



<http://www.Microcontrol.cn> 微控设计网

中国 MSP430 单片机专业网站

MSP430F2 系列 16 位超低功耗单片机模块原理

第 9 章 定时器 B Timer B

版本: 1.3

日期: 2007.4.

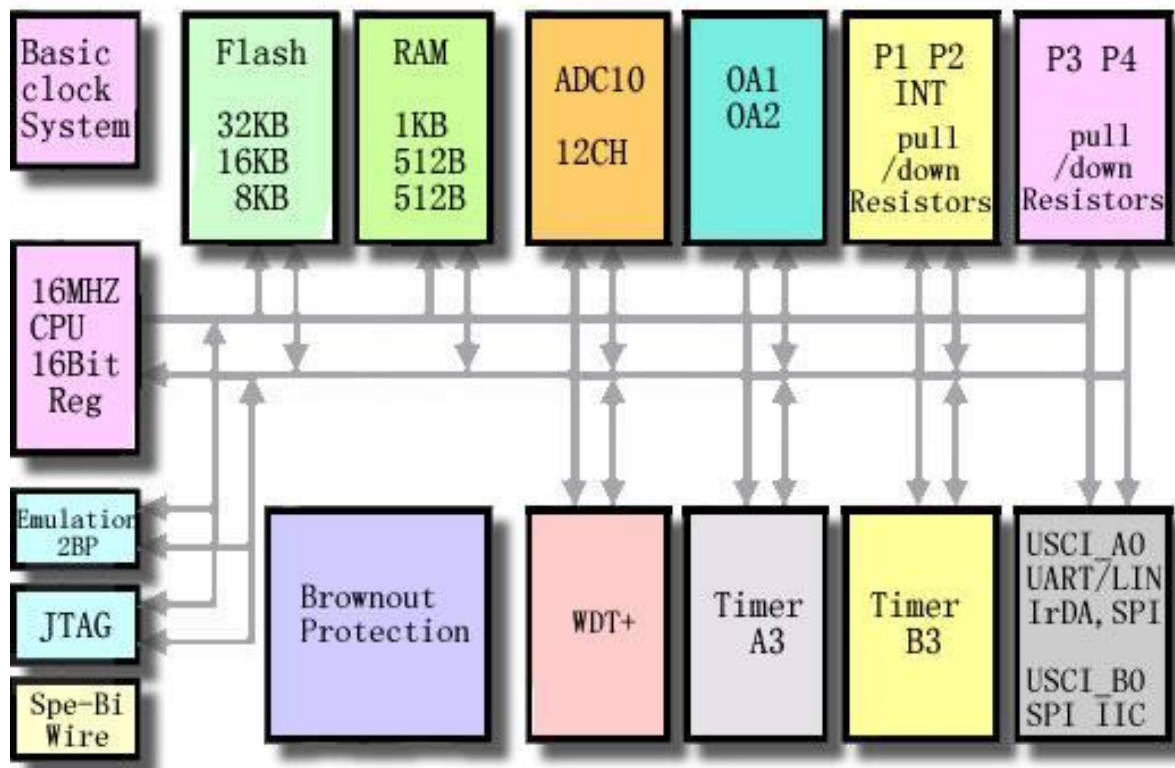
原文: TI MSP430x2xxfamily.pdf

翻译: 李璘 中国计量学院

编辑: DC 微控技术论坛总版主

注: 以下文章是翻译 TI MSP430x2xxfamily.pdf 文件中的部分内容。由于我们翻译水平有限, 有整理过程中难免有所不足或错误; 所以以下内容只供参考. 一切以原文为准。

详情请密切留意微控技术论坛。



定时器 B (Timer_B, 以后简称为 TB) 是一个 16 位的定时/计数器, 并复合了捕获/比较寄存器。Timer_B3 (拥有 3 个捕获比较器) 只存在于 MSP430x2xx 中。

章节

9.1 Timer_B 介绍	9-2
9.2 Timer_B 操作方法	9-4
9.3 Timer_B 寄存器	9-19

9.1 Timer_B 介绍

TB 是一个 16 位的定时/计数器, 拥有 3 个或 7 个捕获/比较寄存器。TB 可以支持捕获/比较功能、PWM 输出和定时器功能。TB 还有扩展中断的功能, 中断可以由定时器溢出产生或捕获比较寄存器产生。

TB 的特性如下:

- I 4 种操作模式的异步 16 位定时/计数器
- I 可选择配置的时钟源
- I 3 个或 7 个可配置的捕获/比较器
- I 可配置的 PWM 输出
- I 加载时同步的双缓冲比较锁存
- I 对所有 TB 中断快速响应的中断向量寄存器

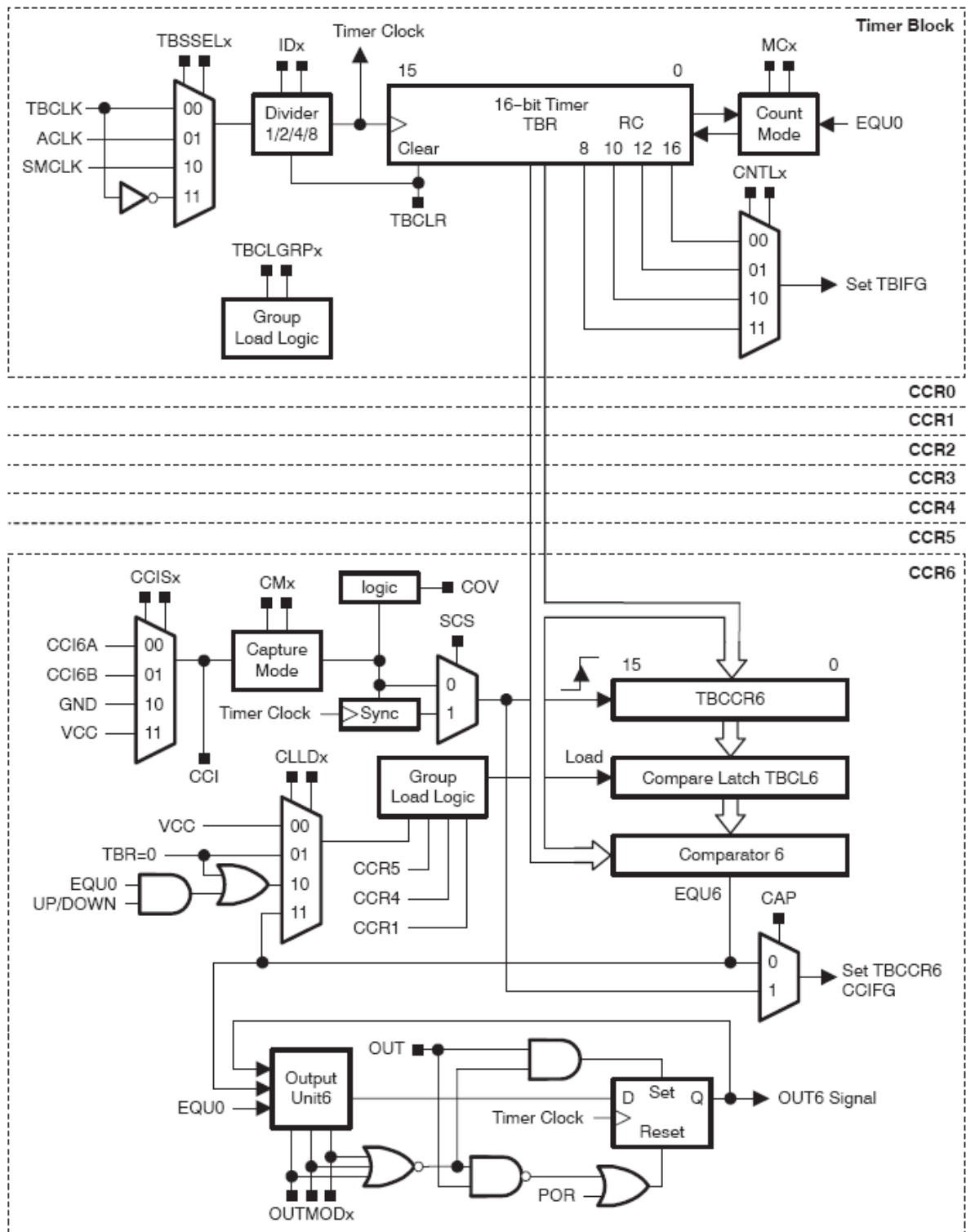
9.1.1 和定时器A的相同点和不同点

定时器B和定时器A的不同点如下:

- I 定时器B的长度是可编程的, 可编程为8, 10, 12, 16位
- I 定时器B TBCCR_x寄存器是双缓冲的, 并可以编组
- I 所有定时器B的输出可以为高阻抗状态
- I SCCI位功能在定时器B中不存在

TB 的结构图见图 9-1.

图 9-1.TB 结构图



9.2 Timer_B 的操作方法

TB 模块由用户软件来配置，TB 的配置将在下面的章节讨论。

9.2.1 16 位定时/计数器

16 位定时/计数器寄存器 TBR, 随着时钟信号的每个上升沿增/减(这由操作模式决定)。TBR 可以被软件读写。另外, 定时器在溢出时可以产生中断。TBR 可以由 TBCLR 位清除, 如果 TB 处于 up/down 模式, TBCLR 置位也会清除时钟分频器和计数方向。

注意:对 TB 寄存器的修改

建议在进行修改定时器的操作(对中断允许、中断标志和 TBCLR 的操作除外)时, 先停止定时器, 以避免产生未知的误操作。当定时器时钟和 CPU 时钟不同步时, 对 TBR 的读会由于定时器的运行而导致所读的结果是不可预料的。因此, 当定时器运行时, 需要多读几次, 通过软件多数表决的方式来原因正确的读数。对 TBR 的写操作是立即生效的。

TBR 的长度

定时器 B 可以通过 CNTLx 位将它配置为 8, 10, 12 或 16 位定时器。最大的计数数值 TBR_{max}, 可以相应为 0FFh, 03FFh, 0FFFh, 和 0FFFFh。在 8-, 10-, 和 12-位模式下, 对 TBR 写数据数据时, 数据的高 4 位必须为 0。

时钟源的选择和分频

定时器的时钟源可以是内部时钟源 ACLK, SMCLK, 或外部源 TBCLK 和 INCLK。时钟源由 TBSSEL 位来选择, 所选择的时钟可以通过 IDx 位进行 2、4 或 8 分频, 当 TBCLR 置位时, 分频器复位。

9.2.2 启动定时器

定时器可以通过以下 2 种方式启动或重新启动:

- I 当定时器计数到 MCX>0 并且时钟源处于活动状态时
- I 当定时器模式为 up 或 up/down 模式时(即单调增和增减模式), 定时器可以通过写 0 到 TBCL0 来停止计数。定时器可以通过写一个非 0 的数值来重新开始计数。在这种情况下, 定时器从 0 开始增计数。

9.2.3 定时器模式控制

定时器有 4 种操作模式, 见表 9-1, 他们分别是停止、单调增、连续和增减模式。操作模式由 MCx 位来选择。

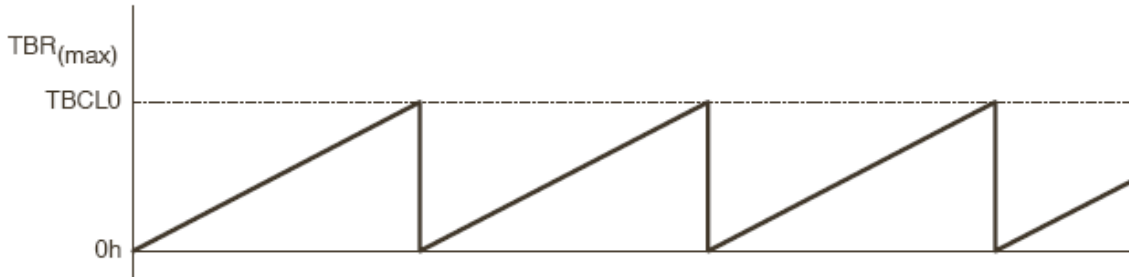
表 9-1 定时器模式

MCx	模式	说明
00	停止模式	定时器暂停
01	单调增模式	定时器循环地从 0 增到 TBCL0 的值
10	连续模式	定时器循环地从 0 连续增加到 0FFFFH
11	增减模式	定时器循环地从 0 增到 TBCL0 的值再连续减至 0

(1) 单调增模式

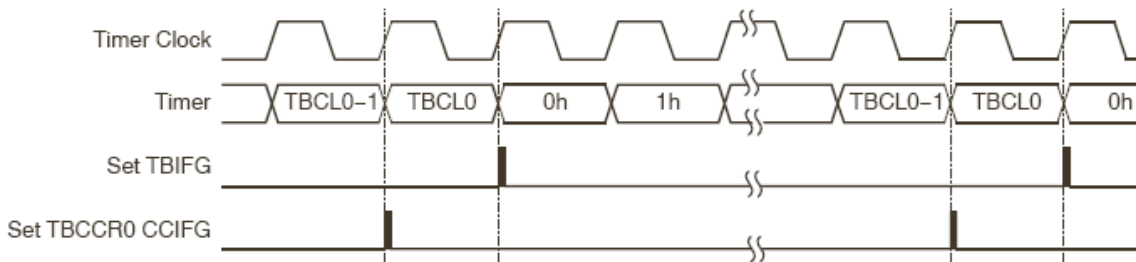
单调增模式用于计数周期不是 $TBR_{(max)}$ 的情况。定时器重复增计数值寄存器TBCL0的值，而TBCL0的值取决于定时周期，如图9-2，定时器计数周期为 $TBCL0+1$ 。当定时器的值等于TBCL0时，定时器就回到0重新计数。如果当定时器的值大于TBCL0，而此时选择单调增模式，定时器立即从0重新开始计数

图 9-2. 单调增模式



当定时器计数到TBCL0的值时，中断标志TBCCR0 CCIFG位置位。当定时器由TBCL0返回0时，TBIFG中断标志置位。图 9-3 说明了标志置位循环

图 9-3. 单调增模式标志位的变化



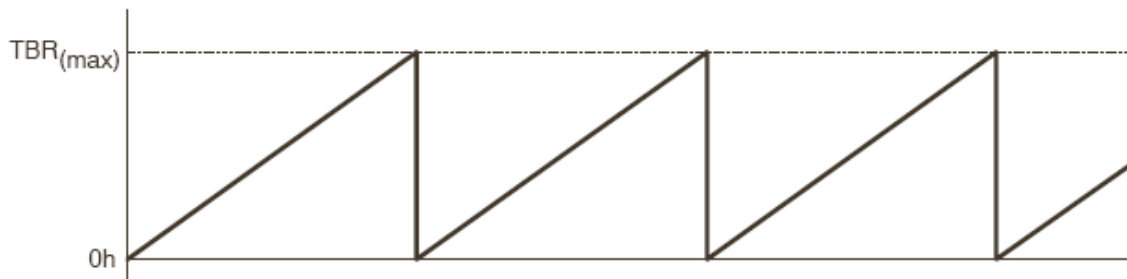
修改周期寄存器 TBCL0

在定时器在运行时修改TBCL0（并且TBCL0加载模式为立即模式），如果新的周期值大于或等于旧的周期值，或大于当前的定时器计数值，那么定时器立刻开始执行新的周期计数；如果新周期小于当前的计数值，那么定时器回到0。但是，在回到0之前会多一个额外的计数。

连续模式

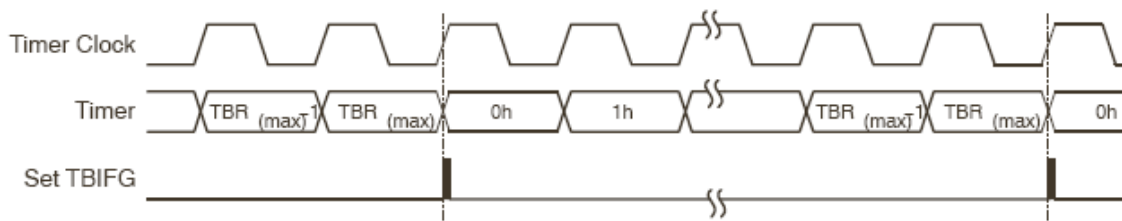
在连续模式中，定时器重复计数到 $TBR_{(max)}$ ，然后重新从0开始增计数，如图9-4。捕获比较寄存器TBCL0以及其他捕获比较器工作方式一样。

图 9-4.连续模式



当定时器从 $TBR(max)$ 到0时，TBIFG中断标志置位。图 9-5 表示了标志位的设置

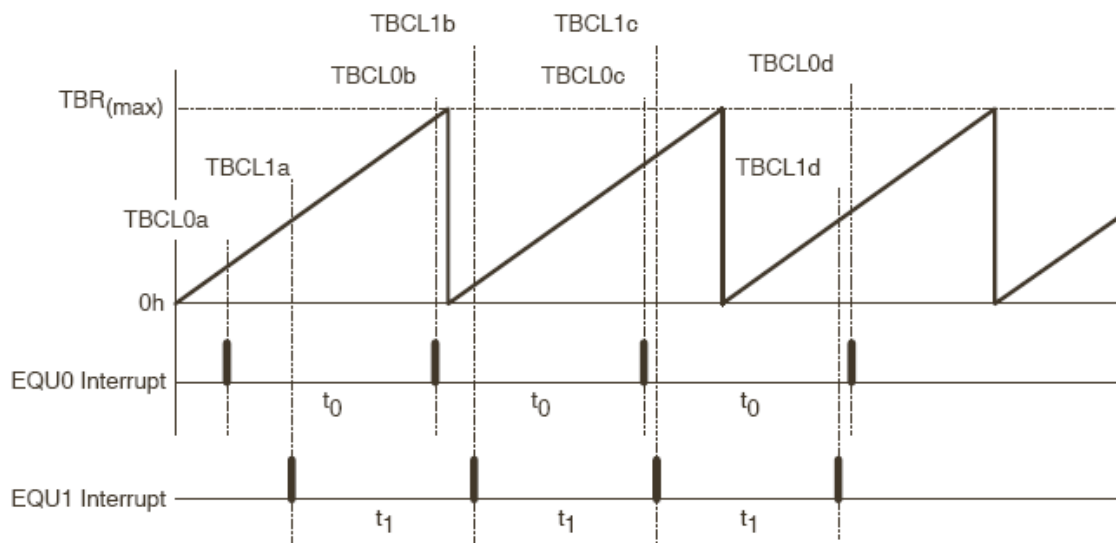
图 9-5. 连续模式标志位的设置



连续模式的使用

连续模式可以用于产生统一的时间间隔和输出频率。当每个时间间隔完成时就产生一个中断。下一个时间间隔的值在进入中断服务子程序时写入 $TBCLx$ 。图9-6显示了2个独立的时间间隔 t_0 和 t_1 写入捕获比较寄存器。在该应用中，时间间隔由硬件控制，与中断响应没有冲突。如果想产生多于3个或7个的时间间隔可以使用捕获比较寄存器。

图 9-6. 连续模式时间间隔



时间间隔可以由其他模式产生， $TBCL0$ 也可以作为周期寄存器使用。如果旧的 $TBCLx$ 的数据之和与新的相比，要比 $TBCL0$ 大，那么操作就会复杂得多。当旧的 $TBCLx$ 的值加上 t_x 比 $TBCL0$ 的值大，那么旧 $TBCL0$ 的值必须被减掉以获得正确的时间间隔。

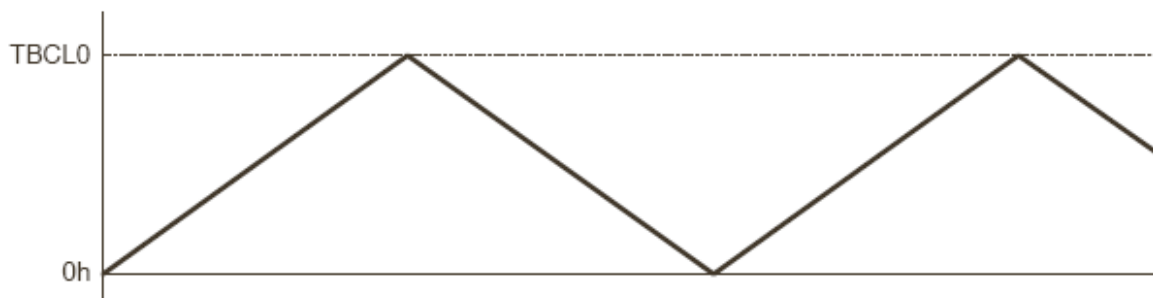
增减模式

增减模式在定时器周期不是 TBR_{max} 且需要产生对称的脉冲时使用。定时器增计数到 $TBCL0$ 再从 $TBCL0$ 减计数到 0，如图 9-7，周期是 $TBCL0$ 值的 2 倍。

注意： $TBCL0 > TBR_{max}$ 的情况

如果 $TBCL0 > TBR_{max}$ ，那么计数方式就和连续模式一样，不会从 TBR_{max} 减到 0。

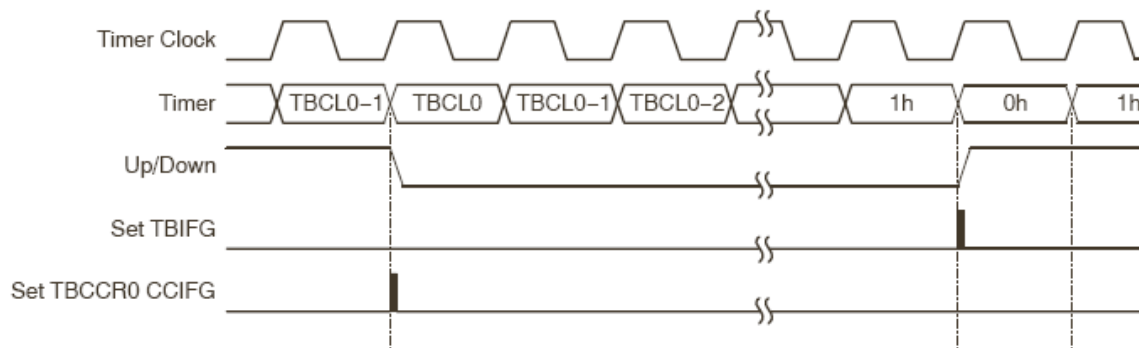
图 9-7.增减模式



该模式下，计数方向是固定的，即让定时器停止后再重新启动定时器，它就会沿着停止时的计数方向和数值开始计数。如果要从 0 开始，就需要将 $TBCLR$ 置位来清除方向。 $TBCLR$ 位也会清除 TBR 的值和定时器的时钟分频。

在增减模式中， $TBCCR0$ $CCIFG$ 中断标志和 $TBIFG$ 中断标志在一个周期中只置位一次，它们相隔 $1/2$ 个定时器周期。当定时器计数到由 $TBCL0-1$ 变到 $TBCL0$ 时， $CCIFG$ 置位；而定时器完成减计数从 $0001h$ 到 $0000h$ 时， $TBIFG$ 置位。图 9-8 表示了标志位的置位状况。

图 9-8.增减模式下的标志



改变 $TBCL0$ 周期寄存器

如果定时器正处于减计数时改变 $TBCL0$ 的值 ($TBCL0$ 的加载模式为立即模式)，定时器会继续减到 0，新的周期在减到 0 后开始；如果正处于增计数状态，并且新的周期已经锁存到 $TBCL0$ ，如果新周期大于等于原来的周期，或比当前计数值要大，定时器会增计数到新的周期再减计数；如果正处于增计数状态，新周期小于原来的周期，定时器立刻开始减计数，但是在定时器开始减计数之前会多计一个数。

增减模式的使用

增减模式支持在输出信号之间有死区时间的应用（参阅 TB 输出章节）。例如，避免出现过载情况，2 个输出驱动一个 H 桥不能同时为高。在图 9-9 的例中， t_{dead} 为

$$t_{dead} = t_{timer} \times (TBCL1 - TBCL3)$$

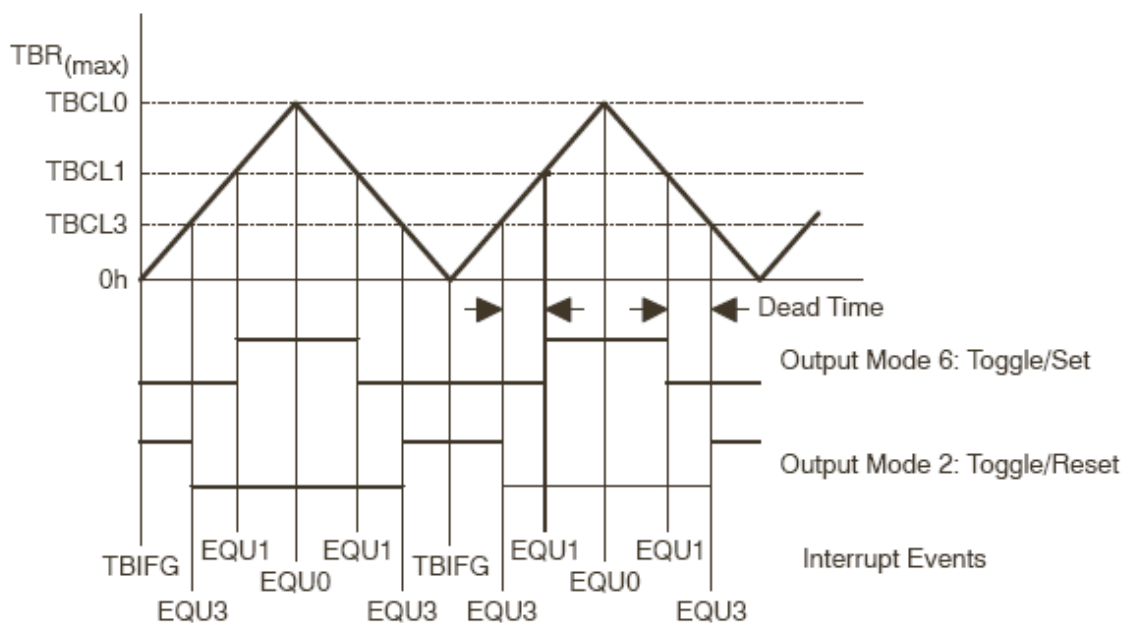
t_{dead} ——同时输出时必须没有反应的时间段

t_{timer} ——定时器时钟周期

TBCL_x——比较锁存器 x 的内容

可以同时加载成组的比较锁存来保证死区时间

图 9-9. 增减模式的输出



9.2.4 捕获比较模块

定时器 B 中有 3 个或 7 个相同的捕获比较模块 TBCCR_x，其中的任何一个模块可以用于定时器数据的捕获或产生时间间隔。

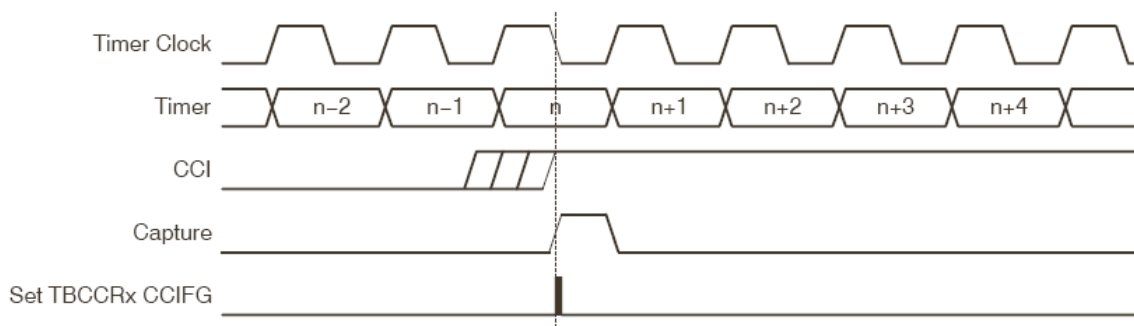
捕获模式

当 CAP=1 时，选择捕获模式。捕获模式用于记录时间事件，比如速度估计或时间测量。捕获输入 CCI_xA 和 CCI_xB 连接外部的引脚或内部的信号，这通过 CCIS_x 位来选择。CM_x 位选择捕获输入信号触发沿：上升沿、下降沿或 2 者都捕获。捕获事件发生于所选择的输入信号的触发沿。如果发生了捕获事件：

- I 定时器的值复制到 TBCCR_x 寄存器中
- I 中断标志位 CCIFG 置位

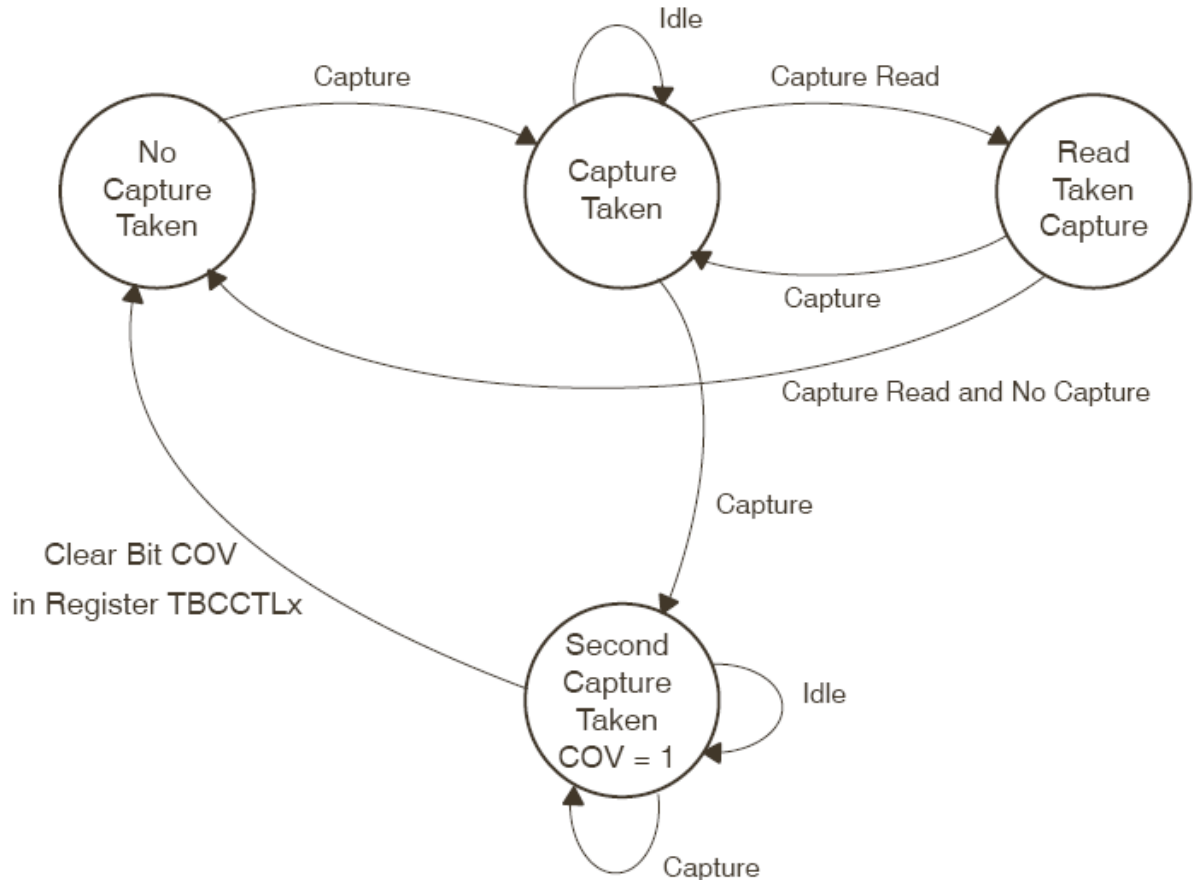
在任何时刻,可以通过 CCI 位读取输入信号的电平。MSP430x2xx 系列的器件允许 CCI_xA 和 CCI_xB 连接于不同的信号(请参考器件手册)。捕获信号可能会和定时器时钟不同步,并导致竞争条件的发生。将 SCS 位置位可以在下个定时器时钟使捕获同步,一般建议将之置位,见图例 9-10

图 9-10. 捕获信号 (SCS=1)



如果第二次捕获在第一次捕获的值被读取之前发生,捕获比较寄存器就会产生一个溢出逻辑,COV 位在此时置位,如图 9-11,COV 位必须软件清除。

图 9-11.捕获循环



通过软件初始化捕获

捕获可以由软件初始化。CMx 位可以配置捕获的触发沿。CCIS1 和 CCIS0 位可以捕获电压在 VCC 和 GND 之间的信号，初始化捕获器举例

```
MOV #CBP+SCS+CCIS1+CM_3,&TBCCTLx ; 配置 TBCCTLx
```

```
XOR #CCIS0,&TBCCTLx ; TBCCTLx = TBR
```

比较模式

比较模式通过将 CBP=0 来进入。比较模式用于选择 PWM 输出信号或在特定的时间间隔中断。当 TBR 计数到 TBCLx 的值时：

- I 中断标志 CCIFG=1;
- I 内部信号 EQUx=1;
- I EQUx 根据输出模式来影响输出信号

比较锁存 TBCLx

The TBCCR_x compare latch, TBCL_x, holds the data for the comparison to the timer value in compare mode. TBCL_x is buffered by TBCCR_x. The buffered compare latch gives the user control over when a compare period updates. The user cannot directly access TBCL_x. Compare data is written to each TBCCR_x and automatically transferred to TBCL_x. The timing of the transfer from TBCCR_x to TBCL_x is user-selectable with the CLLD_x bits as described in Table 9-2.

TBCCR_x的比较锁存器TBCL_x,在比较模式中为定时器值的比较保持数据。TBCL_x由TBCCR_x缓冲。在比较周期更新时,锁存缓冲令用户便于控制。用户不必直接访问TBCL_x,比较数据写入每个TBCCR_x后自动传递到TBCL_x中。从TBCCR_x到TBCL_x传递的时间可以通过CLLD_x位定义,如表9-2

表9-2 TBCL_x加载事件

CLLD _x	说明
00	当更新的数据写入TBCCR _x 时立即加载到TBCL _x 中
01	当TBR计数到0时, TBCCR _x 的值加载到TBCL _x
10	对于增模式和连续模式,当TBR计数到0时, TBCCR _x 的值加载到TBCL _x ; 对于增减模式,当TBR计数到原来的TBCL ₀ 值或0时, TBCCR _x 的值加载到TBCL _x
11	当TBR计数到原来的TBCL _x 值或0时, TBCCR _x 的值加载到TBCL _x

为比较锁存器编组

多个比较锁存寄存器可以通过TBCLGRP_x编组,以便于同步更新。编组后,同组中序号最小的CCR_x的CLLD_x位确定全组的加载事件,包括7个比较寄存器在同一组的情况(TBCLGRP=3),如图9-3。控制TBCCR_x的CLLD_x位不能为0,如果为0的话。所有的比较锁存器就会在他们对应的TBCCR_x写入时立即更新——也就是说不存在比较锁存器编组了。

如果采用分组方法,则需要加载的必须存在2个条件,第1,该组的所有TBCCR_x寄存器必须都更新(即使是TBCCR_x的数据=原来的TBCCR_x数据);第2,加载事件必须发生。

表 9-3. 比较锁存器操作模式

TBCLGRP _x	编组	控制更新的寄存器
00	无	独立控制
01	TBCL1+TBCL2 TBCL3+TBCL4 TBCL5+TBCL6	TBCCR1 TBCCR3 TBCCR5
10	TBCL1+TBCL2+TBCL3 TBCL4+TBCL5+TBCL6	TBCCR1 TBCCR4
11	TBCL0+TBCL1+TBCL2+TBCL3+TBCL4+TBCL5+TBCL6	TBCL0

9.2.5 输出单元

每个捕获比较模块包含一个输出单元。输出单元用于产生如 PWM 这样的信号。每个输出单元可以根据 EQU0 和 EQU_x 产生 8 种模式的信号。TBOUTH 引脚功能可以用于将所有的 Timer_B 输出为高阻抗状态。当 TBOUTH 脚选用此功能并该引脚被上拉时,所有的 Timer_B 输出为高阻抗状态。

输出模式

输出模式由 OUTMOD_x 位来确定,如表 9-4。对于所有模式来说(除了模式 0),OUT_x 信号随着定时器时钟的上升沿而改变。输出模式 2, 3, 6 和 7 对输出单元 0 无效,因为在这些模式下, EQU_x = EQU0。

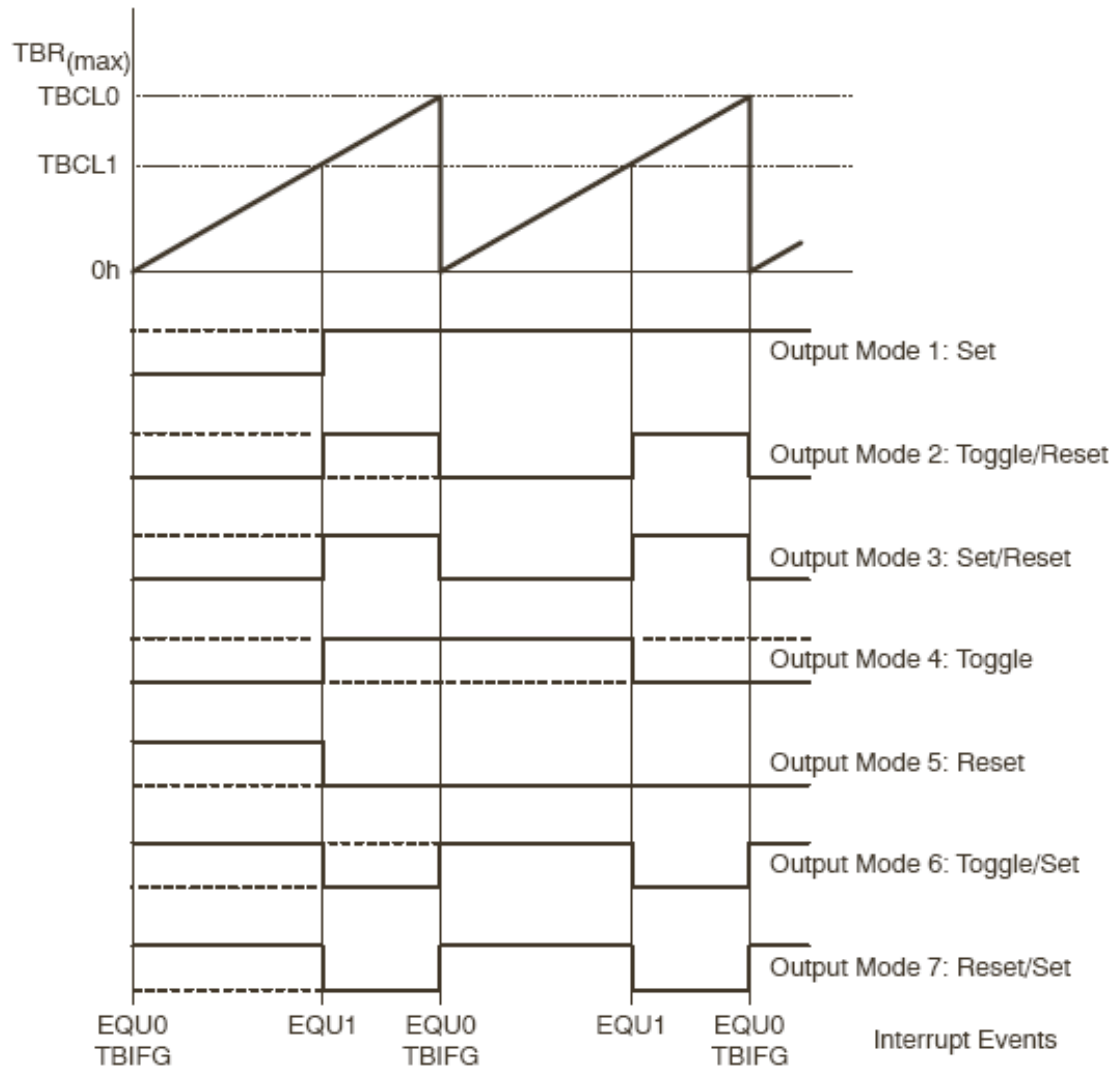
表 9-4. 输出模式

OUTMODx	模式	说明
000	输出	输出信号 OUTx 由 OUTx 位定义。当 OUTx 位更新时，OUTx 信号立刻更新
001	置位	当定时器计数到 TBCLx 值时，输出置位，并保持置位直到定时器复位或选择了另一个输出模式。
010	翻转/复位	当定时器计数到 TBCLx 值时，输出翻转。当定时器计数到 TBCL0 值时，输出复位。
011	置位/复位	当定时器计数到 TBCLx 值时，输出置位。当定时器计数到 TBCL0 值时，输出复位。
100	翻转	当定时器计数到 TBCLx 值时，输出翻转。输出信号的周期是定时器周期的 2 倍。
101	复位	当定时器计数到 TBCLx 值时，输出复位，并保持复位直到选择了另一个输出模式。
110	翻转/置位	当定时器计数到 TBCLx 值时，输出翻转。当定时器计数到 TBCL0 值时，输出置位。
111	复位/置位	当定时器计数到 TBCLx 值时，输出复位。当定时器计数到 TBCL0 值时，输出置位。

输出举例——定时器处于增模式

当定时器计数到 TBCLx 的值或从 TBCL0 到 0 时，OUTx 信号根据输出模式而改变。如图 9-12 使用了 TBCL0 和 TBCCR1

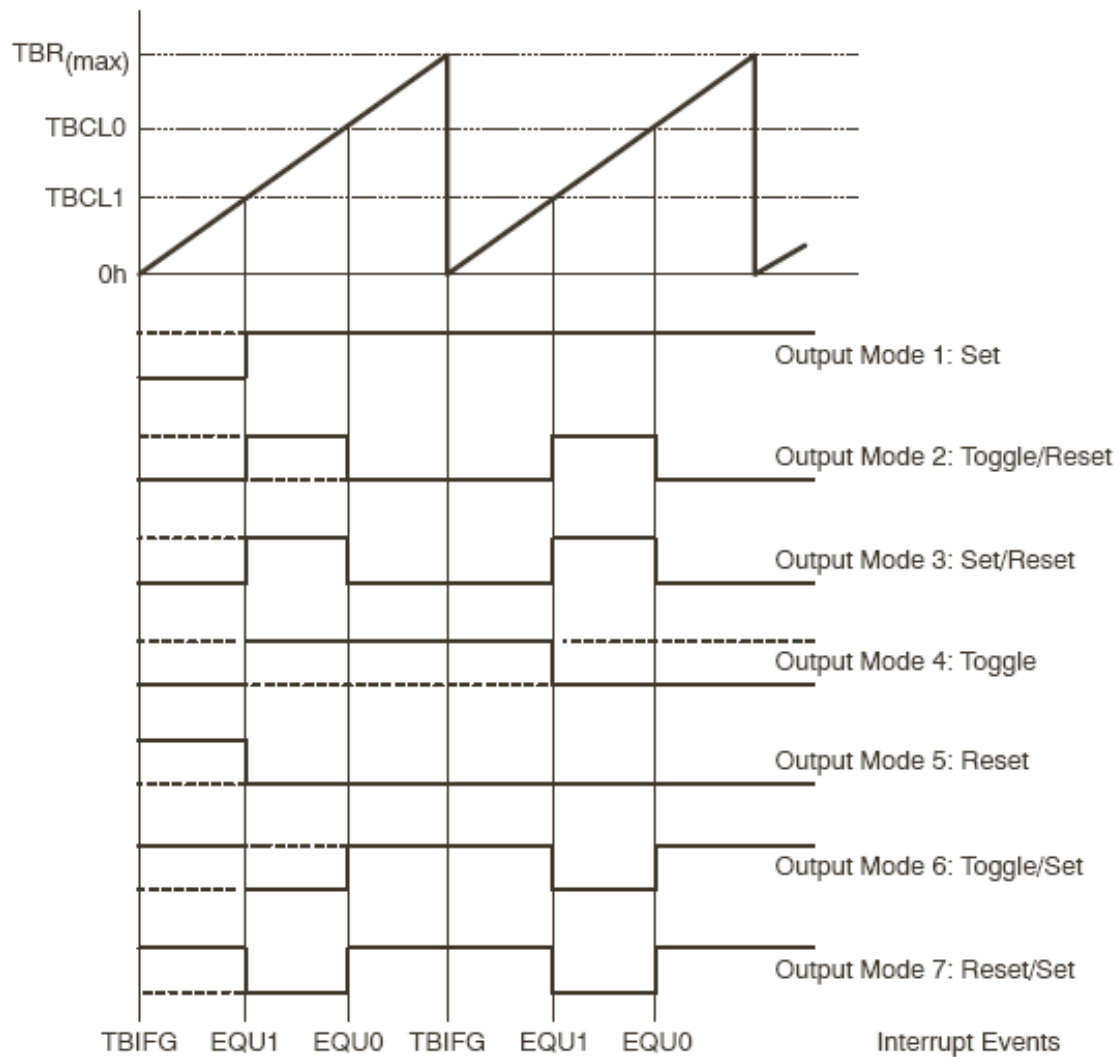
图 9-12. 输出举例——定时器处于增模式



输出举例——定时器处于连续模式

当定时器计数到 $TBCL_x$ 和 $TBCL_0$ 时， OUT_x 信号按选择的输出模式发生改变。如图 9-13 所示，使用了 $TBCL_0$ 和 $TBCL_1$

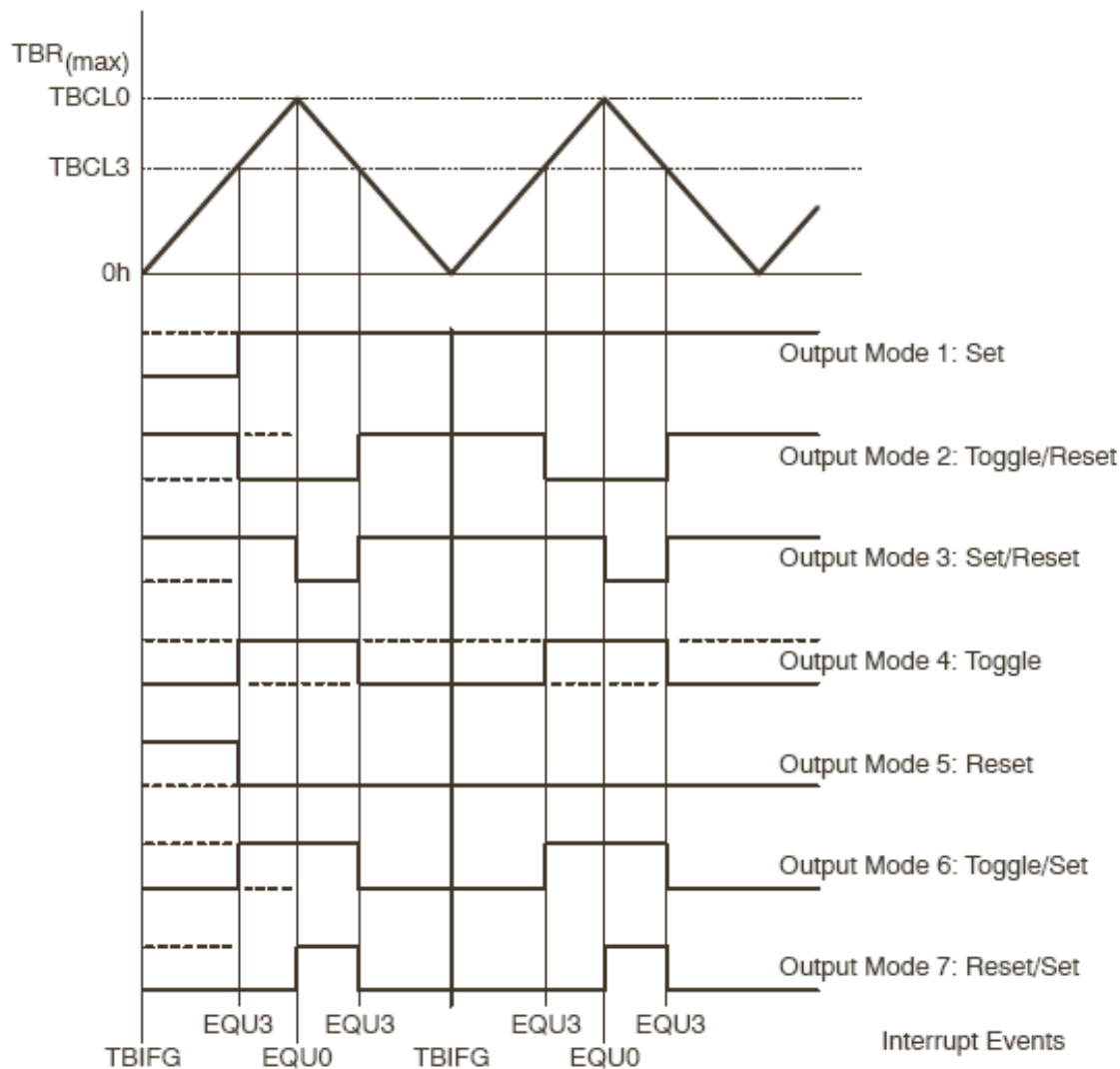
图 9-13. 输出举例——定时器处于连续模式



输出举例——定时器处于增减模式

当定时器的值在任一计数方向上出现了等于 TBCLx 和等于 TBCL0 的值时，OUTx 信号按选择的输出模式发生改变，如图 9-14，使用了 TBCL0 和 TBCL3

图 9-14. 输出举例——定时器处于增减模式



注意：输出模式的切换

当需要在输出模式之间进行切换时， $OUTMOD_x$ 的一个位必须在过度时保持置位，除非是切换到模式 0，否则会由于或非门解码输出模式 0 而导致出现脉冲干扰。输出模式之间的安全切换的方法之一是用输出模式 7 作为过度状态，例如：

`BIS #OUTMOD_7,&TBCCTLx` ; 设置为输出模式 7

`BIC #OUTMODx,&TBCCTLx` ; 清除不需要的位

9.2.6 定时器 B 的中断

16 位定时器 B 有 2 个中断向量：

- I TBCCR0 的 TBCCR0 CCIFG 中断向量
- I 所有其他 CCIFG 和 TBIFG 的 TBIV 中断向量

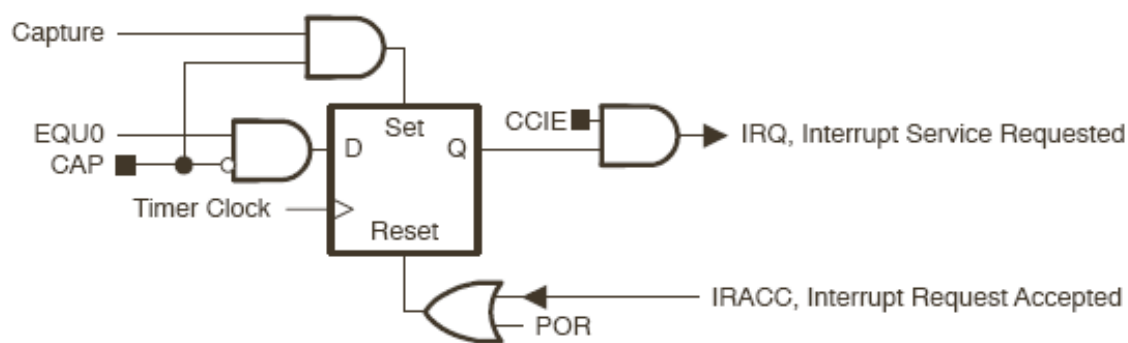
在捕获模式下，当一个定时器的值捕获到相应的 TBCCR_x 寄存器时，CCIFG 标志置位；在比较模式下，如果 TBR 计数到相应的 TBCCR_x 值时，CCIFG 标志置位。软件可以清除或置位任何一个 CCIFG 标志。当相应的 CCIE 和 GIE 置位时，CCIFG 标志就会产生一个中断。

TBCCR0 中断

TBCCR0 CCIFG 标志拥有定时器 B 的最高中断优先级，并有一个专用的中断向量，如图 9-15。当进入 TBCCR0 中断后，TBCCR0 CCIFG 标志自动复位。

如图 9-15。

图 9-15 捕获比较 TBCCR0 中断标志



TBIV, 中断向量发生器

TBCCR_x CCIFG 和 TBIFG 标志共用一个中断向量（不包括 TBCCR0 CCIFG）。中断向量寄存器 TBIV 用于确定它们中的哪个要求响应中断。最高优先级的中断（不包括 TBCCR0 CCIFG）在 TBIV 寄存器中产生一个数字（见寄存器说明），这个数字是规定的数字，可以在程序中识别并自动进入相应的子程序。禁止定时器 B 中断不会影响 TBIV 的值。

对 TBIV 的读写会自动复位最高优先级的挂起中断标志。如果另一个中断标志置位，在结束原先的中断响应后会，该中断响应应立即发生。例如，当中断服务子程序访问 TBIV 时，如果 TBCCR1 和 TBCCR2 CCIFG 标志位置位，TBCCR1 CCIFG 自动复位。在中断服务子程序的 RETI 命令执行后，TBCCR2 CCIFG 标志会产生另一个中断。

TBIV 软件示例

以下软件说明了 TBIV 的使用和操作。TBIV 的值加入 PC 指针来自动跳转到相应的子程序。右边空白处的数字表明 CPU 每条指令需要的周期。不同的中断源的软件包含中断响应时间和返回中断周期，但并不包含任务本身的执行时间。响应时间定义为：

- I 捕获比较模块 TBCCR0 为 11 个时钟周期
- I 捕获比较模块 TBCCR1-6 为 16 个时钟周期
- I 定时器溢出标志 TBIFG 置位为 14 个时钟周期

以下程序举例说明了 TBIV 和 timer_3 的使用

```

; TBCCR0 CCIFG. 的中断处理
CCIFG_0_HND
; ... ; 中断响应开始
RETI
; TBIFG, TBCCR1 和 TBCCR2 CCIFG.的中断处理
TB_HND ... ; 中断响应
ADD &TBIV,PC ; 加偏移量跳转到标号 table_3
RETI ; 中断 0: 无中断
JMP CCIFG_1_HND ; 中断 2: 模块 1
JMP CCIFG_2_HND ; 中断 4: 模块 2
RETI ; 中断 6
RETI ; 中断 8
RETI ; 中断 10
RETI ; 中断 12
TBIFG_HND ; 中断 14: TIMOV Flag
... ; 开始任务
RETI
CCIFG_2_HND ; 中断 4: 模块 2
... ; 开始任务
RETI ; 返回主程序
CCIFG_1_HND ; 中断 2: 模块 1
... ; 开始任务
JMP TB_HND ; 寻找中断域

```

9.3 定时器 B 寄存器

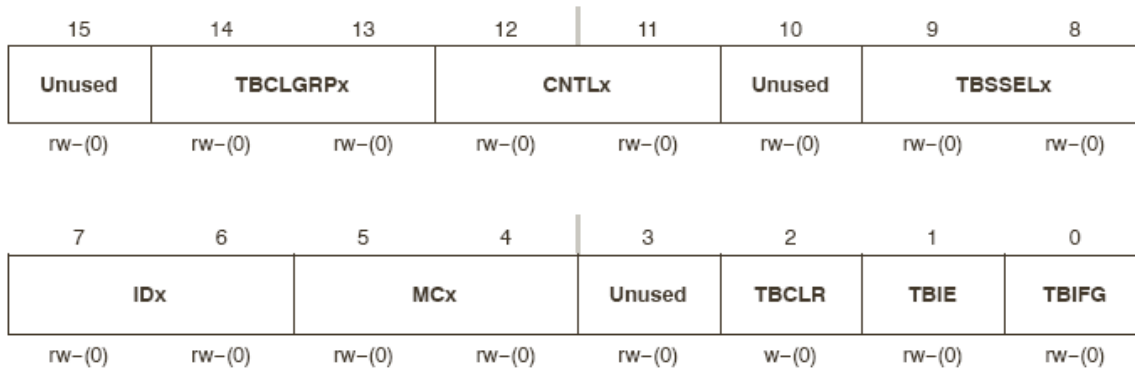
定时器 B 寄存器如表 9-1: († MSP430x20xx 中不存在)

表 9-3. 定时器 B 寄存器

寄存器	简写	寄存器类型	地址	初始状态
TB 控制寄存器	TBCTL	读/写	0180h	POR 复位
TB 计数器	TBR	读/写	0190h	POR 复位
TB 捕获比较控制寄存器 0	TBCCTL0	读/写	0182h	POR 复位
TB 捕获比较 0	TBCL0	读/写	0192h	POR 复位
TB 捕获比较控制寄存器 1	TBCCTL1	读/写	0184h	POR 复位
TB 捕获比较 1	TBCCR1	读/写	0194h	POR 复位
TB 捕获比较控制寄存器 2	TBCCTL2†	读/写	0186h	POR 复位
TB 捕获比较 2	TBCCR2†	读/写	0196h	POR 复位
TB 捕获比较控制寄存器 3	TBCCTL3	读/写	0188h	POR 复位
TB 捕获比较 3	TBCL3	读/写	0198h	POR 复位
TB 捕获比较控制寄存器 4	TBCCTL4	读/写	018Ah	POR 复位

TB 捕获比较 4	TBCCR4	读/写	019Ah	POR 复位
TB 捕获比较控制寄存器 5	TBCCTL5†	读/写	018Ch	POR 复位
TB 捕获比较 5	TBCCR5†	读/写	019Ch	POR 复位
TB 捕获比较控制寄存器 6	TBCCTL6†	读/写	018Eh	POR 复位
TB 捕获比较 6	TBCCR6†	读/写	019Eh	POR 复位
TB 中断向量寄存器	TBIV	只读	011Eh	POR 复位

TBCTL, Timer_B 控制寄存器



Unused	15	未使用位
TBCLGRP	14-13	TBCL _x 编组 00 —— 每个 TBCL _x 锁存器独立加载 01 —— TBCL1+TBCL2 (TBCCR1 CLLD _x 位控制更新) TBCL3+TBCL4 (TBCCR3 CLLD _x bits位控制更新) TBCL5+TBCL6 (TBCCR5 CLLD _x bits位控制更新) TBCL0 独立 10 —— TBCL1+TBCL2+TBCL3 (TBCCR1 CLLD _x bits位控制更新) TBCL4+TBCL5+TBCL6 (TBCCR4 CLLD _x bits位控制更新) TBCL0 独立 11 —— TBCL0+TBCL1+TBCL2+TBCL3+TBCL4+TBCL5+TBCL6 (TBCCR1 CLLD _x bits位控制更新)
CNTL _x	12-11	计数长度 00 16-bit, TBR _(max) = 0FFFFh 01 12-bit, TBR _(max) = 0FFFh 10 10-bit, TBR _(max) = 03FFh 11 8-bit, TBR _(max) = 0FFh
Unused	10	未使用位
TBSSEL _x	9-8	TB 时钟源选择 00 TBCLK 01 ACLK 10 SMCLK 11 TBCLK 的反相信号
ID _x	7-6	输入分频。这些位为输入时钟分频选择 00 /1 01 /2

		10 /4 11 /8
MCx	5-4	模式控制，当 TB 不用于节省功耗时，将 MCx=00h 00 停止模式:定时器停止 01 增模式:定时器计数到 TBCL0 10 连续模式: 定时器计数到 TBmax 11 增减模式: 定时器计数到 TBCL0 然后减到 0000h
Unused	3	未使用位
TBCLR	2	定时器清零位。该位置位会复位 TBR，时钟分频和计数方向。TBCLR 位会自动复位并读出值为 0
TBIE	1	TB 中断允许。改位允许 TBIFG 中断请求 0 中断禁止 1 中断允许
TBIFG	0	TB 中断标志位 0 无中断挂起 1 中断挂起

TBR, Timer_B 寄存器



TBRx 位 15-0

Timer_B 寄存器. TBR 寄存器是 Timer_B 的计数器。

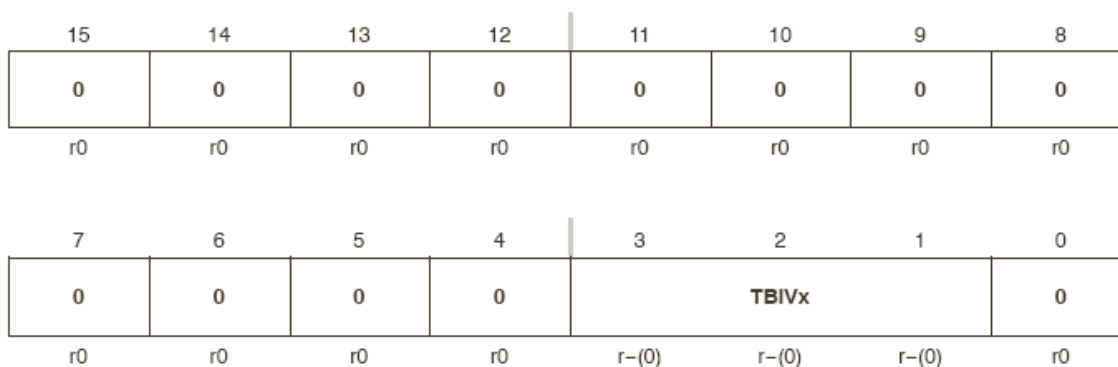
TBCCTLx, 捕获比较控制寄存器

15	14	13	12	11	10	9	8
CMx		CCISx		SCS	CLLDx		CAP
rw-(0)		rw-(0)		rw-(0)	r-(0)		rw-(0)
7	6	5	4	3	2	1	0
OUTMODx			CCIE	CCI	OUT	COV	CCIFG
rw-(0)			rw-(0)	r	rw-(0)	rw-(0)	rw-(0)

CMx	15-14	捕获模式 00 不捕获 01 上升沿捕获 10 下降沿捕获 11 上升和下降沿都捕获
CCISx	13-12	捕获比较选择，该位选择 TBCCR _x 的输入信号，详见器件手册 00 CCI _x A 01 CCI _x B 10 GND 11 VCC
SCS	11	同步捕获源，该位用于将捕获通信和时钟同步 0 异步捕获 1 同步捕获
CLLDx	10-9	比较锁存加载，该位选择比较锁存加载事件 00 ——TBCCR _x 写入时TBCL _x 加载 01 ——TBR计数到0时TBCL _x 加载 10 ——TBR计数到0时TBCL _x 加载(增模式或连续模式时) TBR计数到0时TBCL ₀ 或 0时加载 (增减模式) 11 ——TBR 计数到 TBCL _x 时 TBCL _x 加载
CAP	8	捕获模式 0 比较模式 1 捕获模式
OUTMODx	7-5	输出模式位。由于在模式 2, 3, 6 和 7 下 EQU _x = EQU ₀ ，因此这些模式对 TBCL ₀ 无效 000 OUT 位的值 001 置位 010 翻转/复位 011 置位/复位 100 翻转 101 复位 110 翻转/置位 111 复位/置位
CCIE	4	捕获比较中断允许位，该位允许相应的 CCIFG 标志中断请求 0 中断禁止 1 中断允许

CCI	3	捕获比较输入。所选择的输入信号可以通过该位读取
OUT	2	对于输出模式 0，该位直接控制输出状态 0 输出低电平 1 输出高电平
COV	1	捕获溢出位。该位表示一个捕获溢出发生。COV 必须由软件复位。 0 没有捕获溢出发生 1 有捕获溢出发生
CCIFG	0	捕获比较中断标志位 0 没有中断挂起 1 有中断挂起

TBIV, Timer_B 中断向量寄存器



TBIVx Bits 15-0 Timer_B 中断向量值

TBIV 的内容	中断源	中断标志	中断优先级
00h	无中断挂起	—	
02h	捕获比较 1	TBCCR1 CCIFG	最高
04h	捕获比较 2	TBCCR2 CCIFG	
06h	捕获比较 3 (不是所有的器件都有)	TBCCR3 CCIFG	
08h	捕获比较 4 (不是所有的器件都有)	TBCCR4 CCIFG	
0Bh	捕获比较 5 (不是所有的器件都有)	TBCCR5 CCIFG	
0Ch	捕获比较 6 (不是所有的器件都有)	TBCCR6 CCIFG	
0Eh	定时器溢出	TBIFG	最低



MSP430F22x4 开发板

专业提供 **MSP430** 单片机开发工具