

---

---

## 第 14 章 タイマ

---

---

### ハイライト

本章には次のようなトピックが含まれています。

14.1	はじめに .....	14-2
14.2	タイマの種類 .....	14-3
14.3	制御レジスタ .....	14-6
14.4	動作モード .....	14-9
14.5	タイマ プリスケアラ .....	14-14
14.6	タイマ割り込み .....	14-14
14.7	16 ビット タイマ モジュール レジスタの読み書き .....	14-15
14.8	2 次発振器の 32 kHz クリスタル入力 .....	14-15
14.9	32 ビット タイマの構成方法 .....	14-16
14.10	32 ビット タイマ モードの動作 .....	14-18
14.11	32 ビット タイマの読み書き .....	14-21
14.12	省電力状態でのタイマ動作 .....	14-21
14.13	タイマ モジュールを使用する周辺モジュール .....	14-22
14.14	レジスタ マップ .....	14-23
14.15	関連するアプリケーション ノート .....	14-24
14.16	改版履歴 .....	14-25

## 14.1 はじめに

個々に差異はありますが、PIC24F デバイス ファミリはいくつかの 16 ビット タイマを提供しています。それらのタイマはタイマ 1、タイマ 2、タイマ 3... というように区別されています。

どのタイマ モジュールも 16 ビットのタイマ / カウンタで、次のような読み書き可能なレジスタで構成されています。

- TMRx: 16 ビット タイマ カウント レジスタ
- PRx: 16 ビット タイマ周期レジスタ、タイマごとにある
- TxCON: 16 ビット タイマ制御レジスタ、タイマごとにある

どのタイマ モジュールにも割り込み制御に関連するビットがあります。

- 割り込み有効化制御ビット (TxIE)
- 割り込みフラグ ステータス ビット (TxIF)
- 割り込み優先制御ビット (TxIP<2:0>)

全ての 16 ビット タイマが同じ機能回路を持っていますが、いくつかの例外があります。16 ビット タイマは機能の差異により 3 タイプにグループ分けできます。

- タイプ A タイム ベース
- タイプ B タイム ベース
- タイプ C タイム ベース

いくつかの 16 ビットタイマは、結合して 32 ビットタイマにできます。

本章では、周辺モジュール デバイスと関連させたタイマは説明していません。例えば、これには入力キャプチャや出力コンペア モジュールと関連するタイム ベースが含まれません。

14.2 タイマの種類

PIC24F デバイスに実装されている 16 ビットタイマは機能的に同等ですが、いくつかの例外があります。16 ビットタイマは、タイプ A タイマ、タイプ B タイマ、タイプ C タイマの 3 種類の機能タイプに分けられます。

注：使用可能なタイマとそれらのタイプについては、その個別のデバイスのデータシートを参照して下さい。

14.2.1 タイプ A タイマ

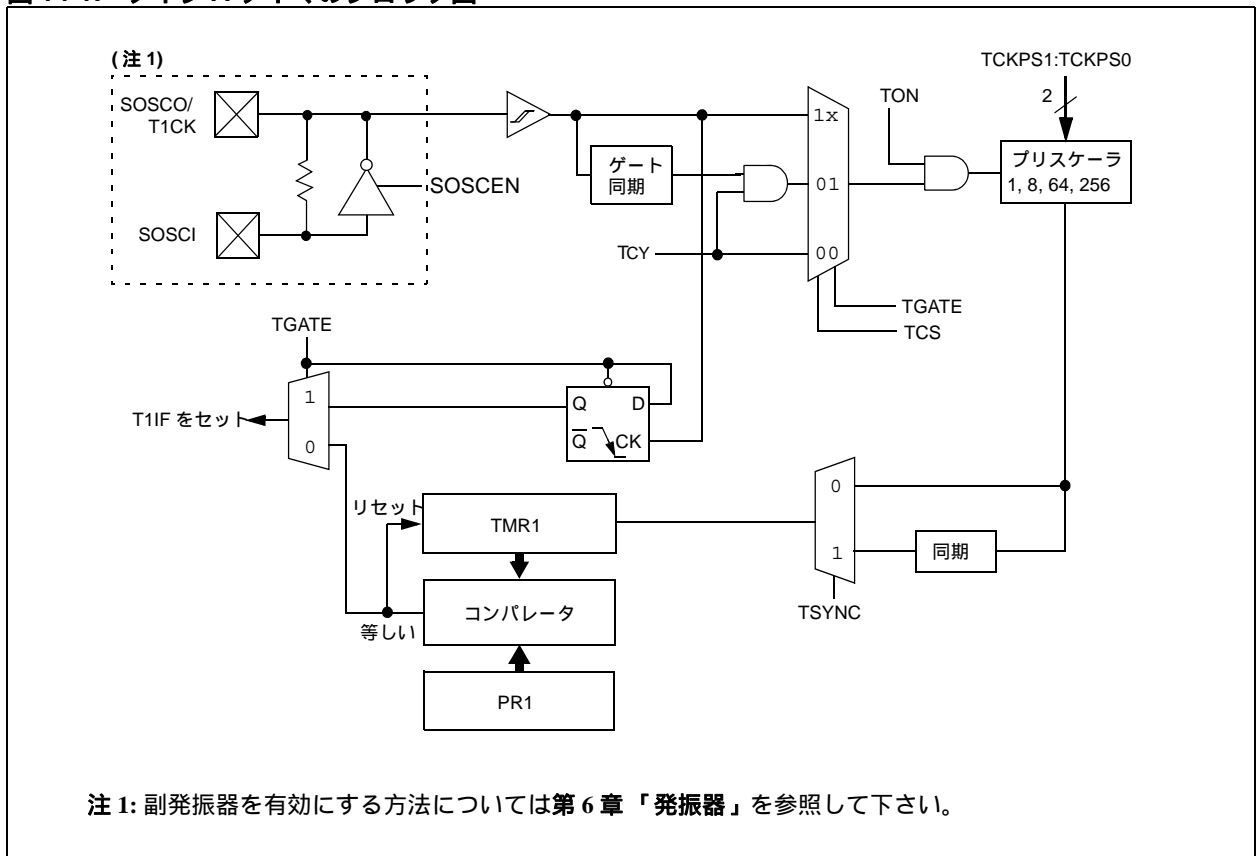
最低 1 個のタイプ A タイマがすべての PIC24F デバイスにあります。タイマ 1 はタイプ A タイマです。タイプ A タイマは他のタイプに対し次のような特徴があります。

- デバイスの低電力の 32kHz 発振器で動作が可能
- 外部クロック源で非同期モードの動作が可能

特に、タイプ A タイマの特徴は、時刻管理や、副システム クロック源として使われることです。

注：ほとんどの PIC24F デバイスは、外付けのハードウェアによる RTCC が必要ないように HW RTCC モジュールをもっています。

図 14-1: タイプ A タイマのブロック図



注 1: 副発振器を有効にする方法については第 6 章「発振器」を参照して下さい。

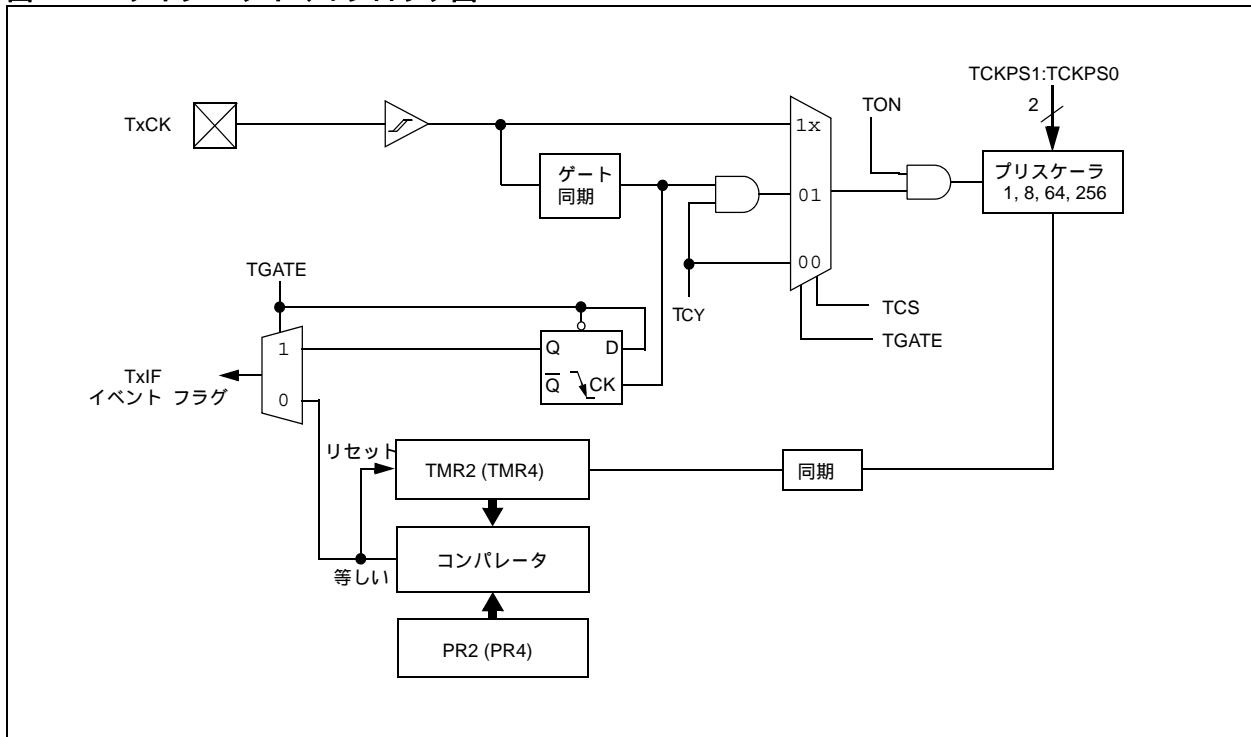
## 14.2.2 タイプ B タイマ

タイマ 2 とタイマ 4 がある場合には、PIC24F デバイスではタイプ B タイマとなります。タイプ B タイマは、他のタイプのタイマに対し次のような特徴があります。

- タイプ B タイマはタイプ C タイマと結合して 32 ビット タイマを構成できる。タイプ B タイマの TxCON レジスタには 32 ビット タイマ機能を有効化するための T32 制御ビットがある
- タイプ B タイマのクロック同期は、プリスケール ロジックの後で動作する。クロック同期がプリスケール ロジックの後にある利点については、14.4.4 項「高速外部クロック源によるタイマ動作」で説明している

タイプ B タイマのブロック図を図 14-2 に示します。

図 14-2: タイプ B タイマのブロック図



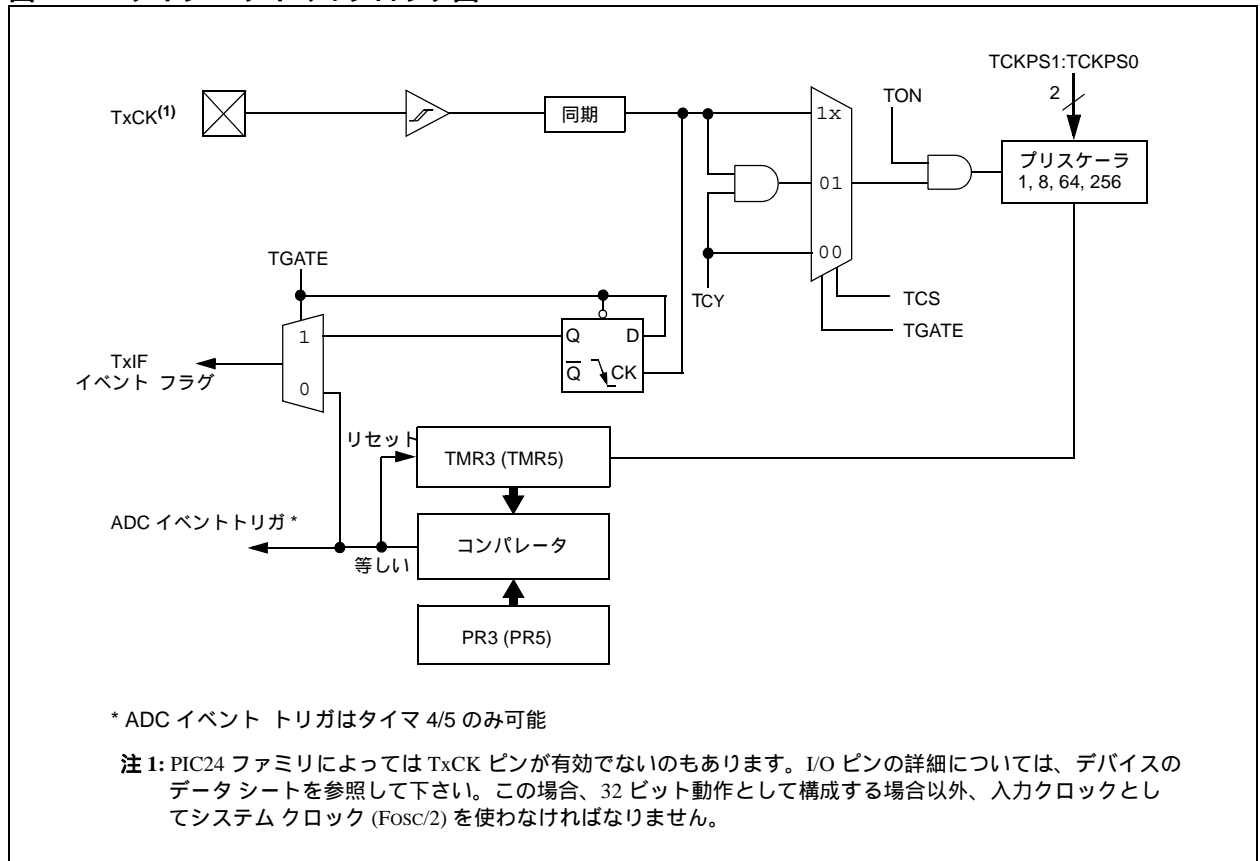
14.2.3 タイプ C タイマ

多くの場合、PIC24F デバイスのタイマ 3 とタイマ 5 はタイプ C タイマです。タイプ C タイマは他のタイマに対し次のような特徴があります。

- タイプ C タイマは、タイプ B タイマと結合して 32 ビット タイマとすることが可能
- どのデバイスも少なくとも 1 個のタイプ C タイマが A/D 変換トリガができる能力を持つ

タイプ C タイマのブロック図を図 14-3 に示します。

図 14-3: タイプ C タイマのブロック図



## 14.3 制御レジスタ

レジスタ 14-1: TxCON: タイプ A タイム ベース制御

R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
TON	—	TSIDL	—	—	—	—	—
ビット 15						ビット 8	

U-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	U-0
—	TGATE	TCKPS<1:0>		—	TSYNC	TCS	—
ビット 7						ビット 0	

<b>凡例:</b>			
R = 読み出し可	W = 書き込み可	U = 未実装、読むと「0」	
-n = POR 後の値	'1' = セット	'0' = クリア	x = 不定

- ビット 15    **TON:** タイマ x オン ビット  
 1 = タイマ スタート  
 0 = タイマ ストップ
- ビット 14    **未実装:** 読むと「0」
- ビット 13    **TSIDL:** アイドル モード中停止ビット  
 1 = デバイスがアイドル モードに入ったらタイマ動作停止  
 0 = アイドル モード中もタイマ動作継続
- ビット 12-7    **未実装:** 読むと「0」
- ビット 6    **TGATE:** タイマ x ゲート時間積算有効化ビット  
When TCS = 1:  
 このビットは無視される  
When TCS = 0:  
 1 = ゲート時間積算有効化  
 0 = ゲート時間積算無効化
- ビット 5-4    **TCKPS<1:0>:** タイマ x 入力クロック分周比選択ビット  
 11 = 1:256 分周比  
 10 = 1:64 分周比  
 01 = 1:8 分周比  
 00 = 1:1 分周比
- ビット 3    **未実装:** 読むと「0」
- ビット 2    **TSYNC:** タイマ x 外部クロック入力同期選択ビット  
When TCS = 1:  
 1 = 外部クロック入力を同期させる  
 0 = 外部クロック入力を同期させない  
When TCS = 0:  
 本ビットは無視、読むと「0」。タイマ x は TCS=0 のときは内部クロックを使用
- ビット 1    **TCS:** タイマ x クロック源選択ビット  
 1 = TxCK ピンからの外部クロック  
 0 = 内部クロック (FOSC/2)
- ビット 0    **未実装:** 読むと「0」

## レジスタ 14-2: TxCON: タイプ B タイム ベース制御

R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
TON	—	TSIDL	—	—	—	—	—
ビット 15						ビット 8	
U-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	U-0
—	TGATE	TCKPS<1:0>		T32	—	TCS	—
ビット 7						ビット 0	

### 凡例:

R = 読み出し可	W = 書き込み可	U = 未実装、読むと「0」
-n = POR 後の値	'1' = セット	'0' = クリア
		x = 不定

- ビット 15    **TON:** タイマ x オンビット  
T32 = 1 の場合 (32 ビットタイマ モード中)  
 1 = 32 ビット TMRx : TMRy タイマ ペア スタート  
 0 = 32 ビット TMRx : TMRy タイマ ペア ストップ  
T32 = 0 の場合 (16 ビットタイマ モード中)  
 1 = 16 ビット タイマ スタート  
 0 = 16 ビット タイマ ストップ
- ビット 14    **未実装:** 読むと「0」
- ビット 13    **TSIDL:** アイドル モード中の停止ビット  
 1 = デバイスがアイドル モードに入ったらタイマ動作停止  
 0 = アイドル モード中もタイマ動作継続
- ビット 12-7    **未実装:** 読むと「0」
- ビット 6    **TGATE:** タイマ x ゲート時間積算有効化ビット  
TCS = 1 の場合  
     本ビットは無視  
TCS = 0 の場合  
 1 = ゲート時間積算有効化  
 0 = ゲート時間積算無効化
- ビット 5-4    **TCKPS<1:0>:** タイマ x 入力クロック分周比選択ビット  
 11 = 1:256 分周比  
 10 = 1:64 分周比  
 01 = 1:8 分周比  
 00 = 1:1 分周比
- ビット 3    **T32:** 32 ビット タイマ x モード選択ビット  
 1 = TMRx と TMRy で 32 ビット タイマを構成する  
 0 = TMRx と TMRy を別々の 16 ビット タイマとする
- ビット 2    **未実装:** 読むと「0」
- ビット 1    **TCS:** タイマ x クロック源選択ビット  
 1 = TxCK ピンからの外部クロック  
 0 = 内部クロック (Fosc/2)
- ビット 0    **未実装:** 読むと「0」

# PIC24F ファミリ リファレンス マニュアル

## レジスタ 14-3: TyCON: タイプ C タイム ベース制御

R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
TON(1)	—	TSIDL	—	—	—	—	—
ビット 15						ビット 8	

U-0	R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	U-0
—	TGATE(1)	TCKPS<1:0>		—	—	TCS	—
ビット 7						ビット 0	

### 凡例:

R = 読み出し可                      W = 書き込み可                      U = 未実装、読むと「0」  
 -n = POR 後の値                      '1' = セット                      '0' = クリア                      x = 不定

- ビット 15     **TON: タイマ<sub>y</sub> オンビット<sup>(1)</sup>**  
 1 = 16 ビットタイマ<sub>y</sub> スタート  
 0 = 16 ビットタイマ<sub>y</sub> ストップ
- ビット 14     **未実装: 読むと「0」**
- ビット 13     **TSIDL: アイドル モード中の停止ビット**  
 1 = デバイスがアイドル モードに入ったらタイマ動作停止  
 0 = アイドル モード中もタイマ動作継続
- ビット 12-7   **未実装: 読むと「0」**
- ビット 6     **TGATE: タイマ<sub>y</sub> ゲート時間積算有効化ビット<sup>(1)</sup>**  
TCS = 1 の場合  
 本ビットは無視  
TCS = 0 の場合  
 1 = ゲート時間積算有効化  
 0 = ゲート時間積算無効化
- ビット 5-4   **TCKPS<1:0>: タイマ<sub>y</sub> 入力クロック分周比選択ビット**  
 11 = 1:256 分周比  
 10 = 1:64 分周比  
 01 = 1:8 分周比  
 00 = 1:1 分周比
- ビット 3-2   **未実装: 読むと「0」**
- ビット 1     **TCS: タイマ<sub>y</sub> クロック選択ビット**  
 1 = TxCK ピンの外部クロック  
 0 = 内部クロック (Fosc/2)
- ビット 0     **未実装: 読むと「0」**

注 1: 32 ビット動作を有効にした場合 (T2CON<3> = 1)、これらのビットはタイマ<sub>y</sub> の動作に影響しない。すべてのタイマ機能は T2CON を介して設定される。



## 14.4 動作モード

各タイマ モジュールは次のモードのいずれかで動作する。

- タイマ
- 同期カウンタ
- ゲート付きタイマ
- 非同期カウンタ (タイプ A とタイプ C タイム ベースのみ)

タイマ モードは次のビットで決まる。

- TCS (TxCON<1>): タイマ クロック源制御ビット
- TSYNC (TxCON<2>): タイマ同期制御ビット (タイプ A タイム ベースのみ)
- TGATE (TxCON<6>): タイマ ゲート制御ビット

各タイマ モジュールは、TON ビット (TxCON <15>) により有効化、無効化されます。

注: タイプ A と C タイム ベースのみが、外部非同期カウンタ モードをサポートしています。

## 14.4.1 タイマ モード

すべてのタイプのタイマが、システム クロックによるタイマ モードで動作できます。タイマ モードでは、タイマへの入力クロックは、内部システムクロック (FOSC/2) から供給されます。これが有効化され、分周比が 1:1 に設定されると、命令サイクルごとに 1 度インクリメントします。タイマ モードは、TCS 制御ビット (TxCON<1>) をクリアすると選択されます。同期モード制御ビット TSYNC (TxCON<2>) は、システム クロック源がタイマ クロックを生成していますから何の影響も与えません。

## 例 14-1: システム クロックを使った 16 ビットタイマの初期化コード例

```

/* The following code example will enable Timer1 interrupts, load the Timer1
   Period register and start Timer1.

   When a Timer1 period match interrupt occurs, the interrupt service
   routine must clear the Timer1 interrupt status flag in software.
*/

T1CON = 0x00;           //Stops the Timer1 and reset control reg.
TMR1 = 0x00;           //Clear contents of the timer register
PR1 = 0xFFFF;         //Load the Period register with the value 0xFFFF
IPC0bits.T1IP = 0x01; //Setup Timer1 interrupt for desired priority level
                       // (This example assigns level 1 priority)
IFS0bits.T1IF = 0;     //Clear the Timer1 interrupt status flag
IEC0bits.T1IE = 1;    //Enable Timer1 interrupts
T1CONbits.TON = 1;    //Start Timer1 with prescaler settings at 1:1 and
                       //clock source set to the internal instruction cycle

/* Example code for Timer1 ISR*/

void __attribute__((__interrupt__, __shadow__)) _T1Interrupt(void)
{
    /* Interrupt Service Routine code goes here          */

    IFS0bits.T1IF = 0; //Reset Timer1 interrupt flag and Return from ISR
}

```

## 14.4.2 外部クロック入力を使った同期カウンタ モード

TCS 制御ビット (TxCON<1>) がセットされると、タイマへのクロック源が外部供給となり、そのタイマは TxCK ピンのクロック入力の立ち上がりエッジ毎にインクリメントします。

タイプ A タイム ベースは、同期カウンタ モードで走らせるときには、外部クロック同期を有効にしなければなりません。これは、TSYNC 制御ビット (TxCON<2>) をセットすることで達成できます。タイプ B とタイプ C タイマの場合は、外部クロック入力は常にシステムの命令サイクルクロック Tcy に同期させられます。

タイマが同期カウンタ モードで動作中は、外部クロックの High と Low 時間の最小値が既定されます。外部クロック源とデバイスの命令クロックとの同期は、外部クロック信号が 1 命令サイクル内で、2 つの異なる時間にサンプリングされることで達成されます。

同期外部クロック源のタイマ動作は、スリープ モード中は動作しません。これはスリープ モード中は同期回路が停止するためです。

注：同期カウンタ モードで動作させる場合には、外部入力クロックは最小の High 時間と Low 時間の要求仕様を満たさなければなりません。

### 例 14-2: 外部クロック入力を使った 16 ビット同期カウンタ モードの初期化コード例

```
/* The following code example will enable Timer1 interrupts, load the
Timer1 Period register and start Timer1 using an external clock
and a 1:8 prescaler setting.

When a Timer1 period match interrupt occurs, the interrupt service
routine must clear the Timer1 interrupt status flag in software.
*/
T1CON = 0x00;           //Stops the Timer1 and reset control reg.
TMR1 = 0x00;           //Clear contents of the timer register
PR1 = 0x8CFF;          //Load the Period register with the value 0x8CFF
IPC0bits.T1IP = 0x01;  //Setup Timer1 interrupt for desired priority level
                        // (this example assigns level 1 priority)
IFS0bits.T1IF = 0;     //Clear the Timer1 interrupt status flag
IEC0bits.T1IE = 1;     //Enable Timer1 interrupts
T1CON = 0x8016;        //Start Timer1 with prescaler settings at 1:8 and
                        //clock source set to the external clock in the
                        //synchronous mode

/* Example code for Timer1 ISR*/

void __attribute__((__interrupt__, __shadow__)) _T1Interrupt(void)
{
    /* Interrupt Service Routine code goes here          */

    IFS0bits.T1IF = 0;  //Reset Timer1 interrupt flag and Return from ISR
}
```

## 14.4.3 タイプ A タイマの外部クロック入力を使った非同期カウンタ モード

タイプ A タイム ベースは、TxCK ピンに接続された外部クロック源を使った非同期カウンタモードの動作ができます。TSYNC 制御ビット (TxCON<2>) をクリアすると、外部クロック入力はデバイスシステムクロック源と同期されません。タイマは内部デバイスクロックに非同期にインクリメントし続けます。

非同期動作のタイムベースは次のようなアプリケーションで有益です。

- スリープモード中もタイマを動作させ、周期レジスタの一致で割り込みを発生して、プロセッサをウェイクアップさせる場合
- タイムベースを副システムクロック源として使う省電力 32kHz で動作させられる場合

**注 1:** タイプ A タイムベースのみが非同期カウンタモードをサポートしています。

**注 2:** 外部クロック入力は、タイマ x を非同期カウンタモードで使う場合には、最小の High 時間と Low 時間の要求仕様を満たす必要があります。

### 例 14-3: 外部クロック入力を使った 16 ビット非同期カウンタモード初期化コード例

```

/* The following code example will enable Timer1 interrupts, load the Timer1
   Period register and start Timer1 using an asynchronous external clock and
   a 1:8 prescaler setting.

   When a Timer1 period match interrupt occurs, the interrupt service
   routine must clear the Timer1 interrupt status flag in software.
*/

T1CON = 0x00;           //Stops the Timer1 and reset control reg.
TMR1 = 0x00;           //Clear contents of the timer register
PR1 = 0x8CFF;          //Load the Period register with the value 0x8CFF
IPC0bits.T1IP = 0x01;  //Setup Timer1 interrupt for desired priority level
                       // (this example assigns level 1 priority)
IFS0bits.T1IF = 0;     //Clear the Timer1 interrupt status flag
IEC0bits.T1IE = 1;     //Enable Timer1 interrupts
T1CON = 0x8012;        //Start Timer1 with prescaler settings at 1:8 and
                       //clock source set to the external clock in the
                       //asynchronous mode

/* Example code for Timer1 ISR*/

void __attribute__((__interrupt__, __shadow__)) _T1Interrupt(void)
{
    /* Interrupt Service Routine code goes here */

    IFS0bits.T1IF = 0;  //Reset Timer1 interrupt flag and Return from ISR
}

```

## 14.4.4 高速外部クロック源によるタイマ動作

アプリケーションによっては、いずれかのタイマをより高速の外部クロック源のクロックエッジをカウント動作させたい場合があります。外部クロック源でカウントさせるにはタイプ A かタイプ B のタイム ベースが適しています。なぜなら、これらのタイマのクロック同期ロジックが、タイマ プリスケアラの後にあるためです (図 14-1 と図 14-2 を参照)。これにより、使用されるより高い周波数の外部クロックでも、プリスケアラに必要な最小の High と Low 時間の仕様に違反しないようにできます。タイプ A またはタイプ B のタイム ベースのタイマ プリスケアラ比が 1:1 でない場合は、外部クロック入力に必要な最小の High と Low 時間が、選択したプリスケアラ比で低減されます。

タイプ A タイム ベースは特別で、プリスケアラのタイミング要件に関わらずに非同期モードで動作させられます。

すべての場合において、外部クロック信号が超えてはならない最小の High と Low 時間があることに注意して下さい。これらの最小時間は、I/O ピンの要求タイミングを満たすために必須です。

タイム ベースに関連する外部クロック タイミング仕様については、個別のデバイスのデータシートを参照して下さい。

## 14.4.5 ゲート時間積算モード

ゲート時間積算モードでは、TxCK ピンの入力信号が High の間だけ内部タイマ レジスタのインクリメントを行います。ゲート時間積算モードでは、タイマのクロック源は内部システム クロックで駆動されます。TxCK ピンの状態が High のとき、タイマ レジスタは周期一致が発生するか、または TxCK ピン状態が Low 状態に変化するまでカウント アップします。ピンの High から Low への遷移により TxIF 割り込みフラグがセットされます。いつエッジが発生したかに依存して、TxCK ピンの信号の立下りエッジのあとに 1 命令サイクルあるいは 2 命令サイクルで割り込みフラグがセットされます。

ゲート時間積算モードを有効化するためには、TGATE 制御ビット (TxCON<6>) をセットしなければなりません。タイマを有効化 (TON (TxCON<15>) = 1) し、タイマクロック源を内部クロックにセット (TCS (TxCON<1>) = 0) しなければなりません。

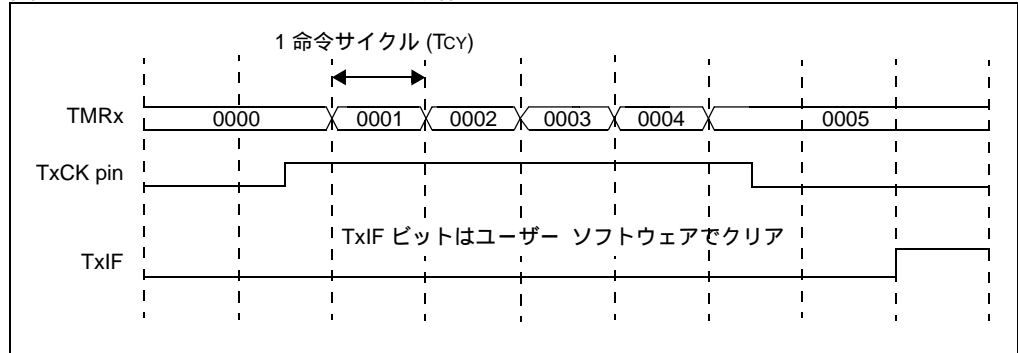
ゲート動作は、TxCK ピンに入力された信号の立ち上がりエッジで開始され、TxCK ピンに入力された信号の立下りエッジで終了します。対応するタイマが外部ゲート信号が High の間だけインクリメントされます。

ゲート信号の立下りエッジで TxIF 割り込みフラグがセットされ、有効ならば割り込みが発生します。

**注：**ゲート時間積算モードでは、タイマの周期一致の発生では、タイマは CPU へ割り込みません。

タイマ カウンタの分解能は、直接タイマ クロック周期に関係します。タイマのプリスケール値が 1:1 ならば、タイマクロック周期は 1 命令サイクルとなります。タイマ プリスケール値が 1:256 ならば、タイマクロック周期は、256 命令サイクルとなります。タイマクロック分解能は、ゲートする信号のパルス幅に関連してきます。ゲートパルス幅の要求についての詳細は、その個別のデバイスのデータシートの「電気的特性」の項を参照して下さい。

図 14-4: ゲート タイマ モードの動作



例 14-4: 16 ビット ゲート時間積算モード用の初期化コード

```

/* The following code example will enable Timer2 interrupts, load the Timer2
Period register and start Timer2 using an internal clock and an external
gate signal. On the falling edge of the gate signal a Timer2 interrupt
occurs. The interrupt service routine must clear the Timer2 interrupt
status flag in software .
*/
*/
T2CON = 0x00;           //Stops the Timer2 and reset control reg.
TMR2 = 0x00;           //Clear contents of the timer register
PR2 = 0xFFFF;         //Load the Period register with the value 0xFFFF
IPC1bits.T2IP = 0x01; //Setup Timer2 interrupt for desired priority level
                        // (this example assigns level 1 priority)
IFS0bits.T2IF = 0;    //Clear the Timer2 interrupt status flag
IEC0bits.T2IE = 1;    //Enable Timer2 interrupts
T2CONbits.TGATE = 1;  //Set up Timer2 for operation in Gated
                        //Time Accumulation mode
T2CONbits.TON = 1;    //Start Timer2

void __attribute__((__interrupt__, __shadow__)) _T2Interrupt(void)
{
    /* Interrupt Service Routine code goes here          */
    IFS0bits.T2IF = 0;  //Reset Timer2 interrupt flag and Return from ISR
}

```

## 14.5 タイマ プリスケーラ

すべての 16 ビットタイマへの入力クロック ( $F_{osc}/2$  または外部クロック) は、1:1、1:8、1:64、1:256 のプリスケーラを選択肢を持っています。クロック プリスケーラ比は、 $TCKPS\langle 1:0 \rangle$  制御ビット ( $TxCON\langle 5:4 \rangle$ ) で選択されます。プリスケーラ カウンタは、次の場合にクリアされます。

- TMRx レジスタへの書き込み
- TON ( $TxCON\langle 15 \rangle$ ) が「0」にクリアされた場合
- いずれかのデバイス リセット

注: TMRx レジスタは、 $TxCON$  への書き込みではクリアされません。

## 14.6 タイマ割り込み

16 ビット タイマは、動作モードにより周期が一致したときか、または外部ゲート信号の立下りで割り込みを生成する能力を持っています。

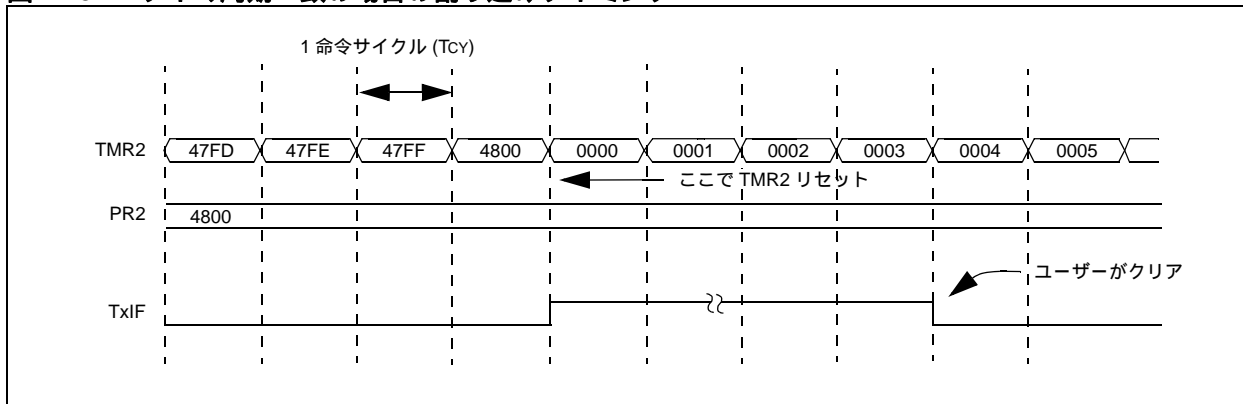
下記条件のうちの一つが成立すると  $TxIF$  ビットがセットされます。

- タイマのカウンタが対応する周期レジスタと一致し、かつタイマ モジュールがゲート時間積算モードで動作していない
- タイマがゲート時間積算モードで動作している場合に「ゲート」信号の立下りエッジが検出された  $TxIF$  ビットはソフトウェアでクリアして下さい。

タイマは対応するタイマ割り込み有効化ビット  $TxIE$  により割り込み源にできます。さらに、タイマを割り込み源とするには、割り込み優先レベルビット ( $TxIP\langle 2:0 \rangle$ ) にゼロでない値を書き込まなければなりません。詳細は第 8 章「割り込み」を参照して下さい。

注: 周期レジスタ  $PRx$  に  $0x0000$  がロードされていてタイマが有効な場合は特別で、この構成ではタイマ割り込みは発生しません。

図 14-5: タイマ周期一致の場合の割り込みタイミング



## 14.7 16 ビット タイマ モジュール レジスタの読み書き

- タイマ モジュールのすべての SFR はバイト (8 ビット) またはワード (16 ビット) で書き込みが可能
- タイマ モジュールのすべての SFR はワード (16 ビット) でしか読み出せない

### 14.7.1 16 ビット タイマへの書き込み

タイマと対応する周期レジスタは、モジュールが動作中でも書き込めます。バイトで書き込む場合には次のことに留意して下さい。

- タイマがインクリメント中にタイマの下位バイトに書き込むときは、上位バイトには影響を与えない。タイマの下位バイトに 0xFF を書き込むと、書き込み後の次のタイマ カウント クロックで下位バイトが 0x00 へロールオーバーし、タイマの上位バイトへの桁上げを生じる
- タイマがインクリメント中に上位バイトに書き込むと、下位バイトには影響を与えない。しかしタイマ下位バイトが 0xFF のときに書き込むと、次のタイマ カウント クロックで下位バイトから桁上げを生じ、タイマの上位バイトのインクリメント要因となる

TMRx レジスタが命令で書き込まれると (ワードでもバイトでも)、TMRx レジスタのインクリメントがマスクされ、その命令サイクルではインクリメントされません。

非同期モードでタイマに書き込むのは、実時間動作のアプリケーションでは避けて下さい。非同期カウンタモードについての情報は 14.4.1 項「タイマモード」を参照して下さい。

### 14.7.2 16 ビット タイマの読み出し

タイマおよび関連 SFR のすべての読み出しは、ワード読み出し (16 ビット) として下さい。バイト読み出しは無効となります (「0」が返されます)。

タイマと対応する周期レジスタは、モジュールが動作中でも読み出せます。TMRx レジスタの読み出し動作は、同じ命令サイクルでのタイマのインクリメント動作を妨げません。

## 14.8 2 次発振器の 32 kHz クリスタル入力

各デバイスのオプションとしてタイプ A タイマ モジュールには、リアルタイム クロック (RTC) アプリケーション用に 32 kHz クリスタル発振器が接続できます。

- 2 次発振器が有効化され、タイマが外部クロック源を使うように構成されていれば、2 次発振器がタイマのクロック源となります。
- 2 次発振器は OSCCON レジスタの SOSCEN 制御ビットをセットすれば有効化される
- 32 kHz クリスタル振動子を SOSCO/SOSCI デバイスピンを接続する

詳細は第 6 章「発振器」を参照して下さい。

## 14.9 32 ビット タイマの構成方法

32 ビット タイマ モジュールはタイプ B とタイプ C の 16 ビット タイマ モジュールを結合して構成します。タイプ C タイム ベースが結合タイマの上位ワード (msw) となり、タイプ B タイム ベースが下位ワード (lsw) 側となります。

32 ビット動作に構成されると、タイプ B タイム ベースの制御ビットが 32 ビット タイマの制御をします。タイプ C タイム ベースの TxCON レジスタ制御ビットは影響を与えません。

結合した 32 ビット タイマの割り込み制御には、タイプ C タイム ベースの割り込み許可ビット、割り込みフラグ、割り込み優先制御ビットを使用します。タイプ B タイム ベースの割り込み制御とステータス ビットは、32 ビットタイマ動作には使用されません。

**注：** 結合可能なタイプ B とタイプ C タイム ベースについての情報は、個別のデバイスのデータシートを参照して下さい。

次の構成設定例ではタイプ C タイム ベースにはタイマ 3 を、タイプ B タイム ベースにはタイマ 2 を仮定しています。

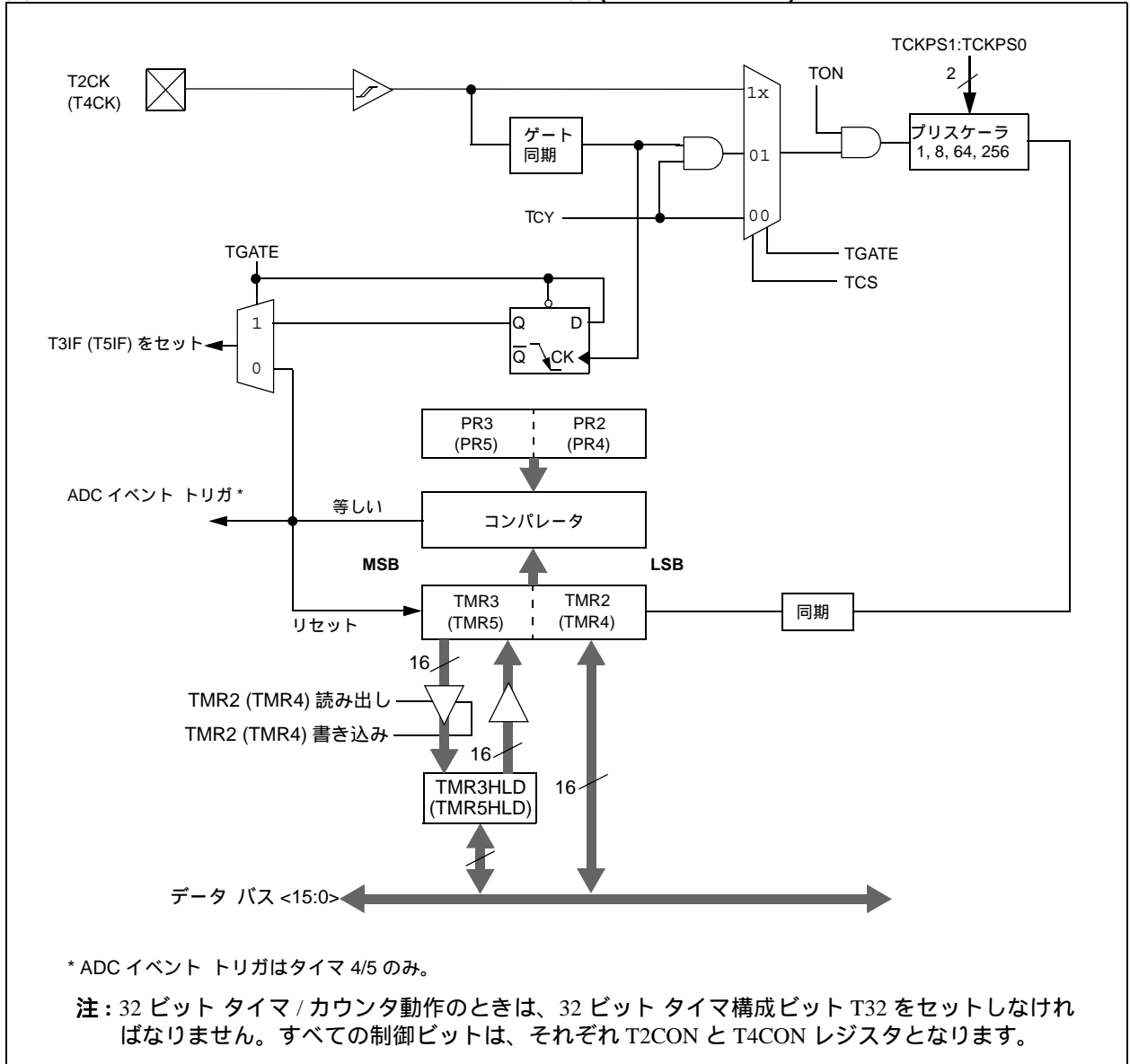
- TON (T2CON<15>) = 1
- T32 (T2CON<3>) = 1
- TCKPS<1:0> ビット (T2CON<5:4>) は、タイマ 2(タイプ B タイム ベース)用のプリスケラ モードの設定に使用される
- TMR3 : TMR2 レジスタ ペアには、タイマ モジュールの 32 ビット値を保持する。  
TMR3(タイプ C タイム ベース) レジスタが 32 ビットタイマ値の上位ワードとなり、TMR2(タイプ B タイム ベース) が下位ワードとなる
- PR3 : PR2 レジスタ ペアが TMR3:TMR2 タイマ値との比較に使用する 32 ビットの周期値を保持する
- T3IE (IEC0<8>) がこの構成での 32 ビットタイマの割り込みを有効化するために使用される
- T3IF (IFS0<8>) がタイマ割り込みのステータス フラグとして使用される
- T3IP<2:0> ビット (IPC2<2:0>) が 32 ビット タイマの割り込み優先レベルを設定する
- T3CON<15:0> は無関係のビットとなる

例として、タイマ 2 とタイマ 3 を使った 32 ビットタイマ モジュールのブロック図を図 14-6 に示します。

**注：** 入力キャプチャと出力コンペアでは、32 ビットタイマ モードにおいては使用できません。



図 14-6: タイプ B/ タイプ C タイマ ペア のブロック図 (32 ビット タイマ)



## 14.10 32 ビット タイマ モードの動作

### 14.10.1 タイマ モード

例 14-5 にタイマ モードにおいて 32 ビットタイマに構成する方法を示します。この例では、タイマ 2 がタイプ B タイム ベース、タイマ 3 がタイプ C タイム ベースと仮定しています。32 ビットタイマ動作とするには、T2CON レジスタ (タイプ B タイム ベース) 内の T32 制御ビットをセットして下さい。タイマ 2 とタイマ 3 が 32 ビット タイマとして構成されると、T3CON 制御ビットは無視されます。T2CON 制御ビットのみが、セットアップと制御に使用されます。タイマ 2 のクロックとゲート入力、32 ビット タイマ モジュール用に必要とされますが、割り込みは T3IF フラグで生成されます。タイマ 2 が 32 ビットタイマの lsw でタイマ 3 が msw です。TMR3 は TMR2 のオーバーフロー (桁上げ) でインクリメントされます。32 ビットタイマは PR2 と PR3 が結合した 32 ビット周期レジスタにあらかじめロードされた値と一致するまでインクリメントされ、その後ロールオーバーして継続します。PR3:PR2 に 0xFFFFFFFF がロードされたとき、32 ビットタイマ の最大カウンタ値となります。周期一致で、許可されていれば割り込みを発生します。

#### 例 14-5: 命令サイクルを入力クロックとして使用する 32 ビットタイマの初期化コード

```
/* The following code example will enable Timer3 interrupts, load the
Timer3:Timer2 Period Register and start the 32-bit timer module
consisting of Timer3 and Timer2.

When a 32-bit period match interrupt occurs, the user must clear the
Timer3 interrupt status flag in software.
*/
*/
T2CON = 0x00;          //Stops any 16/32-bit Timer2 operation
T3CON = 0x00;          //Stops any 16-bit Timer3 operation
TMR3 = 0x00;           //Clear contents of the timer3 register
TMR2 = 0x00;           //Clear contents of the timer2 register
PR3 = 0xFFFF;         //Load the Period register3 with the value 0xFFFF
PR2 = 0xFFFF;         //Load the Period register2 with the value 0xFFFF

IPC2bits.T3IP = 0x01; //Setup Timer3 interrupt for desired priority level
                        //(this example assigns level 1 priority)
IFS0bits.T3IF = 0;    //Clear the Timer3 interrupt status flag
IEC0bits.T3IE = 1;    //Enable Timer3 interrupts
T2CONbits.T32 = 1;    //Enable 32-bit Timer operation
T2CONbits.TON = 1;    //Start 32-bit timer with prescaler
                        //settings at 1:1 and clock source set to
                        //the internal instruction cycle

void __attribute__((__interrupt__, __shadow__)) _T3Interrupt(void)
{
    /* Interrupt Service Routine code goes here          */

    IFS0bits.T3IF = 0; //Reset Timer1 interrupt flag and Return from ISR
}
}
```

## 14.10.2 同期カウンタ モード

同期カウンタ モードのときは 32 ビットタイマは 16 ビットタイマと同様に動作します。例 14-6 に 32 ビットタイマを同期カウンタ モードとして構成する方法を示します。この例では、タイマ 2 がタイプ B タイム ベースで、タイマ 3 がタイプ C タイム ベースとみなしています。

## 例 14-6: 外部クロック入力を使用した 32 ビット同期カウンタ モードの初期化コード

```

/* The following code example will enable Timer2 interrupts, load the
   Timer3:Timer2 Period register and start the 32-bit timer module
   consisting of Timer3 and Timer2.

   When a 32-bit period match interrupt occurs, the user must clear the
   Timer3 interrupt status flag in the software.
*/
*/
T2CON = 0x00;           //Stops any 16/32-bit Timer2 operation
T3CON = 0x00;           //Stops any 16-bit Timer3 operation
TMR3 = 0x00;           //Clear contents of the timer3 register
TMR2 = 0x00;           //Clear contents of the timer2 register
PR3 = 0xFFFF;         //Load the Period register3 with the value 0xFFFF
PR2 = 0xFFFF;         //Load the Period register2 with the value 0xFFFF

IPC2bits.T3IP = 0x01;  //Setup Timer3 interrupt for desired priority level
                        //(this example assigns level 1 priority)
IFS0bits.T3IF = 0;    //Clear the Timer3 interrupt status flag
IEC0bits.T3IE = 1;    //Enable Timer3 interrupts
T2CON = 0x801A;       //Enable 32-bit Timer operation and start
                        //32-bit timer with prescaler settings at 1:8
                        //and clock source set to external clock

void __attribute__((__interrupt__, __shadow__)) _T3Interrupt(void)
{
    /* Interrupt Service Routine code goes here          */

    IFS0bits.T3IF = 0; //Reset Timer1 interrupt flag and Return from ISR
}

```

## 14.10.3 非同期 カウンタ モード

タイプ B、タイプ C タイマいずれも非同期の外部クロックモードをサポートしていません、したがって 32 ビット非同期カウンタ モードはサポートされていません。

## 14.10.4 ゲート時間 積算モード

ゲート時間積算モードでは、32 ビットタイマの動作は 16 ビット タイマと同じです。例 14-7 にゲート時間積算モードにおいて、32 ビットタイマを構成する方法を示します。この例では、タイマ 2 がタイプ B タイム ベースで、タイマ 3 がタイプ C タイム ベースとしてしています。

### 例 14-7: 32 ビットゲート時間積算モードの初期化 コード

```
/* The following code example will enable Timer2 interrupts, load the
   Timer3:Timer2 Period register and start the 32-bit timer module
   consisting of Timer3 and Timer2.
   When a 32-bit period match occurs the timer will simply roll over and
   continue counting.

   However, when at the falling edge of the Gate signal on T2CK an interrupt
   is generated, if enabled. The user must clear the Timer3 interrupt status
   flag in the software.
*/

T2CON = 0x00;           //Stops any 16/32-bit Timer2 operation
T3CON = 0x00;           //Stops any 16-bit Timer3 operation
TMR3 = 0x00;           //Clear contents of the timer3 register
TMR2 = 0x00;           //Clear contents of the timer2 register
PR3 = 0xFFFF;         //Load the Period register3 with the value 0xFFFF
PR2 = 0xFFFF;         //Load the Period register2 with the value 0xFFFF

IPC2bits.T3IP = 0x01;  //Setup Timer3 interrupt for desired priority level
                        //(this example assigns level 1 priority)
IFS0bits.T3IF = 0;    //Clear the Timer3 interrupt status flag
IEC0bits.T3IE = 1;    //Enable Timer3 interrupts
T2CON = 0x8048;       //Enable 32-bit Timer operation and
                        //Start 32-bit timer in gated time accumulation mode.

void __attribute__((__interrupt__, __shadow__)) _T3Interrupt(void)
{
    /* Interrupt Service Routine code goes here          */

    IFS0bits.T3IF = 0;  //Reset Timer1 interrupt flag and Return from ISR
}
```

## 14.11 32 ビット タイマの読み書き

32 ビットタイマの lsw と msw の間で、32 ビット読み書き動作としての同期を取るためには、追加された制御ロジックと保持レジスタを使用します (図 14-6 参照)。タイプ C タイムベースは TMRxHLD と呼ばれるレジスタを持っていて、タイマの レジスタ ペアを読み書きするときに使用されます。TMRxHLD レジスタは、対応するタイマが 32 ビット動作に構成されたときだけ使用されます。

TMR3:TMR2 が 32 ビットタイマ ペアを構成しているとする、まず TMR2 レジスタからタイマ値の lsw を読み出します。lsw を読み出すと自動的に TMR3 の内容が TMR3HLD レジスタに転送されます。この後 TMR3HLD を読み出せばタイマ値の msw が取り出せます。これを例 14-8 に示します。

## 例 14-8: 32 ビット タイマからの読み出し

```

/* The following code segment reads the 32-bit timer formed by the
   Timer3-Timer2 pair into the registers W1(MS Word) and W0(LS Word).
*/
unsigned int temp_lsb;
unsigned int temp_msb;

temp_lsb = TMR2;          //Transfer the LSW into temp_lsb
temp_msb = TMR3HLD;      //Transfer the MSW from the holding register to into
                          //temp_msb

```

TMR3:TMR2 レジスタ ペアに値を書き込むためには、まず TMR3HLD レジスタに msw を書き込みます。TMR2 にタイマ値の lsw を書き込むとき、TMR3HLD の内容が TMR3 レジスタに自動的に転送されます。

## 14.12 省電力状態でのタイマ動作

## 14.12.1 スリープモード中のタイマ動作

デバイスがスリープモードになると、システムクロックが停止します。タイマが内部クロック源 (Fosc/2) で動作していると、これも停止します。

タイプ A タイムベースは他のタイマモジュールと異なり、システムクロック源とは非同期に動作させられます。この違いにより、タイプ A タイムベースモジュールは、スリープモード中でも動作継続できます。スリープモード中でも動作させるには、タイプ A タイムベースを次のように構成しなければなりません。

- タイマ 1 モジュールを有効にする、TON (TxCON<15>) = 1
- タイマ 1 のクロック源として外部を選択、TCS (TxCON<1>) = 1 かつ
- TSYNC ビット (TxCON<2>) を論理「0」とする (非同期カウンタモードを有効にする)

**注:** 非同期カウンタモードはタイマ 1 モジュールだけでサポートされています。

上記条件がすべて満足されると、タイマ 1 はデバイスがスリープモード中にカウントと周期一致検出動作を継続します。タイマと周期レジスタが一致すると、TxIF ビットがセットされ、オプションで、デバイスをスリープからウェイクアップさせる割り込みを生成できます。詳細は第 10 章「省電力機能」を参照して下さい。

## 14.12.2 アイドルモード中のタイマ動作

デバイスがアイドルモードになると、システムクロックは機能継続しますが、CPU はコード実行を停止します。タイマモジュールはオプションでアイドルモード中でも動作継続します。

TSIDL ビット (TxCON<13>) でアイドルモード中にタイマを停止させるか通常動作継続とするかを選択します。TSIDL = 0 の場合、モジュールはアイドルモード中でも動作を継続できます。TSIDL = 1 の場合はアイドルモードでモジュール動作を停止します。

## 14.12.3 ダズモード中のタイマ動作

ダズモード中のタイマ動作は、通常動作と全く同じです。デバイスがダズモードになると、システムクロック源はそのまま動作し、CPU はスロークロックで動作します。詳細は第 10 章「省電力機能」を参照して下さい。

## 14.13 タイマ モジュールを使用する周辺モジュール

### 14.13.1 入力キャプチャ/出力コンペア用のタイム ベース

入力キャプチャと出力コンペア周辺モジュールは、タイム ベースに 2 つのタイマ モジュールのいずれかを選択できます。詳細は第 15 章「入力キャプチャ」、第 16 章「出力コンペア」と、個別のデバイス データ シートの仕様を参照して下さい。

### 14.13.2 A/D 用特殊イベント トリガ

異なるデバイスのそれぞれにおいて、1 個のタイプ C タイム ベースは、16 ビット、32 ビットいずれのモードでも周期一致により、特別な A/D 変換トリガ信号を生成できます。タイマ モジュールは A/D サンプリング ロジックに対し変換開始信号を提供します。

- T32 = 0 の場合、16 ビットタイマ レジスタ (TMRx) と対応する 16 ビット周期レジスタ (PRx) が一致したとき、A/D 特殊イベント トリガ信号を生成する
- T32 = 1 の場合、32 ビットタイマ (TMRx:TMRy) と、32 ビットの対応する結合周期レジスタ (PRx:PRy) が一致したとき、A/D 特殊イベント トリガ信号を生成する

特殊イベント トリガ信号は常にタイマで生成されています。トリガ要因としての選択は、A/D 変換制御レジスタで行わなくてはなりません。さらなる情報は第 17 章「10 ビット A/D コンバータ」と個別のデバイスのデータシートを参照して下さい。

### 14.13.3 外部割り込みピンとしてのタイマ

タイマごとの外部クロック入力ピンは、追加割り込みピンとして使用できます。割り込みを生成するためには、タイマ周期レジスタ PRx にゼロでない値が書き込まれ、TMRx レジスタに周期レジスタに書き込まれている値より 1 少ない値で初期化します。タイマは、1:1 のクロック プリスケール値に設定して下さい。割り込みは、外部クロック信号の次の立ち上がりエッジが検出されると発生します。

### 14.13.4 I/O ピン制御

タイマ モジュールが有効化され、外部クロックか、ゲート動作に構成されるときは、I/O ピン方向が入力に構成されていることを確認しなければなりません。タイマ モジュールを有効化してもピン方向は構成されません。

### 14.13.5 タイマの有効化

タイマを有効化するには、モジュール有効化ビットをセットする際、一緒にタイマ モジュール無効化ビット (PMD1 レジスタの TxMD ビット) をクリアしなければなりません。TxMD ビットをセットすると、モジュールへの全クロック源が無効化されますので、消費電力は絶対最小値となります。この状態では、関連する周辺モジュールの制御とステータス レジスタも無効化されますので、これらのレジスタへの書き込みは無効で、読み出した値も無効なものとなります。

## 14.14 レジスタ マップ

PIC24F のタイマ モジュールに関連する特殊機能レジスタのまとめを表 14-1 に示します。

表 14-1: タイマ モジュール関連特殊機能レジスタ

SFR 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	リセット後
PMD1	T5MD	T4MD	T3MD	T2MD	T1MD	—	—	—	I2C1MD	U2MD	U1MD	SPI2MD	SPI1MD	—	—	ADCMD	0000
TMR1	タイマ 1 レジスタ																xxxx
PR1	周期レジスタ 1																FFFF
T1CON	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	—	TSYNC	TCS	—	0000
TMR2	タイマ 2 レジスタ																xxxx
TMR3HLD	タイマ 3 保持レジスタ (32 ビット タイマ動作のときのみ)																xxxxx
TMR3	タイマ 3 レジスタ																xxxxx
PR2	周期レジスタ 2																FFFF
PR3	周期レジスタ 3																FFFF
T2CON	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	T32	—	TCS	—	0000
T3CON	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	—	—	TCS	—	0000
TMR4	タイマ 4 レジスタ																xxxxx
TMR5HLD	タイマ 5 保持レジスタ (32 ビット タイマ動作のときのみ)																xxxxx
TMR5	タイマ 5 レジスタ																xxxxx
PR4	周期レジスタ 4																FFFF
PR5	周期レジスタ 5																FFFF
T4CON	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	T32	—	TCS	—	0000
T5CON	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	—	—	TCS	—	0000
IFS0	—	—	AD1IF	UITXIF	U1RXIF	SPI1IF	SPF1IF	T3IF	T2IF	OC2IF	IC2IF	—	T1IF	OC1IF	IC1IF	INT01F	0000
IFS1	U2TXIF	U2RXIF	INT2IF	T5IF	T4IF	OC4IF	OC3IF	—	—	—	—	INT1IF	CNIF	CMIF	M2C1IF	SI2C1IF	0000
IEC0	—	—	AD1IE	UITXIE	U1RXIE	SPI1IE	SPF1IE	T3IE	T2IE	OC2IE	IC2IE	—	T1IE	OC1IE	IC1IE	INT01E	0000
IEC1	U2TXIE	U2RXIE	INT2IE	T5IE	T4IE	OC4IE	OC3IE	—	—	—	—	INT1IE	CNIE	CMIE	M2C1IE	SI2C1IE	0000
IPC0	—	T1IP2	T1IP1	T1IP0	—	OC1IP2	OC1IP1	OC1IP0	—	IC1IP2	IC1IP1	IC1IP0	—	INT0IP2	INT0IP1	INT0IP0	4444
IPC1	—	T2IP2	T2IP1	T2IP0	—	OC2IP2	OC2IP1	OC2IP0	—	IC2IP2	IC2IP1	IC2IP0	—	—	—	—	4440
IPC2	—	U1RXIP2	U1RXIP1	U1RXIP0	—	SPI1IP2	SPI1IP1	SPI1IP0	—	SPF1IP2	SPF1IP1	SPF1IP0	—	T3IP2	T3IP1	T3IP0	4444
IPC6	—	T4IP2	T4IP1	T4IP0	—	OC4IP2	OC4IP1	OC4IP0	—	OC3IP2	OC3IP1	OC3IP0	—	—	—	—	4440
IPC7	—	U2TXIP2	U2TXIP1	U2TXIP0	—	U2RXIP2	U2RXIP1	U2RXIP0	—	INT2IP2	INT2IP1	INT2IP0	—	T5IP2	T5IP1	T5IP0	4444

注: メモリ マップの詳細については、その個別のデバイスのデータ シートを参照して下さい。

## 14.15 関連するアプリケーション ノート

この項では、マニュアルのこの章に関連するアプリケーションノートを一覧アップします。これらのアプリケーションノートは、特に PIC24F デバイス ファミリー用に書かれているわけではありませんが、その概念は適切であり、変更、あるいは制限事項も考慮に入れて使用可能です。現状、タイマ モジュールに関連する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 #
非同期クロックモードでのタイマ 1 の使い方	AN580

**注：** PIC24F ファミリ デバイスに関するその他のアプリケーション ノートやコード例についてはマイクロチップ ウェブサイト ([www.microchip.com](http://www.microchip.com)) をご覧下さい。



14.16 改版履歴

リビジョン A (2006 年 4 月)

本文書の初版発行

ノート: