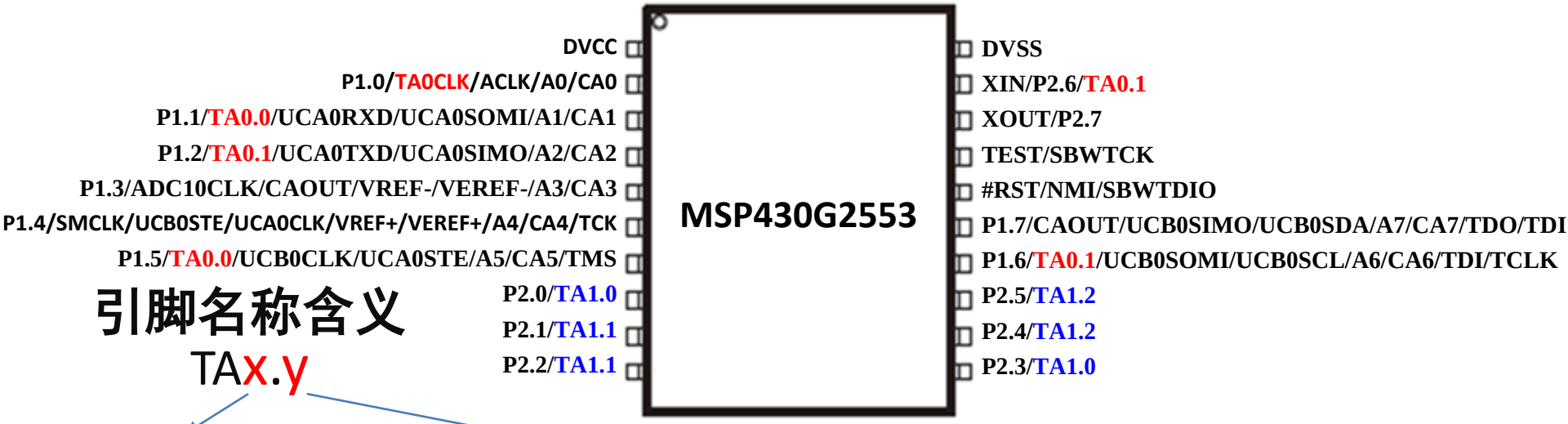
定时器TA的PWM输出（简介版）

定时器TA的详细版本参看 “参考资料 定时器”

20引脚双列直插式

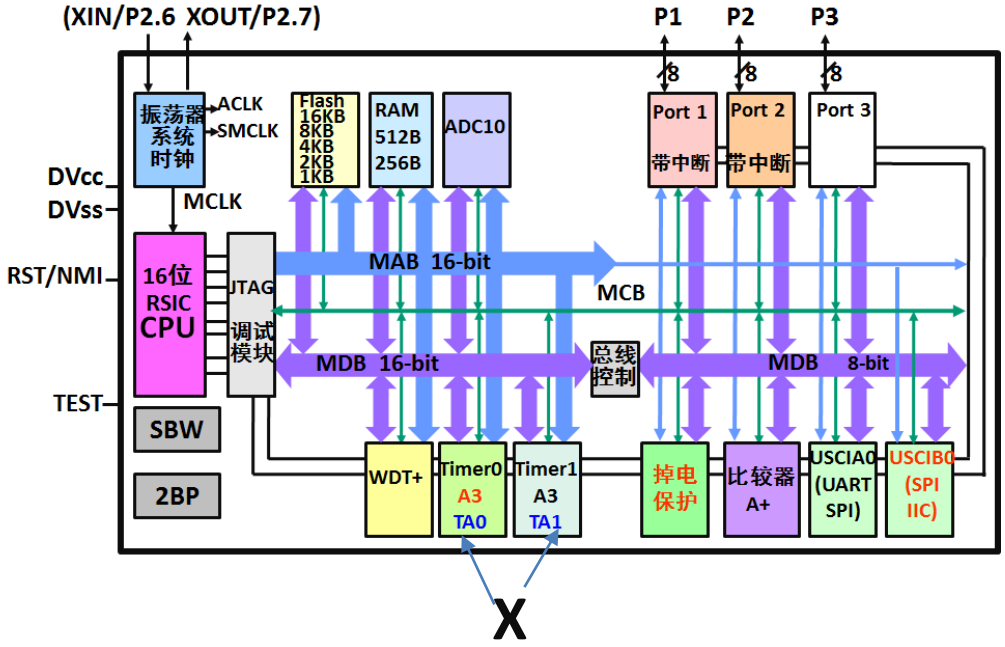


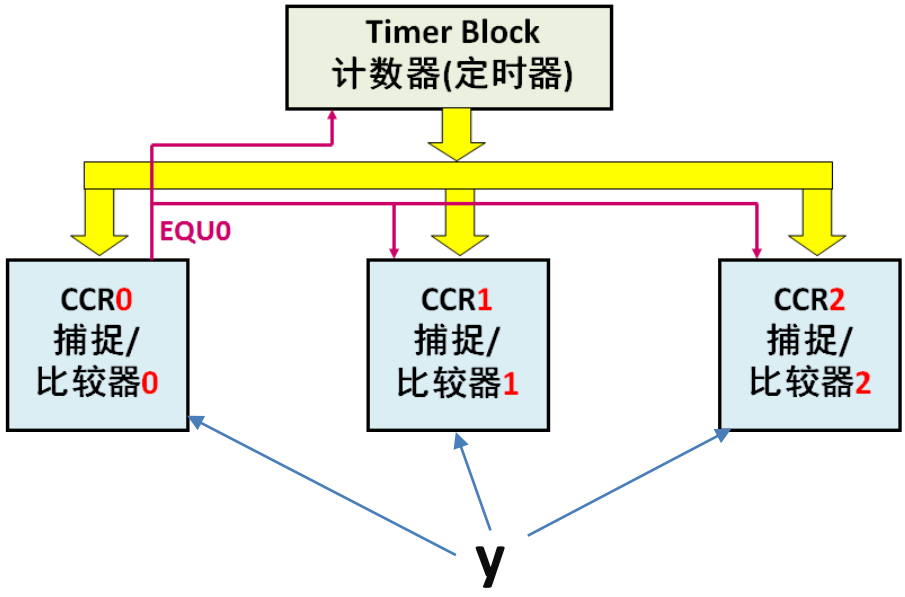
引脚名称含义

TAx.y

x 表示哪个定时器

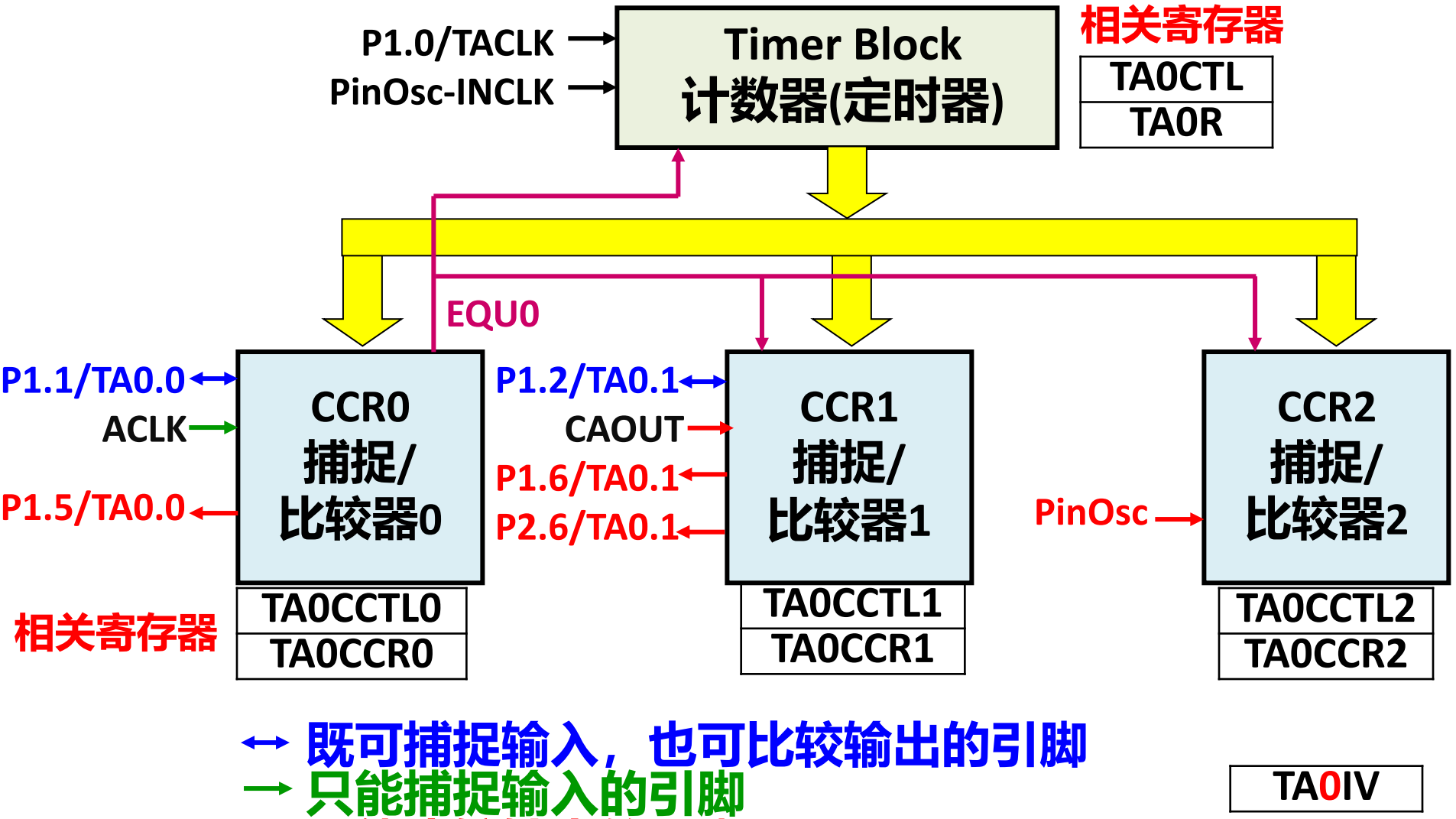
y 表示 定时器中的哪个捕捉/比较器





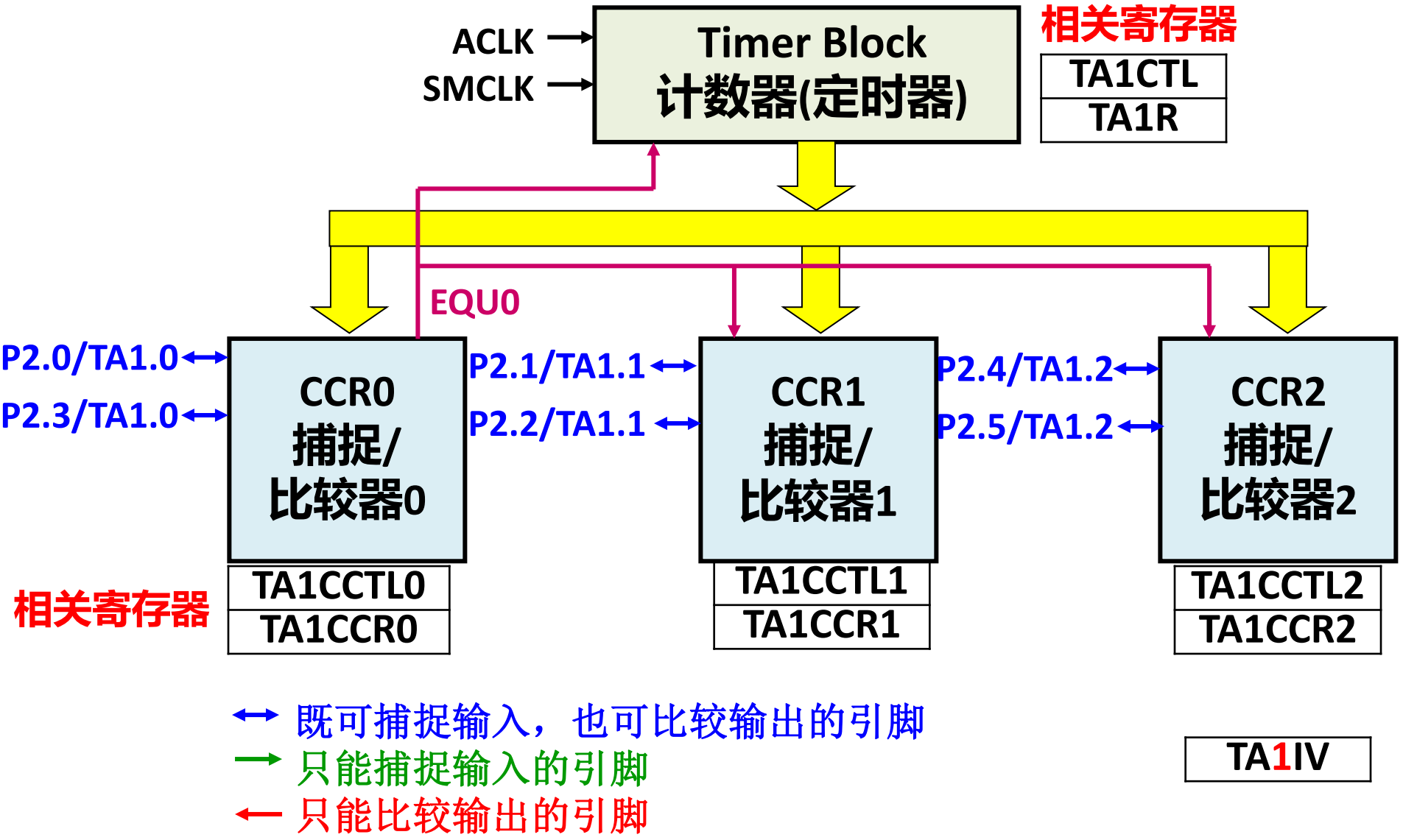
TA0的结构图

由1个计数定时器单元和3个捕捉/比较器单元构成

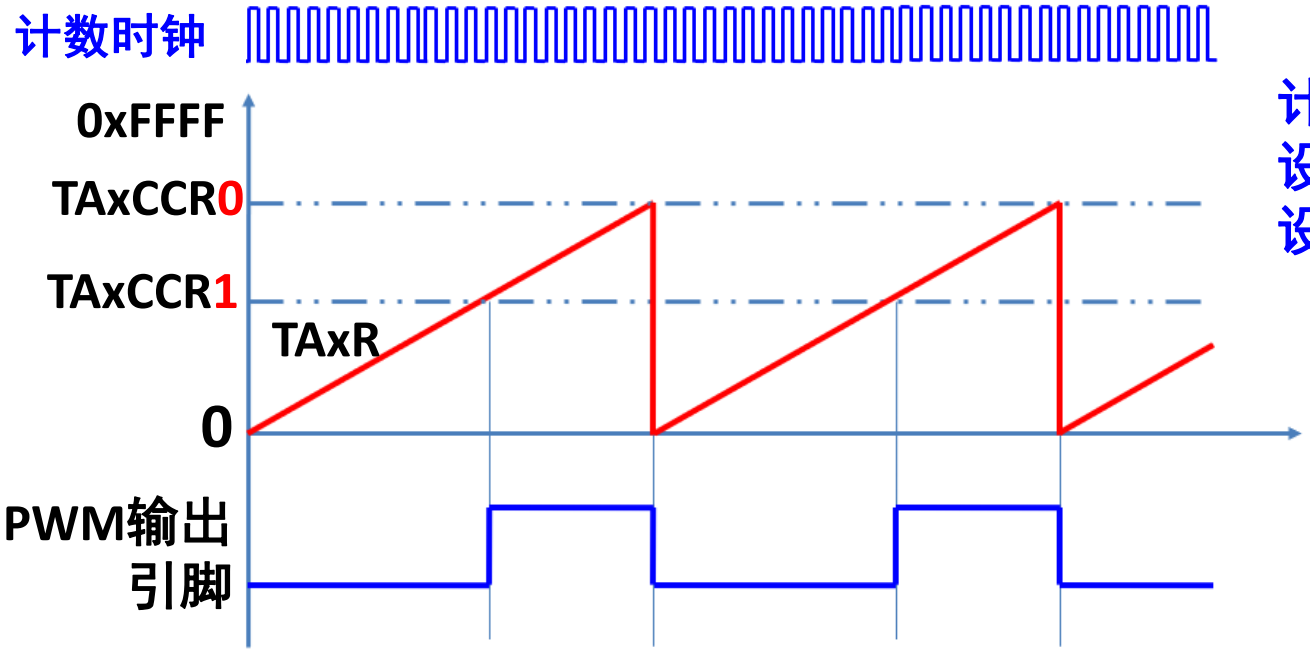


TA1的结构图

由1个计数定时器单元和3个捕捉/比较器单元构成



Pulse Width Modulation脉宽调制输出(比较输出)

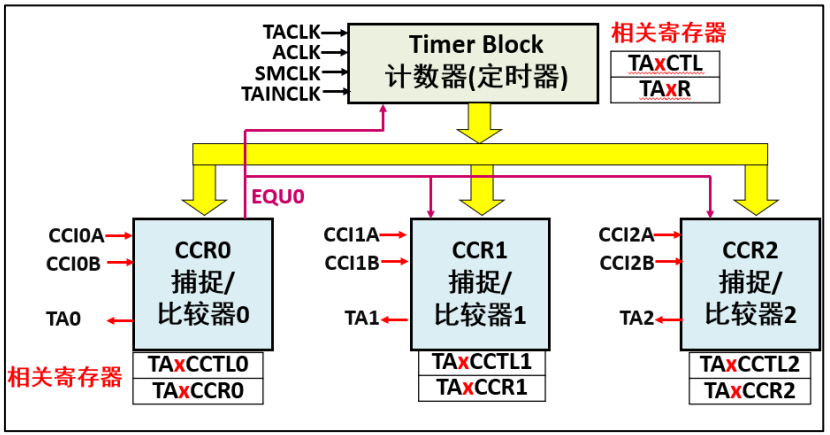


计数寄存器TAxR
设定值寄存器TAxCCR0
设定值寄存器TAxCCR1

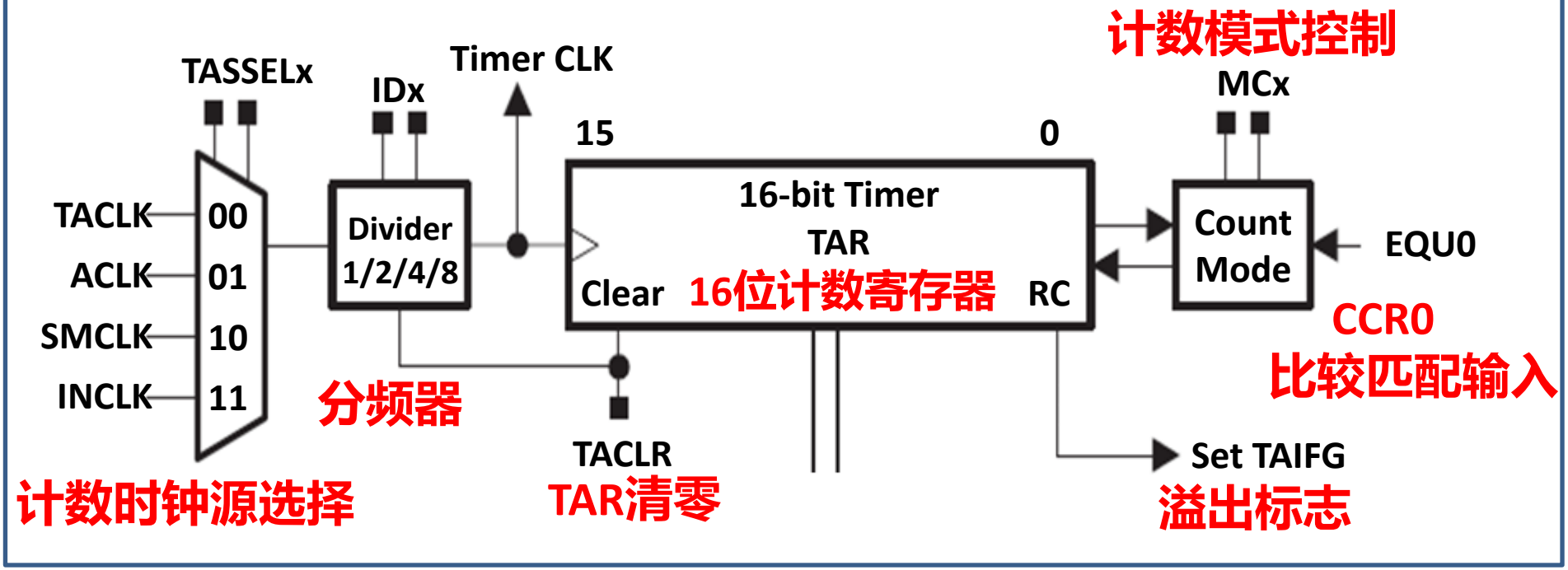
- 计数CLK时钟的上升沿触发1次计数器TAR计数，在增模式下加TAxR自动加1;
- 当TAxR=TAxCCR1, 置引脚输出1；
- 当TAxR=TAxCCR0, 置引脚输出0；并使TAxR回0，从0开始计数

- 改变TAxCCR0，改变脉冲周期和频率;
- 改变TAxCCRy，改变占空比

图中：
PWM的周期=(TAxCCR0+1)*1/计数时钟频率
PWM的频率=计数时钟频率/(TAxCCR0+1)
PWM占空比=(TAxCCR0-TAxCCR1)/(TAxCCR0+1)



TA 计数定时器部分结构图



图中控制位(如TASSELx、 MCx等)中的 x 表示有2位以上, 按顺序可为 0、 1、 2、 ….

相关寄存器:

TAxCTL: TA控制寄存器

TAxR: TA计数寄存器

寄存器名称中的 **x**可为0或1，分别对应定时器TA0，TA1

TA计数寄存器 TAxR

- TAR是1个16位计数寄存器，其内容可读可写;
- Timer CLK时钟 的上升沿触发1次计数器TAR计数，
- 由计数方式决定TAR自动随时钟个数加 1 或减 1、
  到何值时设置溢出中断标志；
- TAR 可由程序设置初值，可由控制寄存器的TACLR 位清零

15	14	13	…	…	…	1	0
TAxR							
rw-(0)	rw-(0)	rw-(0)	…	…	rw-(0)	rw-(0)	rw-(0)

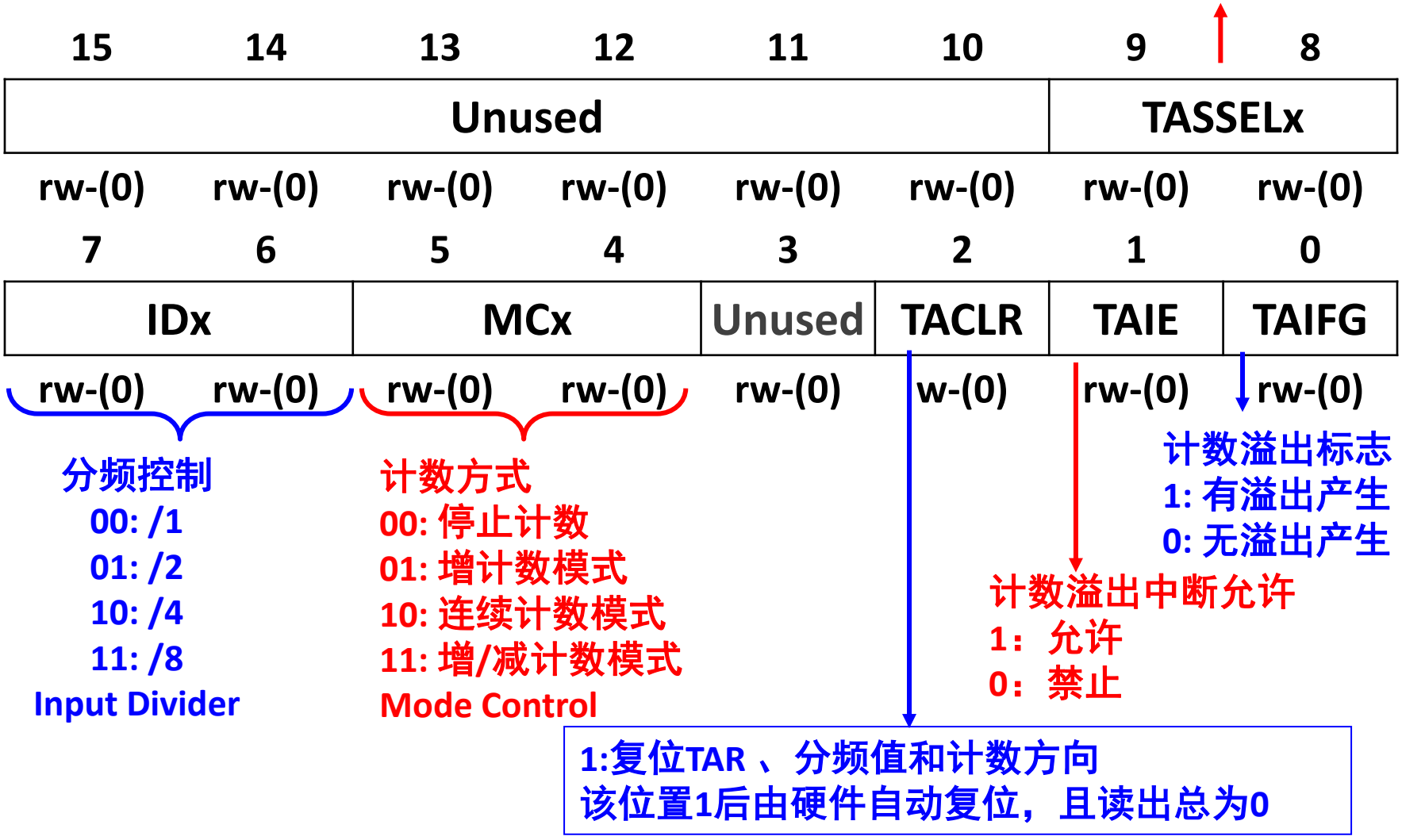
注意： TAxR 中x分别对应定时器TA0，TA1，x可为0或1

TA控制寄存器 TAxCTL

Timer_A control Register

控制计数定时器的操作

Source Select
时钟源选择
00: TACLK
01: ACLK
10: SMCLK
11: INCLK



msp430g2553h中TAxCTL各个位及位域值的符号定义

```
//位定义
#define  TASSEL1          (0x0200)     /* Timer A clock source select 1 */
#define  TASSEL0          (0x0100)     /* Timer A clock source select 0 */
#define  ID1              (0x0080)     /* Timer A clock input divider 1 */
#define  ID0              (0x0040)     /* Timer A clock input divider 0 */
#define  MC1              (0x0020)     /* Timer A mode control 1 */
#define  MC0              (0x0010)     /* Timer A mode control 0 */
#define  TACLR            (0x0004)     /* Timer A counter clear */
#define  TAIE             (0x0002)     /* Timer A counter interrupt enable */
#define  TAIFG            (0x0001)     /* Timer A counter interrupt flag */
```

```
//位域值的定义
#define  MC_0         (0*0x10u)  /* Timer A mode control: 0 - Stop */
#define  MC_1         (1*0x10u)  /* Timer A mode control: 1 - Up to CCR0 */
#define  MC_2         (2*0x10u)  /* Timer A mode control: 2 - Continous up */
#define  MC_3         (3*0x10u)  /* Timer A mode control: 3 - Up/Down */
#define  ID_0         (0*0x40u)  /* Timer A input divider: 0 - /1 */
#define  ID_1         (1*0x40u)  /* Timer A input divider: 1 - /2 */
#define  ID_2         (2*0x40u)  /* Timer A input divider: 2 - /4 */
#define  ID_3         (3*0x40u)  /* Timer A input divider: 3 - /8 */
#define  TASSEL_0     (0*0x100u) /* Timer A clock source select: 0 - TACLK */
#define  TASSEL_1     (1*0x100u) /* Timer A clock source select: 1 - ACLK  */
#define  TASSEL_2     (2*0x100u) /* Timer A clock source select: 2 - SMCLK */
#define  TASSEL_3     (3*0x100u) /* Timer A clock source select: 3 - INCLK
```

TAxCCTLy捕捉/比较器控制寄存器 (x=0,1；y=0,1,2)

capture/compare control register

决定CCRx的工作方式
0: 比较方式
1: 捕捉方式

了解比较匹配、比较输出有关的设置即可，
其他为用上电复位设置

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	r0	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)

000~111
比较输出方式
8种

比较匹配
中断允许
1: 允许
0: 禁止

当OUTMODx=0时
决定比较输出引脚
上输出的电平值

比较匹配
中断标志
1: 有中断
0: 无中断

msp430g2553.h中TAxCCTLy相关各个位及位域值的符号定义

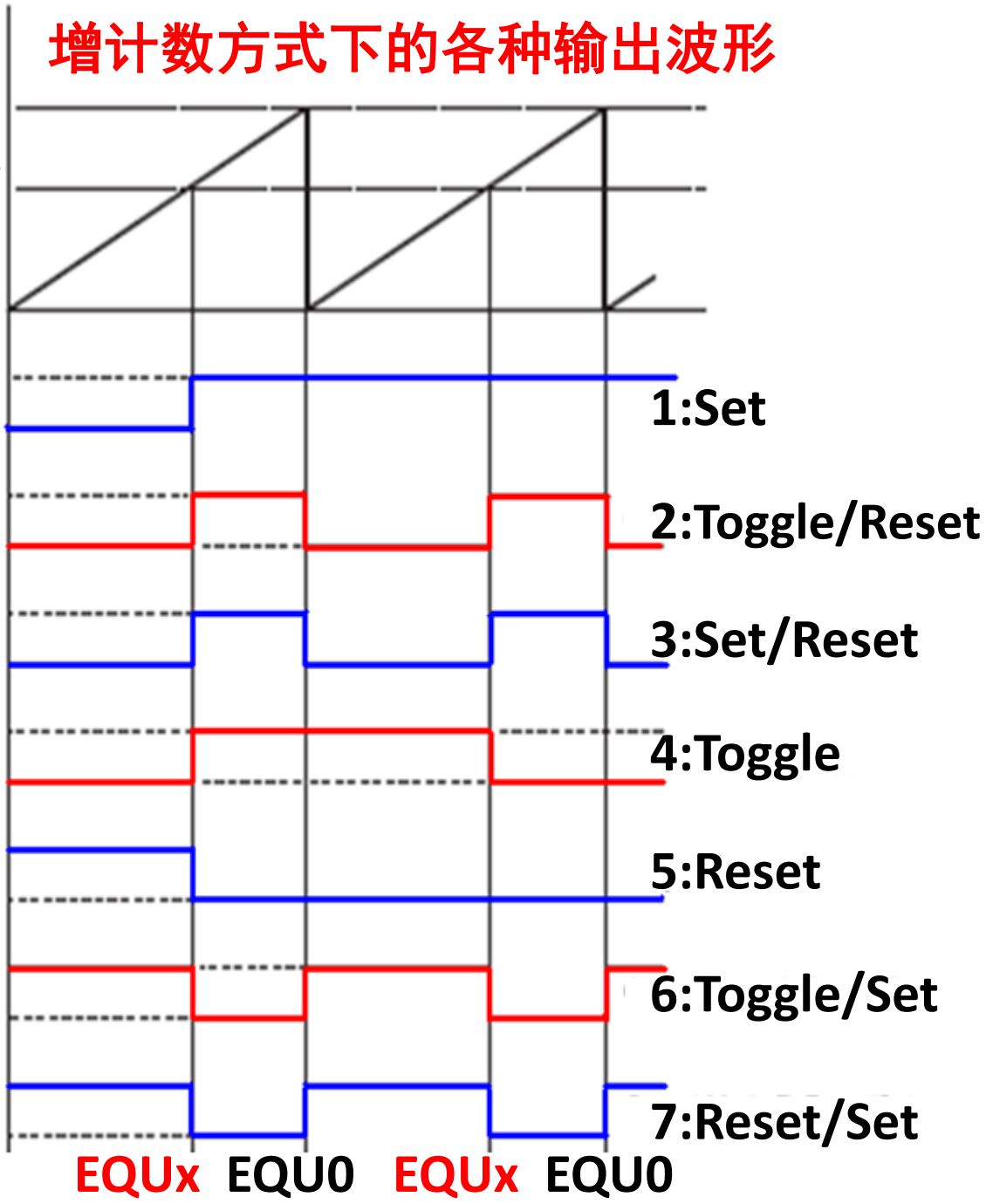
```
#define  CAP          (0x0100)    /* Capture mode: 1 /Compare mode : 0 */
#define  OUTMOD2      (0x0080)    /* Output mode 2 */
#define  OUTMOD1      (0x0040)    /* Output mode 1 */
#define  OUTMOD0      (0x0020)    /* Output mode 0 */
#define  CCIE         (0x0010)    /* Capture/compare interrupt enable */
#define  OUT          (0x0004)    /* PWM Output signal if output mode 0 */
#define  CCIFG        (0x0001)    /* Capture/compare interrupt flag */
#define  OUTMOD_0     (0*0x20u)   /* PWM output mode: 0 - output only */
#define  OUTMOD_1     (1*0x20u)   /* PWM output mode: 1 - set */
#define  OUTMOD_2     (2*0x20u)   /* PWM output mode: 2 - PWM toggle/reset */
#define  OUTMOD_3     (3*0x20u)   /* PWM output mode: 3 - PWM set/reset */
#define  OUTMOD_4     (4*0x20u)   /* PWM output mode: 4 - toggle */
#define  OUTMOD_5     (5*0x20u)   /* PWM output mode: 5 - Reset */
#define  OUTMOD_6     (6*0x20u)   /* PWM output mode: 6 - PWM toggle/set */
#define  OUTMOD_7     (7*0x20u)   /* PWM output mode: 7 - PWM reset/set */
```

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)

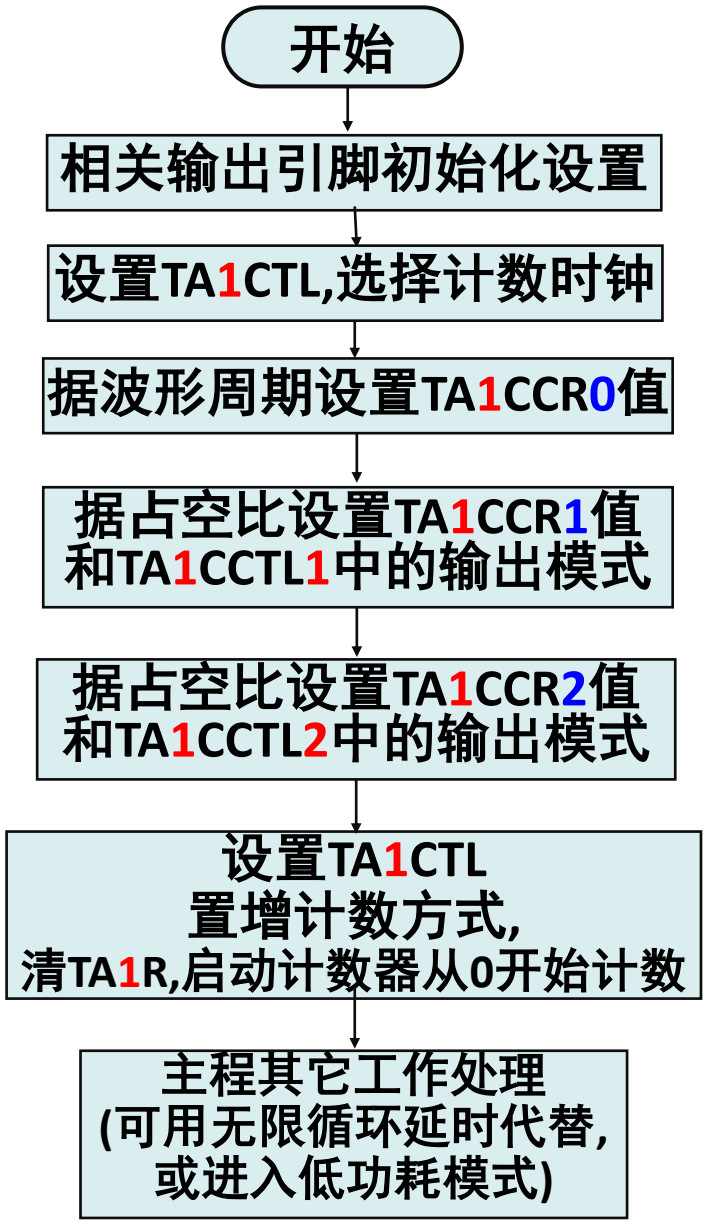
输出模式

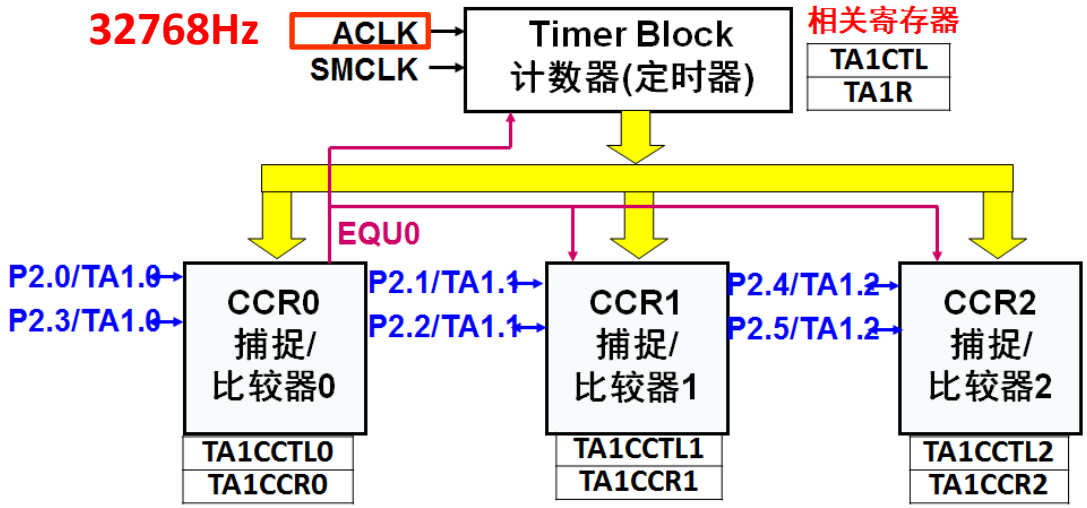
OUT MODx	EQUy时输出	EQU0时输出
000	=OUT位	不影响
001	置位	不影响
010	翻转	复位
011	置位	复位
100	翻转	不影响
101	复位	不影响
110	翻转	置位
111	复位	置位

利用模式0，
可通过OUT位设置
波形的初始值

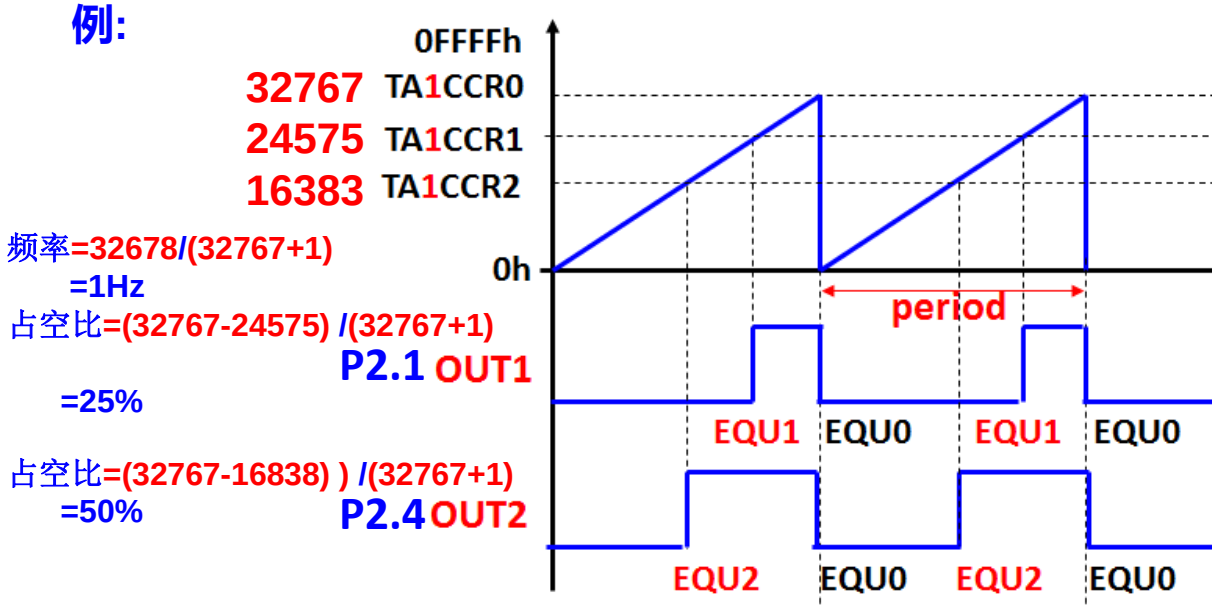


TA_pwm.c程序流程



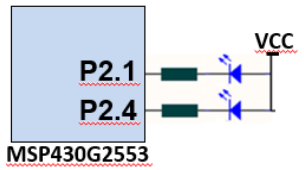


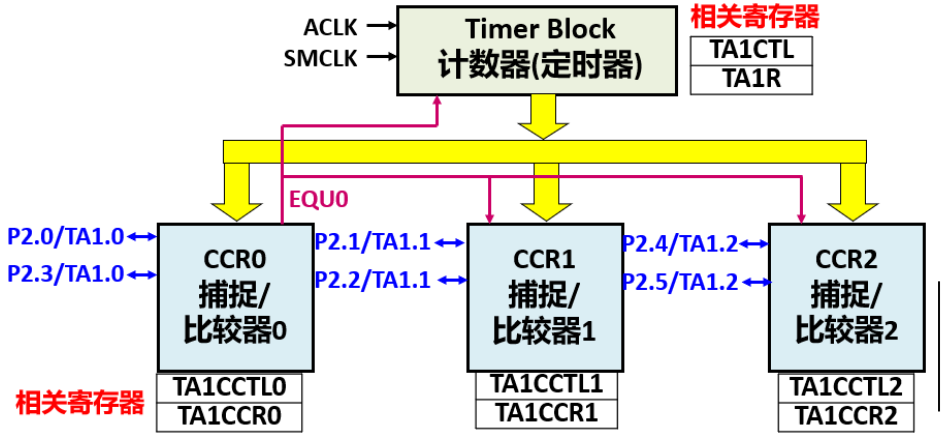
例:



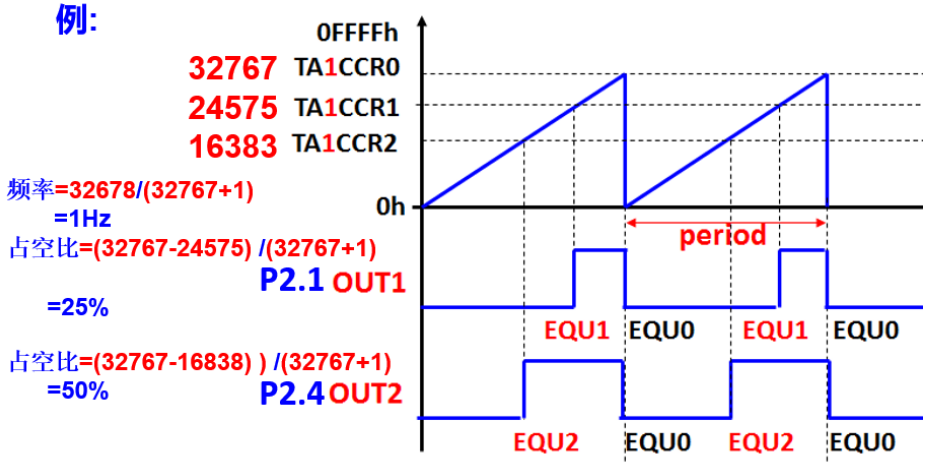
TA_PWM.c

```
#include "msp430.h"
int main ( void )
{   WDTCTL = WDTPW + WDTHOLD; //关闭看门狗
    P2SEL |=BIT1+BIT4;                  //置P2.1和 P2.4为定时器TA1的PWM输出引脚
    P2SEL2 &=~(BIT1+BIT4);              //P2.1为比较器1的PWM输出引脚
    P2DIR |=BIT1+BIT4;                  //P2.4为比较器2的PWM输出引脚

    TA1CTL |=TASSEL0;                   //选择ACLK为TA1计数时钟，ACLK使用上电复位设置，即外部晶振32768Hz
    TA1CCR0=32767;                      //置PWM周期，周期= (TA1CCR0+1)*T =(TA1CCR0+1)/计数时钟频率
                                        //输出的PWM波形频率=1/PWM周期=计数时钟频率/(TA1CCR0+1)
    TA1CCTL1|=OUTMOD1;                  //置TA1比较器1的PWM输出为模式2： 计数到CCR1值翻转，到CCR0值置0
    TA1CCR1=24575;                      //置TA1比较器1设定值CCR1，TA1CCR1=TA1CCR0*(1-PWM波形占空比25%)

   TA1CCTL2|=OUTMOD1;                   //置比较器2的PWM输出为模式2： 计数到CCR2值翻转，到CCR0值置0
   TA1CCR2=16383;                       //置TA1比较器2设定值CCR2，TA1CCR2=TA1CCR0*(1-PWM波形占空比50%)

    TA1CTL |=TACLR+MC0;                 //置增计数方式，使计数器从0开始计数，计数到TA1CCR0后又从0计数。

    while(1){ };                        // 主循环，CPU可做其他事情
}
```



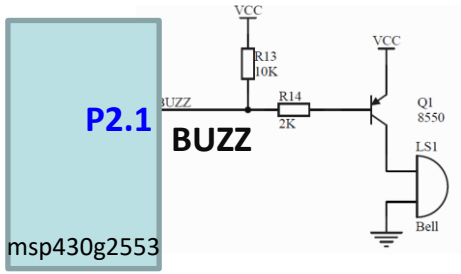


例:



TA_PWM.c

```
#include "msp430.h"
int main ( void )
{   WDTCTL = WDTPW + WDTHOLD; //关闭看门狗
    P2SEL |=BIT1+BIT4;                  //置P2.1和 P2.4为定时器TA1的PWM输出引脚
    P2SEL2 &=~(BIT1+BIT4);              //P2.1为比较器1的PWM输出引脚
    P2DIR |=BIT1+BIT4;                  //P2.4为比较器2的PWM输出引脚

    TA1CTL |=TASSEL0;                   //选择ACLK为TA1计数时钟，ACLK使用上电复位设置，即外部晶振32768Hz
    TA1CCR0=32767;                      //置PWM周期，周期= (TA1CCR0+1)*T =(TA1CCR0+1)/计数时钟频率
                                        //输出的PWM波形频率=1/PWM周期=计数时钟频率/(TA1CCR0+1)
    TA1CCTL1|=OUTMOD1;                  //置TA1比较器1的PWM输出为模式2： 计数到CCR1值翻转，到CCR0值置0
    TA1CCR1=24575;                      //置TA1比较器1设定值CCR1，TA1CCR1=TA1CCR0*(1-PWM波形占空比25%)

   // TA1CCTL2|=OUTMOD1;                //置比较器2的PWM输出为模式2： 计数到CCR2值翻转，到CCR0值置0
   //TA1CCR2=16383;                     //置TA1比较器2设定值CCR2，TA1CCR2=TA1CCR0*(1-PWM波形占空比50%)

    TA1CTL |=TACLR+MC0;                 //置增计数方式，使计数器从0开始计数，计数到TA1CCR0后又从0计数。

    while(1){  };                       // 主循环，CPU可做其他事情
}
```



图中：

PWM的周期 = (TAxCCR0+1)*计数时钟周期
= (TAxCCR0+1)*1/计数时钟频率

PWM的频率 = 计数时钟频率/(TAxCCR0+1)

PWM占空比 =(TAxCCR0-TAxCCR1)/(TAxCCR0+1)

ACLK时钟 32768Hz

C调do 音 262Hz

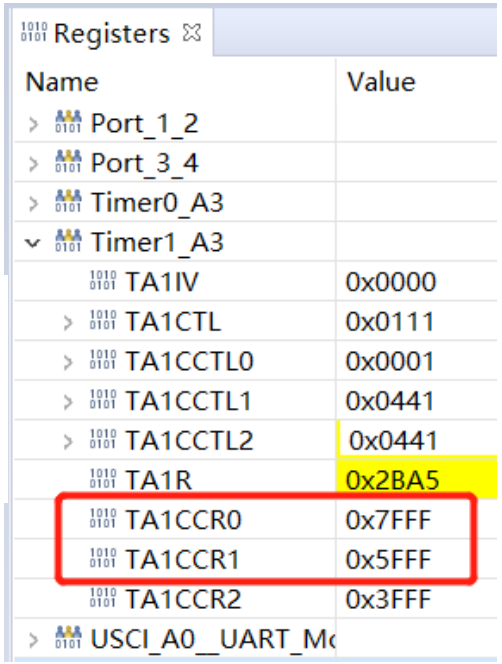
TAxCCR0=计数时钟频率/PWM频率-1

TAxCCR1=（1−占空比）*TAxCCR0

解压下载的“定时器PWM波形输出练习.zip”

综合实验准备-定时器PWM波形输出练习

在提供的参考程序TA_PWM.c基础上做练习：



■ **练习1：**

将P2.1和P2.4两个引脚分别连到两个发光二级管上；
运行TA_PWM.c程序，观察现象；
理解程序实现原理；

ACLK时钟 32768Hz　　C调do 音 262Hz

TAxCCR0=计数时钟频率/PWM频率-1

TAxCCR1=（1−占空比）*TAxCCR0

■ **练习2：**

将P2.1连到Buzz蜂鸣器控制端上；
在DEBUG下设置Timer_TA1的TA1CCR0和TA1CCR1的值，
使P2.1引脚输出频率262Hz，占空比为90%的方波；
通过听蜂鸣器是否发出C调中的do音，确定是否成功；
有示波器的同学可以用示波器测量另一处的P2.1输出的方波信号。

■ **练习3:**

改写TA_PWM.c程序，完成实验任务2功能，使蜂鸣器发出C调中do音；

■ **练习4：**

编程将右图C调各频率对应的TA1CCR0和TA1CCR1值存放在一个数组中，用循环方式，分别取出数组中的各个元素控制TA1CCR0和TA1CCR1，使蜂鸣器分别发出C调的8个音。

可参看定时器TA的PWM输出PPT（简介版）
或定时器TA的详细版本 “参考资料 定时器”