



## MSP430F5 系列 16 位超低功耗单片机模块原理

### 第 12 章 Timer A 定时器 A

版本: 1.1

日期: 2008.10. 最后更新日期:2010.8.

原文: TI slau208.pdf (5xxfamily User's Guide)

翻译: 许俊超 郑州市 软件工程师

编辑: DC 微控网总版主

注: 以下文章是翻译 TI slau208.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。

文章更新详情请密切留意微控技术论坛。

## 第 12 章 Timer A

定时器 A 是一个复合了捕获/比较寄存器的十六位的定时/计数器。本章介绍 MSP430X5XX 的定时器 A。

章节

12.1 定时器 A 介绍

12.2 定时器 A 操作

12.3 定时器 A 寄存器

### 12.1 定时器 A 介绍

定时器 A 是一个十六位的定时/计数器，其捕获/比较寄存器多达七个。定时器 A 支持支持多重捕获/比较，PWM 输出和内部定时。定时器还有扩展中断功能，中断可以由定时器溢出产生或由捕获/比较寄存器产生。

定时器 A 的特性包括：

- 四种运行模式的异步 16 位定时/计数器
- 可选择配置的的时钟源
- 多达七个可配置的捕获/比较寄存器
- 可配置的 PWM 输出
- 异步输入和输出锁存
- 对所有 TA 中断快速响应的中断向量寄存器

定时器 A 的结构图如图 12-1

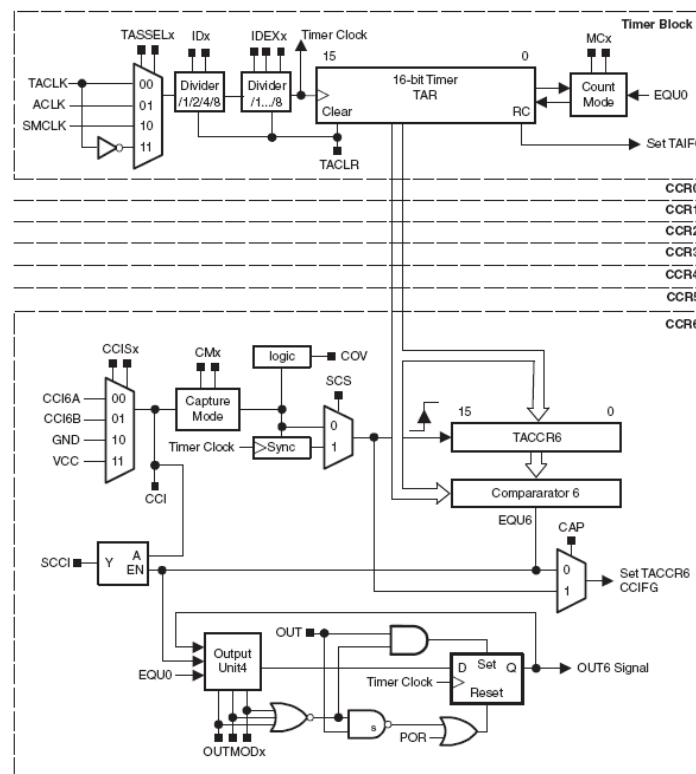


图 12-1 定时器 A 的结构图

### 12.2 TIMER\_A 操作方法

TIMERA 模块由用户软件来配置，TIMERA 的使用在下面的章节中讨论。

#### 12.2.1

16 位定时/计数器寄存器 **TAR**，随着时钟信号的第个上升沿增/减（由模式所决定）。**TAR** 可以由软件读写。除此之外，定时器溢出时可以产生中断。**TAR** 可以通过设置 **TACLR** 位来清除。在 **UP/DOWN** 模式下，设置 **TACLR** 也可以清除时钟分频器和计数方向。

注意：修改 **TIMERA** 寄存器

建议在修改定时器运行模式前先停止定时器（中断使能、中断标志、**TACLR** 例外），以避免产生求知的误操作。

当 **TACLK** 与 **CPU** 时钟不同步时，当定时器没有运行时 **TAR** 的读将产生，其结果也是不可预料的。可以在定时器运行时多读几次，通过软件表决的方式来原因正确的读数。对 **TAR** 任何的写将是立即生效。

### 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** 或者或者外部源 **TACLK**。时钟源由 **TASSELx** 位来选择，通过 **IDEXx** 所选择的时钟源可以直接通过定时器或者分为 2, 4, 8 分频，选择的时钟源可以使用 **IDx** 位进一步分为 2, 3, 4, 5, 6, 7, 8 分频。**TACLR** 置位时，分频器复位。

注意：**TIMER\_A** 分频器

设置 **TACLR** 位将清除 **TAR** 即分频器的内容。当 **TACLR** 位被清除时，定时器时钟将在所选择的 **TASSELx** 位 **TIMER\_A** 时钟源的第一个时钟上升沿开始计时，并持续到分频器设置 **IDx** 和 **IDEXx** 位置位。

### 12.2.2 开始计数器

定时器可以通过下列方式来启动或重新启动：

○当定时器计数到 **MCXZ** 0 并且时钟源处于活动状态

○当定时器模式为 **up** 或 **up/down** 模式时（即单调增和培养模式），定时器可以通过写 0 到 **TACCR0** 业停止计数。定时器可以通过写 不念旧恶非 0 值来重新开始计数。在这种情况下，定时器从 0 开始增计数。

1

### 2.2.3 定时器模式控制

定时器有 4 种操作模式，见表 12-1，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 12-1 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 <b>TACCR0</b> 的值
10	连续模式	定时器循环地从 0 连续增加到 <b>0FFFFH</b>
11	增减模式	定时器循环地从 0 增到 <b>TACCR0</b> 的值再连续减至 0

### 单调增模式

单调增模式用于计数周期不是 **0FFFFH** 的情况。定时器重复增计数值寄存器 **TACCR0** 的值，而 **TACCR0** 的值取决于定时周期，如图 12-2，定时器周期为 **TACCR0+1**。当定时器的值等于 **TACCR0** 时，定时器就回到 0 重新计数。如果当定时器的值大于 **TACCR0**，而此时选择单调增模式，定时器立即从 0 重新开始计数。

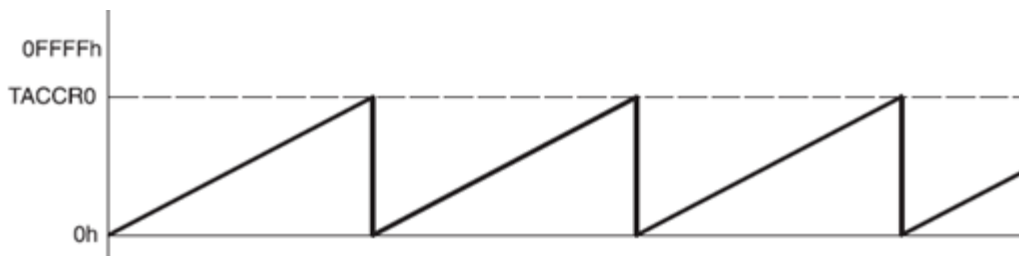


图 12-2 单调增模式

当定时器计数到 TACCR0 的值时，中断标志 CCIFG 位置位。当定时器由 TACCR0 返回 0 时，TAIFG 中断标志置位。图 12-3 说明了标志循环。

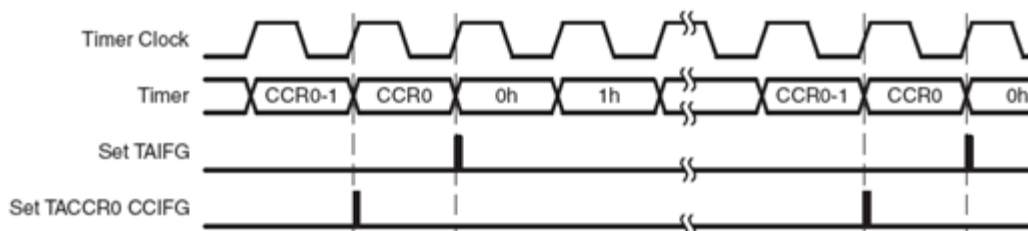


图 12-3 单调增模式标志位的变化

### 修改周期寄存器 TACCR0

在定时器在运行时修改 TACCR0，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 0FFFFh，然后重新从 0 开始增计数，如图 12-4。捕获/比较寄存器 TACCR0 以及其他捕获比较器工作方式一样。

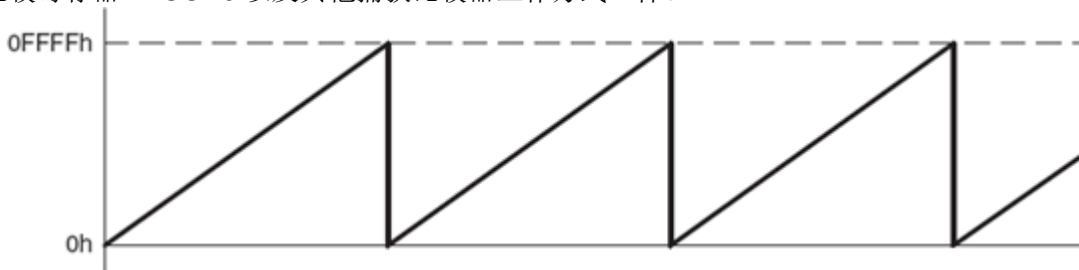


图 12-4 连续模式

当定时器从 0FFFFh 到 0 时，TAIFG 中断标志置位。图 12-5 表示了标志位的设置

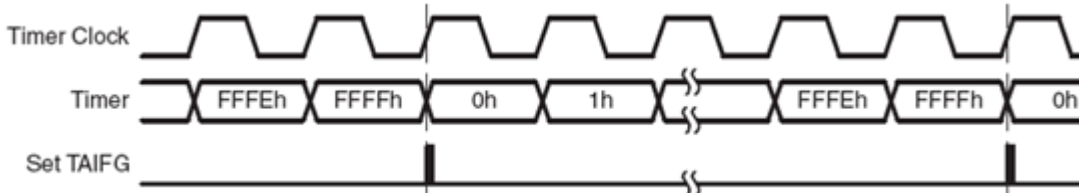


图 12-5 连续模式标志位的设置

### 连续模式的使用

连续模式可以用于产生独立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了程序时写入 TACCRX。图 12-6 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，时间间隔由硬件而不是软件控制，与中断响应没有冲突。如果想产生多于 3 个时间间隔可以使用所有的捕获比较寄存器。

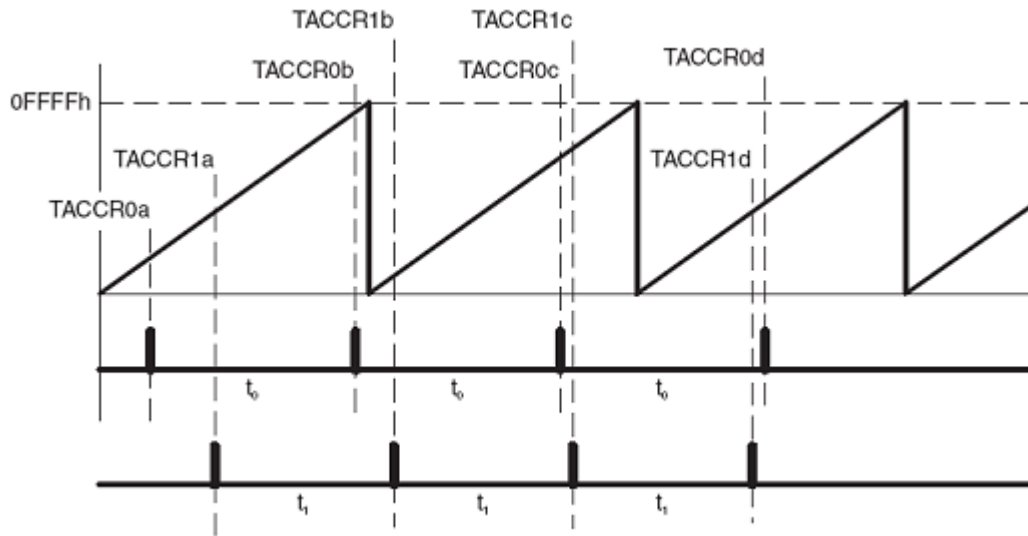


图 12-6 连续模式时间间隔

时间间隔可以由其他模式产生，TACCR0 也可以作为周期寄存器使用。如果旧的 TACCR0X 的数据之和与新的相比，要比 TACCR0 大，那么操作就会复杂得多。当旧的 TACCRX 的值加上  $t_x$  比 TACCR0 的值大，那么 TACCR0 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 0FFFFh 且需要产生对称的脉冲时使用。定时器增计数到 TACCR0 再从 TACCR0 减计数到 0，如图 12-7，周期是 TACCR0 值的 2 倍。

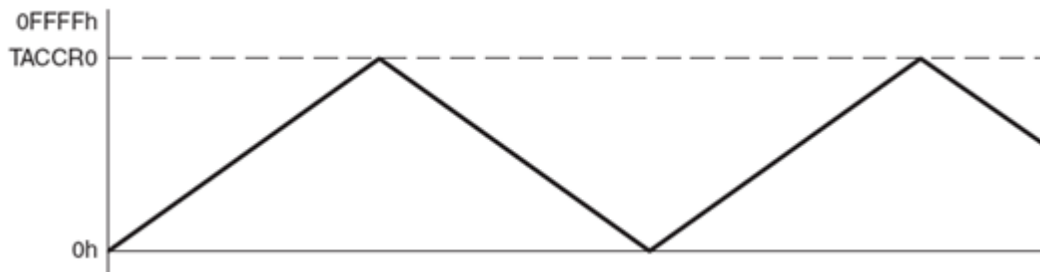


图 12-7 增减模式

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，它就沿着停止时的计数方向和数值开始计数。如果不希望这样，就需要将 TACLR 置位来清除方向。TACLR 位也会清除 TAR 的值和定时器的时钟分频。

在增减模式，在 TACCR0 中，CCIFG 中断标志和 TAIFG 中断标志在一个周期中只能置位一次，由 1/2 定时器周期隔开。当定时器计数到由 TACCR0-1 变到 TACCR0 时，CCIFG 置位；而定时器完成减计数从 0001h 到 0000h 时，TAIFG 置位。图 12-8 表示了标志位的置位状况。

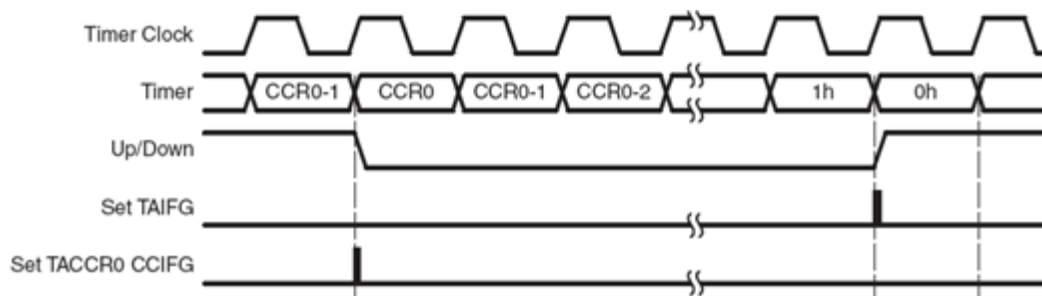


图 12-8 增减模式下的标志位

### 改变 TACCR0 周期寄存器

当定时器运行时，改变 TACCR0 的值，如果正处于减计数的情况，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，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；如果正牌增计数状态，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时间的应用（参阅 TIMERA 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 12-9 的例中，tdead 为

$$T_{dead} = t_{timer} * (TACCR1 - TACCR2)$$

T<sub>dead</sub>—同时输出时必须没有反应的时间段

T<sub>timer</sub>—定时器时钟周期

TACCRX—捕获比较寄存器 X 的内容。

TACCRX 寄存器并不是缓冲，写入时立即更新，因此，任何所要求的死区时间不会自动保留。

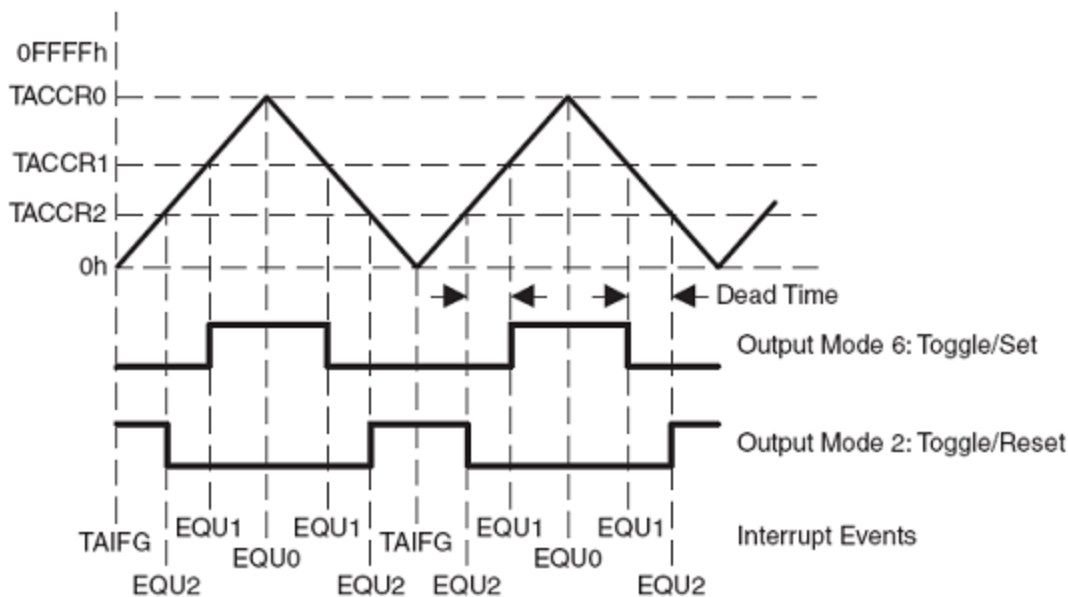


图 12-9 增减模式的输出

### 12.2.4 捕获/比较模块

定时器 A 中有 3 个或 5 个相同的捕获比较模块 TACCRX，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。

#### 捕获模块

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  $CCIXA$  和  $CCIXB$  连接外部的引脚或内部的信号，这通过  $CCISX$  位来选择。 $CMX$  位选择捕获输入信号触发沿；上升沿、下降沿或两者都捕获。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：

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

内部信号可以在任一时间通过  $CCI$  位来读取。 $MSP430X5XX$  系列的器件允许  $CCIXA$  和  $CCIXB$  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，见图例 12-10

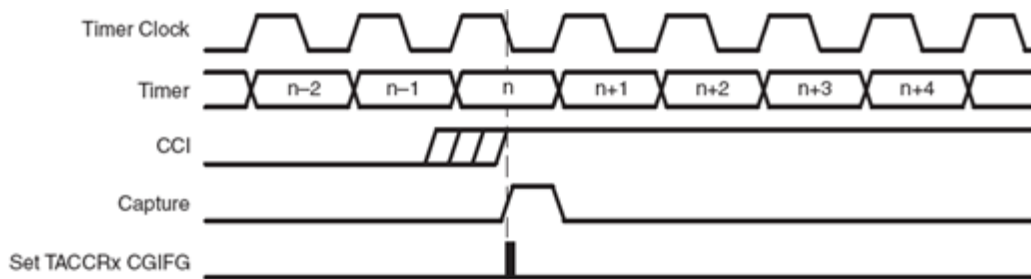


图 12-10 捕获信号 ( $scs=1$ )

如果一个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 12-11， $COV$  位必须软件清除。

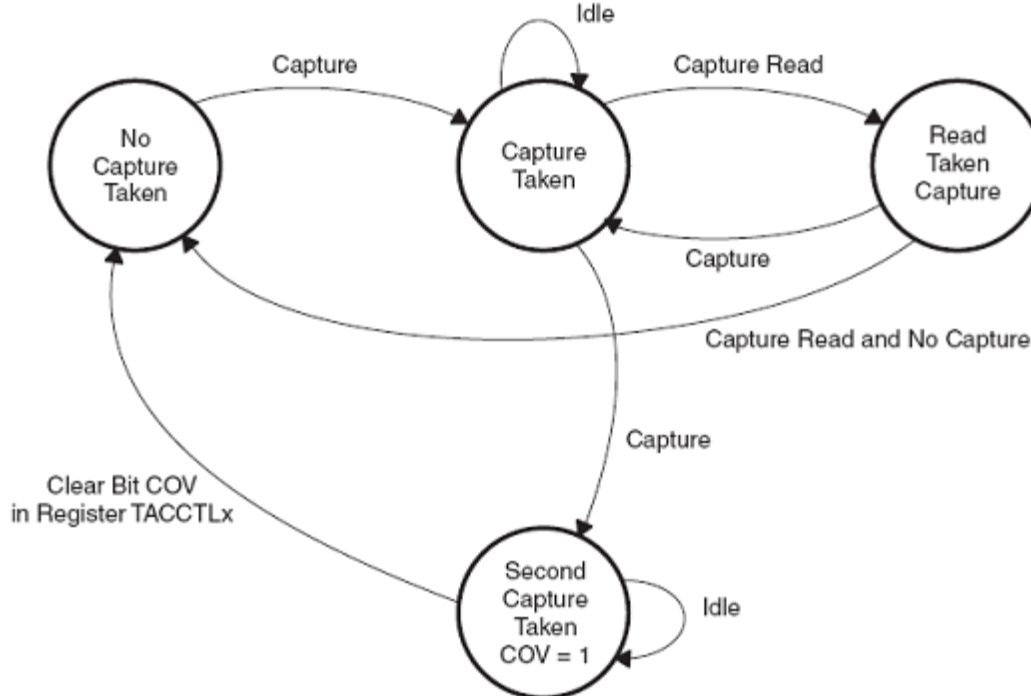


图 12-11 捕获周期

#### 12.2.4.0.1 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。 $CCIS1=1$  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 #CAP+SCS+CCIS+CM_3, &TACCTLx;配置 TACCTLx
XOR #CCIS0,&TACCTLx; TACCTLx=TAR
```

### 比较模式

CAP=0 时选择比较模式。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 TAR 计数到 TACCRx 的值时：

- 中断标志 CCIFG 置位；
- 内部信号 EQUx=1
- EQUx 根据输出模式来影响输出信号
- 输入信号 CCI 锁存到 SCCI

### 12.2.5 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 EQUx 产生 8 种模式的信号。

### 输出模式

输出模式由 OUTMODx 位来确定，如表 12-2。对于所有模式来说（模式 0 除外），OUTx 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，EQUx=EQU0。

表 12-2 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 TACCRX 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 TACCRX 值时，输出翻转。当定时器计数到 TACCR0 值时，输出复位
011	置位/复位	当定时器计数到 TACCRX 值时，输出置位。当定时器计数到 TACCR0 值时，输出复位
100	翻转	当定时器计数到 TACCRX 值时，输出翻转。输出信号的周期是定时器的 2 倍
101	复位	当定时器计数到 TACCRX 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 TACCRX 值时，输出翻转。当定时器计数到 TACCR0 值时，输出置位
111	复位/置位	当定时器计数到 TACCRX 值时，输出复位。当定时器计数到 TACCR0 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 TACCRX 的值或从 TACCR0 到 0 时，OUTx 信号根据输出模式而改变。使用 TACCR0 和 TACCR1 如图 12-12 所示。



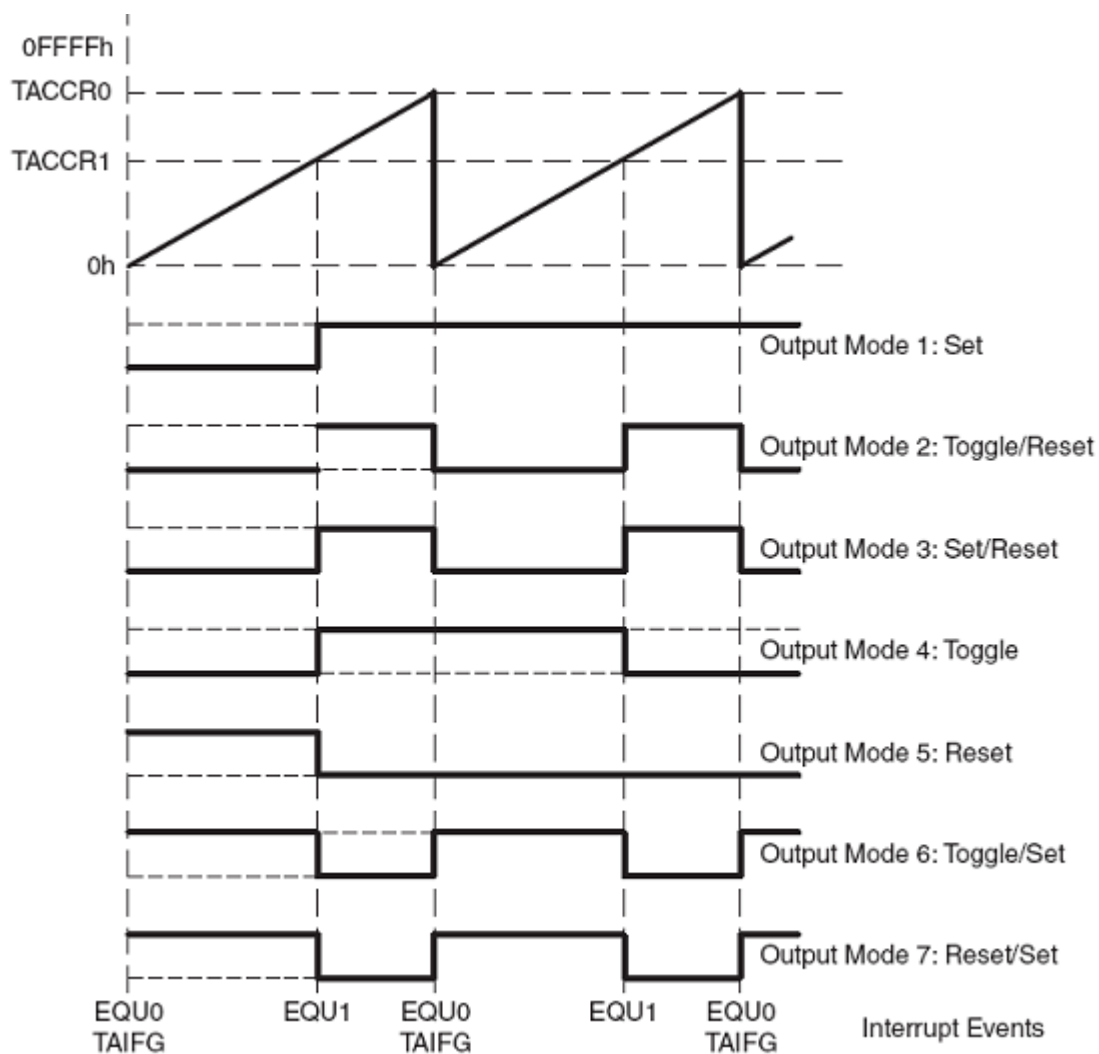
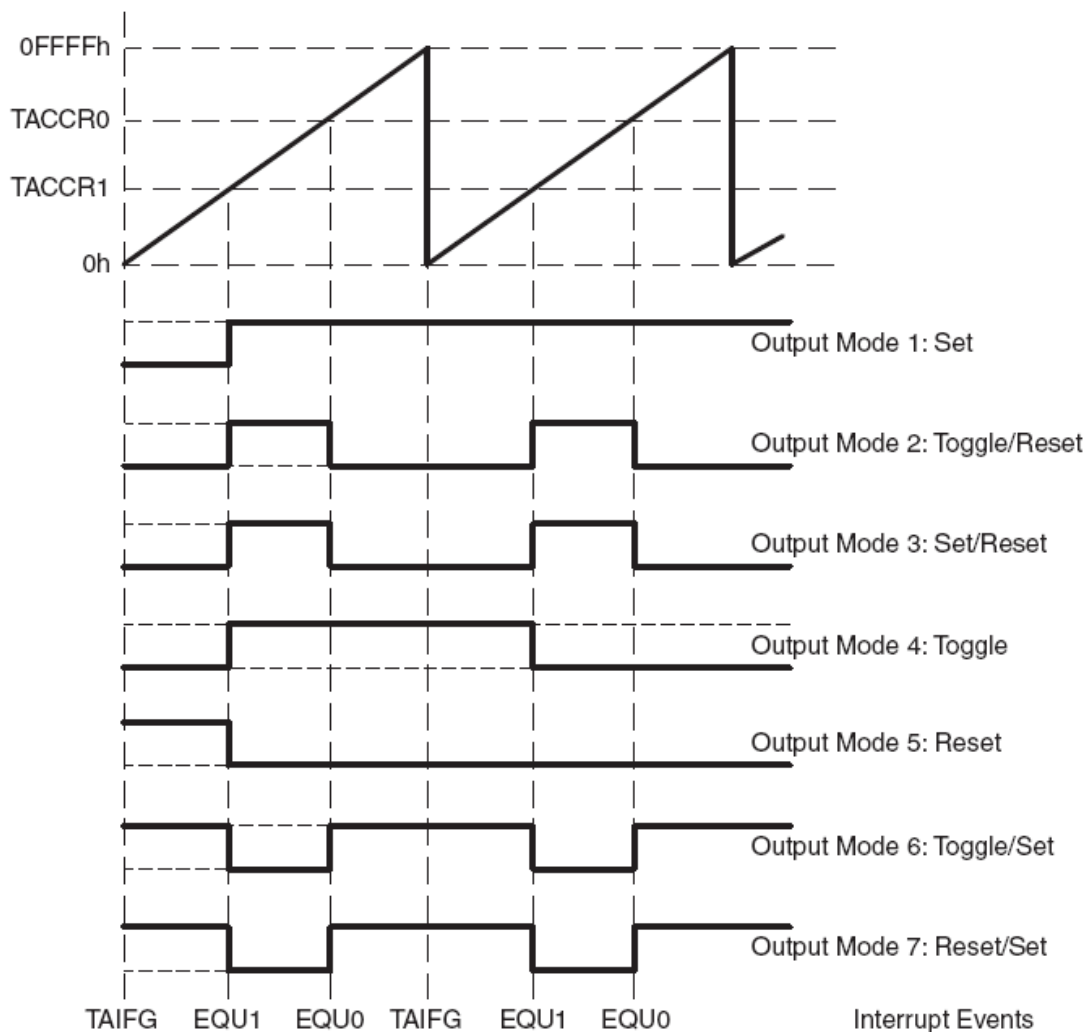


图 12-2 输出举例—定时器处于增模式

#### 输出举例—定时器处于连续模式

当定时器计数到 TACCRX 和 TACCR0 时，OUTx 信号按选择的输出模式发生改变。图 12-13 是使用 TACCR0 和 TACCR1 的例子



**Figure 12-13. Output Example—Timer in Continuous Mode**

图 12-13 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 TACCRX 和等于 TACCR0 的值时，OUTX 信号按选择的输出模式发生改变，使用方法如图 12-14

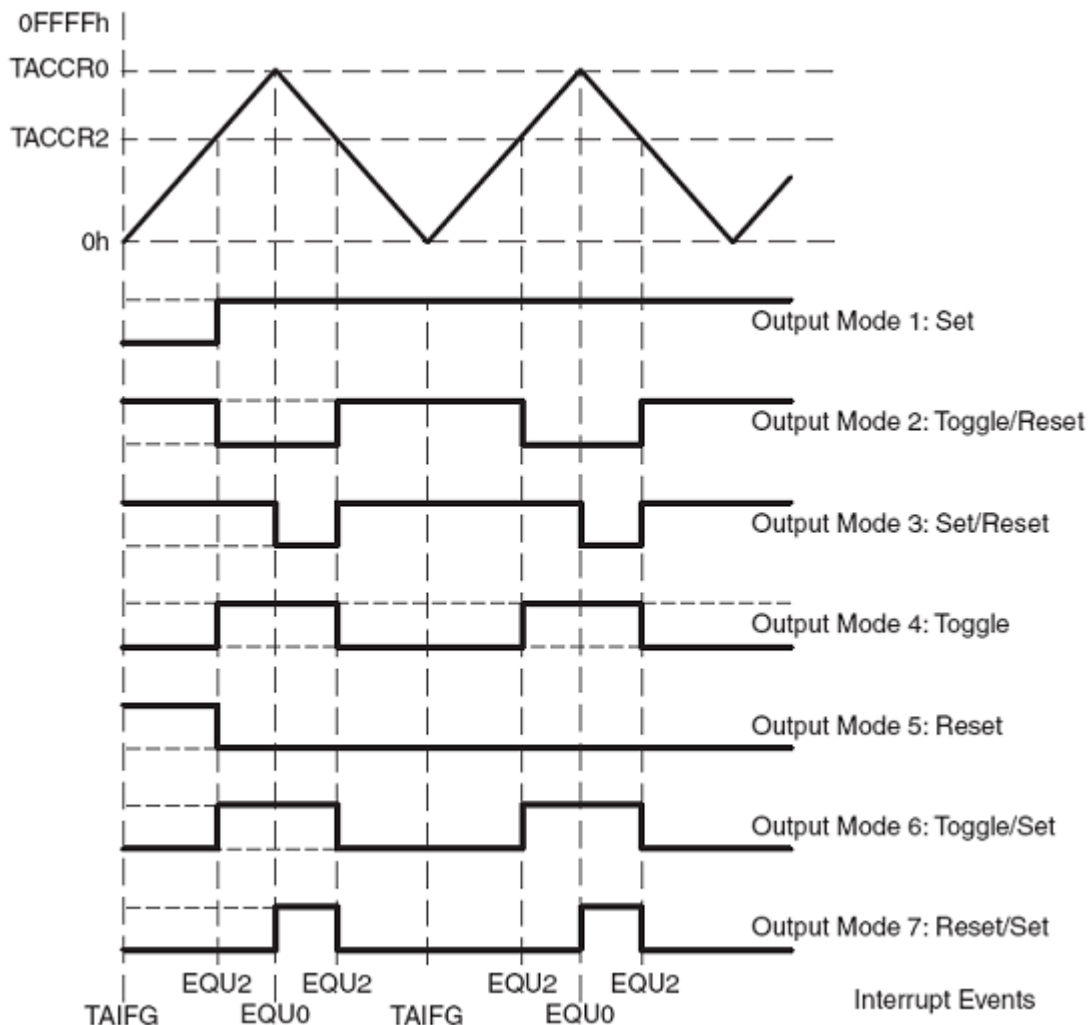


图 12-14 输出举例—定时器处于增减模式

注意：输出模式的切换

当需要在输出模式之间进行切换时，OUTMODX 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, &TACCTLx; 设置为输出模式 7

BIC#OUTMODx,&TACCTLx; 清除不需要的位

### 12.2.6 定时器 A 的中断

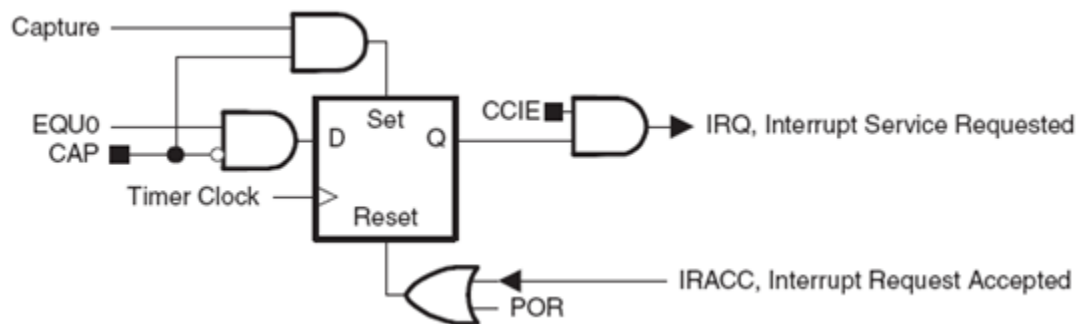
16 位定时器 A 有 2 个中断向量：

- TACCR0 CCIFG 的 TACCR0 中断向量
- 所有其他 CCIFG 和 TAIFG 的 TAIV 中断向量

在捕获模式下，当一个定时器的值捕获到相应的 TACCRx 寄存器时，CCIFG 标志置位。在比较模式下，如果 TAR 计数到相应的 TACCRx 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。

#### TACCR0 中断

TACCR0 CCIFG 标志拥有定时器 A 的最高中断优先级，并有一个专用的中断向量，如图 12-15。当进入 TACCR0 中断后，TACCR0 CCIFG 标志自动复位。



如图 12-15 捕获比较 TACCR0 中断标志

TAIV，中断向量发生器

TACCR1 CCIFG TACCR2 CCIFG 和 TAIFG 标志共用一个中断向量。中断向量寄存器 TAIV 用于确定它们中的那个要求响应中断。最高优先级的中断在 TAIV 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。定时器 A 中断不会影响 TAIV 的值。对 TAIV 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，在结束原告的中断响应后会，该中断响应立即发生。例如，当中断服务子程序访问 TAIV 时，如果 TACCR1 和 TACCR2 CCIFG 标志位置位，TACCR1 CCIFG 自动复位。在中断服务子程序的 RETI 命令执行后，TACCR2 CCIFG 标志会产生另一个中断。

TAIV 软件示例

以下软件说明了 TAIV 的使用和操作。TAIV 的值加入 PC 指针来自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：

- 捕获比较模块 TACCR0 为 11 个时钟周期
- 捕获/比较模块 TACCR1，TACCR2：16 个时钟周期
- 定时器溢出标志 TAIFG：14 个时钟周期

; TACCR0 CCIFG的中断周期.		周期数
CCIFG_0_HND		
; ... ; 中断响应开始		6
RETI		5
; TAIFG, TACCR1 和TACCR2 CCIFG的中断处理.		
TA_HND ... ; 中断响应		6
ADD &TAIV,PC ;加偏移量跳转到标号table		3
RETI ; Vector 0:无中断		5
JMP CCIFG_1_HND ; 中断2: TACCR1		2
JMP CCIFG_2_HND ; 中断4: TACCR2		2
RETI ; 中断6:保留		5
RETI ; 中断8: 保留		5
RETI ; 中断10:保留		5
RETI ; 中断12:保留		5
TAIFG_HND ; 中断14: TAIFG Flag		
... ;开始任务		
RETI		5
CCIFG_2_HND ; 中断4: TACCR2		
... ; 开始任务		
RETI ; 返回主程序		5

CCIFG\_1\_HND ; 中断2: TACCR1  
 ... ; 开始任务  
 RETI ; 返回主程序

5

### 12.3 定时器A寄存器

定时器A的Timer\_A7寄存器如表12-3，这是可以使用的最大配置。在设置使用手册中可以找到基地址。

表，定时器A寄存器

寄存器	简写	寄存器类型	寄存器入口类型	寄存器地址	初始状态
Timer_A7 控制寄存器	TACTL TACTL_L TACTL_H	读/写 读/写 读/写	Word Byte Byte	00h 00h 01h	0000h 00h 00h
Timer_A7 捕获比较控制寄存器 0	TACCTL0 TACCTL0_L TACCTL0_H	读/写 读/写 读/写	Word Byte Byte	02h 02h 03h	0000h 00h 00h
Timer_A7 捕获比较控制寄存器 1	TACCTL1 TACCTL1_L TACCTL1_H	读/写 读/写 读/写	Word Byte Byte	04h 04h 05h	0000h 00h 00h
Timer_A7 捕获比较控制寄存器 2	TACCTL2 TACCTL2_L TACCTL2_H	读/写 读/写 读/写	Word Byte Byte	06h 06h 07h	0000h 00h 00h
Timer_A7 捕获比较控制寄存器 3	TACCTL3 TACCTL3_L TACCTL3_H	读/写 读/写 读/写	Word Byte Byte	08h 08h 09h	0000h 00h 00h
Timer_A7 捕获比较控制寄存器 4	TACCTL4 TACCTL4_L TACCTL4_H	读/写 读/写 读/写	Word Byte Byte	0Ah 0Ah 0Bh	0000h 00h 00h
Timer_A7 捕获比较控制寄存器 5	TACCTL5 TACCTL5_L TACCTL5_H	读/写 读/写 读/写	Word Byte Byte	0Ch 0Ch 0Dh	0000h 00h 00h
Timer_A7 捕获比较控制寄存器 6	TACCTL6 TACCTL6_L TACCTL6_H	读/写 读/写 读/写	Word Byte Byte	0Eh 0Eh 0Fh	0000h 00h 00h
Timer_A7 计数器	TAR TAR_L TAR_H	读/写 读/写 读/写	Word Byte Byte	10h 10h 11h	0000h 00h 00h
Timer_A7 捕获比较控制 0	TACCR0 TACCR0_L TACCR0_H	读/写 读/写 读/写	Word Byte Byte	12h 12h 13h	0000h 00h 00h
Timer_A7 捕获比较控制 1	TACCR1 TACCR1_L TACCR1_H	读/写 读/写 读/写	Word Byte Byte	14h 14h 15h	0000h 00h 00h
Timer_A7 捕获比较控制 2	TACCR2 TACCR2_L TACCR2_H	读/写 读/写 读/写	Word Byte Byte	16h 16h 17h	0000h 00h 00h

Timer_A7 捕获比较控制 3	TACCR3 TACCR3_L TACCR3_H	读/写 读/写 读/写	Word Byte Byte	18h 18h 19h	0000h 00h 00h
Timer_A7 捕获比较控制 4	TACCR4 TACCR4_L TACCR4_H	读/写 读/写 读/写	Word Byte Byte	1Ah 1Ah 1Bh	0000h 00h 00h
Timer_A7 捕获比较控制 5	TACCR5 TACCR5_L TACCR5_H	读/写 读/写 读/写	Word Byte Byte	1Ch 1Ch 1Dh	0000h 00h 00h
Timer_A7 捕获比较控制 6	TACCR6 TACCR6_L TACCR6_H	读/写 读/写 读/写	Word Byte Byte	1Eh 1Eh 1Fh	0000h 00h 00h
Timer_A7 中断向量	TAIV TAIV_L TAIV_H	读/写 读/写 读/写	Word Byte Byte	2Eh 2Eh 2Fh	0000h 00h 00h
Timer_A7 扩展	TAEX0 TAEX0 TAEX0	读/写 读/写 读/写	Word Byte Byte	20h 20h 21h	0000h 00h 00h

**TACTL, Timer\_A 控制寄存器**

15	14	13	12	11	10	9	8
Unused						TASSELx	
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)
7	6	5	4	3	2	1	0
IDx	MCx	Unused	TACLR	TAIE	TAIFG		
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)

未使用	15-10	未使用位
		TA 时钟源选择
		00 TACLK
TASSELx	9-8	01 ACLK
		10 SMCLK
		11 INCLK
		输入分频。这些位为输入时钟分频选择
		00 /1
IDx	7-6	01 /2
		10 /4
		11 /8
		模式控制，当定时器 A 不用于节省功耗时，将 MCx=00h
		00 停止模式：定时器停止
MCx	5-4	01 增模式 定时器计数到 TACCR0
		10 连续模式，定时器计数到 0FFFH
		11 增减模式：定时器计数到 TACCR0 然后送到 0000H
Unused	3	未使用位
TACLR	2	定时器清零位。该位置位会复位 TAR，时钟分频和计数方向。 TACLR 位会自动复位并值为 0
TAIE	1	TA 中断允许。改位允许 TAIFG 中断请求 0 中断禁止

TAIFG	0	1 中断允许
		TA 中断标志位
		0 无中断挂起
		1 中断挂起

## TAR, Timer A 寄存器

[illegible]

TARx 位 15-0

Timer A 寄存器, TAR 寄存器 Timer A 的计数器

## TACCTLx, 捕获比较控制寄存器

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

		捕获模式
		00 不捕获
CMx	15-14	01 上升沿捕获
		10 下降沿捕获
		11 上升和下降沿都捕获
		捕获比较选择, 该位选择 TACCRx 的输入信号, 详见器件手册
		00 CCIxA
CCISx	13-12	01 CCIxB
		10 GND
		11 VCC
		同步捕获源, 该位用于将捕获通信和时钟同步
SCS	11	0 异步捕获
		1 同步捕获
SCCI	10	同步的捕获/比较输入, 所选择的 CCI 输入信号由 EQUx 信号锁
		存, 并可通过该位读取
Unused	9	未使用
		捕获模式
CAP	8	0 比较模式
		1 捕获模式
		输出模式位。由于在模式 2, 3, 6 和 7 下 EQUx
		=EQU0,因此这些模式对 TACCR0 无效
OUTMODx	7-5	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
		捕获比较中断允许位，该位允许相应的 <b>CCIFG</b> 标志中断请求
CCIE	4	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 对于输出模式 <b>0</b> ，该位直接控制输出状态
OUT	2	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。 <b>COV</b> 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

**TAIV, Timer\_A 中断向量寄存器**

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
r0	r0	r0	r0	r0	r0	r0	r0
7	6	5	4	3	2	1	0
0	0	0	0	TAIVx			0
r0	r0	r0	r0	r0	r0	r0	r0

**TAIVx Bits 15-0 Timer\_A 中断向量值**

TAIV 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	TACCR1 CCIFG	最高
04H	捕获比较 2	TACCR2 CCIFG	
06H	捕获比较 3	TACCR3 CCIFG	
08H	捕获比较 4	TACCR4 CCIFG	
0AH	捕获比较 5	TACCR5 CCIFG	
0CH	捕获比较 6	TACCR6 CCIFG	
0EH	定时器溢出	TAIFG	最低位

**TAEX0, Timer\_A 扩展寄存器 0**

15	14	13	12	11	10	9	8
Unused	Unused	Unused	Unused	Unused	Unused	Unused	Unused
r0	r0	r0	r0	r0	r0	r0	r0
7	6	5	4	3	2	1	0
Unused	Unused	Unused	Unused	Unused	IDEX		
r0	r0	r0	r0	r0		rw_(0)	rw_(0)

未使用 位 15—3 未使用，只读，总为 0。

IDEX 位 2—0，输入分频器。由 IDX 的这些位来选择输入时钟

000 /1

001 /2

010 /3

011 /4

100 /5

101 /6

100 /7

111 /8