



## Section 7. Oscillator

### HIGHLIGHTS

This section of the manual contains the following topics:

7.1	Introduction .....	7-2
7.2	CPU Clocking.....	7-4
7.3	Oscillator Configuration Registers .....	7-5
7.4	Special Function Registers .....	7-8
7.5	Primary Oscillator (POSC).....	7-18
7.6	Internal Fast RC (FRC) Oscillator .....	7-22
7.7	Phase-Locked Loop (PLL) .....	7-23
7.8	Low-Power Secondary Oscillator (SOSC).....	7-28
7.9	Low-Power RC (LPRC) Oscillator .....	7-29
7.10	Auxiliary Oscillator .....	7-30
7.11	Auxiliary Phase-Locked Loop (APLL) .....	7-31
7.12	Fail-Safe Clock Monitor (FSCM) .....	7-34
7.13	Clock Switching.....	7-35
7.14	Two-Speed Start-up .....	7-39
7.15	Register Maps .....	7-40
7.16	Related Application Notes.....	7-41
7.17	Revision History .....	7-42

# dsPIC33E/PIC24E Family Reference Manual

---

**Note:** This family reference manual section is meant to serve as a complement to device data sheets. Depending on the device variant, this manual section may not apply to all dsPIC33E/PIC24E devices.

Please consult the note at the beginning of the “**Oscillator**” 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>

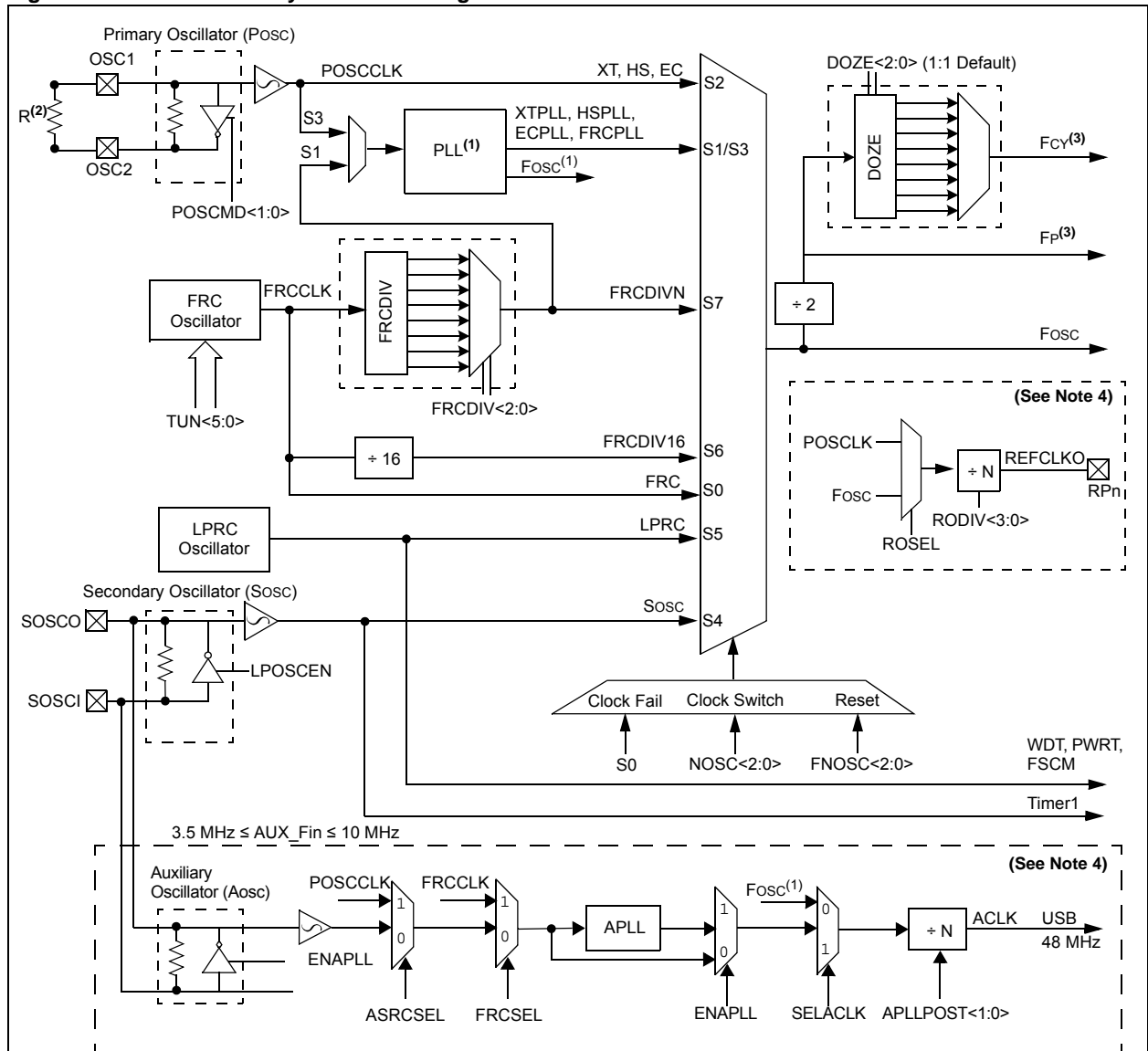
## 7.1 INTRODUCTION

The dsPIC33E/PIC24E oscillator system includes these characteristics:

- Four external and internal oscillator options
- Auxiliary oscillator that provides clock source to the USB module (not available on all devices)
- On-chip Phase-Locked Loop (PLL) to boost internal operating frequency on select internal and external oscillator sources
- On-the-fly clock switching between various clock sources
- Doze mode for system power savings
- Fail-Safe Clock Monitor (FSCM) that detects clock failure and permits safe application recovery or shutdown
- Nonvolatile Configuration bits for clock source selection
- Scalable Reference Clock Out (REFCLKO) (not available on all devices)

A block diagram of the dsPIC33E/PIC24E oscillator system is shown in [Figure 7-1](#).

**Figure 7-1: Oscillator System Block Diagram**



- Note**
- 1: See Figure 7-8 for the source of the Fosc signal.
  - 2: If the oscillator is used with XT or HS modes, an external parallel resistor with the value of 1 MΩ must be connected.
  - 3: The terms Fcy and Fp are used interchangeably throughout this document and are equivalent for the default Doze mode 1:1 ratio.
  - 4: This feature is not available on all devices. Refer to the “Oscillator” chapter in the specific device data sheet for availability.

# dsPIC33E/PIC24E Family Reference Manual

## 7.2 CPU CLOCKING

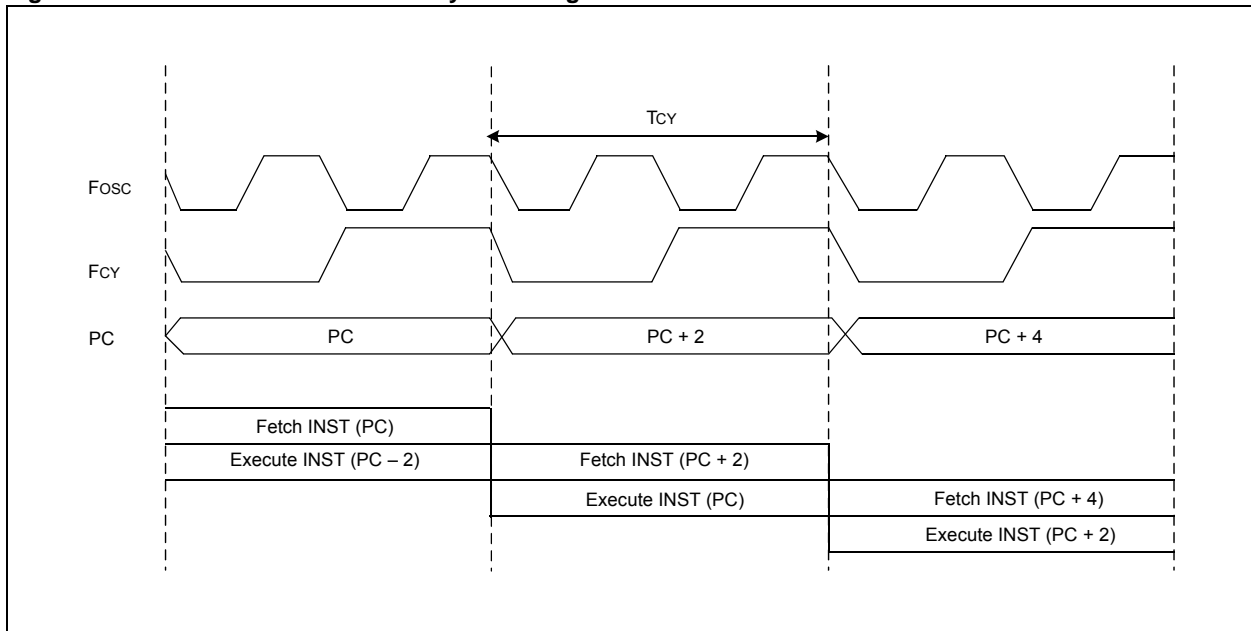
The system clock (Fosc) source can be provided by one of the following options:

- Primary Oscillator (Posc) on the OSC1 and OSC2 pins
- Secondary Oscillator (Sosc) on the SOSCI and SOSCO pins
- Internal Fast RC Oscillator (FRC) with optional clock divider
- Internal Low-Power RC Oscillator (LPRC)
- Primary Oscillator with PLL
- Internal Fast RC Oscillator with PLL

The system clock source is divided by two to produce the internal instruction cycle clock. In this document, the instruction cycle clock is denoted by Fcy. The timing diagram in [Figure 7-2](#) 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 7.5 “Primary Oscillator \(Posc\)”](#).

**Figure 7-2: Clock and Instruction Cycle Timing**



## 7.3 OSCILLATOR CONFIGURATION REGISTERS

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

- **FOSCSEL: Oscillator Source Selection Register**

FOSCSEL selects the initial oscillator source and start-up option. FOSCSEL contains the following Configuration bit:

The FNOSC<2:0> Configuration bits in the 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: Oscillator Configuration Register**

FOSC configures the Primary Oscillator mode, OSCO pin function, peripheral pin select, and the fail-safe and clock switching modes. FOSC contains the following Configuration bits:

- POSCMD (FOSC<1:0>) Configuration bits select the operation mode of the Posc.
- OSCIOFNC (FOSC<2>) Configuration bit selects the OSC2 pin function, except in HS or Medium-Speed Oscillator (XT) mode.

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

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

Table 7-1 lists the configuration settings that select the device oscillator source and operating mode at a POR.

**Table 7-1: Configuration Bit Values for Clock Selection**

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

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

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

# dsPIC33E/PIC24E Family Reference Manual

**Register 7-1: FOSCSEL: Oscillator Source Selection Register**

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

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

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

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

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

1 = Start-up device with Internal FRC, then automatically switch to the user-selected oscillator source when ready

0 = Start-up device with user-selected oscillator source

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

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

111 = Fast RC Oscillator with Divide by N (FRCDIVN)

110 = Fast RC Oscillator with Divide by 16 (FRCDIV16)

101 = Low-Power RC Oscillator (LPRC)

100 = Secondary Oscillator (Sosc)

011 = Primary Oscillator with PLL (XTPLL, HSPLL, ECPLL)

010 = Primary Oscillator (XT, HS, EC)

001 = Fast RC Oscillator with PLL (FRCPLL)

000 = Fast RC Oscillator (FRC)

**Register 7-2: FOSC: Oscillator Configuration Register**

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

R/P	R/P	R/P	U	U	R/P	R/P	R/P
FCKSM<1:0>		IOL1WAY <sup>(1)</sup>	—	—	OSCI0FNC	POSCMD<1:0>	
bit 7							bit 0

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

- bit 15-8 **Reserved:** Reserved bits must be programmed as '1'
- bit 7-6 **FCKSM<1:0>:** Clock Switching Mode bits
  - 1x = Clock switching is disabled, Fail-Safe Clock Monitor is disabled
  - 01 = Clock switching is enabled, Fail-Safe Clock Monitor is disabled
  - 00 = Clock switching is enabled, Fail-Safe Clock Monitor is enabled
- bit 5 **IOL1WAY:** Peripheral Pin Select Configuration bit<sup>(1)</sup>
  - 1 = Allow only one reconfiguration
  - 0 = Allow multiple reconfigurations
- bit 4-3 **Reserved:** Reserved bits must be programmed as '1'
- bit 2 **OSCI0FNC:** OSC2 Pin Function bit (except in XT and HS modes)
  - 1 = OSC2 is the clock output and the instruction cycle (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
  - 11 = Primary oscillator disabled
  - 10 = HS Crystal Oscillator mode (10 MHz to 60 MHz)
  - 01 = XT Crystal Oscillator mode (3.5 MHz to 10 MHz)
  - 00 = EC (External Clock) mode (0 MHz to 60 MHz)

**Note 1:** The IOL1WAY bit is not available on all dsPIC33E/PIC24E devices. For more information, refer to the “**Special Features**” chapter in the specific device data sheet.

## 7.4 SPECIAL FUNCTION REGISTERS

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

- **OSCCON: Oscillator Control Register**

This register controls clock switching and provides status information that allows current clock source, PLL lock and clock fail conditions to be monitored.

- **CLKDIV: Clock Divisor Register**

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

- **PLLFBF: PLL Feedback Divisor Register**

This register selects the PLL feedback divisor.

- **OSCTUN: FRC Oscillator Tuning Register**

This register is used to tune the Internal FRC oscillator frequency in software. It allows the FRC oscillator frequency to be adjusted over a range of  $\pm 12\%$ .

- **ACLKCON3: Auxiliary Clock Control Register**

This register controls and provides prescaler and postscaler values for the Auxiliary PLL module.

- **ACLKDIV3: Auxiliary Clock Divisor Control Register**

This register selects the PLL feedback divisor for the Auxiliary PLL module.



**Register 7-3: OSCCON: Oscillator Control Register**

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

R/S-0	R/W-0	R-0	U-0	R/C-0	U-0	R/W-0	R/W-0
CLKLOCK	IOLOCK <sup>(1)</sup>	LOCK	—	CF	—	LPOSCEN <sup>(2)</sup>	OSWEN
bit 7				bit 0			

<b>Legend:</b>	U = Unimplemented bit, read as '0'	y = Depends on FOSCSEL<FNOSC> bits	
R = Readable bit	W = Writable bit	C = Clearable bit	S = Settable bit
-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 with Divide-by-N (FRCDIVN)
  - 110 = Fast RC Oscillator with Divide-by-16 (FRCDIV16)
  - 101 = Low-Power RC Oscillator (LPRC)
  - 100 = Secondary Oscillator (Sosc)
  - 011 = Primary Oscillator with PLL (XTPLL, HSPLL, ECPLL)
  - 010 = Primary Oscillator (XT, HS, EC)
  - 001 = Fast RC Oscillator with PLL (FRCPLL)
  - 000 = Fast RC Oscillator (FRC)
- bit 11           **Unimplemented:** Read as '0'
- bit 10-8        **NOSC<2:0>:** New Oscillator Selection bits
  - 111 = Fast RC Oscillator with Divide by N (FRCDIVN)
  - 110 = Fast RC Oscillator with Divide by 16 (FRCDIV16)
  - 101 = Low-Power RC Oscillator (LPRC)
  - 100 = Secondary Oscillator (SOSC)
  - 011 = Primary Oscillator with PLL (XTPLL, HSPLL, ECPLL)
  - 010 = Primary Oscillator (XT, HS, EC)
  - 001 = Fast RC Oscillator with PLL (FRCPLL)
  - 000 = Fast RC Oscillator (FRC)
- bit 7           **CLKLOCK:** Clock Lock Enable bit
  - If clock switching is enabled and FSCM is disabled (FCKSM<1:0> (FOSC<7:6>) = 01):
  - 1 = Clock switching is disabled, system clock source is locked
  - 0 = Clock switching is enabled, system clock source may be modified by clock switching
- bit 6           **IOLOCK:** Peripheral Pin Select (PPS) Lock bit<sup>(1)</sup>
  - 1 = Peripheral Pin Select is locked. Writes to Peripheral Pin Select registers are not allowed.
  - 0 = Peripheral Pin Select is not locked. Writes to Peripheral Pin Select registers are allowed.
- bit 5           **LOCK:** PLL Lock Status bit (read-only)
  - 1 = Indicates that PLL is in lock or PLL start-up timer is satisfied
  - 0 = Indicates that PLL is out of lock, start-up timer is in progress or PLL is disabled
- bit 4           **Unimplemented:** Read as '0'

- Note 1:** The IOLOCK bit is not available on all dsPIC33E/PIC24E devices. For more information, refer to “**Oscillator**” chapter in the specific device data sheet.
- 2:** This bit is only reset on a POR.

**Note:** Writes to this register require an unlock sequence. For more information and examples see [7.13 “Clock Switching”](#).

# dsPIC33E/PIC24E Family Reference Manual

---

## Register 7-3: OSCCON: Oscillator Control Register (Continued)

- bit 3      **CF:** Clock Fail Detect bit (read or cleared by application)  
1 = FSCM has detected clock failure  
0 = FSCM has not detected clock failure
- bit 2      **Unimplemented:** Read as '0'
- bit 1      **LPOSCEN:** Secondary Oscillator (Sosc) Enable bit  
1 = Enable secondary oscillator  
0 = Disable secondary oscillator
- bit 0      **OSWEN:** Oscillator Switch Enable bit  
1 = Request oscillator switch to selection specified by the NOSC<2:0> bits  
0 = Oscillator switch is complete

**Note 1:** The IOLOCK bit is not available on all dsPIC33E/PIC24E devices. For more information, refer to “**Oscillator**” chapter in the specific device data sheet.

**2:** This bit is only reset on a POR.

**Note:** Writes to this register require an unlock sequence. For more information and examples see [7.13 “Clock Switching”](#).

**Register 7-4: CLKDIV: Clock Divisor Register**

R/W-0	R/W-0	R/W-1	R/W-1	R/W-0	R/W-0	R/W-0	R/W-1
ROI	DOZE<2:0> <sup>(2,3)</sup>			DOZEN <sup>(1)</sup>	FRCDIV<2:0>		
bit 15				bit 8			

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

**Legend:**

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

- bit 15      **ROI:** Recover on Interrupt bit  
 1 = Interrupts will clear the DOZEN bit and the processor clock, 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<sup>(2,3)</sup>  
 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<sup>(1)</sup>  
 1 = DOZE<2:0> field specifies the ratio between the peripheral clocks and the processor clocks  
 0 = Processor clock and peripheral clock ratio forced to 1:1
- bit 10-8      **FRCDIV<2:0>:** Internal Fast RC Oscillator Postscaler bits  
 111 = FRC divided by 256  
 110 = FRC divided by 64  
 101 = FRC divided by 32  
 100 = FRC divided by 16  
 011 = FRC divided by 8  
 010 = FRC divided by 4  
 001 = FRC divided by 2  
 000 = FRC divided by 1 (default)
- bit 7-6      **PLLPOST<1:0>:** PLL VCO Output Divider Select bits (also denoted as 'N2', PLL postscaler)  
 11 = Output divided by 8  
 10 = Reserved  
 01 = Output divided by 4 (default)  
 00 = Output divided by 2
- bit 5      **Unimplemented:** Read as '0'

- Note 1:** This bit is cleared when the ROI bit is set and an interrupt occurs.
- Note 2:** The DOZE<2:0> bits can only be written when the DOZEN bit is clear. If DOZEN = 1, any writes to DOZE<2:0> are ignored.
- Note 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.

# dsPIC33E/PIC24E Family Reference Manual

---

## Register 7-4: CLKDIV: Clock Divisor Register (Continued)

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

11111 = Input divided by 33

•

•

•

00001 = Input divided by 3

00000 = Input divided by 2 (default)

- Note 1:** This bit is cleared when the ROI bit is set and an interrupt occurs.
- 2:** The DOZE<2:0> bits can only be written when the DOZEN bit is clear. If DOZEN = 1, any writes to DOZE<2:0> are ignored.
- 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.

## Section 7. Oscillator

**Register 7-5: PLLFBD: PLL Feedback Divisor Register**

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

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

**Legend:**

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

bit 15-9      **Unimplemented:** Read as '0'

bit 8-0      **PLLDIV<8:0>:** PLL Feedback Divisor bits (also denoted as 'M', PLL multiplier)

111111111 = 513

•

•

•

000110000 = 50 (default)

•

•

•

000000010 = 4

000000001 = 3

000000000 = 2

# dsPIC33E/PIC24E Family Reference Manual

**Register 7-6: OSCTUN: FRC Oscillator Tuning Register**

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

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	TUN<5:0>					
bit 7						bit 0	

**Legend:**

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

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

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

011111 = Center frequency + 11.625% (8.23 MHz)

011110 = Center frequency + 11.25% (8.20 MHz)

•

•

•

000001 = Center frequency + 0.375% (7.40 MHz)

000000 = Center frequency (7.37 MHz nominal)

111111 = Center frequency – 0.375% (7.345 MHz)

•

•

•

100001 = Center frequency – 11.625% (6.52 MHz)

100000 = Center frequency – 12% (6.49 MHz)

## Section 7. Oscillator

**Register 7-7: REFOCON: Reference Oscillator Control Register**

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ROON	—	ROSSLP	ROSEL	RODIV<3:0> <sup>(1)</sup>			
bit 15				bit 8			
U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 7				bit 0			

**Legend:**

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

- bit 15      **ROON:** Reference Oscillator Output Enable bit  
             1 = Reference oscillator output enabled on REFCLK<sup>(2)</sup> pin  
             0 = Reference oscillator output disabled
- bit 14      **Unimplemented:** Read as '0'
- bit 13      **ROSSLP:** Reference Oscillator Run in Sleep bit  
             1 = Reference oscillator output continues to run in Sleep  
             0 = Reference oscillator output is disabled in Sleep
- bit 12      **ROSEL:** Reference Oscillator Source Select bit  
             1 = Oscillator crystal used as the reference clock  
             0 = System clock used as the reference clock
- bit 11-8    **RODIV<3:0>:** Reference Oscillator Divider bits<sup>(1)</sup>  
             1111 = Reference clock divided by 32,768  
             1110 = Reference clock divided by 16,384  
             1101 = Reference clock divided by 8,192  
             1100 = Reference clock divided by 4,096  
             1011 = Reference clock divided by 2,048  
             1010 = Reference clock divided by 1,024  
             1001 = Reference clock divided by 512  
             1000 = Reference clock divided by 256  
             0111 = Reference clock divided by 128  
             0110 = Reference clock divided by 64  
             0101 = Reference clock divided by 32  
             0100 = Reference clock divided by 16  
             0011 = Reference clock divided by 8  
             0010 = Reference clock divided by 4  
             0001 = Reference clock divided by 2  
             0000 = Reference clock
- bit 7-0     **Unimplemented:** Read as '0'

- Note 1:** The reference oscillator output must be disabled (ROON = 0) before writing to these bits.  
**Note 2:** This pin is remappable. See the “**I/O Ports**” chapter in the specific device data sheet for information.

**Note:** The Reference Clock Out pin and this register is not available on all devices. Refer to the “**Oscillators**” chapter in the specific device data sheet for availability.

# dsPIC33E/PIC24E Family Reference Manual

## Register 7-8: ACLKCON3: Auxiliary Clock Control Register

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0
ENAPLL	—	SELACLK	AOSCMD<1:0>	ASRCSEL	FRCSEL	—	—
bit 15							bit 8
R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0
APLLPOST<2:0>			—	—	APLLPRE<2:0>		
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            **ENAPLL:** Enable Auxiliary PLL (APLL) and Select APLL as USB Clock Source bit  
1 = APLL is enabled, the USB clock source is the APLL output  
0 = APLL is disabled, the USB clock source is the input clock to the APLL
- bit 14            **Unimplemented:** Read as '0'
- bit 13            **SELACLK:** Select Auxiliary Clock Source for Auxiliary Clock Divider bit  
1 = Auxiliary PLL or oscillators provides the source clock for auxiliary clock divider  
0 = Primary PLL provides the source clock for the auxiliary clock divider
- bit 12-11        **AOSCMD<1:0>:** Auxiliary Oscillator Mode bits  
11 = EC (External Clock) Oscillator mode select  
10 = XT (Crystal) Oscillator mode select  
01 = HS (High-Speed) Oscillator mode select  
00 = Auxiliary oscillator disabled (default)
- bit 10            **ASRCSEL:** Select Reference Clock Source for APLL bit  
1 = Primary oscillator is the clock source for APLL  
0 = Auxiliary oscillator is the clock source for APLL
- bit 9             **FRCSEL:** Select FRC as Reference Clock Source for APLL bit  
1 = FRC is clock source for APLL  
0 = Auxiliary oscillator or primary oscillator is the clock source for APLL (determined by ASRCSEL bit)
- bit 8             **Unimplemented:** Read as '0'
- bit 7-5          **APLLPOST<2:0>:** Select PLL VCO Output Divider bits  
111 = Divided by 2  
110 = Divided by 2  
101 = Divided by 4  
100 = Divided by 8  
011 = Divided by 16  
010 = Divided by 32  
001 = Divided by 64  
000 = Divided by 256 (default)
- bit 4-3          **Unimplemented:** Read as '0'
- bit 2-0          **APLLPRE<2:0>:** PLL Phase Detector Input Divider bits  
111 = Divided by 12  
110 = Divided by 10  
101 = Divided by 6  
100 = Divided by 5  
011 = Divided by 4  
010 = Divided by 3  
001 = Divided by 2  
000 = Divided by 1 (default)

**Note:** This register is not available on all devices. Refer to the “Oscillators” chapter in the specific device data sheet for availability.



## Section 7. Oscillator

**Register 7-9: ACLKDIV3: Auxiliary Clock Divisor Control 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	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	APLLDIV<2:0>		
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 15-3 **Unimplemented:** Read as '0'

bit 2-0 **APLLDIV<2:0>:** PLL Feedback Divisor bits (PLL multiplier ratio)

111 = 24

110 = 21

101 = 20

100 = 19

011 = 18

010 = 17

001 = 16

000 = 15 (default)

**Note:** This register is not available on all devices. Refer to the “**Oscillators**” chapter in the specific device data sheet for availability.

# dsPIC33E/PIC24E Family Reference Manual

## 7.5 PRIMARY OSCILLATOR (Posc)

The Primary Oscillator (Posc) is available on the OSC1 and OSC2 pins of the dsPIC33E/PIC24E device family. This connection enables an external crystal (or ceramic resonator) to provide the clock to the device. Optionally, and depending on the device, it can be used with the internal PLL to boost the system frequency ( $F_{OSC}$ ) up to 140 MHz for 70 MIPS execution. 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 25 MHz.

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

If the on-chip oscillator is not used, the EC mode allows the internal oscillator to be bypassed. The device clocks are generated from an external source (0.8 MHz to 60 MHz) and input on the OSC1 pin.

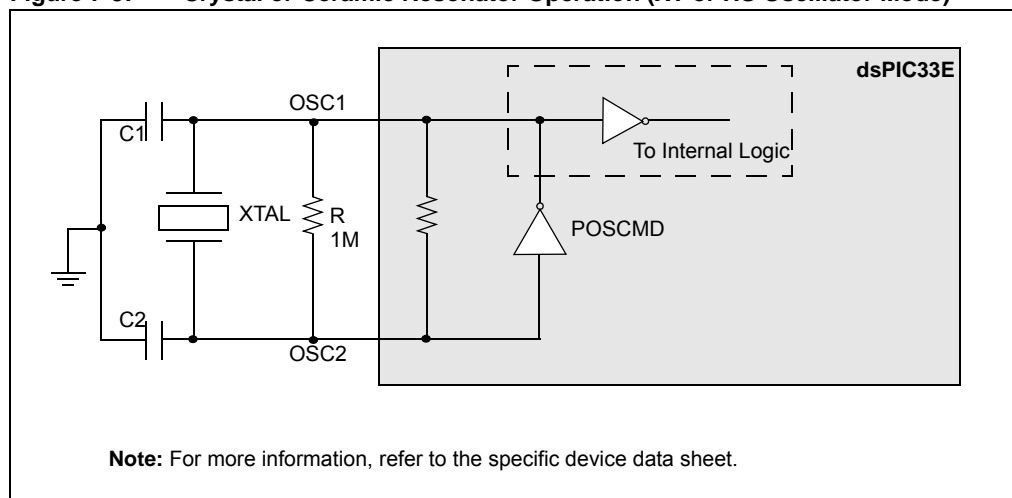
The FNOSC<2:0> Configuration bits in the Oscillator Source Selection register (FOSCSEL<2:0>) specify the primary oscillator clock source at Power-on Reset. The POSCMD<1:0> Configuration bits in the Oscillator Configuration register (FOSC<1:0>) specify the Primary Oscillator mode. [Table 7-2](#) provides the options selected by specific bit configurations, which are programmed at the time of device programming.

**Table 7-2: Primary Oscillator Clock Source Options**

FNOSC Value	POSCMD	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 7-3](#) is a recommended crystal oscillator circuit diagram for the dsPIC33E/PIC24E device. Capacitors, C1 and C2, form the load capacitance for the crystal. The optimum load capacitance ( $C_L$ ) for a given crystal is specified by the crystal manufacturer. Load capacitance can be calculated as shown in [Equation 7-1](#).

**Figure 7-3: Crystal or Ceramic Resonator Operation (XT or HS Oscillator Mode)**



**Equation 7-1: Crystal Load Capacitance**

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

**Note:** Where  $C_S$  is the stray capacitance.

Assuming  $C_1 = C_2$ , [Equation 7-2](#) gives the capacitor value ( $C_1$ ,  $C_2$ ) for a given load and stray capacitance.

**Equation 7-2: External Capacitor for Crystal**

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

For more information on crystal oscillators and their operation, refer to [Section 7.16 “Related Application Notes”](#).

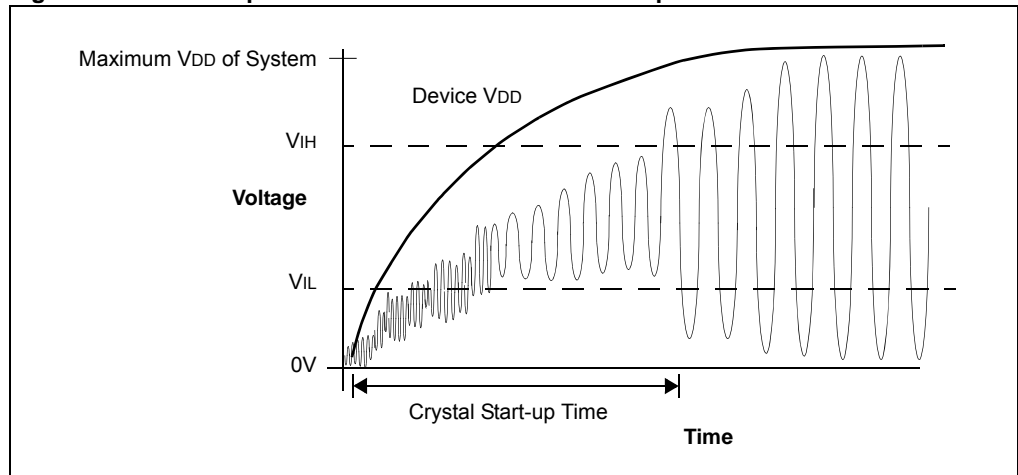
## 7.5.1 Oscillator Start-up Time

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

- Crystal and resonator frequency
- Capacitor values used ( $C_1$  and  $C_2$  in [Figure 7-3](#))
- Device  $V_{DD}$  rise time
- System temperature
- Series resistor value and type if used
- Oscillator mode selection of device (selects the gain of the internal oscillator inverter)
- Crystal quality
- Oscillator circuit layout
- System noise

[Figure 7-4](#) illustrates a plot of a typical oscillator and resonator start-up.

**Figure 7-4: 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) and the Secondary Oscillator (SOSC). The OST is a simple, 10-bit counter that counts 1024 cycles before releasing the oscillator clock to the rest of the system. This time-out period is denoted as  $T_{OST}$ .

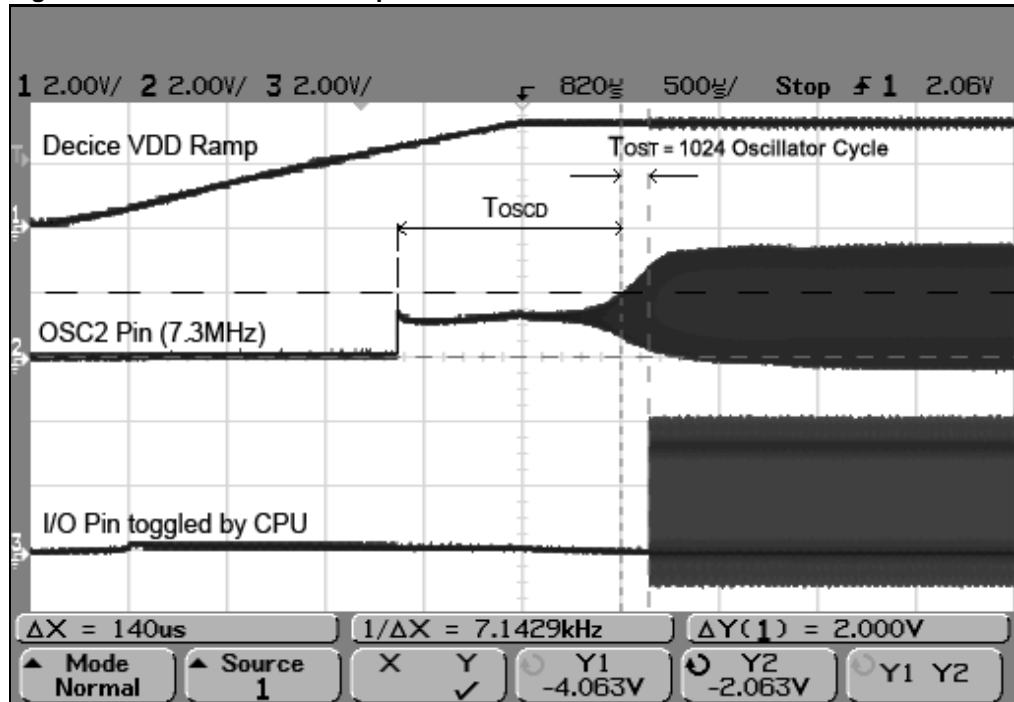
The amplitude of the oscillator signal must reach the  $V_{IL}$  and  $V_{IH}$  thresholds for the oscillator pins before the OST can begin to count cycles. The  $T_{OST}$  interval is required every time the oscillator restarts (that is, on POR, BOR and wake-up from Sleep mode) when XT or HS mode is selected in the Configuration words. The  $T_{OST}$  timer does not exist when EC mode is selected.

# dsPIC33E/PIC24E Family Reference Manual

After the primary oscillator is enabled, it takes a finite amount of time to start oscillating. This delay is denoted as  $T_{OSCD}$ . After  $T_{OSCD}$ , the OST timer takes 1024 clock cycles ( $T_{OST}$ ) to release the clock. The total delay for the clock to be ready is  $T_{OSCD} + T_{OST}$ . If the PLL is used, an additional delay is required for the PLL to lock. For more information, see [Section 7.7 “Phase-Locked Loop \(PLL\)”](#).

Primary oscillator start-up behavior is illustrated in [Figure 7-5](#), where the CPU begins toggling an I/O pin when it starts execution after the  $T_{OSCD} + T_{OST}$  interval.

**Figure 7-5: Oscillator Start-up Characteristics**



## 7.5.2 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: Primary Oscillator Mode Selection bits:**

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

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

- 1 = OSC2 is the clock output, and the instruction cycle (Fcy) clock is output on the OCS2 pin (see [Figure 7-6](#))
- 0 = OSC2 is a general purpose digital I/O pin (see [Figure 7-7](#))

The oscillator pin functions are provided in [Table 7-3](#).

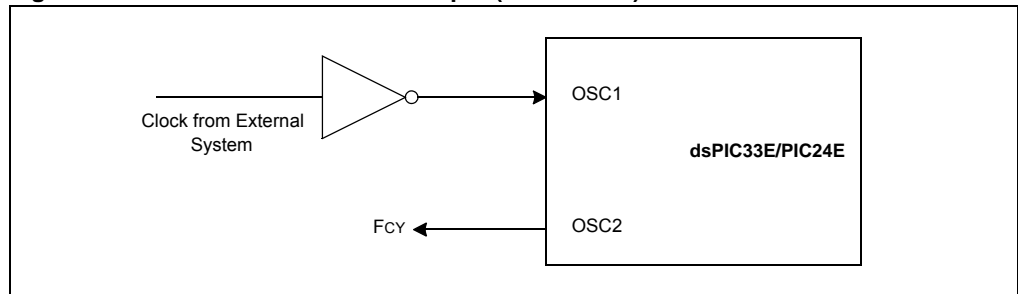
**Table 7-3: Clock Pin Function Selection**

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

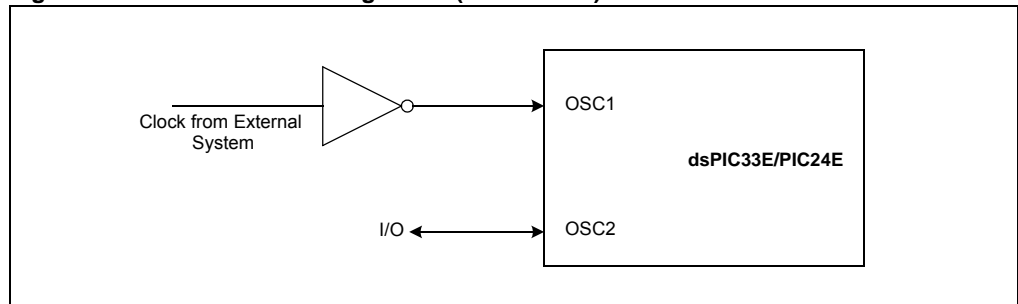
**Note 1:** OSC1 pin function is determined by the Primary Oscillator Mode (POSCMOD<1:0>) Configuration bits.

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

**Figure 7-6: OSC2 Pin for Clock Output (in EC Mode)**



**Figure 7-7: OSC2 Pin for Digital I/O (in EC Mode)**



# dsPIC33E/PIC24E Family Reference Manual

## 7.6 INTERNAL FAST RC (FRC) OSCILLATOR

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

The application software can tune the frequency of the oscillator from -12% to +11.625% (30 kHz steps) of the nominal frequency value using the FRC Oscillator Tuning bits (TUN<5:0>) in the FRC Oscillator Tuning register (OSCTUN<5:0>).

**Note:** Refer to the “Oscillator” 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 Initial Oscillator Source Selection Configuration bits (FNOSC<2:0>) in the Oscillator Source Selection register (FOSCSEL<2:0>) select the FRC clock source. The FRC clock source options at the time of a Power-on Reset are provided in [Table 7-4](#). The Configuration bits are programmed at the time of device programming.

**Table 7-4: FRC Clock Source Options**

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

### 7.6.1 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 Divisor register (CLKDIV<10:8>), which allows 8 settings, from 1:1 to 1:256, to be chosen.

**Table 7-5: 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

Optionally, and depending on the device, the FRC postscaler output can be used with the internal PLL to boost the system frequency (Fosc) up to 140 MHz for 70 MIPS instruction cycle execution speed.

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

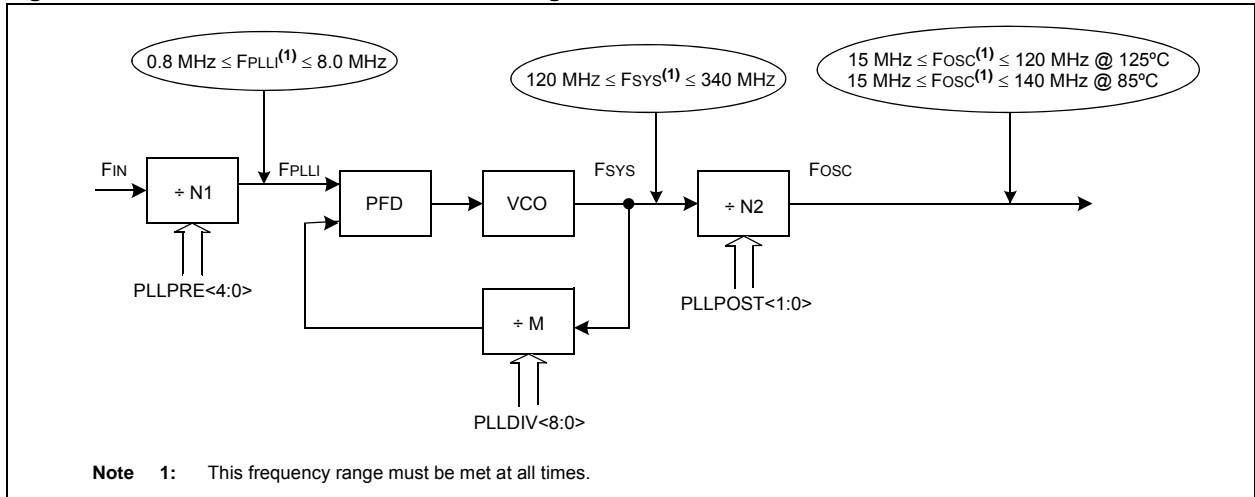
To change the FRC divider:

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

## 7.7 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 7-8 illustrates a block diagram of the PLL module.

Figure 7-8: dsPIC33E/PIC24E PLL Block Diagram



For PLL operation, the Phase Frequency Detector (PFD) input frequency and Voltage Controlled Oscillator (VCO) output frequency must meet the following requirements at all times, no exceptions:

- The PFD input frequency (FPLLI) must be in the range of 0.8 MHz to 8.0 MHz
- The VCO output frequency (FSYS) must be in the range of 120 MHz to 340 MHz

The PLL Phase Detector Input Divider Select bits (PLLPRE<4:0>) in the Clock Divisor register (CLKDIV<4:0>) specify the input divider ratio (N1), which is used to scale down the input clock (FIN) to meet the PFD input frequency range of 0.8 MHz to 8.0 MHz.

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

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

Equation 7-3 provides the relation between input frequency (FIN) and VCO frequency (FSYS).

Equation 7-3: FSYS Calculation

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

Equation 7-4 provides the relation between input frequency (FIN) and output frequency (FOSC).

Equation 7-4: FOSC Calculation

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

Where,

$$N1 = PLLPRE + 2$$

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

$$M = PLLDIV + 2$$

# dsPIC33E/PIC24E Family Reference Manual

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

Table 7-6 provides the default values of the PLL Prescaler, PLL Postscaler and PLL Feedback Divisor Configuration bits at Power-on Reset.

**Table 7-6: PLL Mode Defaults**

Register	Bit Field	Value at POR Reset	PLL Divider Ratio
CLKDIV<4:0>	PLLPRE<4:0>	00000	N1 = 2
CLKDIV<7:6>	PLLPOST<1:0>	01	N2 = 4
PLLFBD<8:0>	PLLDIV<8:0>	000110000	M = 50

Given these Reset values, the following equations provide the relationship between input frequency ( $F_{IN}$ ) and PFD input frequency ( $F_{PLLI}$ ), VCO frequency ( $F_{SYS}$ ) and system clock frequency ( $F_{OSC}$ ) at Power-on Reset.

**Equation 7-5: FPLLI at Power-on Reset**

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

**Equation 7-6: FSYS at Power-on Reset**

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

**Equation 7-7: FOSC at Power-on Reset**

$$F_{OSC} = F_{IN} \left( \frac{M}{N1 \cdot N2} \right) = F_{IN} \left( \frac{50}{2 \cdot 4} \right) = 6.25(F_{IN})$$

To use the PLL and to ensure that the PFD input frequency ( $F_{PLLI}$ ) and the VCO frequency are in the specified frequency range to meet the PLL requirements follow these process:

1. Power-up the device with the internal FRC or the primary oscillator without PLL.
2. Change the PLLDIV, PLLPRE and PLLPOST bit values, based on the input frequency, to meet these PLL requirements:
  - The PFD input frequency ( $F_{PLLI}$ ) must be in the range of 0.8 MHz to 8.0 MHz
  - The VCO output frequency ( $F_{SYS}$ ) must be in the range of 120 MHz to 340 MHz
3. Switch the clock to the PLL mode in software.

**Note:** Due to the default PLL register setting on Reset, it would violate the OSC specification to power-up with primary OSC with PLL enabled for input clock frequencies greater than 13.6 MHz. In that case the user would need to power-up in a non-PLL mode, configure the PLL registers and then perform a clock switch to a 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.



### 7.7.2 PLL Lock Status

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

TLOCK is applied when 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 7.5.1 “Oscillator Start-up Time”](#). Also, refer to the “**Oscillator**” 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 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 Prescaler (PLLPRE) and PLL Feedback Divisor (PLLDIV) 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.

#### 7.7.2.1 SETUP FOR USING PLL WITH 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 that the required system clock frequency is:
 
$$F_{OSC} = 2 \times F_{CY} = 120 \text{ MHz}$$
2. To set up the PLL and meet the requirements of the PLL, follow these steps:
  - a) Select the PLL postscaler to meet the VCO output frequency requirement ( $120 \text{ MHz} < F_{SYS} < 340 \text{ MHz}$ ).
    - Select a PLL postscaler ratio of  $N2 = 2$
    - Ensure that  $F_{SYS} = (F_{OSC} \times N2) = 240 \text{ MHz}$
  - b) Select the PLL prescaler to meet the PFD input frequency requirement ( $0.8 \text{ MHz} < F_{PLLI} < 8.0 \text{ MHz}$ ).
    - Select a PLL prescaler ratio of  $N1 = 2$
    - Ensure that  $F_{PLLI} = (F_{IN} \div N1) = 5 \text{ MHz}$
  - c) Select the PLL feedback divisor to generate the required VCO output frequency based on the PFD input frequency.
    - $F_{SYS} = F_{PLLI} \times M$
    - $M = F_{SYS} \div F_{PLLI} = 48$
  - d) Configure the FNOSC<2:0> bits (FOSCSEL<2:0>) to select a clock source without the PLL (for example, Internal FRC) at Power-on Reset.
  - e) In the main program, change the PLL prescaler, PLL postscaler and PLL feedback divisor values to those just decided in the previous steps, and then perform a clock switch to the PLL mode.

[Example 7-1](#) illustrates code for using the PLL with the primary oscillator. (See also [Section 7.13 “Clock Switching”](#) for example code for clock switching.)

# dsPIC33E/PIC24E Family Reference Manual

## Example 7-1: Code Example for Using PLL with Primary Oscillator (Posc)

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

// Enable Clock Switching and Configure POSC in XT mode

_FOSC(FCKSM_CSECMD & OSCIOFNC_OFF & POSCMD_XT);
int main()
{
// Configure PLL prescaler, PLL postscaler, PLL divisor
PLLFBD=46;                // M=48
CLKDIVbits.PLLPOST=0;    // N2=2
CLKDIVbits.PLLPRE=0;    // N1=2

// 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.COSC != 0b011);

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

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

The following process is used to set up the PLL to operate the device at 60 MIPS with a 7.37 MHz Internal FRC.

1. To execute instruction at 60 MHz, ensure that the system clock frequency is:  
 $F_{OSC} = 2 \times F_{CY} = 120 \text{ MHz}$
2. To set up the PLL and meet the requirements of the PLL, follow these steps:
  - a) Select the PLL postscaler to meet VCO output frequency requirement ( $120 \text{ MHz} < F_{SYS} < 340 \text{ MHz}$ ).
    - Select a PLL postscaler ratio of  $N_2 = 2$
    - Ensure that  $F_{SYS} = (F_{OSC} \times N_2) = 240 \text{ MHz}$
  - b) Select the PLL prescaler to meet PFD input frequency requirement ( $0.8 \text{ MHz} < F_{PLLI} < 8.0 \text{ MHz}$ ).
    - Select a PLL prescaler ratio of  $N_1 = 2$
    - Ensure that  $F_{PLLI} = (F_{IN} \div N_1) = 3.68 \text{ MHz}$
  - c) Select the PLL feedback divisor to generate required VCO output frequency based on the PFD input frequency.
    - $F_{SYS} = F_{PLLI} \times M$
    - $M = F_{SYS} \div F_{PLLI} = 65$
  - d) Configure the FNOSC<2:0> bits (FOSCSEL<2:0>) to select a clock source without PLL (for example, Internal FRC) at Power-on Reset.
  - e) In the main program, change the PLL prescaler, PLL postscaler and PLL feedback divisor to meet the user and PLL requirement, and then perform clock switch to the PLL mode.

Example 7-2 illustrates code for using PLL with a 7.37 MHz Internal FRC. (See also [Section 7.13 “Clock Switching”](#) for example code for clock switching.)

### Example 7-2: Code Example for Using PLL with 7.37 MHz Internal FRC

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

// Enable Clock Switching and Configure Primary Oscillator in XT mode
_FOSC(FCKSM_CSECMD & OSCIOFNC_OFF & POSCMD_NONE);

int main()
{
// Configure PLL prescaler, PLL postscaler, PLL divisor
PLLFBFBD=63; // M=65
CLKDIVbits.PLLPOST=0; // N2=2
CLKDIVbits.PLLPRE=1; // N1=3

// Initiate Clock Switch to FRC oscillator with PLL (NOSC=0b001)
__builtin_write_OSCCONH(0x01);
__builtin_write_OSCCONL(OSCCON | 0x01);
// Wait for Clock switch to occur
while (OSCCONbits.COSC!= 0b001);

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

## 7.8 LOW-POWER SECONDARY OSCILLATOR (Sosc)

The Low-Power Secondary Oscillator (SOSC) enables a 32.768 kHz watch crystal to be attached to the dsPIC33E/PIC24E device as a secondary crystal clock source for low-power operation. It uses the SOSCI and SOSCO pins. The low-power secondary oscillator can also drive Timer1 for Real-Time Clock (RTC) applications.

### 7.8.1 Secondary Oscillator for System Clock

The low-power secondary oscillator is enabled as the system clock when:

- The Initial Oscillator Source Selection Configuration bits (FNOSC<2:0>) in the Oscillator Source Selection register (FOSCSEL<2:0>) are appropriately set to select the secondary oscillator at a Power-on Reset
- The user-assigned software initiates a clock switch to the secondary oscillator for low-power operation.

If the low-power secondary oscillator is not used to provide the system clock, or if the device enters Sleep mode, it is disabled to save power.

### 7.8.2 Secondary Oscillator Start-up Delay

When the low-power secondary oscillator is enabled, it takes a finite amount of time to start oscillating. Refer to [Section 7.5.1 “Oscillator Start-up Time”](#) for details.

### 7.8.3 Continuous Secondary Oscillator Operation

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

There are two reasons to leave the low-power secondary oscillator running:

- Keeping the oscillator on at all times allows a fast switch to the 32 kHz system clock for lower power operation. Returning to the faster main oscillator still requires an oscillator start-up time if it is a crystal type source. For more information, see [Section 7.5.1 “Oscillator Start-up Time”](#).
- The oscillator should remain on at all times when Timer1 is being used as a Real-Time Clock.

<p><b>Note:</b> In Sleep mode, all clock sources (Primary Oscillator, Internal FRC and LPRC Oscillator) are shut down, with the exception of the low-power secondary oscillator. The low-power secondary oscillator can be active in Sleep mode if the Secondary Oscillator Enable bit (LPOSCEN) is set in the Oscillator Control register (OSCCON&lt;1&gt;).</p>
---

### 7.9 LOW-POWER RC (LPRC) OSCILLATOR

The Low-Power RC (LPRC) oscillator provides a nominal clock frequency of 32 kHz. The LPRC is the clock source for the Power-up Timer (PWRT), Watchdog Timer (WDT) and Fail-safe Clock Monitor (FSCM) circuits. It can also be used to provide a low-frequency clock source option for the device in those applications where power consumption is critical and timing accuracy is not required.

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

#### 7.9.1 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 Oscillator Source Selection register (FOSCSEL<2:0>) are appropriately set to select the LPRC oscillator at Power-on Reset
- User-assigned software initiates a clock switch to the LPRC oscillator for low-power operation

#### 7.9.2 Enabling the LPRC Oscillator

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

The LPRC oscillator remains enabled under these conditions:

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

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

**Note:** The LPRC is enabled and running automatically if either 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.

#### 7.9.3 LPRC Oscillator Start-up Delay

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

# dsPIC33E/PIC24E Family Reference Manual

## 7.10 AUXILIARY OSCILLATOR

**Note:** This feature is not available on all devices. Refer to the “Oscillator” chapter in the specific device data sheet for availability.

The Auxiliary Oscillator (ACLK) is used by the Universal Serial Bus (USB) module, which needs to operate at a frequency unrelated to the system clock. The auxiliary oscillator can use one of the following as its clock source:

- Crystal (XT): Crystal and ceramic resonators in the range of 3.5 MHz to 10 MHz
- High-Speed Crystal (HS): Crystals in the range of 10 MHz to 40 MHz. The external crystal is connected to the SOSCI and SOSCO pins
- External Clock (EC): External clock signal up to 60 MHz. The external clock signal is directly applied to the SOSCI pin

### 7.10.1 Enabling the Auxiliary Oscillator

To enable the Auxiliary Oscillator mode, the Enable Auxiliary PLL bit (ENAPLL) must be set in the Auxiliary Clock control register (ACLKCONx<15>). The Auxiliary Oscillator Mode bits (AOSCMD<1:0>) allow four oscillator mode settings, as listed in Table 7-7.

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

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

**Note:** By default, the USB module is clocked by the primary oscillator with PLL.

### 7.10.2 Auxiliary Clock Source

The desired reference clock source for the auxiliary PLL can be selected by setting the appropriate clock source select bits in the Auxiliary Clock Control Register 1 (ACLKCONx).

Set the Auxiliary Reference Clock Select bit (ASRCSEL) to use the primary oscillator as the clock source or clear this bit to use the auxiliary oscillator as the clock source.

Set the FRC Select bit (FRCSEL) to use the FRC as the clock source, or clear this bit to use the auxiliary or primary oscillator selected by the ASRCSEL bit as the clock source.

Set the Select Clock Source to Auxiliary Clock Divider bit (SELACLK) in the Auxiliary Clock Control Register 1 (ACLKCONx) to select the auxiliary PLL or oscillators to provide the clock source for the auxiliary clock divider.

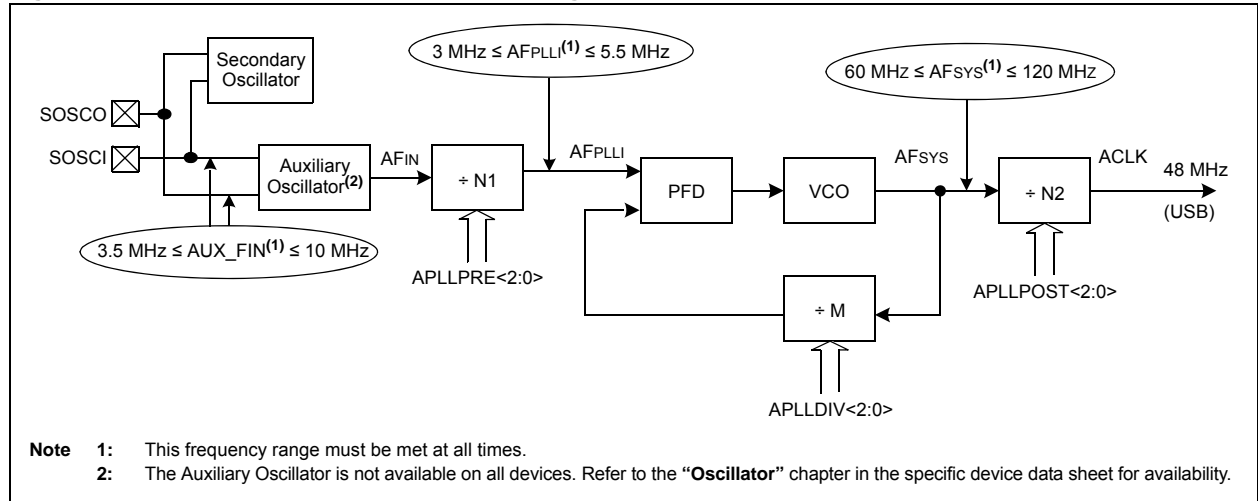
Clearing the SELACLK bit will cause the primary PLL output to act as the clock source to the auxiliary clock divider.

## 7.11 AUXILIARY PHASE-LOCKED LOOP (APLL)

**Note:** This feature is not available on all devices. Refer to the “Oscillator” chapter in the specific device data sheet for availability.

The auxiliary oscillator uses an on-chip PLL to obtain different auxiliary clock speeds. Figure 7-9 shows a block diagram of the APLL module.

Figure 7-9: dsPIC33E/PIC24E APLL Block Diagram



For operation of the APLL, the Auxiliary Phase Frequency Detector (APFD) input frequency and the Auxiliary Voltage Controlled Oscillator (AVCO) output frequency must meet the following requirements:

- The APFD input frequency (AFPLLI) must be in the range of 3 MHz to 5.5 MHz
- The AVCO output frequency (AFSYS) must be in the range of 60 MHz to 120 MHz

The APLL Phase Detector Input Divider bits (APLLPRE<2:0>) in the Auxiliary Clock Control Register 1 (ACLKCONx<2:0>) specify the input divider ratio (N1), which is used to scale down the Auxiliary PLL Input (AFIN) clock to meet the APFD input frequency range of 3 MHz to 5.5 MHz.

The Auxiliary PLL Feedback Divisor bits (APLLDIV<2:0>) in the Auxiliary Clock Control Register 2 (ACLKDOVx<2:0>) specify the divider ratio (M), which scales down the AvCO frequency (AFSYS) for feedback to the APFD. The AvCO frequency (AFSYS) is M times the APFD input frequency (AFPLLI).

The APLL VCO Output Divider Select bits (APLLPOST<2:0>) in the Auxiliary Clock Control Register 1 (ACLKCONx<7:5>) specify the divider ratio (N2).

The correct combination of the APLL Phase Detector Input Divider bits (APLLPRE<2:0>), the Auxiliary PLL Feedback Divisor bits (APLLDIV<2:0>), and the APLL VCO Output Divider bits (APLLPOST<2:0>) will provide the 48 MHz Auxiliary Clock (ACLK) frequency needed by the USB module.

Equation 7-8 shows the relationship between the Auxiliary PLL Input (AFIN) clock frequency and the Auxiliary Clock (ACLK) frequency.

Equation 7-8: ACLK Calculation

$$ACLK = AF_{IN} \times \left( \frac{M}{N1 \times N2} \right)$$

$$= AF_{IN} \times \frac{(APLLDIV + 15)}{(APLLPRE + 1)(APLLPOST + 1)}$$

Where,  
 $N1 = APLLPRE + 1$   
 $N2 = APLLPOST + 1$   
 $M = APLLDIV + 15$

**Note:** When APLLDIV<2:0> = 111, substitute (APLLDIV + 15) with (APLLDIV + 18) in Equation 7-8.

# dsPIC33E/PIC24E Family Reference Manual

Equation 7-9 shows the relationship between the Auxiliary PLL Input (AFIN) clock frequency and the AVCO Frequency (AFSYS).

**Equation 7-9: AFSys Calculation**

$$\begin{aligned} AFSYS &= AFIN \times \left(\frac{M}{N}\right) \\ &= AFIN \times \left(\frac{(PLLDIV + 15)}{(APLLPRE + 1)}\right) \end{aligned}$$

## 7.11.1 APLL Setup

**Note:** This feature is not available on all devices. Refer to the “Oscillator” chapter in the specific device data sheet for availability.

### 7.11.1.1 SETUP FOR USING APLL WITH AUXILIARY OSCILLATOR WITH AN 8 MHz CRYSTAL

1. Clear the ASRCSEL bit to choose the auxiliary oscillator as the clock source for the APLL.
2. Clear the FRCSEL bit to choose the auxiliary oscillator at the clock source for the APLL.
3. Set the SELACLK bit to choose the auxiliary PLL or oscillators to provide the source clock for the auxiliary clock divider.
4. Follow these steps to configure the APLL Phase Detector Input Divider bits (APLLPRE<2:0>), the Auxiliary PLL Feedback Divisor bits (APLLDIV<2:0>) and the APLL VCO Output Divider bits (APLLPOST<2:0>) to set up the APLL for a 48 MHz ACLK (used by the USB module) using an 8 MHz auxiliary oscillator:
  - a) Select the APLL VCO output divider to meet the AVCO output frequency requirement (60 MHz < AFSYS < 120 MHz).
    - Select an APLL VCO output divider ratio of N2 = 2
    - Ensure that AFSYS = (ACLK x N2) = 96 MHz
  - b) Select the APLL phase detector input divider to meet the APFD input frequency requirement (3 MHz < AFPLLI < 5.5 MHz).
    - Select an APLL phase detector input divider ratio of N1 = 2
    - Ensure that AFPLLI = (AFIN ÷ N1) = 4 MHz
  - c) Select the auxiliary PLL feedback divisor to generate the required VCO output frequency based on the PFD input frequency.
    - AFSYS = AFPLLI x M
    - M = AFSYS ÷ AFPLLI = 24
5. Enable the auxiliary PLL by setting the ENAPLL bit.

Example 7-3 provides code for using the APLL with the auxiliary oscillator.

**Example 7-3: Code Example for Using the APLL with the Auxiliary Oscillator**

```
// Configure APLL prescaler, APLL postscaler, APLL divisor
ACLKCON3bits.ASRCSEL = 0; // Select Auxiliary Oscillator as the clock source
ACLKCON3bits.FRCSEL = 0; // Select Auxiliary Oscillator as the clock source
ACLKCON3bits.SELACLK = 1; // Select Auxiliary PLL or oscillators to provide
                          // the source clock for auxiliary clock divider

ACLKDIV3bits.APLLDIV = 0b111; // M = 24
ACLKCON3bits.APLLPRE = 0b001; // N1 = 2
ACLKCON3bits.APLLPOST = 0b111; // N2 = 2
ACLKCON3bits.ENAPLL = 1; // Enable Auxiliary Clock
```



### 7.11.1.2 SETUP FOR USING APLL WITH PRIMARY OSCILLATOR USING AN 8 MHz CRYSTAL

1. Set the ASRCSEL bit to choose the primary oscillator as the clock source for the APLL.
2. Clear the FRCSEL bit to choose the primary oscillator as the clock source for the APLL.
3. Set the SELACLK bit to choose the auxiliary PLL or oscillators to provide the clock source for the auxiliary clock divider.
4. Follow these steps to configure the APLL Phase Detector Input Divider bits (APLLPRE<2:0>), the Auxiliary PLL Feedback Divisor bits (APLLDIV<2:0>) and the APLL VCO Output Divider bits (APLLPOST<2:0>) to set up the APLL for a 48 MHz ACLK (used by the USB module) using an 8 MHz auxiliary oscillator:
  - a) Select the APLL VCO output divider to meet the AVCO output frequency requirement ( $60 \text{ MHz} < \text{AF}_{\text{SYS}} < 120 \text{ MHz}$ ).
    - Select an APLL VCO output divider ratio of  $N2 = 2$
    - Ensure that  $\text{AF}_{\text{SYS}} = (\text{ACLK} \times N2) = 96 \text{ MHz}$
  - b) Select the APLL phase detector input divider to meet the APFD input frequency requirement ( $3 \text{ MHz} < \text{AF}_{\text{PLLI}} < 5.5 \text{ MHz}$ ).
    - Select an APLL phase detector input divider ratio of  $N1 = 2$
    - Ensure that  $\text{AF}_{\text{SYS}} = (\text{AF}_{\text{IN}} \div N1) = 4 \text{ MHz}$
  - c) Select the auxiliary PLL feedback divisor to generate the required VCO output frequency based on the PFD input frequency.
    - $\text{AF}_{\text{SYS}} = \text{AF}_{\text{PLLI}} \times M$
    - $M = \text{AF}_{\text{SYS}} \div \text{AF}_{\text{PLLI}} = 24$
5. Enable the auxiliary PLL by setting the ENAPLL bit.

[Example 7-4](#) provides code for using the APLL with the primary oscillator.

#### Example 7-4: Code Example for Using the APLL with the Primary Oscillator

```
// Configure APLL prescaler, APLL postscaler, APLL divisor
CLKCON3bits.ASRCSEL = 1; // Select Primary Oscillator as the clock source
CLKCON3bits.FRCSEL = 0; // Select Primary Oscillator as the clock source
CLKCON3bits.SELACLK = 1; // Select Auxiliary PLL or oscillators to provide
                          // the source clock for auxiliary clock divider

CLKDIV3bits.APLLDIV = 0b111; // M = 24
CLKCON3bits.APLLPRE = 0b001; // N1 = 2
CLKCON3bits.APLLPOST = 0b111; // N2 = 2
CLKCON3bits.ENAPLL = 1; // Enable Auxiliary Clock
```

## 7.12 FAIL-SAFE CLOCK MONITOR (FSCM)

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

The FSCM monitors the system clock. If it does not detect a system clock within a specific period of time (typically 2 ms, maximum 4 ms), it generates a clock failure trap and switches the system clock to the FRC oscillator. The user-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 FRC oscillator:

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

### 7.12.1 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 or secondary oscillator is selected as the system clock.

For more information, refer to **Section 8. "Reset"** (DS70602). For the recent documentation, visit the Microchip web site at [www.microchip.com](http://www.microchip.com).

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

### 7.12.2 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.

## 7.13 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 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.

Certain dsPIC33E/PIC24E devices feature the Phase-Locked Loop Enable (PLLKEN) bit in the Fuse Configuration register (FWDTC<5>). Setting this bit will cause the device to wait until the PLL locks before switching to the PLL clock source. When this bit is set to '0', the device will not wait for the PLL lock and will proceed with the clock switch. The default setting for this bit is '1'. Refer to **Section 30. "Device Configuration"** (DS70618) for more information.

With few limitations, applications are free to switch between any of the four clock sources (POSC, SOSC, FRC and LPRC) under software control at any time. To limit the possible side effects that could result from this flexibility, dsPIC33E/PIC24E devices have a safeguard lock built into the switch process. That is, the OSCCON register is write-protected during clock switching.

### 7.13.1 Enabling Clock Switching

The Clock Switching Mode Configuration bits (FCKSM<1:0>) in the Oscillator Configuration register (FOSC<7:6>) must be programmed to enable clock switching and the Fail-Safe Clock Monitor (see Table 7-8).

**Table 7-8: Configurable Clock Switching Modes**

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

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

### 7.13.2 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.

# dsPIC33E/PIC24E Family Reference Manual

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 (the Posc or Sosc) 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> bit (OSCCON<10:8>) values are 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) or Sosc (if SOSSEN remains set). The timing of the transition between clock sources is illustrated in Figure 7-10.

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

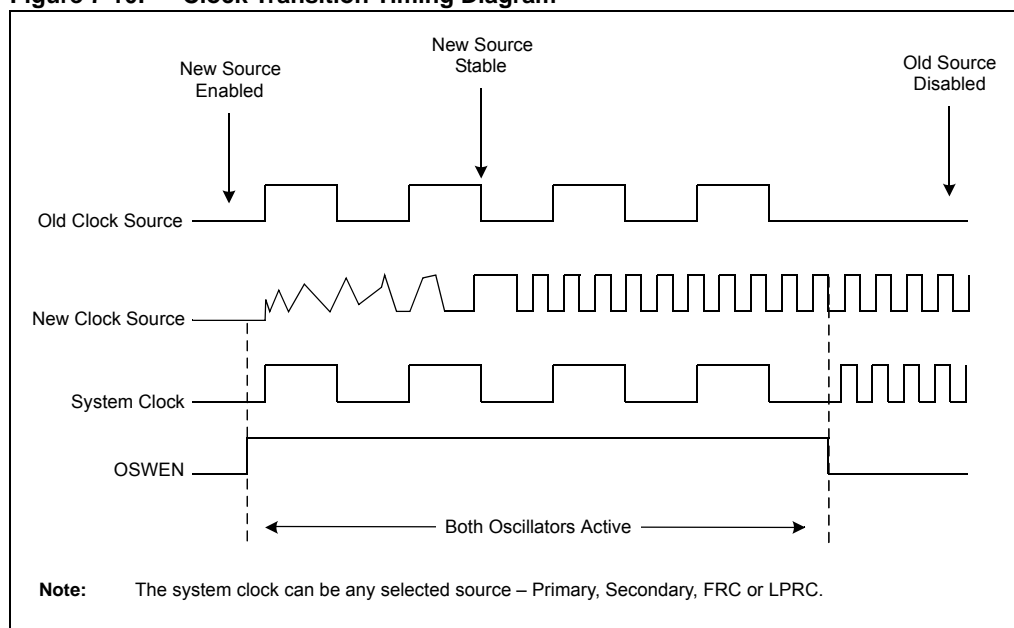
**2:** Direct clock switching between PLL modes is not possible. For example, clock switching should not occur between the primary oscillator with PLL and the internal FRC oscillator with PLL.

**3:** Setting the CLKLOCK bit (OSCCON<7>) prevents clock switching when clock switching is enabled and Fail-Safe Clock Monitoring is disabled by Configuration bits, FCKSM<1:0> (FOSC<7:6>) = 01. The CLKLOCK bit (OSCCON<7>) cannot be cleared after it is set by the software; it clears on Power-on Reset.

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

**5:** The clock switch will not wait for the PLL lock if the PLLKEN bit in the FWDT Fuse Configuration (FWDT<5>) register is set to '0'.

**Figure 7-10: Clock Transition Timing Diagram**



The following are recommended code sequences 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 (OSCCON<0>) is '0'. If it is, the switch was successful.

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

```
__builtin_write_OSCCONH(value)
__builtin_write_OSCCONL(OSCCON | value)
```

For more information, see MPLAB IDE Help.

**Example 7-5** illustrates the code sequence for unlocking the OSCCON register and switching from FRC with PLL clock to the LPRC clock source.

**Example 7-5: 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
```

## 7.13.3 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 good to create the routine in assembler and link it to the application, and then calling it as a function when it is required.
- If the destination clock source is a crystal oscillator, the clock switch time will be dominated by the oscillator start-up time.
- If the new clock source does not start or 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 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, like secondary oscillator, will result in slow device operation.

## 7.13.4 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 7-6](#). 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 7-6: 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
```

## 7.13.5 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 `PWRSVAV` 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.

### 7.14 TWO-SPEED START-UP

The Internal External Start-up Option Configuration bit (IESO) in the Oscillator Source Selection register (FOSCSEL<7>) specifies whether to start the device with a user-selected oscillator source or to initially start with the Internal FRC, 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, regardless of the other oscillator source settings (FOSCSEL<2:0>). Then, the device switches to the specified oscillator when it is ready.

Unless FSCM is enabled, the FRC oscillator is turned off immediately after the clock switch is completed. The Two-Speed Start-up option is a faster way to get the device up and running, and works independently of the state of the Clock Switching Mode Configuration bits, FCKSM<1:0> (FOSC<7:6>).

Two-Speed Start-up is useful when an external oscillator is selected by the FNOSC<2:0> Configuration bits (FOSCSEL<2:0>) and a crystal-based oscillator (either a primary or secondary oscillator) has a longer start-up time. As an internal RC oscillator, the FRC clock source is available immediately following Power-on Reset. With Two-Speed Start-up, the device starts executing code in its default oscillator configuration (FRC). It continues to operate in this mode until the specified external oscillator source becomes stable, at which time, it switches to that source.

User code can check which clock source is currently providing the device clocking by checking the status of the 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 is 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.

## 7.15 REGISTER MAPS

Table 7-9 maps the bit functions for the Oscillator Special Function Control registers. Table 7-10 maps the bit functions for the Oscillator Configuration registers.

**Table 7-9: Oscillator Special Function Control Registers**

File Name	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
OSCCON	—	COSC<2:0>			—	NOSC<2:0>			CLKLOCK	IOLOCK <sup>(2)</sup>	LOCK	—	CF	—	LPOSCEN	OSWEN	7700 <sup>(1)</sup>
CLKDIV	ROI	DOZE<2:0>			DOZEN	FRCDIV<2:0>			PLLPOST<1:0>	—	PLLPRE<4::0>					3140	
PLLFBD	—	—	—	—	—	—	—	PLLDIV<8:0>								0000	
OSCTUN	—	—	—	—	—	—	—	—	—	TUN<5:0>						0000	
ACLKCON3	ENAPLL	—	SELACLK	AOSCMD<1:0>		ASRCSEL	FRCSEL	—	APLLPOST<2:0>			—	—	APLLPRE<2:0>		0000	
ACLKDIV3	—	—	—	—	—	—	—	—	—	—	—	—	—	APLLDIV<2:0>		0000	

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

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

**Note 2:** The IOLOCK bit is not available on all dsPIC33E/PIC24E devices. For more information, refer to “Oscillator” chapter in the specific device data sheet.

**Table 7-10: Oscillator Configuration Registers**

File Name	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
FOSCSEL	—	—	—	—	—	—	—	—	IESO	—	—	—	—	FNOSC<2:0>		
FOSC	—	—	—	—	—	—	—	—	FCKSM<1:0>		IOL1WAY <sup>(1)</sup>	—	—	OSCIOFNC	POSCMD<1:0>	

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

**Note 1:** The IOL1WAY bit is not available on all dsPIC33E/PIC24E devices. For more information, refer to the “Special Features” chapter in the specific device data sheet.



### 7.16 RELATED APPLICATION NOTES

This section lists application notes that pertain to this section of the manual. These application notes may not be written specifically for the dsPIC33E/PIC24E Product Family, but the concepts are pertinent and could be used with modification and possible limitations. The current application notes related to the Oscillator module include:

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

**Note:** Visit the Microchip Web site ([www.microchip.com](http://www.microchip.com)) for additional Application Notes and code examples for the dsPIC33E/PIC24E family of devices.

## 7.17 REVISION HISTORY

### Revision A (January 2009)

This is the initial release of this document.

### Revision B (July 2010)

This revision includes the following updates:

- Removed the watermark from all code examples
- Updated all code examples ([Example 7-1](#) through [Example 7-5](#)) with the exception of [Example 7-6](#)
- Added Note 2 and updated the Oscillator System Block Diagram (see [Figure 7-1](#))
- Added Note 2 to the Oscillator Control Register (see [Register 7-3](#))
- Renamed the Auxiliary Clock Control Register 1 to ACLKCONx (see [Register 7-8](#))
- Renamed the Auxiliary Clock Control Register 2 to ACLKDIVx (see [Register 7-9](#))
- Updated the Crystal or Ceramic Resonator Operation (XT or HS Oscillator Mode) diagram (see [Figure 7-3](#))
- Changed the PFD input frequency (FREF) end range value from 3.5 MHz to 5.5 MHz in step 2 of [7.7.1 “Input Clock Limitation at Start-up for PLL Mode”](#)
- Updated the second paragraph in [7.7.2 “PLL Lock Status”](#)
- Updated step 3 of the second clock switching sequence in [7.13.2 “Clock Switch Sequence”](#)
- Added shaded note after the ACLK Calculation (see [Equation 7-8](#))
- Additional minor corrections such as language and formatting updates are incorporated throughout the document

### Revision C (March 2012)

This revision includes the following updates:

- Changes were made to frequency designators throughout the document. For example, FVCO was changed to FSYS and FREF was changed to FPLLI, among others.
- Updated the Oscillator System Block Diagram (see [Figure 7-1](#))
- Added Notes 2, 3, and 4 to the Clock Divisor Register (see [Register 7-4](#))
- Revised bit 14 in the Auxiliary Clock Control Register 1 to Unimplemented (see [Register 7-8](#) and [Table 7-9](#))
- Updated the system frequency and MIPS ratings in the first paragraph of [7.5 “Primary Oscillator \(Posc\)”](#) and the second paragraph of [7.6.1 “FRC Postscaler Mode \(FRCDIVN\)”](#)
- Updated the dsPIC33E/PIC24E PLL Block Diagram (see [Figure 7-8](#))

---

---

**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.

---

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights.

**Trademarks**

The Microchip name and logo, the Microchip logo, dsPIC, KEELOQ, KEELOQ logo, MPLAB, PIC, PICmicro, PICSTART, PIC<sup>32</sup> logo, rPIC and UNI/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.


FilterLab, Hampshire, HI-TECH C, Linear Active Thermistor, MXDEV, MXLAB, SEEVAL and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, Application Maestro, chipKIT, chipKIT logo, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, dsSPEAK, ECAN, ECONOMONITOR, FanSense, HI-TIDE, In-Circuit Serial Programming, ICSP, Mindi, MiWi, MPASM, MPLAB Certified logo, MPLIB, MPLINK, mTouch, Omniscent Code Generation, PICC, PICC-18, PICDEM, PICDEM.net, PICkit, PICtail, REAL ICE, rLAB, Select Mode, Total Endurance, TSHARC, UniWinDriver, WiperLock and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

All other trademarks mentioned herein are property of their respective companies.

© 2009-2012, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

 Printed on recycled paper.

ISBN: 978-1-62076-152-6

**QUALITY MANAGEMENT SYSTEM  
CERTIFIED BY DNV  
= ISO/TS 16949 =**

*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<sup>®</sup> MCUs and dsPIC<sup>®</sup> DSCs, KEELOQ<sup>®</sup> 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.*



# MICROCHIP

## 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](http://www.microchip.com)

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

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

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

**Cleveland**  
Independence, OH  
Tel: 216-447-0464  
Fax: 216-447-0643

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

**Detroit**  
Farmington Hills, MI  
Tel: 248-538-2250  
Fax: 248-538-2260

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

**Los Angeles**  
Mission Viejo, CA  
Tel: 949-462-9523  
Fax: 949-462-9608

**Santa Clara**  
Santa Clara, CA  
Tel: 408-961-6444  
Fax: 408-961-6445

**Toronto**  
Mississauga, Ontario,  
Canada  
Tel: 905-673-0699  
Fax: 905-673-6509

### ASIA/PACIFIC

**Asia Pacific Office**  
Suites 3707-14, 37th Floor  
Tower 6, The Gateway  
Harbour City, Kowloon  
Hong Kong  
Tel: 852-2401-1200  
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 - Hangzhou**  
Tel: 86-571-2819-3187  
Fax: 86-571-2819-3189

**China - Hong Kong SAR**  
Tel: 852-2401-1200  
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: 86-21-5407-5533  
Fax: 86-21-5407-5066

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

**China - Shenzhen**  
Tel: 86-755-8203-2660  
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

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

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

### ASIA/PACIFIC

**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: 91-20-2566-1512  
Fax: 91-20-2566-1513

**Japan - Osaka**  
Tel: 81-66-152-7160  
Fax: 81-66-152-9310

**Japan - Yokohama**  
Tel: 81-45-471-6166  
Fax: 81-45-471-6122

**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: 886-7-536-4818  
Fax: 886-7-330-9305

**Taiwan - Taipei**  
Tel: 886-2-2500-6610  
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

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

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

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

**Netherlands - Druenen**  
Tel: 31-416-690399  
Fax: 31-416-690340

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

**UK - Wokingham**  
Tel: 44-118-921-5869  
Fax: 44-118-921-5820

11/29/11