliary PLL is not in lock

bit 13 **SELACLK:** Select Clock Source for Auxiliary Clock Divider
1 = Auxiliary PLL or FRC or primary oscillator provides the source clock for the auxiliary clock divider
0 = Primary FVCO provides the source clock for auxiliary clock divider

bit 12-11 **Unimplemented:** Read as '0'

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 = No clock input is selected

bit 6 **FRCSEL:** Select Reference Clock Source for Auxiliary Clock
1 = Select FRC clock for clock source
0 = Input clock source is determined by ASRCSEL bit setting

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

Register 48-10: 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
ROON	—	ROSSLP	ROSEL	RODIV<3:0>			
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 **ROON:** Reference Oscillator Output Enable bit
1 = Reference oscillator output is enabled on REFO pin
0 = Reference oscillator output is disabled

bit 14 **Unimplemented:** Read as '0'

bit 13 **ROSSLP:** Reference Oscillator Run in Sleep bit
1 = Reference oscillator output continues to run in Sleep mode
0 = Reference oscillator output is disabled in Sleep mode

bit 12 **ROSEL:** Reference Oscillator Source Select bit
1 = Oscillator crystal is used as the reference clock
0 = System clock is used as the reference clock

bit 11-8 **RODIV<3:0>:** Reference Oscillator Divider bits
1111 = Divided by 32,768
1110 = Divided by 16,384
1101 = Divided by 8,192
1100 = Divided by 4,096
1011 = Divided by 2,048
1010 = Divided by 1,024
1001 = Divided by 512
1000 = Divided by 256
0111 = Divided by 128
0110 = Divided by 64
0101 = Divided by 32
0100 = Divided by 16
0011 = Divided by 8
0010 = Divided by 4
0001 = Divided by 2
0000 = Reference oscillator source

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

Note: The Reference oscillator module must be disabled (ROON = 0) before writing to these bits.

 © 2009 Microchip Technology Inc.

48.5 PRIMARY OSCILLATOR (POSC)

The Primary Oscillator (POSC) is available on the OSC1 and OSC2 pins of the dsPIC33F device family. 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 (FOSC) 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 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 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 to 64 MHz) and input on the OSC1 pin.

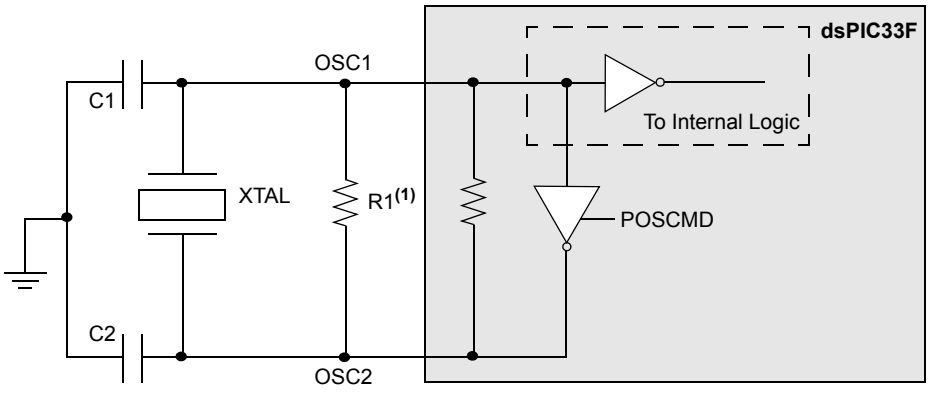
The Initial Oscillator Source Selection (FNOSC<2:0>) Configuration bits in the Oscillator Source Selection (FOSCSEL<2:0>) register specify the primary oscillator clock source at Power-on Reset. The Primary Oscillator Mode Selection (POSCMD<1:0>) Configuration bits in the Oscillator Configuration (FOSC<1:0>) register specify the Primary Oscillator mode. Table 48-2 lists the options selected by specific bit configurations, which are programmed at the time of device programming.

Table 48-2: 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 48-3 shows a recommended crystal oscillator circuit diagram for dsPIC33F 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 by using Equation 48-1.

Figure 48-3: Crystal or Ceramic Resonator Operation (XT or HS Oscillator Mode)



Note 1: If the Oscillator is used with XT or HS modes, an external parallel resistor with the value of 1 MΩ must be connected.

Equation 48-1: Crystal Load Capacitance

$$C_L = C_S + \frac{C1 \cdot C2}{C1 + C2}$$

where:

C_S is the stray capacitance

Assuming $C1 = C2$, Equation 48-2 provides the capacitor value (C1, C2) for a given load and stray capacitance.

Equation 48-2: External Capacitor for Crystal

$$C1 = C2 = 2 \cdot (C_L - C_S)$$

For additional information on crystal oscillators and their operation, refer to **48.16 "Related Application Notes"**.

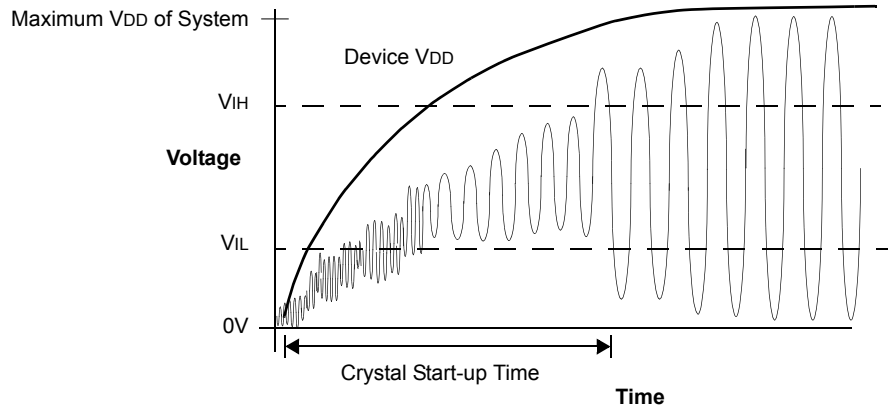
48.5.1 Oscillator Start-up Time

The oscillator starts oscillating as the device voltage increases from VSS. The time required for the oscillator to start oscillating depends on the following factors:

- Crystal/Resonator frequency
- Capacitor values used (C1 and C2 in Figure 48-3)
- Device VDD rise time
- System temperature
- Series resistor value and type if used
- Oscillator mode selection of a device (selects the gain of the internal oscillator inverter)
- Crystal quality
- Oscillator circuit layout
- System noise

Figure 48-4 shows a graph of a typical oscillator/resonator start-up.

Figure 48-4: Example of 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). 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 TOST.

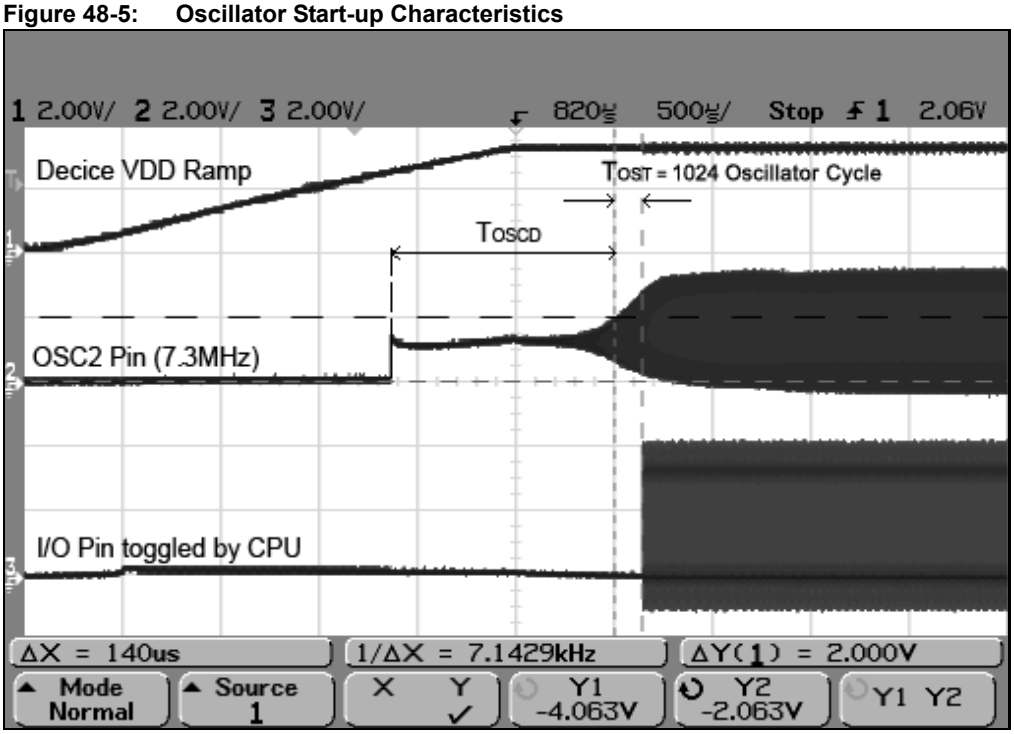
The amplitude of the oscillator signal must reach the VIL and VIH thresholds for the oscillator pins before the OST can begin to count cycles. The TOST interval is required every time the oscillator restarts (i.e., on POR, BOR, and wake-up from Sleep mode).

 © 2009 Microchip Technology Inc.

Once the primary oscillator is enabled, it takes a finite amount of time to start oscillating. This delay is denoted as TOSCD. After TOSCD, the OST timer takes 1024 clock cycles (TOST) to release the clock. The total delay for the clock to be ready is TOSCD + TOST. If the PLL is used, an additional delay is required for the PLL to lock (see **48.7 "Phase-Locked Loop (PLL)"**).

Primary oscillator start-up characteristics are illustrated in Figure 48-5, where the CPU starts toggling an I/O pin when it starts execution after the TOSCD + TOST interval.

Figure 48-5: Oscillator Start-up Characteristics



48.5.2 Primary Oscillator Pin Functionality

The primary oscillator pins (OSC1/OSC2) 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 OSC2 pin function. When FOSC<2> is '0', OSC2 is a general purpose digital I/O pin (see Figure 48-6). When FOSC<2> is '1', OSC2 is a clock output and the instruction cycle (FCY) clock is output on the OSC2 pin (see Figure 48-7).

The oscillator pin functions are listed in Table 48-3.

Table 48-3: Clock Pin Function Selection

Oscillator Source	OSCIOFNC<2> Value	POSCMD<1:0> Value	OSC1(1) Pin Function	OSC2(2) 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	OSC1	OSC2
XT (Crystal)	X	01	OSC1	OSC2
EC (External Clock)	1	00	OSC1	Clock Output (FCY)
EC (External Clock)	0	00	OSC1	Digital I/O

Note 1: The OSC1 pin function is determined by the Primary Oscillator Mode (POSCMD<1:0>) Configuration bits.

2: The OSC1 pin function is determined by the Primary Oscillator Mode (POSCMD<1:0>) and the OSC2 Pin Function (OSCIOFNC<2>) Configuration bits.

Figure 48-6: OSC2 Pin for Digital I/O (in EC Mode), FOSC<2> = 0

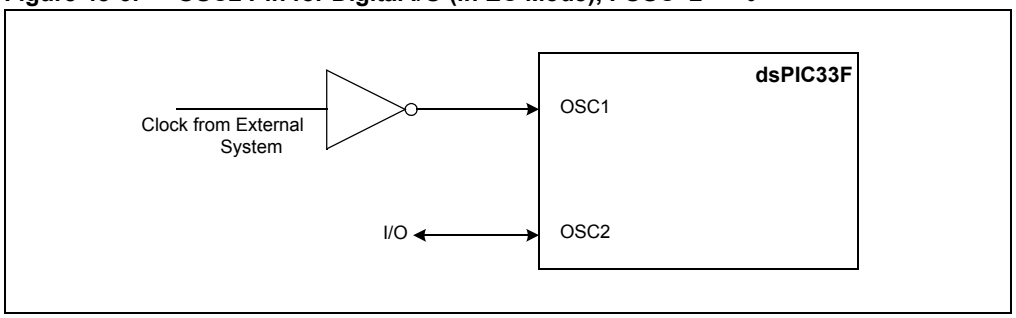
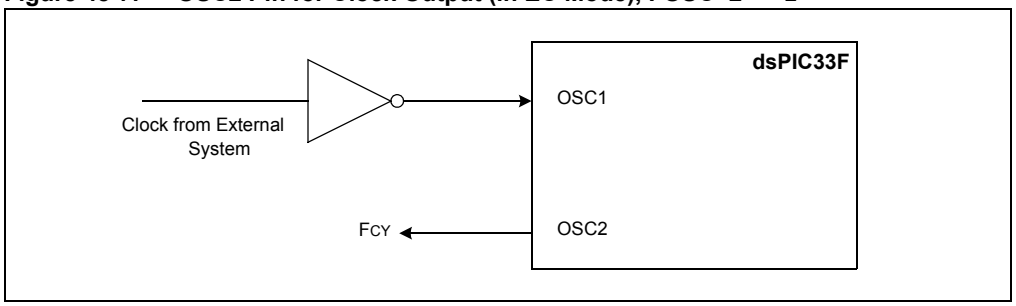


Figure 48-7: OSC2 Pin for Clock Output (in EC Mode), FOSC<2> = 1



 © 2009 Microchip Technology Inc.

48.6 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.

48.6.1 FRC Tuning

48.6.1.1 FREQUENCY SEQUENCING MODE

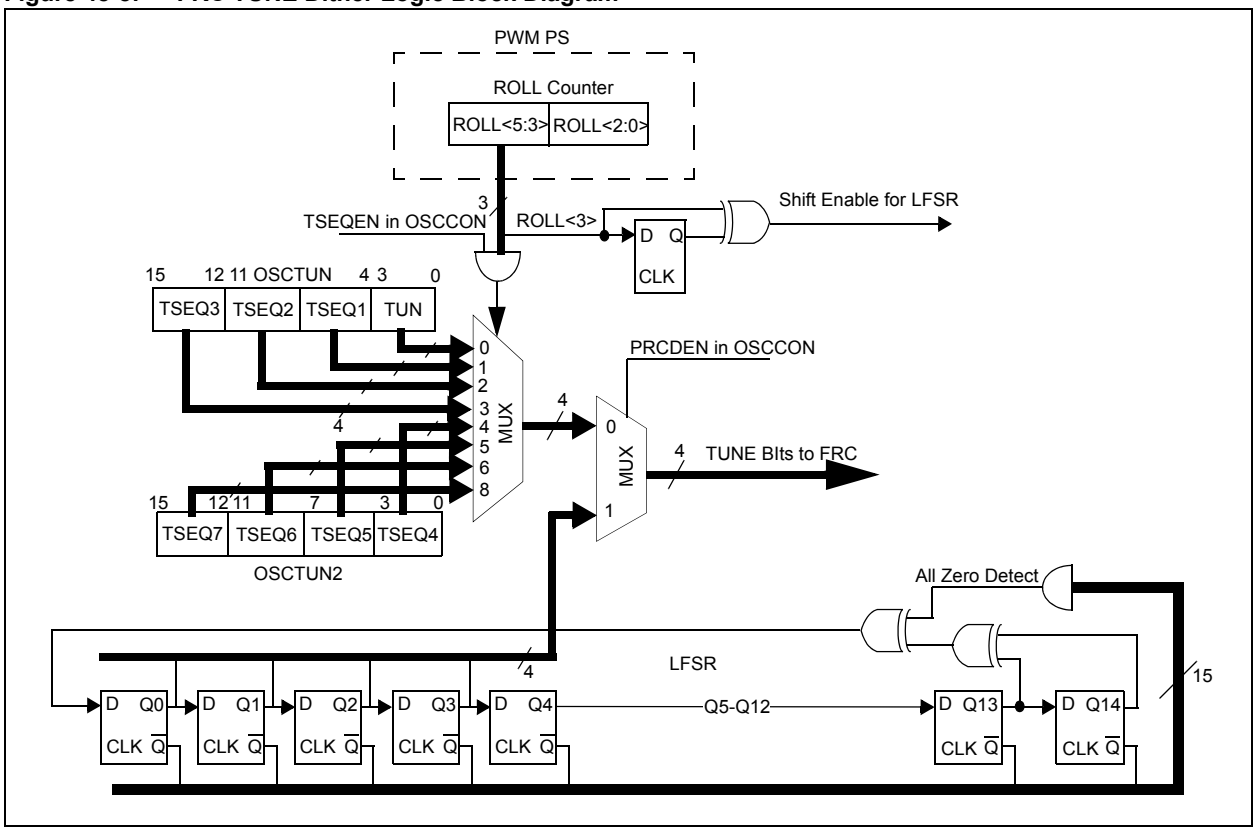
The Frequency Sequencing mode enables the PWM module to select a sequence of eight different FRC TUN values to vary the system frequency with each rollover of the primary PWM time base. The OSCTUN and the OSCTUN2 registers allow the user application to specify eight sequential tune values if the TSEQEN bit is set in the OSCCON register. If the TSEQEN bit is zero, only the TUN bits affect the FRC frequency.

A 4-bit wide multiplexer with eight sets of inputs selects the tuning value from the TUN and the TSEQx bit fields. The multiplexer is controlled by the ROLL<5:3> counter in the PWM module. The ROLL<5:3> counter increments every time the primary time base rolls over after reaching the period value.

48.6.1.2 PSEUDO RANDOM CLOCK DITHERING MODE

The Pseudo Random Clock Dither (PRCD) logic is implemented with a 15-bit Linear Feedback Shift Register (LFSR), which is a shift register with a few exclusive OR gates. The lower four bits of the LFSR provide the FRC TUNE bits. The PRCD feature is enabled by setting the PRCDEN bit in the OSCCON register. The LSFR is “clocked” (enabled to clock) once every time the ROLL<3> bit changes state, which occurs once every eight PWM cycles.

Figure 48-8: FRC TUNE Dither Logic Block Diagram



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 48-4. The Configuration bits are programmed at the time of device programming.

Table 48-4: 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)

48.6.2 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 eight settings, from 1:1 to 1:256, as shown in Figure 48-5.

Table 48-5: 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:

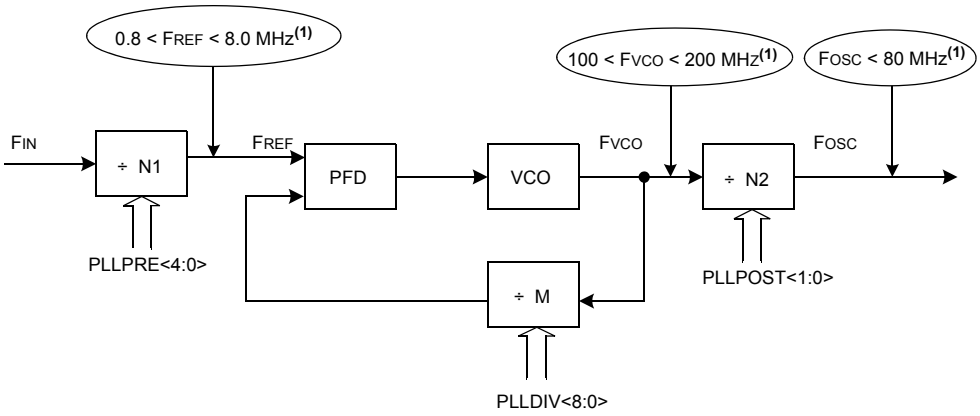
1. Switch the clock to a non-PLL mode (e.g., Internal FRC).
2. Make the necessary changes.
3. Switch the clock back to the PLL mode.

 © 2009 Microchip Technology Inc.

48.7 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 48-9.

Figure 48-9: dsPIC33F PLL Block Diagram



Note 1: This frequency range must be satisfied at all times.

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 (FREF) must be in the range of 0.8 to 8.0 MHz
- The VCO output frequency (FVCO) must be in the range of 100 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 (FIN) to meet the PFD input frequency range of 0.8 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 (FVCO) for feedback to the PFD. The VCO frequency (FVCO) is 'M' times the input reference clock (FREF).

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 (FOSC) to 80 MHz.

Equation 48-3 shows the relation between the input frequency (FIN) and the output frequency (FOSC).

Equation 48-3: Fosc Calculation

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

where:

$N1 = PLLPRE + 2$

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

$M = PLLDIV + 2$

Equation 48-4 shows the relation between the input frequency (FIN) and the VCO frequency (FVCO).

Equation 48-4: Fvco Calculation

$$FVCO = FIN \times \left(\frac{M}{N1}\right) = FIN \times \left(\frac{(PLLDIV + 2)}{(PLLPRE + 2)}\right)$$

48.7.1 Input Clock Limitation at Start-up for PLL Mode

Table 48-6 gives the default values of the PLL Prescaler, PLL Postscaler and PLL Feedback Divisor Configuration bits at Power-on Reset.

Table 48-6: 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 (FIN) and PFD input frequency (FREF), and the VCO frequency (FVCO) and system clock frequency (FOSC) at Power-on Reset.

Equation 48-5: FREF at Power-on Reset

$$FREF = FIN\left(\frac{1}{N1}\right) = 0.5(FIN)$$

Equation 48-6: FVCO at Power-on Reset

$$Fvco = FIN\left(\frac{M}{N1}\right) = FIN\left(\frac{50}{2}\right) = 25(FIN)$$

Equation 48-7: FOSC at Power-on Reset

$$FOSC = FIN\left(\frac{M}{N1 \cdot N2}\right) = 6.25(FIN)$$

Given the above equations at Power-on Reset, the input frequency (FIN) to the PLL module must be limited to 4 MHz < FIN < 8 MHz to comply with the VCO output frequency requirement (100M < FVCO < 200M) 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 MHz < FIN < 8 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, follow the procedure given below:

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 (FREF) must be in the range of 0.8 MHz to 8.0 MHz
   - The VCO output frequency (FVCO) must be in the range of 100 MHz to 200 MHz
3. Switch the clock to the PLL mode in software.

 © 2009 Microchip Technology Inc.

48.7.2 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 (TLOCK) to synchronize to the new settings.

TLOCK is applied when PLL is selected as the clock source at Power-on Reset, or during a clock switching operation. The value of TLOCK is relative to the time at which the clock is available to the PLL input. For example, with the primary oscillator, TLOCK starts after the OST delay. Refer to **48.5.1 "Oscillator Start-up Time"** for detailed information about oscillator start-up delay. Refer to specific device data sheet for typical TLOCK values.

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.

48.7.2.1 SETUP FOR USING PLL WITH PRIMARY OSCILLATOR (Posc)

The following procedure 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 Fosc = 2 • FCY = 80 MHz.
2. Ensure that the default reset values of PLLPRE, PLLPOST and PLLDIV meet the PLL and user requirements:
   - FREF = 0.5 • FIN = 5 MHz
   - FOSC = 6.25 • FIN = 62.5 MHz
   - FVCO = 25 • FIN = 250 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 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) = 5 MHz
   c) Select the PLL feedback divisor to generate the required VCO output frequency based on the PFD input frequency:
      - Fvco = FREF • M
      - M = Fvco/FREF = 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 upon in the previous steps, and then perform a clock switch to the PLL mode.

Example 48-1 illustrates the code sequence for using PLL with the primary oscillator. See **48.12 "Clock Switching"** for a clock switching code example.

Example 48-1: 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, and PLL divisor
PLLFBD=30;             // M = 32
CLKDIVbits.PLLPRE=0;   // N1 = 2
CLKDIVbits.PLLPOST=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) {};

}
```

48.7.2.2 SETUP FOR USING PLL WITH 7.37 MHz INTERNAL FRC

The following procedure 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 FOSC = 2 • FCY = 80 MHz.
2. Ensure that the default Reset values of PLLPRE, PLLPOST and PLLDIV meet the PLL and user requirements:
   - FREF = 0.5 • FIN = 3.68 MHz
   - FOSC = 6.25 • FIN = 46 MHz
   - FVCO = 25 • FIN = 184 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:
      - FVCO = FREF • M
      - M = FVCO/FREF = 43

 © 2009 Microchip Technology Inc.

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 48-2 illustrates the code sequence for using PLL with a 7.37 MHz Internal FRC. See **48.12 "Clock Switching"** for a clock switching code example.

Example 48-2: 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, and PLL divisor
PLLFBD = 41;           // M = 43
CLKDIVbits.PLLPRE=0;   // N1 = 2
CLKDIVbits.PLLPOST=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) {};

}
```

48.8 LOW-POWER SECONDARY OSCILLATOR (SOSC)

The Low-Power Secondary Oscillator (SOSC) enables a 32.768 kHz watch crystal to be attached to the dsPIC33F device as a secondary crystal clock source for low-power operation. It uses the SOSCI and SOSCO pins. The low-power secondary oscillator can also drive Timer1 for Real-Time Clock (RTC) application.

48.8.1 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.
- the user application 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.

48.8.2 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 **48.5.1 "Oscillator Start-up Time"** for details.

48.8.3 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 low-power operation. Returning to the faster main oscillator still requires an oscillator start-up time if it is a crystal type source (see **48.5.1 "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 and LPRC oscillator) are shutdown, 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.

 © 2009 Microchip Technology Inc.

48.9 LOW-POWER RC (LPRC) 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” section in the specific device data sheet for details.

48.9.1 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
- The user application initiates a clock switch to the LPRC oscillator for low-power operation

48.9.2 Enabling the LPRC Oscillator

The LPRC oscillator is the clock source for the PWRT, WDT and FSCM. It 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 contain a non-zero value.

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.

48.9.3 LPRC Oscillator Start-up Delay

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

48.10 AUXILIARY PLL MODULE FOR ADC AND PWM SYSTEM CLOCK

The auxiliary PLL can be used to provide a high-speed clock to peripherals such as the PWM and the ADC. The ACLKCON register selects the reference clock and output dividers for obtaining the necessary auxiliary clock for the PWM and ADC modules. The auxiliary clock for the PWM and ADC can be either:

- Internal FRC (7.37 MHz nominal)
- Primary Oscillator
- FRC with PLL
- Primary Oscillator with PLL
- Auxiliary PLL

48.10.1 Enabling the Auxiliary PLL

To enable the auxiliary PLL, the following steps must be performed:

1. Select the reference clock for the auxiliary PLL by setting the ASRCSEL bit (ACLKCON<7>) for the primary oscillator or by setting the FRCSEL bit (ACLKCON<6>) for the FRC.
2. Enable the auxiliary PLL by setting the ENAPLL bit (ACLKCON<15>).
3. Select the clock source for the auxiliary clock output divider by setting the SELACLK bit (ACLKCON<13>).
4. Select the appropriate clock divider by setting the APSTSCLR<2:0> bits (ACLKCON<10:8>).
5. Ensure that the auxiliary PLL has locked and is ready for operation. This is done by polling the APLLCK bit (ACLKCON<14>).

Example 48-3 illustrates the code sequence that set up the auxiliary PLL for 120 MHz using the internal FRC as a clock reference.

Example 48-3: Enabling the Auxiliary PLL

```
ACLKCONbits.FRCSEL = 1;    /* Internal FRC is a clock source for auxiliary PLL*/
ACLKCONbits.ENAPLL = 1;    /* APLL is enabled */
ACLKCONbits.SELACLK = 1;   /* Auxiliary PLL provides the source clock for the */
                           /* clock divider */
ACLKCONbits.APSTSCLR = 7; /* Auxiliary Clock Output Divider is Divide by 1 */

while(ACLKCONbits.APLLCK != 1){}; /* Wait for auxiliary PLL to lock */

/* Given a 7.5 MHz input from the FRC, the auxiliary clock frequency for the*/
/* ADC and PWM modules is 7.5 MHz * 16 = 120 MHz */
```

Note:	If the primary PLL is used as a source for the auxiliary clock, the primary PLL should be configured up to a maximum operation of 30 MIPS or less.

 © 2009 Microchip Technology Inc.

48.10.2 Auxiliary Clock Output Divider

The Auxiliary Clock Output Divider (APSTSCLR<2:0>) bits in the Auxiliary Clock Control register (ACLKCON<10:8>) divide the auxiliary clock, which allow a lower frequency to be chosen. These bits allow for eight postscaler settings, from 1:1 to 1:256, as shown in Table 48-7.

Table 48-7: PLL Mode Defaults

Register	Bit Field
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)

48.11 FAIL-SAFE CLOCK MONITOR (FSCM)

The Fail-Safe Clock Monitor (FSCM) 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.

Note:	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.

48.11.1 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 oscillator is selected as the system clock.

Refer to **Section 8. "Reset"** (DS70192) in the *"dsPIC33F Family Reference Manual"* for additional information.

Note:	Refer to the "Electrical Characteristics" section of the specific device data sheet for TFSCM values.

48.11.2 FSCM and WDT

The FSCM and 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.

 © 2009 Microchip Technology Inc.

48.12 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.
- FSCM automatically switches to internal FRC oscillator on a clock failure.
- The user application 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 the proper make-before-break sequence is executed (i.e., 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, dsPIC33F devices have a safeguard lock built into the switch process (i.e., the OSCCON register is write-protected during clock switching).

48.12.1 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 48-8).

Table 48-8: 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.

48.12.2 Clock Switch Sequence

The recommended procedure for a clock switch is as follows:

1. Read the COSC bits (OSCCON<14:12>) to determine the current oscillator source (if this 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/secondary) must be turned on, the hardware waits TOSCD until the crystal starts oscillating and then until TOST 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). The timing of the transition between clock sources is as shown in Figure 48-10.

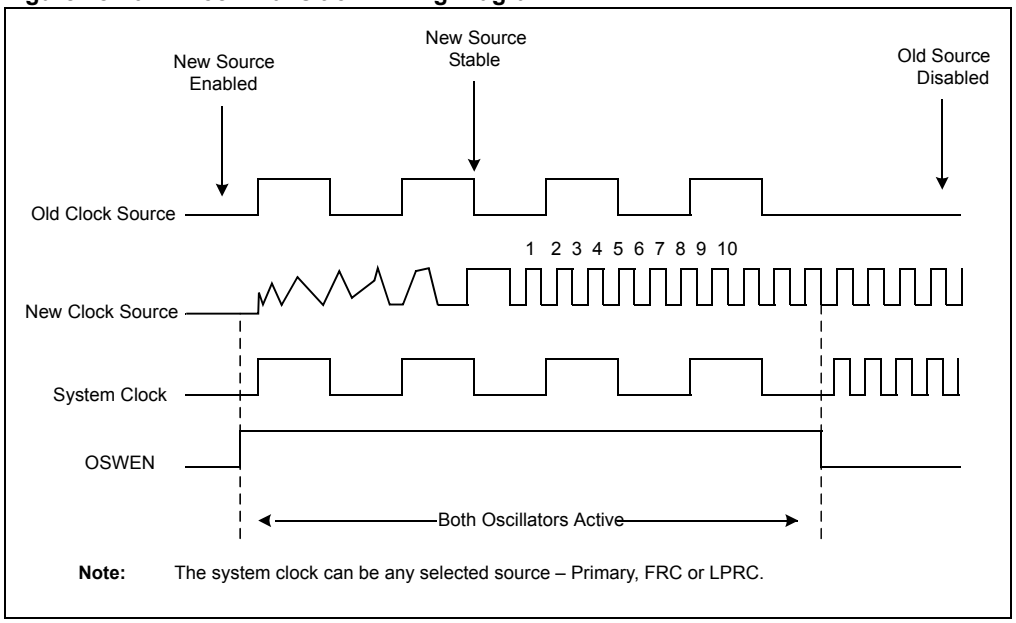
Note 1: Clock switching between XT, HS and EC primary oscillator modes is not possible without reprogramming the device.

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 48-10: Clock Transition Timing Diagram



Note: The system clock can be any selected source – Primary, FRC or LPRC.

 © 2009 Microchip Technology Inc.

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 is 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 48-4 illustrates the code sequence for unlocking the OSCCON register and switching from FRC with PLL clock to the LPRC clock source.

Example 48-4: Code Example for Clock Switching

```
;Place the new oscillator selection (NOSC=0b101) in W0
MOV #0x15,WREG

;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   WREG, OSCCONH; Request clock switching by setting the OSWEN bit
wait:
        btsc    OSCCONL,#OSWEN
        bra     wait
```

48.12.3 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 will result in slow device operation.

48.12.4 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 48-5. 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 48-5: 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
```

48.12.5 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.

 © 2009 Microchip Technology Inc.

48.13 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 application-selected oscillator source or to initially start with the internal FRC and then switch automatically to the user application-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 Clock Switching mode configuration bits FCKSM<1:0> (FOSC<7:6>).

Two-Speed Start-up is particularly useful when an external oscillator is selected by the FOSCSEL Configuration bits (FOSC<2:0) and a crystal-based oscillator has a longer start-up time.

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). The device continues to operate in this mode until the specified external oscillator source becomes stable, and at the same time 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.

48.14 REFERENCE CLOCK OUTPUT

The reference clock output provides a clock signal to any remappable pin (RPx). The reference clock can be either the external oscillator or the system clock.

The ROSEL bit in the Reference Oscillator Control (REFOCON) register selects between the external oscillator and the system clock. The RODIV bits in the REFOCON register scale the reference clock to a desired clock output.

Figure 48-1 shows a block diagram for the reference clock. See the REFOCON register (Register 48-10) for the bits associated with the reference clock output. Refer to the specific device data sheet regarding peripheral remapping.

48.15 REGISTER MAPS

Table 48-9 maps the bit functions for the Oscillator Special Function Control registers. Table 48-10 maps the bit functions for the Oscillator Configuration registers.

Table 48-9: Oscillator Special Function Control Registers

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	—	COSC<2:0>			—	NOSC<2:0>			CLKLOCK	—	LOCK	PRCDEN	CF	TSEQEN	LPOSCEN	OSWEN	0000(1)
CLKDIV	ROI	DOZE<2:0>			DOZEN	FRCDIV<2:0>			PLLPOST<1:0>		—	PLLPRE<4:0>					3040
PLLFBD	—	—	—	—	—	—	—	PLLDIV<8:0>									0030
OSCTUN	TSEQ3<3:0>				TSEQ2<3:0>				TSEQ1<3:0>				TUN<3:0>				0000
OSCTUN2	TSEQ7<3:0>				TSEQ6<3:0>				TSEQ5<3:0>				TSEQ4<3:0>				0000
LFSR	—	LFSR<14:8>							LFSR<7:0>								0000
ACLKCON	ENAPLL	APLLCK	SELACLK	—	—	APSTSCLR<2:0>			ASRCSEL	FRCSEL	—	—	—	—	—	—	0000
REFOCON	ROON	—	ROSSLP	ROSEL	RODIV<3:0>				—	—	—	—	—	—	—	—	0000

Legend: x = unknown value on Reset, — = unimplemented, read as '0'. Reset values are shown in hexadecimal.
Note 1: OSCCON register Reset values are dependent on the FOSCSEL Configuration bits and type of Reset.

Table 48-10: Oscillator Configuration Registers

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
FOSCSEL	—	—	—	—	—	—	—	—	IESO	—	—	—	—	FNOSC<2:0>			xxxx(2)
FOSC	—	—	—	—	—	—	—	—	FCKSM<1:0>		IOL1WAY(1)	—	—	OSCIOFNC	POSCMD<1:0>		xxxx

Legend: x = unknown value on Reset, — = unimplemented, read as '0'. Reset values are shown in hexadecimal.
Note 1: The IOL1WAY bit is not available on all dsPIC33F devices. Refer to the specific device data sheet for more information.
2: Configuration bits are programmed during device programming and it retains the programmed values on Reset.

Preliminary
© 2009 Microchip Technology Inc.

48.16 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 dsPIC33F 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 V) module include:

Title	Application Note #
PICmicro® Microcontroller Oscillator Design Guide	AN588
Low-Power Design using PICmicro® Microcontrollers	AN606
Crystal Oscillator Basics and Crystal Selection for rfPIC® and PICmicro® Devices	AN826

Note: Please visit the Microchip web site (www.microchip.com) for additional Application Notes and code examples for the dsPIC33F family of devices.

48.17 REVISION HISTORY

Revision A (June 2009)

This is the initial released revision of this document.

 © 2009 Microchip Technology Inc.