第 14 章 定时器

目录

本章包括下列主题：

14.1 简介

视具体器件，PIC24F 器件系列提供了几个 16 位定时器。这些定时器被指定为 Timer1、Timer2、Timer3... 等。

每个定时器模块均为 16 位定时器 / 计数器，由以下可读 / 写寄存器组成：

- TMRx：16 位定时器计数寄存器
- PRx：与该定时器相关的 16 位周期寄存器
- TxCON：与该定时器相关的 16 位控制寄存器

每个定时器模块还有与中断控制相关的位：

- 中断允许控制位 （TxIE）
- 中断标志状态位 （TxIF）
- 中断优先级控制位 （TxIP<2:0>）

除了某些例外的功能之外，所有 16 位定时器都有相同的功能电路。16 位定时器分为三种类型以说明其功能上的差异：

- A 类型时基
- B 类型时基
- C 类型时基

有些 16 位定时器可以组合成为 32 位定时器。

本章不介绍与外设器件相关的专用定时器。例如，其中包括与输入捕捉或输出比较模块相关的时基。

14.2 不同的定时器

除了某些例外之处， PIC24F 器件上所有可用的 16 位定时器在功能上是相同的。这些 16 位定时器分为三种功能类型：A 类型定时器、 B 类型定时器和 C 类型定时器。

注：	关于可用的定时器及其相应的类型信息，请参见具体器件数据手册。

14.2.1 A 类型定时器

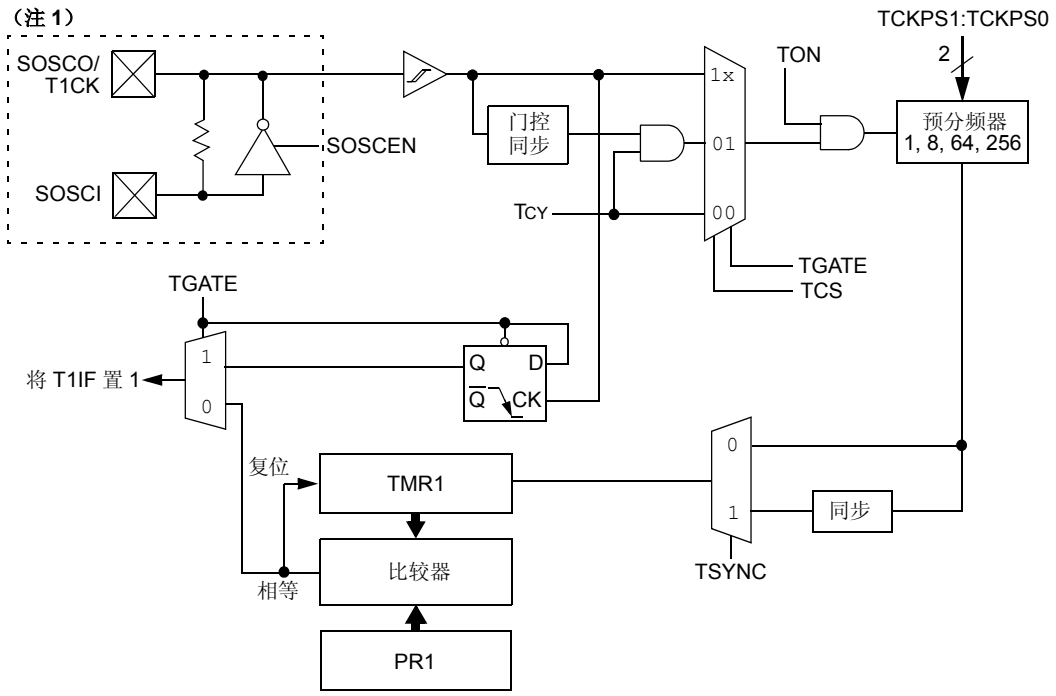
在大多数 PIC24F 器件上，至少有一个 A 类型定时器。对于大多数 PIC24F 器件，Timer1 是 A 类型定时器。与其他类型的定时器相比， A 类型定时器有下列独特的功能：

- 可以使用器件的低功耗 32 kHz 振荡器作为时钟源工作
- 可以在使用外部时钟源的异步模式下工作

特别值得一提的是， A 类型定时器独特的功能使它可以用于计时，或者用作辅助系统时钟源。

注：	大多数 PIC24F 器件具有 HW RTCC 模块，因而无需再使用硬件 RTCC。

图 14-1： A 类型定时器框图



注 1： 关于使能辅助振荡器的信息，请参见**第 6 章 “振荡器”**。

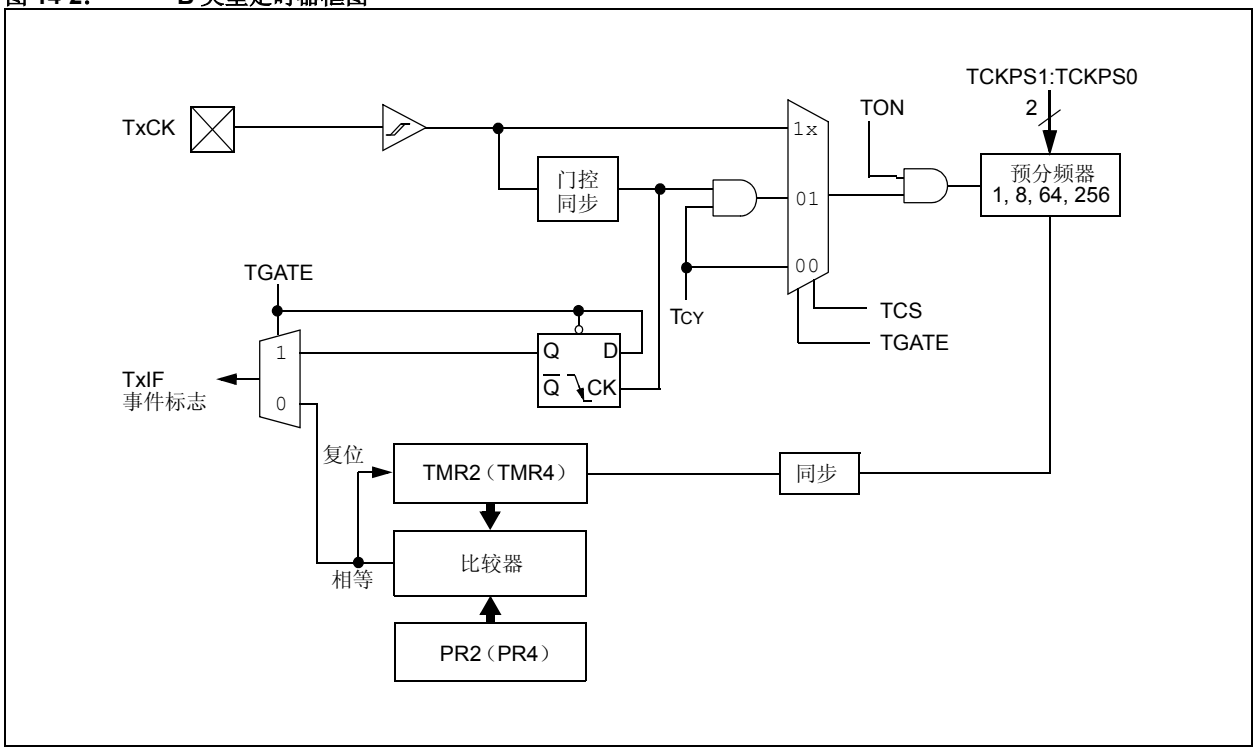
14.2.2 B 类型定时器

在大多数 PIC24F 器件上，如果存在 Timer2 和 Timer4，它们是 B 类型定时器。与其他类型的定时器相比， B 类型定时器有下列独特的功能：

- B类型定时器可以和C类型定时器相连形成32位定时器。B类型定时器的TxCON寄存器具有T32 控制位，用来使能 32 位定时器功能。
- B 类型定时器的时钟同步在预分频逻辑后执行。关于将时钟同步放在预分频逻辑后执行的益处，请参见**第 14.4.4 节 “使用快速外部时钟源的定时器工作原理”**。

图 14-2 给出了 B 类型定时器的框图。

图 14-2： B 类型定时器框图



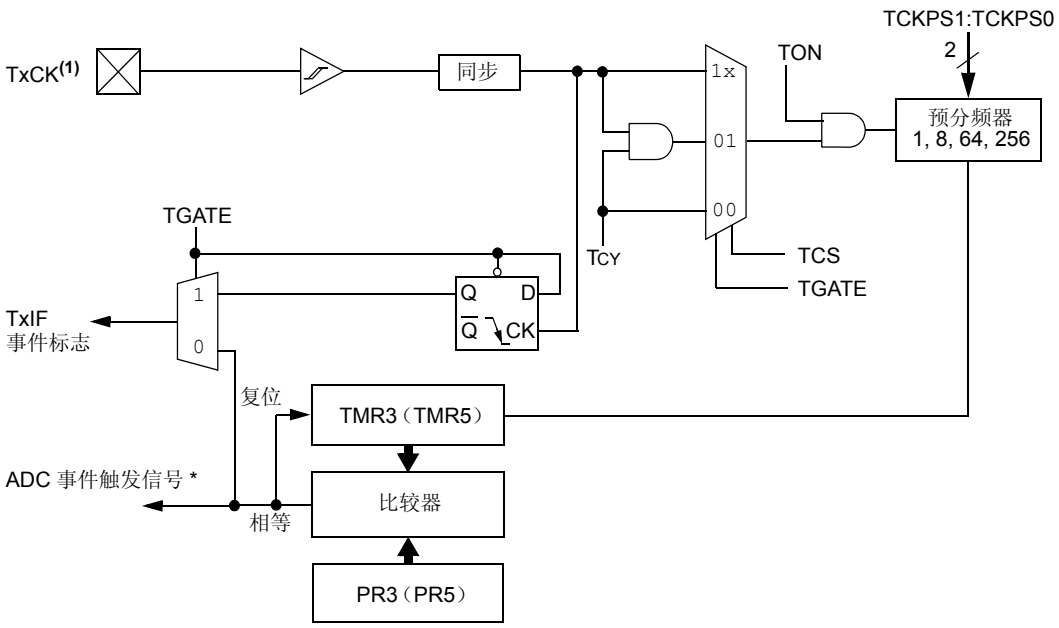
14.2.3 C 类型定时器

在大多数 PIC24F 器件上，Timer3 和 Timer5 是 C 类型定时器。与其他类型的定时器相比，C 类型定时器有下列独特的功能：

- C 类型定时器可以和 B 类型定时器相连形成 32 位定时器。
- 在某个给定的器件上，至少有一个 C 类型定时器能够触发 A/D 转换。

图 14-3 给出了 C 类型定时器的框图。

图 14-3： C 类型定时器框图



* ADC 事件触发信号仅在 Timer4/5 上可用。

注 1： PIC24F 系列的某些器件上可能没有 TxCK 引脚。请参见器件数据手册了解 I/O 引脚的详细信息。在这种情况下，该定时器必须使用系统时钟 （Fosc/2）作为它的输入时钟，除非它被配置为 32 位工作模式。

14.3 控制寄存器

寄存器 14-1： TxCON：A 类型时基控制

R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
TON	—	TSIDL	—	—	—	—	—
bit 15							bit 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	—
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现，读为 0	
-n = POR 值	1 = 置 1	0 = 清零	x = 未知

bit 15 **TON**：Timerx 开控制位
1 = 启动定时器
0 = 停止定时器

bit 14 **未实现**：读为 0

bit 13 **TSIDL**：空闲模式停止位
1 = 当器件进入空闲模式时，定时器停止工作
0 = 在空闲模式下定时器继续工作

bit 12-7 **未实现**：读为 0

bit 6 **TGATE**：Timerx 门控时间累加使能位
当 TCS = 1 时：
该位为无关位。
当 TCS = 0 时：
1 = 使能门控时间累加
0 = 禁止门控时间累加

bit 5-4 **TCKPS<1:0>**：Timerx 输入时钟预分频选择位
11 = 预分频比为 1:256
10 = 预分频比为 1:64
01 = 预分频比为 1:8
00 = 预分频比为 1:1

bit 3 **未实现**：读为 0

bit 2 **TSYNC**：Timerx 外部时钟输入同步选择位
当 TCS = 1 时：
1 = 同步外部时钟输入
0 = 不同步外部时钟输入
当 TCS = 0 时：
该位为无关位。读为 0。当 TCS = 0 时， Timerx 使用内部时钟。

bit 1 **TCS**：Timerx 时钟源选择位
1 = 来自 TxCK 引脚的外部时钟
0 = 内部时钟 （FOSC/2）

bit 0 **未实现**：读为 0

寄存器 14-2： TxCON：B 类型时基控制

R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
TON	—	TSIDL	—	—	—	—	—
bit 15							bit 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	—
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现，读为 0	
-n = POR 值	1 = 置 1	0 = 清零	x = 未知

bit 15 **TON**：Timerx 开控制位
当 T32 = 1 时（在 32 位定时器模式下）：
1 = 启动 32 位 TMRx:TMRy 定时器对
0 = 停止 32 位 TMRx:TMRy 定时器对
当 T32 = 0 时（在 16 位定时器模式下）：
1 = 启动 16 位定时器
0 = 停止 16 位定时器

bit 14 **未实现**：读为 0

bit 13 **TSIDL**：空闲模式停止位
1 = 当器件进入空闲模式时，定时器停止工作
0 = 在空闲模式下定时器继续工作

bit 12-7 **未实现**：读为 0

bit 6 **TGATE**：Timerx 门控时间累加使能位
当 TCS = 1 时：
该位为无关位。
当 TCS = 0 时：
1 = 使能门控时间累加
0 = 禁止门控时间累加

bit 5-4 **TCKPS<1:0>**：Timerx 输入时钟预分频选择位
11 = 预分频比为 1:256
10 = 预分频比为 1:64
01 = 预分频比为 1:8
00 = 预分频比为 1:1

bit 3 **T32**：32 位 Timerx 模式选择位
1 = TMRx 和 TMRy 形成 32 位定时器
0 = TMRx 和 TMRy 为独立的 16 位定时器

bit 2 **未实现**：读为 0

bit 1 **TCS**：Timerx 时钟源选择位
1 = 来自 TxCK 引脚的外部时钟
0 = 内部时钟（Fosc/2）

bit 0 **未实现**：读为 0

寄存器 14-3： TyCON：C 类型时基控制

R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
TON(1)	—	TSIDL	—	—	—	—	—
bit 15							bit 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	—
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现，读为 0	
-n = POR 值	1 = 置 1	0 = 清零	x = 未知

bit 15 **TON：** Timery 开控制位 (1)
1 = 启动 16 位 Timery
0 = 停止 16 位 Timery

bit 14 **未实现：** 读为 0

bit 13 **TSIDL：** 空闲模式停止位
1 = 当器件进入空闲模式时，定时器停止工作
0 = 在空闲模式下定时器继续工作

bit 12-7 **未实现：** 读为 0

bit 6 **TGATE：** Timery 门控时间累加使能位 (1)
当 TCS = 1 时：
该位为无关位。
当 TCS = 0 时：
1 = 使能门控时间累加
0 = 禁止门控时间累加

bit 5-4 **TCKPS<1:0>：** Timery 输入时钟预分频选择位
11 = 预分频比为 1:256
10 = 预分频比为 1:64
01 = 预分频比为 1:8
00 = 预分频比为 1:1

bit 3-2 **未实现：** 读为 0

bit 1 **TCS：** Timery 时钟源选择位
1 = 来自 TxCK 引脚的外部时钟
0 = 内部时钟 （FOSC/2）

bit 0 **未实现：** 读为 0

注 1： 当使能 32 位工作模式 （T2CON<3> = 1）时，这些位对 Timery 操作无影响；所有定时器功能都通过 T2CON 进行设置。

14.4 工作模式

每个定时器模块均可工作在以下几种模式之一：

- 作为定时器
- 作为同步计数器
- 作为门控定时器
- 作为异步计数器 （仅 A 类型和 C 类型时基）

定时器模式由以下位决定：

- TCS （TxCON<1>）：定时器时钟源控制位
- TSYNC （TxCON<2>）：定时器同步控制位 （仅 A 类型时基）
- TGATE （TxCON<6>）：定时器门控控制位

使用 TON 位 （TxCON <15>）使能或禁止每个定时器模块。

注：	只有 A 类型和 C 类型时基支持外部异步计数器模式。

14.4.1 定时器模式

所有类型的定时器都可以在基于系统时钟的定时器模式下工作。在定时器模式下，定时器的输入时钟由内部系统时钟（Fosc/2）提供。当使能为该模式时，对于 1:1 的预分频器设置，定时器的计数值在每个指令周期都会递增。通过清零 TCS 控制位 （TxCON<1>）选择定时器模式。同步模式控制位 TSYNC （TxCON<2>）在该模式下不起作用，因为使用了系统时钟源产生定时器时钟。

例 14-1： 使用系统时钟的 16 位定时器的初始化代码

```
/*  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 使用外部时钟输入的同步计数器模式

当 TCS 控制位（TxCON<1>）置 1 时，定时器的时钟源由外部提供，所选的定时器在 TxCK 引脚上的时钟输入的每个上升沿递增。

对于 A 类型时基，必须使能外部时钟同步以使其运行在同步计数器模式下。这通过将 TSYNC 控制位（TxCON<2>）置 1 实现。对于 B 类型和 C 类型时基，外部时钟输入总是与系统指令周期时钟 TCY 同步。

当定时器在同步计数器模式下工作时，对外部时钟高电平和低电平有一个最短时间的要求。通过在一个指令周期内的两个不同时间对外部时钟信号进行采样，可以实现外部时钟源与器件指令时钟的同步。

使用同步的外部时钟源工作的定时器在休眠模式下不工作，因为同步电路在休眠模式下是关闭的。

注：	当在同步计数器模式下使用时，外部输入时钟必须满足一定的高电平和低电平最短时间要求。

例 14-2： 使用外部时钟输入的 16 位同步计数器模式的初始化代码

```
/*  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 使用外部时钟输入的 A 类型定时器异步计数器模式

通过使用连接到 TxCK 引脚的外部时钟源， A 类型时基能够在异步计数模式下工作。当 TSYNC 控制位（TxCON<2>）清零时，外部时钟输入不与器件系统时钟源同步。该时基继续进行与内部器件时钟异步的递增计数。

异步工作的时基对以下应用是有益的：

- 时基可以在休眠模式下工作，并能够在发生周期寄存器匹配时产生中断，唤醒处理器。
- 可以使用低功耗 32 kHz 振荡器作为时基的时钟源，以提供一个辅助系统时钟源。

注 1：只有 A 类型时基支持异步计数器模式。

2：当在异步计数器模式下使用 Timerx 时，外部输入时钟必须满足一定的高电平和低电平最短时间要求。

例 14-3： 使用外部时钟输入的 16 位异步计数器模式的初始化代码

```
/*  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 使用快速外部时钟源的定时器工作原理

在一些应用中，可能希望使用其中一个定时器为来自频率相对较高的外部时钟源的时钟沿计数。在这些情况下，A 类型和 B 类型时基是对外部时钟源计数最合适的选择，因为这些定时器的时钟同步逻辑位于定时器预分频器之后 （见图 14-1 和图 14-2）。这样就可使用更高的外部时钟频率而不违反预分频器所要求的高电平和低电平最短时间。当为 A 类型或 B 类型时基选择的定时器预分频器分频比不是 1:1 时，外部时钟输入的高电平和低电平最短时间会被选择的预分频比减小。

A 类型时基的独特之处在于它可以在异步模式下工作，从而消除了任何预分频器的时序要求。

请注意，在所有情况下都不能超出外部时钟信号的高电平和低电平最短时间要求。这些最短时间是满足 I/O 引脚的时序要求所必需的。

欲了解与时基相关的外部时钟时序规范，请参见具体器件数据手册。

14.4.5 门控时间累加模式

门控时间累加模式允许内部定时器寄存器根据加在 TxCK 引脚上的高电平时间进行递增计数。在门控时间累加模式下，定时器时钟源来自内部系统时钟。当 TxCK 引脚为高电平状态时，定时器寄存器将递增计数，直到发生周期匹配或 TxCK 引脚变为低电平状态。引脚状态从高电平到低电平的转变会将 TxIF 中断标志位置 1。根据边沿发生的时间，中断标志位在 TxCK 引脚上信号下降沿产生后的 1 或 2 个指令周期置 1。

必须将 TGATE 控制位（TxCON<6>）置 1 以使能门控时间累加模式。必须使能定时器（TON（TxCON<15>）= 1），并且把内部时钟设置为定时器时钟源 （TCS （TxCON<1>）= 0）。

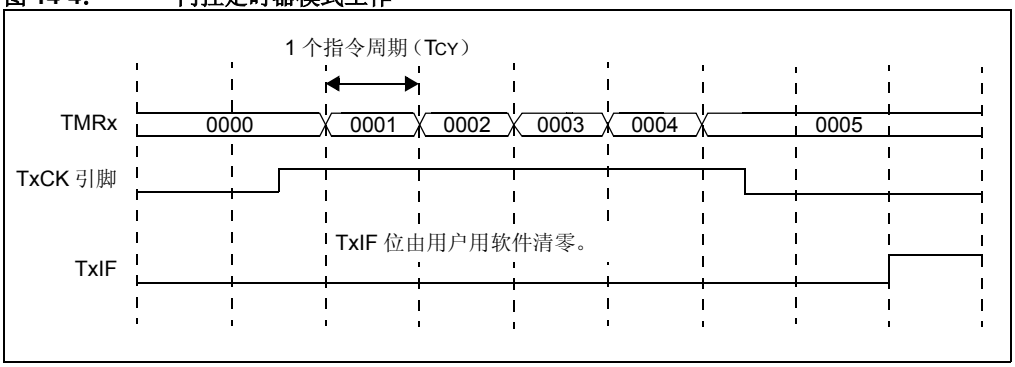
当加在 TxCK 引脚上的信号出现上升沿时，门控控制电路开始工作；当加在 TxCK 引脚上的信号出现下降沿时，门控控制电路终止工作。当外部门控信号为高电平时，对应的定时器将进行递增计数。

门控信号的下降沿将 TxIF 中断标志置 1 并产生中断 （如果允许）。

注：	在门控时间累加模式下，当发生定时器周期匹配时，定时器不会中断 CPU。

定时器计数的精度与定时器时钟周期直接相关。对于预分频比为 1:1 的定时器预分频器，定时器时钟周期为一个指令周期。对于预分频比为 1:256 的定时器预分频器，定时器时钟周期为 256 个指令周期。可将定时器时钟精度与门控信号的脉冲宽度相关联。关于门控脉冲宽度要求的更多详细信息，请参见具体器件数据手册中的 **“电气规范”** 部分。

图 14-4：　　门控定时器模式工作



例 14-4：　　16 位门控时间累加模式的初始化代码

```
/*  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 定时器预分频器

所有 16 位定时器的输入时钟 （FOSC/2 或外部时钟）都有 1:1、 1:8、 1:64 和 1:256 的预分频比选项。使用 TCKPS<1:0> 控制位 （TxCON<5:4>）选择时钟的预分频比。当发生以下情况中的任何一种时，预分频器计数器清零：

- 写 TMRx 寄存器
- 清零 TON （TxCON<15>）
- 任何器件复位

注： 写 TxCON 时， TMRx 寄存器不会清零。

14.6 定时器中断

根据工作模式的不同， 16 位定时器可以在发生周期匹配或外部门控信号的下降沿出现时产生中断。

当以下条件之一为真时， TxIF 位置 1：

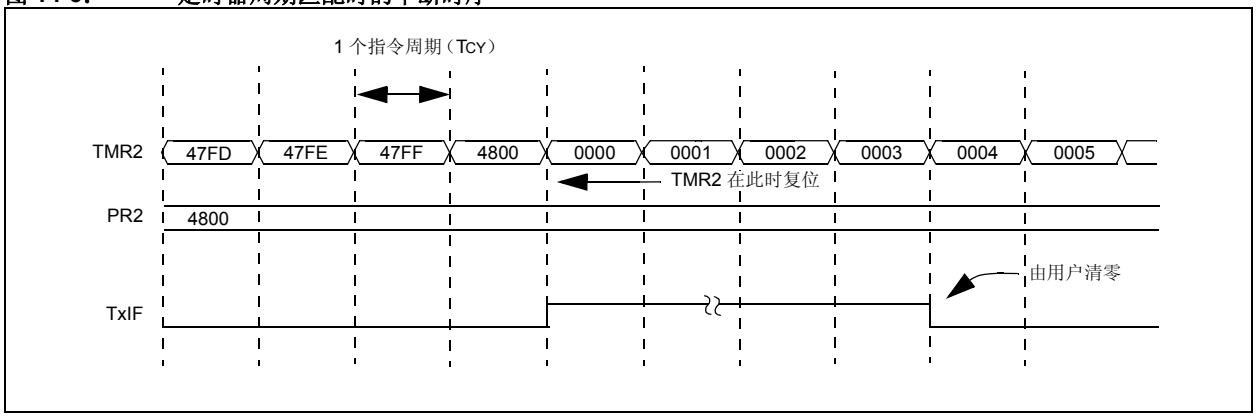
- 定时器的计数值与对应的周期寄存器相符，而且该定时器模块不工作在门控时间累加模式。
- 当定时器工作在门控时间累加模式下时，检测到 “门控”信号的下降沿。

TxIF 位必须用软件清零。

通过对应的定时器中断允许位 TxIE，可以将定时器使能为中断源。此外，为了使该定时器成为中断源，必须对中断优先级级别位 （TxIP<2:0>）写入非零值。更多详细信息，请参见**第 8 章 “中断”**。

注： 当周期寄存器 PRx 装入了 0x0000 且定时器被使能时，会发生特殊情形。在这种配置下，将不会产生定时器中断。

图 14-5： 定时器周期匹配时的中断时序



14.7 读写 16 位定时器模块寄存器

- 定时器模块的所有 SFR 都可以字节 （8 位）或字 （16 位）为单位**写入**。
- 定时器模块的所有 SFR 只能作为一个字 （16 位）**读取**。

14.7.1 写 16 位定时器

当定时器模块工作时，可以对该定时器及其对应的周期寄存器进行写操作。在执行字节写操作时用户应该对以下情况有所认识：

- 如果定时器正在递增计数，此时对定时器的低字节进行写操作，那么该定时器的高字节不受影响。如果将 0xFF 写入该定时器的低字节，在此次写操作之后的下一个定时器计数时钟将使低字节计满返回到 0x00，并产生向定时器的高字节进位。
- 如果定时器正在递增计数，此时对定时器的高字节进行写操作，那么该定时器的低字节不受影响。如果当此次写操作发生时定时器的低字节为 0xFF，那么在下一个定时器计数时钟将从定时器低字节产生进位，并且该进位会使定时器的高字节递增。

当通过一条指令将字或字节写入 TMRx 寄存器时，TMRx 寄存器的递增计数被屏蔽且在该指令周期内都不会发生递增计数。

在实时计时应用中，应该避免对使用异步模式的定时器进行写操作。关于异步计数器模式的信息，请参见**第 14.4.1 节 “定时器模式”**。

14.7.2 读 16 位定时器

对定时器及其相关 SFR 的所有读操作必须以字为单位读取 （16 位）。字节读取不起作用 （将返回 0）。

当该模块工作时，可以对定时器及其对应的周期寄存器进行读操作。读 TMRx 寄存器不会阻止定时器在同一个指令周期进行递增计数。

14.8 辅助振荡器 32 kHz 晶振输入

在各种不同的器件中， A 类型定时器模块均可使用 32 kHz 晶振用于实时时钟 （RTC）应用。

- 当辅助振荡器使能且该定时器被配置为使用外部时钟源时，辅助振荡器成为该定时器的时钟源。
- 通过将 OSCCON 寄存器中的 SOSCEN 控制位置 1，可以使能辅助振荡器。
- 32 kHz 晶振连接到 SOSCO/SOSCI 器件引脚。

更多详细信息，请参见**第 6 章 “振荡器”**。

14.9 32 位定时器配置

B 类型和 C 类型 16 位定时器模块可以组合形成 32 位定时器模块。C 类型时基成为组成的定时器的最高有效字 （msw），而 B 类型时基是最低有效字 （lsw）。

当配置为 32 位工作时，B 类型时基的控制位控制 32 位定时器的工作。C 类型时基的 TxCON 寄存器中的控制位不起作用。

组合的 32 位定时器使用 C 类型时基的中断允许、中断标志和中断优先级控制位进行中断控制。在 32 位定时器工作中，不使用 B 类型时基的中断控制和状态位。

注： 关于可以组合的 B 类型和 C 类型时基的信息，请参见具体器件数据手册。

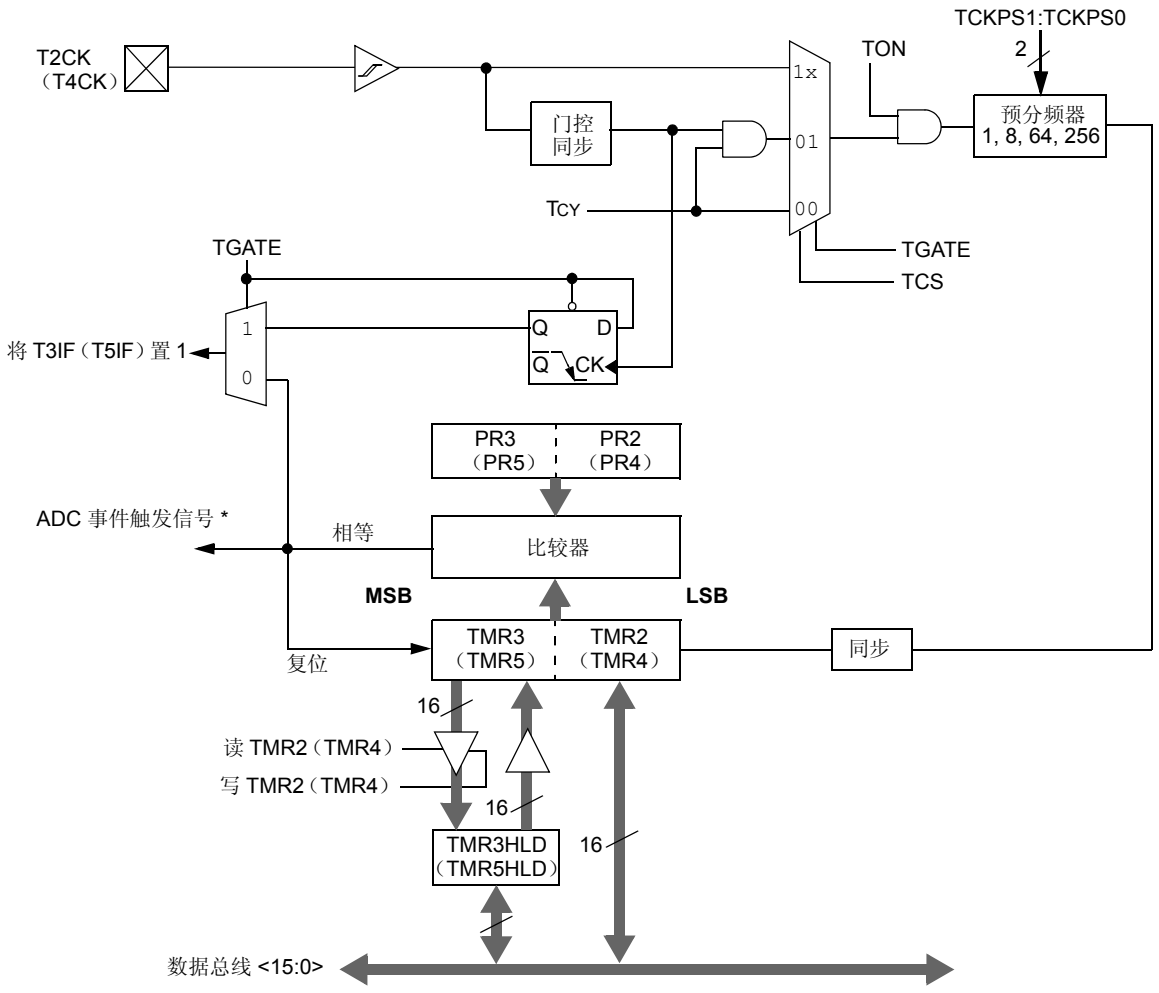
以下配置设置假设 Timer3 是 C 类型时基，而 Timer2 是 B 类型时基：

- TON （T2CON<15>）= 1。
- T32 （T2CON<3>）= 1。
- TCKPS<1:0> 位 （T2CON<5:4>）用于为 Timer2 设置预分频器模式 （B 类型时基）。
- TMR3:TMR2 这对寄存器包含定时器模块的 32 位值。TMR3（C 类型时基）寄存器是该 32 位定时器值的最高有效字，而 TMR2 （B 类型时基）寄存器是该 32 位定时器值的最低有效字。
- PR3:PR2 这对寄存器包含 32 位周期值，该值用于与 TMR3:TMR2 定时器值作比较。
- T3IE （IEC0<8>）用于允许该配置的 32 位定时器中断。
- T3IF （IFS0<8>）用作该定时器中断的状态标志。
- T3IP<2:0> 位 （IPC2<2:0>）为该 32 位定时器设置中断优先级级别。
- T3CON<15:0> 为 “无关”位。

图 14-6 所示为使用 Timer2 和 Timer3 的 32 位定时器模块示例的框图。

注： 输入捕捉和输出比较在 32 位定时器模式下不可用。

图 14-6： B 类型 /C 类型定时器对框图 （32 位定时器）



* ADC 事件触发信号仅在 Timer4/5 上可用。

注： 对于32位定时器/计数器工作，必须将32位定时器配置位T32置1。所有控制位都与T2CON和T4CON寄存器对应。

14.10 32 位定时器的工作模式

14.10.1 定时器模式

例 14-5 展示了如何在定时器模式下配置 32 位定时器。此例假设 Timer2 是 B 类型时基，而 Timer3 是 C 类型时基。对于 32 位定时器工作模式，必须将 T2CON 寄存器（B 类型时基）中的 T32 控制位置 1。当 Timer2 和 Timer3 被配置为 32 位定时器时，T3CON 控制位被忽略。设置和控制只需要使用 T2CON 控制位。32 位定时器模块使用 Timer2 时钟和门控输入，但是会产生中断将 T3IF 标志位置 1。 Timer2 和 Timer3 分别是 32 位定时器的 lsw 和 msw。来自 TMR2 的溢出（进位）使 TMR 进行递增计数。 32 位定时器进行递增计数，直到与由 PR2 和 PR3 组合形成的 32 位周期寄存器中预先装入的值相匹配，然后计满回零并继续计数。要使 32 位定时器能计数到最大值，应将值 0xFFFFFFFF 值装入 PR3:PR2。若允许中断，中断将在周期匹配时产生。

例 14-5： 使用指令周期作为输入时钟的 32 位定时器的初始化代码

```
/*  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 同步计数器模式

在同步计数器模式下，32 位定时器与 16 位定时器的工作方式类似。例 14-6 所示为如何在同步计数器模式下配置 32 位定时器。此例假设 Timer2 是 B 类型时基，而 Timer3 是 C 类型时基。

例 14-6： 使用外部时钟输入的 32 位同步计数器模式的初始化代码

```
/*  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 异步计数器模式

B 类型和 C 类型时基不支持异步外部时钟模式，所以不支持任何 32 位异步计数器模式。

14.10.4 门控时间累加模式

在门控时间累加模式下，32 位定时器与 16 位定时器的工作方式类似。例 14-7 所示为如何在门控时间累加模式下配置 32 位定时器。此例假设 Timer2 是 B 类型时基，而 Timer3 是 C 类型时基。

例 14-7： 32 位门控时间累加模式的初始化代码

```
/*  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 读写 32 位定时器

为了使 32 位读 / 写操作在 32 位定时器的 lsw 和 msw 之间同步，使用了额外的控制逻辑电路和保持寄存器 （见图 14-6）。每个 C 类型时基都有一个称为 TMRxHLD 的寄存器，当读或写该定时器的一对寄存器时使用它。只有其对应的定时器被配置为 32 位工作时才会使用 TMRxHLD 寄存器。

假设 TMR3:TMR2 形成一对 32 位定时器，用户应该首先从 TMR2 寄存器读取定时器值的 lsw。读 lsw 将会自动把 TMR3 的内容传送给 TMR3HLD 寄存器。然后用户可以读 TMR3HLD，以得到定时器值的 msw。此过程如例 14-8 所示：

例 14-8： 读 32 位定时器

```
/*  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
```

要将值写入 TMR3:TMR2 这对寄存器，用户应该首先将 msw 写入 TMR3HLD 寄存器。当定时器值的 lsw 被写入 TMR2 时， TMR3HLD 的内容将会自动传送到 TMR3 寄存器。

14.12 省电状态下的定时器工作

14.12.1 休眠模式下的定时器工作

当器件进入休眠模式后，系统时钟被禁止。如果定时器模块使用内部时钟源 （FOSC/2）运行，则该定时器也会被禁止。

A 类型定时器与其他定时器模块不同，因为它能使用系统时钟源异步工作。由于这个差别， A 类型时基模块可以在休眠模式下继续工作。要在休眠模式下工作， A 类型时基必须作如下配置：

- Timer1 模块使能， TON （TxCON<15>）= 1 ；
- 选择外部时钟源作为 Timer1 时钟源， TCS （TxCON<1>）= 1，且
- 将 TSYNC 位 （TxCON<2>）设置为逻辑 0 （异步计数器模式使能）。

注： 只有 Timer1 模块支持异步计数器工作模式。

满足了上述所有条件后，当器件处于休眠模式时，Timer1 将继续计数并检测周期匹配。当定时器和周期寄存器发生匹配时，TxIF 位将被置 1 并可以产生中断，从而选择将器件从休眠模式唤醒。更多详细信息，请参见**第 10 章 “省电特性”**。

14.12.2 空闲模式下的定时器工作

当器件进入空闲模式后，系统时钟源保持工作，但 CPU 停止执行代码。定时器模块可以选择在空闲模式下继续工作。

TSIDL 位 （TxCON<13>）选择在空闲模式下定时器模块是停止还是继续正常工作。如果 TSIDL = 0，在空闲模式下该模块将继续工作。如果 TSIDL = 1，在空闲模式下该模块将停止工作。

14.12.3 打盹模式下的定时器工作

打盹模式下的定时器工作和正常模式下一样。当器件进入打盹模式后，系统时钟源保持工作，CPU 可工作在较低时钟频率下。更多详细信息，请参见**第 10 章 “省电特性”**。

14.13 使用定时器模块的外设

14.13.1 输入捕捉 / 输出比较的时基

输入捕捉和输出比较外设可以选择两个定时器模块之一作为它们的时基。更多详细信息，请参见**第 15 章 “输入捕捉”**、**第 16 章 “输出比较”**和具体器件数据手册。

14.13.2 A/D 特殊事件触发信号

各种不同器件在 16 位和 32 位模式下发生周期匹配时，C 类型时基都能够产生特殊 A/D 转换触发信号。定时器模块为 A/D 采样逻辑提供了转换起始信号。

- 如果 T32 = 0，当 16 位定时器寄存器（TMRx）与各自对应的 16 位周期寄存器（PRx）之间发生匹配时，会产生 A/D 特殊事件触发信号。
- 如果 T32 = 1，当 32 位定时器寄存器 （TMRx:TMRy）与对应的 32 位组合的周期寄存器（PRx:PRy）之间发生匹配时，会产生 A/D 特殊事件触发信号。

特殊事件触发信号总是由定时器产生。必须在 A/D 转换器控制寄存器中选择触发源。更多信息，请参见**第 17 章 “10 位 A/D 转换器”**和具体器件数据手册。

14.13.3 定时器作为外部中断引脚

每个定时器的外部时钟输入引脚都可以用作额外的中断引脚。为了提供中断，应向定时器周期寄存器 PRx 写入非零值，并将 TMRx 寄存器初始化为一个比写入周期寄存器的值小 1 的值。定时器必须配置一个 1:1 的时钟预分频器。当检测到外部时钟信号的下一个上升沿时，将产生中断。

14.13.4 I/O 引脚控制

当定时器模块使能，并且配置为外部时钟或门控工作时，用户必须确保 I/O 引脚方向被配置为输入。使能该定时器模块不会配置引脚方向。

14.13.5 使能定时器

要使能定时器，必须清零定时器模块禁止位 （PMD1 寄存器中的 TxMD），同时将模块使能位置 1。将 TxMD 位置 1 会禁止该模块的所有时钟源，从而将其功耗降至最低。在此状态下，与外设相关的控制和状态寄存器也被禁止，因此对那些寄存器的写操作不起作用，读取值的操作也无效。

14.14 寄存器映射

表 14-1 中提供了与 PIC24F 定时器模块相关的特殊功能寄存器汇总。

表 14-1： 与定时器模块相关的特殊功能寄存器

SFR 名称	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	所有复位时
PMD1	T5MD	T4MD	T3MD	T2MD	T1MD	—	—	—	I2C1MD	U2MD	U1MD	SPI2MD	SPI1MD	—	—	ADCMD	0000
TMR1	Timer1 寄存器																xxxx
PR1	周期寄存器 1																FFFF
T1CON	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	—	TSYNC	TCS	—	0000
TMR2	Timer2 寄存器																xxxx
TMR3HLD	Timer3 保持寄存器 （仅限 32 位定时器工作）																xxxx
TMR3	Timer3 寄存器																xxxx
PR2	周期寄存器 2																FFFF
PR3	周期寄存器 3																FFFF
T2CON	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	T32	—	TCS	—	0000
T3CON	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	—	—	TCS	—	0000
TMR4	Timer4 寄存器																xxxx
TMR5HLD	Timer5 保持寄存器 （仅限 32 位定时器工作）																xxxx
TMR5	Timer5 寄存器																xxxx
PR4	周期寄存器 4																FFFF
PR5	周期寄存器 5																FFFF
T4CON	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	T32	—	TCS	—	0000
T5CON	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	—	—	TCS	—	0000
IFS0	—	—	AD1IF	U1TXIF	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	U1TXIE	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： 关于存储器映射的详细信息，请参见具体器件数据手册。

14.15 相关应用笔记

本节列出了与手册本章内容相关的应用笔记。这些应用笔记可能并不是专为 PIC24F 器件系列而编写的，但其概念是相关的，通过适当修改即可使用，但在使用中可能会受到一定限制。当前与定时器模块相关的应用笔记有：

标题	应用笔记编号
Using Timer1 in Asynchronous Clock Mode	AN580

注： 如需获取更多 PIC24F 系列器件的应用笔记和代码示例，请访问 Microchip 网站（www.microchip.com）。

14.16 版本历史

版本 A （2006 年 4 月）

这是本文档的初始发行版。

注：

