CPU定时器

CPU定时器

定时器是用来准确控制时间的工具。在生活中，古时用的沙漏，现在用的闹钟等都属于定时器。DSP为了能够精确的控制时间，以满足控制某些特定事件的要求，定时器是不可缺少的内容。F28335内部具有3个32位的CPU定时器——Timer0、Timer1和Timer2。

CPU定时器工作原理

CPU定时器(0/1/2)的内部结构如图9-1所示。

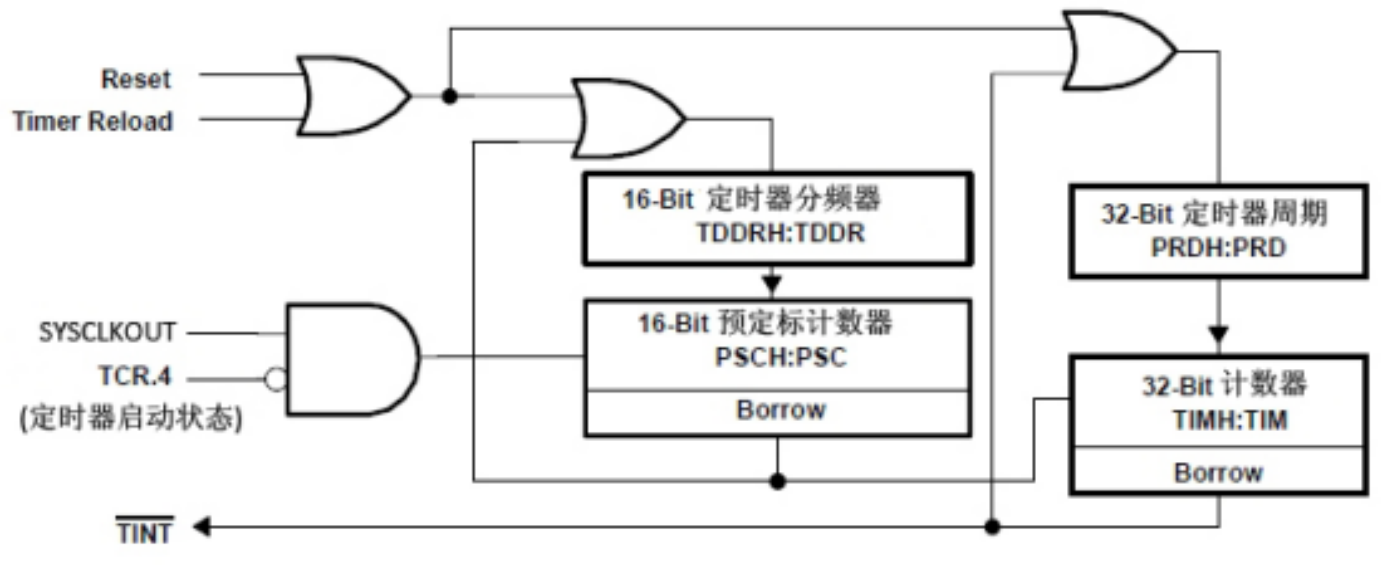

图9-1 CPU定时器内部结构

CPU定时器工作原理

从图9-1可以看到CPU定时器的几个寄存器，32位的定时器周期寄存器PRDH：PRD，32位的计数器寄存器TIMH：TIM，16位的定时器分频器寄存器TDDRH：TDDR，16位的预定标计数器寄存器PSCH：PSC。这里第一次遇到“XH：X”形式表示寄存器的方式，顺带介绍一下。因为F28335的存储器是16位的，但是CPU定时器是32位的，例如定时器周期寄存器、定时器计数器寄存器都是32位的，那如何用16位的存储器表示32位的呢？很显然，可以用2个16位的存储器XH和X来表示32位的寄存器，其中XH表示高16位，而X表示低16位。

CPU定时器工作原理

在讲CPU定时器工作原理之前，先来看看生活中的例子。比如，每天上班最痛苦的莫过于早上起床了，爱睡懒觉的朋友可能没有办法只好用闹钟把自己闹醒。首先前一天晚上睡觉前把闹钟设定好，闹钟每秒走动1次，当闹钟显示的时间和设定的时间相同时，闹钟就开始打铃，把睡觉中的主人给叫醒。这是生活中常见的例子，其实CPU定时器的工作原理与其类似，下面详细讲解。

CPU定时器工作原理

图9-2为CPU定时器的工作原理图。在CPU定时器工作前，先要根据实际的需求，计算好CPU定时器周期寄存器的值，然后给周期寄存器PRDH:PRD赋值，这就好比给闹钟设定时间一样。当启动定时器开始计数时，周期寄存器PRDH：PRD里面的值装载进定时器计数寄存器TIMH：TIM中。好比闹钟每隔1s走动一下一样，计数器寄存器TIMH：TIM里面的值每隔一个TIMCLK就减小1，直到计数到0，完成一个周期的计数。闹钟到点后会打铃，而CPU定时器这时候就会产生一个中断信号，关于中断的知识将在下一章中详细介绍。完成一个周期的计数后，在下一个定时器输入时钟周期开始时，周期寄存器PRDH：PRD里面的值重新装载入计数器寄存器TIMH：TIM中，周而复始地循环下去。一个CPU定时器周期所经历的时间就等于(PRDH:PRD+1)*TIMCLK。

CPU定时器工作原理

计数器寄存器TIMH：TIM每隔TIMCLK时间减少1，那TIMCLK究竟是多久呢？这个就是定时器分频器TDDRH：TDDR和定时器预定标器PSCH：PSC来控制的。先给定时器分频器TDDRH：TDDR赋值，然后装载入预定标器PSCH：PSC中，每隔一个SYSCLKOUT脉冲，PSCH：PSC中的值减1，当PSCH：PSC中的值为0的时候，就会输出一个TIMCLK，从而TIMH：TIM减1。在下一个定时器输入时钟周期开始时，TDDRH：TDDR中的值重新装载入PSCH：PSC中，周而复始的循环下去。因此，TIMCLK就等于(TDDRH:TDDR+1)个系统时钟的时间。

CPU定时器工作原理

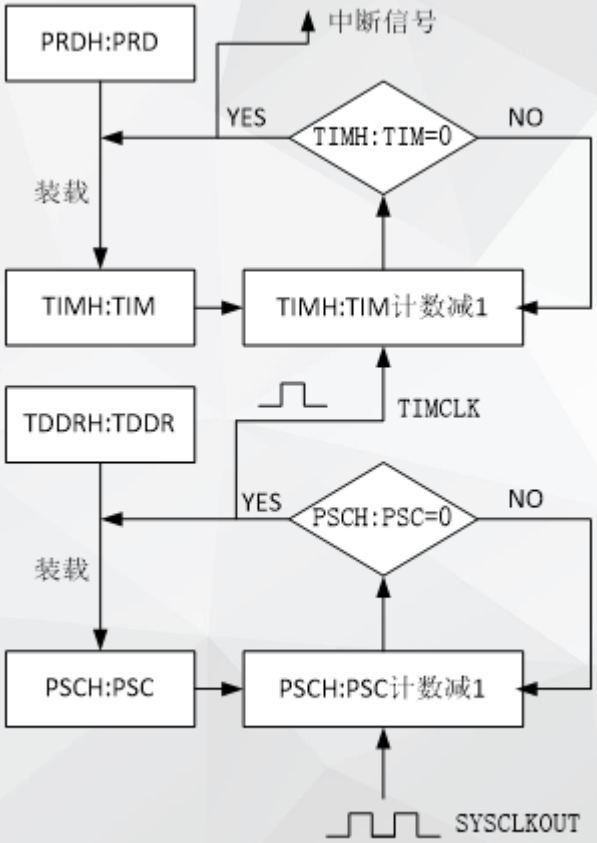

图9-2 CPU定时器工作原理

CPU定时器工作原理

从上面的介绍可以看到，如果想要用CPU定时器来计量一段时间的话，需要设定的寄存器有两个，一个是周期寄存器PRDH：PRD，一个是分频器寄存器TDDRH：TDDR。分频器寄存器TDDRH：TDDR决定了CPU定时器计数时每一步的时间。假设系统时钟SYSCLKOUT的值为X MHz，那么计数器每走一步，所需要的时间为：

$$\mathrm{TIMCLK} = \frac{\mathrm{TDDRH{:}\,TDDR} + 1}{\mathrm{X}} * 10^{-6}\mathrm{s} \qquad (9-1)$$

因为CPU定时器一个周期计数了(PRDH:PRD+1)次，因此CPU定时器一个周期所计量的时间为：

$$\mathrm{T} = (\mathrm{PRDH{:}\,PRD} + 1) * \frac{\mathrm{TDDRH{:}\,TDDR} + 1}{\mathrm{X}} * 10^{-6}\mathrm{s} \qquad (9-2)$$

CPU定时器工作原理

实际应用的时候，通常是确定了要定时的时间T和CPU的系统时钟X，来确定周期寄存器PRDH：PRD的值。TDDRH：TDDR通常可以取为0，如果取0的时候，PRDH:PRD的值超过了32位寄存器的范围，那么TDDRH：TDDR可以取其他值，使得PRDH：PRD的值小一些，从而能放得下32位寄存器中。

CPU定时器寄存器

该部分所列举的是CPU定时器的所有寄存器，寄存器的具体内容可以在C2000助手中查看。

名称	地址	大小(*16)	说明
TIMER0TIM	0x0000 0C00	1	CPU定时器0计数器寄存器低位
TIMER0TIMH	0x0000 0C01	1	CPU定时器0计数器寄存器高位
TIMER0PRD	0x0000 0C02	1	CPU定时器0周期寄存器低位
TIMER0PRDH	0x0000 0C03	1	CPU定时器0周期寄存器高位
TIMER0TCR	0x0000 0C04	1	CPU定时器0控制寄存器
Reserved	0x0000 0C05	1	保留
TIMER0TPR	0x0000 0C06	1	CPU定时器0预定标寄存器低位
TIMER0TPRH	0x0000 0C07	1	CPU定时器0预定标寄存器高位
TIMER1TIM	0x0000 0C08	1	CPU定时器1计数器寄存器低位
TIMER1TIMH	0x0000 0C09	1	CPU定时器1计数器寄存器高位
TIMER1PRD	0x0000 0C0A	1	CPU定时器1周期寄存器低位
TIMER1PRDH	0x0000 0C0B	1	CPU定时器1周期寄存器高位
TIMER1TCR	0x0000 0C0C	1	CPU定时器1控制寄存器
Reserved	0x0000 0C0D	1	保留
TIMER1TPR	0x0000 0C0E	1	CPU定时器1预定标寄存器低位
TIMER1TPRH	0x0000 0C0F	1	CPU定时器1预定标寄存器高位
TIMER2TIM	0x0000 0C9	1	CPU定时器2计数器寄存器低位
TIMER2TIMH	0x0000 0C11	1	CPU定时器2计数器寄存器高位
TIMER2PRD	0x0000 0C12	1	CPU定时器2周期寄存器低位
TIMER2PRDH	0x0000 0C13	1	CPU定时器2周期寄存器高位
TIMER2TCR	0x0000 0C14	1	CPU定时器2控制寄存器
Reserved	0x0000 0C15	1	保留
TIMER2TPR	0x0000 0C16	1	CPU定时器2预定标寄存器低位
TIMER2TPRH	0x0000 0C17	1	CPU定时器2预定标寄存器高位

表9-1 CPU定时器寄存器列表

（此处可右键选择“放大”功能查看图像）

CPU定时器寄存器

1.定时器计数器寄存器低位

定时器计数器寄存器低位TIMERxTIM(x=0,1,2)如图9-3所示。

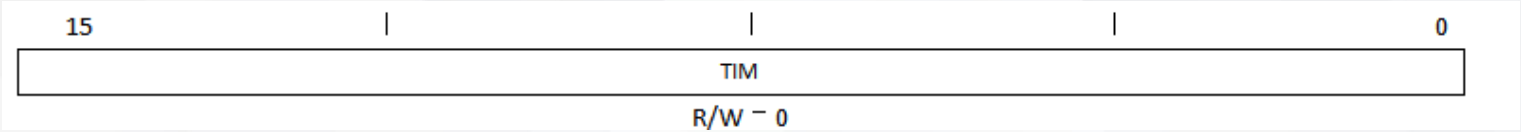

图9-3定时器计数器寄存器低位TIMERxTIM(x=0,1,2)

注：R=可读，W=可写，–0=复位后的值。

位	名称	定义
15～0	TIM	定时器计数寄存器（TIMH:TIM）：TIM寄存器是当前32位定时器的低16位，TIMH寄存器是当前32位定时器的高16位。TIMH：每隔（TDDRH:TDDR+1）个时钟周期，TIMH：TIM减1，其中，TDDRH:TDDR是定时器预定标分频值。当TIMH:TIM减到0时，TIMH:TIM重新装载 PRDH:PRD寄存器内所包含的周期值，同时产生定时器中断TINT信号。

2.定时器计数器寄存器高位

定时器计数器寄存器高位TIMERxTIMH(x=0,1,2)如图9-4所示。

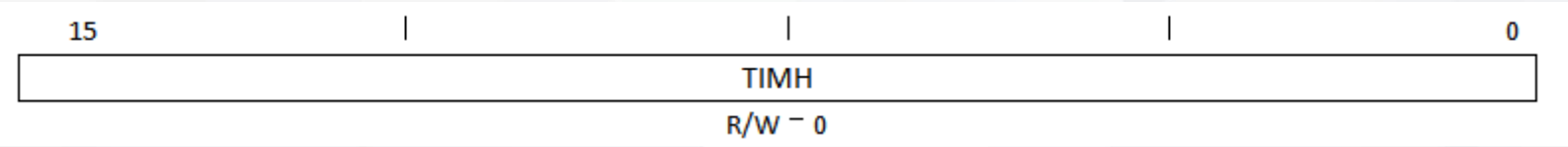

图9-4定时器计数器寄存器高位TIMERxTIMH(x=0,1,2)

注：R=可读，W=可写，–0=复位后的值。

位	名称	定义
15～0	TIMH	请参考TIMERxTIM的说明。

CPU定时器寄存器

3.定时器周期寄存器低位

定时器周期寄存器低位TIMERxPRD(x=0,1,2)如图9-5所示。

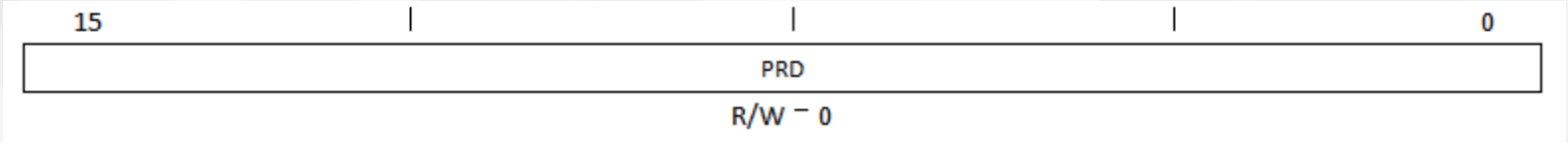

图9-5定时器周期寄存器低位TIMERxPRD(x=0,1,2)

注：R=可读，W=可写，–0=复位后的值。

位	名称	定义
15～0	PRD	定时器周期寄存器（PRDH:PRD）：PRD寄存器是32位周期寄存器的低16位，PRDH寄存器是32位周期周期寄存器的高16位。当TIMH:TIM减到0时，在下一个定时器输入时钟周期开始时（预定标器的输出），TIMH:TIM寄存器重载PRDH:PR寄存器内所包含的周期值。当用户在定时器控制寄存器（TCR）中对重装位（TRB）进行了设置时，PRDH:PR的内容也装到TIMH:TIM中。

4.定时器周期寄存器高位

定时器周期寄存器高位TIMERxPRDH(x=0,1,2)如图9-6所示。

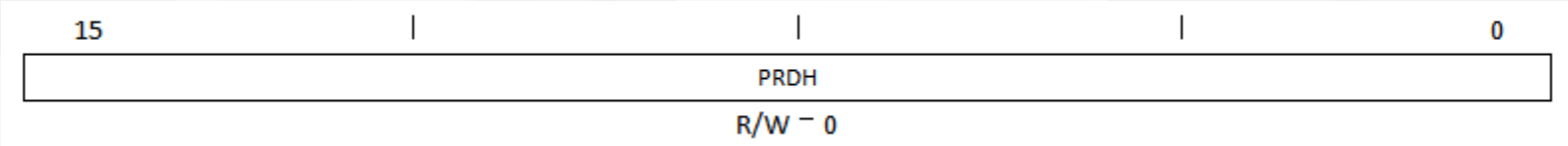

图9-5定时器周期寄存器高位TIMERxPRDH(x=0,1,2)

注：R=可读，W=可写，–0=复位后的值。

位	名称	定义
15～0	PRDH	请参考TIMERxPRD的说明。

CPU定时器寄存器

5.定时器控制寄存器

定时器控制寄存器TIMERxTCR(x=0,1,2)如图9-7所示。

15	14	13 12	11	9	9 8
TIF	TIE	Reserved	FREE	SOFT	Reserved
R/W−0	R/W−0	R−0	R/W−0	R/W−0	R−0

7 6	5	4	3 2 1 0
Reserved	TRB	TSS	Reserved

图9-7 定时器控制寄存器TIMERxTCR(x=0,1,2)

注：R=可读，W=可写，–0=复位后的值。

CPU定时器寄存器

位	名称	定义
15	TIF	定时器中断标志位。当定时器减到0时，标志位将置1，可通过软件写1对该位清0，但是只有计数器递减到0该位才会被置位。 对该位写1将清除该位，写0无效。
14	TIE	定时器中断使能位。如果定时器计数器递减到0，该位置1，定时器将会向CPU提出中断请求。
13～12	Reserved	保留。
11	FREE	定时器仿真方式：FREE和下面的SOFT位是专用于仿真的，这些位决定了在高级语言编程调试中，遇到断点时，定时器的状态。如果FREE位为1，那么在遇到断点时，定时器继续运行（即自由运行），在这种情况下，SOFT位不起作用。但是，如果FREE为0，则SOFT起作用。在此情形下，如果SOFT=0，定时器在下一个TIMH:TIM递减操作完成后停止。如果SOFT位为1，那么定时器在TIMH:TIM递减到0后停止。
9	SOFT	FREE SOFT定时器仿真方式 00 定时器在下一个TIMH:TIM递减操作完成后停止（硬停止）； 01 定时器在TIMH:TIM递减到0后停止（软停止）； 9 自由运行； 11 自由运行。
9～6	Reserved	保留。
5	TRB	定时器重装位。当向TRB写1时，PRDH:PRD的值装入TIMH:TIM，并且把定时器分频寄存器（TDDRH:TDDR）中的值装入预定标计数器（PSCH:PSC）。TRB位一直读作0。
4	TSS	定时器停止状态位。TSS是停止或启动定时器的一个标志位。要停止定时器，置TSS为1。要启动或重启动定时器，置TSS为0。在复位时，TSS清0并且定时器立即启动。
3～0	Reserved	保留。

CPU定时器寄存器

6.定时器预定标计数器低位

定时器预定标计数器低位TIMERxTPR(x=0,1,2)如图9-8所示。

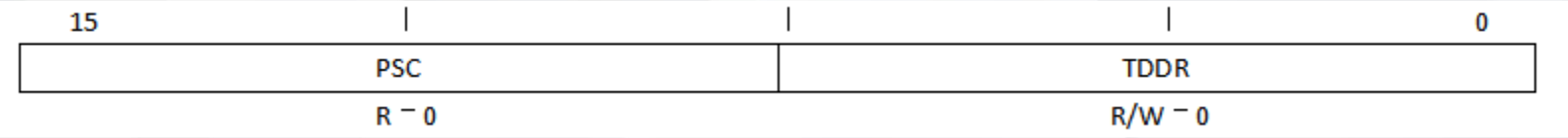

图9-8定时器预定标计数器低位TIMERxTPR(x=0,1,2)

注：R=可读，W=可写，–0=复位后的值。

位	名称	定义
15～8	PSC	定时器预定标器计数器。PSC是预定标计数器的低8位。PSCH是预定标计数器的高8位。对每一个定时器时钟周期，PSCH:PSC的值大于0，PSCH:PSC逐个减计数。PSCH:PSC到0后是一个定时器时钟（定时器预定标器的输出）周期，TDDRH:TDDR 的值装入PSCH:PSC，定时器计数器寄存器（TIMH:TIM）减1。无论何时，定时器重装位（TRB）由软件置1时，也重装PSCH:PSC。复位时，PSCH:PSC置为0。
7～0	TDDR	定时器分频器。TDDR是定时器分频器的低8位。TDDRH是定时器分频器的高8位。每过一个（TDDRH:TDDR+1）个定时器时钟周期，定时器计数器寄存器（TIMH:TIM）减1。复位时，TDDRH:TDDR位清0。当预定标器计数器（PSCH:PSC）值为0，一个定时器时钟源周期后，PSCH:PSC重装TDDRH:TDDR内的值，并使TIMH:TIM减1。无论何时，用软件置定时器重装位（TRB）为1， PSCH:PSC就会重装TDDRH:TDDR的值。

CPU定时器寄存器

7.定时器预定标计数器高位

定时器预定标计数器高位TIMERxTPRH(x=0,1,2)如图9-9所示。

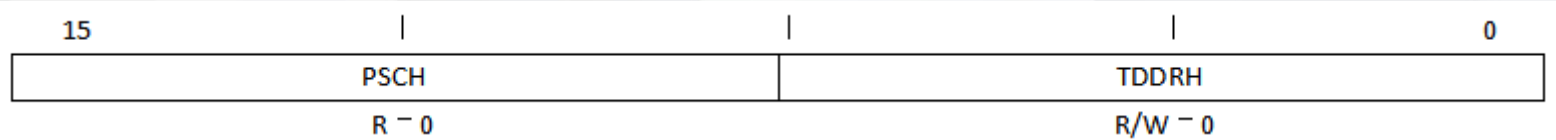

图9-9定时器预定标计数器高位TIMERxTPRH(x=0,1,2)

注：R=可读，W=可写，–0=复位后的值。

位	名称	定义
15 ~ 8	PSCH	请参考TIMERxTPR的说明。
7~0	TDDRH	请参考TIMERxTPR的说明。

CPU定时器

因为CPU定时器通常使用的时候是结合其周期中断来使用的，就是定时一个周期后去处理一些事件。由于还没有介绍中断的知识，所以此处暂时不介绍CPU定时器的应用，在下一章讲中断的时候，再结合中断的知识，来详细介绍CPU定时器的实际应用。

*程序清单 略