Oscillator Module with High-Speed PLL

HIGHLIGHTS

This section of the manual contains the following major topics:

Note: This family reference manual section is meant to serve as a complement to device data sheets. This document applies to all dsPIC33/PIC24 devices. Some dsPIC33/PIC24 devices are dual core and contain both a Master and Slave CPU core. For single core dsPIC33/PIC24 devices, disregard any Slave-specific references.

Please consult the note at the beginning of the **"Oscillator Configuration"** chapter in the current device data sheet to check whether this document supports the device you are using.

Device data sheets and family reference manual sections are available for download from the Microchip Worldwide Web site at: http://www.microchip.com.

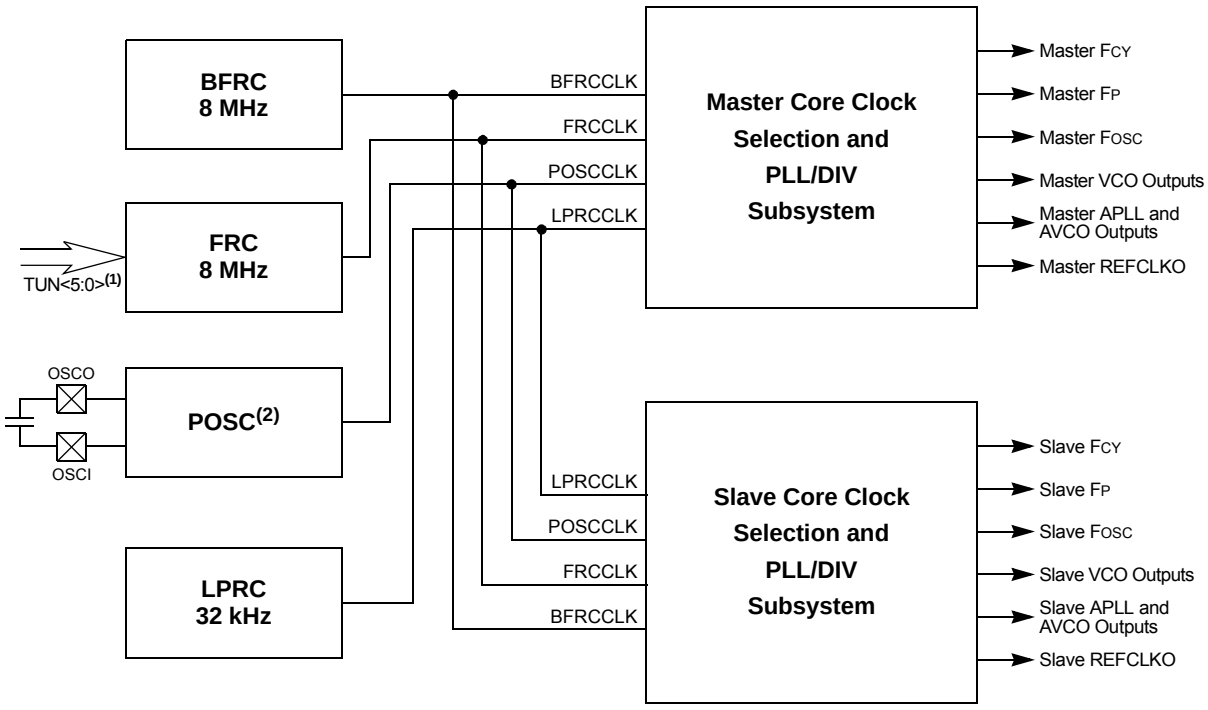
1.0 INTRODUCTION

The Oscillator Module with High-Speed PLL includes these characteristics:

- Master and Slave Core Subsystems
- Internal and External Oscillator Sources Shared between Master and Slave Cores
- Master and Slave Independent On-Chip Phase-Locked Loop (PLL) to Boost Internal Operating Frequency on Select Internal and External Oscillator Sources
- Master and Slave Independent Auxiliary PLL (APLL) Clock Generator to Boost Operating Frequency for Peripherals.
- Master and Slave Independent Doze mode for System Power Savings
- Master and Slave Independent Scalable Reference Clock Output (REFCLKO)
- On-the-Fly Clock Switching between Various Clock Sources
- Fail-Safe Clock Monitoring (FSCM) that Detects Clock Failure and Permits Safe Application Recovery or Shutdown

A block diagram of the dsPIC33/PIC24 shared core oscillator system is shown in Figure 1-1.

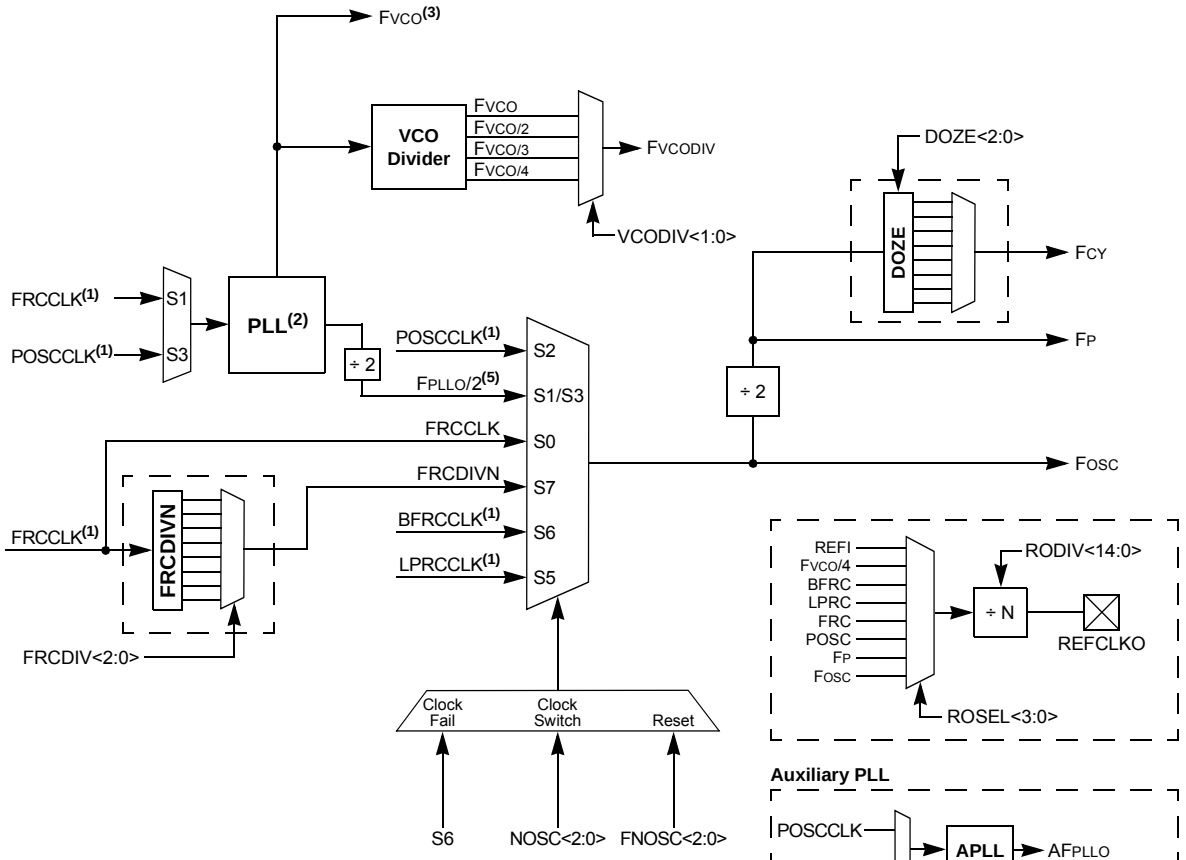
Figure 1-1: Master and Slave Core Shared Clock Sources Block Diagram



Note 1: FRC Oscillator tuning bits are configured in the Master core OSCTUN register.

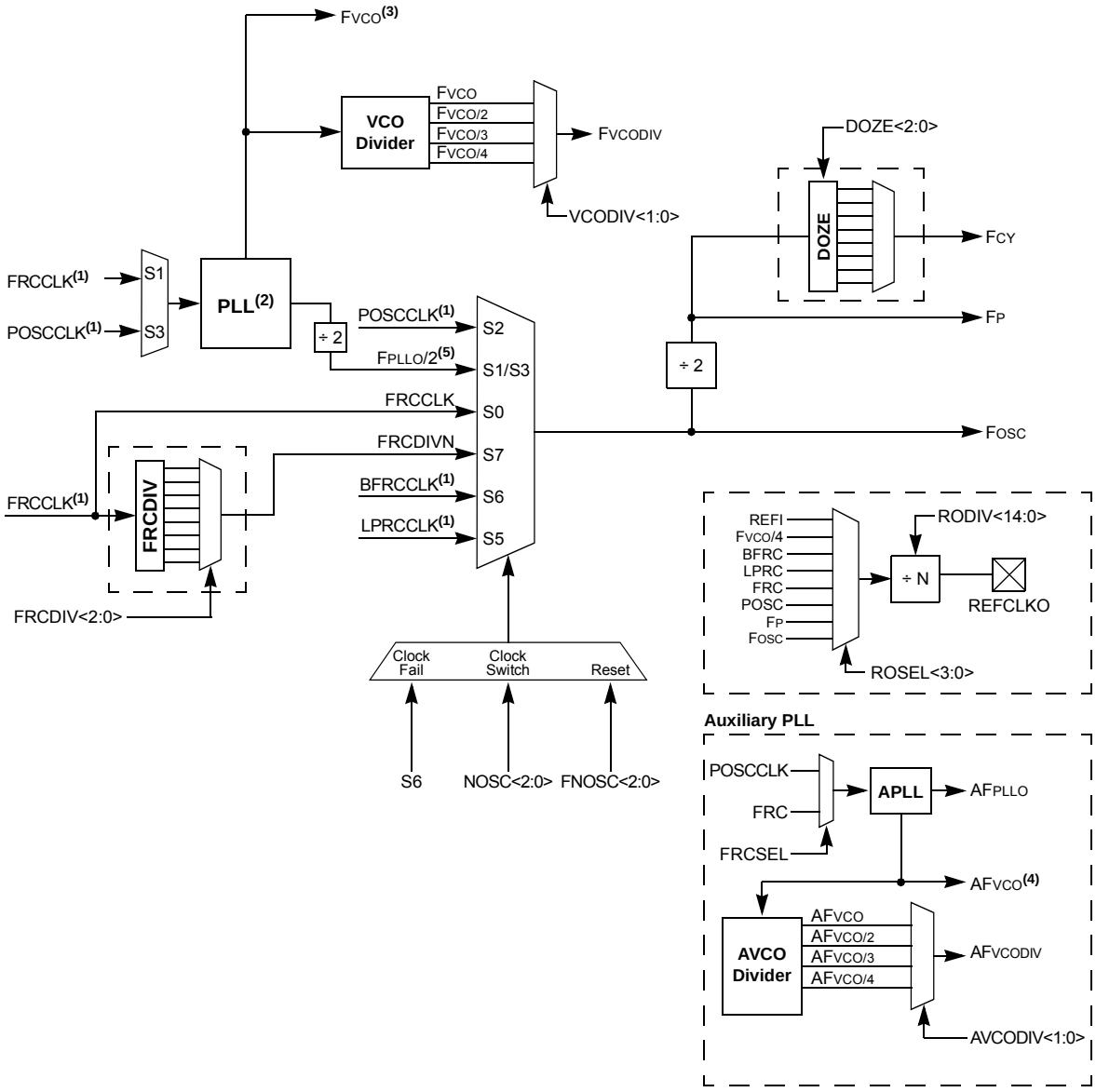
2: POSC is configured through the POSCMD<1:0> bits in the Master FOSC Configuration register.

Figure 1-2: Master Core Oscillator Subsystem



Note 1: From Master and Slave core shared oscillator source.
2: See Figure 8-1 for details of the PLL module.
3: See Figure 8-1 for the source of FVCO.
4: See Figure 10-1 for the source of AFvco.
5: XTPLL, HSPLL, ECPLL, FRCPLL (FPLLO).

Figure 1-3: Slave Core Oscillator Subsystem



Note 1: From Master and Slave core shared oscillator source.
2: See Figure 9-1 for details of the PLL module.
3: See Figure 9-1 for the source of FVCO.
4: See Figure 11-1 for the source of AFVCO.
5: XTPLL, HSPLL, ECPLL, FRCPLL (FPLLO).

2.0 CPU CLOCKING

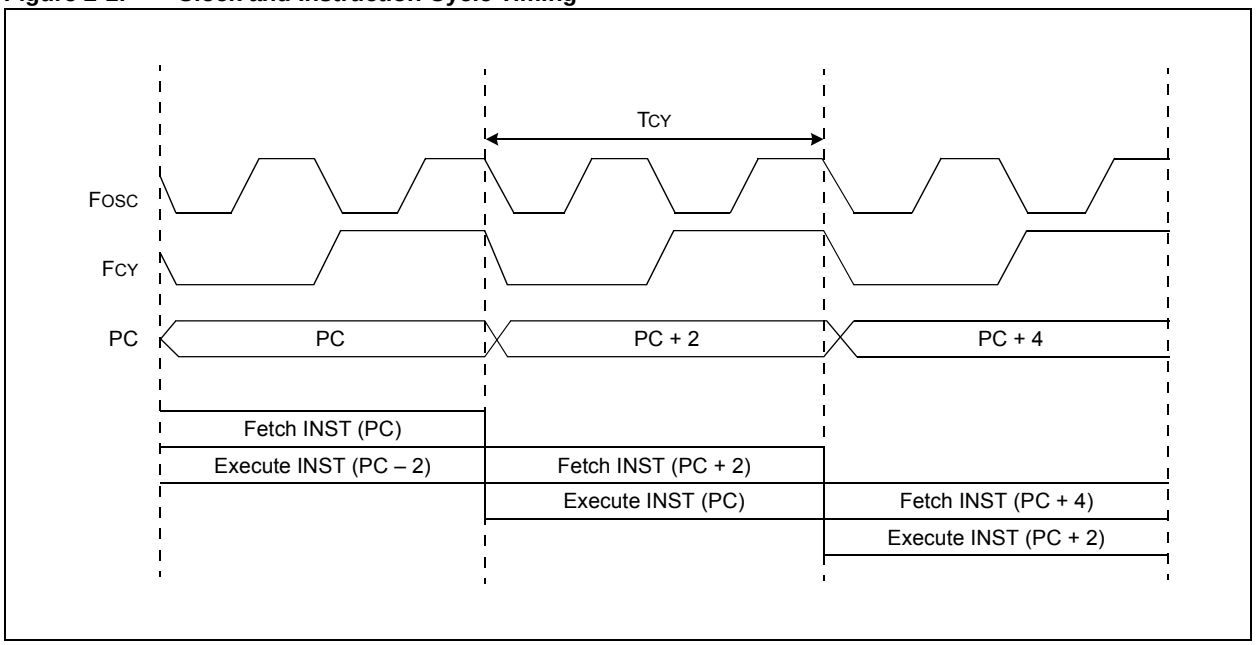
While the Master and Slave subsystems share access to a single set of oscillator sources, all other clocking logic is implemented individually. The Master and Slave core can be configured independently to use any of the following clock configurations:

- Primary Oscillator (POSC) on the OSC1 and OSC2 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 (FRCPLL)
- Backup Internal Fast RC Oscillator (BFRC)

Each core's system clock source is divided by two to produce the internal instruction cycle clock. In this document, the instruction cycle clock is denoted by FCY. The timing diagram in Figure 2-1 illustrates 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. For more information, see **Section 5.0 "Primary Oscillator (POSC)"**.

Figure 2-1: Clock and Instruction Cycle Timing



3.0 OSCILLATOR CONFIGURATION REGISTERS

3.1 Master Oscillator Configuration Registers

- **FOSCSEL: Master Oscillator Source Selection Register**

  FOSCSEL selects the Master core's initial oscillator source and start-up option. FOSCSEL contains the following Configuration bits:

  - The FNOSC<2:0> Configuration bits in the Master Oscillator Source Selection register (FOSCSEL<2:0>) determine the clock source that is used at a Power-on Reset (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.

- **FOSC: Master Oscillator Configuration Register**

  FOSC configures the Primary Oscillator mode, OSC2 pin function, Peripheral Pin Select (PPS) and the Fail-Safe and Clock Switching modes. FOSC contains the following Configuration bits:

  - The POSCMD<1:0> (FOSC<1:0>) Configuration bits select the operation mode of the POSC that is available to both the Master and Slave cores.
  - The OSCIOFNC (FOSC<2>) Configuration bit selects the OSC2 pin function, except in High-Speed or Medium Speed Oscillator (XT) mode.

    If OSCIOFNC is unprogrammed ('1'), the Master FCY clock is output on the OSC2 pin.

    If OSCIOFNC is programmed ('0'), the OSC2 pin becomes a general purpose I/O pin.

Table 3-1 lists the configuration settings that select the device's Master core oscillator source and operating mode at a POR.

Table 3-1: Configuration Bit Values for Clock Selection

Oscillator Source	Oscillator Mode	FNOSC<2:0> Value	POSCMD<1:0>(3) Value	Notes
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	Reserved	100	xx	
S5	Low-Power RC Oscillator (LPRC)	101	xx	1
S6	Backup FRC (BFRC)	110	xx	1
S7	Fast RC Oscillator with ÷ N Divider (FRCDIVN)	111	xx	1, 2

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

2: This is the default oscillator mode for an unprogrammed (erased) device.

3: The POSCMD<1:0> bits are only available in the Master Oscillator Configuration register, FOSC.

Register 3-1: FOSCSEL: Master Oscillator Source Selection Register

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 23							bit 16

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 15							bit 8

R/PO-1	U-1	U-1	U-1	U-1	R/PO-1	R/PO-1	R/PO-1
IESO	—	—	—	—	FNOSC2	FNOSC1	FNOSC0
bit 7							bit 0

Legend:	PO = Program Once bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘1’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 23-8 **Unimplemented:** Read as ‘1’

bit 7 **IESO:** Internal External Start-up Option bit
- 1 = Starts up device with the Internal FRC Oscillator, then automatically switches to the user-selected oscillator source when ready
- 0 = Starts up device with the user-selected oscillator source

bit 6-3 **Unimplemented:** Read as ‘1’

bit 2-0 **FNOSC<2:0>:** Initial Oscillator Source Selection bits
- 111 = Fast RC Oscillator with Divide-by-N (FRCDIVN)
- 110 = Backup FRC Oscillator (BFRC)
- 101 = Low-Power RC Oscillator (LPRC)
- 100 = Reserved
- 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)

Register 3-2: FOSC: Master Oscillator Configuration Register

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 23							bit 16

U-1	U-1	U-1	R/PO-1	R/PO-1	R/PO-1	U-1	R/PO-1
—	—	—	XTBST	XTCFG1	XTCFG0	—	PLLKEN
bit 15							bit 8

R/PO-1	R/PO-1	U-1	U-1	U-1	R/PO-1	R/PO-1	R/PO-1
FCKSM1	FCKSM0	—	—	—	OSCIOFNC(1)	POSCMD1(2)	POSCMD0(2)
bit 7							bit 0

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

bit 23-13 **Unimplemented:** Read as '1'

bit 12 **XTBST:** Kick-Starter Programmability for Oscillator bit
1 = Boosts the kick-start
0 = Default kick-start

bit 11-10 **XTCFG<1:0>:** Crystal Oscillator Drive Select bits
Current gain programmability for oscillator (output drive).
11 = Gain3 (use for 24-32 MHz crystals)
10 = Gain2 (use for 16-24 MHz crystals)
11 = Gain1 (use for 8-16 MHz crystals)
11 = Gain0 (use for 4-8 MHz crystals)

bit 9 **Unimplemented:** Read as '1'

bit 8 **PLLKEN:** PLL Lock Enable bit
1 = Source for PLL lock signal is the lock detect
0 = Source for PLL lock signal is the PLL enable signal

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 5-3 **Unimplemented:** Read as '1'

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

bit 1-0 **POSCMD<1:0>:** Primary Oscillator Mode Selection bits(2)
11 = Primary Oscillator is disabled
10 = HS Crystal Oscillator mode (10 MHz to 32 MHz)
01 = XT Crystal Oscillator mode (3.5 MHz to 10 MHz)
00 = EC (External Clock) mode (0 MHz to 64 MHz)

Note 1: If both the Master core OSCIOFNC and Slave core S1OSCIOFNC bits are set, the Master core OSCIOFNC bit has priority.

2: The POSCMD<1:0> bits are only available in the Master Oscillator Configuration register, FOSC. This setting configures the Primary Oscillator for use by either core.

3.2 Slave Oscillator Configuration Registers

- **FS1OSCSEL: Slave Oscillator Source Selection Register**

  FS1OSCSEL selects the initial oscillator source and start-up option. FS1OSCSEL contains the following Configuration bits:

  - The S1FNOSC<2:0> Configuration bits in the Slave Oscillator Source Selection register (FS1OSCSEL<2:0>) determine the clock source that is used at a Power-on Reset (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.

- **FS1OSC: Slave Oscillator Configuration Register**

  FS1OSC configures the Primary Oscillator mode, OSC2 pin function, Peripheral Pin Select (PPS), and the Fail-Safe and Clock Switching modes.

Table 3-2 lists the configuration settings that select the device's Slave core oscillator source and operating mode at a POR.

Table 3-2: Configuration Bit Values for Clock Selection

Oscillator Source	Oscillator Mode	S1FNOSC<2:0> Value	POSCMD<1:0>(3) Value	Notes
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	Reserved	100	xx	1
S5	Low-Power RC Oscillator (LPRC)	101	xx	1
S6	Backup FRC Oscillator (BFRC)	110	xx	1
S7	Fast RC Oscillator with ÷ N Divider (FRCDIVN)	111	xx	1, 2

Note 1: The OSC2 pin function is determined by the S1OSCIOFNC Configuration bit. If both the Master core OSCIOFNC and Slave core S1OSCIOFNC bits are set, the Master core OSCIOFNC bit has priority.

2: This is the default oscillator mode for an unprogrammed (erased) device.

3: The POSCMD<1:0> bits are only available in the Master Oscillator Configuration register, FOSC. This setting configures the Primary Oscillator for use by either core.

Register 3-3: FS1OSCSEL: Slave Oscillator Source Selection Register

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 23							bit 16

U1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 15							bit 8

R/PO-1	U-1	U-1	U-1	U-1	R/PO-1	R/PO-1	R/PO-1
S1IESO	—	—	—	—	S1FNOSC2	S1FNOSC1	S1FNOSC0
bit 7							bit 0

Legend:	PO = Program Once bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘1’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 23-8 **Unimplemented:** Read as ‘1’

bit 7 **S1IESO:** Internal External Start-up Option bit
- 1 = Starts up device with the Internal FRC Oscillator, then automatically switches to the user-selected oscillator source when ready
- 0 = Starts up device with the user-selected oscillator source

bit 6-3 **Unimplemented:** Read as ‘1’

bit 2-0 **S1FNOSC<2:0>:** Initial Oscillator Source Selection bits
- 111 = Fast RC Oscillator with Divide-by-N (FRCDIVN)
- 110 = Backup FRC Oscillator (BFRC)
- 101 = Low-Power RC Oscillator (LPRC)
- 100 = Reserved
- 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)

Register 3-4: FS1OSC: Slave Oscillator Configuration Register

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 23							bit 16

U-1	U-1	U-1	U-1	U-1	U-1	U-1	R/PO-1
—	—	—	—	—	—	—	S1PLLKEN
bit 15							bit 8

R/PO-1	R/PO-1	U-1	U-1	U-1	R/PO-1	U-1	U-1
S1FCKSM1	S1FCKSM0	—	—	—	S1OSCIOFNC(1)	—	—
bit 7							bit 0

Legend:	PO = Program Once bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘1’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 23-9 **Unimplemented:** Read as ‘1’

bit 8 **S1PLLKEN:** PLL Lock Enable bit
1 = Clock switches will wait for PLL lock signal
0 = Source for PLL lock signal is the PLL enable signal

bit 7-6 **S1FCKSM<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 5-3 **Unimplemented:** Read as ‘1’

bit 2 **S1OSCIOFNC:** OSC2 Pin Function bit (except in XT and HS modes)(1)
1 = OSC2 is the clock output and the Slave core instruction cycle (Slave FCY) clock is output on the OSC2 pin
0 = OSC2 is a general purpose digital I/O pin

bit 1-0 **Unimplemented:** Read as ‘1’

Note 1: If both the Master core OSCIOFNC and Slave core S1OSCIOFNC bits are set, the Master core OSCIOFNC bit has priority.

4.0 SPECIAL FUNCTION REGISTERS

4.1 Master Special Function Registers

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

- **OSCCON: Oscillator Control Register(1)**

  This 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 Divider Register**

  This register controls Doze mode, the FRC divider and PLL prescaler.

- **PLLFBD: PLL Feedback Divider Register**

  This register selects the PLL feedback divider.

- **OSCTUN: FRC Oscillator Tuning Register**

  This register is available to the Master core only and provides the ability to tune the Internal FRC Oscillator frequency in software.

- **PLLDIV: PLL Output Divider Register**

  This register controls the VCO divider and both PLL post-dividers.

- **ACLKCON1: Auxiliary Clock Control Register**

  This register is used to configure and enable the Auxiliary PLL. APLL clock source selection, APLL prescaler setting and APLL lock status are all contained in this register.

- **APLLFBD1: APLL Feedback Divider Register**

  This register controls the APLL feedback divider.

- **APLLDIV1: APLL Output Divider Register**

  This register controls the AVCO divider and both APLL post-dividers.

  CANCLKCON: CAN Clock Control Register(4)

  This register controls the CAN clock source and divider selection.

- **REFOCONL: Reference Clock Control Low Register**

  This register is used to configure and enable the Reference Clock output.

- **REFOCONH: Reference Clock Control High Register**

  This register controls the Reference Clock output divider selection.

Register 4-1: OSCCON: Oscillator Control Register(1)

U-0	R-0	R-0	R-0	U-0	R/W-y	R/W-y	R/W-y
—	COSC2	COSC1	COSC0	—	NOSC2(2)	NOSC1(2)	NOSC0(2)
bit 15							bit 8

R/W-0	U-0	R-0	U-0	R/W-0	U-0	U-0	R/W-0
CLKLOCK	—	LOCK	—	CF(3)	—	—	OSWEN
bit 7							bit 0

Legend:	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 **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 (FRCDIVN)
110 = Backup FRC Oscillator (BFRC)
101 = Low-Power RC Oscillator (LPRC)
100 = Reserved
011 = Primary Oscillator (XT, HS, EC) with PLL (XTPLL, HSPLL, ECPLL)
010 = Primary Oscillator (XT, HS, EC)
001 = Fast RC Oscillator (FRC) with PLL (FRCPLL)
000 = Fast RC Oscillator (FRC)

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

bit 10-8 **NOSC<2:0>:** New Oscillator Selection bits(2)
111 = Fast RC Oscillator (FRC) with Divide-by-N (FRCDIVN)
110 = Backup FRC Oscillator (BFRC)
101 = Low-Power RC Oscillator (LPRC)
100 = Reserved
011 = Primary Oscillator (XT, HS, EC) with PLL (XTPLL, HSPLL, ECPLL)
010 = Primary Oscillator (XT, HS, EC)
001 = Fast RC Oscillator (FRC) with PLL (FRCPLL)
000 = Fast RC Oscillator (FRC)

bit 7 **CLKLOCK:** Clock Lock Enable bit
1 = If (FCKSM0 = 1), then the clock and PLL configurations are locked; if (FCKSM0 = 0), then the clock and PLL configurations may be modified
0 = Clock and PLL selections are not locked, configurations may be modified

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

bit 5 **LOCK:** PLL Lock Status bit (read-only)
1 = Indicates that PLL is in lock or the PLL start-up timer is satisfied
0 = Indicates that PLL is out of lock, the start-up timer is in progress or PLL is disabled

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

bit 3 **CF:** Clock Fail Detect bit(3)
1 = FSCM has detected a clock failure
0 = FSCM has not detected a clock failure

Note 1: Writes to this register require an unlock sequence.

2: Direct clock switches between any Primary Oscillator mode with PLL and FRCPLL modes are not permitted. This applies to clock switches in either direction. In these instances, the application must switch to FRC mode as a transitional clock source between the two PLL modes.

3: This bit should only be cleared in software. Setting the bit in software (= 1) will have the same effect as an actual oscillator failure and will trigger an oscillator failure trap.

Register 4-1: OSCCON: Oscillator Control Register(1) (Continued)

bit 2-1	**Unimplemented:** Read as '0'
bit 0	**OSWEN:** Oscillator Switch Enable bit 1 = Requests oscillator switch to the selection specified by the NOSC<2:0> bits 0 = Oscillator switch is complete

Note 1: Writes to this register require an unlock sequence.

2: Direct clock switches between any Primary Oscillator mode with PLL and FRCPLL modes are not permitted. This applies to clock switches in either direction. In these instances, the application must switch to FRC mode as a transitional clock source between the two PLL modes.

3: This bit should only be cleared in software. Setting the bit in software (= 1) will have the same effect as an actual oscillator failure and will trigger an oscillator failure trap.

Register 4-2: CLKDIV: Clock Divider 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	DOZE2(1)	DOZE1(1)	DOZE0(1)	DOZEN(2,3)	FRCDIV2	FRCDIV1	FRCDIV0
bit 15							bit 8

U-0	U-0	r-0	r-0	R/W-0	R/W-0	R/W-0	R/W-1
—	—	—	—	PLLPRE3(4)	PLLPRE2(4)	PLLPRE1(4)	PLLPRE0(4)
bit 7							bit 0

Legend:	r = Reserved 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 **ROI:** Recover on Interrupt bit
1 = Interrupts will clear the DOZEN bit and the processor clock, and the 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(1)
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(2,3)
1 = DOZE<2:0> bits field specifies the ratio between the peripheral clocks and the processor clocks
0 = Processor clock and peripheral clock ratio is 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 **Unimplemented:** Read as '0'

bit 5-4 **Reserved:** Read as '0'

Note 1: The DOZE<2:0> bits can only be written to when the DOZEN bit is clear. If DOZEN = 1, any writes to DOZE<2:0> are ignored.

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

3: The DOZEN bit cannot be set if DOZE<2:0> = 000. If DOZE<2:0> = 000, any attempt by user software to set the DOZEN bit is ignored.

4: PLLPRE<3:0> bits may be updated while the PLL is operating, but the VCO may overshoot.

Register 4-2: CLKDIV: Clock Divider Register (Continued)

bit 3-0 **PLLPRE<3:0>:** PLL Phase Detector Input Divider Select bits (also denoted as 'N1', PLL prescaler)(4)

1111 = Reserved
• • •
1001 = Reserved
1000 = Input divided by 8
0111 = Input divided by 7
0110 = Input divided by 6
0101 = Input divided by 5
0100 = Input divided by 4
0011 = Input divided by 3
0010 = Input divided by 2
0001 = Input divided by 1 (power-on default selection)
0000 = Invalid selection

Note 1: The DOZE<2:0> bits can only be written to when the DOZEN bit is clear. If DOZEN = 1, any writes to DOZE<2:0> are ignored.

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

3: The DOZEN bit cannot be set if DOZE<2:0> = 000. If DOZE<2:0> = 000, any attempt by user software to set the DOZEN bit is ignored.

4: PLLPRE<3:0> bits may be updated while the PLL is operating, but the VCO may overshoot.

Register 4-3: PLLFBD: PLL Feedback Divider Register

U-0	U-0	U-0	U-0	r-0	r-0	r-0	r-0
—	—	—	—	—	—	—	—
bit 15							bit 8

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

Legend:	r = Reserved 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 15-12 **Unimplemented:** Read as '0'

bit 11-8 **Reserved:** Read as '0'

bit 7-0 **PLLFBDIV<7:0>:** PLL Feedback Divider bits (also denoted as 'M', PLL multiplier)

11111111 = Reserved
• • •
11001000 = 200 Maximum(1)
• • •
10010110 = 150 (default)
• • •
00010000 = 16 Minimum(1)
• • •
00000010 = Reserved
00000001 = Reserved
00000000 = Reserved

Note 1: The allowed range is 16-200 (decimal). The rest of the values are reserved and should be avoided. The power on the default feedback divider is 150 (decimal) with an 8 MHz FRC input clock. The VCO frequency is 1.2 GHz.

Register 4-4: 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 = Maximum frequency deviation of 1.74% (8.139 MHz)
011110 = Center frequency + 1.693% (8.135 MHz)
• • •
000001 = Center frequency + 0.047% (8.038 MHz)
000000 = Center frequency (8.00 MHz nominal)
111111 = Center frequency – 0.047% (7.962 MHz)
• • •
100001 = Center frequency – 1.693% (7.865 MHz)
100000 = Minimum frequency deviation of -1.74% (7.861 MHz)

Register 4-5: PLLDIV: PLL Output Divider Register

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	VCODIV<1:0>	
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-1
—	POST1DIV<2:0>(1,2)			—	POST2DIV<2:0>(1,2)		
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 15-10 **Unimplemented:** Read as '0'

bit 9-8 **VCODIV<1:0>:** PLL VCO Output Divider Select bits
11 = VCO clock
10 = VCO/2 clock
01 = VCO/3 clock
00 = VCO/4 clock

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

bit 6-4 **POST1DIV<2:0>:** PLL Output Divider #1 Ratio bits(1,2)
POST1DIV<2:0> bits can have a valid value from 1 to 7 (POST1DIVx value should be greater than or equal to the POST2DIVx value). The POST1DIVx divider is designed to operate at higher clock rates than the POST2DIVx divider.

bit 3 **Unimplemented:** Read as '0'

bit 2-0 **POST2DIV<2:0>:** PLL Output Divider #2 Ratio bits(1,2)
POST2DIV<2:0> bits can have a valid value from 1 to 7 (POST2DIVx value should be less than or equal to the POST1DIVx value). The POST1DIVx divider is designed to operate at higher clock rates than the POST2DIVx divider.

Note 1: The POST1DIVx and POST2DIVx divider values must not be changed while the PLL is operating.
2: The default values for POST1DIVx and POST2DIVx are 4 and 1, respectively, yielding a 150 MHz Master PLL output.

Register 4-6: ACLKCON1: Auxiliary Clock Control Register

R/W-0	R/W-0	U-0	U-0	U-0	U-0	U-0	R/W-0
APLLEN(1)	APLLCK	—	—	—	—	—	FRCSEL
bit 15							bit 8

U-0	U-0	r-0	r-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	APLLPRE3	APLLPRE2	APLLPRE1	APLLPRE0
bit 7							bit 0

Legend:	r = Reserved 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 **APLLEN:** Auxiliary PLL Enable/Bypass Select bit(1)
- 1 = AFPLLO is connected to the APLL post-divider output (bypass disabled)
- 0 = AFPLLO is connected to the APLL input clock (bypass enabled)

bit 14 **APLLCK:** APLL Phase-Locked Loop State Status bit
- 1 = Auxiliary PLL is in lock
- 0 = Auxiliary PLL is not in lock

bit 13-9 **Unimplemented:** Read as '0'

bit 8 **FRCSEL:** FRC Clock Source Select bit
- 1 = FRC is the clock source for APLL
- 0 = Primary Oscillator is the clock source for APLL

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

bit 5-4 **Reserved:** Read as '0'

bit 3-0 **APLLPRE<3:0>:** Auxiliary PLL Phase Detector Input Divider Select bits
- 1111 = Reserved
- • • •
- 1001 = Reserved
- 1000 = Input divided by 8
- 0111 = Input divided by 7
- 0110 = Input divided by 6
- 0101 = Input divided by 5
- 0100 = Input divided by 4
- 0011 = Input divided by 3
- 0010 = Input divided by 2
- 0001 = Input divided by 1 (power-on default selection)
- 0000 = Invalid selection

Note 1: Even with the APLLEN bit set, another peripheral must generate a clock request before the APLL will start.

Register 4-7: APLLFBD1: APLL Feedback Divider Register

U-0	U-0	U-0	U-0	r-0	r-0	r-0	r-0
—	—	—	—	—	—	—	—
bit 15							bit 8

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

Legend:	r = Reserved 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 15-12 **Unimplemented:** Read as '0'

bit 11-8 **Reserved:** Read as '0'

bit 7-0 **APLLFBDIV<7:0>:** APLL Feedback Divider bits

11111111 = Reserved
• • •
11001000 = 200 Maximum(1)
• • •
10010110 = 150 (Default)
• • •
00010000 = 16 Minimum(1)
• • •
00000010 = Reserved
00000001 = Reserved
00000000 = Reserved

Note 1: The allowed range is 16-200 (decimal). The rest of the values are reserved and should be avoided. The power-on default feedback divider is 150 (decimal) with an 8 MHz FRC input clock; the VCO frequency is 1.2 GHz.

Register 4-8: APLLDIV1: APLL Output Divider Register

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	AVCODIV<1:0>	
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-1
—	APOST1DIV<2:0>(1,2)			—	APOST2DIV<2:0>(1,2)		
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 15-10 **Unimplemented:** Read as '0'

bit 9-8 **AVCODIV<1:0>:** APLL VCO Output Divider Select bits
11 = VCO clock
10 = VCO/2 clock
01 = VCO/3 clock
00 = VCO/4 clock

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

bit 6-4 **APOST1DIV<2:0>:** APLL Output Divider #1 Ratio bits(1,2)
APOST1DIV<2:0> bits can have a valid value from 1 to 7 (the APOST1DIVx value should be greater than or equal to the APOST2DIVx value). The APOST1DIVx divider is designed to operate at higher clock rates than the APOST2DIVx divider.

bit 3 **Unimplemented:** Read as '0'

bit 2-0 **APOST2DIV<2:0>:** APLL Output Divider #2 Ratio bits(1,2)
APOST2DIV<2:0> bits can have a valid value from 1 to 7 (the APOST2DIVx value should be less than or equal to the APOST1DIVx value). The APOST1DIVx divider is designed to operate at higher clock rates than the APOST2DIVx divider.

Note 1: The APOST1DIVx and APOST2DIVx values must not be changed while the PLL is operating.
2: The default values for APOST1DIVx and APOST2DIVx are 4 and 1, respectively, yielding a 150 MHz Master APLL output.

Register 4-9: CANCLKCON: CAN Clock Control Register(4)

R/W-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
CANCLKEN	—	—	—	CANCLKSEL<3:0>(1)			
bit 15							bit 8

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-1
—	CANCLKDIV<6:0>(2,3)						
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 **CANCLKEN:** Enables the CAN Clock Generator
1 = CAN clock generation circuitry is enabled
0 = CAN clock generation circuitry is disabled

bit 14-12 **Unimplemented:** Read as ‘0’

bit 11-8 **CANCLKSEL<3:0>:** CAN Clock Source Select bits(1)
1011-1111 = Reserved (no clock selected)
1010 = $AF_{VCO}/4$
1001 = $AF_{VCO}/3$
1000 = $AF_{VCO}/2$
0111 = AF_{VCO}
0110 = AF_{PLLO}
0101 = $F_{VCO}/4$
0100 = $F_{VCO}/3$
0011 = $F_{VCO}/2$
0010 = F_{PLLO}
0001 = F_{VCO}
0000 = 0 (no clock selected)

bit 7 **Unimplemented:** Read as ‘0’

bit 6-0 **CANCLKDIV<6:0>:** CAN Clock Divider Select bits(2,3)
1111111 = Divide-by-128
• • •
0000010 = Divide-by-3
0000001 = Divide-by-2
0000000 = Divide-by-1

Note 1: The user must ensure the input clock source is 640 MHz or less.
2: The CANCLKDIVx divider value must not be changed during CAN module operation.
3: The user must ensure the maximum clock output frequency of the divider is 80 MHz or less.
4: CAN is not implemented in all devices. Refer to the device-specific data sheet to check if CAN is included in your device.

Register 4-10: REFOCONL: Reference Clock Control Low Register

R/W-0	U-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0, HC	R-0, HSC
ROEN	—	ROSIDL	ROOUT	ROSLP	—	ROSWEN	ROACTIV
bit 15							bit 8

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	ROSEL3	ROSEL2	ROSEL1	ROSEL0
bit 7							bit 0

Legend:	HC = Hardware Clearable bit	HSC = Hardware Settable/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 **ROEN:** Reference Clock Enable bit
1 = Reference Oscillator is enabled on the REFO pin
0 = Reference Oscillator is disabled

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

bit 13 **ROSIDL:** Reference Clock Stop in Idle bit
1 = Reference Oscillator is disabled in Idle mode
0 = Reference Oscillator continues to run in Idle mode

bit 12 **ROOUT:** Reference Clock Output Enable bit
1 = Reference Clock external output is enabled and available on the REFO pin
0 = Reference Clock external output is disabled

bit 11 **ROSLP:** Reference Clock Stop in Sleep bit
1 = Reference Oscillator continues to run in Sleep modes
0 = Reference Oscillator is disabled in Sleep modes

bit 10 **Unimplemented:** Read as '0'

bit 9 **ROSWEN:** Reference Clock Output Enable bit
1 = Clock divider change is requested or is in progress (set in software, cleared by hardware upon completion)
0 = Clock divider change has completed or is not pending

bit 8 **ROACTIV:** Reference Clock Status bit
1 = Reference Clock is active; do not change clock source
0 = Reference Clock is stopped; clock source and configuration may be safely changed

bit 7-4 **Unimplemented:** Read as '0'

bit 3-0 **ROSEL<3:0>:** Reference Clock Source Select bits
1111 = Reserved
••• = Reserved
1000 = Reserved
0111 = REFI pin
0110 = FVCO/4
0101 = BFRC
0100 = LPRC
0011 = FRC
0010 = Primary Oscillator
0001 = FOSC/2 (FP)
0000 = FOSC

Register 4-11: REFOCONH: Reference Clock Control High Register

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	RODIV<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
RODIV<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 14-0 **RODIV<14:0>:** Reference Clock Integer Divider Select bits

Divider for the selected input clock source is two times the selected value.

111 1111 1111 1111 = Base clock value divided by 65,534 (2 * 7FFFh)
111 1111 1111 1110 = Base clock value divided by 65,532 (2 * 7FFEh)
111 1111 1111 1101 = Base clock value divided by 65,530 (2 * 7FFDh)
• • •
000 0000 0000 0010 = Base clock value divided by 4 (2 * 2)
000 0000 0000 0001 = Base clock value divided by 2 (2 * 1)
000 0000 0000 0000 = Base clock value

4.2 Slave Special Function Registers

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

- **OSCCON: Oscillator Control Register(1)**

  This 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 Divider Register**

  This register controls the Doze mode, FRC divider and PLL prescaler.

- **PLLFBD: PLL Feedback Divider Register**

  This register selects the PLL feedback divider.

- **PLLDIV: PLL Output Divider Register**

  This register controls the VCO divider and both PLL post-dividers.

- **ACLKCON1: Auxiliary Clock Control Register**

  This register is used to configure and enable the Auxiliary PLL. APLL clock source selection, APLL prescaler setting and APLL lock status are all contained in this register.

- **APLLFBD1: APLL Feedback Divider Register**

  This register controls the APLL feedback divider.

- **APLLDIV1: APLL Output Divider Register**

  This register controls the AVCO divider and both APLL post-dividers.

- **REFOCONL: Reference Clock Control Low Register**

  This register is used to configure and enable the Reference Clock output.

- **REFOCONH: Reference Clock Control High Register**

  This register controls the Reference Clock output divider selection.

Register 4-12: OSCCON: Oscillator Control Register(1)

U-0	R-0	R-0	R-0	U-0	R/W-y	R/W-y	R/W-y
—	COSC2	COSC1	COSC0	—	NOSC2(2)	NOSC1(2)	NOSC0(2)
bit 15							bit 8

R/W-0	U-0	R-0	U-0	R/W-0	U-0	U-0	R/W-0
CLKLOCK	—	LOCK	—	CF(3)	—	—	OSWEN
bit 7							bit 0

Legend:	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 **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 (FRCDIVN)
110 = Backup FRC Oscillator (BFRC)
101 = Low-Power RC Oscillator (LPRC)
100 = Reserved
011 = Primary Oscillator (XT, HS, EC) with PLL (XTPLL, HSPLL, ECPLL)
010 = Primary Oscillator (XT, HS, EC)
001 = Fast RC Oscillator (FRC) with PLL (FRCPLL)
000 = Fast RC Oscillator (FRC)

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

bit 10-8 **NOSC<2:0>:** New Oscillator Selection bits(2)
111 = Fast RC Oscillator (FRC) with Divide-by-N (FRCDIVN)
110 = Backup FRC Oscillator (BFRC)
101 = Low-Power RC Oscillator (LPRC)
100 = Reserved
011 = Primary Oscillator (XT, HS, EC) with PLL (XTPLL, HSPLL, ECPLL)
010 = Primary Oscillator (XT, HS, EC)
001 = Fast RC Oscillator (FRC) with PLL (FRCPLL)
000 = Fast RC Oscillator (FRC)

bit 7 **CLKLOCK:** Clock Lock Enable bit
1 = If (FCKSM0 = 1), then clock and PLL configurations are locked; if (FCKSM0 = 0), then clock and PLL configurations may be modified
0 = Clock and PLL selections are not locked, configurations may be modified

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 **Unimplemented:** Read as '0'

bit 3 **CF:** Clock Fail Detect bit(3)
1 = FSCM has detected a clock failure
0 = FSCM has not detected a clock failure

Note 1: Writes to this register require an unlock sequence.

2: Direct clock switches between any Primary Oscillator mode with PLL and FRCPLL mode are not permitted. This applies to clock switches in either direction. In these instances, the application must switch to FRC mode as a transitional clock source between the two PLL modes.

3: This bit should only be cleared in software. Setting the bit in software (= 1) will have the same effect as an actual oscillator failure and will trigger an oscillator failure trap.

Register 4-12: OSCCON: Oscillator Control Register(1) (Continued)

bit 2-1	**Unimplemented:** Read as '0'
bit 0	**OSWEN:** Oscillator Switch Enable bit 1 = Requests oscillator switch to the selection specified by the NOSC<2:0> bits 0 = Oscillator switch is complete

Note 1: Writes to this register require an unlock sequence.

2: Direct clock switches between any Primary Oscillator mode with PLL and FRCPLL mode are not permitted. This applies to clock switches in either direction. In these instances, the application must switch to FRC mode as a transitional clock source between the two PLL modes.

3: This bit should only be cleared in software. Setting the bit in software (= 1) will have the same effect as an actual oscillator failure and will trigger an oscillator failure trap.

Register 4-13: CLKDIV: Clock Divider 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	DOZE2(1)	DOZE1(1)	DOZE0(1)	DOZEN(2,3)	FRCDIV2	FRCDIV1	FRCDIV0
bit 15							bit 8

U-0	U-0	r-0	r-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	PLLPRE3(4)	PLLPRE2(4)	PLLPRE1(4)	PLLPRE0(4)
bit 7							bit 0

Legend:	r = Reserved 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 **ROI:** Recover on Interrupt bit
1 = Interrupts will clear the DOZEN bit and the processor clock, and the 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(1)
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(2,3)
1 = DOZE<2:0> bits field specifies the ratio between the peripheral clocks and the processor clocks
0 = Processor clock and peripheral clock ratio is 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 **Unimplemented:** Read as '0'

bit 5-4 **Reserved:** Read as '0'

Note 1: The DOZE<2:0> bits can only be written to when the DOZEN bit is clear. If DOZEN = 1, any writes to DOZE<2:0> are ignored.

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

3: The DOZEN bit cannot be set if DOZE<2:0> = 000. If DOZE<2:0> = 000, any attempt by user software to set the DOZEN bit is ignored.

4: PLLPRE<3:0> may be updated while the PLL is operating, but the VCO may overshoot.

Register 4-13: CLKDIV: Clock Divider Register (Continued)

bit 3-0 **PLLPRE<3:0>:** PLL Phase Detector Input Divider Select bits (also denoted as 'N1', PLL prescaler)(4)

1111 = Reserved
• • •
1001 = Reserved
1000 = Input divided by 8
0111 = Input divided by 7
0110 = Input divided by 6
0101 = Input divided by 5
0100 = Input divided by 4
0011 = Input divided by 3
0010 = Input divided by 2
0001 = Input divided by 1 (power-on default selection)
0000 = Invalid selection

Note 1: The DOZE<2:0> bits can only be written to when the DOZEN bit is clear. If DOZEN = 1, any writes to DOZE<2:0> are ignored.

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

3: The DOZEN bit cannot be set if DOZE<2:0> = 000. If DOZE<2:0> = 000, any attempt by user software to set the DOZEN bit is ignored.

4: PLLPRE<3:0> may be updated while the PLL is operating, but the VCO may overshoot.

Register 4-14: PLLFBD: PLL Feedback Divider Register

U-0	U-0	U-0	U-0	r-0	r-0	r-0	r-0
—	—	—	—	—	—	—	—
bit 15							bit 8

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

Legend:	r = Reserved 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 15-12 **Unimplemented:** Read as '0'

bit 11-8 **Reserved:** Read as '0'

bit 7-0 **PLLFBDIV<7:0>:** PLL Feedback Divider bits (also denoted as 'M', PLL multiplier)

11111111 = Reserved
• • •
11001000 = 200 Maximum(1)
• • •
10010110 = 150 (default)
• • •
00010000 = 16 Minimum(1)
• • •
00000010 = Reserved
00000001 = Reserved
00000000 = Reserved

Note 1: The allowed range is 16-200 (decimal). The rest of the values are reserved and should be avoided. The power on the default feedback divider is 150 (decimal) with an 8 MHz FRC input clock. The VCO frequency is 1.2 GHz.

Register 4-15: PLLDIV: PLL Output Divider Register

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	VCODIV<1:0>	
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-1
—	POST1DIV<2:0>(1,2)			—	POST2DIV<2:0>(1,2)		
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 15-10 **Unimplemented:** Read as '0'

bit 9-8 **VCODIV<1:0>:** PLL VCO Output Divider Select bits
11 = VCO clock
10 = VCO/2 clock
01 = VCO/3 clock
00 = VCO/4 clock

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

bit 6-4 **POST1DIV<2:0>:** PLL Output Divider #1 Ratio bits(1,2)
POST1DIV<2:0> bits can have a valid value from 1 to 7 (POST1DIVx value should be greater than or equal to the POST2DIVx value). The POST1DIVx divider is designed to operate at higher clock rates than the POST2DIVx divider.

bit 3 **Unimplemented:** Read as '0'

bit 2-0 **POST2DIV<2:0>:** PLL Output Divider #2 Ratio bits(1,2)
POST2DIV<2:0> bits can have a valid value from 1 to 7 (POST2DIVx value should be less than or equal to the POST1DIVx value). The POST1DIVx divider is designed to operate at higher clock rates than the POST2DIVx divider.

Note 1: The POST1DIVx and POST2DIVx divider values must not be changed while the PLL is operating.
2: The default values for POST1DIVx and POST2DIVx are 4 and 1, respectively, yielding a 150 MHz Slave PLL output.

Register 4-16: ACLKCON1: Auxiliary Clock Control Register

R/W-0	R/W-0	U-0	U-0	U-0	U-0	U-0	R/W-0
APLLEN(1)	APLLCK	—	—	—	—	—	FRCSEL
bit 15							bit 8

U-0	U-0	r-0	r-0	R/W-0	R/W-0	R/W-0	R/W-1
—	—	—	—	APLLPRE3	APLLPRE2	APLLPRE1	APLLPRE0
bit 7							bit 0

Legend:	r = Reserved 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 **APLLEN:** Auxiliary PLL Enable/Bypass Select bit(1)
1 = AFPLLO is connected to the APLL post-divider output (bypass disabled)
0 = AFPLLO is connected to the APLL input clock (bypass enabled)

bit 14 **APLLCK:** APLL Phase-Locked State Status bit
1 = Auxiliary PLL is in lock
0 = Auxiliary PLL is not in lock

bit 13-9 **Unimplemented:** Read as '0'

bit 8 **FRCSEL:** FRC Clock Source Select bit
1 = FRC is the clock source for APLL
0 = Primary Oscillator is the clock source for APLL

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

bit 5-4 **Reserved:** Read as '0'

bit 3-0 **APLLPRE<3:0>:** Auxiliary PLL Phase Detector Input Divider bits
1111 = Reserved
• • •
1001 = Reserved
1000 = Input divided by 8
0111 = Input divided by 7
0110 = Input divided by 6
0101 = Input divided by 5
0100 = Input divided by 4
0011 = Input divided by 3
0010 = Input divided by 2
0001 = Input divided by 1 (power-on default selection)
0000 = Invalid selection

Note 1: Even with the APLLEN bit set, another peripheral must generate a clock request before the APLL will start.

Register 4-17: APLLFBD1: APLL Feedback Divider Register

U-0	U-0	U-0	U-0	r-0	r-0	r-0	r-0
—	—	—	—	—	—	—	—
bit 15							bit 8

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

Legend:	r = Reserved 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 15-12 **Unimplemented:** Read as '0'

bit 11-8 **Reserved:** Read as '0'

bit 7-0 **APLLFBDIV<7:0>:** APLL Feedback Divider bits

11111111 = Reserved
• • •
11001000 = 200 Maximum(1)
• • •
10010110 = 150 (Default)
• • •
00010000 = 16 Minimum(1)
• • •
00000010 = Reserved
00000001 = Reserved
00000000 = Reserved

Note 1: The allowed range is 16-200 (decimal). The rest of the values are reserved and should be avoided. The power on the default feedback divider is 150 (decimal) with an 8 MHz FRC input clock; the VCO frequency is 1.2 GHz.

Register 4-18: APLLDIV1: APLL Output Divider Register

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	AVCODIV<1:0>	
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-1
—	APOST1DIV<2:0>(1,2)			—	APOST2DIV<2:0>(1,2)		
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 15-10 **Unimplemented:** Read as ‘0’

bit 9-8 **AVCODIV<1:0>:** APLL VCO Output Divider Select bits
11 = VCO clock
10 = VCO/2 clock
01 = VCO/3clock
00 = VCO/4 clock

bit 7 **Unimplemented:** Read as ‘0’

bit 6-4 **APOST1DIV<2:0>:** APLL Output Divider #1 Ratio bits(1,2)
APOST1DIV<2:0> bits can have a valid value from 1 to 7 (the APOST1DIVx value should be greater than or equal to the APOST2DIVx value). The APOST1DIVx divider is designed to operate at higher clock rates than the APOST2DIVx divider.

bit 3 **Unimplemented:** Read as ‘0’

bit 2-0 **APOST2DIV<2:0>:** APLL Output Divider #2 Ratio bits(1,2)
APOST2DIV<2:0> bits can have a valid value from 1 to 7 (the APOST2DIVx value should be less than or equal to the APOST1DIVx value). The APOST1DIVx divider is designed to operate at higher clock rates than the APOST2DIVx divider.

Note 1: The APOST1DIVx and APOST2DIVx divider values must not be changed while the PLL is operating.
2: The default values for APOST1DIVx and APOST2DIVx are 4 and 1, respectively, yielding a 150 MHz Slave APLL output.

Register 4-19: REFOCONL: Reference Clock Control Low Register

R/W-0	U-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0, HC	R-0, HSC
ROEN	—	ROSIDL	ROOUT	ROSLP	—	ROSWEN	ROACTIV
bit 15							bit 8

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	ROSEL3	ROSEL2	ROSEL1	ROSEL0
bit 7							bit 0

Legend:	HC = Hardware Clearable bit	HSC = Hardware Settable/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 **ROEN:** Reference Clock Enable bit
1 = Reference Oscillator is enabled on the REFO pin
0 = Reference Oscillator is disabled

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

bit 13 **ROSIDL:** Reference Clock Stop in Idle bit
1 = Reference Oscillator is disabled in Idle mode
0 = Reference Oscillator continues to run in Idle mode

bit 12 **ROOUT:** Reference Clock Output Enable bit
1 = Reference Clock external output is enabled and available on the REFO pin
0 = Reference Clock external output is disabled

bit 11 **ROSLP:** Reference Clock Stop in Sleep bit
1 = Reference Oscillator continues to run in Sleep modes
0 = Reference Oscillator is disabled in Sleep modes

bit 10 **Unimplemented:** Read as '0'

bit 9 **ROSWEN:** Reference Clock Output Enable bit
1 = Clock divider change is requested or is in progress (set in software, cleared by hardware upon completion)
0 = Clock divider change has completed or is not pending

bit 8 **ROACTIV:** Reference Clock Status bit
1 = Reference Clock is active; do not change clock source
0 = Reference Clock is stopped; clock source and configuration may be safely changed

bit 7-4 **Unimplemented:** Read as '0'

bit 3-0 **ROSEL<3:0>:** Reference Clock Source Select bits
1111 = Reserved
••• = Reserved
1000 = Reserved
0111 = REFI pin
0110 = FVCO/4
0101 = BFRC Oscillator
0100 = LPRC Oscillator
0011 = FRC Oscillator
0010 = Primary Oscillator
0001 = FOSC/2 (FP)
0000 = FOSC

Register 4-20: REFOCONH: Reference Clock Control High Register

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	RODIV<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
RODIV<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 14-0 **RODIV<14:0>:** Reference Clock Integer Divider Select bits

Divider for the selected input clock source is two times the selected value.

111 1111 1111 1111 = Base clock value divided by 65,534 (2 * 7FFFh)
111 1111 1111 1110 = Base clock value divided by 65,532 (2 * 7FFEh)
111 1111 1111 1101 = Base clock value divided by 65,530 (2 * 7FFDh)
• • •
000 0000 0000 0010 = Base clock value divided by 4 (2 * 2)
000 0000 0000 0001 = Base clock value divided by 2 (2 * 1)
000 0000 0000 0000 = Base clock value

5.0 PRIMARY OSCILLATOR (POSC)

The dsPIC33/PIC24 devices contain one instance of the Primary Oscillator (POSC), which is available to both the Master and Slave clock subsystems. The Primary Oscillator is available on the OSC1 and OSC2 pins of the dsPIC33/PIC24 devices. This connection enables an external crystal (or ceramic resonator) to provide the clock to the device. The Primary Oscillator provides three modes of operation:

- **Medium Speed Oscillator (XT Mode)**
  - The XT mode is a Medium Gain, Medium Frequency mode used to work with crystal frequencies of 3.5 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 32 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 (from 0 MHz to up to 64 MHz) and input on the OSC1 pin.

The FNOSC<2:0> and S1FNOSC<2:0> Configuration bits in the Master and Slave Oscillator Source Selection registers (FOSCSEL<2:0>, FS1OSCSEL<2:0>) specify the system clock source at Power-on Reset.

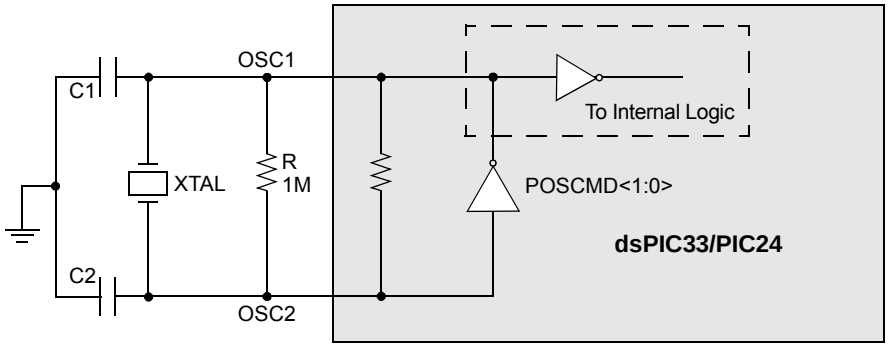
The POSCMD<1:0> Configuration bits in the Oscillator Configuration register (FOSC<1:0>) specify the Primary Oscillator mode. Table 5-1 provides the options selected by specific bit configurations, which are programmed at the time of device programming.

Table 5-1: Primary Oscillator Clock Source Options

FNOSC<2:0> S1FNOSC<2:0> Value	POSCMD<1:0> Value	Primary Oscillator Source and Mode
010	00	Primary Oscillator: External Clock Mode (EC)
010	01	Primary Oscillator: Medium Frequency Mode (XT)
010	10	Primary Oscillator: High-Frequency Mode (HS)
011	00	Primary Oscillator with PLL: External Clock Mode (ECPLL)
011	01	Primary Oscillator with PLL: Medium Frequency Mode (XTPLL)
011	10	Primary Oscillator with PLL: High-Frequency Mode (HSPLL)

Figure 5-1 is a recommended crystal oscillator circuit diagram for the dsPIC33/PIC24 devices. Capacitors, C1 and C2, form the Load Capacitance (C_L) for the crystal. The optimum Load Capacitance for a given crystal is specified by the crystal manufacturer. Load Capacitance can be calculated as shown in Equation 5-1.

Figure 5-1: Crystal or Ceramic Resonator Operation (XT or HS Oscillator Mode)



Note: For more information, refer to the specific device data sheet.

Equation 5-1: Crystal Load Capacitance

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

Note: Where C_S is the stray capacitance.

Assuming C1 = C2, Equation 5-2 gives the capacitor value (C1, C2) for a given load and stray capacitance.

Equation 5-2: External Capacitor for Crystal

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

For more information on crystal oscillators and their operation, refer to **Section 17.0 "Related Application Notes"**.

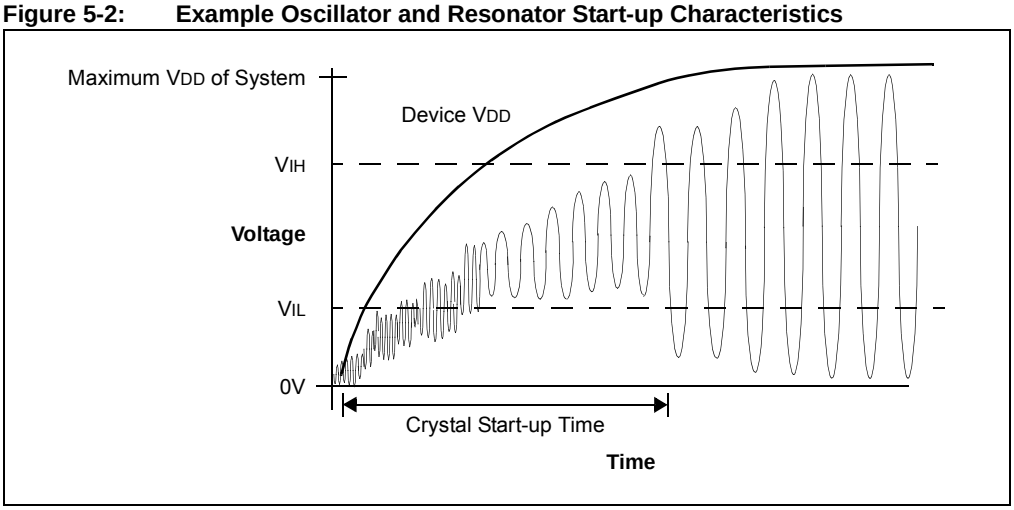
5.1 Oscillator Start-up Time

As the device voltage increases from VSS, the oscillator will start its oscillations. The time required for the oscillator to start oscillating depends on these factors:

- Crystal and resonator frequency
- Capacitor values used (C1 and C2 in Figure 5-1)
- Device VDD 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 5-2 illustrates a plot of a typical oscillator and resonator start-up.

Figure 5-2: Example Oscillator and Resonator Start-up Characteristics



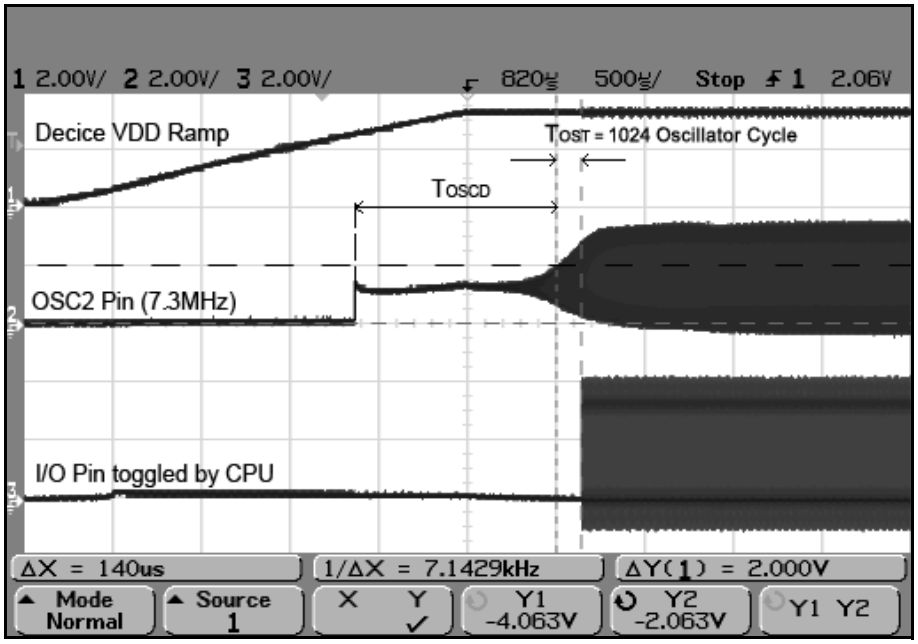
To ensure that a crystal oscillator (or ceramic resonator) has started and stabilized, an Oscillator Start-up Timer (OST) is provided with the Primary Oscillator (POSC). The OST is a simple, 10-bit counter that counts 1024 cycles before releasing the oscillator clock to the rest of the system. This time-out 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 (that is, on POR, BOR and wake-up from Sleep mode) when XT or HS mode is selected in the Configuration Words. The TOST timer does not exist when EC mode is selected.

After 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. For more information, see **Section 8.0 "Master Phase-Locked Loop (PLL)"**.

Primary Oscillator start-up behavior is illustrated in Figure 5-3, where the CPU begins toggling an I/O pin when it starts execution after the TOSCD + TOST interval.

Figure 5-3: Oscillator Start-up Characteristics



5.2 Primary Oscillator Pin Functionality

The Primary Oscillator pins (OSC1 and OSC2) can be used for other functions when the oscillator is not being used. The POSCMD<1:0> Configuration bits in the Oscillator Configuration register (FOSC<1:0>) determine the oscillator pin function. The OSCIOFNC bit (FOSC<2>) determines the OSC2 pin function.

POSCMD<1:0>: Primary Oscillator Mode Selection bits:

- `11` = Primary Oscillator mode is disabled
- `10` = HS Oscillator mode is selected
- `01` = XT Oscillator mode is selected
- `00` = External Clock mode is selected

OSCIOFNC: OSC2 Pin Function bit (except in XT and HS modes):

- `1` = OSC2 is the clock output and the Master instruction cycle (Master FCY) clock is output on the OCS2 pin (see Figure 5-4)
- `0` = OSC2 is a general purpose digital I/O pin (see Figure 5-5)

S1OSCIOFNC: OSC2 Pin Function bit (except in XT and HS modes):

- `1` = OSC2 is the clock output and the Slave instruction cycle (Slave FCY) clock is output on the OCS2 pin (see Figure 5-4)
- `0` = OSC2 is a general purpose digital I/O pin (see Figure 5-5)

The oscillator pin functions are provided in Table 5-2.

Table 5-2: Clock Pin Function Selection

Oscillator Source	OSCIOFNC Value	S1OSCIOFNC Value	POSCMD<1:0> Value	OSC1 Pin Function(1)	OSC2 Pin Function(2)
Primary Oscillator Disabled	1	x	11	Digital I/O	Master Clock Output (FCY)
Primary Oscillator Disabled	0	1	11	Digital I/O	Slave Clock Output (Slave FCY)
Primary Oscillator Disabled	0	0	11	Digital I/O	Digital I/O
HS	x	x	10	OSC1	OSC2
XT	x	x	01	OSC1	OSC2
EC	1	x	00	OSC1	Master Clock Output (FCY)
EC	0	1	00	—	Slave Clock Output (Slave FCY)
EC	0	0	00	—	Digital I/O

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

2: OSC2 pin function is determined by the Primary Oscillator Mode Selection (POSCMOD<1:0>), OSCIOFNC and S1OSCIOFNC Configuration bits.

3: If both the Master core OSCIOFNC and Slave core S1OSCIOFNC bits are set, the Master core OSCIOFNC bit has priority.

Figure 5-4: OSC2 Pin for Clock Output (in EC Mode)

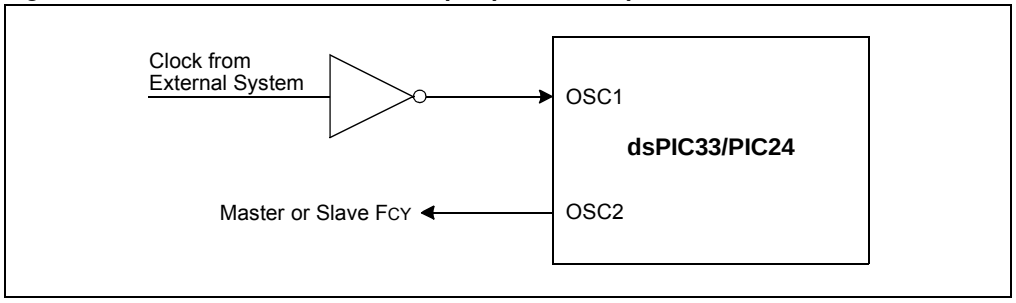
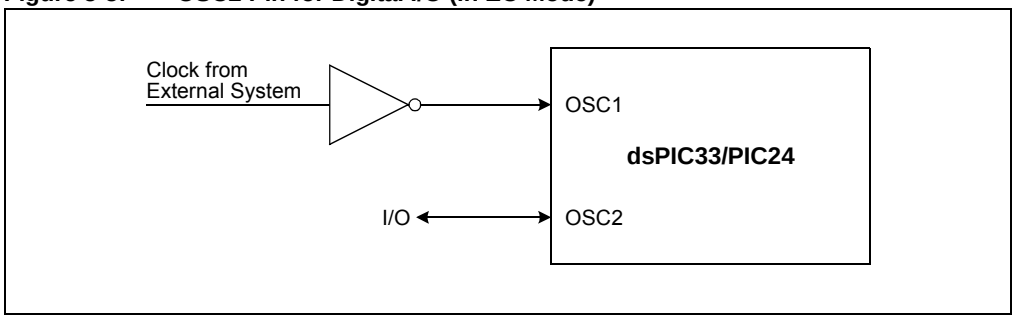


Figure 5-5: OSC2 Pin for Digital I/O (in EC Mode)



6.0 INTERNAL FAST RC (FRC) OSCILLATOR

The dsPIC33/PIC24 devices contain one instance of the Internal Fast RC (FRC) Oscillator, which is available to both the Master and Slave clock subsystems. The FRC Oscillator provides a nominal 8 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 using the FRC Oscillator Tuning bits (TUN<5:0>) in the FRC Oscillator Tuning register (OSCTUN<5:0>).

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

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

The FNOSC<2:0> and S1FNOSC<2:0> Configuration bits in the Master and Slave Oscillator Source Selection registers (FOSCSEL<2:0>, FS1OSCSEL<2:0>) specify the system clock source at Power-on Reset. The FRC clock source options at the time of a Power-on Reset are provided in Table 6-1. The Configuration bits are programmed at the time of device programming.

Table 6-1: FRC Clock Source Options

FNOSC<2:0> S1FNOSC<2:0> Value	Primary Oscillator Source and Mode
000	FRC Oscillator (FRC)
001	FRC Oscillator: Postscaler with PLL (FRCPLL)
110	Backup FRC Oscillator (BFRC)
111	FRC Oscillator: Postscaler Divide-by-N (FRCDIVN)

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 bits (FRCDIV<2:0>) in the Clock Divider register (CLKDIV<10:8>), which allows 8 settings, from 1:1 to 1:256, to be chosen.

Table 6-2: Internal Fast RC Oscillator Postscaler Settings

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

7.0 LOW-POWER RC (LPRC) OSCILLATOR

The dsPIC33/PIC24 devices contain one instance of the Low-Power RC (LPRC) Oscillator that is available to both the Master and Slave clock subsystems. The LPRC Oscillator provides a nominal clock frequency of 32 kHz and is the clock source for the Power-up Timer (PWRT), Watchdog Timer (WDT) and Fail-Safe Clock Monitor (FSCM) circuits in each core clock subsystem.

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 more information.

7.1 LPRC Oscillator for System Clock

The LPRC Oscillator is selected as the system clock when:

- The Initial Oscillator Source Selection bits (FNOSC<2:0>) in the Master or Slave core Oscillator Source Selection register (FOSCSEL<2:0> and FS1OSCSEL<2:0>, respectively) are appropriately set to select the LPRC Oscillator at a Power-on Reset
- User-assigned software initiates a Master or Slave core clock switch to the LPRC Oscillator for low-power operation

7.2 Enabling the LPRC Oscillator

The LPRC Oscillator is the clock source for the PWRT, WDT and FSCM in both the Master and Slave cores. The LPRC Oscillator is enabled at power-on.

The LPRC Oscillator remains enabled under these conditions:

- The Master or Slave FSCM is enabled
- The Master or Slave 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 is enabled and running automatically if either the WDT or clock fail detect is enabled. The LPRC runs in Sleep mode only if the Watchdog Timer is enabled. Under all other conditions, LPRC is disabled in Sleep mode.

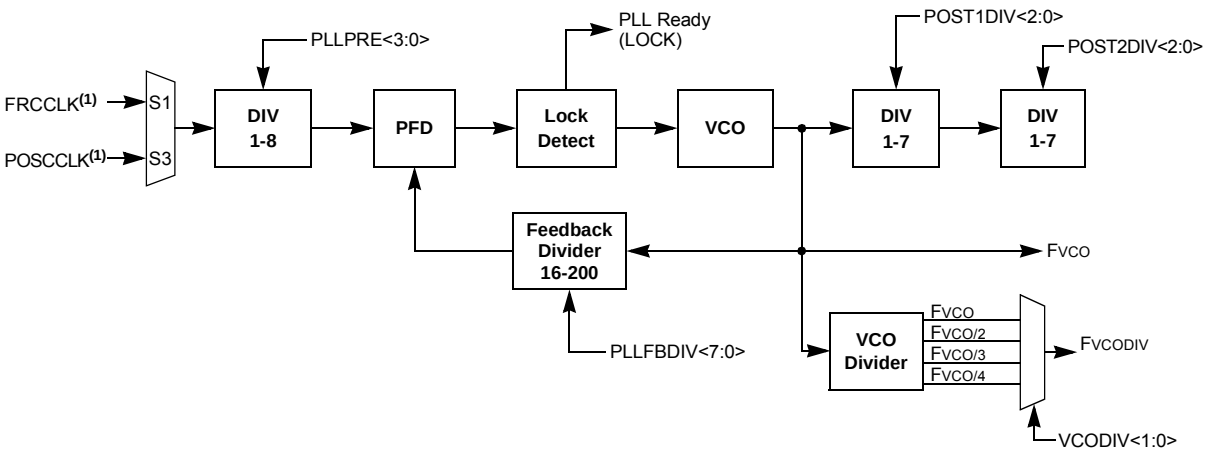
7.3 LPRC Oscillator Start-up Delay

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

8.0 MASTER PHASE-LOCKED LOOP (PLL)

The Primary Oscillator and Internal FRC Oscillator sources can optionally use an on-chip PLL to obtain higher operating speeds. Figure 8-1 illustrates a block diagram of the PLL module.

Figure 8-1: Master Core PLL and VCO Detail



Note 1: From Master and Slave core shared oscillator source.

For PLL operation, the following requirements must be met at all times without exception:

- The PLL Input Frequency (FPLLI) must be in the range of 8 MHz to 64 MHz
- The PFD Input Frequency (FPFD) must be in the range of 8 MHz to (FVCO/16) MHz
- The VCO Output Frequency (FVCO) must be in the range of 400 MHz to 1600 MHz

The PLL Phase Detector Input Divider Select bits (PLLPRE<3:0>) in the Clock Divider register (CLKDIV<3:0>) specify the input divider ratio (N1), which is used to scale down the input clock (FPLLI) to meet the PFD input frequency range of 8 MHz to (FVCO/16) MHz.

The PLL Feedback Divider bits (PLLFBDIV<7:0>) in the PLL Feedback Divider register (PLLFBD<7:0>) specify the divider ratio (M), which scales down the VCO Output Frequency (FVCO) for feedback to the PFD input. The VCO Frequency (FVCO) is 'M' times the PFD Input Frequency (FPFD).

There are two PLL VCO output dividers configured through the POST1DIV<2:0> and POST2DIV<2:0> select bits. These bits are located in the PLL Output Divider register (PLLDIV<6:4> and PLLDIV<2:0>) and specify the divider ratios (N2 and N3) that limit the PLL Output Frequency (FPLLO). FPLLO must not exceed 400 MHz (100 MIPS) when used as the Master core system clock.

Equation 8-1 provides the relationship between the PLL Input Frequency (FPLLI) and VCO Output Frequency (FVCO).

Equation 8-1: Master Core FVCO Calculation

$$FVCO = FPLLI \times \left(\frac{M}{N1}\right) = FPLLI \times \left(\frac{PLLFBDIV<7:0>}{PLLPRE<3:0>}\right)$$

Equation 8-2 provides the relationship between the PLL Input Frequency (FPLLI) and PLL Output Frequency (FPLLO).

Equation 8-2: Master Core FPLLO Calculation

$$FPLLO = FPLLI \times \left(\frac{M}{N1 \times N2 \times N3}\right) = FPLLI \times \left(\frac{PLLFBDIV<7:0>}{PLLPRE<3:0> \times POST1DIV<2:0> \times POST2DIV<2:0>}\right)$$

Where:

M = $PLLFBDIV<7:0>$
$N1$ = $PLLPRE<3:0>$
$N2$ = $POST1DIV<2:0>$
$N3$ = $POST2DIV<2:0>$

8.1 Input Clock Limitation at Start-up for PLL Mode

Table 8-1 provides the default values of the PLL prescaler, PLL feedback divider and both PLL postscalers at Power-on Reset (POR).

Table 8-1: Master Core PLL Mode Defaults

Register	Bit Field	Value at POR Reset	PLL Divider Ratio
CLKDIV<3:0>	PLLPRE<3:0>	0001	N1 = 1
PLLDIV<6:4>	POST1DIV<2:0>	100	N2 = 4
PLLDIV<2:0>	POST2DIV<2:0>	001	N3 = 1
PLLFBD<7:0>	PLLFBDIV<7:0>	10010110	M = 150

Given these Reset values, the following equations provide the PLL Input Frequency (FPLLI) and VCO Output Frequency (FVCO) at Power-on Reset.

Equation 8-3: Master Core FVCO at Power-on Reset

$$FVCO = FPLLI\left(\frac{M}{N1}\right) = FPLLI\left(\frac{150}{1}\right) = 150\ FPLLI$$

Equation 8-4: Master Core FPLLO at Power-on Reset

$$FPLLO = FPLLI\left(\frac{M}{N1 \times N2 \times N3}\right) = FPLLI\left(\frac{150}{1 \times 4 \times 1}\right) = 37.5\ FPLLI$$

To use the PLL with other than default settings, and to ensure all PLL requirements are met, follow this process:

1. Power up the device with the Internal FRC or the Primary Oscillator without PLL.
2. Change the PLLFBDIVx, PLLPREx, POST1DIVx and POST2DIVx bit values, based on the input frequency, to meet these PLL requirements:
   - The PLL Input Frequency (FPLLI) must be in the range of 8 MHz to 64 MHz
   - The PFD Input Frequency (FPFD) must be in the range of 8 MHz to (FVCO/16) MHz
   - The VCO Output Frequency (FVCO) must be in the range of 400 MHz to 1600 MHz
3. Switch the clock to the PLL mode in software.

Note: The PLL Phase Detector Input Divider Select (PLLPREx) and the PLL Feedback Divider (PLLFBDIVx) bits should not be changed when operating in PLL mode. Therefore, the user must start in either a non-PLL mode, or clock switch to a non-PLL mode (e.g., Internal FRC Oscillator), to make any necessary changes and then clock switch to the desired PLL mode.

It is not permitted to directly clock switch from one PLL clock source to a different PLL clock source. The user would need to transition between PLL clock sources with a clock switch to a non-PLL clock source.

8.2 PLL Lock Status

Whenever the PLL input frequency, the PLL prescaler or the PLL feedback divider is changed, the PLL requires a finite amount of time (TLOCK) to synchronize to the new settings.

TLOCK is applied when the 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 POSC, TLOCK starts after the OST delay. For more information about oscillator start-up delay, see **Section 5.1 "Oscillator Start-up Time"**. Also, refer to the **"Oscillator Configuration"** chapter in the specific device data sheet for more information about typical TLOCK values.

The LOCK bit in the Oscillator Control register (OSCCON<5>) is a read-only status bit that indicates the lock status of the PLL. The LOCK bit is cleared at a 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 advisable to wait for the LOCK bit to be set, before executing other code, after a clock switch event in which the PLL is enabled.

Note:	The PLL Phase Detector Input Divider Select (PLLPREx) and PLL Feedback Divider (PLLFBDIVx) bits 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.

8.3 PLL Setup

8.3.1 SETUP FOR USING PLL WITH THE PRIMARY OSCILLATOR (POSC)

The following process is used to set up the PLL to operate the device at 50 MIPS with a 10 MHz external crystal:

1. To execute instructions at 50 MHz, a PLL output frequency of 200 MHz will be required.
2. To set up the PLL and meet the requirements of the PLL, follow these steps:
   a) Select the PLL prescaler to meet the PFD input frequency requirement (8 MHz ≤ FPFD ≤ (FVCO/16) MHz).
      - Select a PLL prescaler value of N1 = 1
      - FPLLI = 10 MHz
      - FPFD = 10 MHz(1/N1) = 10 MHz(1) = 10 MHz

   b) Select the feedback divider to meet the VCO output frequency requirement (400 MHz ≤ FVCO ≤ 1.6 GHz), as well as achieve the desired FVCO frequency.
      - Select a feedback divider value of M = 100
      - FVCO = FPLLI x (M/N1) = 10 MHz x (100/1) = 1 GHz

   c) Select values for the first and second PLL postscalers to achieve the required FPLLO frequency.
      - Select values for the first and second postscalers of N2 = 5 and N3 = 1
      - FPLLO = FVCO/(N2 x N3) = 1 GHz/5 = 200 MHz

   d) Configure the FNOSC<2:0> bits (FOSCSEL<2:0>) to select a clock source without the PLL (for example, Internal FRC Oscillator) at Power-on Reset.
   e) In the main program, change the PLL prescaler, PLL postscalers and PLL feedback divider values to those just decided in the previous steps, and then perform a clock switch to the PLL mode.

Example 8-1 illustrates code for using the PLL with the Primary Oscillator. (Also, see **Section 13.0 "Clock Switching"** for example code for clock switching.)

Example 8-1: Code Example for Using PLL with the Primary Oscillator (POSC)

```
//code example for 50 MIPS system clock using POSC with 10 MHz external crystal

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

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

int main()
{
    // Configure PLL prescaler, both PLL postscalers, and PLL feedback divider
    CLKDIVbits.PLLPRE   = 1;          // N1=1
    PLLFBDbits.PLLFBDIV = 100;        // M = 100
    PLLDIVbits.POST1DIV = 5;          // N2=5
    PLLDIVbits.POST2DIV = 1;          // N3=1

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

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

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

8.3.2 SETUP FOR USING PLL WITH 8 MHz INTERNAL FRC

The following process is used to set up the PLL to operate the device at 50 MIPS with an 8 MHz Internal FRC.

1. To execute instructions at 50 MHz, a PLL output frequency of 200 MHz will be required.
2. To set up the PLL and meet the requirements of the PLL, follow these steps:
   a) Select the PLL prescaler to meet the PFD input frequency requirement (8 MHz ≤ FPFD ≤ (FVCO/16) MHz).
      - Select a PLL prescaler value of N1 = 1
      - FPLLI = 8 MHz
      - FPFD = 8 MHz(1/N1) = 8 MHz(1) = 8 MHz

   b) Select the feedback divider to meet the VCO output frequency requirement (400 MHz ≤ FVCO ≤ 1.6 GHz), as well as achieve the desired FVCO frequency.
      - Select a feedback divider value of M = 125
      - FVCO = FPLLI x (M/N1) = 8 MHz x (125/1) = 1 GHz

   c) Select values for the first and second PLL postscalers to achieve the required FPLLO frequency.
      - Select values for the first and second postscalers of N2 = 5 and N3 = 1
      - FPLLO = FVCO/(N2 x N3) = 1 GHz/5 = 200 MHz

   d) Configure the FNOSC<2:0> bits (FOSCSEL<2:0>) to select a clock source without the PLL (for example, Internal FRC Oscillator) at Power-on Reset.

   e) In the main program, change the PLL prescaler, PLL postscalers and PLL feedback divider values to those just decided in the previous steps, and then perform a clock switch to the PLL mode.

Example 8-2 illustrates code for using the PLL with an 8 MHz Internal FRC Oscillator. (Also, see **Section 13.0 "Clock Switching"** for example code for clock switching.)

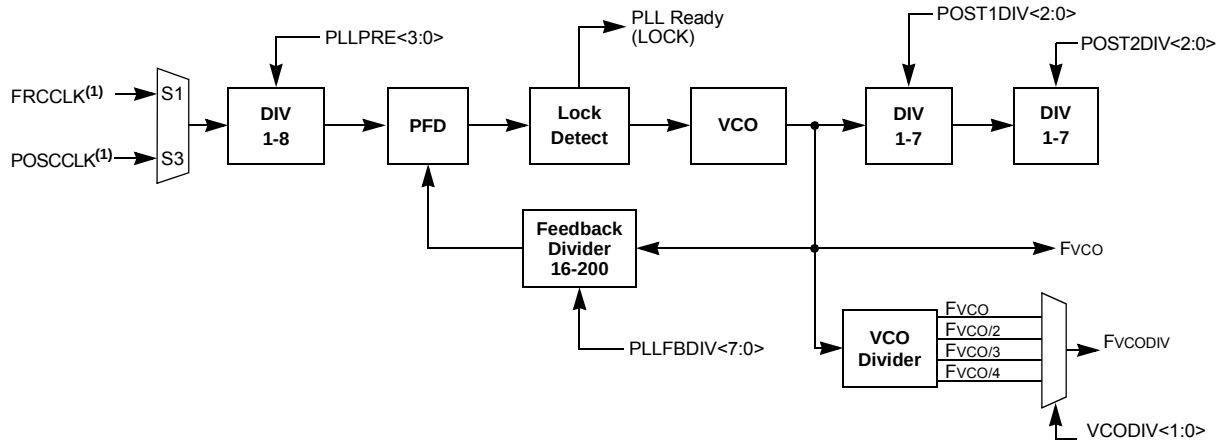
Example 8-2: Code Example for Using PLL with 8 MHz Internal FRC

```
//code example for 50 MIPS system clock using 8MHz FRC

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

// Enable Clock Switching
_FOSC(FCKSM_CSECMD);

int main()
{
    // Configure PLL prescaler, both PLL postscalers, and PLL feedback divider
    CLKDIVbits.PLLPRE   = 1;          // N1=1
    PLLFBDbits.PLLFBDIV = 125;        // M = 125
    PLLDIVbits.POST1DIV = 5;          // N2=5
    PLLDIVbits.POST2DIV = 1;          // N3=1

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

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

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

9.0 SLAVE PHASE-LOCKED LOOP (PLL)

The Primary Oscillator and Internal FRC Oscillator sources can optionally use an on-chip PLL to obtain higher operating speeds. Figure 9-1 illustrates a block diagram of the PLL module.

Figure 9-1: Slave Core PLL and VCO Detail



Note 1: From Master and Slave core shared oscillator source.

For PLL operation, the following requirements must be met at all times without exception:

- The PLL Input Frequency (FPLLI) must be in the range of 8 MHz to 64 MHz
- The PFD Input Frequency (FPFD) must be in the range of 8 MHz to (FVCO/16) MHz
- The VCO Output Frequency (FVCO) must be in the range of 400 MHz to 1600 MHz

The PLL Phase Detector Input Divider Select bits (PLLPRE<3:0>) in the Clock Divider register (CLKDIV<3:0>) specify the input divider ratio (N1), which is used to scale down the input clock (FPLLI) to meet the PFD input frequency range of 8 MHz to (FVCO/16) MHz.

The PLL Feedback Divider bits (PLLFBDIV<7:0>) in the PLL Feedback Divider register (PLLFBD<7:0>) specify the divider ratio (M), which scales down the VCO Frequency (FVCO) for feedback to the PFD input. The VCO Frequency (FVCO) is 'M' times the PFD Input Frequency (FPFD).

There are two PLL VCO output dividers configured through the POST1DIV<2:0> and POST2DIV<2:0> select bits. These bits are located in the PLL Output Divider register (PLLDIV<6:4> and PLLDIV<2:0>) and specify the divider ratios (N2 and N3) that limit the PLL Output Frequency (FPLLO). FPLLO must not exceed 480 MHz (120 MIPS) when used as the Slave core system clock.

Equation 9-1 provides the relationship between the PLL Input Frequency (FPLLI) and VCO Output Frequency (FVCO).

Equation 9-1: Slave Core Fvco Calculation

$$FVCO = FPLLI \times \left(\frac{M}{N1}\right) = FPLLI \times \left(\frac{PLLFBDIV<7:0>}{PLLPRE<3:0>}\right)$$

Equation 9-2 provides the relationship between the PLL Input Frequency (FPLLI) and PLL Output Frequency (FPLLO).

Equation 9-2: Slave Core FPLLO Calculation

$$FPLLO = FPLLI \times \left(\frac{M}{N1 \times N2 \times N3}\right) = FPLLI \times \left(\frac{PLLFBDIV<7:0>}{PLLPRE<3:0> \times POST1DIV<2:0> \times POST2DIV<2:0>}\right)$$

Where:

M = *PLLFBDIV*<7:0>
N1 = *PLLPRE*<3:0>
N2 = *POST1DIV*<2:0>
N3 = *POST2DIV*<2:0>

9.1 Input Clock Limitation at Start-up for PLL Mode

Table 9-1 provides the default values of the PLL prescaler, PLL postscaler and PLL feedback divider bits at a Power-on Reset.

Table 9-1: Slave Core PLL Mode Defaults

Register	Bit Field	Value at POR Reset	PLL Divider Ratio
CLKDIV<3:0>	PLLPRE<3:0>	0001	N1 = 1
PLLDIV<6:4>	POST1DIV<2:0>	100	N2 = 4
PLLDIV<2:0>	POST2DIV<2:0>	001	N3 = 1
PLLFBD<7:0>	PLLFBDIV<7:0>	10010110	M = 150

Given these Reset values, the following equations provide the relationship between the PLL Input Frequency (FPLLI), PFD Input Frequency (FPFD), VCO Output Frequency (FVCO) and System Clock Frequency (FOSC) at a Power-on Reset.

Equation 9-3: Slave Core FVCO at Power-on Reset

$$FVCO = FPLLI\left(\frac{M}{N1}\right) = FPLLI\left(\frac{150}{1}\right) = 150\ FPLLI$$

Equation 9-4: Slave Core FPLLO at Power-on Reset

$$FPLLO = FPLLI\left(\frac{M}{N1 \times N2 \times N3}\right) = FPLLI\left(\frac{150}{1 \times 4 \times 1}\right) = 37.5\ FPLLI$$

To use the PLL with other than default settings, and to ensure all PLL requirements are met, follow this process:

1. Power up the device with the Internal FRC or the Primary Oscillator without PLL.
2. Change the PLLFBDIVx, PLLPREx, POST1DIVx and POST2DIVx bit values, based on the input frequency, to meet these PLL requirements:
   - The PLL Input Frequency (F_{PLLI}) must be in the range of 8 MHz to 64 MHz
   - The PFD Input Frequency (F_{PFD}) must be in the range of 8 MHz to (F_{VCO}/16) MHz
   - The VCO Output Frequency (F_{VCO}) must be in the range of 400 MHz to 1600 MHz
3. Switch the clock to the PLL mode in software.

Note: The PLL Phase Detector Input Divider Select (PLLPREx) and PLL Feedback Divider (PLLFBDIVx) bits should not be changed when operating in PLL mode. Therefore, the user must either start in a non-PLL mode or clock switch to a non-PLL mode (e.g., Internal FRC Oscillator) to make the necessary changes and then clock switch to the desired PLL mode.

It is not permitted to directly clock switch from one PLL clock source to a different PLL clock source. The user would need to transition between PLL clock sources with a clock switch to a non-PLL clock source.

9.2 PLL Lock Status

Whenever the PLL input frequency, the PLL prescaler or the PLL feedback divider is changed, the PLL requires a finite amount of time (TLOCK) to synchronize to the new settings.

TLOCK is applied when the PLL is selected as the clock source at a 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 POSC, TLOCK starts after the OST delay. For more information about oscillator start-up delay, see **Section 5.1 "Oscillator Start-up Time"**. Also, refer to the **"Oscillator Configuration"** chapter in the specific device data sheet for more information about typical TLOCK values.

The LOCK bit in the Oscillator Control register (OSCCON<5>) is a read-only status bit that indicates the lock status of the PLL. The LOCK bit is cleared at a 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 advisable to wait for the LOCK bit to be set before executing other code after a clock switch event in which the PLL is enabled.

Note:	The PLL Phase Detector Input Divider Select (PLLPREx) and PLL Feedback Divider (PLLFBDIVx) bits should not be changed when operating in PLL mode. You must clock switch to a non-PLL mode (e.g., Internal FRC Oscillator) to make the necessary changes and then clock switch back to the PLL mode.

9.3 PLL Setup

9.3.1 SETUP FOR USING PLL WITH THE PRIMARY OSCILLATOR (POSC)

The following process is used to set up the PLL to operate the device at 60 MIPS with a 10 MHz external crystal:

1. To execute instructions at 60 MHz, ensure the required system clock frequency is: FOSC = 2 x FCY = 120 MHz.
2. To set up the PLL and meet the requirements of the PLL, follow these steps:
   a) Select the PLL prescaler to meet the PFD input frequency requirement (8 MHz ≤ FPFD ≤ (FVCO/16) MHz).
      - Select a PLL prescaler value of N1 = 1
      - FPLLI = 10 MHz
      - FPFD = 10 MHz (1/N1) = 10 MHz(1) = 10 MHz

   b) Select the feedback divider to meet the VCO output frequency requirement (400 MHz ≤ FVCO ≤ 1.6 GHz) as well as achieve the desired FVCO frequency.
      - Select a feedback divider value of M = 120
      - FVCO = FPLLI x (M/N1) = 10 MHz x (120/1) = 1.2 GHz

   c) Select values for the first and second PLL postscalers to achieve the required FPLLO frequency.
      - Select values for the first and second postscalers of N2 = 5 and N3 = 1
      - FPLLO = FVCO/(N2 x N3) = 1.2 GHz/5 = 240 MHz

   d) Configure the FNOSC<2:0> bits (FOSCSEL<2:0>) to select a clock source without the PLL (for example, Internal FRC Oscillator) at a Power-on Reset.
   e) In the main program, change the PLL prescaler, PLL postscalers and PLL feedback divider values to those just decided in the previous steps, and then perform a clock switch to the PLL mode.

Example 9-1 illustrates code for using the PLL with the Primary Oscillator. (Also, see **Section 13.0 "Clock Switching"** for example code for clock switching.)

Example 9-1: Code Example for Using PLL with the Primary Oscillator (POSC)

```
//code example for 60 MIPS system clock using POSC with 10 MHz external crystal
// Select Internal FRC at POR
_FS1OSCSEL(S1FNOSC_FRC & S1IESO_OFF);

// Enable Clock Switching
_FS1OSC(S1FCKSM_CSECMD);

//Configure POSC in XT mode in Master core FOSC configuration register
_FOSC(POSCMD_XT);

int main()
{
    // Configure PLL prescaler, both PLL postscalers, and PLL feedback divider
    CLKDIVbits.PLLPRE   = 1;          // N1=1
    PLLFBDbits.PLLFBDIV = 120;        // M = 120
    PLLDIVbits.POST1DIV = 5;          // N2=5
    PLLDIVbits.POST2DIV = 1;          // N3=1

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

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

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

9.3.2 SETUP FOR USING PLL WITH 8 MHz INTERNAL FRC

The following process is used to set up the PLL to operate the device at 60 MIPS with an 8 MHz Internal FRC Oscillator.

1. To execute instructions at 60 MHz, a PLL output frequency of 240 MHz will be required.
2. To set up the PLL and meet the requirements of the PLL, follow these steps:
   a) Select the PLL prescaler to meet the PFD input frequency requirement (8 MHz ≤ F_{PFD} ≤ (F_{VCO}/16) MHz).
      - Select a PLL prescaler value of N1 = 1
      - F_{PLLI} = 8 MHz
      - F_{PFD} = 8 MHz (1/N1) = 8 MHz(1) = 8 MHz

   b) Select the feedback divider to meet the VCO output frequency requirement (400 MHz ≤ F_{VCO} ≤ 1.6 GHz), as well as achieve the desired F_{VCO} frequency.
      - Select a feedback divider value of M = 150
      - F_{VCO} = F_{PLLI} x (M/N1) = 8 MHz x (150/1) = 1.2 GHz

   c) Select values for the first and second postscalers to achieve the required F_{PLLO} frequency.
      - Select values for the first and second postscalers of N2 = 5 and N3 = 1
      - F_{PLLO} = F_{VCO}/(N2 x N3) = 1.2 GHz/5 = 240 MHz

   d) Configure the FNOSC<2:0> bits (FOSCSEL<2:0>) to select a clock source without the PLL (for example, Internal FRC Oscillator) at Power-on Reset.

   e) In the main program, change the PLL prescaler, PLL postscalers and PLL feedback divider to those just decided in the previous steps, and then perform a clock switch to the PLL mode.

Example 9-2 illustrates code for using the PLL with an 8 MHz Internal FRC Oscillator. (Also, see **Section 13.0 "Clock Switching"** for example code for clock switching.)

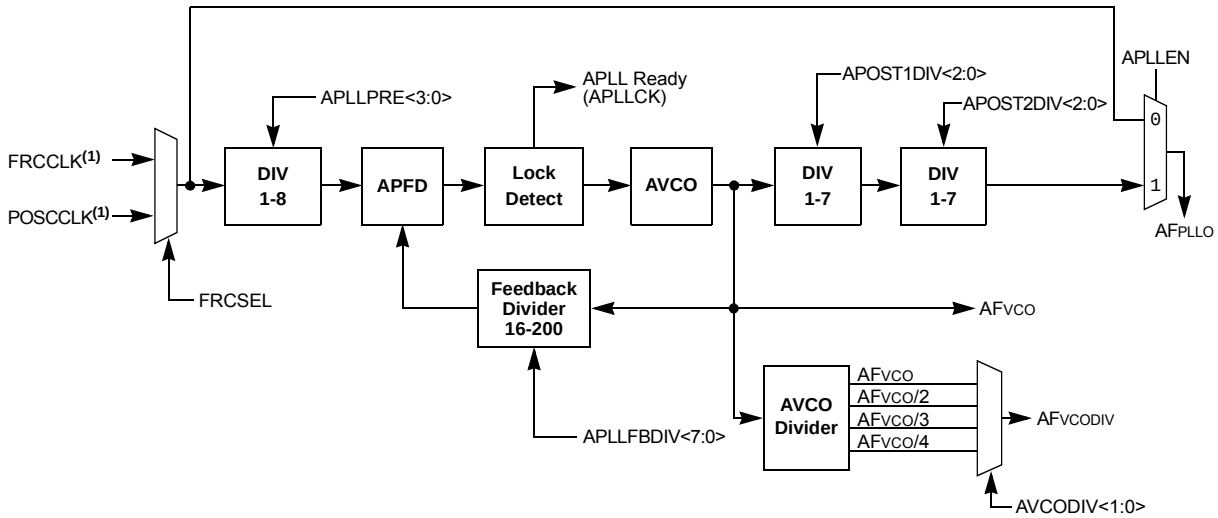
Example 9-2: Code Example for Using PLL with 8 MHz Internal FRC

```
//code example for 60 MIPS system clock using 8MHz FRC

// Select Internal FRC at POR
_FS1OSCSEL(S1FNOSC_FRC & S1IESO_OFF);

// Enable Clock Switching
_FS1OSC(S1FCKSM_CSECMD);


int main()
{
    // Configure PLL prescaler, both PLL postscalers, and PLL feedback divider
    CLKDIVbits.PLLPRE   = 1;          // N1=1
    PLLFBDbits.PLLFBDIV = 150;        // M = 150
    PLLDIVbits.POST1DIV = 5;          // N2=5
    PLLDIVbits.POST2DIV = 1;          // N3=1

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

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

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

10.0 MASTER AUXILIARY PHASE-LOCKED LOOP (APLL)

The dsPIC33/PIC24 device families implement an Auxiliary PLL module for each core present. The Auxiliary PLL is used to generate various peripheral clock sources independent of the system clock. Figure 10-1 shows a block diagram of the Master core APLL module.

Figure 10-1: Master Core APLL and VCO Detail



Note 1: From Master and Slave core shared oscillator source.

For APLL operation, the following requirements must be met at all times without exception:

- The APLL Input Frequency (AFPLLI) must be in the range of 8 MHz to 64 MHz
- The APFD Input Frequency (AFPFD) must be in the range of 8 MHz to (AFVCO/16) MHz
- The AVCO Output Frequency (AFVCO) must be in the range of 400 MHz to 1600 MHz

The APLL Phase Detector Input Divider Select bits (APLLPRE<3:0>) in the Auxiliary Clock Control register (ACLKCON1<3:0>) specify the input divider ratio (N1), which is used to scale down the input clock (AFPLLI) to meet the APFD input frequency range of 8 MHz to (AFVCO/16) MHz.

The APLL Feedback Divider bits (APLLFBDIV<7:0>) in the APLL Feedback Divider register (APLLFBD1<7:0>) specify the divider ratio (M), which scales down the AVCO Frequency (AFVCO) for feedback to the APFD input. The AVCO Output Frequency (AFVCO) is 'M' times the APFD Input Frequency (AFPFD).

There are two PLL VCO output dividers configured through the APOST1DIV<2:0> and APOST2DIV<2:0> select bits. These bits are located in the APLL Output Divider register (APLLDIV1<6:4> and APLLDIV1<2:0>) and specify the divider ratios (N2 and N3) that limit the APLL Output Frequency (AFPLLO).

Equation 10-1 provides the relationship between the APLL Input Frequency (AFPLLI) and the AVCO Output Frequency (AFVCO).

Equation 10-1: Master Core AFVCO Calculation

$$AF_{VCO} = AF_{PLLI} \times \left(\frac{M}{N1}\right) = AF_{PLLI} \times \left(\frac{APLLFBDIV<7:0>}{APLLPRE<3:0>}\right)$$

Equation 10-2 provides the relationship between the APLL Input Frequency (AF_{PLLI}) and APLL Output Frequency (AF_{PLLO}).

Equation 10-2: Master Core AF_{PLLO} Calculation

$$AF_{PLLO} = AF_{PLLI} \times \left(\frac{M}{N1 \times N2 \times N3}\right) = AF_{PLLI} \times \left(\frac{APLLFBDIV<7:0>}{APLLPRE<3:0> \times APOST1DIV<2:0> \times APOST2DIV<2:0>}\right)$$

Where:

M = $APLLFBDIV<7:0>$
$N1$ = $APLLPRE<3:0>$
$N2$ = $APOST1DIV<2:0>$
$N3$ = $APOST2DIV<2:0>$

10.1 APLL Setup

10.1.1 SETUP FOR USING APLL WITH THE 8 MHz INTERNAL FRC OSCILLATOR

1. Set the FRCSEL bit to choose the Internal FRC Oscillator as the clock source for the APLL.
2. To set up the APLL and meet the requirements of the APLL, follow these steps:
   a) Select the APLL prescaler to meet the APFD input frequency requirement (8 MHz ≤ AF_{PFD} ≤ (AF_{VCO}/16) MHz).
      - Select an APLL prescaler value of N1 = 1
      - AF_{PLLI} = 8 MHz
      - AF_{PFD} = 8 MHz (1/N1) = 8 MHz(1) = 8 MHz

   b) Select the feedback divider to meet the AVCO output frequency requirement (400 MHz ≤ AF_{VCO} ≤ 1.6 GHz) as well as achieve the desired AF_{VCO} frequency.
      - Select a feedback divider value of M = 125
      - AF_{VCO} = AF_{PLLI} x (M/N1) = 8 MHz x (125/1) = 1 GHz

   c) Select values for the first and second postscalers to achieve the required AF_{PLLO} frequency based on the AF_{VCO} frequency.
      - Select values for the first and second postscalers of N2 = 2 and N3 = 1
      - AF_{PLLO} = AF_{VCO}/(N2 x N3) = 1 GHz/2 = 500 MHz
3. Enable the APLL post-divider output and disable the APLL reference bypass output by setting the APLLEN bit.

Example 10-1 provides code for using the APLL with the Internal FRC Oscillator.

Example 10-1: Code Example for Using the APLL with the Internal FRC Oscillator

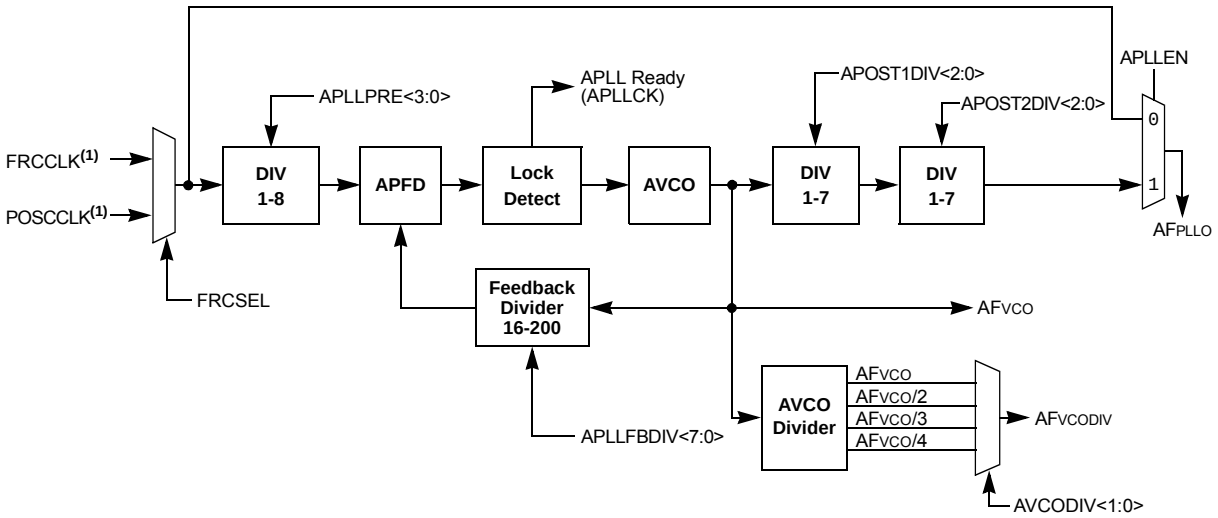
```
//code example for AFVCO = 1 GHz and APLLO = 500 MHz using 8 MHz internal FRC

// Configure the source clock for the APLL
ACLKCON1bits.FRCSEL = 1;          // Select internal FRC as the clock source

// Configure the APLL prescaler, APLL feedback divider, and both APLL postscalers.
ACLKCON1bits.APLLPRE   = 1;       // N1 = 1
APLLFBD1bits.APLLFBDIV = 125;     // M = 125
APLLDIV1bits.APOST1DIV = 2;       // N2 = 2
APLLDIV1bits.APOST2DIV = 1;       // N3 = 1

// Enable APLL
ACLKCON1bits.APLLEN = 1;
```

Note: Even with the APLLEN bit set, another peripheral must generate a clock request before the APLL will start.

11.0 SLAVE AUXILIARY PHASE-LOCKED LOOP (APLL)

The dsPIC33/PIC24 device families implement an Auxiliary PLL module for each core present. The Auxiliary PLL is used to generate various peripheral clock sources independent of the system clock. Figure 11-1 shows a block diagram of the Slave core APLL module.

Figure 11-1: Slave Core APLL and VCO Detail



Note 1: From Master and Slave core shared oscillator source.

For APLL operation, the following requirements must be met at all times without exception:

- The APLL Input Frequency (AFPLLI) must be in the range of 8 MHz to 64 MHz
- The APFD Input Frequency (AFPFD) must be in the range of 8 MHz to (AFVCO/16) MHz
- The AVCO Output Frequency (AFVCO) must be in the range of 400 MHz to 1600 MHz

The APLL Phase Detector Input Divider Select bits (APLLPRE<3:0>) in the Auxiliary Clock Control register (ACLKCON1<3:0>) specify the input divider ratio (N1), which is used to scale down the input clock (AFPLLI) to meet the APFD input frequency range of 8 MHz to (AFVCO/16) MHz.

The APLL Feedback Divider bits (APLLFBDIV<7:0>) in the APLL Feedback Divider register (APLLFBD1<7:0>) specify the divider ratio (M), which scales down the AVCO Frequency (AFVCO) for feedback to the APFD input. The AVCO Output Frequency (AFVCO) is 'M' times the APFD Input Frequency (AFPFD).

There are two PLL VCO output dividers configured through the APOST1DIV<2:0> and APOST2DIV<2:0> select bits. These bits are located in the APLL Output Divider register (APLLDIV1<6:4> and APLLDIV1<2:0>) and specify the divider ratios (N2 and N3) that limit the APLL Output Frequency (AFPLLO).

Equation 11-1 provides the relationship between the APLL Input Frequency (AFPLLI) and AVCO Output Frequency (AFVCO).

Equation 11-1: Slave Core AFVCO Calculation

$$AF_{VCO} = AF_{PLLI} \times \left(\frac{M}{N1}\right) = AF_{PLLI} \times \left(\frac{APLLFBDIV<7:0>}{APLLPRE<3:0>}\right)$$

Equation 11-2 provides the relationship between the APLL Input Frequency (AFPLLI) and APLL Output Frequency (AFPLLO).

Equation 11-2: Slave Core AFPLLO Calculation

$$AFPLLO = AFPLLI \times \left(\frac{M}{N1 \times N2 \times N3}\right) = AFPLLI \times \left(\frac{APLLFBDIV<7:0>}{APLLPRE<3:0> \times APOST1DIV<2:0> \times APOST2DIV<2:0>}\right)$$

Where:

M = $APLLFBDIV<7:0>$
$N1$ = $APLLPRE<3:0>$
$N2$ = $APOST1DIV<2:0>$
$N3$ = $APOST2DIV<2:0>$

11.1 APLL Setup

11.1.1 SETUP FOR USING APLL WITH THE 8 MHz INTERNAL FRC OSCILLATOR

1. Set the FRCSEL bit to choose the Internal FRC Oscillator as the clock source for the APLL.
2. To set up the APLL and meet the requirements of the APLL, follow these steps:
   a) Select the APLL prescaler to meet the APFD input frequency requirement (8 MHz ≤ AFPFD ≤ (AFVCO/16) MHz).
      - Select an APLL prescaler value of N1 = 1
      - AFPLLI = 8 MHz
      - AFPFD = 8 MHz(1/N1) = 8 MHz(1) = 8 MHz

   b) Select the feedback divider to meet the AVCO output frequency requirement (400 MHz ≤ AFVCO ≤ 1.6 GHz), as well as achieve the desired AFVCO frequency.
      - Select a feedback divider value of M = 125
      - AFVCO = AFPLLI x (M/N1) = 8 MHz x (125/1) = 1 GHz

   c) Select values for the first and second postscalers to achieve the required AFPLLO frequency.
      - Select values for the first and second postscalers of N2 = 2 and N3 = 1
      - AFPLLO = AFVCO/(N2 x N3) = 1 GHz/2 = 500 MHz
3. Enable the APLL post-divider output and disable the APLL reference bypass output by setting the APLLEN bit.

Example 11-1 provides code for using the APLL with the Internal FRC Oscillator.

Example 11-1: Code Example for Using the APLL with the Internal FRC Oscillator

```
//code example for AFVCO = 1 GHz and APLLO = 500 MHz using 8 MHz internal FRC

// Configure the source clock for the APLL
ACLKCON1bits.FRCSEL    = 1;       // Select internal FRC as the clock source

// Configure the APLL prescaler, APLL feedback divider, and both APLL postscalers.
ACLKCON1bits.APLLPRE   = 1;       // N1 = 1
APLLFBD1bits.APLLFBDIV = 125;     // M = 125
APLLDIV1bits.APOST1DIV = 2;       // N2 = 2
APLLDIV1bits.APOST2DIV = 1;       // N3 = 1

// Enable APLL
ACLKCON1bits.APLLEN = 1;
```

Note: Even with the APLLEN bit set, another peripheral must generate a clock request before the APLL will start.

12.0 FAIL-SAFE CLOCK MONITOR (FSCM)

The dsPIC33/PIC24 device families implement Fail-Safe Clock Monitoring (FSCM) protection for each of their cores. 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 Configuration bits (FCKSM<1:0>) in the Master or Slave Oscillator Configuration register (FOSC<7:6> and FS1OSC<7:6>, respectively) at the time of device programming. When FSCM is enabled (FCKSM<1:0> = 00) for any core, 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 Backup FRC Oscillator. The user-assigned application has the option to either attempt to restart the oscillator or execute a controlled shutdown.

Note:	When the device is in Sleep mode, if the clock fails, the FSCM does not wake-up the device.

The FSCM module takes the following actions when it switches to the Backup FRC Oscillator:

- The Current Oscillator Selection bits, COSC<2:0> (OSCCON<14:12>), are loaded with '110' (Backup FRC Oscillator).
- The Clock Fail (CF) Detect bit (OSCCON<3>) is set to indicate the clock failure.
- The Oscillator Switch Enable (OSWEN) bit (OSCCON<0>) is cleared to cancel any pending clock switches.

12.1 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.

For more information, refer to **"Reset"** in the *"dsPIC33/PIC24 Family Reference Manual"* (DS39712). For recent documentation, visit the Microchip web site at www.microchip.com.

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

12.2 FSCM and WDT

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

13.0 CLOCK SWITCHING

Clock switching can be initiated as a result of a hardware event or a software request. A typical scenario includes:

- 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 Backup FRC Oscillator on a clock failure.
- User-assigned 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<2:0> 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. That is, the new clock source is ready before the old clock is deactivated and code continues to execute as clock switching occurs.

With few limitations, applications are free to switch between any of the three clock sources (POSC, FRC and LPRC), under software control, at any time. To limit the possible side effects that could result from this flexibility, dsPIC33/PIC24 devices have a safeguard lock built into the switch process. That is, the OSCCON register is write-protected during clock switching.

13.1 Enabling Clock Switching

The Clock Switching Mode Configuration bits (FCKSM<1:0> or S1FCKSM<1:0>) in the Master or Slave Oscillator Configuration register (FOSC<7:6> and FS1OSC<7:6>, respectively) must be programmed to enable clock switching and the Fail-Safe Clock Monitor for that core (see Table 13-1).

Table 13-1: 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.

13.2 Clock Switch Sequence

The recommended process for a clock switch is as follows:

1. Read the COSC<2:0> 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<2:0> 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 previous steps are completed, the clock switch logic performs the following tasks:

1. The clock switching hardware compares the COSC<2:0> status bits (OSCCON<14:12>) with the new value of the NOSC<2:0> control bits (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 CF (OSCCON<3>) status bits are cleared.
3. The new oscillator is turned on by the hardware (if it is not running). If a crystal oscillator must be turned on, the hardware waits for TOSCD until the crystal starts oscillating and 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<2:0> bits (OSCCON<10:8>) value is transferred to the COSC<2:0> 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 is enabled). The timing of the transition between clock sources is illustrated in Figure 13-1.

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

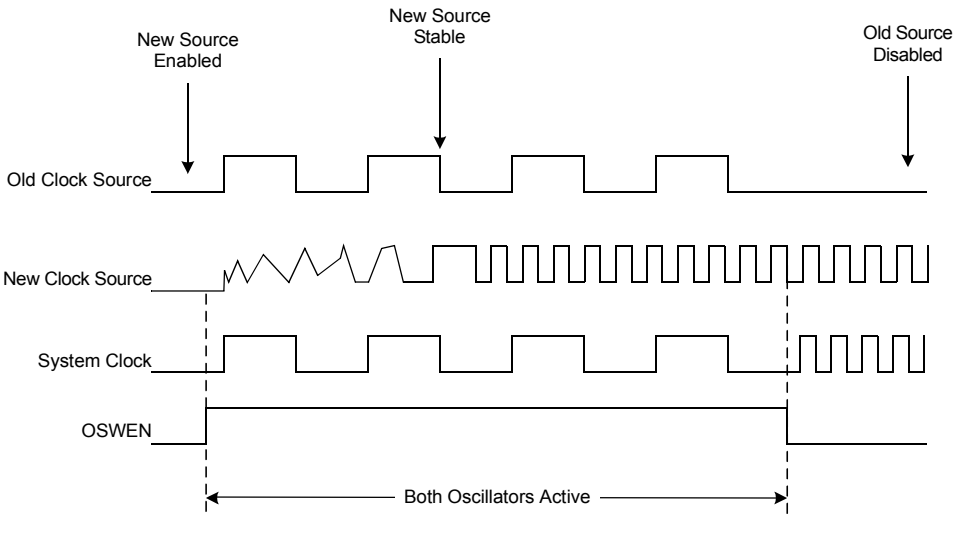
2: Direct clock switching between PLL modes is not possible. For example, clock switching should not occur between the Primary Oscillator with PLL and the Internal FRC Oscillator with PLL.

3: Setting the CLKLOCK bit (OSCCON<7>) prevents clock switching when clock switching is enabled and Fail-Safe Clock Monitoring is disabled. The CLKLOCK bit cannot be cleared after it is set by the software; it clears on a 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.

5: The clock switch for a given core will not wait for the PLL lock if the (S1)PLLKEN bit in the Master or Slave Oscillator Configuration register (FOSC<8> and FS1OSC<8>, respectively) is set to '0'.

Figure 13-1: Clock Transition Timing Diagram



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

The following steps are the recommended code sequence for a clock switch:

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 0x78 to OSCCON<15:8>
   - Write 0x9A to OSCCON<15:8>
3. In the instruction immediately following the unlock sequence, write the new oscillator source to the NOSC<2:0> control bits (OSCCON<10:8>).
4. Execute the unlock sequence for the OSCCON low byte.
   In two back-to-back instructions:
   - Write 0x46 to OSCCON<7:0>
   - Write 0x57 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 the OSWEN bit 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_OSCCONH(value)
__builtin_write_OSCCONL(OSCCON | value)
```

For more information, see the MPLAB IDE Help file.

Example 13-1 illustrates the code sequence for unlocking the OSCCON register and switching from FRC with the PLL clock to the LPRC clock source.

Example 13-1: Code Example for Clock Switching

```
;Place the New Oscillator Selection (NOSC=0b101) in W0
MOV     #0x5, WREG

;OSCCONH (high byte) Unlock Sequence
MOV     #OSCCONH, w1
MOV     #0x78, w2
MOV     #0x9A, w3
MOV.B   w2, [w1]        ;Write 0x78
MOV.B   w3, [w1]        ;Write 0x9A

;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 0x46
MOV.B   w3, [w1]        ;Write 0x57

; Enable Clock Switch
BSET    OSCON, #0       ;Request Clock Switching by Setting OSWEN bit

wait:
    btsc    OSCCONL, #OSWEN
    bra     wait
```

13.3 Clock Switching Consideration

When you incorporate clock switching into an application, consider these points when designing the code:

- The OSCCON unlock sequence is extremely timing-critical. The OSCCON register byte is only writable for one instruction cycle following the sequence. Some high-level languages, such as C, may not preserve the timing-sensitive sequence of instructions when compiled. When clock switching is required for an application written in a high-level language, it is a good idea to create the routine in an assembler and link it to the application, and then call 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 (OST).
- 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. User-assigned 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 the lock has been achieved. User-assigned 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.

13.4 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-up 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 13-2. 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 13-2: 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
```

13.5 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 useful to perform a clock switch to the Internal FRC Oscillator before entering Sleep mode, as this will ensure fast wake-up from Sleep mode.

14.0 TWO-SPEED START-UP

The Internal External Start-up Option Configuration bit (IESO and S1IESO) in the Master or Slave Oscillator Source Selection register (FOSCSEL<7> and FS1OSCSEL<7>, respectively) specifies whether to start the device with a user-selected oscillator source or to initially start with the Internal FRC Oscillator and then switch to the user-selected oscillator. If this bit is set to '1', the device will always power-up on the Internal FRC Oscillator, then switch to the specified oscillator when it is ready.

The Two-Speed Start-up option is a faster way to get the device up and running, and works independently from the state of the Master or Slave Clock Switching Mode Configuration bits, FCKSM<1:0> and S1FCKSM<1:0> (FOSC<7:6> and FS1OSC<7:6>, respectively).

Two-Speed Start-up is useful when an external oscillator with a long start-up delay is selected by the Master or Slave core FNOSC<2:0> and S1FNOSC<2:0> Configuration bits (FOSCSEL<2:0> and FS1OSCSEL<2:0>, respectively). As an Internal RC Oscillator, the FRC clock source is available immediately following a 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; at which time, it switches to that source.

User code can check which clock source is currently providing clocking to a given core by checking the status of the Master or Slave core COSC<2:0> bits (OSCCON<14:12>) against the NOSC<2:0> bits (OSCCON<10:8>). If these two sets of bits match, the clock switch has 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.

15.0 REFERENCE CLOCK OUTPUT

The dsPIC33/PIC24 device families provide a Reference Clock output for each implemented core. The Reference Clock output provides a clock signal to any remappable pin (RPn).

The ROSEL<3:0> bits (REFOCONL<3:0>) in the Master or Slave Reference Clock Control Low register select between various clock sources for the Reference Clock output for a given core.

The RODIV<14:0> bits (REFOCONH<14:0>) in the Master or Slave Reference Clock Control High register configure the Reference Clock output divider for a given core.

Figure 1-1 shows a block diagram for the Reference Clock. See the Master or Slave REFOCONL/H register (Register 4-10, Register 4-11, Register 4-19 and Register 4-20, respectively) for the bits associated with the Reference Clock output. Refer to the specific device data sheet for more information on peripheral remapping.

16.0 REGISTER MAPS

Table 16-1 and Table 16-2 map the bit functions for the Oscillator Module with High-Speed PLL Special Function Control registers. Table 16-3 and Table 16-4 map the bit functions for the Oscillator Configuration registers.

Table 16-1: Master 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	—	CF	—	—	OSWEN	7700(1)
CLKDIV	ROI	DOZE<2:0>			DOZEN	FRCDIV<2:0>			—	—	r(2)	r(2)	PLLPRE<3:0>				3001
PLLFBD	—	—	—	—	r(2)	r(2)	r(2)	r(2)	PLLFBDIV<7:0>								0096
PLLDIV	—	—	—	—	—	—	VCODIV<1:0>		—	POST1DIV<2:0>			—	POST2DIV<2:0>			0031
OSCTUN	—	—	—	—	—	—	—	—	—	—	TUN<5:0>						0000
ACLKCON1	APLLEN	APLLCK	—	—	—	—	—	FRCSEL	—	—	r(2)	r(2)	APLLPRE<3:0>				0001
APLLFBD1	—	—	—	—	r(2)	r(2)	r(2)	r(2)	APLLFBDIV<7:0>								0096
APLLDIV1	—	—	—	—	—	—	AVCODIV<1:0>		—	APOST1DIV<2:0>			—	APOST2DIV<2:0>			0031
CANCLKCON	CANCLKEN	—	—	—	CANCLKSEL<3:0>				—	CANCLKDIV<6:0>							0001
REFOCONL	ROEN	—	ROSIDL	ROOUT	ROSLP	—	ROSWEN	ROACTIV	—	—	—	—	ROSEL<3:0>				0000
REFOCONH	—	RODIV<14:0>															0000

Legend: — = unimplemented, read as '0'; r = Reserved bit, read as '0'. Reset values are shown in hexadecimal.

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

2: Bits are reserved, maintain as '0'.

Table 16-2: Slave 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	—	CF	—	—	OSWEN	7700(1)
CLKDIV	ROI	DOZE<2:0>			DOZEN	FRCDIV<2:0>			—	—	r(2)	r(2)	PLLPRE<3:0>				3001
PLLFBD	—	—	—	—	r(2)	r(2)	r(2)	r(2)	PLLFBDIV<7:0>								0096
PLLDIV	—	—	—	—	—	—	VCODIV<1:0>		—	POST1DIV<2:0>			—	POST2DIV<2:0>			0031
ACLKCON1	APLLEN	APLLCK	—	—	—	—	—	FRCSEL	—	—	r(2)	r(2)	APLLPRE<3:0>				0001
APLLFBD1	—	—	—	—	r(2)	r(2)	r(2)	r(2)	APLLFBDIV<7:0>								0096
APLLDIV1	—	—	—	—	—	—	AVCODIV<1:0>		—	APOST1DIV<2:0>			—	APOST2DIV<2:0>			0031
REFOCONL	ROEN	—	ROSIDL	ROOUT	ROSLP	—	ROSWEN	ROACTIV	—	—	—	—	ROSEL<3:0>				0000
REFOCONH	—	RODIV<14:0>															0000

Legend: — = unimplemented, read as '0'; r = Reserved bit, read as '0'. Reset values are shown in hexadecimal.

Note 1: The Slave OSCCON register Reset values are dependent on the FS1OSCSEL Configuration bits and by the type of Reset.

2: Bits are reserved, maintain as '0'.

Table 16-3: Master Oscillator Configuration Registers

File Name	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	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>			0xFFFFF
FOSC	—	—	—	—	—	—	—	—	—	—	—	XTBST	XTCFG<1:0>		—	PLLKEN	FCKSM<1:0>		—	—	—	OSCIOFNC	POSCMD<1:0>		0xFFFFF

Legend: — = unimplemented, read as '1'. Reset values are shown in hexadecimal.

Table 16-4: Slave Oscillator Configuration Registers

File Name	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	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
FS1OSCSEL	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	S1IESO	—	—	—	—	S1FNOSC2:0>			0xFFFFF
FS1OSC	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	S1PLLKEN	S1FCKSM<1:0>		—	—	—	S1OSCIOFNC	—	—	0xFFFFF

Legend: — = unimplemented, read as '1'. Reset values are shown in hexadecimal.

17.0 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 dsPIC33/PIC24 product families, but the concepts are pertinent and could be used with modification and possible limitations. The current application notes related to the Oscillator Module with High-Speed PLL 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 dsPIC33/PIC24 device families.

18.0 REVISION HISTORY

Revision A (April 2016)

This is the initial revision of this document.

Revision B (July 2017)

This revision includes the following updates:

- Sections:
  - Updated **External Clock Source Operation (EC Mode)** in **Section 5.0 “Primary Oscillator (POSC)”**, **Section 5.2 “Primary Oscillator Pin Functionality”**, **Section 6.0 “Internal Fast RC (FRC) Oscillator”**, **Section 7.1 “LPRC Oscillator for System Clock”**, **Section 8.0 “Master Phase-Locked Loop (PLL)”**, **Section 8.3 “PLL Setup”**, **Section 8.3.2 “Setup for Using PLL with 8 MHz Internal FRC”**, **Section 9.0 “Slave Phase-Locked Loop (PLL)”**, **Section 9.3.1 “Setup for Using PLL with the Primary Oscillator (POSC)”**, **Section 9.3.2 “Setup for Using PLL with 8 MHz Internal FRC”**, **Section 10.0 “Master Auxiliary Phase-Locked Loop (APLL)”**, **Section 10.1.1 “Setup for Using APLL with the 8 MHz Internal FRC Oscillator”**, **Section 11.0 “Slave Auxiliary Phase-Locked Loop (APLL)”**, **Section 11.1.1 “Setup for Using APLL with the 8 MHz Internal FRC Oscillator”**, **Section 12.0 “Fail-Safe Clock Monitor (FSCM)”**, **Section 13.0 “Clock Switching”** and **Section 14.0 “Two-Speed Start-up”**.
- Figures:
  - Updated Figure 1-2, Figure 1-3, Figure 5-4, Figure 8-1, Figure 9-1, Figure 10-1 and Figure 11-1.
- Registers:
  - Updated Register 3-2, Register 3-3, Register 3-4, Register 4-3, Register 4-5, Register 4-7, Register 4-8, Register 4-9, Register 4-10, Register 4-14, Register 4-15, Register 4-17, Register 4-18 and Register 4-19.
- Tables:
  - Updated Table 3-2, Table 5-1, Table 5-2, Table 6-1, Table 8-1, Table 9-1, Table 16-2, Table 16-3 and Table 16-4.
- Equations:
  - Updated Equation 8-2, Equation 8-4, Equation 9-2, Equation 9-4, Equation 10-2 and Equation 11-2.
- Examples:
  - Updated Example 8-1, Example 8-2, Example 9-1, Example 9-2, Example 10-1 and Example 11-1.

Additional minor updates to text and formatting were incorporated throughout the document.

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
═ ISO/TS 16949 ═

Trademarks

ISBN: 978-1-5224-1905-1

Worldwide Sales and Service

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://www.microchip.com/support
Web Address:
www.microchip.com

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Austin, TX
Tel: 512-257-3370

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Novi, MI
Tel: 248-848-4000

Houston, TX
Tel: 281-894-5983

Indianapolis
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453
Tel: 317-536-2380

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608
Tel: 951-273-7800

Raleigh, NC
Tel: 919-844-7510

New York, NY
Tel: 631-435-6000

San Jose, CA
Tel: 408-735-9110
Tel: 408-436-4270

Canada - Toronto
Tel: 905-695-1980
Fax: 905-695-2078

ASIA/PACIFIC

Asia Pacific Office
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon

Hong Kong
Tel: 852-2943-5100
Fax: 852-2401-3431

Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Tel: 86-10-8569-7000
Fax: 86-10-8528-2104

China - Chengdu
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

China - Chongqing
Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

China - Dongguan
Tel: 86-769-8702-9880

China - Guangzhou
Tel: 86-20-8755-8029

China - Hangzhou
Tel: 86-571-8792-8115
Fax: 86-571-8792-8116

China - Hong Kong SAR
Tel: 852-2943-5100
Fax: 852-2401-3431

China - Nanjing
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

China - Qingdao
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

China - Shanghai
Tel: 86-21-3326-8000
Fax: 86-21-3326-8021

China - Shenyang
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

China - Shenzhen
Tel: 86-755-8864-2200
Fax: 86-755-8203-1760

China - Wuhan
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

China - Xian
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

ASIA/PACIFIC

China - Xiamen
Tel: 86-592-2388138
Fax: 86-592-2388130

China - Zhuhai
Tel: 86-756-3210040
Fax: 86-756-3210049

India - Bangalore
Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

India - New Delhi
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

India - Pune
Tel: 91-20-3019-1500

Japan - Osaka
Tel: 81-6-6152-7160
Fax: 81-6-6152-9310

Japan - Tokyo
Tel: 81-3-6880- 3770
Fax: 81-3-6880-3771

Korea - Daegu
Tel: 82-53-744-4301
Fax: 82-53-744-4302

Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 or
82-2-558-5934

Malaysia - Kuala Lumpur
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

Malaysia - Penang
Tel: 60-4-227-8870
Fax: 60-4-227-4068

Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

Taiwan - Hsin Chu
Tel: 886-3-5778-366
Fax: 886-3-5770-955

Taiwan - Kaohsiung
Tel: 886-7-213-7830

Taiwan - Taipei
Tel: 886-2-2508-8600
Fax: 886-2-2508-0102

Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

Finland - Espoo
Tel: 358-9-4520-820

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

France - Saint Cloud
Tel: 33-1-30-60-70-00

Germany - Garching
Tel: 49-8931-9700

Germany - Haan
Tel: 49-2129-3766400

Germany - Heilbronn
Tel: 49-7131-67-3636

Germany - Karlsruhe
Tel: 49-721-625370

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Germany - Rosenheim
Tel: 49-8031-354-560

Israel - Ra'anana
Tel: 972-9-744-7705

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Italy - Padova
Tel: 39-049-7625286

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Norway - Trondheim
Tel: 47-7289-7561

Poland - Warsaw
Tel: 48-22-3325737

Romania - Bucharest
Tel: 40-21-407-87-50

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

Sweden - Gothenberg
Tel: 46-31-704-60-40

Sweden - Stockholm
Tel: 46-8-5090-4654

UK - Wokingham
Tel: 44-118-921-5800
Fax: 44-118-921-5820