

## セクション 7. オシレータ

### ハイライト

本セクションには下記の主要項目を記載しています。

7.1	はじめに .....	7-2
7.2	CPU クロック .....	7-3
7.3	オシレータ コンフィグレーション レジスタ .....	7-4
7.4	特殊機能レジスタ .....	7-7
7.5	プライマリ オシレータ (Posc).....	7-14
7.6	内部高速 RC (FRC) オシレータ .....	7-18
7.7	位相ロックループ (PLL) .....	7-20
7.8	セカンダリ オシレータ (Sosc).....	7-25
7.9	低消費電力 RC (LPRC) オシレータ .....	7-26
7.10	フェイルセーフ クロックモニタ (FSCM).....	7-27
7.11	クロック切り換え .....	7-28
7.12	2 段階起動 .....	7-32
7.13	レジスタマップ .....	7-33
7.14	関連アプリケーション ノート .....	7-34
7.15	改訂履歴 .....	7-35

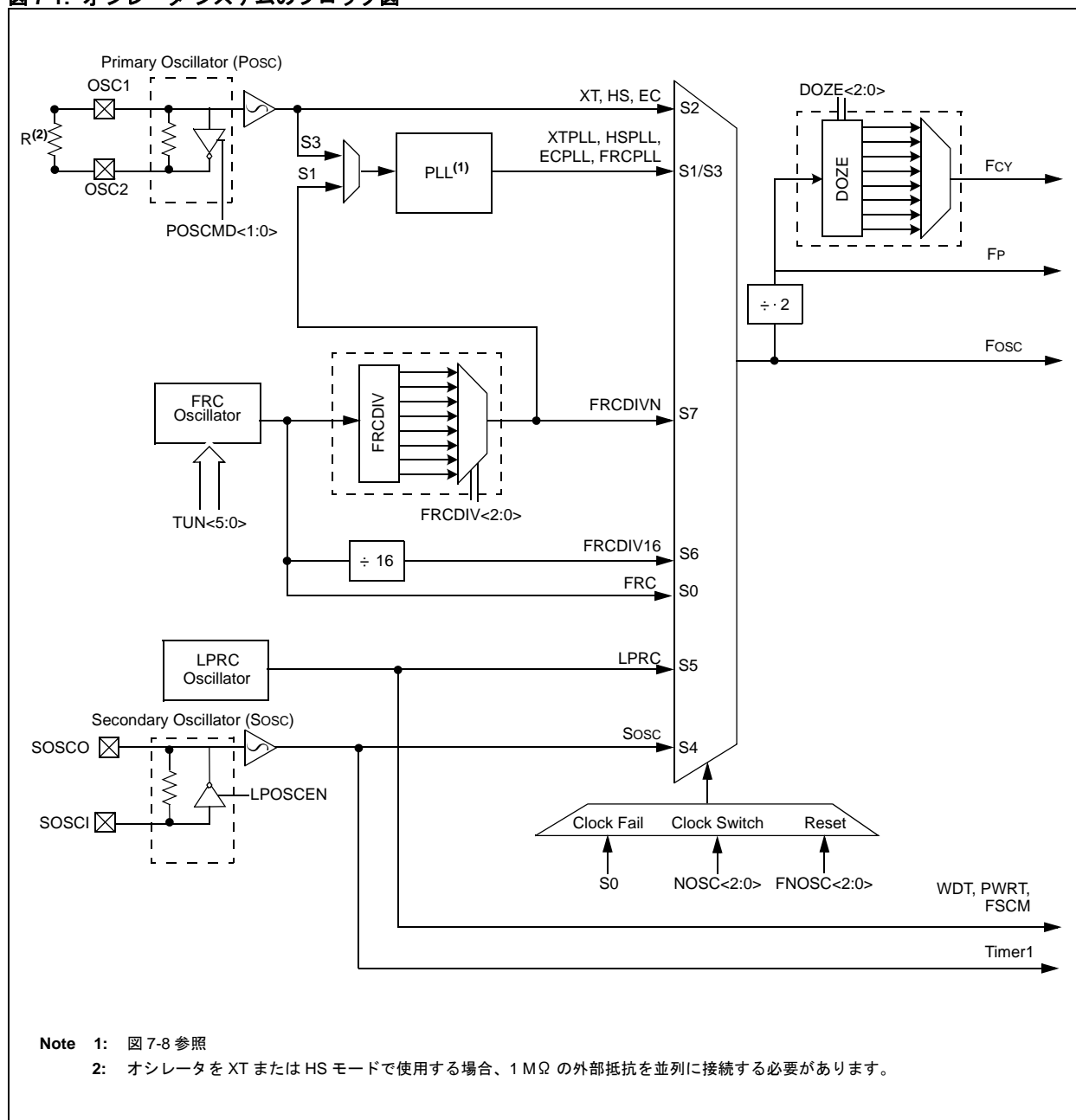
## 7.1 はじめに

dsPIC33F のオシレータ システムは下記の特徴を備えます。

- 4 つの外部および内部オシレータ オプション
- 選択した内部および外部発振源に内蔵位相ロックループ (PLL) を併用して内部動作周波数を通倍
- 動作中に各種クロック源を切り換え可能
- Doze モードによる省電力化
- クロック障害を検出し、アプリケーションを安全に回復またはシャットダウンするフェイルセーフ クロックモニタ (FSCM)
- クロック源選択用の不揮発性コンフィグレーション ビット

dsPIC33F オシレータ システムのブロック図を図 7-1 に示します。

図 7-1: オシレータ システムのブロック図



## 7.2 CPU クロック

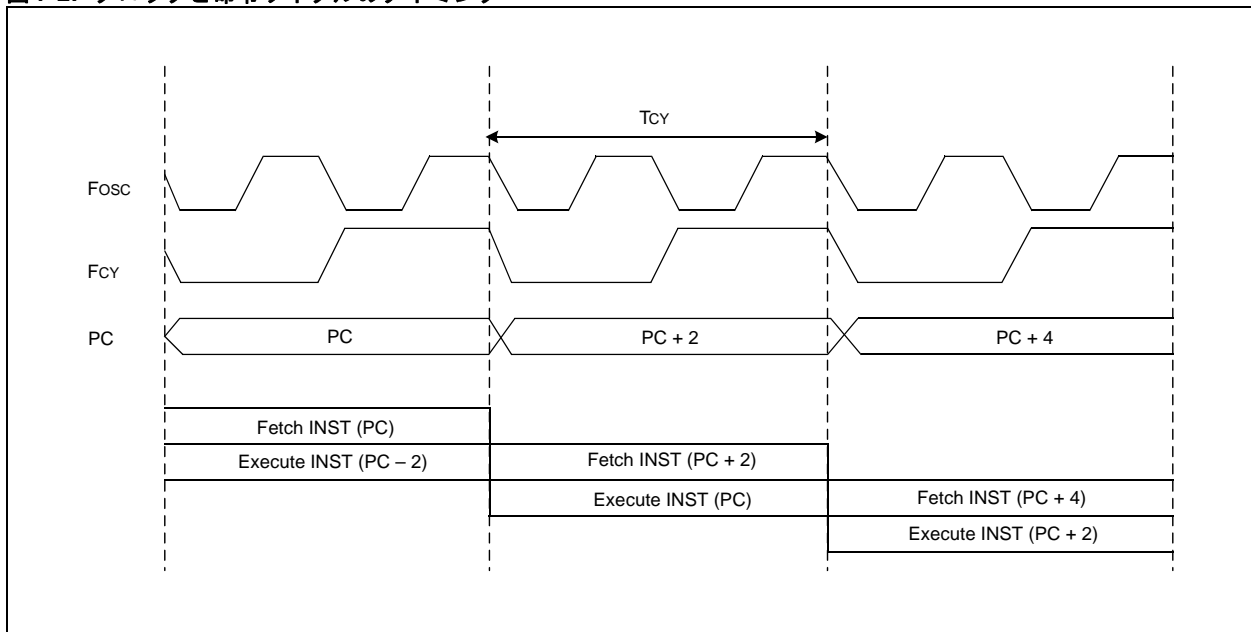
システムクロック (Fosc) は下記の方法で供給できます。

- プライマリ オシレータ (Posc): OSC1 および OSC2 ピンを使用
- セカンダリ オシレータ (Sosc): SOSCI および SOSCO ピンを使用
- 内部高速 RC (FRC) オシレータ (クロック分周比を選択可能)
- 低消費電力の内部 RC (LPRC) オシレータ
- PLL 併用 Posc
- PLL 併用内部 FRC オシレータ

源クロック Fosc を 2 分周する事により、内部命令サイクルクロックを生成します。本書では、命令サイクルクロックを Fcy と表記します。図 7-2 に Fosc、命令サイクルクロック (Fcy)、プログラム カウンタ (PC) のタイミングを示します。

クロック源としてプライマリ オシレータモードまたは高速オシレータ (HS) モードを選択しない場合には、Fcy を OSC2 I/O ピンへ出力できます (セクション 7.5 「プライマリ オシレータ (Posc)」参照)。

図 7-2: クロックと命令サイクルのタイミング



## 7.3 オシレータ コンフィグレーション レジスタ

オシレータ コンフィグレーション レジスタは特殊機能レジスタ (SFR) ではありません。これら 2 つのレジスタは共にプログラムメモリ領域内に配置され、デバイス プログラミング時に設定します。

### • FOSCSEL: クロック源選択レジスタ

FOSCSELは内部クロック源と起動オプションを選択します。FOSCSELは下記のコンフィグレーション ビットを格納します。

クロック源選択 (FOSCSEL<2:0>) レジスタ内の FNOSC<2:0> コンフィグレーション ビットは、POR ( パワーオン リセット ) 時に使用するクロック源を選択します。POR 後に、クロック切り換え操作により他のクロック源へ切り換える事ができます。

既定値 ( デバイスの初期状態 ) では、ポストスケーラ (FRCDIVN) を使用する内部 FRC オシレータを選択します。

### • FOSC: オシレータ コンフィグレーション レジスタ

FOSC はプライマリ オシレータモード、OSCO ピン機能、周辺モジュール用ピン選択モード、フェイルセーフおよびクロック切り換えモードを設定します。FOSC は下記のコンフィグレーション ビットを格納します。

- POSCMD (FOSC<1:0>) コンフィグレーション ビットは Posc の動作モードを選択します。
- OSCIOFNC (FOSC<2>) コンフィグレーション ビットは、HS および中速オシレータ (XT) モード以外での OSC2 ピンの機能を選択します。  
OSCIOFNC が「1」 ( 既定値 ) の場合、OSC2 ピンは Fcy クロックを出力します。  
OSCIOFNC を「0」に変更した場合、OSC2 ピンは汎用 I/O ピンとして機能します。

デバイスクロック源と POR 時の動作モードを選択するコンフィグレーション ビットの設定を表 7-1 に示します。

表 7-1: クロック選択コンフィグレーション ビットの設定

クロック源	オシレータモード	FNOSC	POSCMD	Note
S0	高速 RC オシレータ (FRC)	000	xx	1
S1	PLL 併用高速 RC オシレータ (FRCPLL)	001	xx	1
S2	プライマリ オシレータ (EC)	010	00	1
S2	プライマリ オシレータ (XT)	010	01	
S2	プライマリ オシレータ (HS)	010	10	
S3	PLL 併用プライマリ オシレータ (ECPLL)	011	00	1
S3	PLL 併用プライマリ オシレータ (XTPLL)	011	01	
S3	PLL 併用プライマリ オシレータ (HSPLL)	011	10	
S4	セカンダリ オシレータ (Sosc)	100	xx	1
S5	低消費電力 RC (LPRC) オシレータ	101	xx	1
S6	16 分周高速オシレータ (FRCDIV16)	110	xx	1
S7	N 分周高速オシレータ (FRCDIVN)	111	xx	1, 2

**Note 1:** OSC2 ピンの機能は OSCIOFNC コンフィグレーション ビットで選択します。

**2:** 既定値オシレータモード ( デバイス初期状態またはプログラム消去後の状態 )

レジスタ 7-1: FOSCSEL: クロック源選択レジスタ

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

**凡例:**

R = 読み出し可能ビット      P = 書き込み可能ビット      U = 未使用ビット、「1」にセット  
 -n = POR 時の値      1 = ビットをセット      0 = ビットをクリア      x = ビットは未知

bit 15-8 **未実装:** 「1」 として読み出し

bit 7 **IESO:** 内部 / 外部起動オプションビット

1 = 内部 FRC オシレータを使用してデバイスを起動し、安定化後にユーザ選択クロック源へ切り換え  
 0 = ユーザ選択クロック源を使用してデバイスを起動

bit 6-3 **未実装:** 「1」 として読み出し

bit 2-0 **FNOSC<2:0>:** 初期クロック源選択ビット

111 = N 分周高速 RC オシレータ (FRCDIVN)  
 110 = 16 分周高速 RC オシレータ (FRCDIV16)  
 101 = 低消費電力 RC (LPRC) オシレータ  
 100 = セカンダリ オシレータ (SOSC)  
 011 = PLL 併用プライマリ オシレータ (XTPLL,、HSPLL、ECPLL)  
 010 = プライマリ オシレータ (XT、HS、EC)  
 001 = PLL 併用高速 RC オシレータ (FRCPLL)  
 000 = 高速 RC オシレータ (FRC)

# dsPIC33F ファミリ リファレンス マニュアル

レジスタ 7-2: FOSC: オシレータ コンフィグレーション レジスタ

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>	—	—	—	OSCIOFNC	POSCMD<1:0>	—
bit 7						bit 0	

## 凡例:

R = 読み出し可能ビット	P = 書き込み可能ビット	U = 未使用ビット、「1」にセット
-n = POR 時の値	1 = ビットをセット	0 = ビットをクリア      x = ビットは未知

bit 15-8 **未実装**: 「1」として読み出し

bit 7-6 **FCKSM<1:0>**: クロック切り換えモードビット

1x = クロック切り換え無効 / フェイルセーフ クロックモニタ (FSCM) 無効  
 01 = クロック切り換え有効 / FSCM 無効  
 00 = クロック切り換え有効 / FSCM 有効

bit 5 **IOL1WAY**: 周辺モジュール用ピン選択モード コンフィグレーション ビット<sup>(1)</sup>

1 = 1 回の設定変更のみ許可  
 0 = 複数回の設定変更を許可

bit 4-3 **未実装**: 「1」として読み出し

bit 2 **OSCIOFNC**: OSC2 ピン機能ビット (XT および HS モード以外)

1 = OSC2 ピンをクロック出力として使用し、命令サイクル (Fcy) クロックを出力  
 0 = OSC2 ピンを汎用デジタル I/O ピンとして使用

bit 1-0 **POSCMD<1:0>**: プライマリ オシレータのモード選択ビット

11 = プライマリ オシレータは無効  
 10 = HS 水晶振動子モード  
 01 = XT 水晶振動子モード  
 00 = EC ( 外部クロック ) モード

**Note 1:** 一部の dsPIC33F デバイスでは IOL1WAY ビットを利用できません。詳細は各デバイスのデータシートを参照してください。

## 7.4 特殊機能レジスタ

下記の特種機能レジスタ (SFR) は、オシレータ システムのランタイム制御とステータス情報を提供します。

- **OSCCON: オシレータ制御レジスタ<sup>(2)</sup>**

OSCCON はクロック切り換えを制御し、使用中のクロック源、PLL ロック、クロック障害条件を監視するためのステータス情報を提供します。

- **CLKDIV: クロック分周比レジスタ**

CLKDIV は Doze モードを制御し、PLL プリスケラ、PLL ポストスケラ、FRC ポストスケラを選択します。

- **PLLFBF: PLL フィードバック分周比レジスタ**

PLLFBF は PLL フィードバック分周比を選択します。

- **OSCTUN: FRC オシレータ調整レジスタ**

OSCTUN は、ソフトウェアによる内部 FRC オシレータ周波数の調整に使用します。これにより、内部 FRC オシレータ周波数を ± 12% の範囲で調整できます。

**Note:** オシレータ SFR (OSCCON、CLKDIV、PLLFBF、OSCTUN) は POR 時にのみリセットされます。

# dsPIC33F ファミリ リファレンス マニュアル

レジスタ 7-3: OSCCON: オシレータ制御レジスタ (2)

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	OSWEN
bit 7				bit 0			

<b>凡例:</b>	U = 未実装ビット、「0」として読み出し y = FOSCSEL<FNOSC> ビットに依存						
R = 読み出し可能ビット	W = 書き込み可能ビット	C = クリア可能ビット	S = セット可能ビット				
-n = POR 時の値	1 = ビットをセット	0 = ビットをクリア	x = ビットは未知				

bit 15 **未実装:** 「0」として読み出し

bit 14-12 **COSC<2:0>:** 切り換え前クロック源選択ビット (読み出し専用)

111 = N 分周高速 RC オシレータ (FRCDIVN)  
 110 = 16 分周高速 RC オシレータ (FRCDIV16)  
 101 = 低消費電力 RC (LPRC) オシレータ  
 100 = セカンダリ オシレータ (Sosc)  
 011 = PLL 併用プライマリ オシレータ (XTPLL, HSPLL, ECPLL)  
 010 = プライマリ オシレータ (XT, HS, EC)  
 001 = PLL 併用高速 RC オシレータ (FRCPLL)  
 000 = 高速 RC オシレータ (FRC)

bit 11 **未実装:** 「0」として読み出し

bit 10-8 **NOSC<2:0>:** 切り換え後クロック源選択ビット

111 = N 分周高速 RC オシレータ (FRCDIVN)  
 110 = 16 分周高速 RC オシレータ (FRCDIV16)  
 101 = 低消費電力 RC (LPRC) オシレータ  
 100 = セカンダリ オシレータ (Sosc)  
 011 = PLL 併用プライマリ オシレータ (XTPLL, HSPLL, ECPLL)  
 010 = プライマリ オシレータ (XT, HS, EC)  
 001 = PLL 併用高速 RC オシレータ (FRCPLL)  
 000 = 高速 RC オシレータ (FRC)

bit 7 **CLKLOCK:** クロックロック イネーブルビット

クロック切り換え有効 / FSCM 無効の場合: FCKSM<1:0> (FOSC<7:6>) = 01  
 1 = クロック切り換えは無効 (システム クロック源を固定)  
 0 = クロック切り換えは有効 (クロック切り換えによるシステム クロック源の変更が可能)

bit 6 **IOLOCK:** 周辺モジュール用ピン選択モード (PPS) ロックビット <sup>(1)</sup>

1 = 周辺モジュール用ピン選択モードをロック (周辺モジュール用ピン選択モードレジスタへの書き込みを禁止)  
 0 = 周辺モジュール用ピン選択モードのロックを解除 (周辺モジュール用ピン選択モードレジスタへの書き込みを許可)

bit 5 **LOCK:** PLL ロック ステータスビット (読み出し専用)

1 = PLL はロック中、または PLL 起動タイマ条件が成立  
 0 = PLL はロック解除中、または起動タイマが作動中、または PLL は無効

bit 4 **未実装:** 「0」として読み出し

**Note 1:** 一部の dsPIC33F デバイスでは IOLOCK ビットを利用できません。詳細は各デバイスのデータシートを参照してください。

**2:** このレジスタへ書き込むには、シーケンスのロックを解除する必要があります。詳細と例は 7.11 「クロック切り換え」を参照してください。



### レジスタ 7-3: OSCCON: オシレータ制御レジスタ (2)

bit 3	<b>CF:</b> クロック障害検出ビット (アプリケーションによる読み出しまたはクリア) 1 = FSCM がクロック障害を検出した 0 = FSCM はクロック障害を検出していない
bit 2	<b>未実装:</b> 「0」として読み出し
bit 1	<b>LPOSCEN:</b> セカンダリ オシレータ イネーブルビット 1 = セカンダリ オシレータは有効 0 = セカンダリ オシレータは無効
bit 0	<b>OSWEN:</b> クロック源切り換えイネーブルビット 1 = NOSC<2:0> ビットが指定するクロック源への切り換えを要求 0 = クロック源の切り換えは完了

- Note 1:** 一部の dsPIC33F デバイスでは IOLOCK ビットを利用できません。詳細は各デバイスのデータシートを参照してください。
- 2:** このレジスタへ書き込むには、シーケンスのロックを解除する必要があります。詳細と例は 7.11 「クロック切り換え」を参照してください。

# dsPIC33F ファミリ リファレンス マニュアル

レジスタ 7-4: CLKDIV: クロック分周比レジスタ

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

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

## 凡例:

R = 読み出し可能ビット      W = 書き込み可能ビット      U = 未実装ビット、「0」として読み出し  
 -n = POR 時の値      1 = ビットをセット      0 = ビットをクリア      x = ビットは未知

- bit 15      **ROI:** 割り込み回復ビット  
 1 = 割り込みは DOZEN ビットとプロセッサ クロックをクリアし、周辺モジュール用クロック分周比を 1:1 へ設定  
 0 = 割り込みは DOZEN ビットに影響しない
- bit 14-12      **DOZE<2:0>:** プロセッサ クロック分周比選択ビット <sup>(2)</sup>  
 111 = Fcy を 128 分周  
 110 = Fcy を 64 分周  
 101 = Fcy を 32 分周  
 100 = Fcy を 16 分周  
 011 = Fcy を 8 分周 (既定値)  
 010 = Fcy を 4 分周  
 001 = Fcy を 2 分周  
 000 = Fcy を分周しない
- bit 11      **DOZEN:** Doze モード イネーブルビット <sup>(1,2)</sup>  
 1 = DOZE<2:0> フィールドにより周辺モジュール用クロックとプロセッサ クロック間の比を指定  
 0 = 周辺モジュール用クロックとプロセッサ クロック間の比を 1:1 に固定
- bit 10-8      **FRCDIV<2:0>:** 内部高速 RC オシレータのポストスケーラ ビット  
 111 = FRC を 256 分周  
 110 = FRC を 64 分周  
 101 = FRC を 32 分周  
 100 = FRC を 16 分周  
 011 = FRC を 8 分周  
 010 = FRC を 4 分周  
 001 = FRC を 2 分周  
 000 = FRC を分周しない (既定値)
- bit 7-6      **PLLPOST<1:0>:** PLL VCO 出力分周比選択ビット (「N2」、PLL ポストスケーラとも呼ぶ)  
 00 = 出力を 2 分周  
 01 = 出力を 4 分周 (既定値)  
 10 = 予約  
 11 = 出力を 8 分周
- bit 5      **未実装:** 「0」として読み出し

- Note 1:** ROI ビットをセットした状態で割り込みが発生すると、このビットはクリアされます。  
**2:** Doze モードの詳細はセクション 9. 「ウォッチドッグ タイマと省電力モード」 (DS70196) を参照してください。

### レジスタ 7-4: CLKDIV: クロック分周比レジスタ ( 続き )

bit 4-0      **PLLPRE<4:0>:** PLL 位相検出器入力分周比選択ビット ( 「N1」、PLL プリスケアラとも呼ぶ )  
11111 = 入力を 33 分周  
•  
•  
•  
00001 = 入力を 3 分周  
00000 = 入力を 2 分周 ( 既定値 )

**Note 1:** ROI ビットをセットした状態で割り込みが発生すると、このビットはクリアされます。

**2:** Doze モードの詳細はセクション 9. 「ウォッチドッグ タイマと省電力モード」 (DS70196) を参照してください。

# dsPIC33F ファミリ リファレンス マニュアル

レジスタ 7-5: PLLFBD: PLL フィードバック分周比レジスタ

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

<b>凡例:</b>			
R = 読み出し可能ビット	W = 書き込み可能ビット	U = 未実装ビット、「0」として読み出し	
-n = POR 時の値	1 = ビットをセット	0 = ビットをクリア	x = ビットは未知

bit 15-9	<b>未実装:</b> 「0」として読み出し
bit 8-0	<b>PLLDIV&lt;8:0&gt;:</b> PLL フィードバック分周比ビット (「M」、PLL マルチプライヤとも呼ぶ)
	11111111 = 513
	•
	•
	•
	000110000 = 50 (既定値)
	•
	•
	•
	000000010 = 4
	000000001 = 3
	000000000 = 2

レジスタ 7-6: OSCTUN: FRC オシレータ調整レジスタ

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	

**凡例:**

R = 読み出し可能ビット      W = 書き込み可能ビット      U = 未実装ビット、「0」として読み出し  
 -n = POR 時の値      1 = ビットをセット      0 = ビットをクリア      x = ビットは未知

bit 15-6      **未実装:** 「0」として読み出し  
 bit 5-0      **TUN<5:0>:** 高速 RC オシレータ調整ビット  
              011111 = 中心周波数 + 11.625% (8.23 MHz)  
              011110 = 中心周波数 + 11.25% (8.20 MHz)  
              •  
              •  
              •  
              000001 = 中心周波数 + 0.375% (7.40 MHz)  
              000000 = 中心周波数 (7.37 MHz 公称値)  
              111111 = 中心周波数 - 0.375% (7.345 MHz)  
              •  
              •  
              •  
              100001 = 中心周波数 - 11.625% (6.52 MHz)  
              100000 = 中心周波数 - 12% (6.49 MHz)

## 7.5 プライマリ オシレータ (Posc)

dsPIC33F は Posc を OSC1 および OSC2 ピンで生成します。この接続により、外部の水晶またはセラミック振動子からデバイスへクロックを供給できます。あるいは内部 PLL と併用する事により、Fosc を 80 MHz へ逡倍して 40 MIPS での実行が可能です。Posc は下記の動作モードを備えます。

- **中速オシレータ (XT モード)**

XT モードは中ゲイン / 中周波数のモードです。このモードは 3 ~ 10 MHz の水晶振動子周波数で動作させる場合に使用します。

- **高速オシレータ (HS モード)**

HS モードは高ゲイン / 高周波数のモードです。このモードは 10 ~ 40 MHz の水晶振動子周波数で動作させる場合に使用します。

- **外部クロック源動作 (EC モード)**

EC モードでは、内部オシレータを使用せずにバイパスします。この場合、外部クロック源 (0.8 ~ 64 MHz) からデバイスへクロックを供給します (OSC1 ピンで入力)。

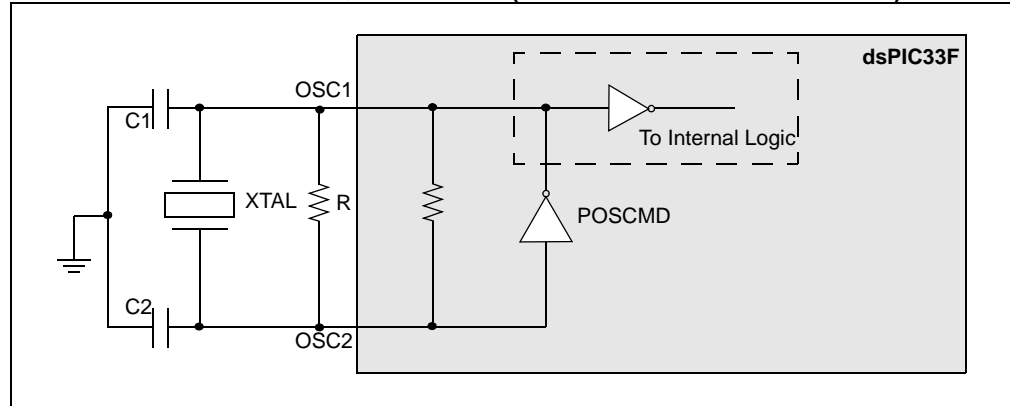
クロック源選択レジスタ (FOSCSEL<2:0>)内の FNOSC<2:0> コンフィグレーションビットは、POR 時の Posc クロック源を指定します。オシレータ コンフィグレーション レジスタ (FOSC<1:0>) 内の POSCMD<1:0> コンフィグレーション ビットは、プライマリ オシレータ モードを指定します。表 7-2 にビット設定によるオプション選択を示します。これらのビット設定は、デバイス プログラミング時に書き込みます。

表 7-2: プライマリ オシレータ クロック源のオプション選択

FNOSC	POSCMD	プライマリ クロック源およびモード
010	00	プライマリ オシレータ : 外部クロックモード (EC)
010	01	プライマリ オシレータ : 中速モード (XT)
010	10	プライマリ オシレータ : 高速モード (HS)
011	00	PLL 併用プライマリ オシレータ : 外部クロックモード (ECPLL)
011	01	PLL 併用プライマリ オシレータ : 中速モード (XTPLL)
011	10	PLL 併用プライマリ オシレータ : 高速モード (HSPLL)

dsPIC33F 向けに推奨する水晶振動子回路を図 7-1 に示します。

図 7-3: 水晶またはセラミック振動子の回路図 (XT または HS オシレータモード用)



コンデンサ C1 および C2 は、水晶振動子の負荷容量を形成します。

負荷容量 (CL) の値は振動子メーカーの指定に従います。CL は式 7-1 のように計算できます。

## 式 7-1: 水晶振動子の負荷容量

$$CL = Cs + \frac{C1 \times C2}{C1 + C2}$$

Cs は浮遊容量

C1 = C2 とした場合のコンデンサ容量 (C1、C2) は、負荷容量と浮遊容量から式 7-2 により求められます。

## 式 7-2: 水晶振動子の外部コンデンサ容量

$$C1 = C2 = 2 \times (CL - Cs)$$

水晶振動子とその動作に関する詳細は 7.14「関連アプリケーションノート」を参照してください。

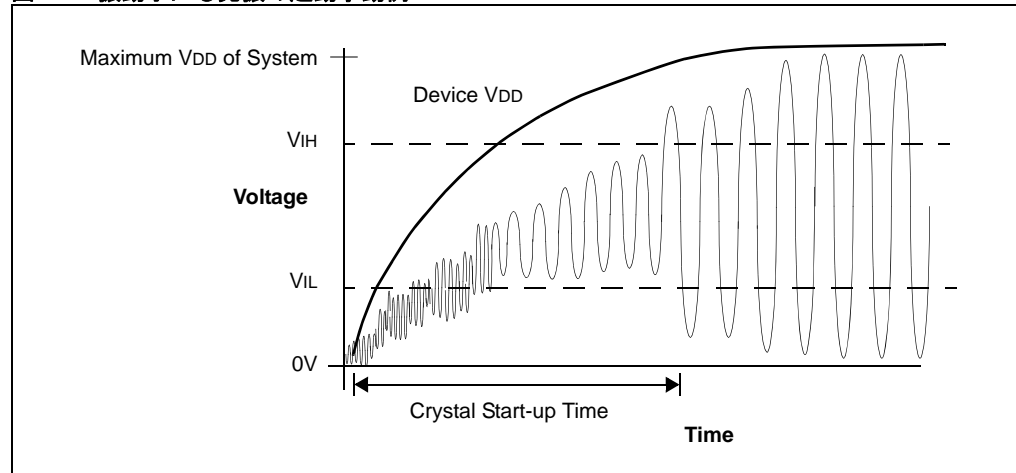
### 7.5.1 オシレータ起動時間

デバイスの電圧が VSS から上昇するにつれオシレータの発振が始まります。オシレータが発振を開始するまでに要する起動時間には、下記を含む多くの要因が影響します。

- 水晶 / セラミック振動子の周波数
- コンデンサの容量
- デバイスの VDD 立ち上がり時間
- システムの温度
- 直列抵抗を使用する場合は、その抵抗値とタイプ
- デバイスのオシレータモード (内部オシレータ インバータのゲイン選択)
- 水晶振動子の品質
- オシレータ回路のレイアウト
- システムノイズ

図 7-4 に振動子による発振の典型的な起動挙動を示します。

図 7-4: 振動子による発振の起動挙動例



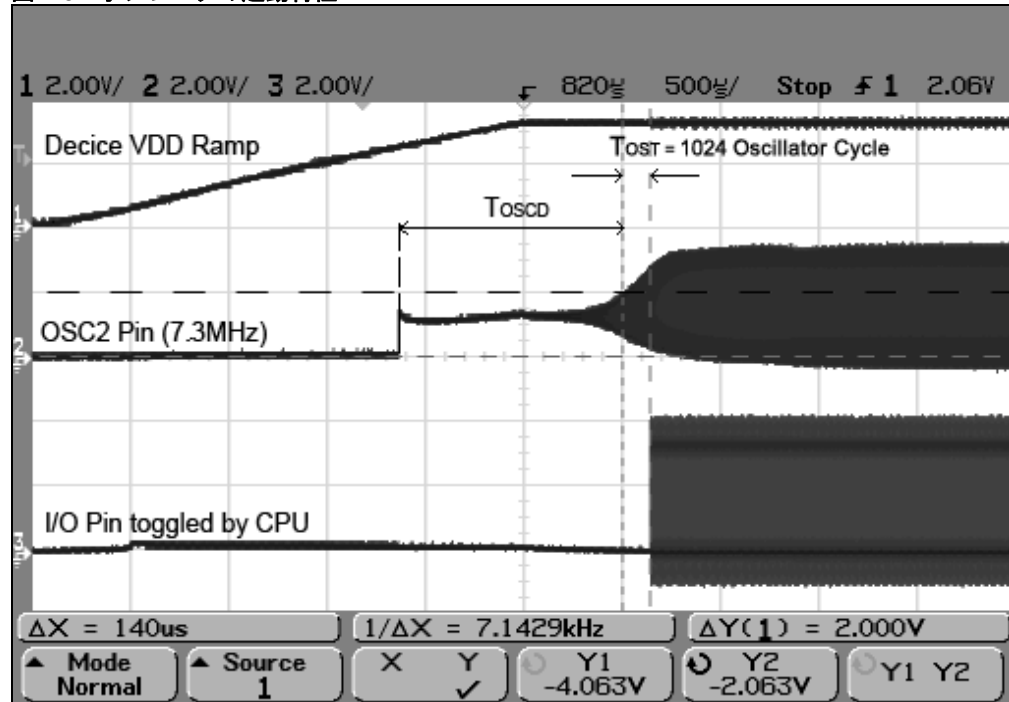
水晶またはセラミック振動子を確実に起動して安定させるために、Posc と Sosc はオシレータ起動タイマ (OST) を備えます。OST は 1024 サイクルをカウントする単純な 10 ビットカウンタです。このカウントの後にシステムへのオシレータ クロックの供給を開始します。この遅延時間を TOST と表記します。

オシレータ出力ピン上のオシレータ信号振幅がしきい値  $V_{IL}$  および  $V_{IH}$  に達した後に、OST がサイクルのカウントを開始します。オシレータを再起動するたびに  $T_{OST}$  遅延が必要です (POR、BOR、スリープモードからのウェイクアップ時等)。

$Posc$  を起動してから発振が始まるまでに一定の時間がかかります。この遅延時間を  $T_{OSCD}$  と表記します。 $T_{OSCD}$  の後に、OST タイマが 1024 クロックサイクル ( $T_{OST}$ ) をカウントしてから、クロックの供給が始まります。 $T_{OSCD}$  と  $T_{OST}$  の合計時間が、クロック供給が可能になるまでの総遅延時間です。PLL を使用する場合、PLL をロックするために追加の遅延時間が必要です (7.7 「位相ロックループ (PLL)」 参照)。

$Posc$  の起動挙動を図 7-5 に示します。 $T_{OSCD} + T_{OST}$  の遅延後に CPU が動作 (I/O ピンのトグル) を開始しています。

図 7-5: オシレータの起動特性



## 7.5.2 $Posc$ ピンの機能

プライマリ オシレータを使用しない場合、 $Posc$  ピン (OSC1 と OSC2) を他の機能向けに使用できます。オシレータピン機能は、オシレータ コンフィグレーション レジスタ ( $FOSC<1:0>$ ) の  $POSCMD<1:0>$  コンフィグレーション ビットで指定します。OSC2 ピンの機能は  $OSCIOFNC$  ビット ( $FOSC<2>$ ) で指定します。

### $POSCMD$ : プライマリ オシレータモードの選択ビット

- 11 = プライマリ オシレータモードは無効
- 10 = HS オシレータモードを選択
- 01 = XT オシレータモードを選択
- 00 = 外部クロックモードを選択

### $OSCIOFNC$ : OSC2 ピンの機能ビット (XT および HS モード以外)

- 1 = OSC2 をクロック出力ピンとして使用し、命令サイクル ( $F_{CY}$ ) クロックを出力 (図 7-6 参照)
- 0 = OSC2 を汎用デジタル I/O ピンとして使用 (図 7-7 参照)



オシレータピン機能の一覧を表 7-3 に示します。

表 7-3: クロックピン機能の選択

クロック源	OSCIOFNC	POSCMD<1:0>	OSC1 <sup>(1)</sup> ピン機能	OSC2 <sup>(2)</sup> ピン機能
Posc 無効	1	11	デジタル I/O	クロック出力 (FcY)
Posc 無効	0	11	デジタル I/O	デジタル I/O
HS	x	10	OSC1	OSC2
XT	x	01	OSC1	OSC2
EC	1	00	OSC1	クロック出力 (FcY)
EC	0	00	OSC1	デジタル I/O

**Note 1:** OSC1 ピンの機能は、プライマリ オシレータモード コンフィグレーション ビット (POSCMOD<1:0>) で指定します。

**2:** OSC2 ピンの機能は、プライマリ オシレータモード (POSCMOD<1:0>) と OSC2 ピン機能コンフィグレーション ビット (OSCIOFNC) で指定します。

図 7-6: OSC2 ピンをクロック出力として使用する場合 (EC モード)

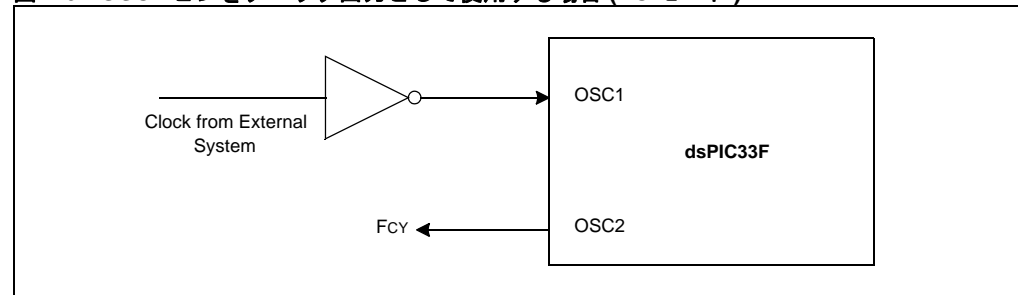
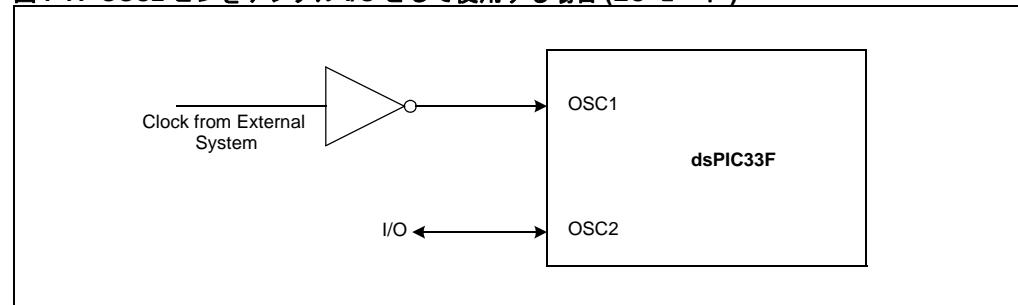


図 7-7: OSC2 ピンをデジタル I/O として使用する場合 (EC モード)



## 7.6 内部高速 RC (FRC) オシレータ

内部 FRC オシレータは、外部の水晶またはセラミック振動子を必要とせずに 7.37 MHz ( 公称値 ) のクロックを供給します。このため、精密なクロック リファレンスを要求しないアプリケーションでは、システムコストを削減できます。

アプリケーション ソフトウェアは、公称周波数 -12 ~ +11.625% の範囲 (30 kHz ステップ) で周波数を調整できます。この調整には、FRC オシレータ調整レジスタ (OSCTUN<5:0>) 内の FRC オシレータ調整ビット (TUN<5:0>) を使用します。

- Note 1:** FRC クロック周波数精度の温度および電圧依存性は各デバイスのデータシートを参照してください。
- 2:** PLL 併用内部 FRC で動作中に FRC オシレータ調整ビット (TUN<5:0>) を変更しないでください。
- この場合の FRC オシレータ調整ビットの変更方法
- a) クロックを非 PLL モード ( 内部 FRC モード等 ) へ切り換えます。
  - b) FRC オシレータ調整ビットを変更します。
  - c) クロックを PLL 併用モードへ戻します。

内部 FRC オシレータは瞬時に起動します。水晶振動子は発振を開始するまでに数 ms を要する場合がありますが、内部 FRC オシレータは即座に発振を開始します。

FRC クロック源は、クロック源選択レジスタ (FOSCSEL<2:0>) 内の初期クロック源選択コンフィグレーション ビットで選択します。POR 時の FRC クロック源オプションを表 7-4 に示します。これらのコンフィグレーション ビットは、デバイス プログラミング時に書き込みます。

表 7-4: POR 時の FRC クロック源オプション

FNOSC<2:0>	プライマリ クロック源およびモード
000	FRC オシレータ (FRC)
001	PLL 併用 FRC オシレータ (FRCPLL)
110	FRC オシレータ : 16 分周ポストスケーラ (FRCDIV16)
111	FRC オシレータ : N 分周ポストスケーラ (FRCDIVN)

### 7.6.1 FRC ポストスケーラ モード (FRCDIV)

FRC ポストスケーラ モードでは、可変ポストスケーラを使用して FRC クロック出力を分周する事により、クロック周波数よりも低い周波数を選択できます。クロック分周比レジスタ (CLKDIV<10:8>) 内の内部 FRC オシレータ ポストスケーラ ビット (FRCDIV<2:0>) でポストスケーラを制御する事により、1:1 ~ 1:256 の範囲で 8 段階の分周比を選択できます (表 7-5 参照)。

表 7-5: 内部 FRC オシレータのポストスケーラ設定

FRCDIV<2:0>	内部 FRC オシレータの設定
000	FRC を分周しない ( 既定値 )
001	FRC を 2 分周
010	FRC を 4 分周
011	FRC を 8 分周
100	FRC を 16 分周
101	FRC を 32 分周
110	FRC を 64 分周
111	FRC を 256 分周

あるいは、FRC ポストスケーラ出力と内部 PLL を併用する事により、Fosc を 80 MHz へ通倍して 40 MIPS での実行が可能です。

**Note:** PLL 併用内部 FRC モードで動作中に FRC 分周比を変更しないでください。

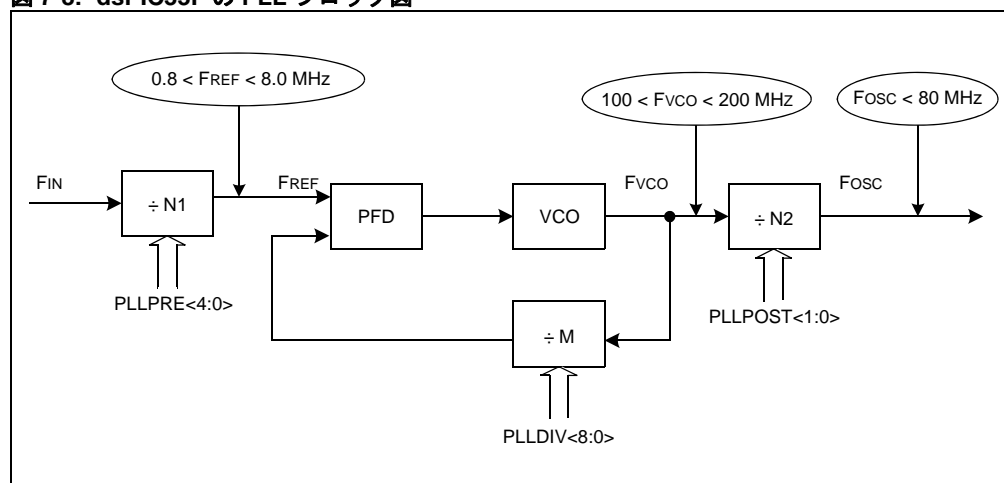
この場合の FRC 分周比の変更方法

1. クロックを非 PLL モード ( 内部 FRC モード等 ) へ切り換えます。
2. FRC オシレータ調整ビットを変更します。
3. クロックを PLL 併用モードへ戻します。

## 7.7 位相ロックループ (PLL)

Posc または内部 FRC クロック源に内蔵 PLL を併用すると、動作速度を高める事ができます。図 7-8 に PLL モジュールのブロック図を示します。

図 7-8: dsPIC33F の PLL ブロック図



PLL が正しく動作するには、位相周波数検出器 (PFD) の入力周波数と電圧制御オシレータ (VCO) の出力周波数が下記の要件を満たす必要があります。

- PFD 入力周波数 (FREF) が 0.8 ~ 8.0 MHz である事
- VCO 出力周波数 (FVCO) が 100 ~ 200 MHz である事

クロック分周比レジスタ (CLKDIV<4:0>) 内の PLL 位相検出器入力分周比選択ビット (PLLPRE<4:0>) で入力分周比 (N1) を指定する事により、入力周波数 (FIN) を PFD 入力周波数の許容範囲内 (0.8 ~ 8 MHz) へ低減します。

PLL フィードバック分周比レジスタ (PLLFBD<8:0>) 内の PLL フィードバック分周比ビット (PLLDIV<8:0>) では、PFD へのフィードバック周波数 FVCO を低減するために分周比 (M) を指定します。FVCO は FREF の M 倍の周波数を持ちます。

クロック分周比レジスタ (CLKDIV<7:6>) 内の PLL VCO 出力分周比選択ビット (PLLPOST<1:0>) では、FOSC を許容範囲 (< 80 MHz) に制限するために分周比 (N2) を指定します。

式 7-3 に FIN と FOSC の関係を示します。

式 7-3: Fosc (< 80 MHz) の計算式

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

$$N1 = PLLPRE + 2$$

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

$$M = PLLDIV + 2$$

式 7-4 に FIN と FVCO の関係を示します。

式 7-4: Fvco の計算式

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

## 7.7.1 PLL モードにおける起動時入力クロックの制限

PLL プリスケアラ、PLL ポストスケアラ、PLL フィードバック分周比コンフィグレーションビットの POR 時の既定値を表 7-6 に示します。

表 7-6: PLL モジュールの既定値

レジスタ	ビットフィールド	POR 時の値	PLL 分周比
CLKDIV<4:0>	PLLPRE<4:0>	00	N1 = 2
CLKDIV<7:6>	PLLPOST<1:0>	01	N2 = 4
PLLFBD<8:0>	PLLDIV<8:0>	000110000	M = 50

これらの既定値を使用した場合の POR 時の  $F_{IN}$ 、 $F_{REF}$ 、 $F_{VCO}$ 、 $F_{OSC}$  の関係は下式で示されます。

式 7-5: POR 時の  $F_{REF}$

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

式 7-6: POR 時の  $F_{VCO}$

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

式 7-7: POR 時の  $F_{OSC}$

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

上式から、 $PLLPRE$ 、 $PLLPOST$ 、 $PLLDIV$  の既定値を使用した場合、 $F_{VCO}$  を許容範囲 (100 ~ 200 MHz) に収めるには、PLL モジュールへの入力周波数  $F_{IN}$  を 4 ~ 8 MHz の範囲に制限する必要があります。

$Posc$  は下記の入力周波数に対応可能です。これらは POR 時の周波数要件 (4 MHz <  $F_{IN}$  < 8 MHz) を超えています。

- XT モード時の  $Posc$ : 3 ~ 10 MHz の水晶振動子
- HS モード時の  $Posc$ : 10 ~ 40 MHz の水晶振動子
- EC モード時の  $Posc$ : 0.8 ~ 64 MHz の周波数入力

許容範囲 (4 ~ 8 MHz) を超える入力周波数で PLL を使用するには、下記の操作が必要です。

1. PLL を使用せずに内部 FRC オシレータまたは  $Posc$  を使用してデバイスを起動します。
2. その入力周波数に対して、下記の PLL 要件を満足するように  $PLLDIV$ 、 $PLLPRE$ 、 $PLLPOST$  ビットの値を変更します。
  - $F_{REF}$  の許容範囲: 0.8 ~ 8.0 MHz
  - $F_{VCO}$  の許容範囲: 100 ~ 2000 MHz
3. ソフトウェアでクロックを PLL モードへ切り換えます。

## 7.7.2 PLL ロック状態

PLL 入力周波数、PLL プリスケーラ、PLL フィードバック分周比のいずれかを変更した場合、PLL を新たな設定に同期させるために一定の時間 (TLOCK) が必要です。

POR 時またはクロック切り換え中のクロック源として PLL を選択した場合、TLOCK が適用されます。TLOCK は PLL 入力へのクロック供給開始時点を起点とする追加の遅延時間です。例えば Posc を使用する場合、OST 遅延の後に TLOCK を追加します。オシレータ起動時の遅延については 7.5.1「オシレータ起動時間」を参照してください。TLOCK の値は各デバイスのデータシートを参照してください。

オシレータ制御レジスタ (OSCCON<5>) 内の LOCK ビットは、PLL のロック状態を示す読み出し専用のステータスビットです。PLL を切り換え先クロック源として選択した場合、POR 時およびクロック切り換え動作中に LOCK ビットがクリアされます。PLL を併用しないクロック源を選択した場合の LOCK ビットはクリア状態を維持します。PLL を使用するクロック切り換えイベントの後では、LOCK ビットがセットされるまで待機し、その後に他のコードを実行する事を推奨します。

**Note:** PLL モードで動作中に PLLPRE ビットと PLLDIV ビットを変更しないでください。変更する場合は、一度クロックを非 PLL モード (内蔵 FRC モード等) へ切り換え、変更後に PLL モードへ戻す必要があります。

### 7.7.2.1 POSC で PLL を併用する場合のセットアップ

10 MHz の外部水晶振動子を使用してデバイスを 40 MIPS で動作させる場合の PLL セットアップ手順は以下の通りです。

1. 40 MHz で命令を実行するために、下記のシステムクロック周波数を確保します。  
 $F_{OSC} = 2 \times F_{CY} = 80 \text{ MHz}$
2. PLLPRE、PLLPOST、PLLDIV の既定値設定が PLL 要件とユーザ要件を満たしている事を確認します。
3. PLL およびユーザ要件を満たしている場合、POR 時に PLL 併用 Posc を選択するように POR.FNOSC<2:0> ビット (FOSCSEL<2:0>) を直接設定します。  
PLL およびユーザ要件を満たしていない場合、下記の手順に従います。
  - a) VCO 出力周波数要件 ( $100 < F_{VCO} < 200 \text{ MHz}$ ) を満たす PLL ポストスケーラを選択します。
  - b) PFD 入力周波数要件 ( $0.8 < F_{REF} < 8 \text{ MHz}$ ) を満たす PLL プリスケーラを選択します。
  - c) その PFD 入力周波数に対して適正な VCO 出力周波数を生成する PLL フィードバック分周比を選択します。
  - d) POR 時に非 PLL クロック源 (内部 FRC 等) を選択するように FNOSC<2:0> ビット (FOSCSEL<2:0>) を設定します。
  - e) メインプログラム内で PLL プリスケーラ、PLL ポストスケーラ、PLL フィードバック分周比を上記で選択した値へ変更してから、クロックを PLL 併用モードへ切り換えます。

例 7-1 に、PLL 併用 Posc 向けのサンプルコードを示します ( クロック切り換えのサンプルコードはセクション 7.11 「クロック切り換え」 参照 )。

**例 7-1: PLL 併用 Posc 向けサンプルコード**

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

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

int main()
{

// Configure PLL prescaler, PLL postscaler, PLL divisor
PLLFBFBD=30;           // M = 32
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(0x01);

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

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

}
```

## 7.7.2.2 PLL 併用 7.37 MHz 内部 FRC のセットアップ

7.37 MHz の内部 FRC を使用してデバイスを 40 MIPS で動作させる場合の PLL セットアップ手順は下記に従います。

1. 40 MHz で命令を実行するために、下記のシステムクロック周波数を確保します。

$$F_{OSC} = 2 \times F_{CY} = 80 \text{ MHz}$$

2. PLLPRE、PLLPOST、PLLDIV の既定値設定が PLL 要件とユーザ要件を満たしている事を確認します。
3. PLL およびユーザ要件を満たしている場合、POR 時に PLL 併用内部 FRC モードを選択するように、POR.FNOSC<2:0> ビット (FOSCSEL<2:0>) を直接設定します。

PLL およびユーザ要件を満たしていない場合、下記の手順に従います。

- a) VCO 出力周波数要件 ( $100 < F_{VCO} < 200 \text{ MHz}$ ) を満たす PLL ポストスケーラを選択します。
- b) PFD 入力周波数要件 ( $0.8 < F_{REF} < 8 \text{ MHz}$ ) を満たす PLL プリスケラを選択します。
- c) その PFD 入力周波数に対して適正な VCO 出力周波数を生成する PLL フィードバック分周比を選択します。
- d) POR 時に非 PLL クロック源 (内部 FRC 等) を選択するように FNOSC<2:0> ビット (FOSCSEL<2:0>) を設定します。
- e) メインプログラム内で、PLL プリスケラ、PLL ポストスケーラ、PLL フィードバック分周比を PLL およびユーザ要件を満たすように変更してから、クロックを PLL 併用モードへ切り換えます。

例 7-2 に、PLL 併用 7.37 MHz 内部 FRC 向けのサンプルコードを示します ( クロック切り換えのサンプルコードは 7.11 「クロック切り換え」 参照 )。

## 例 7-2: PLL 併用 7.37 MHz 内部 FRC 向けのサンプルコード

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

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

int main()
{

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

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

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

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

}
```



## 7.8 セカンダリ オシレータ (Sosc)

セカンダリ オシレータ (Sosc) を使用すると、低消費電力動作向けの補助的な水晶クロック源として 32.768 kHz 水晶振動子を dsPIC33F デバイスへ追加できます。これには SOSCI および SOSCO ピンを使用します。Sosc はリアルタイム クロック (RTC) アプリケーション向けに Timer1 を駆動する事もできます。

**Note:** Sosc は省電力化に有効であることから、低消費電力セカンダリ オシレータと呼ばれる事もあります。このオシレータを LPRC オシレータと混同しないように注意してください。

### 7.8.1 セカンダリ オシレータをシステムクロックとして使用する

下記の場合に Sosc をシステムクロックとして使用します。

- クロック源選択レジスタ (FOSCSEL<2:0>) 内の初期クロック源選択コンフィグレーション ビット (FNOSC<2:0>) で、POR 時のオシレータとして Sosc を選択した場合
- ユーザ アプリケーションが低消費電力動作のためにクロックを Sosc へ切り換えた場合

Sosc をシステムクロック源として使用していない場合、またはデバイスがスリープモードへ切り換わった場合には、Sosc は省電力化のために停止します。

### 7.8.2 セカンダリ オシレータの起動遅延

Sosc を起動してから発振が始まるまでに一定の時間が必要です。詳細は 7.5.1「オシレータ起動時間」を参照してください。

### 7.8.3 セカンダリ オシレータの連続動作

必要に応じて Sosc を常時動作させる事もできます。オシレータ制御レジスタ (OSCCON<1>) 内のセカンダリ オシレータ イネーブルビット (LPOSCEN) をセットすると、Sosc は常時動作します。

Sosc を常時動作させる理由として、下記の 2 つが挙げられます。

- Sosc を常時動作させておくと、低消費電力動作の 32 kHz システムクロックへ高速に切り換える事ができます。しかし、高速メインクロック源が水晶振動子の場合、32 kHz から高速メインクロック源に戻る際に、やはり一定のオシレータ起動時間が必要です (7.5.1「オシレータ起動時間」参照)。
- RTC として Timer1 を使用する場合、セカンダリ オシレータを常時動作させる必要があります。

**Note:** Posc、内部 FRC オシレータ、LPRC オシレータはスリープモード時に停止します。これらに対し Sosc は、オシレータ制御レジスタ (OSCCON<1>) 内のセカンダリ オシレータ イネーブルビット (LPOSCEN) をセットする事により、スリープモード中でも動作可能です。

## 7.9 低消費電力 RC (LPRC) オシレータ

低消費電力 RC (LPRC) オシレータは、公称周波数 32 kHz のクロックを供給します。LPRC オシレータはパワーアップ タイマ (PWRT)、ウォッチドッグ タイマ (WDT)、フェイルセーフ クロック モニタ (FSCM) 回路へクロックを供給します。また、タイミング精度よりも低消費電力性を重視するアプリケーションでは、LPRC オシレータを使用してデバイスに低周波数のクロックを供給する事もできます。

**Note:** LPRC のクロック周波数はデバイスの電圧と動作温度によって変化します。詳細は各デバイス データシート内の「電気的特性」を参照してください。

### 7.9.1 LPRC オシレータをシステムクロックとして使用する

下記の場合に LPRC オシレータをシステムクロックとして使用します。

- クロック 源選択レジスタ (FOSCSEL<2:0>) 内の初期クロック 源選択コンフィグレーション ビット (FNOSC<2:0>) で、POR 時に使用するオシレータとして LPRC を選択した場合
- ユーザ ソフトウェアが低消費電力動作向けにクロック 源を LPRC オシレータへ切り換えた場合

### 7.9.2 LPRC オシレータの有効化

LPRC オシレータは PWRT、WDT、FSCM 向けのクロック 源です。POR コンフィグレーション ヒューズ レジスタ (FPOR<2:0>) 内のパワーオン リセットタイマ値選択ビット (FPWRT) をセットすると、LPRC オシレータが有効になります。

LPRC オシレータは、下記の条件で動作を維持します。

- FSCM が有効な場合
- WDT が有効な場合
- LPRC オシレータをシステムクロックとして選択した場合

上記の条件がいずれも真でない場合、LPRC オシレータは PWRT 経過後に停止します。LPRC オシレータはスリープモード時に停止します。

**Note:** LPRC は、WDT が有効な場合にのみ、スリープモードでも動作します。これ以外の場合 LPRC はスリープモード時に停止します。

### 7.9.3 LPRC オシレータの起動遅延

水晶振動子は発振を開始するまでに数msを要しますが、LPRCオシレータは即座に起動します。

## 7.10 フェイルセーフ クロックモニタ (FSCM)

フェイルセーフ クロックモニタ (FSCM) は、オシレータ障害発生時のデバイスの動作維持を可能にします。FSCM 機能を有効にするには、デバイス プログラミング時にオシレータ コンフィグレーション レジスタ (FOSC<7:6>) 内のクロック切り換えモード コンフィグレーション ビット (FCKSM<1:0>) を設定します。FSCM を有効 (FCKSM<1:0> = 00) に設定すると、LPRC オシレータはスリープモード時を除いて常時動作します。

FSCM はシステムクロックを監視します。FSCM は指定された時間内 (通常 2 ms、最大 4 ms) にシステムクロックを検出できなかった場合にクロック障害トラップを発生し、システムクロックを内部 FRC オシレータへ切り換えます。この時ユーザ アプリケーションは、オシレータの再起動を試みるか、あるいは制御シャットダウンを実行できます。

**Note:** スリープモード中にクロック障害が発生しても、FSCM はデバイスをウェイクアップしません。

内部 FRC オシレータへ切り換える場合、FSCM モジュールは下記のように動作します。

- ・ オシレータ選択ビット COSC<2:0> (OSCCON<14:12>) を「000」(内部 FRC オシレータ) に設定します。
- ・ クロック障害検出ビット CF (OSCCON<3>) をセットしてクロック障害の発生を示します。
- ・ オシレータ切り換えイネーブル制御ビット OSWEN (OSCCON<0>) をクリアして保留中の全てのクロック切り換えをキャンセルします。

### 7.10.1 FSCM 遅延

FSCM は、システムクロックが動作状態となった後、さらに一定の遅延時間 ( $T_{FSCM}$ ) が経過してからシステムクロックの監視を開始します。

FSCM が有効かつ Posc または Sosc をシステムクロックとして選択している場合、FSCM 遅延を適用します。

**Note:**  $T_{FSCM}$  の値については、各デバイス データシート内の「電気的特性」を参照してください。

その他の情報はセクション 8. 「リセット」(DS70192) を参照してください。最新の文書はマイクロチップ社のウェブサイト ([www.microchip.com](http://www.microchip.com)) でご覧になれます。

### 7.10.2 FSCM と WDT

FSCM と WDT はどちらもタイムベースとして LPRC オシレータを使用します。クロック障害が発生しても WDT は影響を受けず、LPRC オシレータを使用して動作を続けます。

## 7.11 クロック切り換え

クロック切り換えは、ハードウェア イベントまたはソフトウェア要求により開始できます。切り換え動作の代表例を以下に挙げます。

- POR 時の 2 段階起動シーケンスは、まず内部 FRC オシレータを使用して高速に起動し、その後、選択したクロック源の動作が安定してから自動的にそのクロック源へ切り換えます。
- クロック障害時に、FSCM が自動的に内部 FRC オシレータへ切り換えます。
- ユーザアプリケーション ソフトウェアが OSWEN ビット (OSCCON<0>) をセットしてクロック切り換えを要求すると、ハードウェアはクロック源を NOSC<2:0> ビット (OSCCON<10:8>) が指定するクロック源へ切り換えます (そのクロックが安定してから切り換え)。

これらの切り換え動作はメークビフォーブレーク シーケンスに従います。すなわち、必ず新しいクロックへの切り換えが完了してから、古いクロックを停止します。また、クロック切り換え中でもコード実行を継続します。

一部の dsPIC33F デバイスは、FWDТ ヒューズ コンフィグレーション レジスタ (FWDТ<5>) 内に位相ロックループ イネーブルビット (PLLKEN) を備えます。このビットをセットすると、デバイスは PLL がロックするまで待機し、その後に PLL クロック源へ切り換えます。このビットが「0」の場合、デバイスは PLL ロックを待たずにクロックを切り換えます。このビットの既定値は「1」です。詳細は dsPIC33F ファミリ リファレンス マニュアルのセクション 25.「デバイス設定」(DS70194) を参照してください。

アプリケーションは、ソフトウェアが制御する 4 つのクロック源 (POSC, SOSC, FRC, LPRC) を常時ほぼ無制限に切り換える事ができます。このような柔軟な切り換え動作による予期せぬ結果を防ぐために、dsPIC33F は切り換えプロセスに保護ロック機能を組み込んでいます。この機能は、クロック切り換え中に OSCCON レジスタを書き込み保護します。

### 7.11.1 クロック切り換えの有効化

クロック切り換えと FSCM を有効にするには、オシレータ コンフィグレーション レジスタ (FOSC<7:6>) 内のクロック切り換えモード コンフィグレーション ビット (FCKSM<1:0>) を設定する必要があります。

表 7-7: 設定可能なクロック切り換えモード

FCKSM<1:0>	クロック切り換え	FSCM
1x	無効	無効
01	有効	無効
00	有効	有効

上位ビットはクロック切り換えの有効 (0)/ 無効 (1) を指定し、下位ビットは FSCM の有効 (0)/ 無効 (1) を指定します。FSCM はクロック切り換えが有効な場合にのみ有効にできます。クロック切り換えが無効 (1) の場合、下位ビットは無視されます。

### 7.11.2 クロック切り換えシーケンス

クロック切り換えの推奨手順は下記の通りです。

1. COSC<2:0> ビット (OSCCON<14:12>) を読み出して、現在使用中のクロック源を取得します (この情報がアプリケーションに必要な場合)
2. ロック解除シーケンスを実行し、OSCCON レジスタの上位バイトへの書き込みを許可します。
3. 切り換え先のクロック源を指定する値を NOSC 制御ビット (OSCCON<10:8>) に書き込みます。
4. ロック解除シーケンスを実行し、OSCCON レジスタの下位バイトへの書き込みを許可します。
5. OSWEN ビット (OSCCON<0>) をセットしてオシレータ切り換えを開始します。

上記の手順が完了した後に、クロック切り換えロジックは下記の手順を実行します。

1. クロック切り換えハードウェアは COSC<2:0> ステータスビット (OSCCON<14:12>) と NOSC<2:0> 制御ビット (OSCCON<10:8>) の値を比較します。これらの値が一致する場合、クロック切り換えは冗長動作です。この場合 OSWEN ビット (OSCCON<0>) を自動的にクリアしてクロック切り換えを中止します。
2. クロック切り換えを開始した場合、PLL ロック ステータスビット (OSCCON<5>) と クロック障害ステータスビット (OSCCON<3>) をクリアします。
3. 切り換え先のオシレータが動作中でなければ、ハードウェアはそのオシレータを起動します。水晶振動子 (Posc または Sosc) を起動する必要がある場合、ハードウェアは発振が始まるまでの遅延時間 (T<sub>OSCD</sub>) とこれに続く T<sub>OST</sub> が経過するまで待機します。切り換え先のクロック源が PLL を併用する場合、ハードウェアは PLL ロック (LOCK = 1) を検出するまで待機します。
4. ハードウェアは、切り換え先のクロック源が安定するまで待機してからクロックを切り換えます。
5. ハードウェアは OSWEN ビット (OSCCON<0>) をクリアし、クロックの切り換えが正常に完了した事を示します。加えて、NOSC<2:0> ビット (OSCCON<10:8>) の値を COSC<2:0> ステータスビット (OSCCON<14:12>) へ転送します。
6. この時点で切り換え前に使用していたクロック源を停止します。ただし、LPRC は WDT または FSCM が有効であれば停止しません。また、Sosc は SOSCEN がセットされたままであれば停止しません。クロック切り換え時のタイミングを図 7-9 に示します。

**Note 1:** XT、HS、EC プライマリ オシレータモード間でクロックを切り換えるには、デバイスの再プログラミングが必要です。

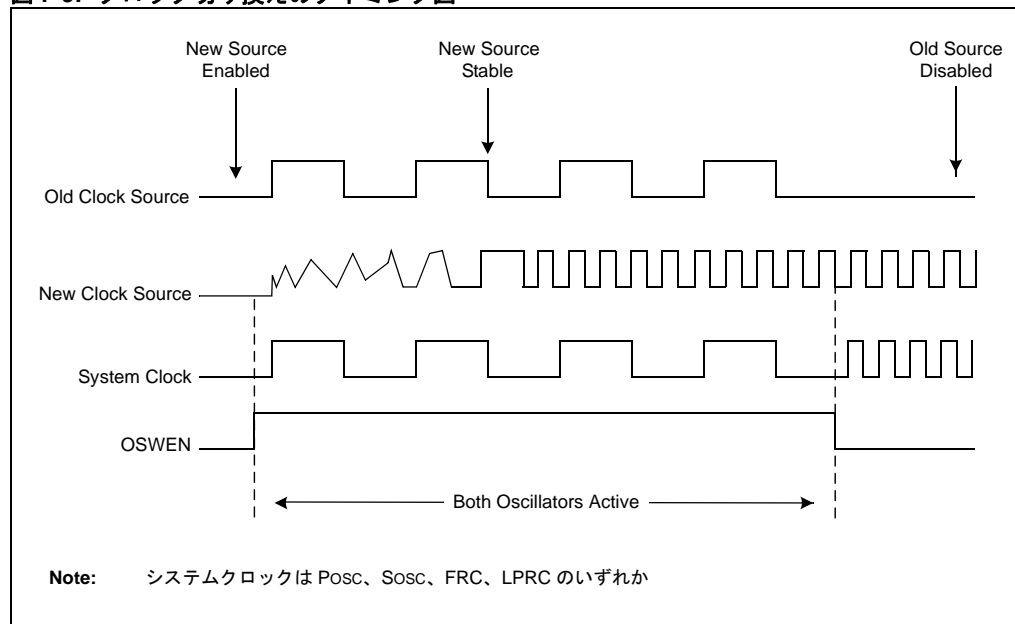
**2:** 各種 PLL モード間の直接的なクロック切り換えはできません。例えば、PLL 併用 Posc と PLL 併用内部 FRC オシレータ間のクロック切り換えを行わないでください。

**3:** コンフィグレーション ビット FCKSM<1:0> (FOSC<7:6>) の設定が 01 (クロック切り換え有効 / FSCM 無効) の場合に CLKLOCK ビット (OSCCON<7>) をセットするとクロック切り換えを回避できます。ソフトウェアからセットした CLKLOCK ビット (OSCCON<7>) をクリアする事はできません。このビットは POR 時にクリアされます。

**4:** プロセッサはクロック切り換えシーケンス中でもコードを実行し続けます。この間にタイミングが重要なコードを実行しないでください。

**5:** FWDT ヒューズ コンフィグレーション レジスタ (FWDT<5>) 内の PLLKEN ビットを「0」に設定すると、クロック切り換えは PLL ロックを待機しません。

図 7-9: クロック切り換えのタイミング図



クロック切り換えの推奨コードシーケンスを以下に示します。

1. OSCCON レジスタのロック解除および書き込みシーケンス中の割り込みを禁止します。
2. OSCCON の上位バイトに対するロック解除シーケンスを実行します (下記 2 つの命令を連続して実行)。
  - 0x78 を OSCCON<15:8> へ書き込む
  - 0x9A を OSCCON<15:8> へ書き込む
3. ロック解除シーケンス直後の命令で、NOSC<2:0> 制御ビット (OSCCON<10:8>) に切り換え先のクロック源を設定します。
4. OSCCON の下位バイトに対するロック解除シーケンスを実行します (下記 2 つの命令を連続して実行)。
  - 0x46 を OSCCON<7:0> へ書き込む
  - 0x57 を OSCCON<7:0> へ書き込む
5. ロック解除シーケンス直後の命令で、OSWEN ビット (OSCCON<0>) をセットします。
6. クロックに影響されにくいコードの実行を継続します (必須ではない)。
7. OSWEN ビット (OSCCON<0>) が「0」になっている事を確認します。このビットが「0」であれば、クロック切り換えが成功した事を意味します。

**Note:** dsPIC DSC 向け MPLAB®C コンパイラは、OSCCON レジスタのロック解除用に下記のビルトイン C 関数を提供します。

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

詳細は MPLAB C コンパイラのヘルプを参照してください。

OSCCON レジスタのロックを解除してクロック源を PLL 併用 FRC から LPRC へ切り換える場合のコードシーケンスを例 7-3 に示します。

## 例 7-3: クロック切り換えのサンプルコード

```
;Place the New Oscillator Selection (NOSC=0b101) in W0  
MOV #0x15, WREG  
  
;OSCCONH (high byte) Unlock Sequence  
MOV #OSCCONH, w1  
MOV #0x78, w2  
MOV #0x9A, w3  
MOV.B w2, [w1] ; Write 0x78  
MOV.B w3, [w1] ; Write 0x9A  
  
;Set New Oscillator Selection  
MOV.B WREG, OSCCONH  
  
; 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  
MOV.B w0, [w1] ; Request Clock Switching by Setting OSWEN bit  
  
wait:  
    btsc    OSCCONL, #OSWEN  
    bra     wait
```

## 7.11.3 クロック切り換えに関する注意点

クロック切り換えをアプリケーションに組み込む場合のコード作成では、下記の点に注意してください。

- OSCCON のロック解除シーケンスではタイミングが極めて重要です。OSCCON レジスタバイトへの書き込みは、ロック解除シーケンス直後の 1 命令サイクル中でのみ可能です。C 言語等の高級言語では、タイミングが重要な命令の順序がコンパイル時に維持されない場合があります。高級言語で記述したアプリケーションでクロック切り換えを実行する場合、切り換えルーチンをアセンブラで作成してアプリケーションヘリンクし、関数として呼び出す事を推奨します。
- クロック源を水晶振動子へ切り換える場合、クロック切り換え時間はオシレータの起動時間によって大きく左右されます。
- 切り換え先のクロック源が起動しない場合や存在しない場合、クロック切り換えハードウェアはそのクロック源が利用可能になるまで待機し続けます。この状態では OSWEN ビット (OSCCON<0>) がセットされたままになるため、ソフトウェアで問題を検出する事ができます。
- 切り換え先のクロック源が PLL を使用する場合、ロックが完了するまでクロックは切り換わりません。この状態では LOCK ビット (OSCCON<5>) がクリアされ OSWEN ビット (OSCCON<0>) がセットされるため、ソフトウェアで問題を検出する事ができます。
- セカンダリ オシレータ等の周波数の低いクロック源へ切り換えると、デバイスの動作速度が低下します。

## 7.11.4 クロック切り換えの中止

クロック切り換えが完了しない場合、OSWEN ビット (OSCCON<0>) をクリアする事により、クロック切り換えロジックをリセットできます。OSWEN ビットをクリアすると、クロック切り換え処理が中止されます。動作中の OST は停止 / リセットされ、動作中の PLL は停止します。

例 7-4 にクロック切り換えを中止するためのアセンブリ サンプルコードを示します。クロック切り換え処理はいつでも中止できます。また、クロック切り換えの実行中に別のクロック切り換えを実行する事によって、既に実行中の切り換えを中止する事もできます。

例 7-4: クロック切り換えの中止

```
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
```

## 7.11.5 クロック切り換え中のスリープモードへの移行

クロック切り換え中にデバイスがスリープモードへ移行した場合、そのクロック切り換えは中止されます。プロセッサは切り換え前のクロックで動作を継続し、OSWEN ビットはクリアされます。その後 PWRSAV 命令が通常通り実行されます。

スリープモードへ移行する前に内部 FRC オシレータへ切り換えておくと、スリープから高速にウェイクアップできます。

## 7.12 2 段階起動

クロック源選択レジスタ (FOSCSEL<7>) 内の内部 / 外部起動オプション コンフィグレーション ビット (IESO) の設定により、デバイス起動時にユーザ選択オシレータを使用するのか、それとも最初に内部 FRC オシレータで起動してからユーザ選択オシレータへ自動的に切り換えるのかを選択できます。このビットを「1」に設定すると、デバイスはクロック源の設定 (FOSCSEL<2:0>) に関係なく常に内部 FRC オシレータを使用して起動します。その後、指定されたオシレータが使用可能になると、自動的にそのオシレータへ切り換わります。

FSCM が有効でない場合、内部 FRC オシレータはクロック切り換え完了直後に自動的に停止します。2 段階起動オプションはデバイスの起動を高速化します。この動作は、オシレータ コンフィグレーション レジスタ (FOSC<7:6>) 内のクロック切り換えモードビット (FCKSM<1:0>) の状態に影響されません。

2 段階起動は、FNOSC<2:0> コンフィグレーション ビット (FOSCSEL<2:0>) で起動時間の長い水晶振動子タイプの外部オシレータを選択した場合に特に効果的です。内部 RC オシレータである FRC クロック源は、POR 後ほぼ即座に使用可能です。2 段階起動では、POR 後にデバイスはまず既定値のオシレータ設定 (FRC) を使用してコード実行を開始します。指定された外部クロック源が使用可能になると、クロック源は自動的に FRC オシレータから外部オシレータへ切り換わります。

ユーザコードで COSC<2:0> ビット (OSCCON<14:12>) と NOSC<2:0> ビット (OSCCON<10:8>) のステータスを比較する事により、デバイスが現在どちらのクロック源を使用して動作しているのかを確認できます。これら 2 つのビットが一致すれば、クロック切り換えが正常に完了し、デバイスは指定されたクロック源を使用して動作している事を意味します。

<b>Note:</b> クロック源に FRC を選択する場合、2 段階起動は冗長動作です。
--



7.13 レジスタマップ

表 7-8 に、オシレータ特殊機能制御レジスタにおけるビット割り当てを示します。表 7-9 に、オシレータ コンフィグレーション レジスタにおけるビット割り当てを示します。

表 7-8: オシレータ特殊機能制御レジスタ

レジスタ名	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	全リセット
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>				3040	
PLLFBD	—	—	—	—	—	—	—	PLLDIV<8:0>									0030
OSCTUN	—	—	—	—	—	—	—	—	—	—	TUN<5:0>						0000

- 凡例: x = リセット時に未知の値、— = 未実装、「0」として読み出し、リセット値は 16 進数で表記
- Note 1: OSCCON レジスタのリセット値は、FOSCSEL コンフィグレーション ビットの設定とリセットのタイプによって異なります。
- 2: 一部の dsPIC33F デバイスでは IOLOCK ビットを利用できません。詳細は各デバイスのデータシートを参照してください。

表 7-9: オシレータ コンフィグレーション レジスタ

レジスタ名	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>	

- 凡例: x = リセット時に未知の値、— = 未実装、「0」として読み出し、リセット値は 16 進数で表記
- Note 1: 一部の dsPIC33F デバイスでは IOL1WAY ビットを利用できません。詳細は各デバイスのデータシートを参照してください。

## 7.14 関連アプリケーション ノート

本セクションに関連するアプリケーション ノートの一覧を下に記載します。一部のアプリケーション ノートは dsPIC33F 製品ファミリ向けではありません。ただし概念は共通しており、変更が必要であったり制限事項が存在するものの利用が可能です。オシレータ モジュールに関連する現在のアプリケーション ノートは次の通りです。

タイトル	アプリケーション ノート番号
PIC <sup>®</sup> マイクロコントローラ オシレータ設計ガイド	AN588
PIC <sup>®</sup> マイクロコントローラによる低消費電力設計	AN606
水晶振動子の基礎と rfPIC <sup>®</sup> および PIC <sup>®</sup> デバイス向け水晶振動子の選択	AN826

<b>Note:</b> dsPIC33F デバイス ファミリ向けのその他のアプリケーション ノートとサンプルコードはマイクロチップ社のウェブサイト ( <a href="http://www.microchip.com">www.microchip.com</a> ) でご覧ください。
--

## 7.15 改訂履歴

### リビジョン A (2007 年 1 月)

本書の初版

### リビジョン B (2008 年 7 月)

改訂内容

- レジスタ
  - FOSCSEL: クロック源選択レジスタ (レジスタ 7-1): Bit 5 を下記に変更  
**予約:** 予約ビットは「1」として書き込む必要があります。
  - OSCTUN: FRC オシレータ調整レジスタ (レジスタ 7-6): Bit 5 を下記に変更  
**TUN<5:0>:** FRC オシレータ調整ビット  
 0111111 = 中心周波数 + 11.625% (8.23 MHz)  
 0111110 = 中心周波数 + 11.25% (8.20 MHz)
- 表
  - オシレータ特殊機能制御レジスタテーブル (表 7-9): Bit 5 を「未実装」に変更
- 表現および体裁の変更等、本書全体の細部を修正

### リビジョン C (2008 年 12 月)

改訂内容

- セクション 7.11 「クロック切り換え」の第 2 段落の後に、クロック切り換えを制御する位相ロックループ (PLL) イネーブル (PLLKEN) ビットの仕様に関する記述を追加
- セクション 7.11.2 「クロック切り換えシーケンス」に PLLKEN ビットに関する Note 5 を追加
- 表現および体裁の変更等、本書全体の細部を修正

### リビジョン D (2009 年 8 月)

改訂内容

- レジスタ
  - レジスタ 7-3 に Note 2 を追加
  - レジスタ 7-4 に Note 2 を追加
- 図
  - 図 7-1 を更新
  - 図 7-1 を更新
- 7.6 「内部高速 RC (FRC) オシレータ」に、FRC オシレータ調整 (TUN<5:0>) ビットに関する Note 2 を追加
- サンプルコード (例 7-1、例 7-2、例 7-3) を更新
- 7.3 「オシレータ コンフィギュレーション レジスタ」にレジスタに関する記述を追加
- 表現および体裁の変更等、本書全体の細部を修正

ISBN: 978-1-60932-503-9

NOTE: