第9章：增强型脉宽调制器ePWM

PWM的学习思路

1. 什么是PWM？什么是互补PWM？
2. ePWM的构成和使用要点？
3. ePWM的中断使用方法？
4. 重要的寄存器？
5. 目标是：配置需要的PWM，完成需要的PWM功能

- 增强型脉宽调制器（ePWM）外设是控制许多商业和工业设备的电力电子系统的一个重要组件。
- 这些系统包括数字电机控制系统、开关电源控制系统、不间断电源（UPS）系统和其他形式的电源转换系统。
- ePWM外设执行数模转换（DAC）功能（占空比等效于DAC模拟值）；有时称为电源DAC（Power DAC）。

- 一个有效的PWM外设必须能够保证在CPU开销最少的前提下产生复杂的脉宽波形。
- 它必须高度可编程与高度灵活，同时还要容易理解与使用。
- ePWM单元通过为每个PWM通道分配所有必需的时序和控制资源，完全满足这些需求。
- ePWM由带单独资源的小型单通道模块构成，并且这些资源在构成一个系统时可以一起工作，从而避免了资源交叉耦合或共享的问题；这种组合方法使得ePWM变为正交结构。

9.1 ePWM总体结构

主要特性：

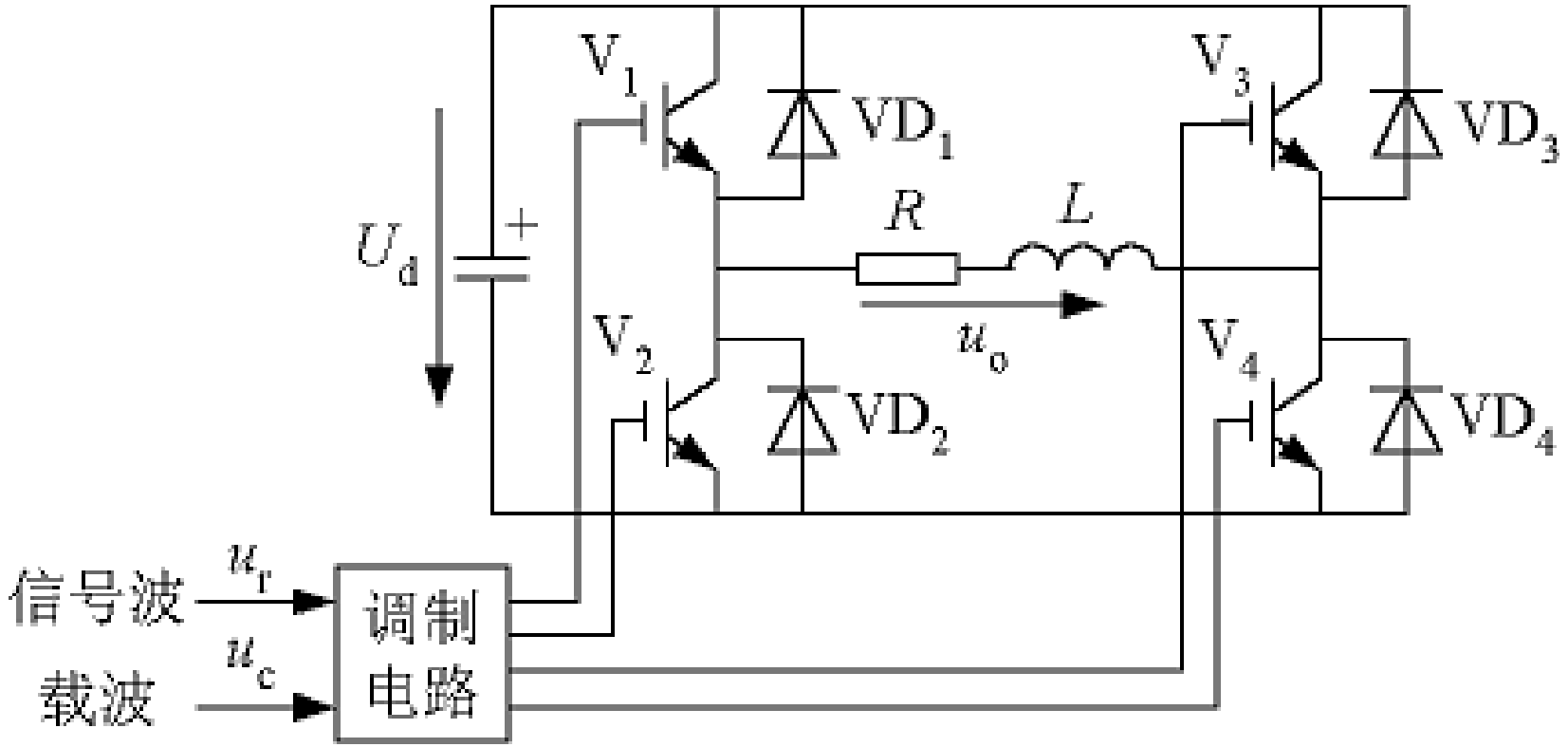
- 专用的16位时基计数器，可控制PWM输出的频率或周期

- 两个PWM输出（EPWMxA和EPWMxB），它们可以配置成：
  ---两个独立的、单边沿操作的PWM输出
  ----两个独立的、双边沿对称操作的PWM输出
  ----一个独立的、双边沿非对称操作的PWM输出

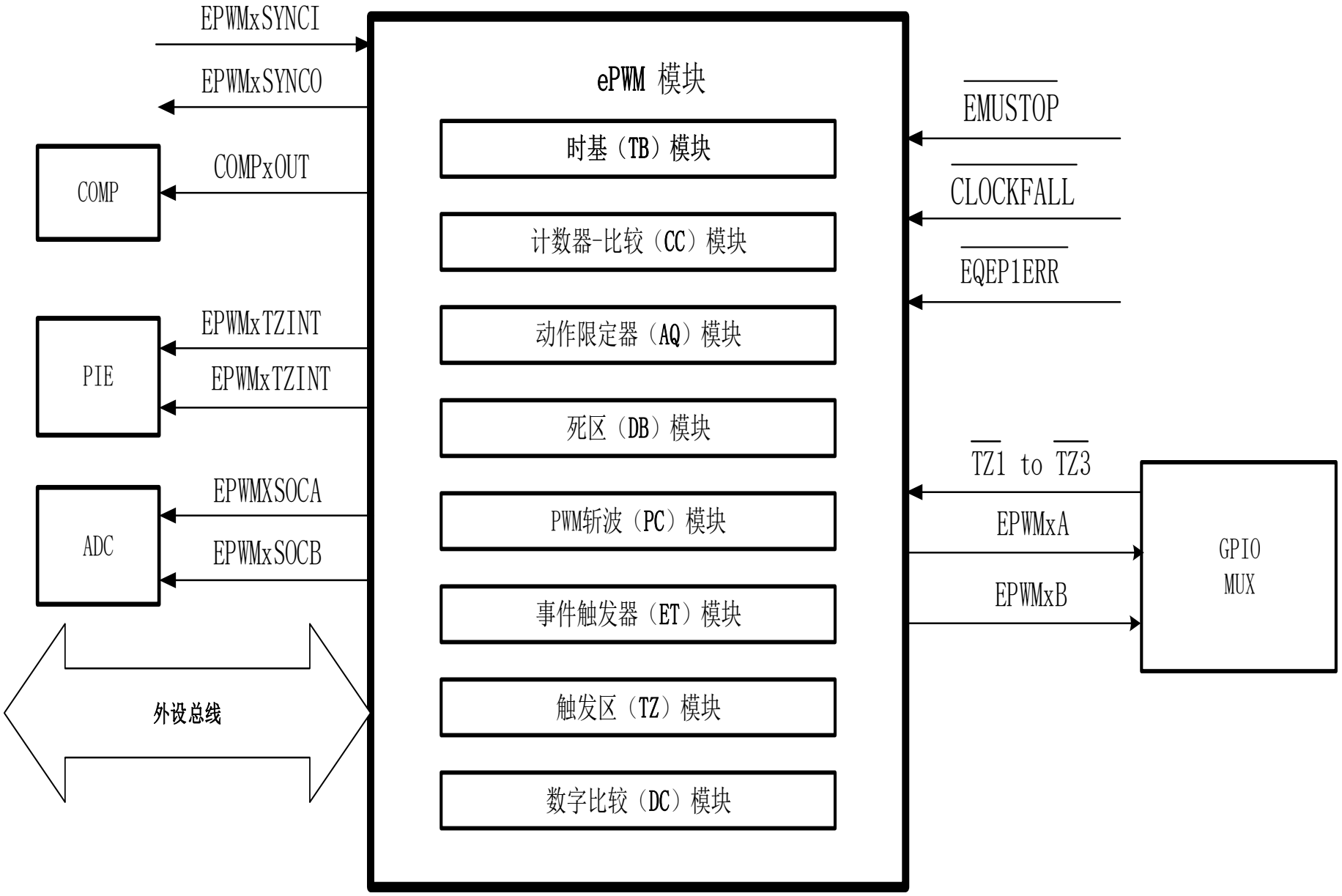
主要特性：

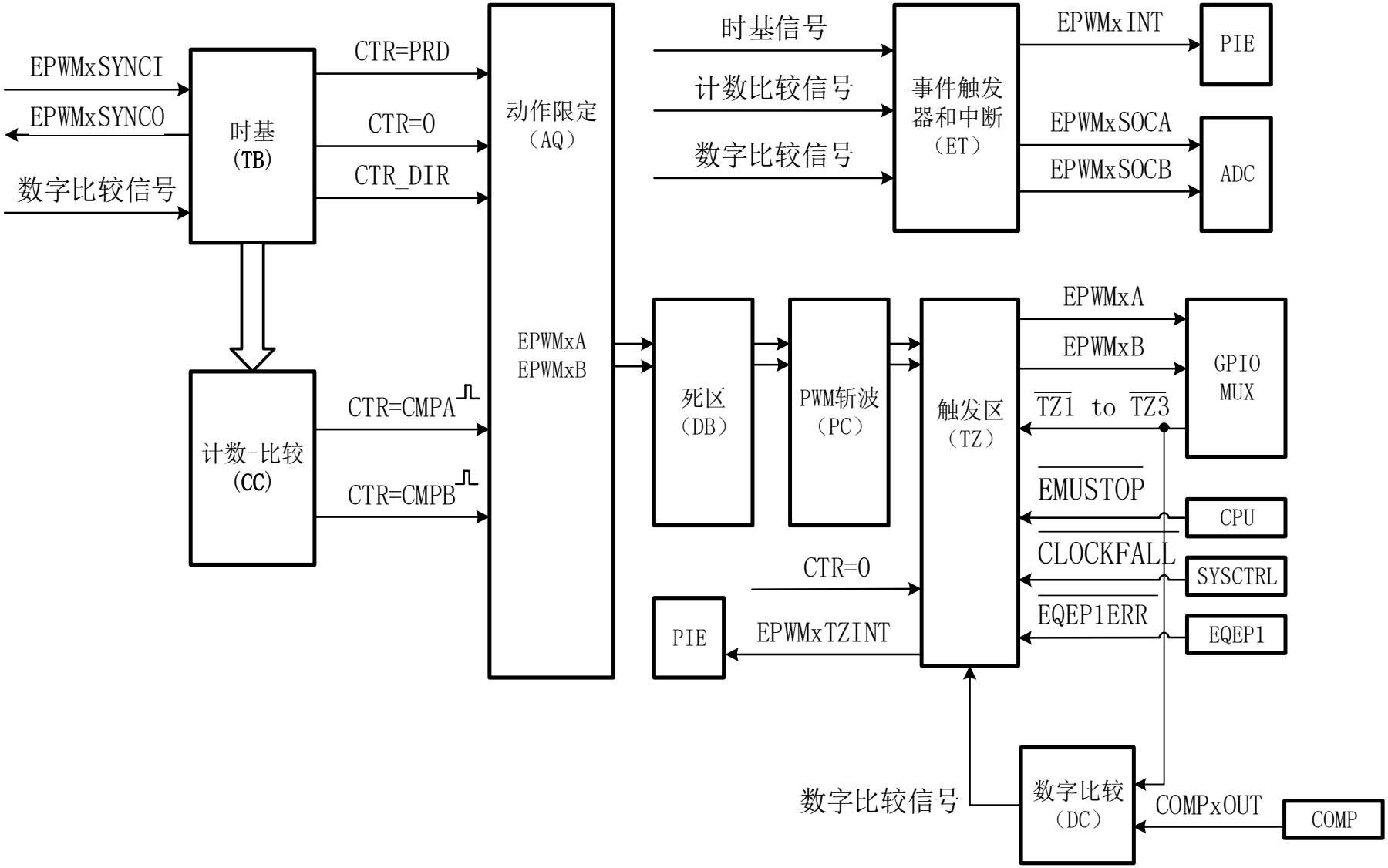
- 通过软件异步控制PWM信号
- 可编程的相位控制，配置相对于其它ePWM模块的相位（领先或滞后多少）
- 周期性地硬件锁定（同步）相位关系
- **死区发生器，带独立的上升沿延迟和下降沿延迟控制**
- 触发区配置可编程，可配置成在出现故障时周期性触发或单次触发
- 产生故障触发条件后（trip condition）可将PWM输出强制变为高电平、低电平或高阻状态

为什么需要互补PWM？



- 比较器模块输出和触发区输入可以产生事件、滤波（filtered）事件或故障触发条件
- 所有事件都可以触发CPU中断和ADC开始转换（SOC）
- 事件预分频因子（prescaling）可编程，使得中断的CPU开销最少
- ~~PWM被高频载波信号斩波，这对于脉冲变压器门极驱动非常有用~~


EPWMxSYNCI
EPWMxSYNCO
ePWM 模块
EMUSTOP
COMP
COMPxOUT
时基（TB）模块
CLOCKFALL
计数器-比较（CC）模块
EQEP1ERR
EPWMxTZINT
PIE
EPWMxTZINT
动作限定器（AQ）模块
死区（DB）模块
TZ1 to TZ3
EPWMXSOCA
ADC
EPWMxSOCB
PWM斩波（PC）模块
EPWMxA
GPIO
MUX
事件触发器（ET）模块
EPWMxB
外设总线
触发区（TZ）模块
数字比较（DC）模块


EPWMxSYNCI
EPWMxSYNCO
数字比较信号
时基
(TB)
CTR=PRD
CTR=0
CTR_DIR
计数-比较
(CC)
CTR=CMPA
CTR=CMPB
动作限定
（AQ）
EPWMxA
EPWMxB
时基信号
计数比较信号
数字比较信号
事件触发
器和中断
（ET）
EPWMxINT
PIE
EPWMxSOCA
EPWMxSOCB
ADC
死区
（DB）
PWM斩波
（PC）
触发区
（TZ）
EPWMxA
EPWMxB
GPIO
MUX
TZ1 to TZ3
EMUSTOP
CPU
CLOCKFAIL
SYSCTRL
EQEP1ERR
EQEP1
CTR=0
PIE
EPWMxTZINT
数字比较信号
数字比较
（DC）
COMPxOUT
COMP

八个子模块：

1.Time-base(TB) module

2.Counter-compare(CC) module

3.Action-qaulifier(AQ) module

4.Dead-band(DB) module

~~5.PWM-chopper(PC) module~~

6.Event-Trigger(ET) module

~~7.Trip-zone(TZ) module~~

~~8.Digital-compare(DC) module~~

八个子模块作用和关系：

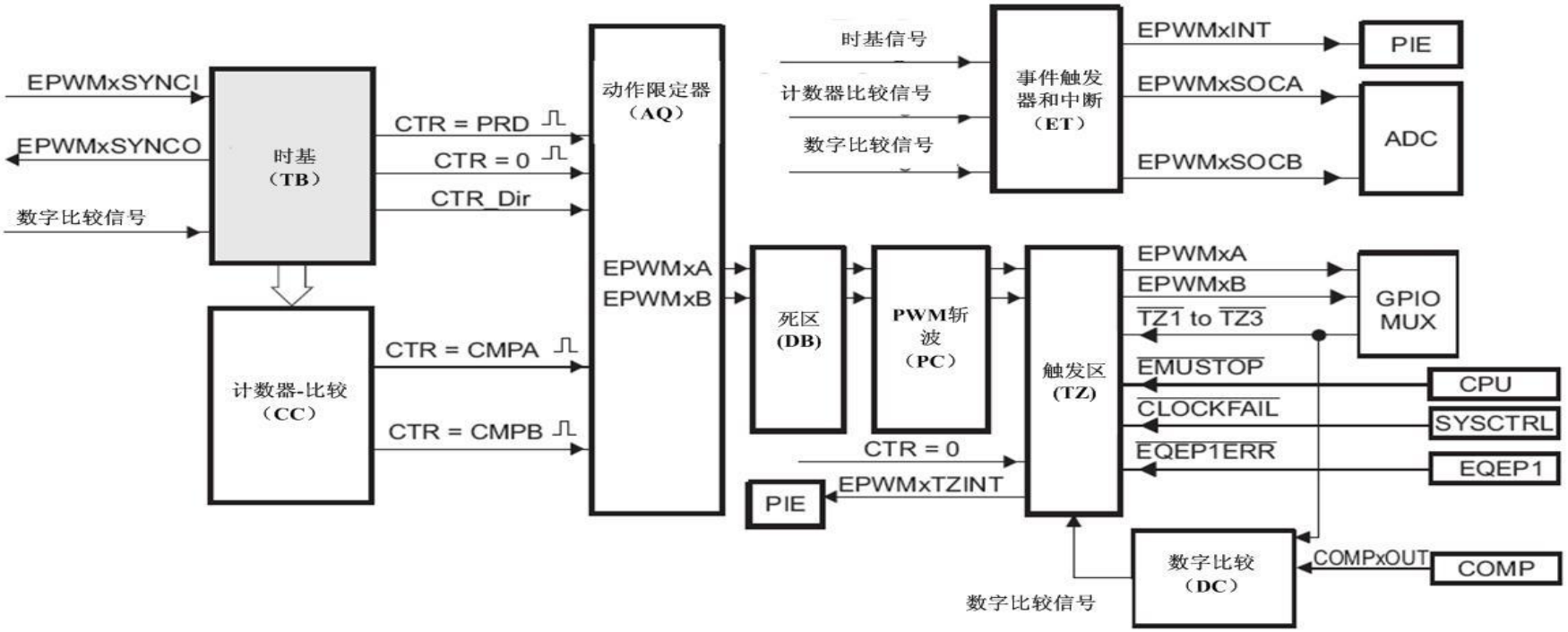
1. TB： PWM计数器方式、两个事件（最大点、过零点）和计数方向
2. CC： 产生两套比较事件，最多四个
3. AQ： 对TB和CC所产生事件的动作（置1,清0，翻转，空动作）
4. DB： 互补PWM的死区，AQ的生效方式
6. ET： AQ的附加作用，触发ADC和中断

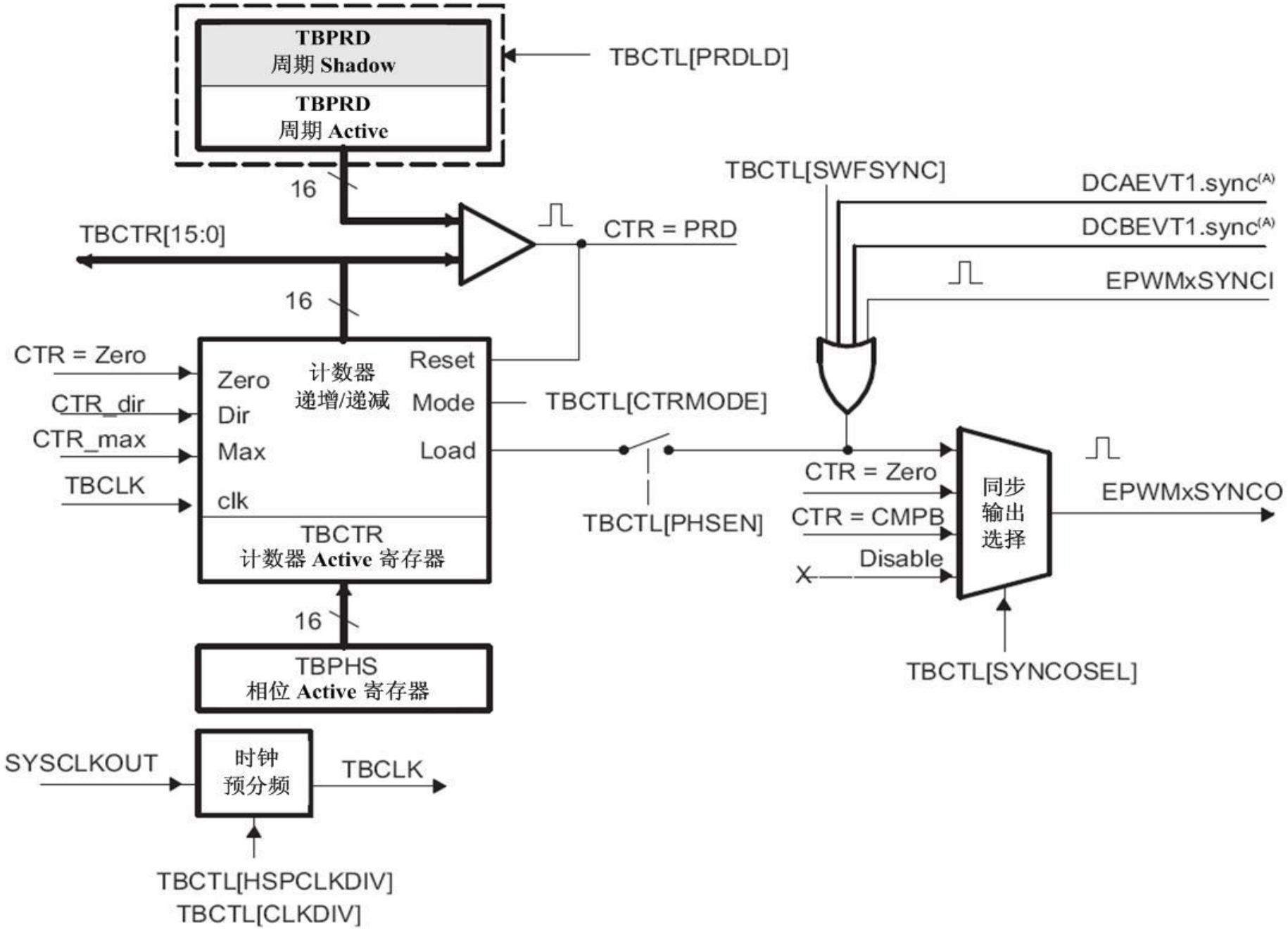
名称	偏移量	大小（×16）	映射	EALLOW	描述
					时基子模块寄存器
TBCTL	0x0000	1	否		时基控制寄存器
TBSTS	0x0001	1	否		时基状态寄存器
TBPHS	0x0003	1	否		时基相位寄存器
TBCTR	0x0004	1	否		时基计数器寄存器
TBPRD	0x0005	1	是		时基周期寄存器
					计数器-比较子模块寄存器
CMPCTL	0x0007	1	否		计数器-比较控制寄存器
CMPA	0x0009	1	是		计数器-比较A寄存器
CMPB	0x000A	1	是		计数器-比较B寄存器
					动作限定器子模块寄存器
AQCTLA	0x000B	1	否		输出A（EPWMxA）动作限定器控制寄存器
AQCTLB	0x000C	1	否		输出B（EPWMxB）动作限定器控制寄存器
AQSFRC	0x000D	1	否		动作限定器软件强制寄存器
AQCSFRC	0x000E	1	是		动作限定器连续S/W强制寄存器组
					死区发生器子模块寄存器
DBCTL	0x000F	1	否		死区发生器控制寄存器
DBRED	0x0010	1	否		死区发生器上升沿延迟计数寄存器
DBFED	0x0011	1	否		死区发生器下降沿延迟计数寄存器
					事件触发器子模块寄存器
ETSEL	0x0019	1			事件触发器选择寄存器
ETPS	0x001A	1			事件触发器预分频寄存器
ETFLG	0x001B	1			事件触发器标志寄存器
ETCLR	0x001C	1			事件触发器清零寄存器
ETFRC	0x001D	1			事件触发器强制寄存器

9.2 ePwm的子模块

9.2.1 Time-base(TB) module

- 每个ePWM模块都有自己的时基子模块，它决定ePWM模块所有事件的时序。
- 内置的同步逻辑电路使得几个ePWM模块的时基可以当作一个系统一起工作。





注A：这些信号由数字比较（DC）子模块产生

信号	描述
EPWMxSYNCI	时基同步输入 该输入脉冲用于同步时基计数器与稍早位于同步链的ePWM模块的计数器。ePWM外设可以配置成使用或忽略这个信号。对于第一个ePWM模块（EPWM1）而言，该信号来自器件管脚。对于其他ePWM模块而言，该信号是从另外一个ePWM模块传输过来的。例如，EPWM2SYNCI由ePWM1外设产生，EPWM3SYNCI由ePWM2产生，依此类推。要了解特定器件的同步顺序，请参阅.2)小节“时基计数器的同步”
EPWMxSYNCO	时基同步输出 该输出脉冲用于同步稍后位于同步链的ePWM模块的计数器。ePWM模块在出现以下其中一种事件时产生该信号： 1. EPWMxSYNCI（同步输入脉冲） 2. CTR = 0：时基计数器的计数值等于0（TBCTR = 0x0000） 3. CTR = CMPB：时基计数器的计数值等于计数器-比较B（TBCTR = CMPB）寄存器
CTR = PRD	时基计数器等于指定周期 每当计数器值等于active周期寄存器值时便产生该信号。即TBCTR = TBPRD时
CTR = Zero	时基计数器等于0 每当计数器值等于零时便产生该信号。即TBCTR等于0x0000时

信号	描述
CTR = CMPB	时基计数器等于active计数器-比较B寄存器（TBCTR = CMPB）该事件由计数器-比较子模块产生，供同步输出逻辑电路使用
CTR = dir	时基计数器方向 用于指示ePWM时基计数器的当前方向。计数器递增时该信号为高电平，计数器递减时该信号为低电平
CTR = max	时基计数器等于最大值（TBCTR = 0xFFFF） TBCTR值到达其最大值时产生的事件。该信号仅被用作状态位
TBCLK	时基时钟 这是系统时钟（SYSCLKOUT）经过预分频得到的值，供ePWM内的所有子模块使用。该时钟决定时基计数器递增或递减的速率

1. 时基子模块的用途

- 配置时基时钟的速率；CPU系统时钟（SYSCLKOUT）经过预分频得到的值。时基计数器从而能够以较慢的速率递增/递减
- 指定ePWM时基计数器（TBCTR）的频率或周期，以控制事件发生的频率
- 管理与其他ePWM模块之间的时基同步，维持与其他ePWM模块之间的相位关系
- 时基计数器三种计数模式：递增、递减或者“先递增后递减”
- 产生以下事件：

  CTR = PRD（TBCTR = TBPRD）和CTR = 0（TBCTR = 0x0000）

主要用途：

简而言之：

- TB模块决定PWM的时钟、
- 计数器工作模式、
- 计数器周期（PWM周期），
- 产生计数值等于PWM周期和计数值等于0两个事件，
- 并实现PWM单元间的同步。

2. 计算PWM周期和频率

时基计数器三种工作模式（TBCTL选择）：

先递增后递减模式(Up-Down mode)：

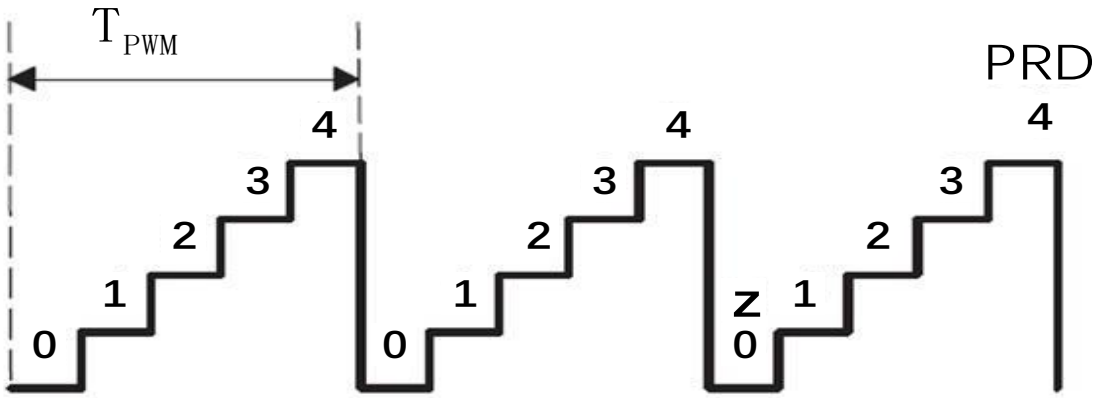
时基计数器从零开始递增计数直到到达周期值（TBPRD），然后在达到周期值时，时基计数器又开始递减直到为0，然后计数器重复这种模式又开始递增。

递增模式(Up mode)：

时基计数器从0开始递增计数直到到达周期寄存器（TBPRD）中的值。当达到周期值时，时基计数器复位到0并再次开始递增计数。

递减模式(Down mode)：

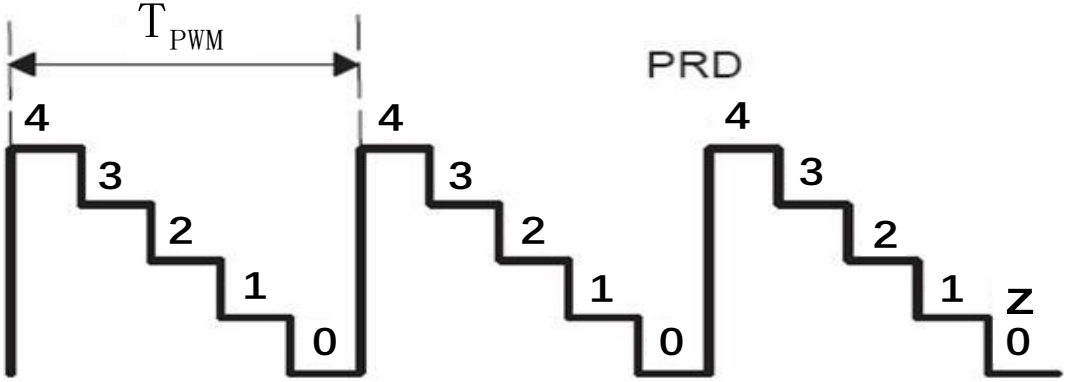
时基计数器从周期（TBPRD）值开始递减计数直到为0。当它到达0时，时基计数器复位到周期值并再次开始递减计数。

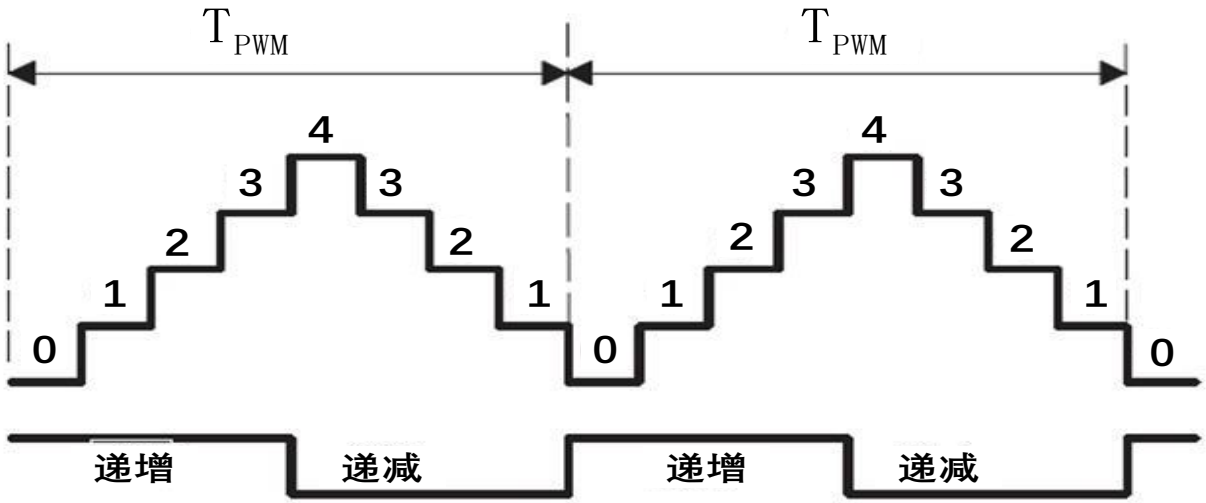


递增或递减计数模式:

$T_{PWM} = (TBPRD+1) \times T_{TBCLK}$

$F_{PWM} = 1/(T_{PWM})$





递增或递减计数模式:

$T_{PWM} = 2 \times TBPRD \times T_{TBCLK}$

$F_{PWM} = 1/(T_{PWM})$

3. 时基周期的映射（shadow）寄存器

时基周期寄存器（TBPRD）包含一个映射寄存器。映射寄存器使得寄存器的更新与硬件同步。

定义用于描述ePWM模块中的**所有**映射寄存器：

- **有效（active）寄存器**
  有效寄存器控制硬件，并且负责由硬件导致或引发的动作

- **映射（shadow）寄存器**
  映射寄存器缓冲或暂时保存有效寄存器的存储单元，它不直接影响任意控制硬件。在特定的时刻，映射寄存器的内容会被转移到有效寄存器，这样做可以避免因寄存器被软件异步修改而发生的错误或虚假操作。

映射周期寄存器的存储器地址与有效寄存器相同。
哪个寄存器被写或被读由TBCTL[PRDLD]位决定。

- **时基周期映射模式：**
  TBPRD 映射寄存器在TBCTL[PRDLD] = 0时使能。向TBPRD存储器地址执行读写操作便可以进入映射寄存器。当时基计数器等于0（TBCTR = 0x0000）时，映射寄存器的内容被转移到有效寄存器（TBPRD (有效)←TBPRD（映射）。缺省情况下，TBPRD 映射寄存器是使能的。

- **时基周期立即装载模式：**
  如果选中立即装载模式（TBCTL[PRDLD] = 1），那么向TBPRD存储器地址执行读或写操作都可以直接进入有效寄存器。

4. 时基计数器的同步机制

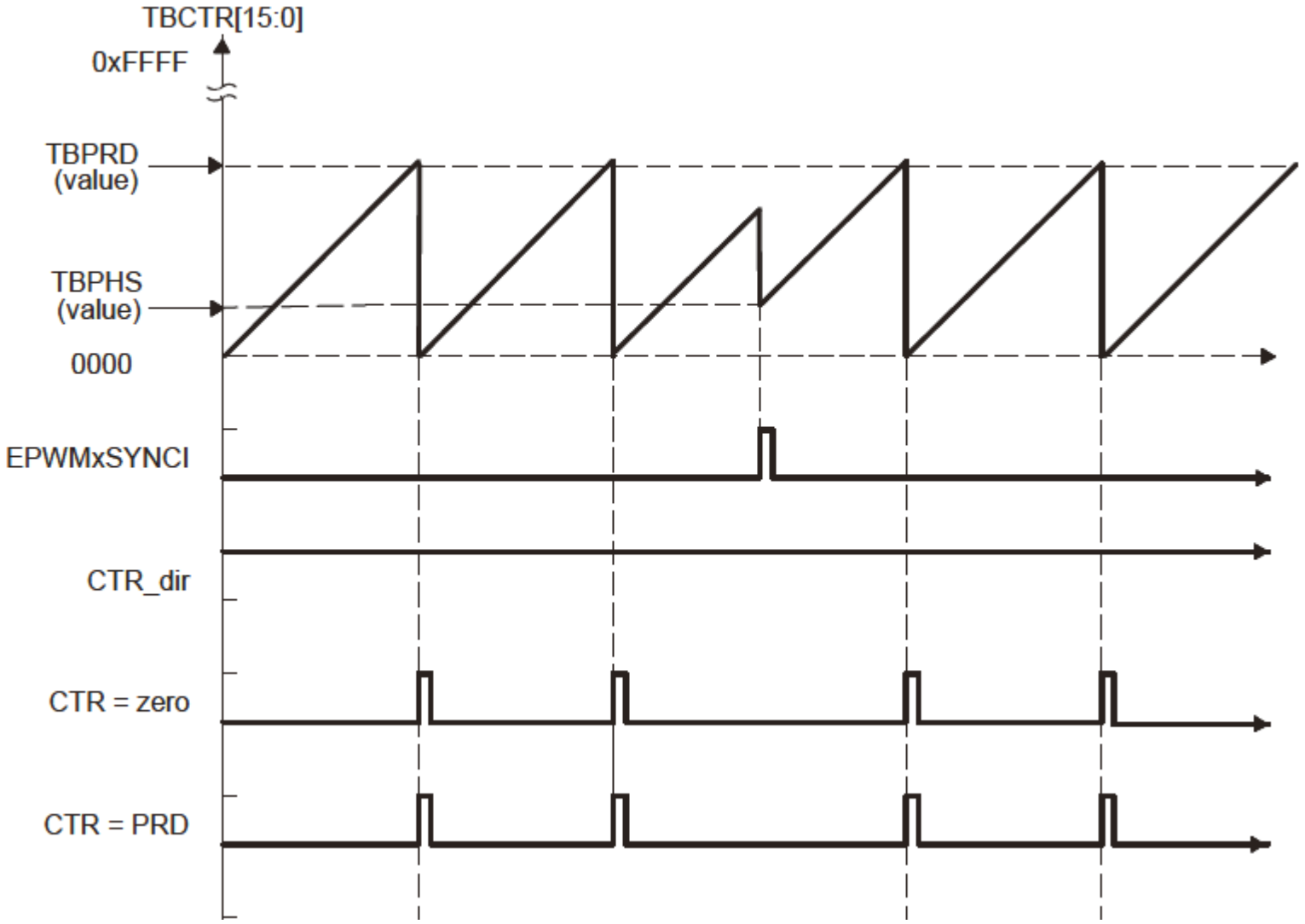
时基同步机制连接了器件上的所有**ePWM**模块。

每个**ePWM**模块都包含**:**

一个同步输入信号（**EPWMxSYNCI**）
一个同步输出信号（**EPWMxSYNCO**）。

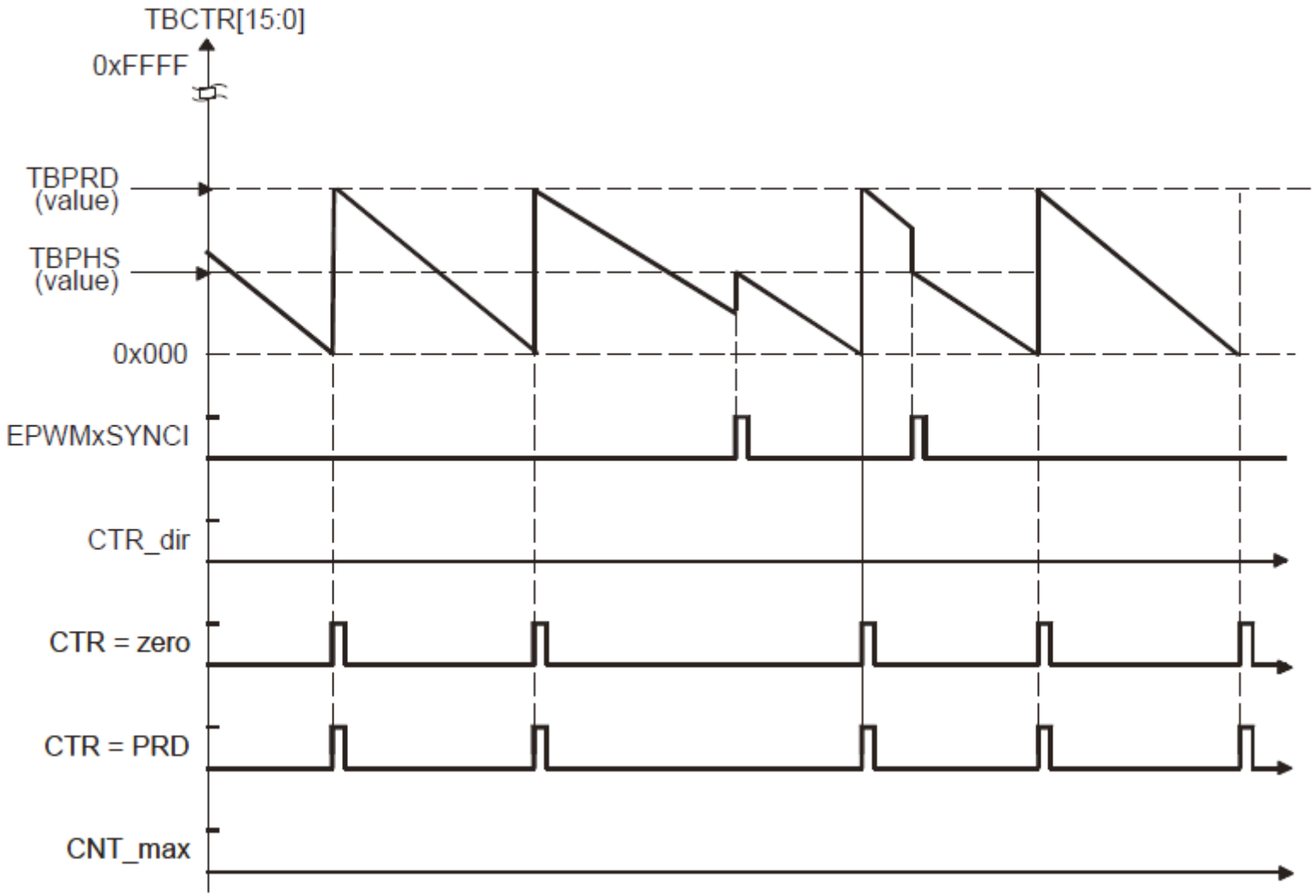
相位控制(up)

Figure 10. Time-Base Up-Count Mode Waveforms



down

Figure 11. Time-Base Down-Count Mode Waveforms



up
-down

Figure 12. Time-Base Up-Down-Count Waveforms, TBCTL[PHSDIR = 0] Count Down On Synchronization Event

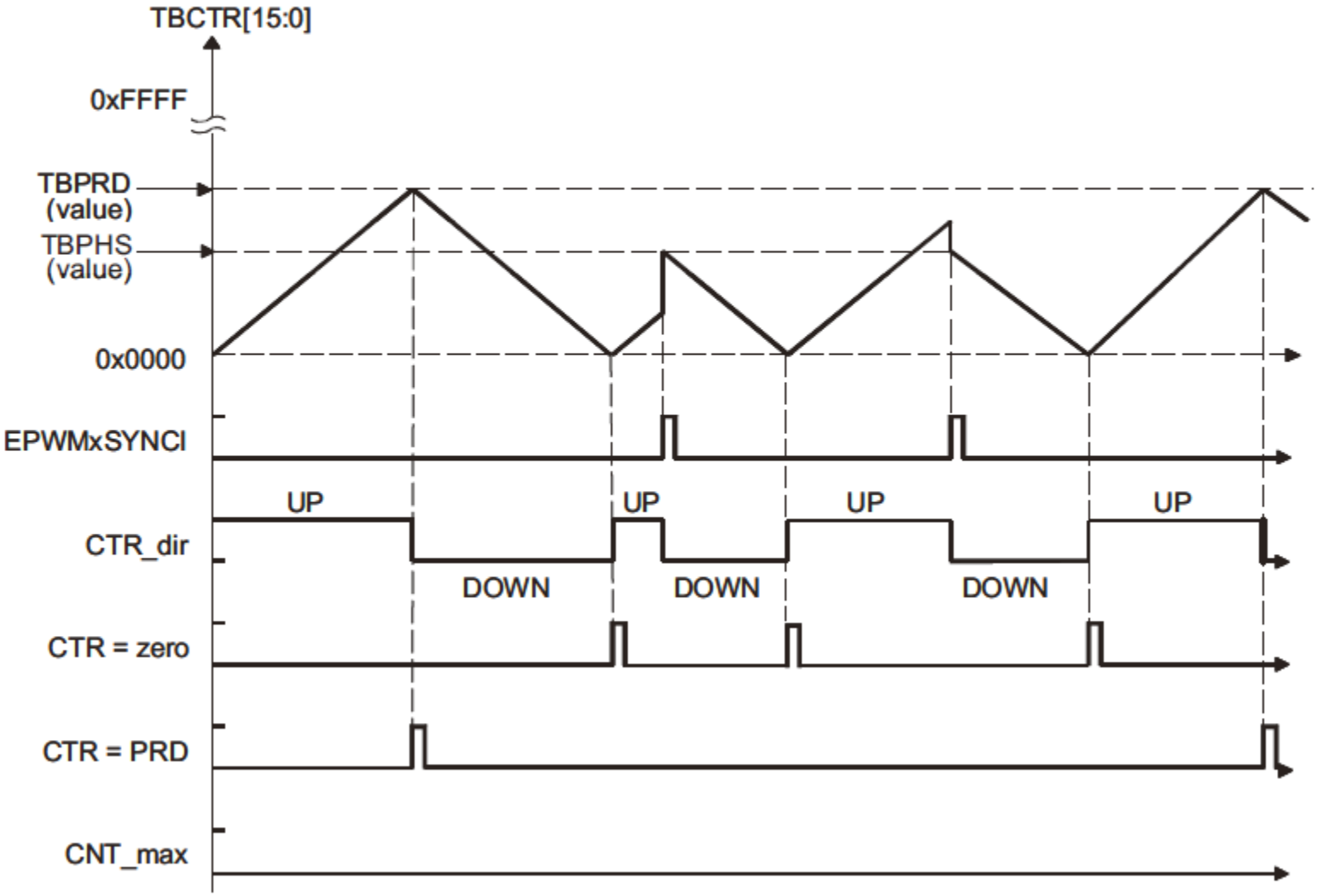
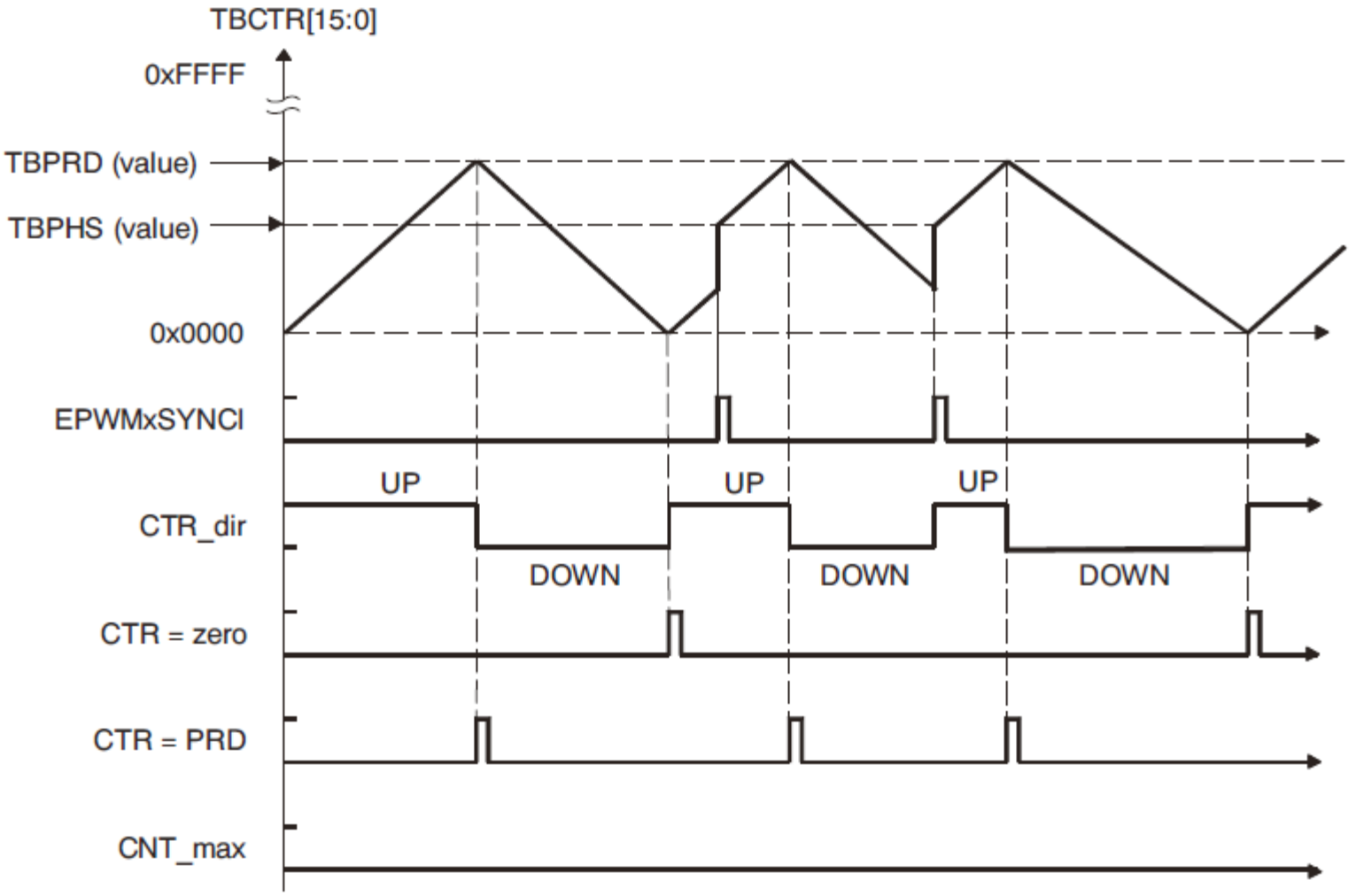
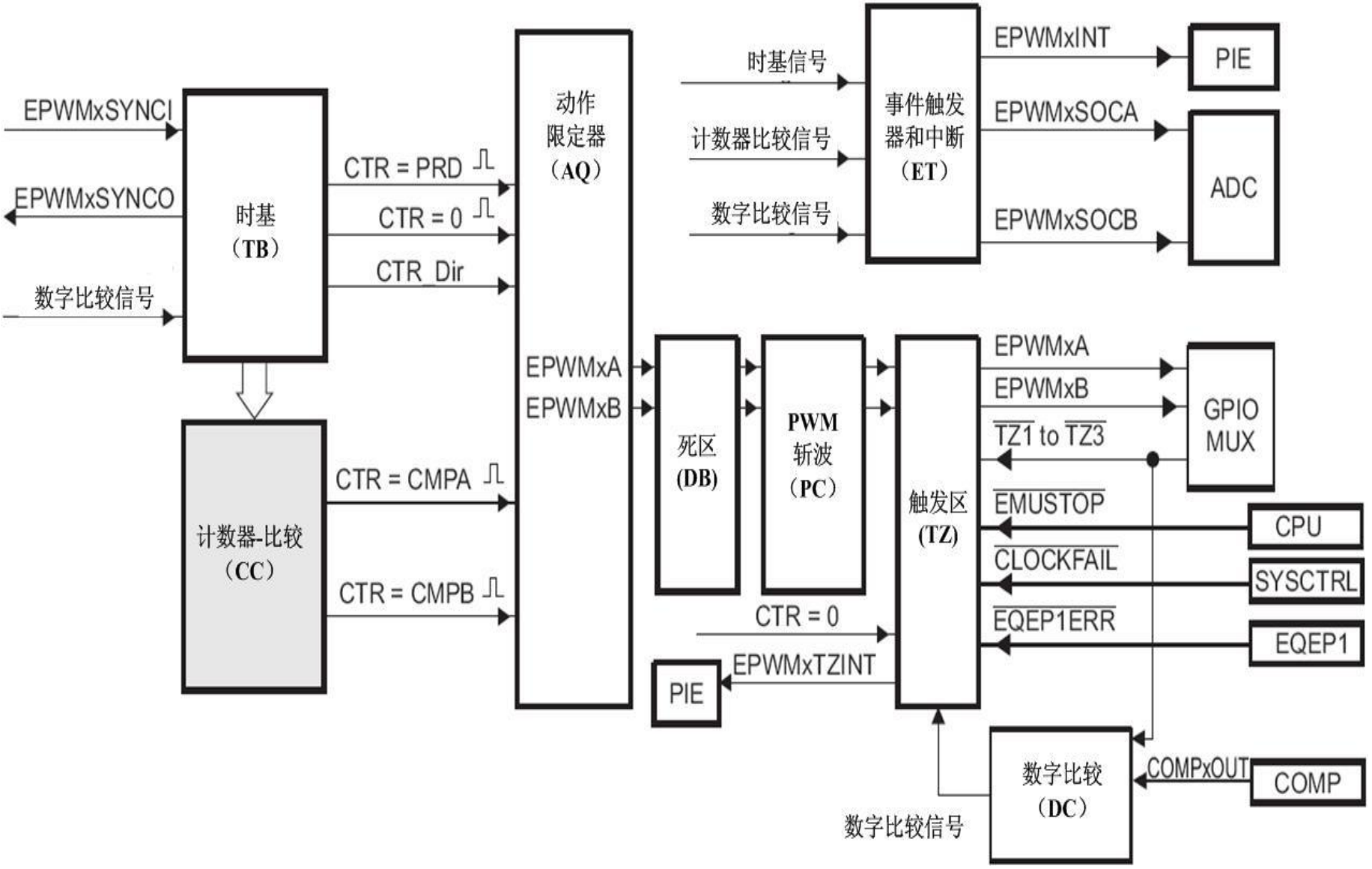
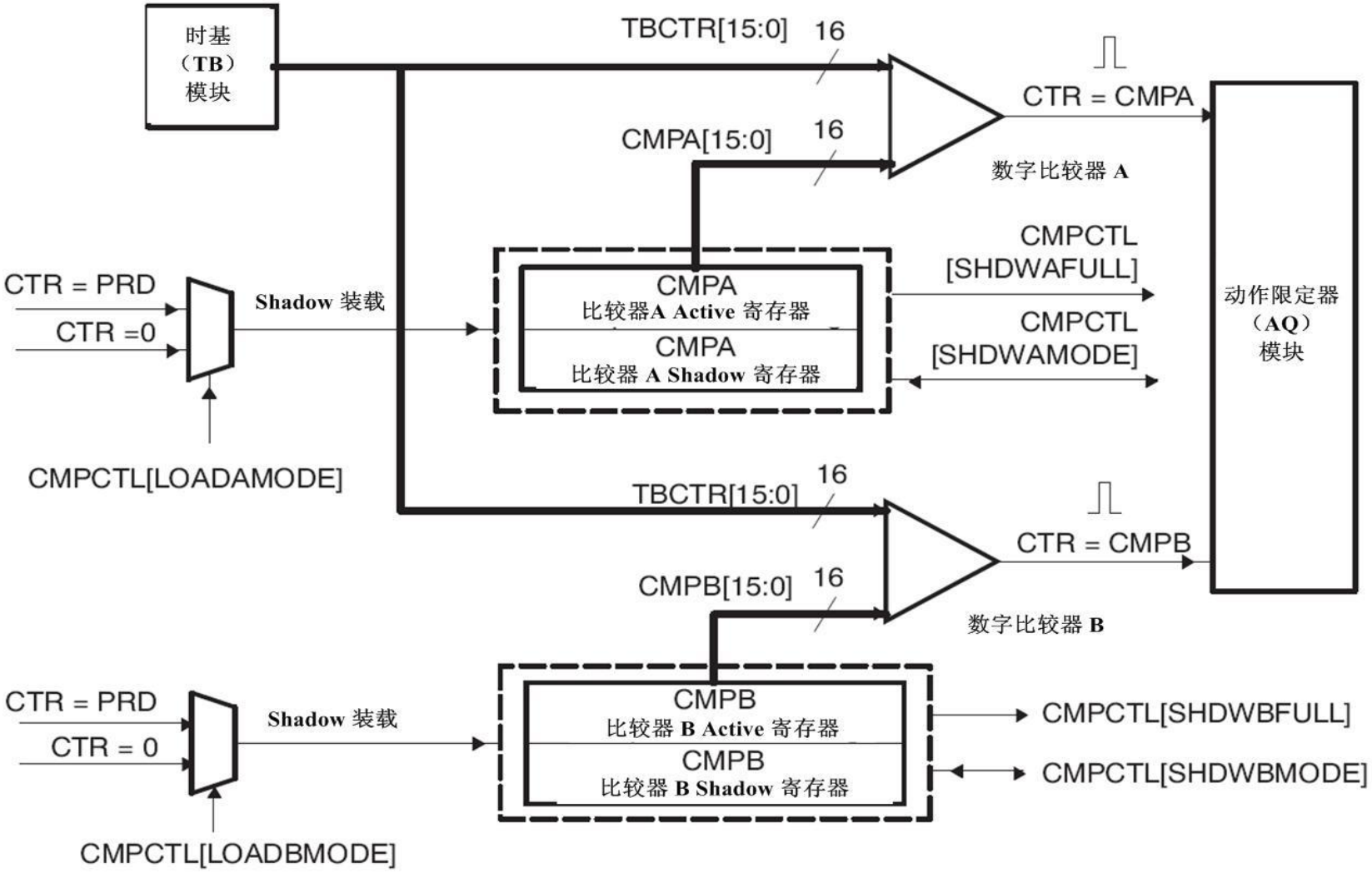


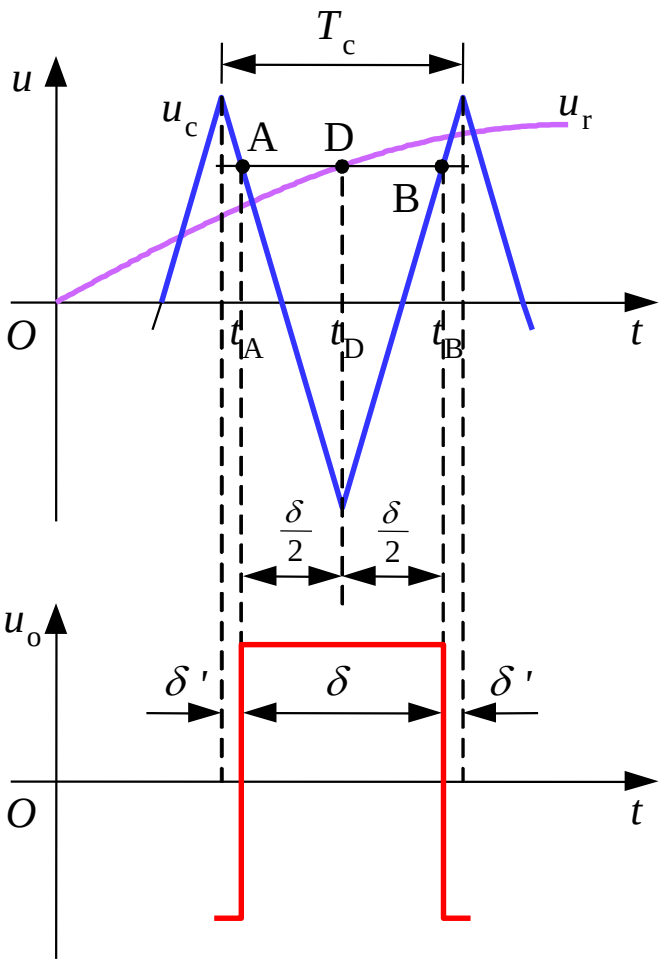
Figure 13. Time-Base Up-Down Count Waveforms, TBCTL[PHSDIR = 1] Count Up On Synchronization Event



9.2.2 计数器-比较（CC）子模块

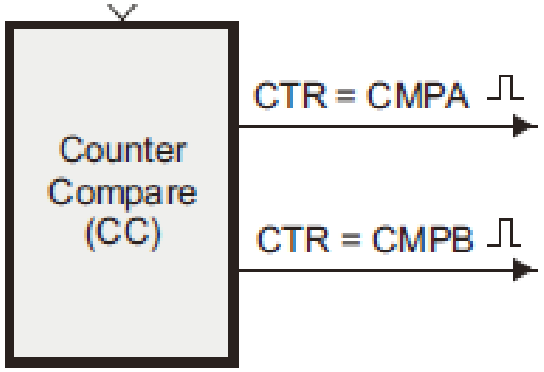



时基
（TB）
模块
TBCTR[15:0]
16
CTR = CMPA
CMPA[15:0]
16
数字比较器 A
CMPCTL
[SHDWAFULL]
CMPA
比较器A Active 寄存器
CMPA
比较器 A Shadow 寄存器
CMPCTL
[SHDWAMODE]
CTR = PRD
CTR =0
Shadow 装载
CMPCTL[LOADAMODE]
动作限定器
（AQ）
模块
TBCTR[15:0]
16
CTR = CMPB
CMPB[15:0]
16
数字比较器 B
CMPB
比较器 B Active 寄存器
CMPB
比较器 B Shadow 寄存器
CMPCTL[SHDWBFULL]
CMPCTL[SHDWBMODE]
CTR = PRD
CTR = 0
Shadow 装载
CMPCTL[LOADBMODE]


T_c
u
u_c
u_r
A
D
B
O
t_A
t_D
t_B
t
$\frac{\delta}{2}$
$\frac{\delta}{2}$
u_o
δ'
δ
δ'
O
t

模块功能:

当计数值到达设定的比较值时，触发相应的事件，如下图所示



CMPA=TBCTR
或 CMPB=TBCTR时
模块会给出事件触发信号给下一个模块（AQ）进行处理。

信号	事件描述	寄存器比较
CTR = CMPA	时基计数器的计数值等于计数器-比较Active寄存器的值	TBCTR = CMPA
CTR = CMPB	时基计数器的计数值等于计数器-比较Active寄存器的值	TBCTR = CMPA
CTR = PRD	时基计数器的计数值等于active周期。 用于从shadow寄存器那里装载active计数器-比较A和B寄存器	TBCTR = TBPRD
CTR = ZERO	时基计数器的计数值等于0。 用于从shadow寄存器那里装载active计数器-比较A和B寄存器	TBCTR = 0x0000

1. 计数器-比较子模块的工作要点

- 计数器-比较子模块负责根据两个比较寄存器产生两个独立的比较事件：
  CTR = CMPA： 时基计数器等于计数器-比较A寄存器
  CTR = CMPB： 时基计数器等于计数器-比较B寄存器
- 如果为递增或递减模式，每个事件每个周期仅发生一次。
- 如果为“先递增后递减”模式且比较值在0x0000～TBPRD之间，则每个事件每个周期发生两次，若比较值等于0x0000或等于TBPRD，则每个事件每个周期发生一次。
- 这些事件被馈送到动作限定器子模块，在那里，它们受计数器方向的限制并在使能时被转换成各种动作。

- 计数器-比较寄存器CMPA和CMPB都包含相关的shadow寄存器。shadow寄存器使得寄存器的更新与硬件同步。
- 当使用了shadow寄存器时，active寄存器只在重要时刻进行更新。这样可以避免因寄存器因软件异步修改而发生的讹误或谬误操作。
- active寄存器的存储器地址与shadow寄存器相同。至于哪个寄存器被写或被读，这由CMPCTL[SHDWAMODE]和CMPCTL[SHDWBMODE]位决定。这些位可以分别使能和禁用CMPA shadow寄存器和CMPB shadow寄存器。

shadow模式：

立即装载模式：

shadow模式：

CMPCTL[SHDWAMODE]=0 或CMPCTL[SHDWBMODE]=0

由CMPCTL[LOADAMODE]和CMPCTL[LOADBMODE]寄存器位决定

对于CMPA：

LOADAMODE=0：时基计数器的计数值等于周期（TBCTR = TBPRD）

LOADAMODE=1：时基计数器的计数值等于零（TBCTR = 0x0000）

LOADAMODE=2：时基计数器的计数值等于周期或零

对于CMPB：

LOADBMODE=0：时基计数器的计数值等于周期（TBCTR = TBPRD）

LOADBMODE=1：时基计数器的计数值等于零（TBCTR = 0x0000）

LOADBMODE=2：时基计数器的计数值等于周期或零

计数器-比较子模块只对有效寄存器的内容进行比较。

立即装载模式：

CMPCTL[SHDWAMODE]=1 或CMPCTL[SHDWBMODE]=1

有效寄存器装载：

相对应的CMPA或CMPB寄存器操作都会直接操作有效寄存器。

2. 计数模式的时序波形

CC模块在以下三种计数模式下都会产生比较事件：

- 递增计数模式：用来产生一个非对称的PWM波形
- 递减计数模式：用来产生一个非对称的PWM波形
- 先递增后递减模式：用来产生一个对称的PWM波形
- 一般情况下比较值不能设置为比周期大，如果比较值设置为比周期大可能会发生不会产生比较动作的情况

9.2.3 限定器（AQ）子模块

动作限定器子模块在构造波形以及产生PWM方面扮演着最重要的角色。它决定将哪些事件转换成哪种动作，从而在EPWMxA和EPWMxB输出上产生所需的开关波形。

动作限定器子模块的用途

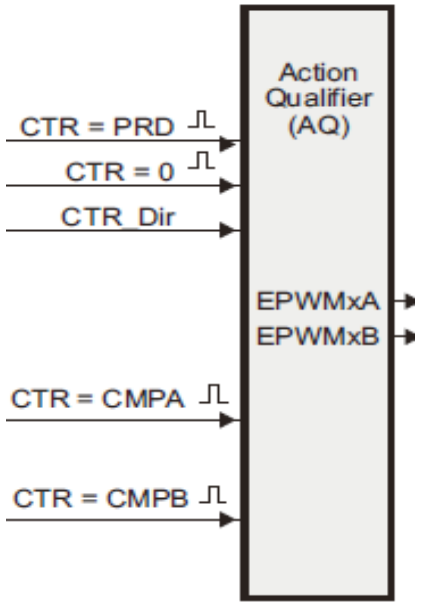
动作限定器子模块负责以下事情：

根据以下事件限定并产生动作（**置位、清零、切换**）

- CTR = PRD
- CTR = Zero
- CTR = CMPA
- CTR = CMPB

在这些事件同时发生时管理优先级

在时基计数器递增和递减时单独控制这些事件



施加到EPWMxA和EPWMxB输出信号上的动作可能是：

置1（高电平）：

将EPWMxA或EPWMxB输出信号设置成高电平

清零（低电平）：

将EPWMxA或EPWMxB输出信号设为低电平

高低电平切换（Toggle）：

如果当前EPWMxA或EPWMxB被拉高，则将输出拉低。

如果当前EPWMxA或EPWMxB被拉低，则将输出拉高。

Do Nothing（不采取任何动作）：

EPWMxA或EPWMxB输出信号保持为当前设置值。尽管“Do Nothing”选项使得事件不可以在EPWMxA和EPWMxB输出信号上采取动作，但该事件仍然可以触发中断和ADC开始转换。

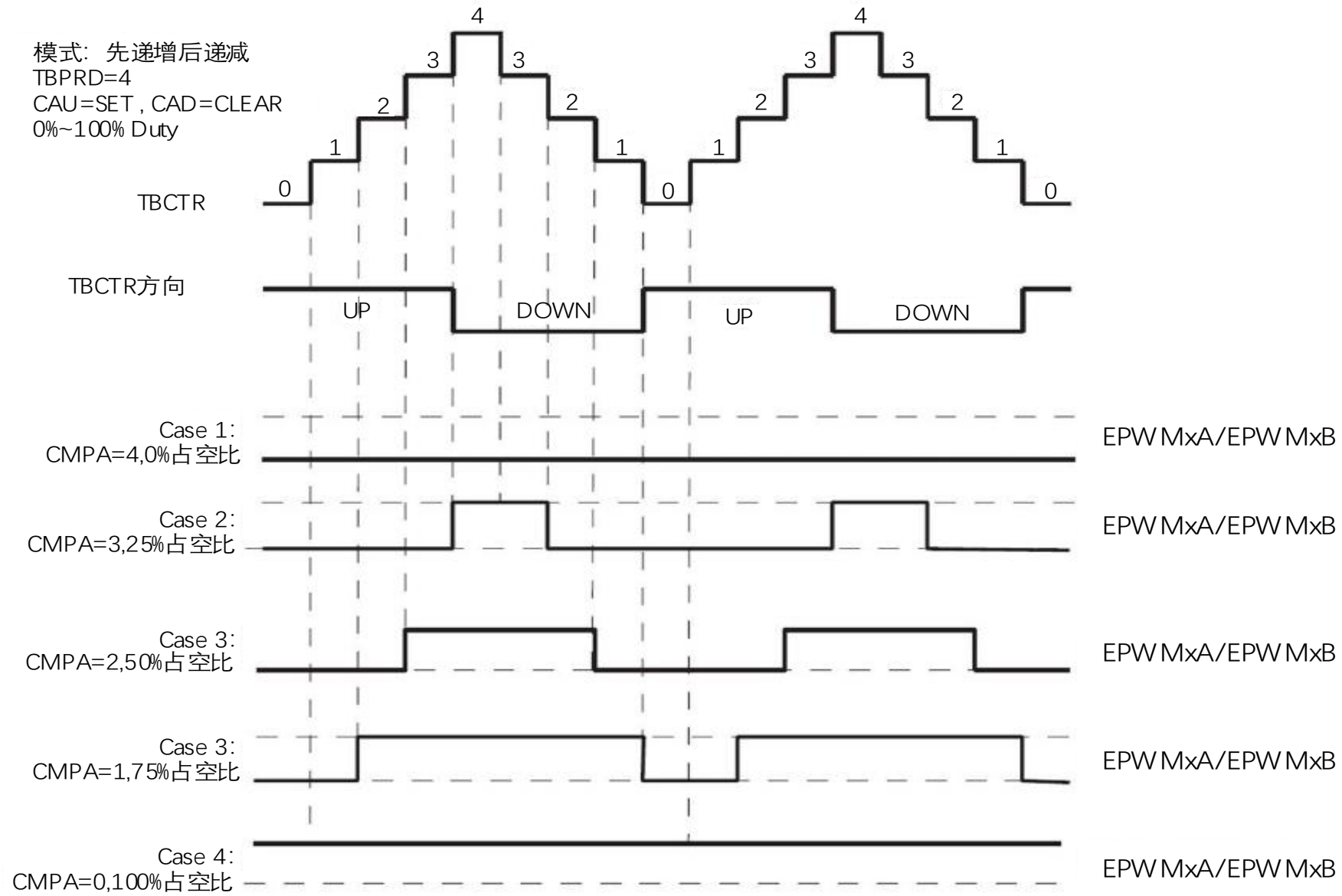
每个输出信号（EPWMxA或EPWMxB）都被单独指定了动作。

所有事件都可以配置成在某个特定输出上产生动作。

例如，CTR = CMPA和CTR = CMPB都可以在EPWMxA输出上产生动作。

动作限定器事件的优先级

使用“先递增后递减”计数模式产生一个对称的PWM：

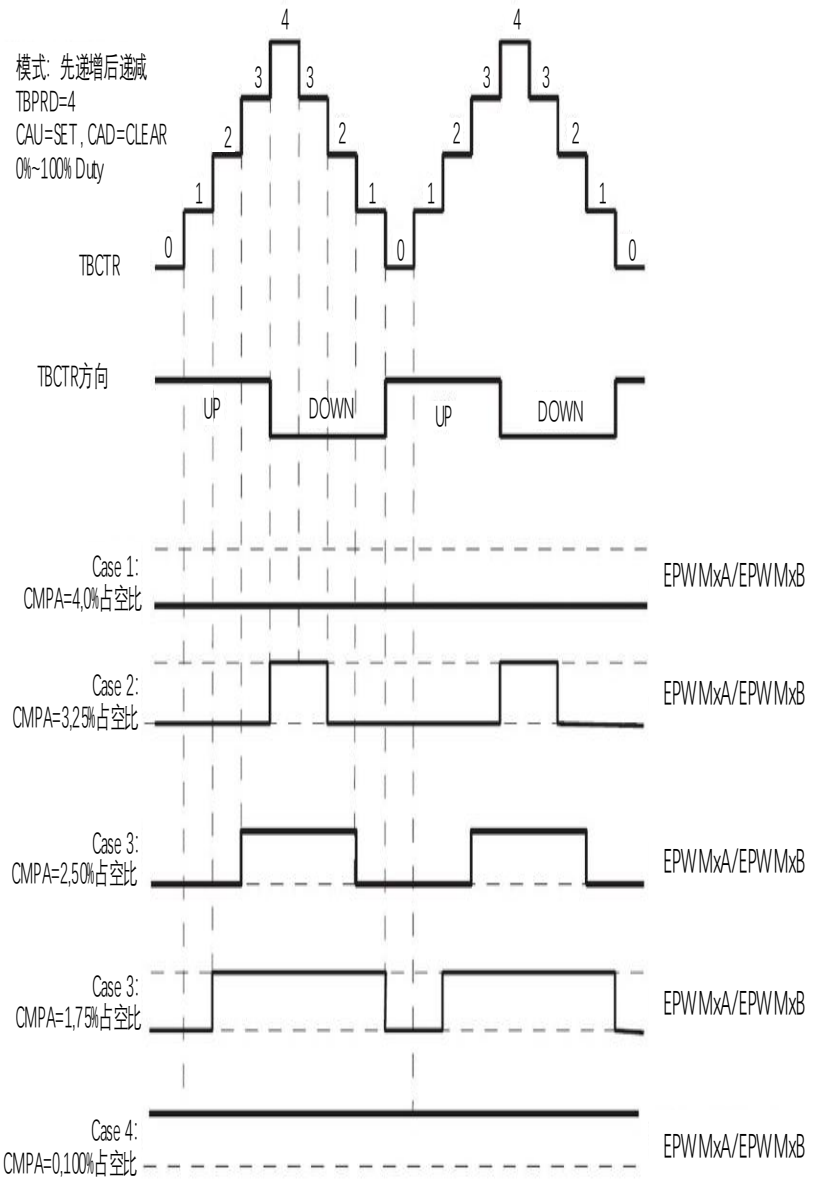


如何输出指定占空比 λ 的PWM？

1. 确定PWM的周期Tc

2. 确定动作限定器的输出：
   （1）CAU=SET / CAD=CLEAR
   （2）CAU=CLEAR / CAD=SET

3. **计算CMPA**
   （1）CMPA = Tc * （1- λ）
   （2）CMPA = Tc * λ



如何实现对称PWM输出？
（规则采样法PWM/单更新）

1. 确定PWM的周期Tc

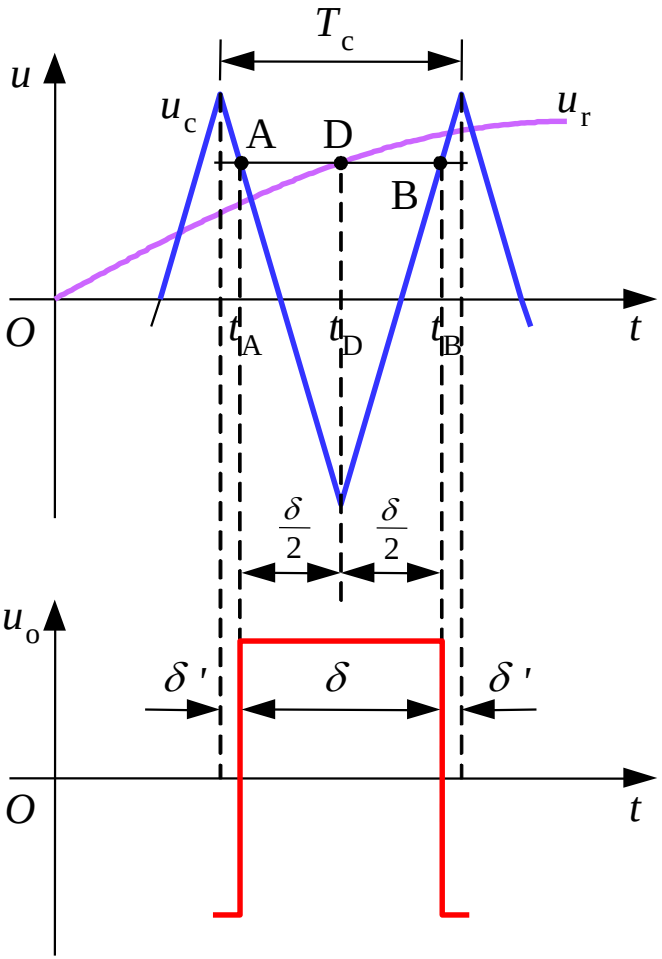
2. 确定动作限定器的输出：
   CAD=SET / CBU=CLEAR
   CAD=SET / CAU=CLEAR

3. 根据 δ 计算CMPA

   $u_r = a\sin\omega t$

   $\delta = 0.5Tc(1+a\sin\omega t_D)$

   CMPA = δ , CMPB = δ



如何实现不对称PWM输出？
（自然采样PWM/双更新）

1. 确定PWM的周期Tc

2. 确定动作限定器的输出：

   CAD=SET / CBU=CLEAR

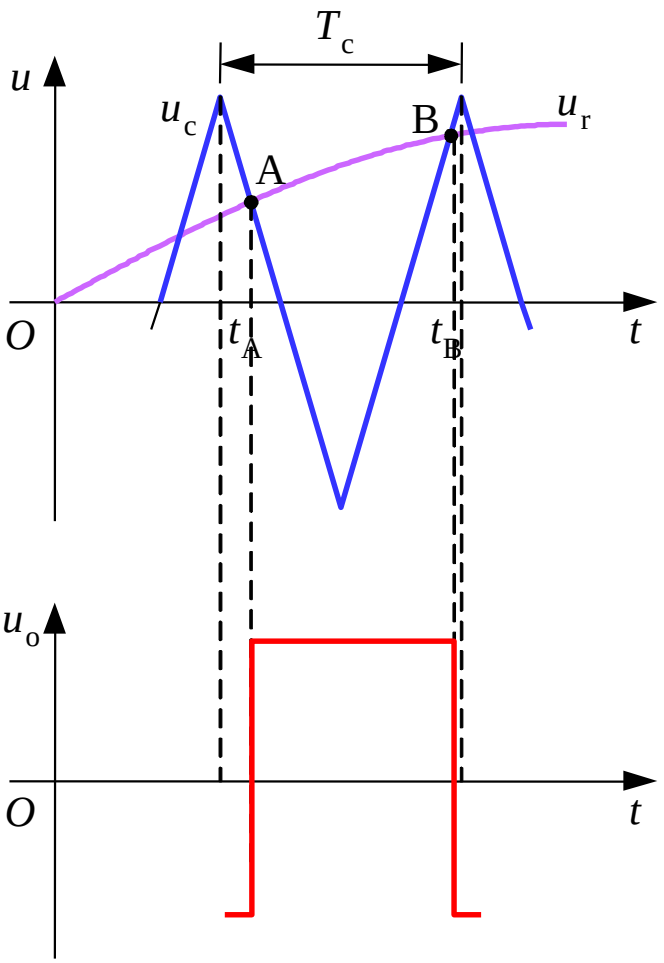
   CAD=SET / CAU=CLEAR

3. **计算CMPA和CMPB**

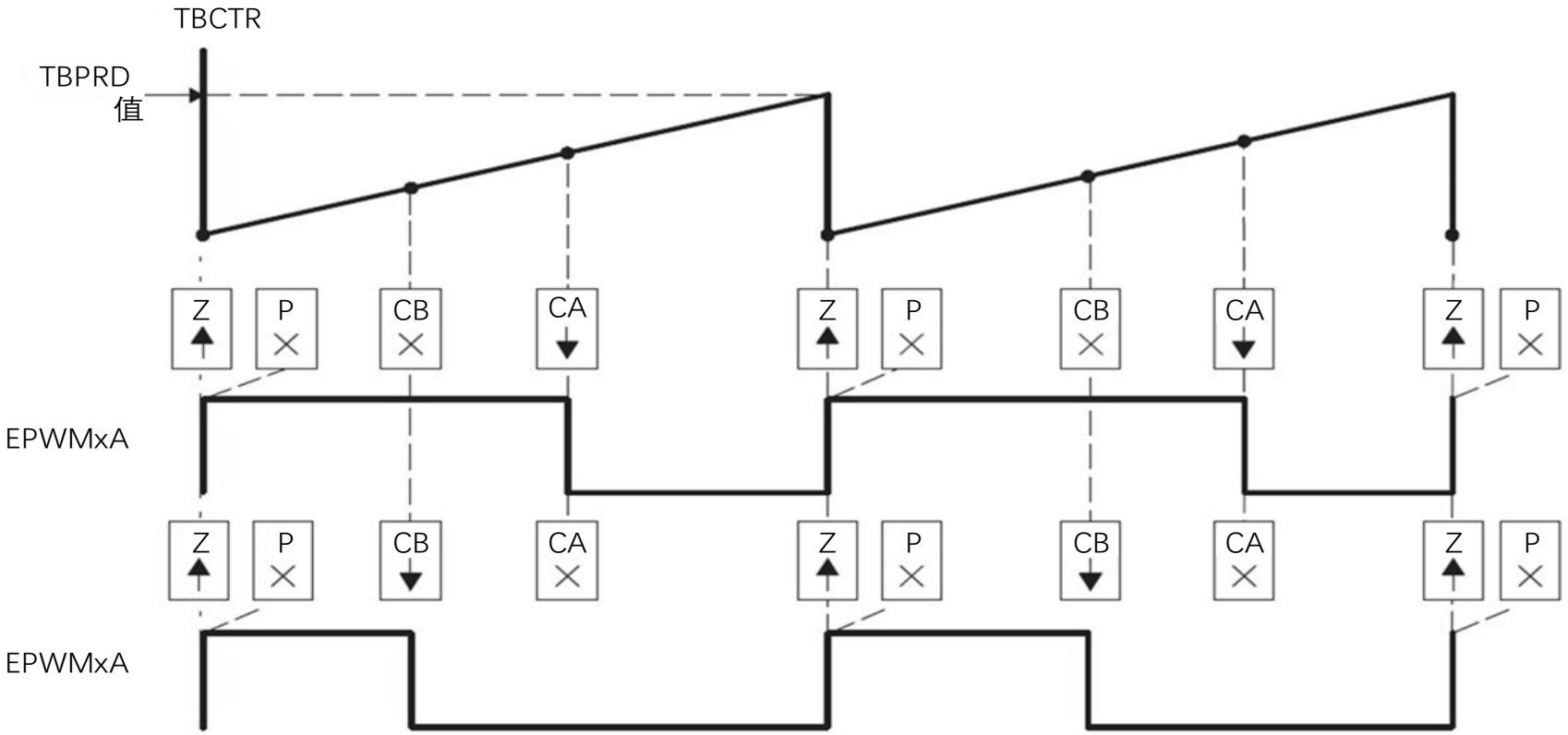
   CMPA = 0.5Tc(1+asinωtA)

   CMPB = 0.5Tc(1+asinωtB)

   CMPA = 0.5Tc(1+asinωtB)



使用递减计数模式或递增模式产生一个非对称的PWM：

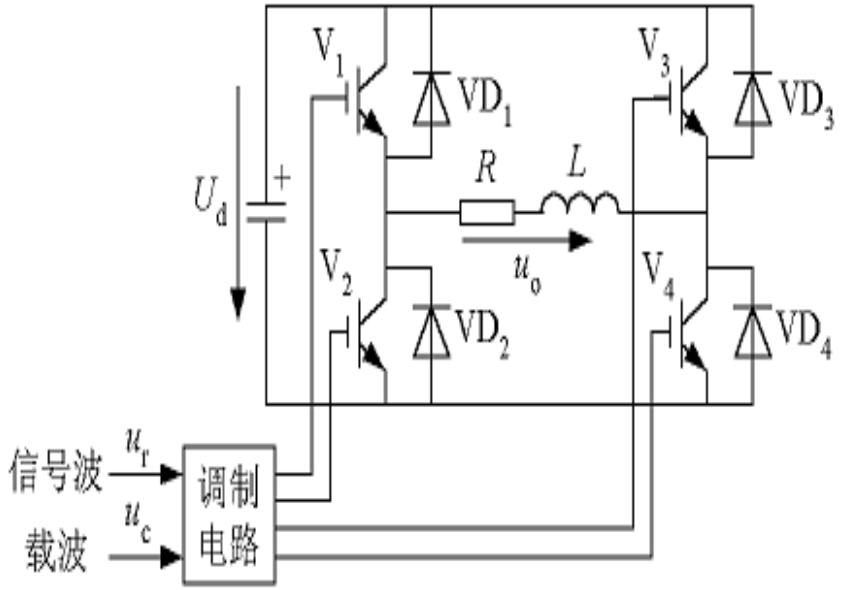


$$\text{PWM period} = (\text{TBPRD} + 1) \times T_{\text{TBCLK}}$$

9.2.4 死区发生器（DB）子模块

为什么需要死区（DB）模块：

在利用dsp对整流器或逆变器进行控制时，为了避免桥臂的上下管同时导通而导致短路，需要对输出PWM信号的上升沿或者下降沿进行延时，也就是设置PWM死区。



死区子模块的主要功能如下：

- 产生具有死区的信号对（EPWMxA和EPWMxB），可从单个EPWMxA或EPWMxB输入得到
- 将信号对编程为：
  高电平有效（AH）、低电平有效（AL）、
  高电平有效互补（AHC）和低电平有效互补（ALC）
- 在上升沿添加可编程延迟（RED）
- 在下降沿添加可编程延迟（FED）
- 在信号通道可完全旁路

死区模块相关术语

1. AH（Active High）：高有效
2. AL（Active Low）：低有效
3. AHC（Active High Complementary）：互补高有效
4. ALC（Active Low Complementary）：互补低有效
5. RED:上升沿延时
6. FED:下降沿延时

寄存器名称	地址偏移量	是否有shadow寄存器	描述
DBCTL	0x000F	否	死区控制寄存器
DBRED	0x0010	否	死区上升沿延迟计数寄存器
DBFED	0x0011	否	死区下降沿延迟计数寄存器

1. 死区子模块的使用要点

- **输入源选择：**
  选择死区模块的输入信号（通过使用DBCTL[IN_MODE]控制位）：

— EPWMxA In是下降沿延迟和上升沿延迟的信号源。这是缺省模式
— EPWMxA In是下降沿延迟的信号源，EPWMxB In是上升沿延迟的信号源
— EPWMxA In是上升沿延迟的信号源，EPWMxB In是下降沿延迟的信号源
— EPWMxB In是下降沿延迟和上升沿延迟的信号源

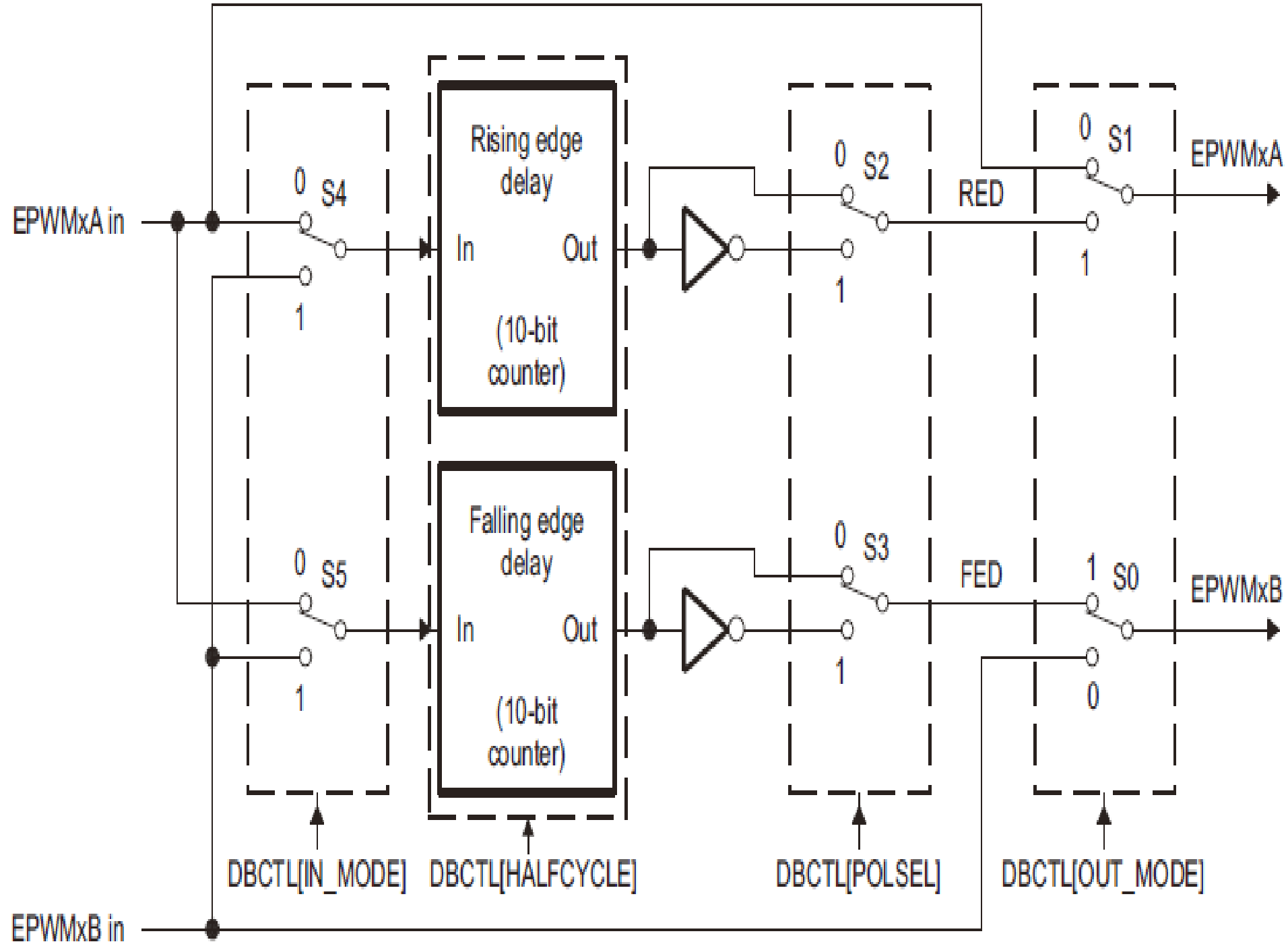
- **半周期计时：**
  死区子模块可以用半周期计时方式来计时，以便分辨率翻倍（即计数器以“2×TBCLK”计时）

全周期：$FED = DBFED \times T_{TBCLK}$，$RED = DBRED \times T_{TBCLK}$
半周期：$FED = DBFED \times T_{TBCLK}/2$，$RED = DBRED \times T_{TBCLK}/2$

1. 死区子模块的使用要点

- **输出模式控制：**
  输出模式由DBCTL[OUT_MODE]位配置，决定输入信号是添加了下降沿延迟，还是上升沿延迟或者是下降沿与上升沿两种延迟，亦或两种延迟都没有添加。

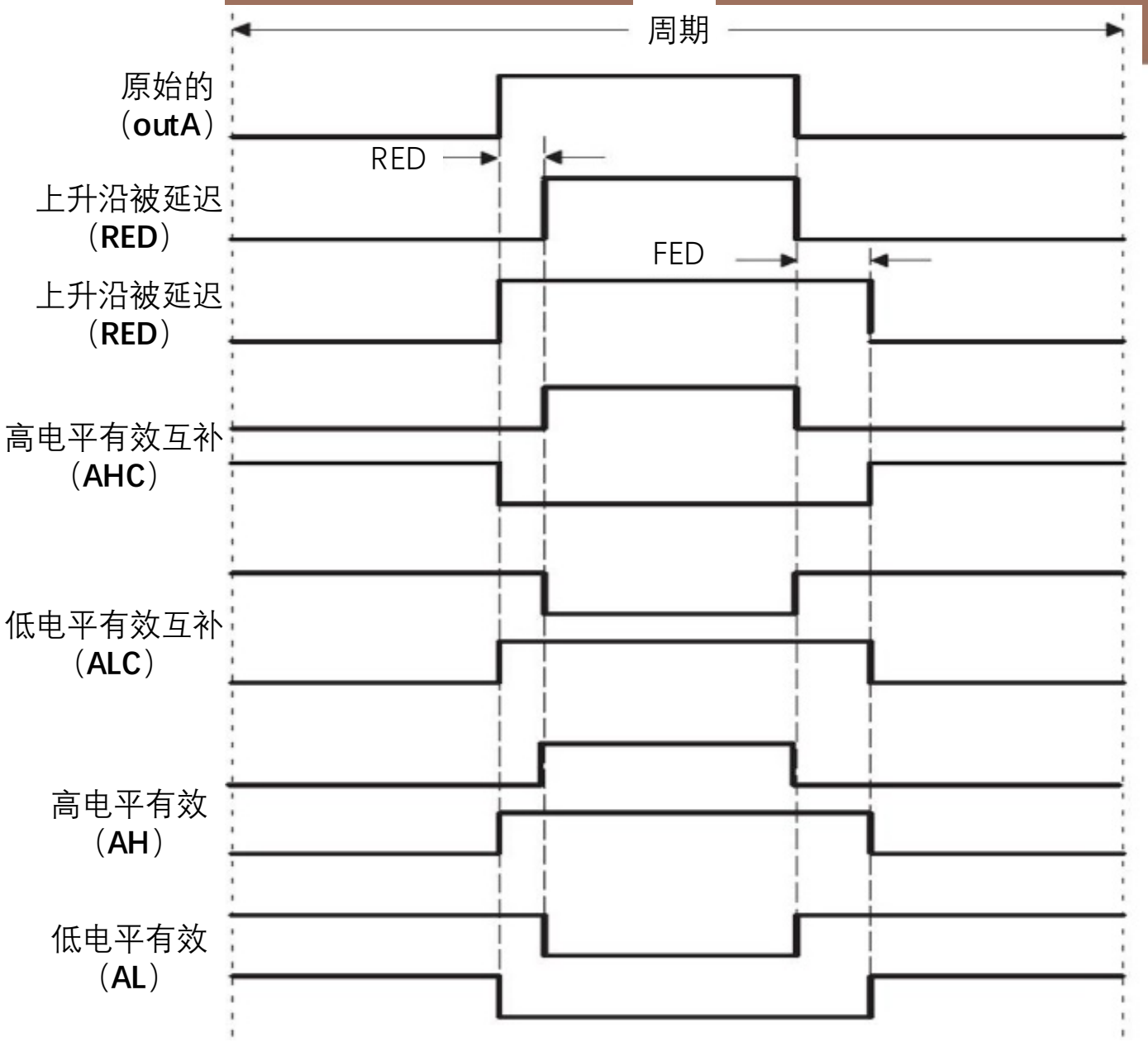
- **极性控制**
  极性控制（DBCTL[POLSEL]让用户可以确定上升沿延迟信号和/或下降沿延迟信号在从死区子模块发送出去之前是否被反相。


EPWMxA in
EPWMxB in
S4
S5
S2
S3
S1
S0
0
1
Rising edge delay
Falling edge delay
In
Out
(10-bit counter)
RED
FED
EPWMxA
EPWMxB
DBCTL[IN_MODE]
DBCTL[HALFCYCLE]
DBCTL[POLSEL]
DBCTL[OUT_MODE]

DB模块可以设置成如下几种典型模式

模式	模式描述	DBCTL[POLSEL]		DBCTL[OUT_MODE]	
		S3	S2	S1	S0
1	EPWMxA 和 EPWMxB 直通（无延迟）	×	×	0	0
2	高电平有效互补（AHC）	1	0	1	1
3	低电平有效互补（ALC）	0	1	1	1
4	高电平有效（AH）	0	0	1	1
5	低电平有效（AL）	1	1	1	1
6	EPWMxA Out = 无延迟的 EPWMxA In EPWMxB Out = E 带下降沿延迟的 PWMxA In	0 或 1	0 或 1	0	1
7	EPWMxA Out = 带上升沿延迟的 EPWMxA In EPWMxB Out = 无延迟的 EPWMxB In	0 或 1	0 或 1	1	0

实际应用中，并不只有以上这7种模式可供选择，
可根据实际需要进行自定义设置。


周期
原始的
（outA）
RED
上升沿被延迟
（RED）
FED
上升沿被延迟
（RED）
高电平有效互补
（AHC）
低电平有效互补
（ALC）
高电平有效
（AH）
低电平有效
（AL）

DB子模块掌握重点

28027的DB做得非常灵活，掌握重点：

1）如何配置互补PWM？

DBCTL.IN_MODE

DBCTL.POSEL ,AHC,ALC

DBCTL.OUT_MODE

2）如何配置死区？

DBCTL.OUT_MODE

DBRED，DBFED

9.2.5 事件触发器（ET）子模块

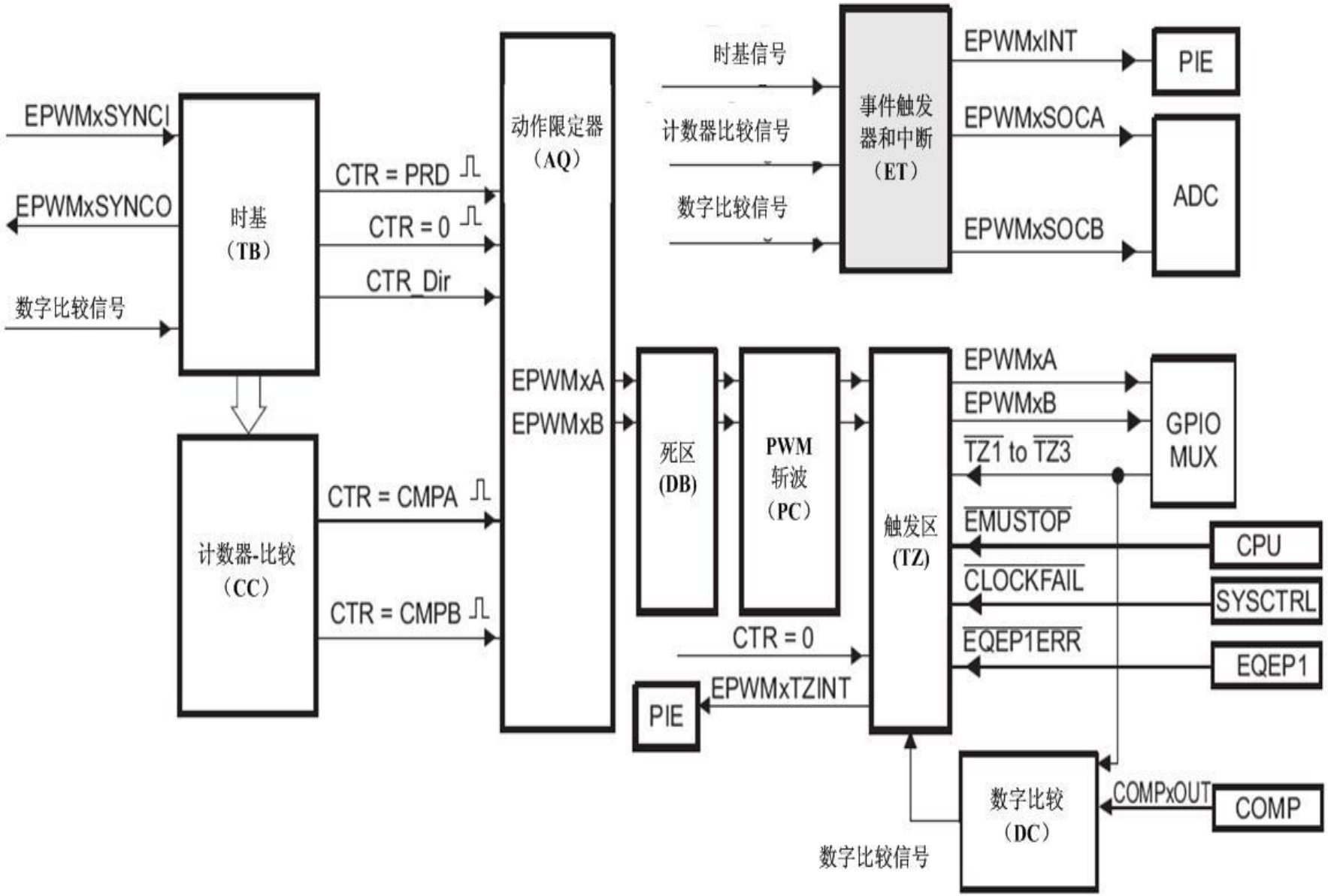
事件触发模块（PWM中断控制）

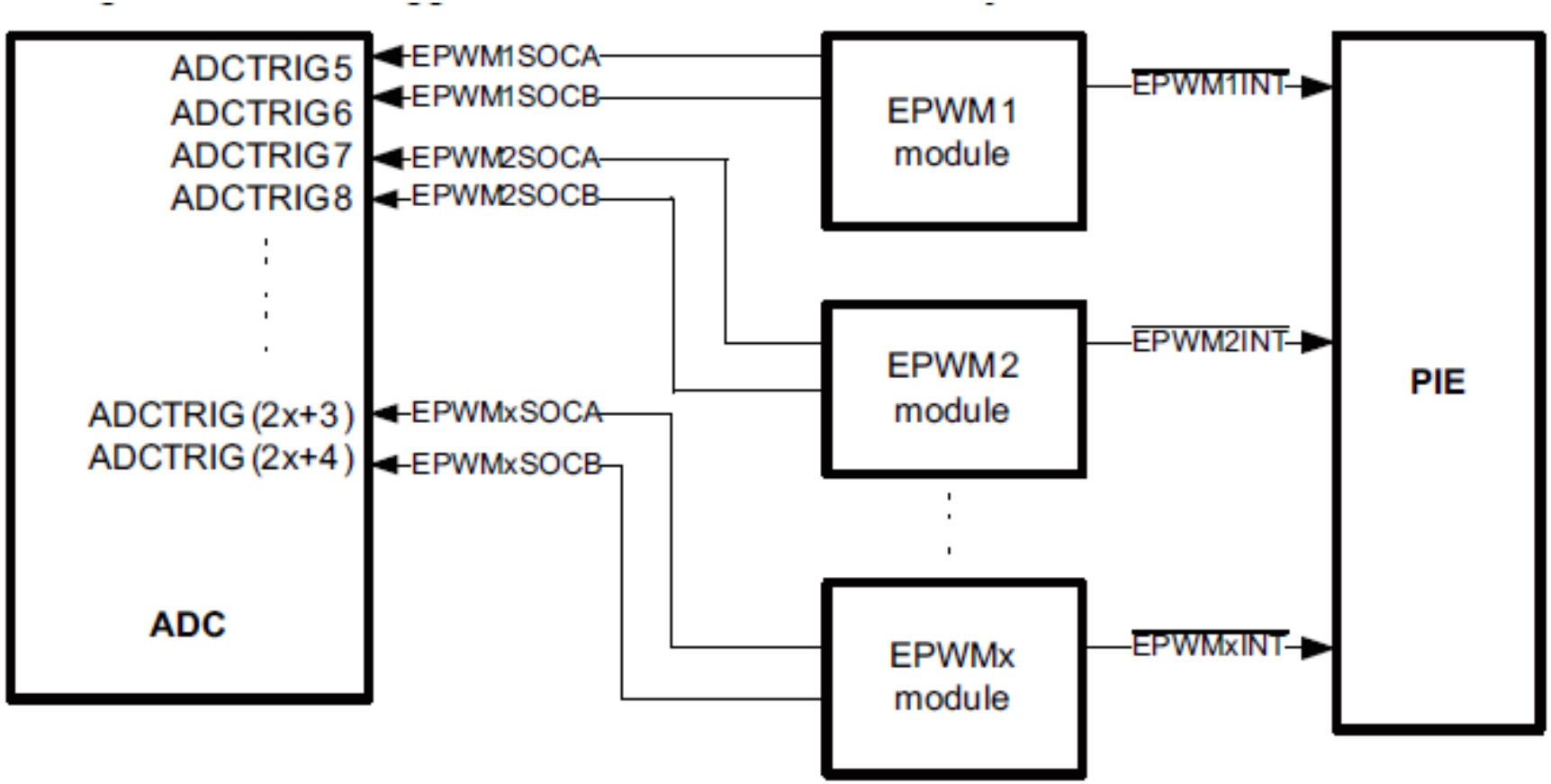
1.接收来自TB模块、CC模块、DC模块事件输入

2.可设置每发生1/2/3个事件，产生一个中断或启动ADC

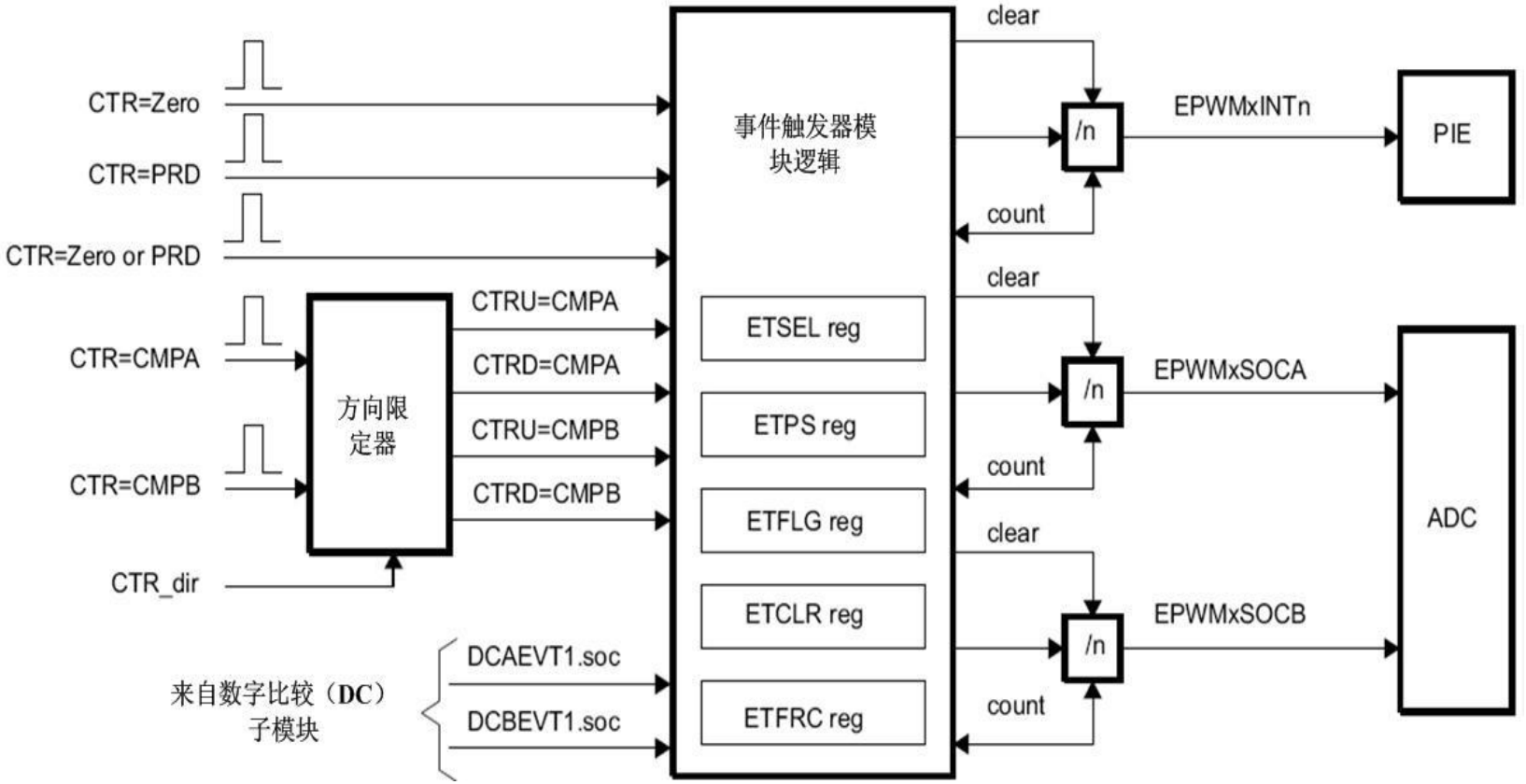
3.全可视的寄存器和寄存器控制

4.允许软件强制触发中断和启动ADC


EPWMxSYNCI
EPWMxSYNCO
数字比较信号
时基
（TB）
CTR = PRD
CTR = 0
CTR_Dir
计数器-比较
（CC）
CTR = CMPA
CTR = CMPB
动作限定器
（AQ）
EPWMxA
EPWMxB
时基信号
计数器比较信号
数字比较信号
事件触发
器和中断
（ET）
EPWMxINT
EPWMxSOCA
EPWMxSOCB
PIE
ADC
死区
(DB)
PWM
斩波
（PC）
触发区
(TZ)
EPWMxA
EPWMxB
TZ1 to TZ3
GPIO
MUX
EMUSTOP
CPU
CLOCKFAIL
SYSCTRL
EQEP1ERR
EQEP1
CTR = 0
EPWMxTZINT
PIE
数字比较
（DC）
COMPxOUT
COMP
数字比较信号


ADCTRIG5
ADCTRIG6
ADCTRIG7
ADCTRIG8
ADCTRIG (2x+3)
ADCTRIG (2x+4)
ADC
EPWM1SOCA
EPWM1SOCB
EPWM2SOCA
EPWM2SOCB
EPWMxSOCA
EPWMxSOCB
EPWM1 module
EPWM2 module
EPWMx module
EPWM1INT
EPWM2INT
EPWMxINT
PIE

事件触发器子模块监控各种事件状态，并且可以配置为在发出中断请求或ADC开始转换之前先预分频（prescale）这些事件。事件触发器其预分频逻辑电路发出中断请求和ADC开始转换的速率是：每一个、二个、三个事件



9.3 ePwm 寄存器介绍

名称	偏移量	大小（×16）	映射	EALLOW	描述
					时基子模块寄存器
TBCTL	0x0000	1	否		时基控制寄存器
TBSTS	0x0001	1	否		时基状态寄存器
TBPHS	0x0003	1	否		时基相位寄存器
TBCTR	0x0004	1	否		时基计数器寄存器
TBPRD	0x0005	1	是		时基周期寄存器
					计数器-比较子模块寄存器
CMPCTL	0x0007	1	否		计数器-比较控制寄存器
CMPA	0x0009	1	是		计数器-比较A寄存器
CMPB	0x000A	1	是		计数器-比较B寄存器
					动作限定器子模块寄存器
AQCTLA	0x000B	1	否		输出A（EPWMxA）动作限定器控制寄存器
AQCTLB	0x000C	1	否		输出B（EPWMxB）动作限定器控制寄存器
AQSFRC	0x000D	1	否		动作限定器软件强制寄存器
AQCSFRC	0x000E	1	是		动作限定器连续S/W强制寄存器组
					死区发生器子模块寄存器
DBCTL	0x000F	1	否		死区发生器控制寄存器
DBRED	0x0010	1	否		死区发生器上升沿延迟计数寄存器
DBFED	0x0011	1	否		死区发生器下降沿延迟计数寄存器
					事件触发器子模块寄存器
ETSEL	0x0019	1			事件触发器选择寄存器
ETPS	0x001A	1			事件触发器预分频寄存器
ETFLG	0x001B	1			事件触发器标志寄存器
ETCLR	0x001C	1			事件触发器清零寄存器
ETFRC	0x001D	1			事件触发器强制寄存器

9.3.1 时基子模块的寄存器

（1）时基周期寄存器（TBPRD）：RW-0

用于设置PWM频率。

该寄存器的映射寄存器通过TBCTL[PRDLD]位使能和禁用。

默认情况下该寄存器的映射寄存器是使能的。

（2）时基相位寄存器（TBPHS）： RW-0

设置所选ePWM相对于时基（提供同步输入信号）的相位。

如果TBCTL[PHSEN] = 0，那么同步事件将被忽略且时基计数器不会装载相位；

如果TBCTL[PHSEN] = 1，那么当同步事件发生时，时基计数器（TBCTR）将会装载相位（TBPHS）。

同步事件可以由同步输入信号（EPWMxSYNCI）发起或通过“软件强制同步”启动

（3） 时基计数器寄存器（TBCTR）： RW-0

读该寄存器可以得到时基计数器的当前值；

写操作可以设置当前时基计数器的值，一但进行写操作后就进行更新；

写操作与时基时钟（TBCLK）不同步，且该寄存器无映射寄存器。

（4）时基控制寄存器（TBCTL）：

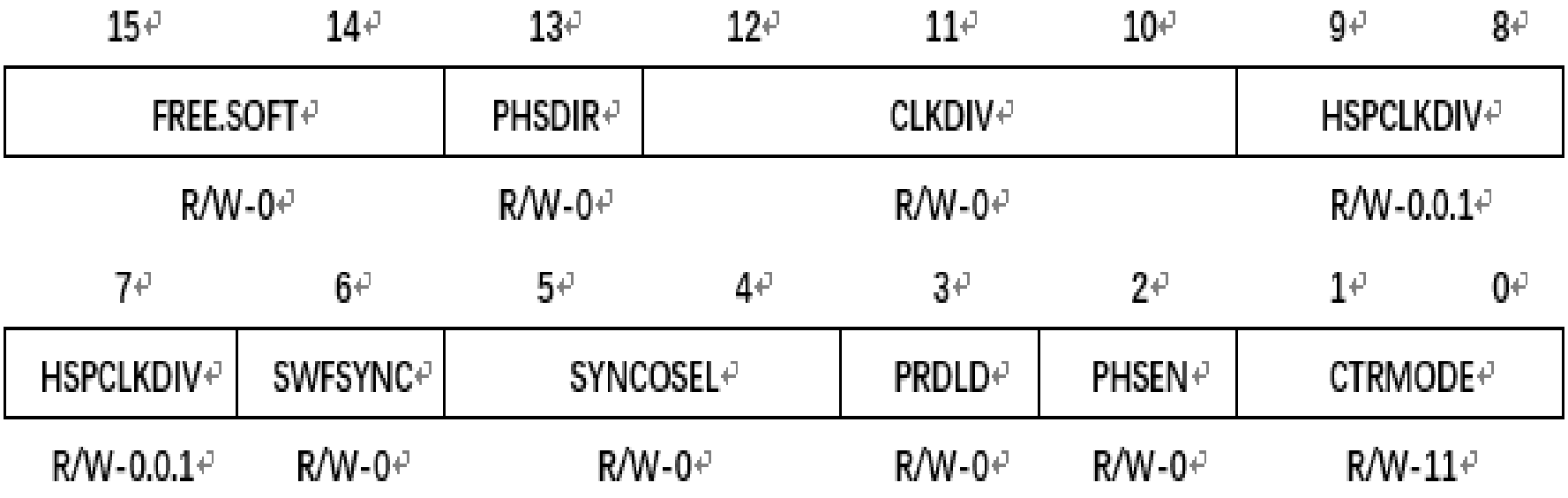


图 9.17　时基控制寄存器（TBCTL）

（4）时基控制寄存器（TBCTL）：

位	名称	值	描述
15:14	FREE, SOFT		仿真模式位。这些位选择 ePWM 时基计数器在仿真期间的行为。
13	PHSDIR		相位方向位 这个位仅在时基计数器被配置为“先递增后递减”计数模式时才使用。 PHSDIR 位用于指示在发生同步事件后时基计数器（TBCTR）采用何种计数方式计数并从相位（TBPHS）寄存器装载新相位值。这与同步事件发生之前计数器的方向无关。在递增和递减计数模式中都不需要这个位。
		0	在同步事件后递减计数
		1	这同步事件后递增计数
12:10	CLKDIV		时基时钟预分频位：这些位决定时基时钟的预分频值 TBCLK = SYSCLKOUT / (HSPCLKDIV× CLKDIV)
		000	/1（复位时的默认值）
		001	/2
		010	/4
		011	/8
		100	/16
		101	/32
		110	/64
		111	/128

（4）时基控制寄存器（TBCTL）：

位	名称	值	描述
6	SWFSYNC		软件强制同步脉冲
		0	写 0 无影响，读取时返回 0
		1	写 1 强制产生一个一次（one-time）同步脉冲
			该事件与 ePWM 模块的 EPWMxSYNCI 输入相或
			SWFSYNC 仅在 EPWMxSYNCI 被 SYNCOSEL = 00 选中时有效（工作）
5:4	SYNCOSEL		同步输出选择。这些位选择 EPWMxSYNCO 信号的源。
		00	EPWMxSYNC：
		01	CTR = 0：时基计数器等于 0（TBCTR = 0x0000）
		10	CTR = CMPB：时基计数器等于计数器比较 B（TBCTR = CMPB）
		11	禁用 EPWMxSYNCO 信号
3	PRDLD		有效周期寄存器是否从映射寄存器装载的选择位
		0	当 TBCTR=0 时，周期寄存器（TBPRD）从映射寄存器那里装载
		1	立即装载 TBPRD 寄存器，无需使用映射寄存器
			对 TBPRD 寄存器执行写或读操作都直接访问有效寄存器。
2	PHSEN		计数器寄存器从相位寄存器装载的使能位
		0	不从时基相位寄存器（TBPHS）那里装载时基计数器（TBCTR）
		1	当 EPWMxSYNCI 输入信号出现或者 SWFSYNC 位强制进行软件同步，抑或发生数字比较同步事件时，时基计数器从相位寄存器那里装载。
1	CTRMODE		计数模式
		00	递增计数模式
		01	递减计数模式
		10	先递增后递减
		11	停止-停顿模式（复位时的默认值）

（5）时基状态寄存器（TBSTS）：

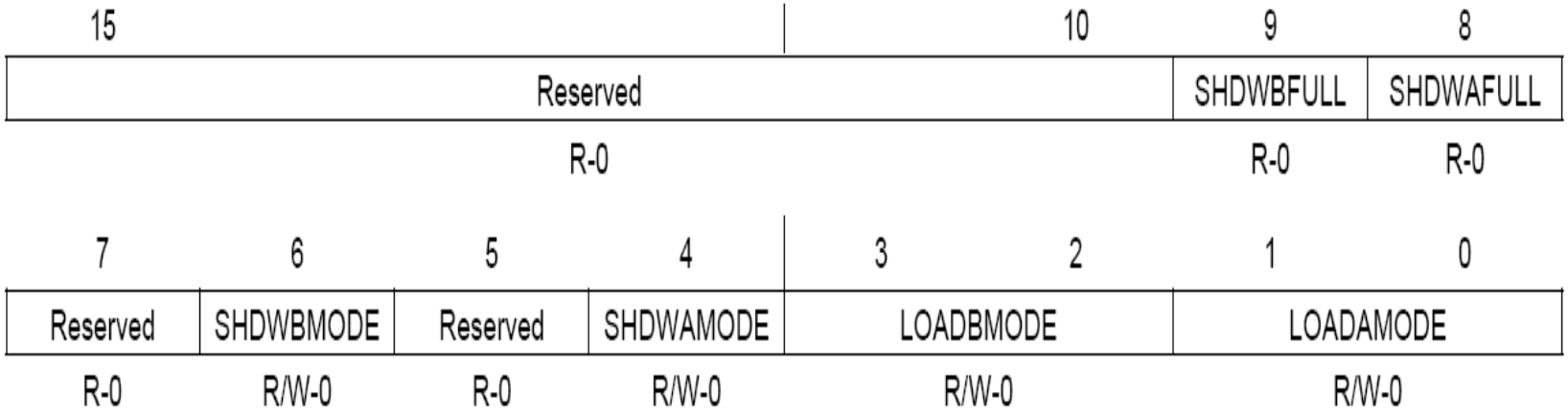
7	3	2	1	0
Reserved		CTRMAX	SYNCI	CTRDIR
R-0		R/W1C-0	R/W1C-0	R-1

图 99.18 时基状态寄存器（TBSTS）

表 9.7 时基状态寄存器（TBSTS）字段描述

名称	值	描述
Reserved		保留
CTRMAX		时基计数器最大值锁存状态位
	0	为 0 时，表示时基计数器从没到达过最大值。向该位写 0 没有作用。
	1	为 1 时，表示时基计数器到达最大值 0xFFFF。向该位写 1 会清除已被锁存事件
SYNCI		输入同步锁存状态位
	0	向该位写 0 没有作用。当为 0 时，表示没有发生外部同步事件。
	1	当为 1 时，表示发生了一个外部同步事件（EPWMxSYNCI）。向该位写 1 会清除已被锁存事件
CTRDIR		时基计数器方向状态位。复位时，该计数器停止工作；因此，这个位没有意义。
		要该位有意义，必须通过 TBCTL[CTRMODE]设置正确的计数模式。
	0	时基计数器当前正在递减
	1	时基计数器当前正在递增

（3）计数器-比较控制寄存器（CMPCTL）：



15						10	9	8
Reserved							SHDWBFULL	SHDWAFULL
R-0							R-0	R-0

7	6	5	4	3	2	1	0
Reserved	SHDWBMODE	Reserved	SHDWAMODE	LOADBMODE		LOADAMODE	
R-0	R/W-0	R-0	R/W-0	R/W-0		R/W-0	

（3）计数器-比较控制寄存器（CMPCTL）：

位	名称	值	描述
15-10	Reserved		保留
9	SHDWBFULL		计数器-比较 B（CMPB）映射寄存器满状态标志
			一旦发生装载选通，这个位就会自己清零。
		0	CMPB 映射 FIFO 还未满
		1	表示 CMPB 映射 FIFO 满了；CPU 写操作会覆盖当前映射值
8	SHDWAFULL		计数器-比较 A（CMPA）映射寄存器满状态标志
			该标志在向 CMPA:CMPAHR 寄存器执行 32 位写操作时置位：或在向
			CMPA 寄存器执行 16 位写操作时置位。向 CMPAHR 寄存器执行 16
			位写操作不会影响这个标志。
			一旦发生装载选通，这个位就会自己清零。
		0	CMPA 映射 FIFO 还未满
		1	表示 CMPA 映射 FIFO 满了；CPU 写操作会覆盖当前映射值
7	Reserved		保留
6	SHDWBMODE		计数器-比较 B（CMPB）寄存器工作模式
		0	映射模式。像一个双缓冲器一样工作。所有通过 CPU 的写操作都可
			以访问映射寄存器。
		1	立即模式。只使用映射比较 B 寄存器。为执行立即比较动作，所有写
			操作和读操作都直接访问映射寄存器。

（3）计数器-比较控制寄存器（CMPCTL）：

4	SHDWAMODE		计数器-比较 A（CMPA）寄存器工作模式
		0	映射模式。像一个双缓冲器一样工作。所有通过 CPU 的写操作都可以访问映射寄存器。
		1	立即模式。只使用有效比较 A 寄存器。为执行立即比较动作，所有写操作和读操作都直接访问有效寄存器。
3-2	LOADBMODE		有效计数器-比较 B（CMPB）从映射装载的模式选择位
			这个位不影响立即模式（CMPCTL[SHDWBMODE] = 1）
		00	在 CTR = 0 时装载：时基计数器等于 0（TBCTR = 0x0000）
		01	在 CTR = PRD 时装载：时基计数器等于周期（TBCTR = TBPRD）在
		10	CTR = 0 或 CTR = PRD 时装载
		11	停顿（不装载）
1-0:	LOADAMODE		Active 计数器-比较 A（CMPA）从映射装载的模式选择位
			这个位不影响立即模式（CMPCTL[SHDWAMODE] = 1）
		00	在 CTR = 0 时装载：时基计数器等于 0（TBCTR = 0x0000）
		01	在 CTR = PRD 时装载：时基计数器等于周期（TBCTR = TBPRD）在
		10	CTR = 0 或 CTR = PRD 时装载
		11	停顿（不装载）

9.3.3 动作限定器子模块的寄存器

（1）动作限定器输出A控制寄存器（AQCTLA）

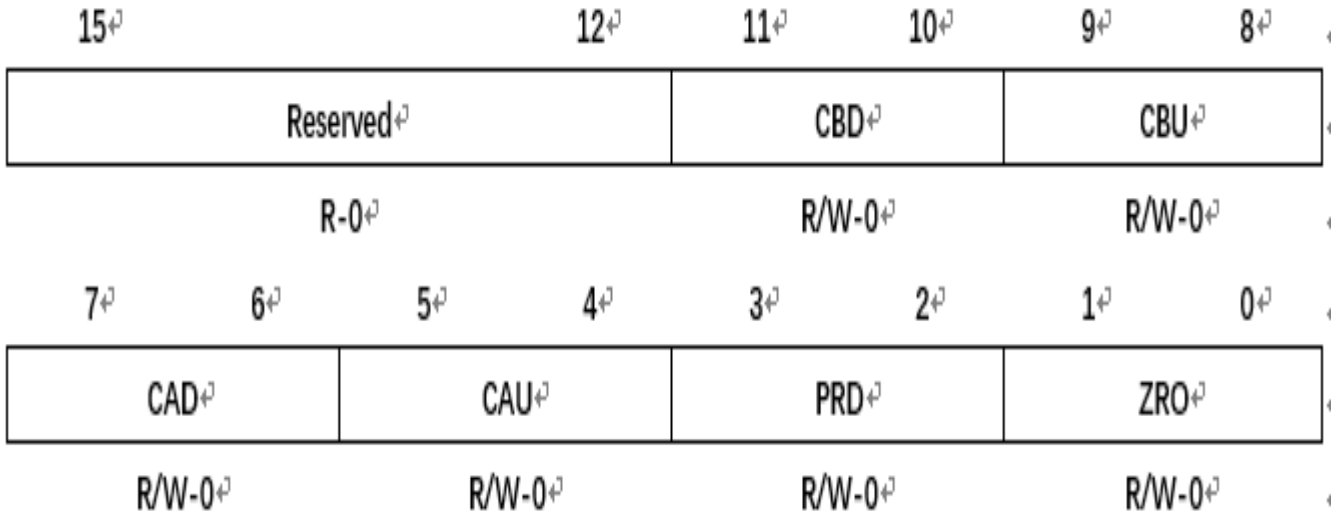


图 9.20　动作限定器输出 A 控制寄存器（AQCTLA）

（1）动作限定器输出A控制寄存器（AQCTLA）

表 9.9　动作限定器输出 A 控制寄存器（AQCTLA）字段描述

位	名称	值	描述
15-12	Reserved		保留
11-10	CBD		当时基计数器等于有效 CMPB 寄存器且计数器在递减时的动作。
		00	什么也不做（不采取动作）
		01	清零：将 EPWMxA 输出强制变低
		10	置位：将 EPWMxA 输出强制变高
		11	切换：将 EPWMxA 输出从低强制变为高，从高强制变为低
9-8	CBU		当时基计数器等于有效 CMPB 寄存器且计数器在递增时的动作。
		00	什么也不做（不采取动作）
		01	清零：将 EPWMxA 输出强制变低
		10	置位：将 EPWMxA 输出强制变高
		11	切换：将 EPWMxA 输出从低强制变为高，从高强制变为低
7-6	CAD		当时基计数器等于有效 CMPA 寄存器且计数器在递减时的动作。
		00	什么也不做（不采取动作）
		01	清零：将 EPWMxA 输出强制变低
		10	置位：将 EPWMxA 输出强制变高
		11	切换：将 EPWMxA 输出从低强制变为高，从高强制变为低

（1）动作限定器输出A控制寄存器（AQCTLA）

5-4	CAU		当时基计数器等于有效 CMPA 寄存器且计数器在递增时的动作。
		00	什么也不做（不采取动作）
		01	清零：将 EPWMxA 输出强制变低
		10	置位：将 EPWMxA 输出强制变高
		11	切换：将 EPWMxA 输出从低强制变为高，从高强制变为低
3-2	PRD		当计数器等于周期时的动作。
			注：通过定义，在“先递增后递减”计数模式中，当计数器等于周期时，计数方向被定义成 0 或递减。
		00	什么也不做（不采取动作）
		01	清零：将 EPWMxA 输出强制变低
		10	置位：将 EPWMxA 输出强制变高
		11	切换：将 EPWMxA 输出从低强制变为高，从高强制变为低
1-0	ZRO		当计数器等于 0 时的动作。
			注：通过定义，在“先递减后递增”计数模式中，当计数器等于 0 时，计数方向被定义成 1 或递增。
		00	什么也不做（不采取动作）
		01	清零：将 EPWMxA 输出强制变低
		10	置位：将 EPWMxA 输出强制变高
		11	切换：将 EPWMxA 输出从低强制变为高，从高强制变为低

9.3.3 动作限定器子模块的寄存器

（2）动作限定器输出B控制寄存器（AQCTLB）

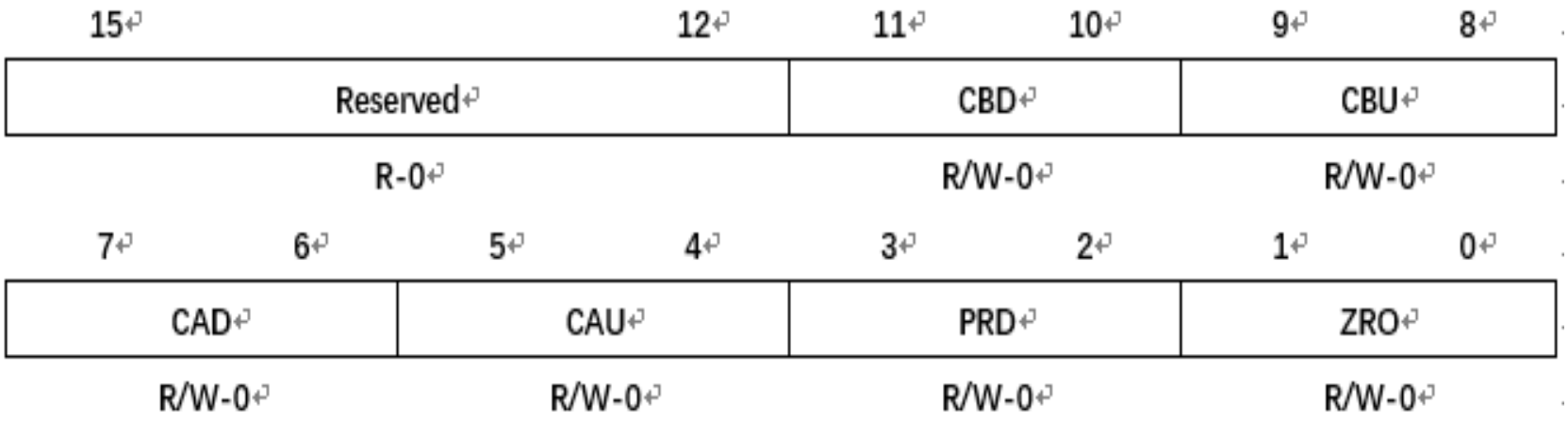


图 9.21 动作限定器输出 B 控制寄存器（AQCTLB）

位定义和AQCTLA一样，只是动作的引脚为EPWMxB

（3）动作限定器软件强制寄存器（AQSFRC）和动作限定器连续软件强制寄存器（AQCSFRC）：

9.3.4 死区子模块的寄存器

（1）死区发生器控制寄存器（DBCTL）：

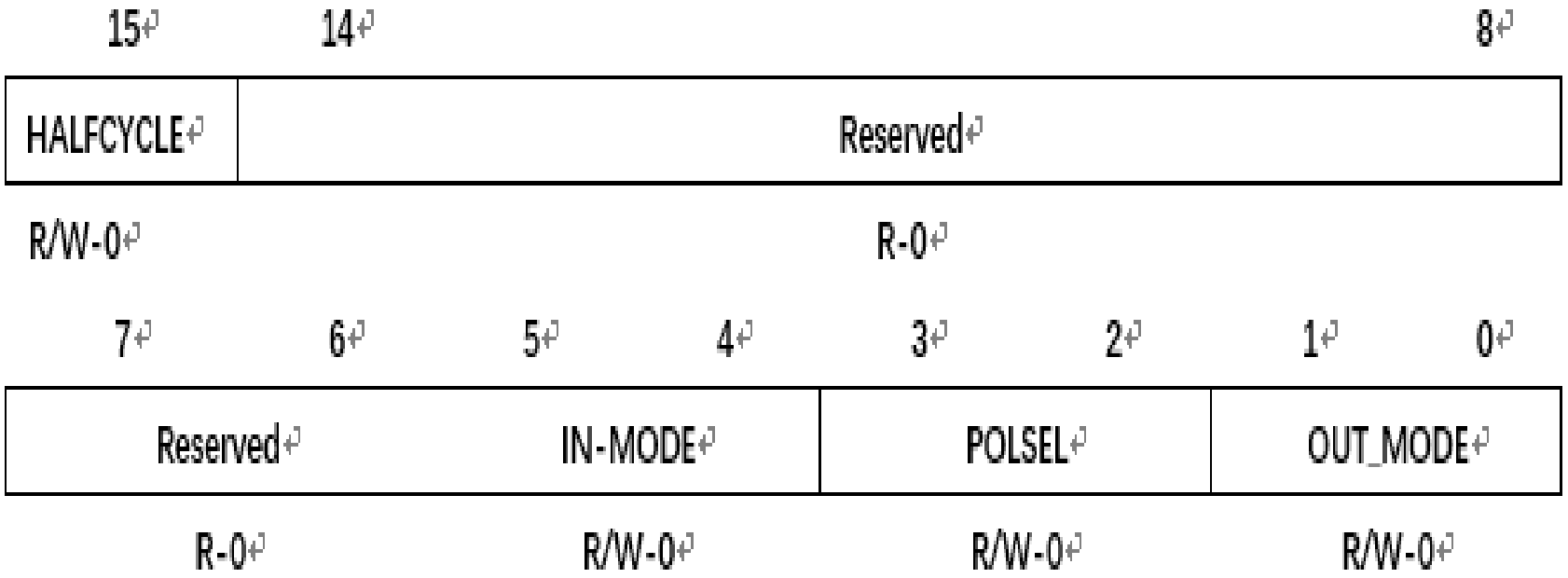


图 9.24 死区发生器控制寄存器（DBCTL）

（1）死区发生器控制寄存器（DBCTL）：

表 9.12 死区发生器控制寄存器（DBCTL）字段描述

位	名称	值	描述
15	HALFCYCLE		半周期计时使能位：
		0	全周期计时使能。死区计数器以 TBCLK 速率计时
		1	半周期计时使能。死区计数器以 TBCLK*2 速率计时
14-6	Reserved		保留
5-4	IN_MODE		死区输入模式控制
			位 5 控制 S5 开关，位 4 控制 S4 开关，如图 9.11 所示。
			这个位可用于选择下降沿延迟和上升沿延迟的输入源
			要产生传统的死区波形，默认情况下，下降沿延迟和上升沿延迟的源都必须是 EPWMxA In
		00	EPWMxA In（来自动作限定器）是下降沿延迟和上升沿延迟的源
		01	EPWMxB In（来自动作限定器）是上升沿延迟信号的源
			EPWMxA In（来自动作限定器）是下降沿延迟信号的源
		10	EPWMxA In（来自动作限定器）是上升沿延迟信号的源
			EPWMxB In（来自动作限定器）是下降沿延迟信号的源
		11	EPWMxB In（来自动作限定器）是上升沿延迟和下降沿延迟信号的源

（1）死区发生器控制寄存器（DBCTL）：

3-2	POLSEL		极性选择控制 位 3 控制 S3 开关，位 2 控制 S2 开关，如图 9.11 所示。 这个位可用于选择在延迟信号发送到死区子模块之前将它反相 以下描述与数字电机控制变极器中一条臂（1eg）的传统上端/下端开关控制对应。？ 这些位假定 DBCTL[OUT_MODE] = 1,1 且 DBCTL[IN_MODE] = 0,0。其它增强模式也可行，但不是典型模式。
		00	高电平有效（AH）模式。EPWMxA 和 EPWMxB 都不反相（缺省）
		01	低电平有效互补（ALC）模式。EPWMxA 反相
		10	高电平有效互补（AHC）模式。EPWMxB 反相
		11	低电平有效（AL）模式。EPWMxA 和 EPWMxB 都反相
1-0	OUT_MODE		死区输出模式控制 位 1 控制 S1 开关，位 0 控制 S0 开关，如图 9.11 所示。 这个位可用于为下降沿延迟和上升沿延迟选择使能或旁路死区发生单元
		00	两个输出信号的死区发生单元都被旁路。在该模式下，来自动作限定器的 EPWMxA 和 EPWMxB 输出信号都被直接传输到 PWM 斩波子模块。 在该模式下，POLSEL 和 IN_MODE 位没什么作用。
		01	禁止上升沿延迟。来自动作限定器的 EPWMxA 信号直接传输到 PWM 斩波子模块的 EPWMxA 输入。 下降沿延迟信号在 EPWMxB 输出上出现。该延迟的输入信号由 DBCTL[IN]MODE]决定。
		10	上升沿延迟信号在 EPWMxA 输出上出现。该延迟的输入信号由 DBCTL[IN]MODE]决定。 禁止下降沿延迟。来自动作限定器的 EPWMxB 信号直接传输到 PWM 斩波子模块的 EPWMxB 输入。
		11	针对 EPWMxA 输出上的上升沿延迟以及 EPWMxB 输出上的下降沿延迟，死区完全使能。延迟的输入信号由 DBCTL[IN_MODE]决定。

（2）死区发生器上升沿延迟寄存器（DBRED）：

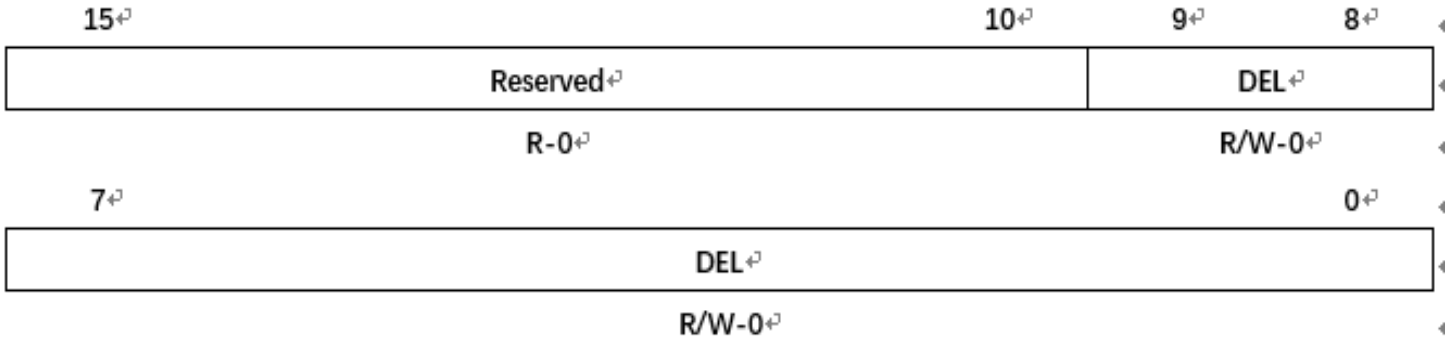


图 9.25　死区发生器上升沿延迟寄存器（DBRED）

（3）死区发生器下降沿延迟寄存器（DBFED）：

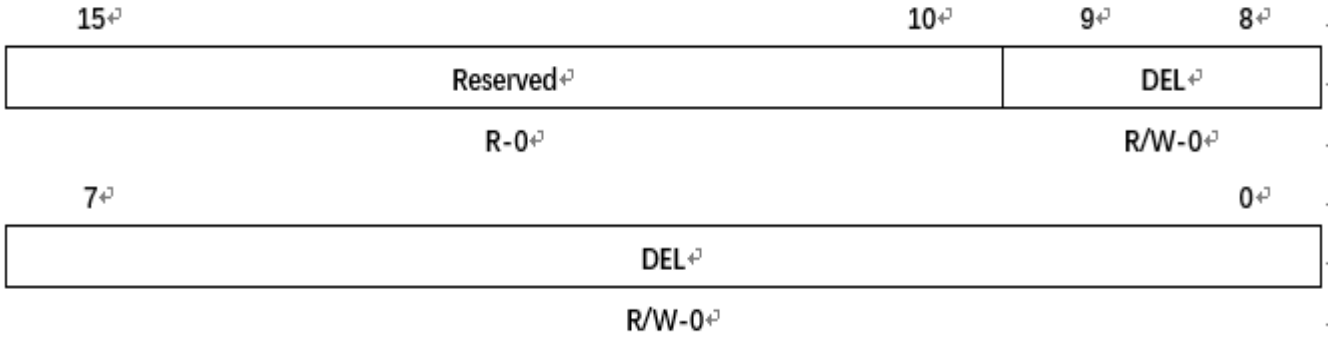


图 9.26　死区发生器下降沿延迟寄存器（DBFED）

9.3.5事件触发器子模块寄存器

（1）事件触发器选择寄存器（ETSEL）：

15	14	12	11	10	8
SOCBEN	SOCBSEL		SOCAEN	SOCASEL	
R/W-0	R/W-0		R/W-0	R/W-0	

7	4	3	2	0
	Reserved	INTEN	INTSEL	
	R-0	R/W-0	R/W-0	

图 9.27　事件触发器选择寄存器（ETSEL）

（1）事件触发器选择寄存器（ETSEL）：

表 9.13　事件触发器选择寄存器（ETSEL）字段描述

位	名称	值	描述
15	SOCB		使能 ADC 开始转换 B（EPWMxSOCB）脉冲
		0	禁用 EPWMxSOCB
		1	使能 EPWMxSOCB
14-12	SOCBSEL		EPWMxSOCB 选项
			这些位决定何时产生 EPWMxSOCB 脉冲。
		000	使能 DCBEVT1.soc 事件
		001	在时基计数器等于 0（TBCTR = 0x0000）时使能事件
		010	在时基计数器等于周期（TBCTR = TBPRD）时使能事件
		011	在时基计数器等于 0 或周期（TBCTR = 0x0000 或 TBCTR = TBPRD）
			时使能事件。这种模式在“先递增后递减”计数模式中非常有用
		100	在定时器正在递增时，当时基计数器等于 CMPA 时使能事件
		101	在定时器正在递减时，当时基计数器等于 CMPA 时使能事件
		110	在定时器正在递增时，当时基计数器等于 CMPB 时使能事件
		111	在定时器正在递减时，当时基计数器等于 CMPB 时使能事件

（1）事件触发器选择寄存器（ETSEL）：

11	SOCAEN		使能 ADC 开始转换 A（EPWMxSOCA）脉冲
		0	禁用 EPWMxSOCA
		1	使能 EPWMxSOCA
10-8	SOCASEL		EPWMxSOCA 选项
			这些位决定何时产生 EPWMxSOCA 脉冲。
		000	使能 DCAEVT1.soc 事件
		001	在时基计数器等于 0（TBCTR = 0x0000）时使能事件
		010	在时基计数器等于周期（TBCTR = TBPRD）时使能事件
		011	在时基计数器等于 0 或周期（TBCTR = 0x0000 或 TBCTR = TBPRD）
			时使能事件。这种模式在“先递增后递减”计数模式中非常有用
		100	在定时器正在递增时，当时基计数器等于 CMPA 时使能事件
		101	在定时器正在递减时，当时基计数器等于 CMPA 时使能事件
		110	在定时器正在递增时，当时基计数器等于 CMPB 时使能事件
		111	在定时器正在递减时，当时基计数器等于 CMPB 时使能事件

（1）事件触发器选择寄存器（ETSEL）：

7-4	Reserved		保留
3	INTEN		ePWM 中断（EPWMx_INT）使能位
		0	禁止产生 EPWMx_INT 中断
		1	允许产生 EPWMx_INT 中断
2-0	INTSEL		ePWM 中断（EPWMx_INT）选项
		000	保留
		001	在时基计数器等于 0（TBCTR = 0x0000）时使能事件
		010	在时基计数器等于周期（TBCTR = TBPRD）时使能事件
		011	在时基计数器等于 0 或周期（TBCTR = 0x0000 或 TBCTR = TBPRD）
			时使能事件。这种模式在“先递增后递减”计数模式中非常有用
		100	在定时器正在递增时，当时基计数器等于 CMPA 时使能事件
		101	在定时器正在递减时，当时基计数器等于 CMPA 时使能事件
		110	在定时器正在递增时，当时基计数器等于 CMPB 时使能事件
		111	在定时器正在递减时，当时基计数器等于 CMPB 时使能事件

（2）事件触发器预分频寄存器（ETPS）：

图 9.28 事件触发器预分频寄存器（ETPS）

SOCBCNT为ePWM ADC开始转换事件EPWMxSOCB计数值（发生次数）：
0=没有发生事件，1~3表示已经发生了1~3个事件

SOCACNT为ePWM ADC开始转换事件EPWMxSOCA计数值（发生次数）：
0=没有发生事件，1~3表示已经发生了1~3个事件

INTCNT为ePWM中断事件（EPWMx_INT）计数值（发生次数）：
0=没有发生事件，1~3表示已经发生了1~3个事件

（2）事件触发器预分频寄存器（ETPS）：

SOCBPRD为ePWM ADC开始转换B事件周期选择位：

00=禁用SOCB事件计数器。没有产生EPWMxSOCB脉冲
01=在第一个事件时产生EPWMxSOCB脉冲
10=在第二个事件时产生EPWMxSOCB脉冲
11=在第三个事件时产生EPWMxSOCB脉冲

SOCAPRD为ePWM ADC开始转换B事件周期选择位：

00=禁用SOCA事件计数器。没有产生EPWMxSOCA脉冲
01=在第一个事件时产生EPWMxSOCA脉冲
10=在第二个事件时产生EPWMxSOCA脉冲
11=在第三个事件时产生EPWMxSOCA脉冲

INTPRD为ePWM中断（EPWMx_INT）周期选择：

00=禁用中断事件计数器。没有产生中断且ETFRC[INT]被忽略
01=在第一个事件时产生中断
10=在第二个事件时产生中断
11=在第三个事件时产生中断

（3）事件触发器标志寄存器（ETFLG）：

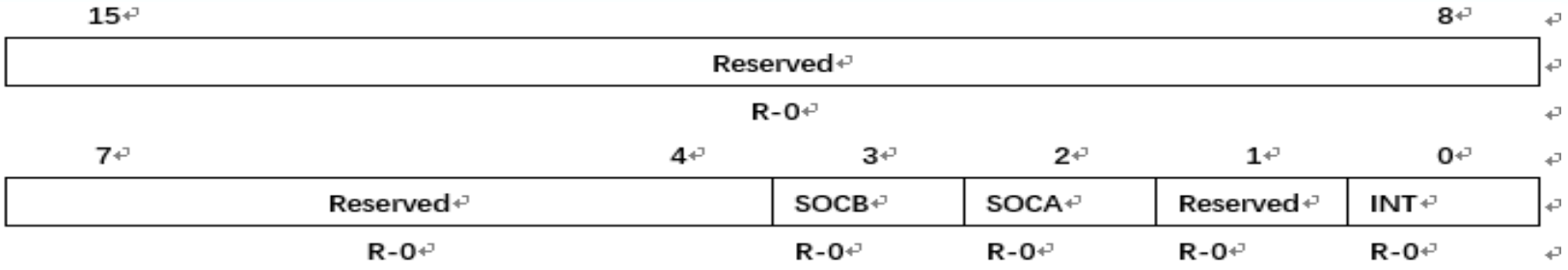


图 9.29 事件触发器标志寄存器（ETFLG）

表 9.15 事件触发器标志寄存器（ETFLG）字段描述

位	名称	值	描述
15-4	Reserved		保留
3	SOCB		被锁存的 ePWM ADC 开始转换 B（EPWMxSOCB）的状态标志
		0	表示没有发生 EPWMxSOCB 事件
		1	表示在 EPWMxSOCB 上产生了一个开始转换脉冲。即使该标志位置位，也将继续产生 EPWMxSOCB 输出
2	SOCA		被锁存的 ePWM ADC 开始转换 A（EPWMxSOCB）的状态标志
		0	表示没有发生 EPWMxSOCA 事件
		1	表示在 EPWMxSOCA 上产生了一个开始转换脉冲。即使该标志位置位，也将继续产生 EPWMxSOCA 输出
1	Reserved		保留
0	INT		被锁存的 ePWM 中断（EPWMx_INT）的状态标志
		0	表示没有事件发生
		1	表示产生了一个 ePWMx 中断（EPWMx_INT）。在该标志位清零之前不会再产生中断。在 ETFLG[INT]位仍然置位的同时最多可以挂起一个中断。如果一个中断正在挂起，那么它必须到 ETFLG[INT]位清零后才能产生。

（4）事件触发器清零寄存器（ETCLR）：

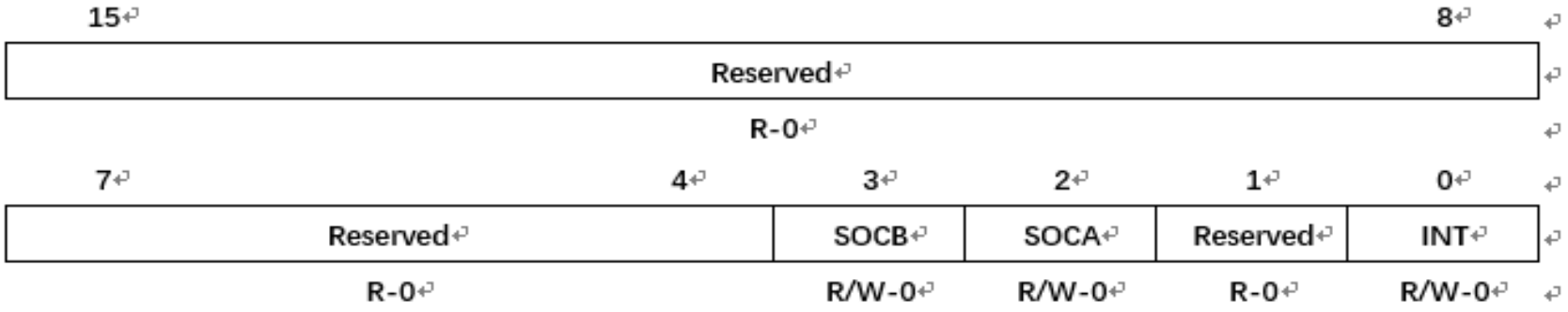


图 9.30 事件触发器清零寄存器（ETCLR）

表 9.16 事件触发器清零寄存器（ETCLR）字段描述

位	名称	值	描述
15-4	Reserved		保留
3	SOCB		ePWM ADC 开始转换 B（EPWMxSOCB）标志清零位
		0	写 0 没有影响。读这个位总是返回 0
		1	清零 ETFLG[SOCB]标志位
2	SOCA		ePWM ADC 开始转换 A（EPWMxSOCA）标志清零位
		0	写 0 没有影响。读这个位总是返回 0
		1	清零 ETFLG[SOCA]标志位
1	Reserved		保留
0	INT		ePWM 中断（EPWMx_INT）标志清零位
		0	写 0 没有影响。读这个位总是返回 0
		1	清零 ETFLG[INT]标志位并允许再产生其它中断脉冲

9.3.6 ePWM寄存器结构体说明

头文件中包括一个寄存器结构体EPWM_REGS，该寄存器结构体中的位定义与前面介绍的寄存器位定义相同。

```
struct EPWM_REGS {
   union  TBCTL_REG      TBCTL;       //时基控制寄存器
   union  TBSTS_REG       TBSTS;       //时基状态寄存器
   union  TBPHS_HRPWM_GROUP  TBPHS;  // TBPHS:TBPHSHR联合
   Uint16    TBCTR;       //时基计数器
   Uint16    TBPRD;       //时基周期寄存器
   Uint16    TBPRDHR;    //时基周期高位寄存器组
   union  CMPCTL_REG     CMPCTL;      //比较控制
   union  CMPA_HRPWM_GROUP  CMPA;   // CMPA:CMPAHR联合
   Uint16    CMPB;                    // CMPB寄存器
    ……
    };

  volatile struct EPWM_REGS EPwm1Regs;
  volatile struct EPWM_REGS EPwm2Regs;
  volatile struct EPWM_REGS EPwm3Regs;
  volatile struct EPWM_REGS EPwm4Regs;
```

9.4 ePWM应用举例

9.4.1 ePWM用于三相电机驱动控制

- 三相电机驱动控制中需要用到三相互补且带死区的PWM。
- 在图4.7中PWM1A/B、PWM2A/B和PWM4A/B可用于三相电机驱动控制。PWM的频率为10kHz。

设置PWM及中断流程

1.关全局中断（INTM）
2.关Pwm中断
3.TBCLKSYNC=0
4.设置Pwm相关寄存器
5.清除中断标志位，开PWM中断
6. TBCLKSYNC=1
7.开全局中断（INTM）

PWM4A/B初始化程序示例

```
void InitPWM4()
{
  int PWMPRD,DeadTime;
  EALLOW;
  SysCtrlRegs.PCLKCR0.bit.TBCLKSYNC = 0;
  EDIS;
  PWMPRD = 3000; // 10kHz
  DeadTime = 120; //
  EPwm4Regs.TBPRD = PWMPRD;
  EPwm4Regs.TBPHS.half.TBPHS = 0; //设置相位寄存器为0
  EPwm4Regs.TBCTL.bit.CLKDIV = 0; //CLKDIV = 0;
  EPwm4Regs.TBCTL.bit.HSPCLKDIV = 0; //HSPCLKDIV = 0;
  EPwm4Regs.TBCTL.bit.CTRMODE = 2; //对称模式
  EPwm4Regs.TBCTL.bit.PHSEN =0; //主模式
  EPwm4Regs.TBCTL.bit.PRDLD = 0;
  EPwm4Regs.TBCTL.bit.SYNCOSEL = TB_CTR_ZERO;
  EPwm4Regs.CMPCTL.bit.SHDWAMODE = 0;
  EPwm4Regs.CMPCTL.bit.SHDWBMODE = 0;
```

PWM4A/B初始化程序示例

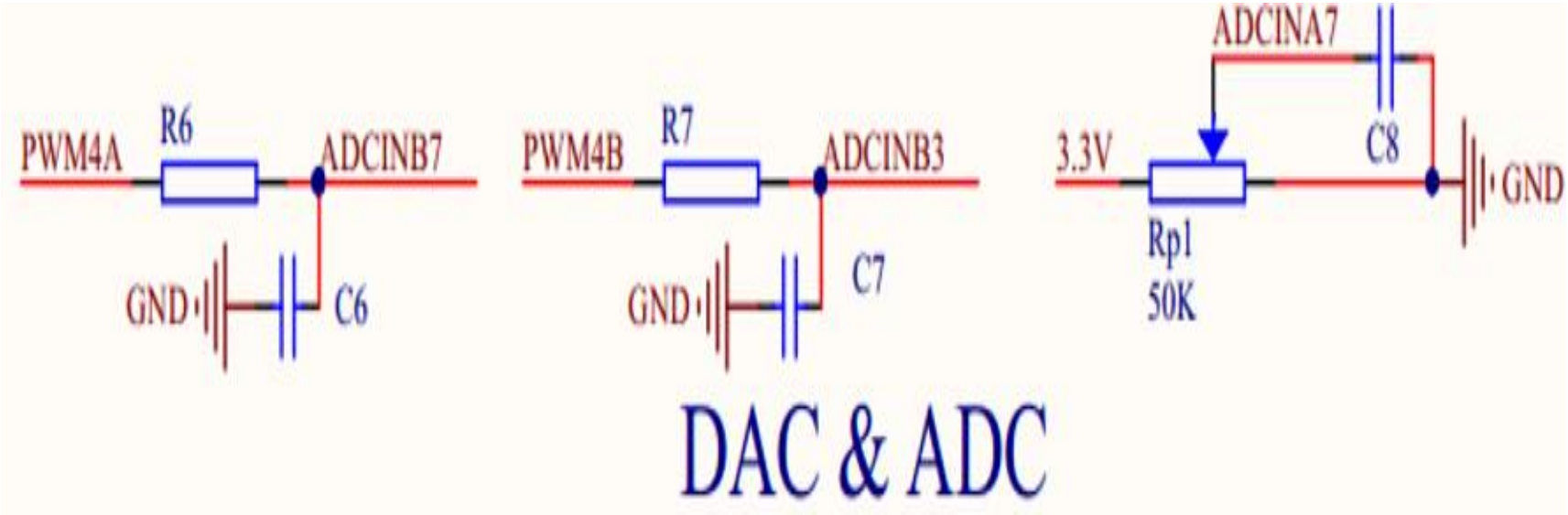
```
EPwm4Regs.CMPCTL.bit.LOADAMODE = 2; //在CTR=0或者CTR=PRD的时候装载
EPwm4Regs.CMPCTL.bit.LOADBMODE = 2; //同上
EPwm4Regs.AQCTLA.bit.CAU = 1; // EPWM4A在CAU时置1
EPwm4Regs.AQCTLA.bit.CAD = 2;
EPwm4Regs.AQCTLA.bit.CBU = 0;
EPwm4Regs.AQCTLA.bit.CBD = 0;
EPwm4Regs.DBCTL.bit.OUT_MODE = 3; // 使能死区
EPwm4Regs.DBCTL.bit.POLSEL =  2; // 互补高有效
EPwm4Regs.DBFED = DeadTime;
EPwm4Regs.DBRED = DeadTime;
EPwm4Regs.CMPA.half.CMPA=PWMPRD/2;
EPwm4Regs.CMPB = PWMPRD/2;
// Enable CNT_zero interrupt using EPWM4 Time-base
 EPwm4Regs.ETSEL.bit.INTEN = 1;   //使能EPWM4中断
 EPwm4Regs.ETSEL.bit.INTSEL = 1;  //CNT_zero时中断
 EPwm4Regs.ETPS.bit.INTPRD = 1;   //在第一个事件处触发中断
 EPwm4Regs.ETCLR.bit.INT = 1;     //清中断标志
EALLOW;
SysCtrlRegs.PCLKCR0.bit.TBCLKSYNC = 1;  //同步
EDIS;
 }
```

9.3 ePwm 寄存器介绍

名称	偏移量	大小（×16）	映射	EALLOW	描述
					时基子模块寄存器
TBCTL	0x0000	1	否		时基控制寄存器
TBSTS	0x0001	1	否		时基状态寄存器
TBPHS	0x0003	1	否		时基相位寄存器
TBCTR	0x0004	1	否		时基计数器寄存器
TBPRD	0x0005	1	是		时基周期寄存器
					计数器-比较子模块寄存器
CMPCTL	0x0007	1	否		计数器-比较控制寄存器
CMPA	0x0009	1	是		计数器-比较A寄存器
CMPB	0x000A	1	是		计数器-比较B寄存器
					动作限定器子模块寄存器
AQCTLA	0x000B	1	否		输出A（EPWMxA）动作限定器控制寄存器
AQCTLB	0x000C	1	否		输出B（EPWMxB）动作限定器控制寄存器
AQSFRC	0x000D	1	否		动作限定器软件强制寄存器
AQCSFRC	0x000E	1	是		动作限定器连续S/W强制寄存器组
					死区发生器子模块寄存器
DBCTL	0x000F	1	否		死区发生器控制寄存器
DBRED	0x0010	1	否		死区发生器上升沿延迟计数寄存器
DBFED	0x0011	1	否		死区发生器下降沿延迟计数寄存器
					事件触发器子模块寄存器
ETSEL	0x0019	1			事件触发器选择寄存器
ETPS	0x001A	1			事件触发器预分频寄存器
ETFLG	0x001B	1			事件触发器标志寄存器
ETCLR	0x001C	1			事件触发器清零寄存器
ETFRC	0x001D	1			事件触发器强制寄存器

9.4.2 ePWM用于DAC

在现在很多场合把PWM用于DAC，高频PWM信号经低通滤波后，其占空比就是模拟输出电压。



9.4.2 ePWM用于DAC

举例：PWM4A输出一个10Hz直流偏置为1.65V、峰值为3.3V的三角波（也就是ADCINB7为直流偏置为1.65V、峰值为3.3V的三角波）

设计说明：

1.PWM4A占空比为0时ADCINB7=0V，
PWM4A占空比为50%时ADCINB7=1.65V，
PWM4A占空比为100%时ADCINB7=3.3V。

2. 因此要实现DAC输出，
需要0.05S内占空比从0增加到100%，
再在0.05S内占空比从100%降到0。

9.4.2 ePWM用于DAC

3. PWM4可在上面的设置基础上作出一些修改（禁止死区或者死区时间为0）即可。
PWM4A的频率为10kHz，每次CNT=0时中断，意味着：500次PWM中断占空比从0变化到100%（CMPA从0变化到周期值3000），
再500次PWM中断占空比从100%变化到0（CMPA从3000变化到0）。

4.CMPA的变化在PWM4中断服务程序里完成即可，在下面PWM4中断服务程序中PWMTimes为全局变量，用于记录进入PWM4中断的次数。

9.4.2 ePWM用于DAC

```
interrupt void Epwm4Int_isr(void)
{
        PWMTimes++;
        if(PWMTimes >999)PWMTimes=0;
        if(PWMTimes >500)
           EPwm4Regs.CMPA.half.CMPA=(1000- PWMTimes)*6;
        else EPwm4Regs.CMPA.half.CMPA=PWMTimes * 6;

        EPwm4Regs.ETCLR.bit.INT = 1;
        PieCtrlRegs.PIEACK.all = PIEACK_GROUP3;
        EINT;
    }
```

•PWM用作DAC需要说明的：

1）这种DAC的分辨率取决于PWM的周期，PWM周期越长，分辨率越高。

2）这种DAC输出的模拟信号的频率受制于低通滤波器的特性、PWM的频率和模拟信号的分辨率。

这种DAC往往用在低成本、低频场合。