

PWM 用户手册

目录

1 SPWM.....	4
1.1 操作步骤.....	4
1.2 寄存器介绍.....	5
2 SVPWM.....	10
2.1 特性.....	10
2.2 操作步骤.....	10
2.3 寄存器介绍.....	11
3 EPWM.....	17
3.1 特性.....	17
3.1.1 独立频率控制多个 Buck 转换电路.....	19
3.1.2 相同频率控制多个 Buck 转换电路.....	21
3.1.3 控制多个 H 半桥电路（HHB）转换电路.....	22
3.1.4 控制电机的双三相逆变器（ACI 和 PMSM）.....	24
3.1.5 PWM 模块间相位控制的实际应用.....	26
3.1.6 控制三相交错 DC/DC 交换电路.....	27
3.1.7 控制 0 电压转换的全桥转换电路（ZVSFB）.....	29
3.1.8 控制峰值电流模式（Peak Current Mode）控制 Buck 模块.....	30
3.1.9 控制 H 桥 LLC 谐振变换器.....	31
3.2 子模块介绍.....	32
3.2.1 时基计数器子模块（Time-base）.....	32
3.2.2 时基计数比较器子模块（Counter-comparator）.....	33
3.2.3 动作产生子模块（Action-qualifier）.....	33
3.2.4 死区产生子模块（Dead-band）.....	33
3.2.5 事件触发子模块（Event-trigger）.....	33
3.2.6 故障区子模块（Trip-zone）.....	34
3.3 寄存器介绍.....	34
4 历史.....	71

关于本手册

本手册提供了 PWM 中所有模块的描述，包括模块介绍、使用步骤和寄存器等。包括了以下模块：SPWM、SVPWM 和 EPWM。

关于一些记数约定

本手册使用了以下通用约定，这些约定在本手册内有效：

- 十六进制数表示为 **suffix'h**。例如：16'ha301，表示该数的二进制数位宽为 16,十六进制数形式值为 a301。
- 十进制数表示为 **suffix'd**。例如：2'd3，表示该数的二进制数位宽为 2，十进制数形式值为 3。
- 二进制数表示为 **suffix'b**。例如：3b'001，表示该数的二进制数位宽为 3,二进制数形式值为 001。

1 SPWM

SPWM，正弦脉冲宽度调制，可以在 IO 输出指定占空比和周期的 PWM 波。支持三电平模式和两电平模式，支持单规则采样，即一个三角形周期只采样一次占空比。SPWM 的模块框图如下：

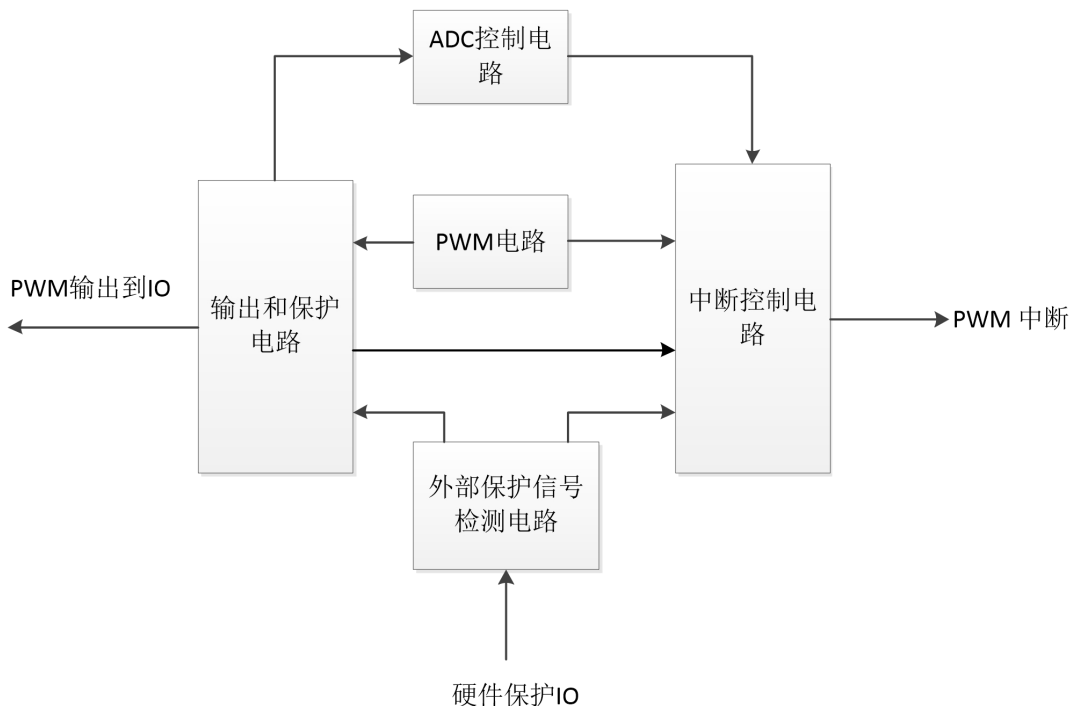


图 1-1 SPWM 的模块框图

1.1 操作步骤

- (1) 配置 SPWM_CON 寄存器，将 SFR_CFG_EN (SPWM_CON[24]) 设为 1。
- (2) 配置 SPWM_PERIOD0 和 SPWM_PERIOD1 寄存器：设置周期和死区时间，最小脉冲宽度推荐写 0。
- (3) 配置 SPWM_CNT0, SPWM_CNT1, SPWM_CNT2 寄存器：设置为三电平或者两电平并选择是否打开 load data 中断。其他全部选择默认值即可。
- (4) 配置 FAULT_INFO 和 FAULT_INVERT 寄存器，可关闭指定的故障检测 (FAULT_INFO[15:0])，并且可以设置低电平或高电平故障有效 (FAULT_INVERT[15:0])。
- (5) 配置 SPWM_MATCH 和 SPWM_ADCC 寄存器，可以在 SPWM 周期内任意一个时间产生匹配 pending 并触发中断，并且可以在 SPWM_ADCC 中配置是否触发 ADC 采样。
- (6) 配置 SPWM_CON 寄存器：设置 FAULT_DEBOUNCE (SPWM_CON[23:20]) 进行故障检测防抖时间；设置 SYSERR_PT_EN (SPWM_CON[25]) = 1，打开系统错误检测。设置 SPWM_ADCC 寄存器可以触发某个或某几个 ADC 采样。
- (7) 配置 SPWM_CON 寄存器，写 SET_PWM_EN (SPWM_CON[0]) 为 1，写

1SPWM

SFR_CFG_EN (SPWM_CON[24]) 为 0 关闭 SFR 写使能。

- (8) 定时将当前数据写入 PWM_BUFDATA (SPWM_DATABUF0/1/2[15:0])。也可以通过将寄存器 SPWM_CNT0, 1, 2 的 LOAD_INT_EN (SPWM_CNT0/1/2[18]) 置 1 来触发更新数据到 PWM_BUFDATA。

1.2 寄存器介绍

1-1 SPWM_CON

地址偏移: 0x00

Bit	名称	复位值	读/写	功能
31:25	保留	7'b0	R	-
26	AUTO_SCALE_DIS	0	RW	Auto scale 使能位。 0: 使能, 如果数据小于死区时间的一半, SPWM 在三电平模式下会输出 0; 如果数据小于死区时间但大于死区时间的一半, SPWM 会正常输出脉冲 1: 不使能, 如果数据小于死区时间, SPWM 在三电平模式下会输出 0
25	SYSERR_PT_EN	0	RW	系统故障保护使能位。 0: 不使能 1: 如果系统时钟停止或系统断电, SPWM 将自动停止。
24	SFR_CFG_EN	0	RW	PWM 配置 SFR 的写操作使能位, 包括 SPWM_CON[14:0], FAULT_INFO 和 FAULTY_INVERT。 0: CPU 无法向上述 SFR 写入数据 1: CPU 可以向上述 SFR 写入数据
23:20	FAULT_DEBOUNCE	0000	RW	硬件保护的故障防抖动时间的设置, 单位为 64 个系统时钟。
19	CLR_DATA_ERROR	0	W	数据缺失故障状态清除控制位。 0: 无作用 1: 清除数据缺失故障状态
18	-	0	R	-
17	CLR_HWP_PD	0	W	硬件保护状态的清除控制位。 0: 无作用 1: 清除
16	CLR_PWMEN	0	W	PWM 清除使能位。 0: 无作用 1: 清除 PWM 使能, PWM 关闭
15	DATA_ERROR	0	R	数据缺失故障状态位。 0: 正常 1: 发生数据缺失故障, 即 CPU 没有及时写入数据到 SPWM

1SPWM

14	DER_PROTECT_DIS	0	RW	数据缺失故障保护使能位。 0: 使能数据缺失故障保护, 如果发生数据缺失故障, PWM 将自动关闭 1: 关闭数据缺失故障保护
13	PWM_WORK	0	R	PWM 工作状态位。 0: PWM 空闲 1: PWM 正在工作
12	DER_INT_EN	0	R	数据缺失故障中断使能位。 0: 不使能 1: 使能
11	HWP_PENDING	0	R	硬件保护 pending 状态位。 0: 未发生硬件保护 1: 发生硬件保护
10	HWP_INT_EN	0	RW	硬件保护中断使能位。 0: 不使能 1: 使能
9	-	0	R	-
8	HWP_DIS	0	RW	硬件保护使能控制位。 0: 打开硬件保护 1: 关闭硬件保护
7	PWM_INVERT	0	RW	PWM 取反使能控制位。 0: 不开启 PWM 取反, 输出 1 代表 IGBT ON 1: 打开 PWM 取反, 输出 0 代表 IGBT ON.
6	-	0	R	-
5	-	0	R	-
4	-	0	R	-
3	PWM_SEL2	0	RW	PWM2 使用控制位。 0: 不使用 PWM2, 当 PWM_INVERT = 0 时其输出总为 0 或当 PWM_INVERT = 1 时为 1 1: 使用 PWM2
2	PWM_SEL1	0	RW	PWM1 使用控制位。 0: 不使用 PWM1, 当 PWM_INVERT = 0 时其输出总为 0 或当 PWM_INVERT = 1 时为 1 1: 使用 PWM1
1	PWM_SELO	0	RW	PWM0 使用控制位。 0: 不使用 PWM0, 当 PWM_INVERT = 0 时其输出总为 0 或当 PWM_INVERT = 1 时为 1 1: 使用 PWM0
0	SET_PWM_EN	0	W	PWM 使能控制位。 0: 无作用 1: 使能选中的 PWM

1-2 SPWM_PERIOD0

1SPWM

地址偏移: 0x04

Bit	名称	复位值	读/写	功能
31:16	保留	16'b0	-	-
15:0	PWM_PERIOD	16'b0	R/W	配置 PWM 周期, 单位为系统时钟。

1-3 SPWM_PERIOD1

地址偏移: 0x28

Bit	名称	复位值	读/写	功能
31:24	保留	8'b0	-	-
23:12	MIN_PULSE	22'b0	R/W	配置最小脉冲, 单位为系统时钟*2。
11:0	DEAD_TIME	12'b0	R/W	配置死区时间, 单位为系统时钟*2, 必须大于 1。

1-4 SPWM_MATCH

地址偏移: 0x2c

Bit	名称	复位值	读/写	功能
31:21	保留	11'b0	R	-
20	CLR_MATCH_PENDING	0	W	匹配 pending 清除控制位。 0: 无作用 1: 清除 match pending
19	MATCH_PENDING	0	R	Counter 与 MATCH_VALUE 的值的比较结果。 0: 不相等 1: 相等
18	MATCH_INTEN	0	R/W	Match 中断使能位。 0: 不使能 1: 使能
17:16	MATCH_PWM_SEL	00	R/W	选择哪一个 PWM 与 MATCH_VALUE 的值做比较。 00: 关闭比较 01: PWM0 10: PWM1 11: PWM2
15:0	MATCH_VALUE	16'b0	R/W	如果 PWM 的 counter 的值等于 MATCH_VALUE 的值, 那么 MATCH_PENDING 会被置 1, 直到被清除。MATCH_VALUE 的范围是 0 到 (PWM_PERIOD-1)。

1-5 SPWM_ADCC

地址偏移: 0x30

Bit	名称	复位值	读/写	功能
31:14	保留	18'b0	R	-
13:0	ADC_CONTROL_EN	14'b0	R/W	ADC 控制使能位, 选择控制哪一个 ADC。 0: 不使能 1: 使能, 当 MATCH_PENDING (SPWM_MATCH[19]) 变为 1, 立即启动 ADC

1SPWM

				转换
--	--	--	--	----

1-6 SPWM_CNT0~2

地址偏移: 0x34/0x38/0x3c

Bit	名称	复位值	读/写	功能
31:23	保留	9'b0	R	-
22	TWO_LEVEL	0	RW	电平模式选择位。 0: 三电平模式 1: 两电平模式
21	DEAD_BALANCE_MODE	1	R/W	死区平衡模式选择位, 该模式只有 PWM_CARRIER_MODE = 0 时才可用; 如果 PWM_CARRIER_MODE = 1, 该位无效。推荐 DEAD_BALANCE_MODE 设置为 1。 0: 不使能 1: 使能, 无论数据正负都要平衡死区时间, 但有可能导致脉冲时间小于设置的最小脉冲 (即 MIN_PULSE)
20	CLR_LOAD_PD	0	W	PWM 的 load data pending 清除控制位。 0: 无作用 1: 清除
19	LOAD_PENDING	0	R/W	SPWM 数据加载完成标志位。 0: 数据未加载到 PWM_DATAUSE 1: 已将数据从 PWM_BUFDATA 加载进 PWM_DATAUSE, CPU 可以写入新的数据到 PWM_BUFDATA。
18	LOAD_INT_EN	0	R/W	数据加载中断使能位。 0: 不使能 1: 使能
17	PWM_DER_DETECT_DIS	0	R/W	PWM 数据缺失错误检测使能位。 0: 使能数据缺失错误检测 1: 不使能数据缺失错误检测
16	PWM_CARRIER_MODE	0	R/W	PWM 三角载波模式选择位。 0: Mode0, 正负值都有相同的相位; 当三角载波计数值等于 0 时, 数据会被加载到 PWM_DATAUSE 寄存器内 1: Mode1, 正负值有不同的相位; 当三角载波计数值等于最大值时, 数据会被加载到 PWM_DATAUSE 中, 推荐使用该模式。
15	PWM_CNT_INI_DIR	0	R/W	PWM 初始化计数的方向选择位。 0: 启动 PWM 时计数增加 1 启动 PWM 时计数减少
14:0	PWM_CNT_INI_VAL	15'b0	R/W	PWM 计数初始值。

1SPWM

1-7 FAULT_INFO

地址偏移: 0x20

Bit	名称	复位值	读/写	功能
31:16	FAULT_PENDING	16'b0	RW	一共有 16 个故障源, 映射到以下引脚 {PB28~PB31, PC0~PC11}, 如果故障源有效, 这个 pending 设为 1。
15:0	FAULT_DIS	16'b0	RW	故障检测使能位。 0: 使能故障检测 1: 不使能故障检测

1-8 FAULT_INVERT

地址偏移: 0x24

Bit	名称	复位值	读/写	功能
31:16	保留	16'b0	RW	-
15:0	FAULT_INVERT	16'b0	RW	故障源反转控制位。 0: 当输入为 1 时故障源有效 1: 当输入为 0 时故障源有效

1-9 SPWM_DATAUSE0/1/2

地址偏移: 0x08/0x10/0x18

Bit	名称	复位值	读/写	功能
31:16	保留	16'b0	RW	-
15:0	PWM_USEDATA	16'b0	RW	PWM 使用数据, PWM 将这些数据发送到 PWM 发生器中。

1-10 SPWM_DATABUF0/1/2

地址偏移: 0x0c/0x14/0x1c

Bit	名称	复位值	读/写	功能
31:16	保留	16'b0	RW	-
15:0	PWM_BUFDATA	16'b0	RW	PWM buffer 的数据。 当 PWM 内部计数 counter 等于特定的值时 (由 PWM_CARRIER_MODE 选择), PWM 将此数据加载到 SPWM_DATAUSE0/1/2 寄存器。

2 SVPWM

TMF6200 集成了一个基于空间矢量脉冲宽度调制模块，本模块的主要功能是根据提供的 abc 坐标的输入数据，自动转换为三相 2 电平或者三相 3 电平的互补对称 PWM 输出。模块提供了可配置的死区控制和必要的硬件保护模式，当遇到紧急情况，可及时停止 PWM 输出，以保护外围硬件。

2.1 特性

- (1) 支持三相 abc 信号：Ua、Ub、Uc，位宽为 16bit。
- (2) 支持三相 2 电平或三相 3 电平的 SVPWM。
- (3) 支持可配置的死区控制。
- (4) 支持硬件保护。

SVPWM 的模块框图如下：

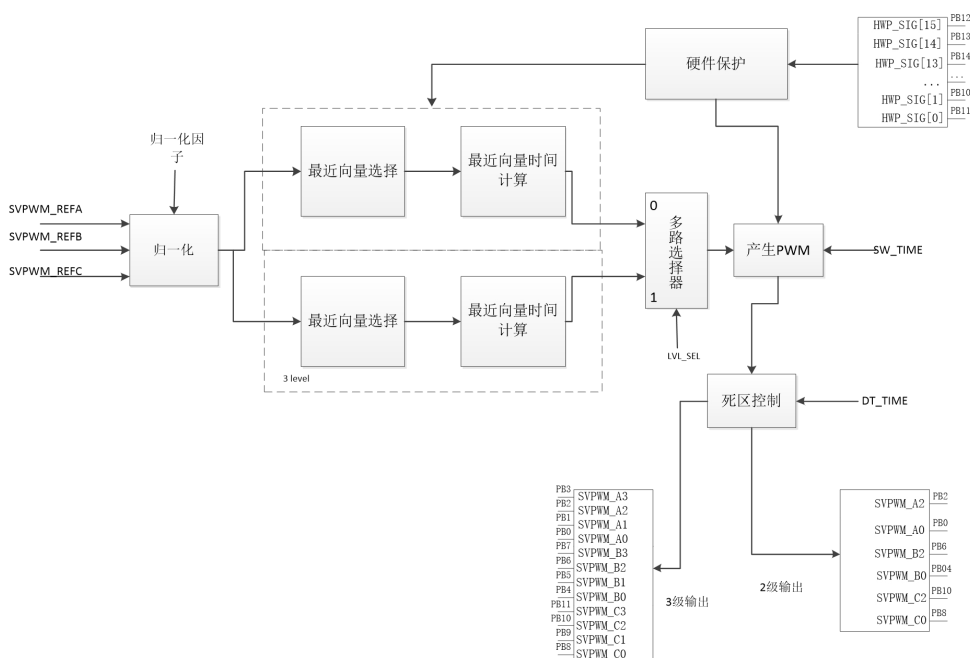


图 2-1 SVPWM 的模块框图

2.2 操作步骤

- (1) 出于安全控制的考虑，需将寄存器 SVPWM_CON0 的 SVPWM_CFG_EN 设为 1，部分寄存器才能进行写操作。用户可根据需要（详情请参考下方详细的寄存器介绍）进行配置。
- (2) 将寄存器 SVPWM_CON0 的 LVL_SEL 设为三电平或两电平输出。
- (3) 在寄存器 SVPWM_CON1 的 SW_TIME 处设置需要的 PWM 开关周期。
- (4) 配置外部硬件保护：HWPRT_EN（SVPWM_CON0[4]）、FAULT_DEAT_DIS（SVPWM_CON4[15:0]）和 FAULT_INVERT（SVPWM_CON5[15:0]）；你可以关闭

2 SVPWM

指定的故障检测，设置故障信号高电平或低电平时有效。根据需要，可配置硬件故障信号的防抖动时间：FAULT_DEBC_TIME（SVPWM_CON0）。

- (5) 根据需要配置输入信号的归一化因子：NORM_FACTOR_SEL（SVPWM_CON0[3]）、NORM_FACTOR（SVPWM_CON3[15:0]）。
- (6) 配置 SVPWM_MATCH_CON 和 SVPWM_ADC_CON 寄存器，可生成用于 ADC 采样的匹配标志位，该匹配位置可以是 SVPWM 周期内任一时间，并且可以选择是否触发中断。
- (7) 配置 LOADDATA_INT_EN（SVPWM_CON0[5]）、HWP_INT_EN（SVPWM_CON0[6]）、MATCH_INT_EN（SVPWM_CON0[26]）和 DER_INT_EN（SVPWM_CON0[27]）
 - a) 进行中断设置。
- (8) 在 SVPWM_REFA, SVPWM_REFB, SVPWM_REFC 中配置 abc 坐标数据。
- (9) 启动 SVPWM：配置 M_EN（SVPWM_CON0[0]）为 1。
- (10) 等待数据被写入，即当 LOADDATA_PND 变为 1 或 load data 中断发生时；然后将数据更新到寄存器 SVPWM_REFA, SVPWM_REFB, SVPWM_REFC 中。
- (11) 如果想要关闭 SVPWM，配置 CLR_SVPWMEN（SVPWM_CON0[14]）为 1。

2.3 寄存器介绍

2-1 SVPWM_CON0

地址偏移：0x0000

Bit	名称	复位值	读/写	功能
31:28	保留	4'b0	RO	读操作：返回 0；写操作：无任何效果
27	DER_INT_EN	0	RW	数据更新出错中断使能控制位，仅当 DATAERR_PND=1 时有效。 0：不使能 1：使能
26	MATCH_INT_EN	0	RW	SVPWM 的计数器匹配中断使能位。 0：不使能 1：使能
25	CLR_MATCH_PND	0	WO	SVPWM 的计数器匹配标志清除控制位。 0：无清除 1：清除匹配标志
24	SVPWM_MATCH_PND	0	RO	SVPWM 的计数器匹配标志位，当 SVPWM 内部计数器值与 SVPWM_MATCH_CON 设置的值相等时，此标志位将被置 1。 0：不相等 1：相等
23	SVPWM_MATCH_EN	0	RW	SVPWM 的计数器匹配功能使能位。 0：不使能 1：使能
22	PWMOUTINVT	0	RW	SVPWM 的 PWM 输出反相控制位。 0：不转换 1：转换

2 SVPWM

21	SVPWM_ON	0	RO	SVPWM 工作状态标志位。 0: SVPWM 处于空闲状态，模块没有工作 1: SVPWM 正在工作
20	DATAERR_PND	0	RO	数据出错指示标志位。在不恰当的时刻更新 abc 坐标数据，将会导致数据丢失。 读操作： 0: 数据更新正确 1 在 SVPWM 周期内发生数据丢失 写操作：没有任何效果
19	LOADDATA_PND	0	RO	输入数据是否已被使用的标志。 0: 写入的输入数据没有被使用前，不能再次写入新的数据，否则前一个输入数据将会丢失。 1: 写入的数据已经被使用，可以写入新的数据。
18	HWP_PND	0	RW	硬件保护标志位，仅当 HWP_EN=1 时有效。 0: 处于硬件保护模式 1: 处于非硬件保护模式
17	CLR DERP	0	WO	DATAERR_PND 的清除控制位。 0: 无任何效果 1: 清除 DATAERR_PND，写操作后 DATAERR_PND =0。
16	CLR_LDP	0	WO	LOADDATA_PND 的清除控制位。 0: 无任何效果 1: 清除 LOADDATA_PND，写操作后 LOADDATA_PND =0
15	CLR_HWP	0	WO	HWP_PND 的清除控制位。 0: 无任何效果 1: 清除 HWP_PND，写操作后 HWP_PND=0
14	CLR_SVPWMEN	0	WO	软件关闭 SVPWM 功能。 0: 无任何效果 1: 关闭 SVPWM
13	SVPWM_CFG_EN	0	RW	SVPWM 相关控制寄存器的写使能控制位。 0: SVPWM 先关控制寄存器无法写入 1: SVPWM 相关控制寄存器允许写入
12:9	FAULT_DEBC_TIME	4'b0	RW	输入的硬件故障信号的去抖动时间，当检测到出现硬件错误时，模块自动进入硬件保护状态，并会自动将 svpwm 模块关闭（需配置 HWPRT_EN=1）。 实际配置的时间为： time = FAULT_DEBC_TIME * 64 * Tclk; Tclk: 模块工作时钟周期; 0: (0+1)*64*Tclk; the protection time; ...



2 SVPWM

				15: (15+1)*64*Tclk;
8	Reserved	0	RO	读操作: 返回 0 写操作: 无任何效果
7	DERP_EN	0	RW	数据出错后 (DATAERR_PND=1), 是否自动关闭 SVPWM 模块的控制位。 0: 不自动关闭 1: 自动关闭
6	HWP_INT_EN	0	RW	硬件保护中断产生控制位, 当硬件保护标志位 HWP_PND = 1 时有效。 0: 不产生中断 1: 产生中断
5	LOADDATA_INT_EN	0	RW	中断产生控制位, 当输入数据使用情况指示标志位 LOADDATA_PND = 1 时有效。 0: 不产生中断 1: 产生中断
4	HWPRT_EN	0	RW	硬件保护启动控制位。 0: 不启动 1: 启动
3	NORM_FACTOR_SEL	0	RW	ABC 坐标输入数据的归一化因子选择位。 0: 选择内置归一化因子。对于两电平, 内部归一化因子为 $\frac{1}{2}\sqrt{3}$; 对于三电平, 归一化因子为 $\sqrt{3}$; 1: 选择有用户配置的归一化因子, 通过寄存器 VPWM_CON3 配置
2	DT_EN	0	RW	死区控制使能。使能后, 死区时间通过寄存器 SVPWM_CON2 进行配置。 0: 打开死区控制功能 1: 关闭死区控制功能
1	LVL_SEL	0	RW	两电平、三电平 SVPWM 选择位。 0: 两电平 SVPWM 1: 三电平 SVPWM
0	M_EN	0	RW	软件打开 SVPWM 功能。当出现以下三种情况之一, 该位会被自动置零并且关闭 SVPWM 功能。 (1) 当 DERP_EN=1 时, 数据出错后 DATAERR_PND=1 (2) 当 HWPRT_EN=1 时, 硬件保护标志位 HWP_PND = 1 (3) 当 CLR_SVWPMEN=1 时, 0: 关闭 1: 打开

2-2 SVPWM_CON1

2 SVPWM

地址偏移: 0x0004

Bit	名称	复位值	读/写	功能
31:16	Reserved	0	RO	读操作: 返回 0; 写操作: 无任何效果
15:0	SW_TIME	0	RW	SVPWM 开关周期时间, 配置值应为实际开关时间的 1/2。 $SW_TIME = \text{round}(1/2 * T_{sw}/T_{clk})$ T_{sw} : 实际开关周期时间 T_{clk} : SVPWM 工作时钟周期 该寄存器需要设置 SVPWM_CFG_EN=1 后才能进行写操作。

2-3 SVPWM_CON2

地址偏移: 0x0008

Bit	名称	复位值	读/写	功能
31:12	Reserved	20'b0	RO	读操作: 返回 0; 写操作: 无任何效果
11:0	DT_TIME	12'b0	RW	SVPWM 死区控制时间。 $DT_TIME = \text{round}(T_{dt}/T_{clk})$, T_{dt} : 实际死区控制时间, T_{clk} : SVPWM 工作时钟周期。 该寄存器需要设置 SVPWM_CFG_EN=1 后, 才能进行配置。

2-4 SVPWM_CON3

地址偏移: 0x000C

Bit	名称	复位值	读/写	功能
31:16	Reserved	16'b0	RO	读操作: 返回 0; 写操作: 没有任何效果
15:0	NORM_FACTOR	16'b0	RW	abc 坐标输入数据的归一化因子。当 NORM_FACTOR_SEL=1 是配置才有效。 公式为: $NORM_FACTOR = m\sqrt{3}$, m 需小于 0.866, 定标方法为 Q2.14。

2-5 SVPWM_CON4

地址偏移: 0x0010

Bit	名称	复位值	读/写	功能
31:16	HWP_SIG	16'b0	RO	R : 指示引起硬件保护的信号。 W : 无任何效果
15:0	FAULT_DECT_DIS	16'b0	RW	硬件故障检测使能位。 0: 打开硬件检测 1: 关闭硬件检测 需要设置 SVPWM_CFG_EN = 1 后, FAULT_DECT_DIS 才能进行写操作。

2-6 SVPWM_CON5

地址偏移: 0x0014

Bit	名称	复位值	读/写	功能
31:16	保留	16'b0	RO	读操作: 返回 0; 写操作: 无任何效果

2 SVPWM

15:0	FAULT_INVERT	16'b0	RW	模块内部对 HWP_SIG 信号进行硬件错误判断时，默认 HWP_SIG[n]=1 时，表示发生错误；若需要 HWP_SIG[n]=0 时，表示发生错误，配置相应的 FAULT_INVERT[n]=1。 0: 当 HWP_SIG[n]=1 表示发生错误 1: 当 HWP_SIG[n]=0 表示发生错误
------	--------------	-------	----	---

2-7 SVPWM_REFA

地址偏移: 0x0018

Bit	名称	复位值	读/写	功能
31:16	保留	16'b0	RO	读操作: 返回 0; 写操作: 无任何效果
15:0	SVPWM_REFA	16'b0	RW	A 相输入参考信号 (abc 坐标)。

2-8 SVPWM_REFB

地址偏移: 0x001C

Bit	名称	复位值	读/写	功能
31:16	保留	16'b0	RO	读操作: 返回 0; 写操作: 无任何效果
15:0	SVPWM_REFB	16'b0	RW	B 相输入参考信号 (abc 坐标)。

2-9 SVPWM_REFC

地址偏移: 0x0020

Bit	名称	复位值	读/写	功能
31:16	保留	16'b0	RO	读操作: 返回 0; 写操作: 无任何效果
15:0	SVPWM_REFC	16'b0	RW	C 相输入参考信号 (abc 坐标)。

2-10 SVPWM_STC0

地址偏移: 0x0024

Bit	名称	复位值	读/写	功能
31:16	SVPWM_STC_B	16'b0	RO	在一个开关周期中，向量 B 的有效时间。
15:0	SVPWM_STC_A1	16'b0	RO	在一个开关周期中，向量 A1 的有效时间。

2-11 SVPWM_STC1

地址偏移: 0x0028

Bit	名称	复位值	读/写	功能
31:16	SVPWM_STC_A2	0	RO	在一个开关周期中，向量 A2 的有效时间。
15:0	SVPWM_STC_C	0	RO	在一个开关周期中，向量 C 的有效时间。

2-12 SVPWM_MATCH_CON

地址偏移: 0x002c

Bit	名称	复位值	读/写	功能
31:17	保留	15'b0	RO	读操作: 返回 0; 写操作: 无任何效果
16:0	MATCH_CON	17'b0	RW	当 SVPWM 内部周期计数器的值与 MATCH_CON 的值相等时，将会产生一个脉冲信号标志位和一个中断，此位用于配置对应的匹配值。

2-13 SVPWM_ADC_CON

地址偏移: 0x0030

Bit	名称	复位值	读/写	功能
-----	----	-----	-----	----

2 SVPWM

31:14	保留	18'b0	RO	读操作：返回 0；写操作：无任何效果
13:0	ADC_AUTO_EN	14'b0	RW	SVPWM 的 ADC 转换使能位，触发信号为匹配脉冲信号，由 SVPWM_MATCH_CON 控制。 0：不触发 ADC 转换 1：触发 ADC 转换 注意：TMF6200 内部集成了 14 个 ADC，此位用于选择出发哪一个 ADC 进行采样。

3 EPWM

EPWM 外设在许多商业及工业级电力电子系统控制中是一个关键模块。这些系统包括数字电机控制、电源开关模式控制、不间断电源 UPS 和其他形式的电源转换模块。EPWM 有时可作为 DAC 使用，占空比即为 DAC 输出的模拟量。

3.1 特性

在 TMF6200 中一共有 7 个 EPWM 模块，每个 EPWM 模块都支持以下功能：

1. 专用的 16 位时间控制器，可用于周期和频率的控制。
2. 两个 PWM 输出（EPWMxA 和 EPWMxB）可进行以下配置：
 - 两个独立的 PWM 输出进行单边控制；
 - 两个独立的 PWM 输出进行双边对称控制；
 - 一个独立的 PWM 输出进行双边不对称控制；
3. 通过软件对 PWM 信号进行异步覆盖控制。
4. 各 EPWM 可与其他 EPWM 模块进行可编程的超前或滞后相位控制。
5. 可在每个周期对各个 EPWM 进行硬件锁定（同步）相位关系。
6. 具有独立的上升沿和下降沿死区延迟控制。
7. 可编程故障区（trip zone）用于故障时的周期循环（cycle-by-cycle trip）控制和单次（one-shot trip）控制。
8. 可通过 CPU，IO 或者比较器使 PWM 输出为强制高、低或高阻抗逻辑电平。
9. EPWM 模块中的所有事件都可以触发 CPU 中断和启动 ADC 开始转换（ADC SOC）。
10. 可编程事件有效降低了中断时 CPU 的负担。

如图所示，每个 EPWM 模块都连接着输入输出信号。

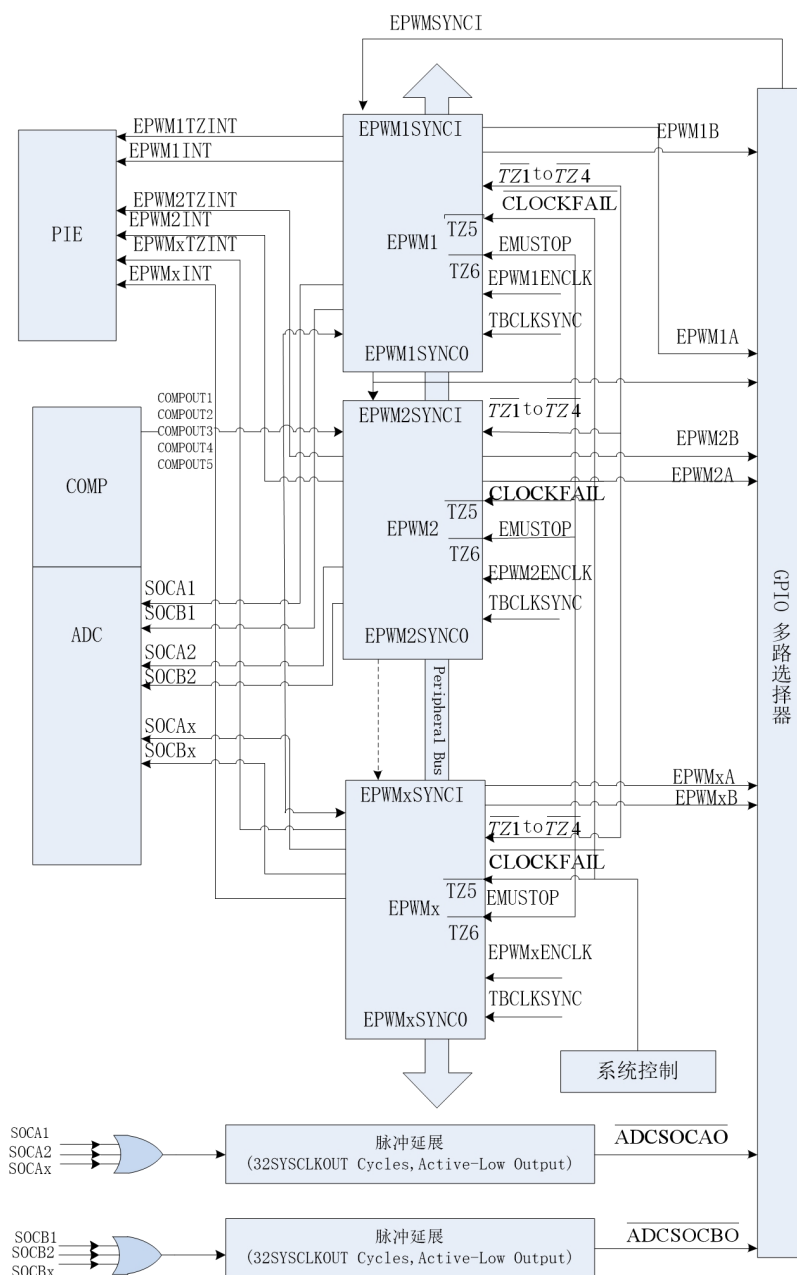


图 3-1 EPWM 模块总框图

有关各 EPWM 的同步功能和主要配置选项如下：

1. 选择同步（Sync in）信号源；
2. 使能同步选通脉冲开关，出现同步脉冲时会将相位寄存器中的数值加载进计数器；
3. 也可不做任何操作，关掉同步选通脉冲开关；
4. SyncOut 的连接选择：
 - Sync flow-through —— 将 SyncOut 直接与 SyncIn 相连；
 - Master 模式下，在 PWM 边界提供一个 sync 信号——SyncOut 与 CTR == 0 连接；
 - Master 模式下，在任意可编程时间点下提供一个 sync 信号——SyncOut 与 CTR == CMPA 相连或者 CTR == CMPB 相连；
 - 模块是单独一个模式并且和其他模块不同步时——SyncOut 与 X 相连（不使能）；

最常见的两种——Master 模式和 Slave 模式——如下图所示：

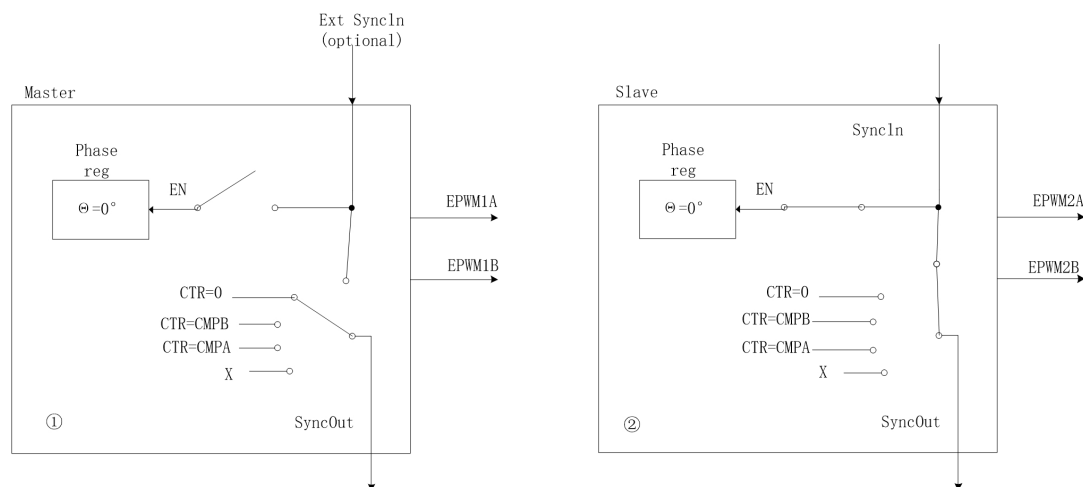
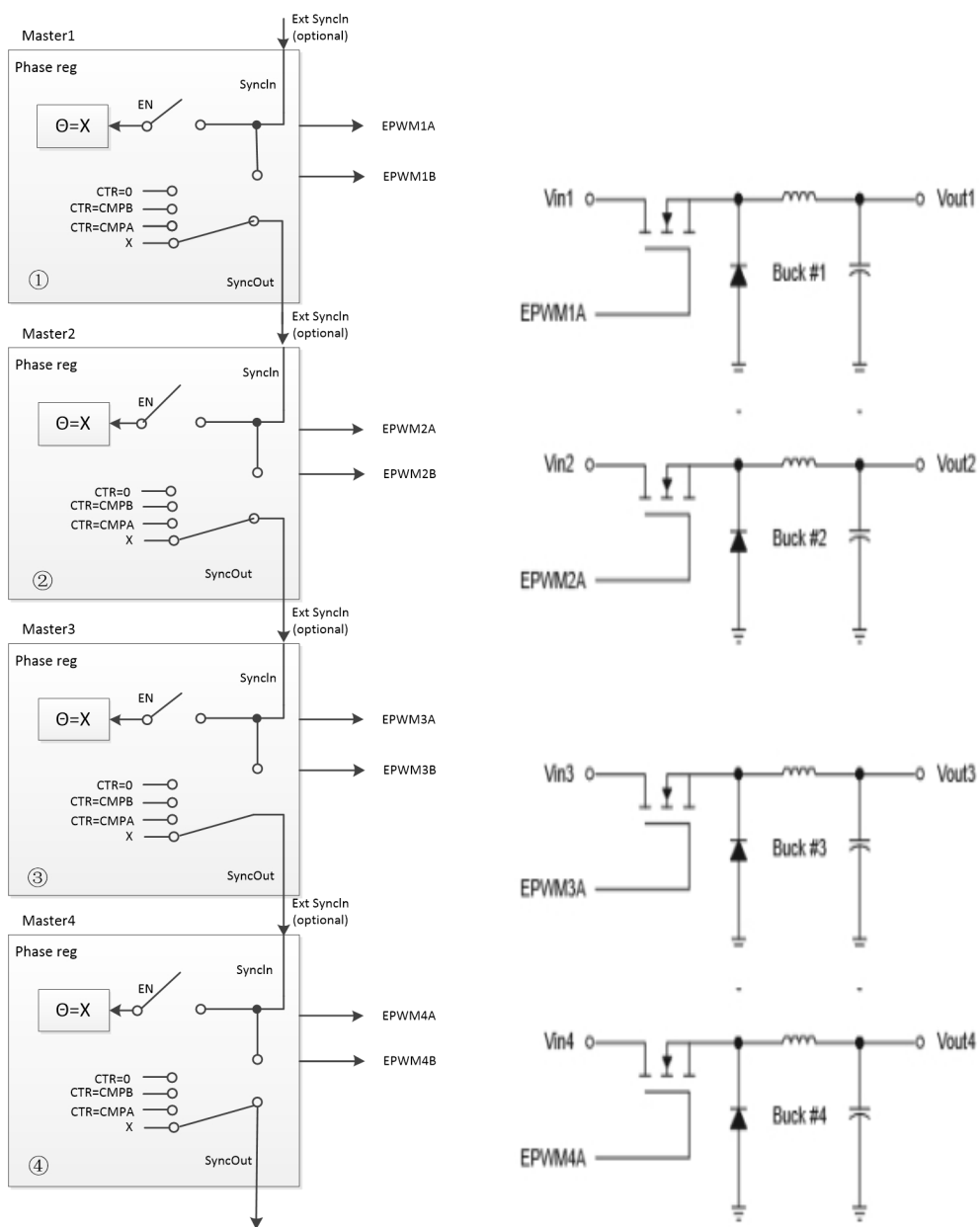


图 3-2 EPWM1 配置为典型 Master，EPWM2 配置为 Slave

3.1.1 独立频率控制多个 Buck 转换电路

最简单的功率变换电路拓扑之一是 buck。将一个 EPWM 模块配置为 master 时，可以用一个频率控制两个 buck 级。如果要求每个 buck 电路用独立频率控制，那么每个 converter 级都需要分配一个 EPWM 模块。下图 3-3 显示了 4 个 buck 级，每个都工作在独立频率上。在这种情况下，所有的四个 EPWM 模块则都被配置为 Master 并且均不同步。图 3-4 为图 3-3 的配置下生成的波形；注意即使有四个 buck 级，但是一共只有三种波形。



Note: $\Theta = X$ indicates value in phase register is a "don't care"

图 3-3 四个 Buck 级的控制,此时 $EPWM1 \neq EPWM2 \neq EPWM3 \neq EPWM4$

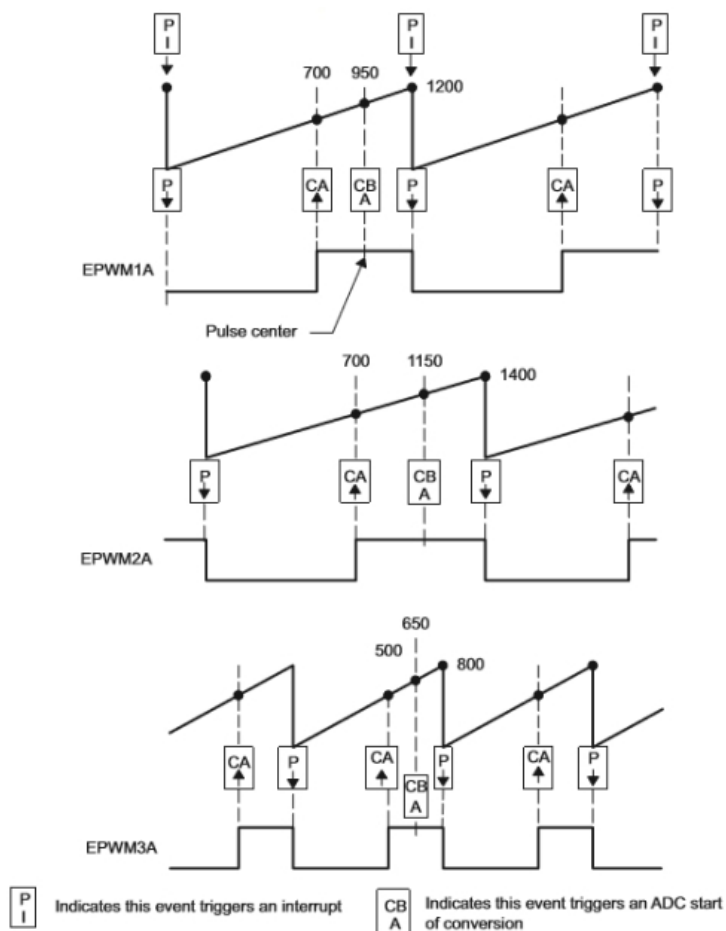


图 3-4 图 3-3 的 Buck 波形图（注意：这里只显示了三个 Buck 电路）

3.1.2 相同频率控制多个 Buck 转换电路

如果要求两个 Buck 电路同步，那么 EPWM2 可以被配置为 **slave** 并且工作在 EPWM1 的整数倍频率下。从 **master** 到 **slave** 的 **sync** 信号可以确保这些模块保持锁定。图 3-5 显示的就是这种设置下的模型。图 3-6 是图 3-5 所示配置下产生的波形。

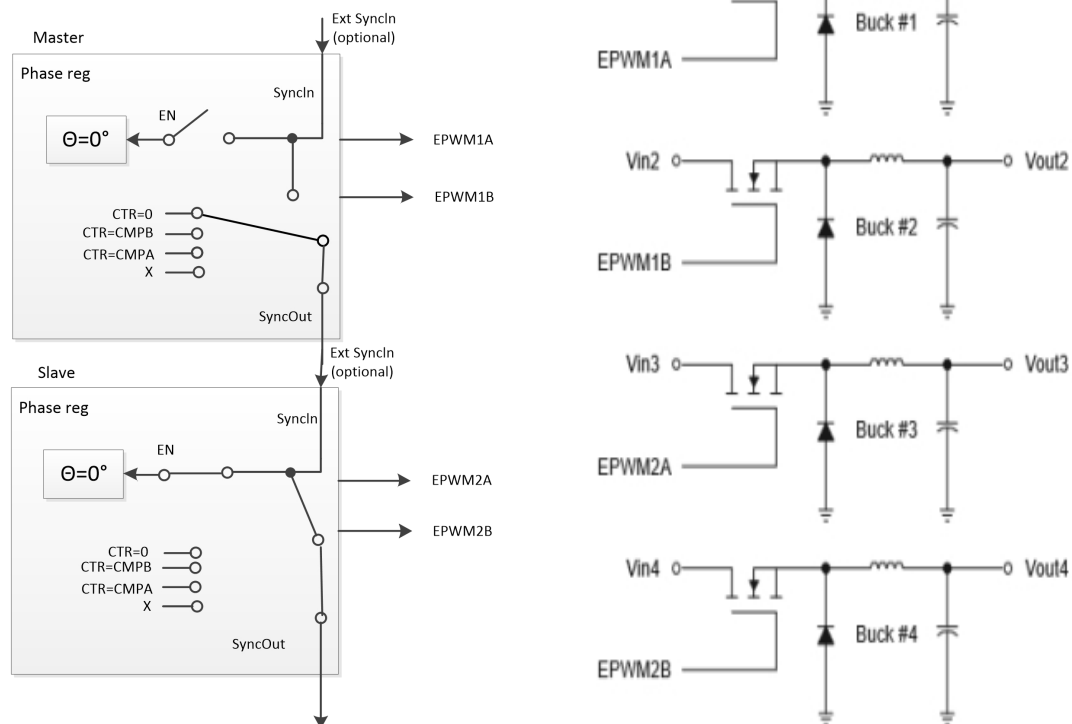


图 3-5 控制四个 Buck 电路（注意：EPWM2 = N x EPWM1）

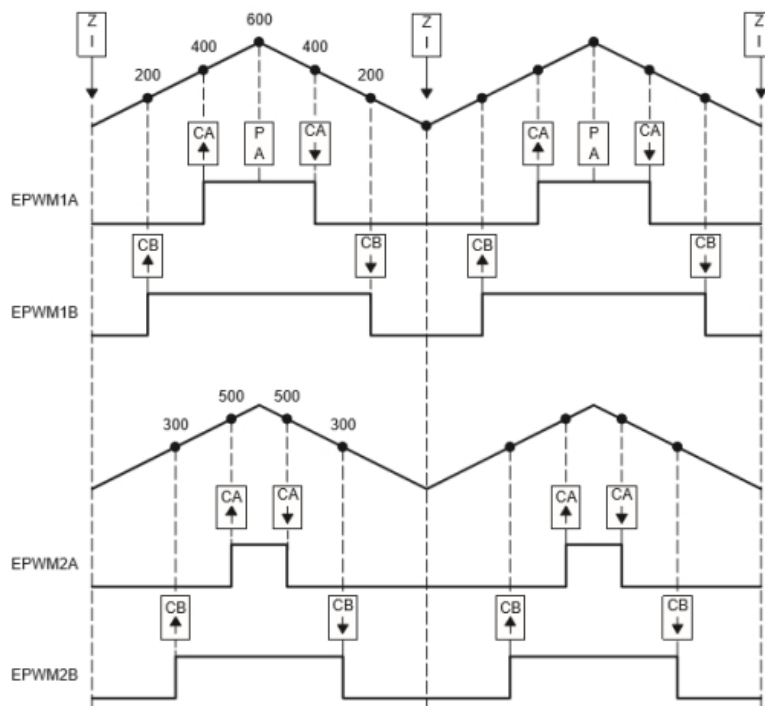


图 3-6 图 3-5 的 Buck 波形图（注意：EPWM2 = EPWM1）

3.1.3 控制多个 H 半桥电路（HHB）转换电路

3 EPWM

控制多个开关元件的拓扑也可以用相同的 EPWM 模块处理。用一个 EPWM 模块控制一个 H 半桥级电路是有可能的。这种控制的方法还可以被扩充至多级 H 半桥级电路。图 3-7 显示在控制两个同步的 H 半桥级电路时，可以使第二级电路工作在第一级电路的工作频率的整数倍下。图 3-8 是在这种设置下产生的波形。

Slave 被配置为 Sync flow-through。如果需要，可以增加第三个 H 半桥由第三个 EPWM 模块控制，并且必须与 master 同步。

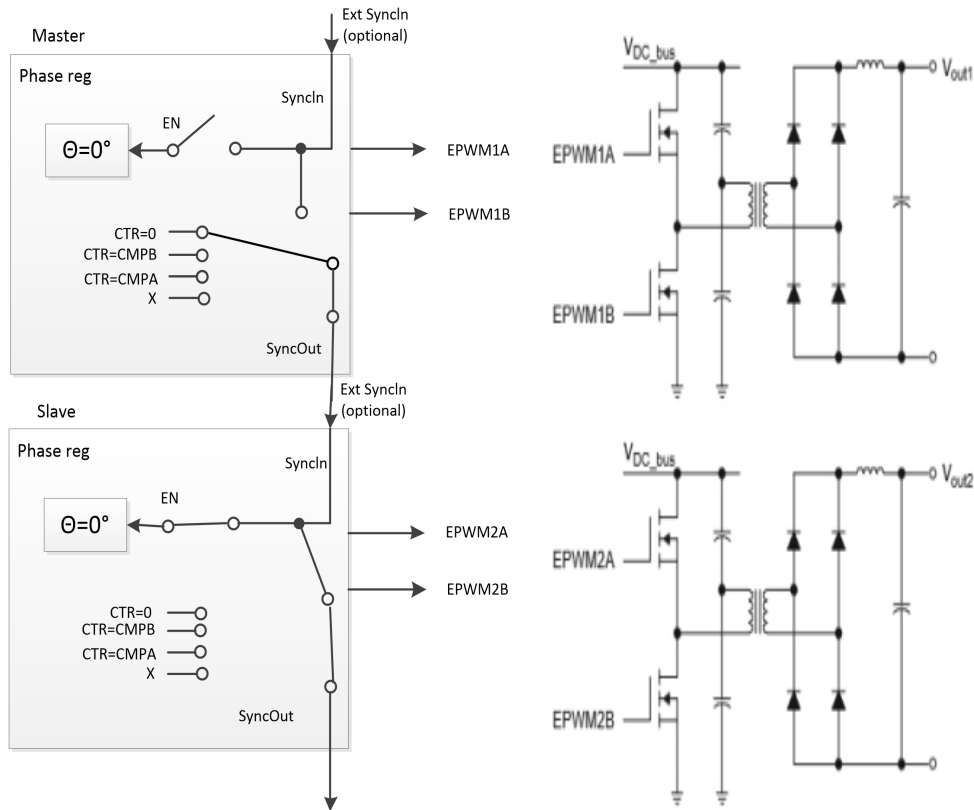


图 3-7 控制两个 H 半桥级 (EPWM2 = N x EPWM1)

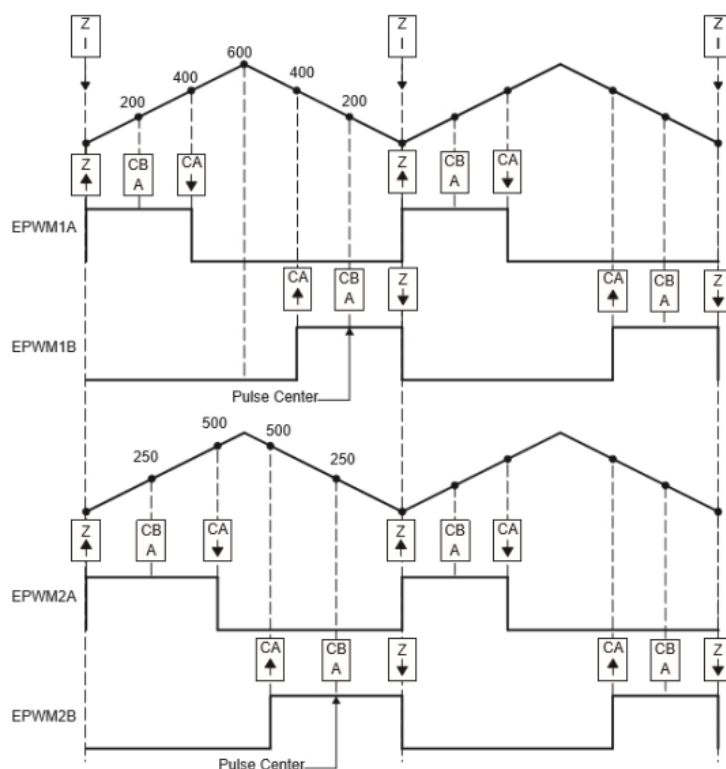


图 3-8 图 3-7 的 H 半桥波形图（注意：此时 EPWM2 = EPWM1）

3.1.4 控制电机的双三相逆变器（ACI 和 PMSM）

多模块控制单个功率级的思想可以扩充至三相逆变器中。在这种情况下，6 个开关元件可以由三个 PWM 模块控制，每支路一个。每条支路都必须在相同的频率下开关并且所有的支路都要同步。所以，1 个 master+2 个 slave 的组合可以轻松地解决这样的需求。图 3-9 显示了 6 个 PWM 模块是如何控制 2 个独立的三相逆变器的，每一个模块都运行着一个电机。

如前几节所示，我们可以让每个逆变器都工作在不同的频率下（例如图 3-9 的例子中模块 1 和 4 都作为 master）；或者设置 1 个为 master、5 个为 slave，使两个逆变器同步。在这种情况下，模块 4、5、6（三个模块的频率相等）的频率是模块 1、2、3（三个模块的频率相等）的频率的整数倍。

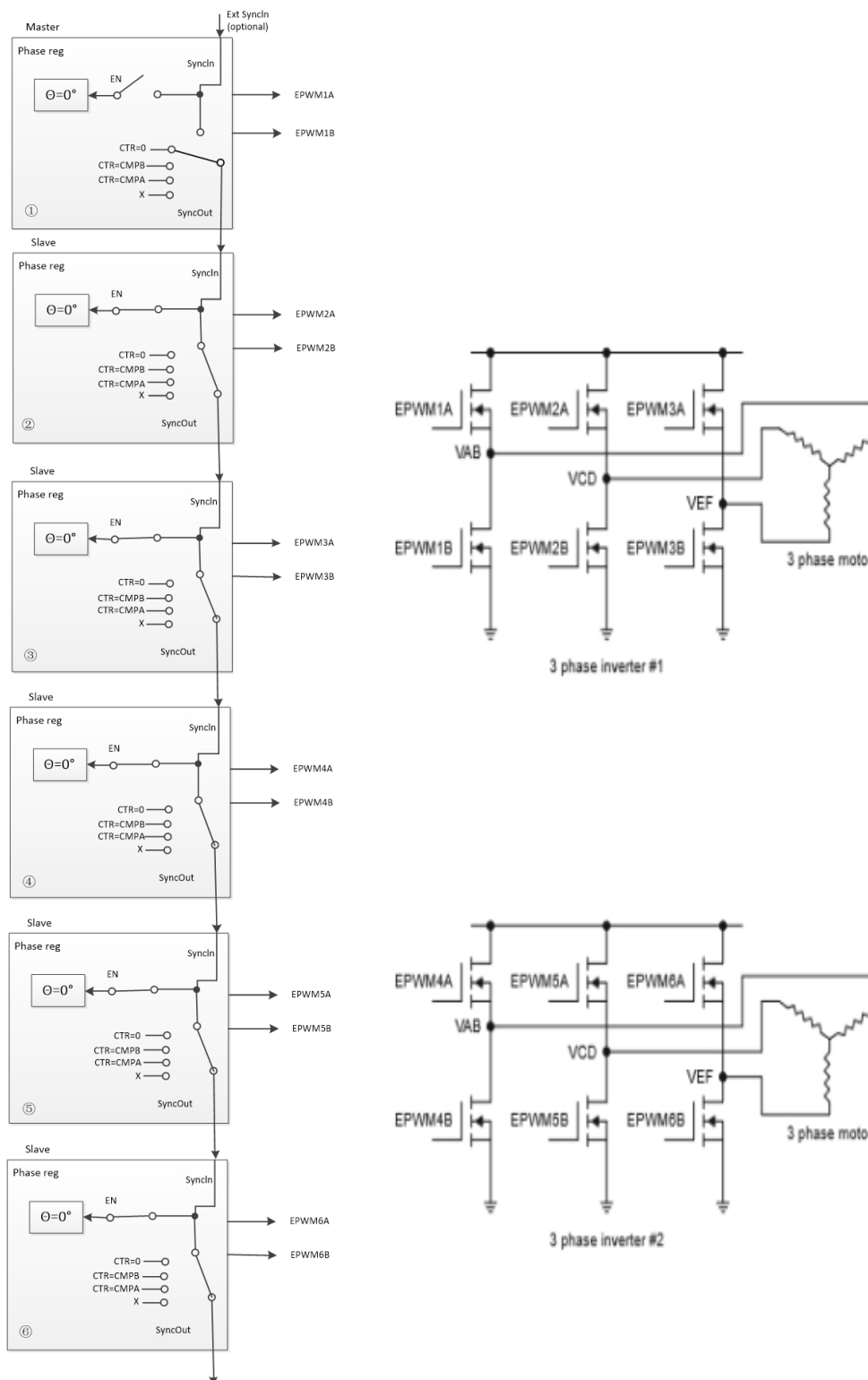


图 3-9 在电机控制中常用的双三相逆变电路控制

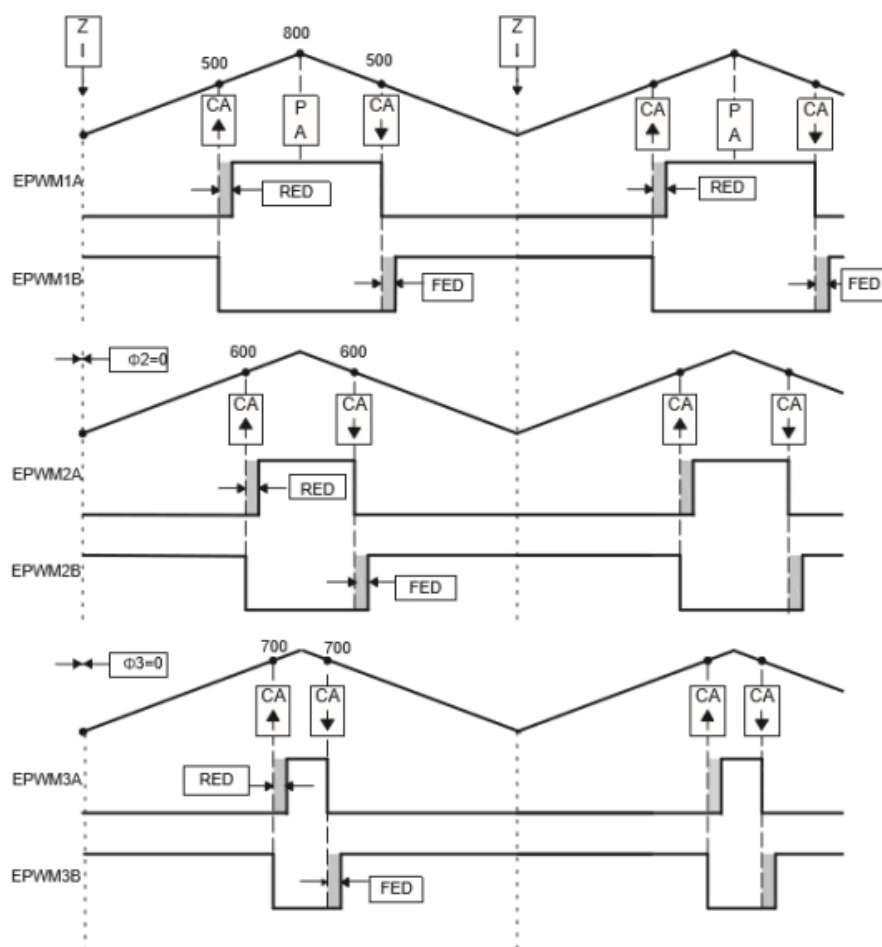


图 3-10 图 3-9 的双三相逆变电路波形图（只显示一个逆变电路波形图）

3.1.5 PWM 模块间相位控制的实际应用

直到目前为止，没有一种例子使用了相位寄存器（TBPHS）。相位寄存器不是被设置为 0 就是被完全忽略该值。然而，通过给 TBPHS 寄存器配置适当的值，多个 PWM 模块可以处理其他的功率拓扑级，这些拓扑级依靠各支路间的相位关系可以进行正确的操作。一个 PWM 模块可以配置 SyncIn 脉冲使得 TBPHS 寄存器可以加载进 TBCTR 寄存器。如上述观点所示，图 3-11 显示了一个相位关系为 120° 的 master 和 slave。

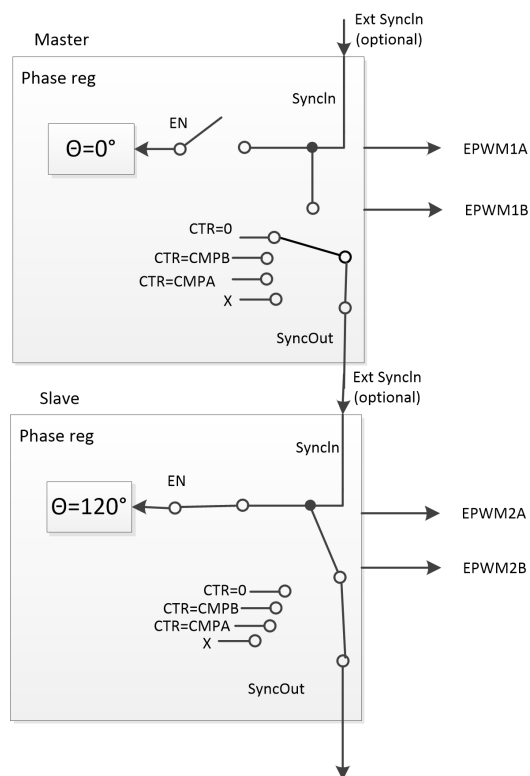


图 3-11 配置两个 PWM 模块进行相位控制

而图 3-12 显示了这个配置下产生的时序波形图。Master 和 slave 的 TBPRD 都等于 600。Slave 的 TBPHS = 200 ($200/600 \times 360^\circ = 120^\circ$)。当 master 产生 SyncIn 脉冲时，TBPHS = 200 的值就会被加载进 slave 的 TBCTR 寄存器。所以 slave 的时基总是超前于 master 的时基 120° 。

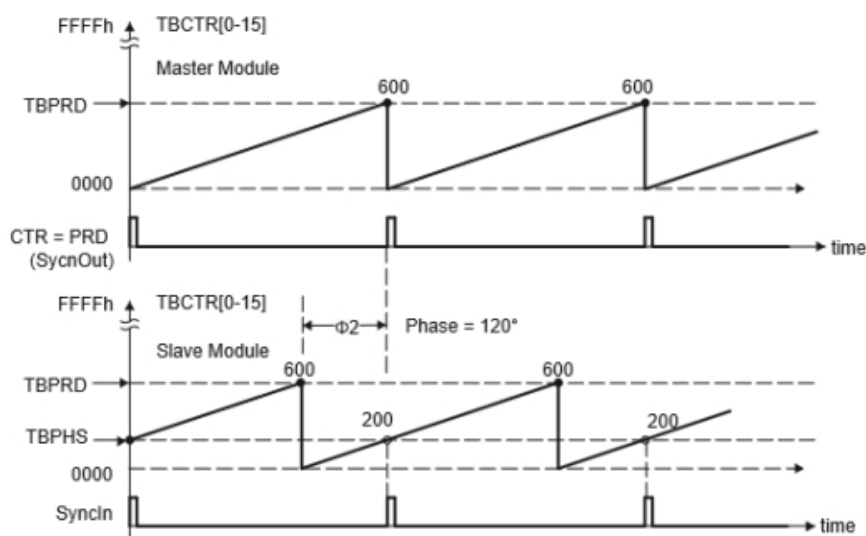


图 3-12 两个模块间相位控制的时序图

3.1.6 控制三相交错 DC/DC 交换电路

3 EPWM

如图 3-13，通用的功率拓扑是利用了模块之间的相位偏移。该系统使用了 3 个 PWM 模块，其中模块 1 被配置为 master。工作时，相邻的模块之间的相位关系为 $F = 120^\circ$ 。这是通过分别设置 slave 的 TBPHS 寄存器 2 和 TBPHS 寄存器 3 的值为周期的 1/3 和 2/3 来实现的。例如，如果周期寄存器被载入一个 900 计数的值，那么 $TBPHS(\text{slave2}) = 300$ 并且 $TBPHS(\text{slave3}) = 600$ 。所有的 slave 模块都要与 master1 模块同步。这个思想通过适当设置 TBPHS 寄存器的值可以扩充 4 个甚至更多的相位。下列公式提供了 TBPHS 寄存器 N 个相位值：

$$TBPHS(N, M) = (TBPRD/N) * (-1)$$

其中，N 为相位值的个数

M 为 PWM 模块的个数

例如，在 3 个相位值的情况下， $TBPRD = 900$ ，

$$TBPHS(3, 2) = (900/3) * (2-1) = 300 \text{ (这是 slave 模块 2 的相位值);}$$

$$TBPHS(3, 3) = (600) \text{ (这是 slave 模块 3 的相位值)。}$$

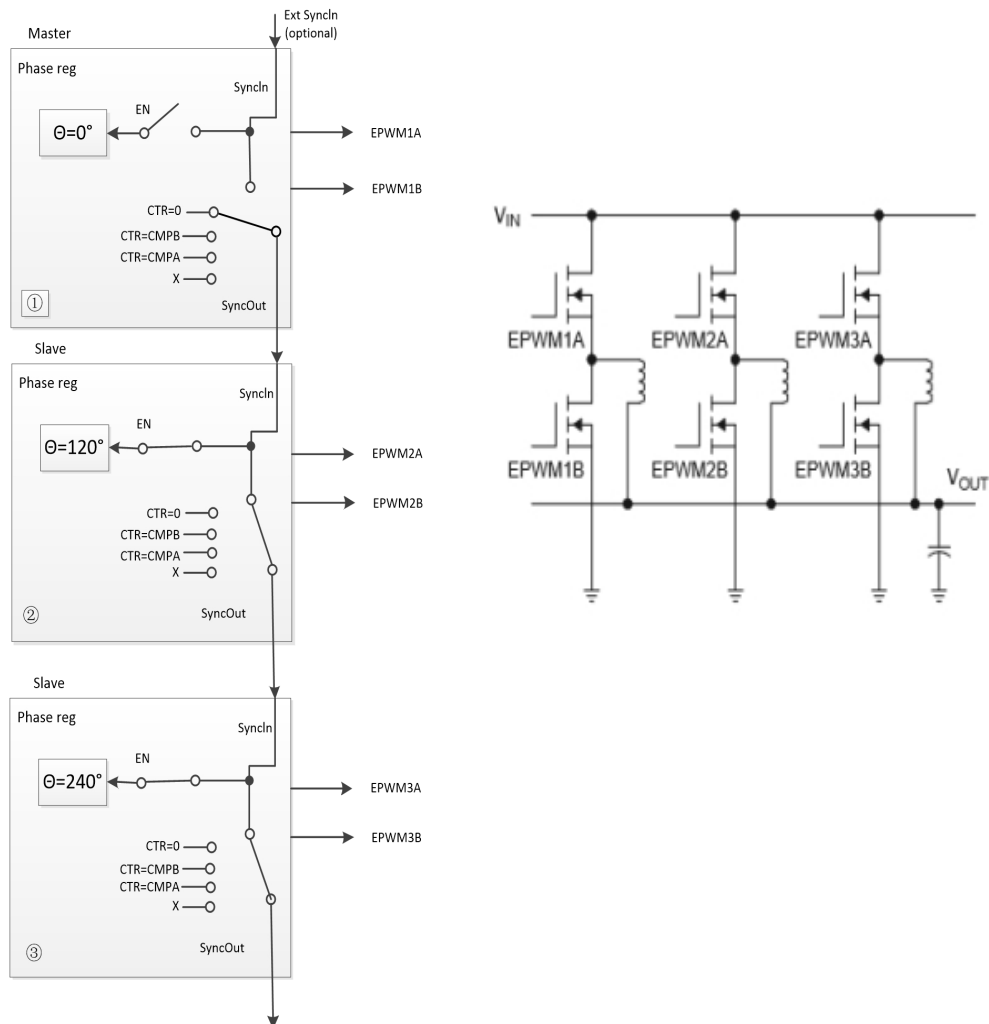


图 3-13 三相交错 DC/DC 交换电路的控制

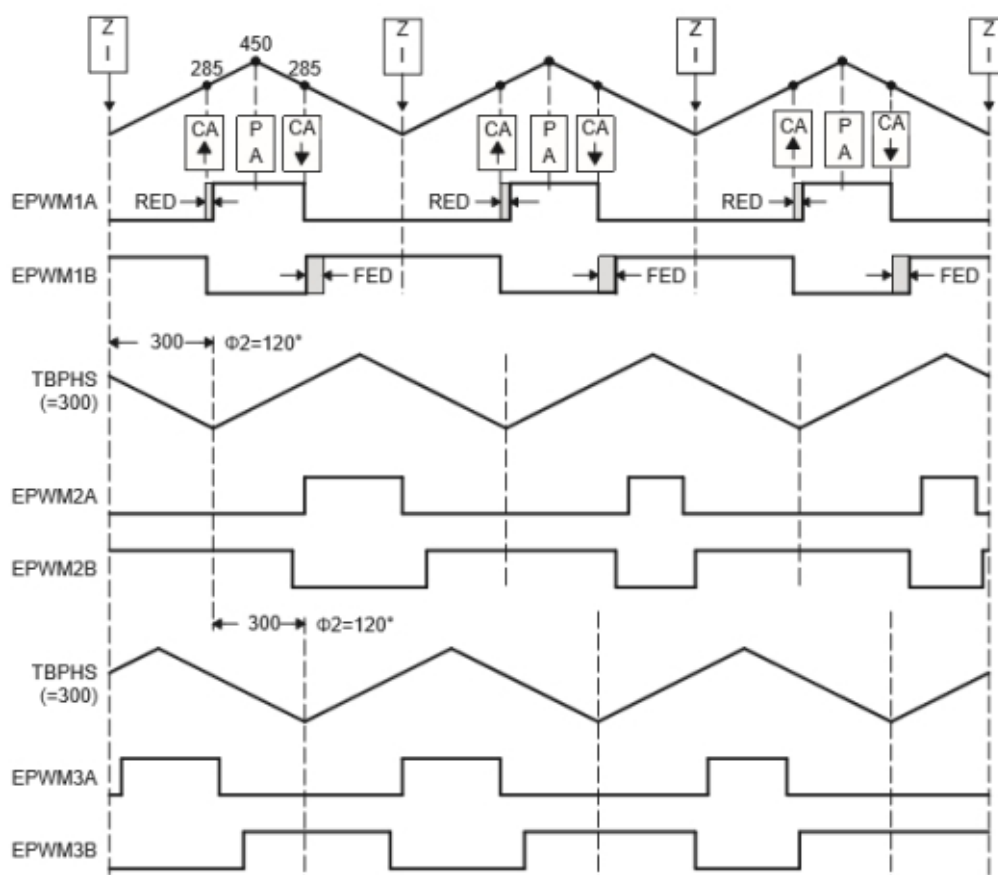


图 3-14 图 3-13 的三相交错 DC/DC 交换电路的波形图

3.1.7 控制 0 电压转换的全桥转换电路 (ZVSFB)

图 3-15 假设了一种在各模块之间静态的或恒定不变的相位关系。在这种情况下，可以通过调制占空比在实现控制。也有可能在循环的基础中动态地改变相位值。这个特性使得模块本身可以像相移全桥或 0 电压转换全桥一样控制功率级拓扑。在这里控制的参数不是占空比（占空比保持不变或保持大约为 50%），而是个支路之间的相位关系。通过两个 PWM 模块的资源配置控制单功率级，就可以实现这样的一个系统，这就要求控制 4 个开关元件。图 3-16 显示一个同步的 master/slave 组合共同控制一个 H 全桥。在这种情况下，master 和 slave 都被要求在相同的 PWM 频率下进行转换。可以通过 slave 的相位寄存器 (TBPHS) 来控制相位。不使用 master 的相位寄存器，因此 master 的相位寄存器应初始化为 0。

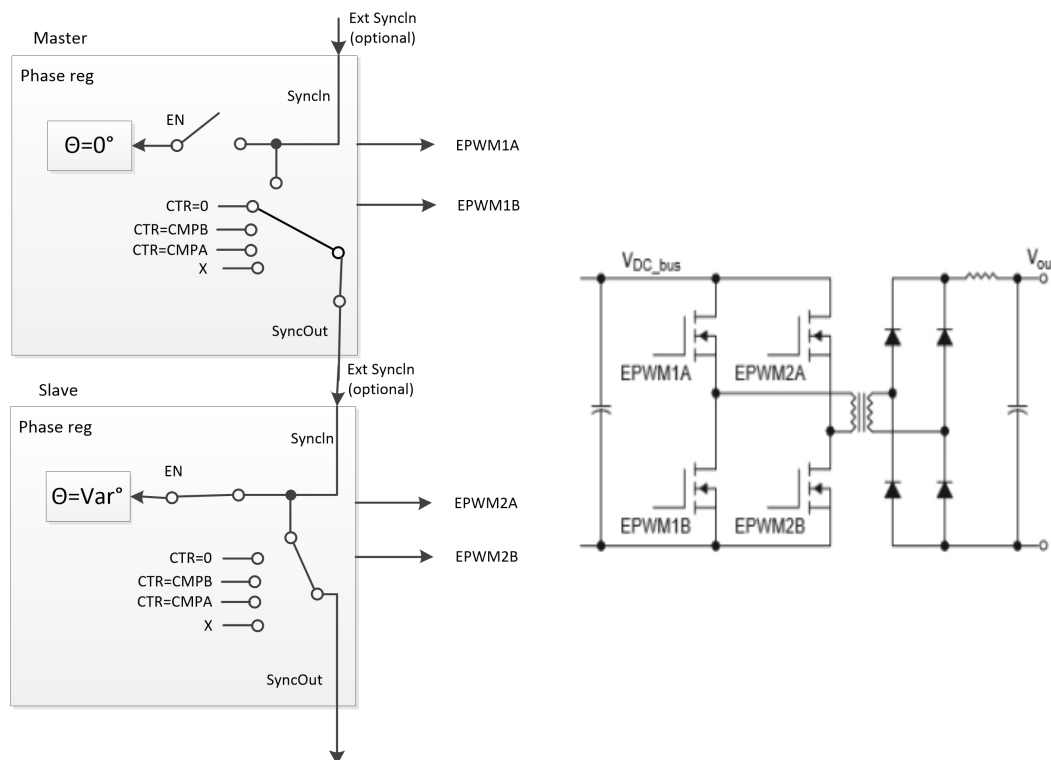


图 3-15 H 全桥的控制（EPWM2 = EPWM1）

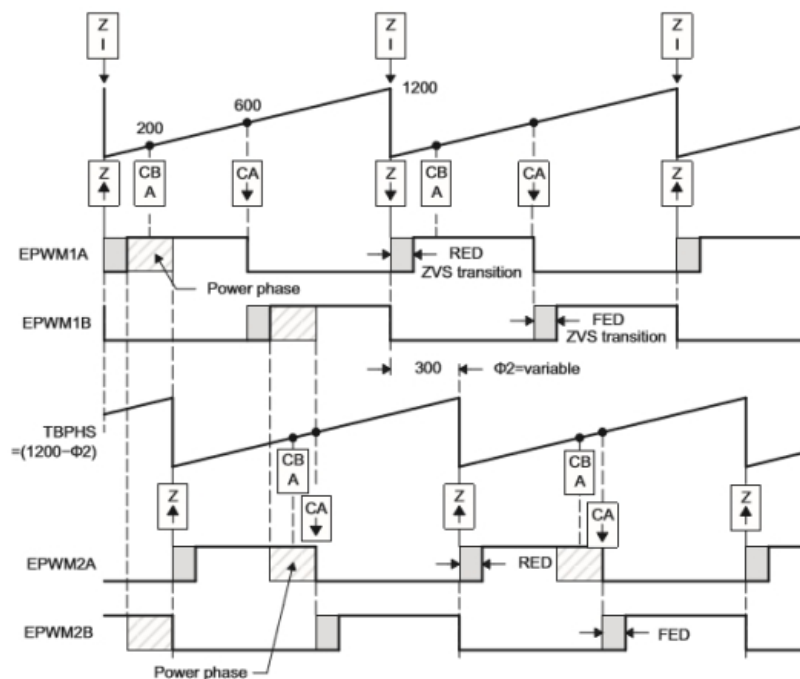


图 3-16 ZVS H 全桥波形图

3.1.8 控制峰值电流模式（Peak Current Mode）控制 Buck 模块

峰值电流模式控制技术提供了自动过流限流、输入电压变化的快速校正以及还原磁饱和等优点。图 3-17 显示了 EPWM1A 和片上比较器在 Buck 变换器拓扑上的应用。输出电流可以通过电流感应电阻感应并转到片上比较器的正极。内部可编程 10 位 DAC 提供一个参

3 EPWM

考峰值电流给比较器的负极。作为选择，一个外部参考电流也连到该输入端。比较器的输出是数字比较器子模块的输入。EPWM 模块这么配置，是为了当感应电流达到峰值参考电流时立即触发 EPWM 的输出。采用了一种周期循环的机制。图 3-18 显示了这种配置下产生的波形。

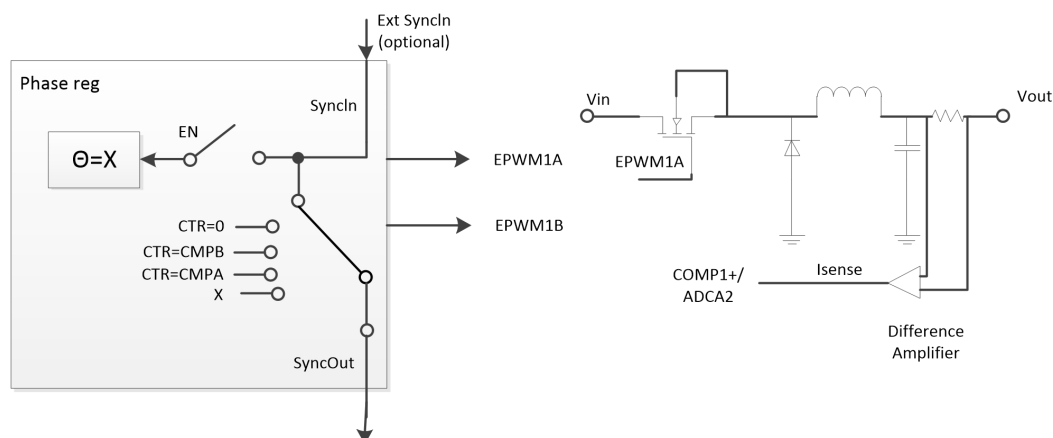


图 3-17 一个 Buck 转换电路的峰值电流模式控制

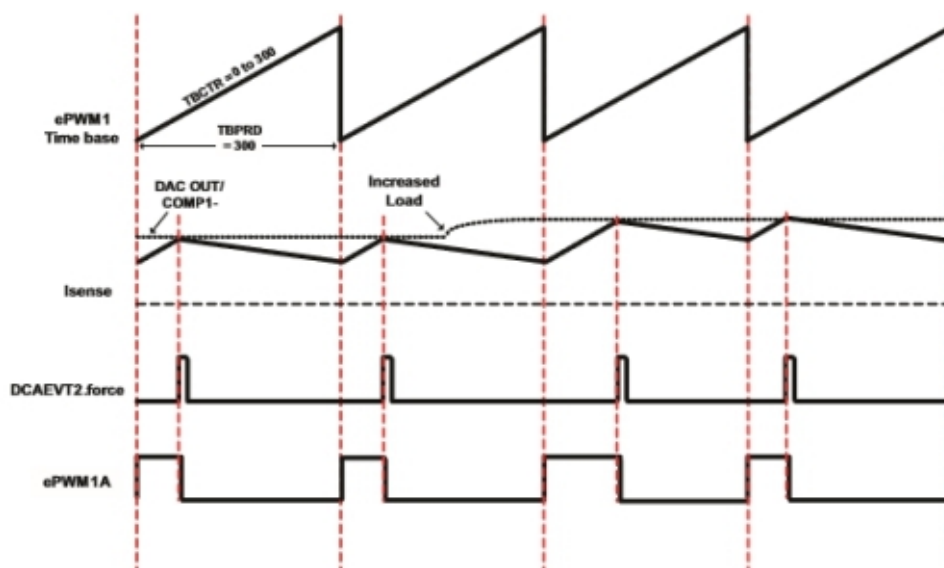


图 3-18 图 3-17 的峰值电流模式控制波形图

3.1.9 控制 H 桥 LLC 谐振变换器

多年来，多种谐振变换器的拓扑在电力电子领域都很有名。另外，近来在要求高效和高功率密度的消费级电子应用方面，H 桥 LLC 谐振变换器拓扑十分受欢迎。例如，EPWM1 的单通道配置十分详细，但其配置可以简单扩充为多通道。控制参数是频率而不是占空比（占空比应保持不变或保持在大约 50%）。虽然无法控制死区时间并且保持 300ns 不变（30@100MHz TBCLK），但是用户需要实时更新死区时间，通过调整足够的软交换时延来提高效率。

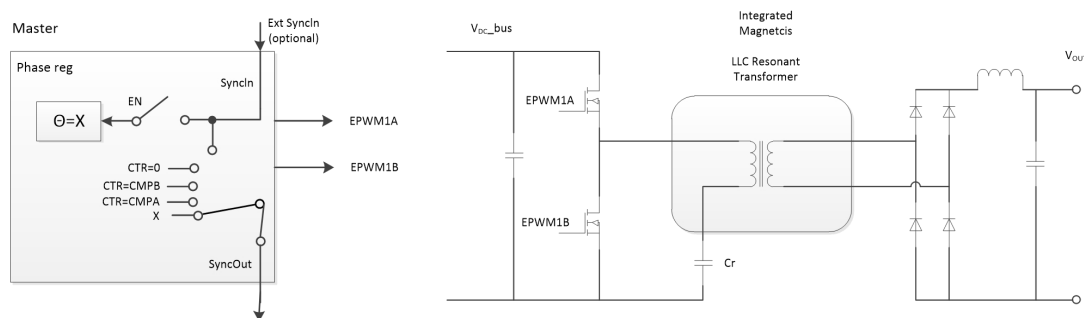


图 3-19 控制两个谐振变换电流

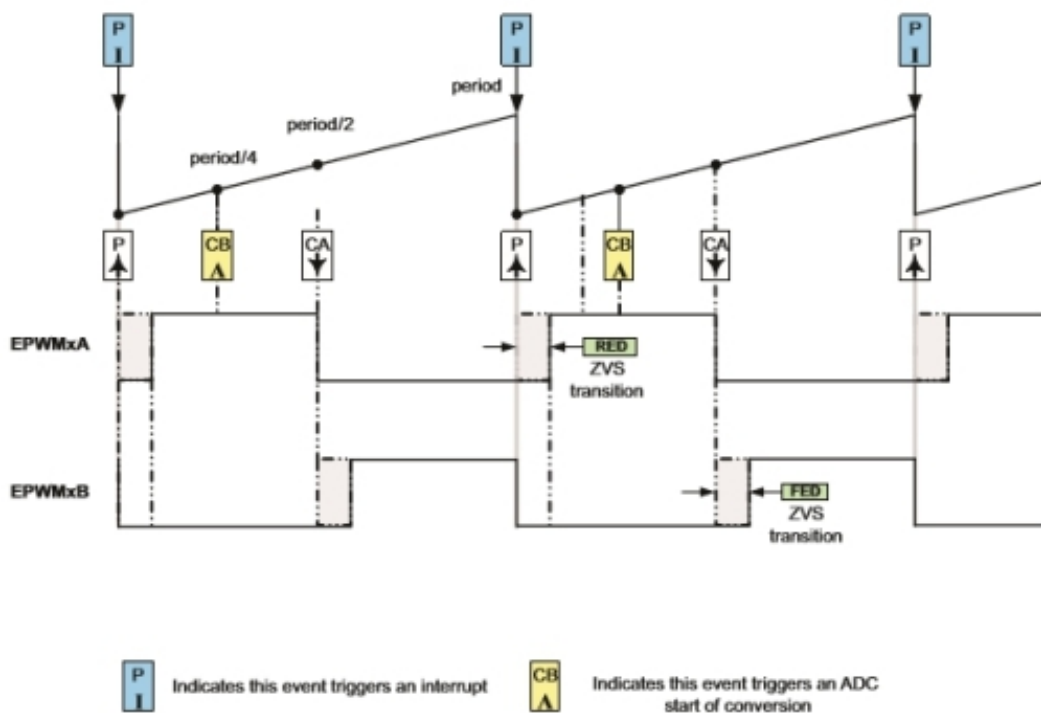


图 3-20 H 桥 LLC 谐振变换 PWM 波形图

3.2 子模块介绍

3.2.1 时基计数器子模块 (Time-base)

时基计数器子模块有如下配置选择：

1. 标定与系统时钟有关的时基计数器时钟；
2. 配置 PWM 时基计数器的频率和周期；
3. 设置时基计数器的模式：
 - 递增计数
 - 递减计数
 - 增减计数
4. 配置与另一 EPWM 模块的时基相位关系；
5. 通过软件或硬件同步不同模块之间的时基计数器；
6. 在同步事件发生后配置时基计数器的方向（递增、递减或增减计数）；

3 EPWM

7. 指定 EPWM 模块的同步输出信号源：

- 同步输入信号
- 时基计数器等于 0
- 时基计数器等于 CMPA
- 时基计数器等于 CMPB
- 没有输出同步信号产生

3.2.2 时基计数比较器子模块（Counter-comparator）

1. 指定 EPWMxA 和 EPWMxB 上输出的 PWM 波的占空比；
2. 指定 EPWMxA 和 EPWMxB 上开关转换的时间；
3. 产生事件触发信号

3.2.3 动作产生子模块（Action-qualifier）

动作产生子模块可以将事件设置转换为各种动作类型，从而在 EPWMxA 和 EPWMxB 输出需要的波形。

1. 动作产生子模块具有以下作用：产生动作（置位、清除、反转）；
2. 通过软件强制控制 PWM 输出状态；
3. 软件控制和配置 PWM 的死区；

3.2.4 死区产生子模块（Dead-band）

1. 控制开关的传统互补死区关系；
2. 指定输出上升沿延时值；
3. 指定输出下降沿延时值；
4. 忽略死区模块，这种情况下 PWM 波形无任何变化地通过；

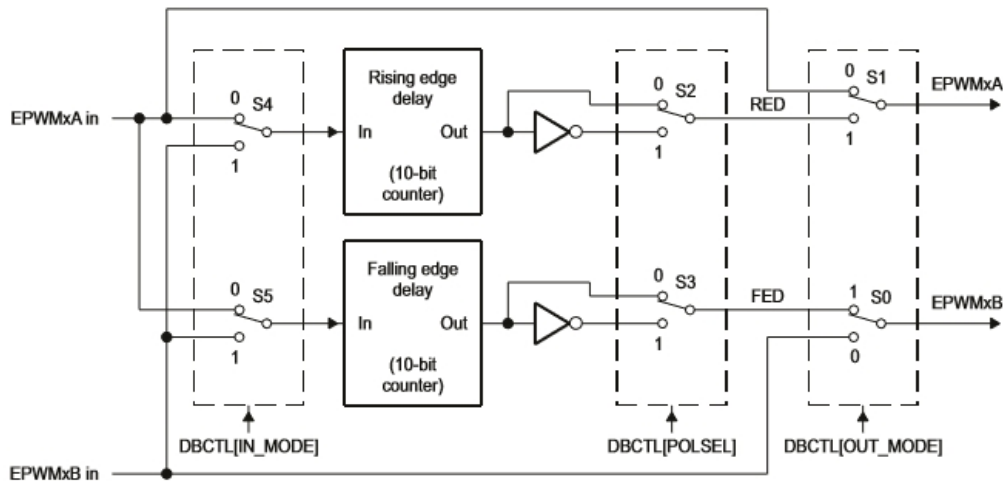


图 3-21 死区子模块的选择配置

3.2.5 事件触发子模块（Event-trigger）

事件触发子模块主要功能如下：

1. 接收由时基计数器或时基计数比较器子模块产生的事件输入；
2. 通过时基计数器方向限定事件 up/down；
3. 在下列情况下使用计数逻辑产生中断请求和启动 adc 转换：
 - 每次事件
 - 每两次事件
 - 每三次事件
4. 通过事件计数器和标志提供事件产生的可视化；

3 EPWM

5. 允许软件强制产生中断和启动 ADC 转换;
6. 当事件产生时事件触发子模块管理时基子模块的事件的产生和计数比较子模块产生 CPU 中断, 产生 ADC 开始转换的脉冲;

3.2.6 故障区子模块 (Trip-zone)

Trip-Zone 子模块的主要功能如下:

1. 故障区输入信号 TZ1 到 TZ4 可以灵活地映射到任何一个 EPWM 模块;
2. 根据故障情况, EPWMxA 和 EPWMxB 输出可以被强制输出为下列信号:
 - 强制输出高电平
 - 强制输出低电平
 - 强制输出高阻抗
 - 没有动作发生
3. 支持在短路或过流情况下的单次 trip;
4. 支持周期循环 (CBC) 限流操作;
5. 每个 Trip-zone 输入引脚都可以被分派为单次或周期循环的操作;
6. 在任何一个 Trip-zone 引脚上都有可能发生中断;
7. 支持软件强制 trip;
8. 如果不需要, 可以忽略 Trip-zone 子模块;

3.3 寄存器介绍

3-1 EPWM_TTCTL

地址偏移: 0x00

Bit	名称	复位值	读/写	功能
31	EPWM6_TZINT	0	R	EPWM6 的故障区 (Trip zone) 的 INT PENDING
30	EPWM5_TZINT	0	R	EPWM5 的故障区 (Trip zone) 的 INT PENDING
29	EPWM4_TZINT	0	R	EPWM4 的故障区 (Trip zone) 的 INT PENDING
28	EPWM3_TZINT	0	R	EPWM3 的故障区 (Trip zone) 的 INT PENDING
27	EPWM2_TZINT	0	R	EPWM2 的故障区 (Trip zone) 的 INT PENDING
26	EPWM1_TZINT	0	R	EPWM1 的故障区 (Trip zone) 的 INT PENDING
25	EPWM0_TZINT	0	R	EPWM0 的故障区 (Trip zone) 的 INT PENDING
24	EPWM6_ETINT	0	R	EPWM6 事件触发 INT PENDING
23	EPWM5_ETINT	0	R	EPWM5 事件触发 INT PENDING
22	EPWM4_ETINT	0	R	EPWM4 事件触发 INT PENDING
21	EPWM3_ETINT	0	R	EPWM3 事件触发 INT PENDING
20	EPWM2_ETINT	0	R	EPWM2 事件触发 INT PENDING
19	EPWM1_ETINT	0	R	EPWM1 事件触发 INT PENDING
18	EPWM0_ETINT	0	R	EPWM0 事件触发 INT PENDING

3 EPWM

17:11	保留	0	R	-
10	CPUTZ	0	RW	0: 强制 tz6_输出为 1 1: 强制 tz6_输出为 0
9:7	保留	0	RW	-
6	EPWM6_EN	0	RW	EPWM6 的使能位。 0: 不使能 1: 使能, 必须先完成所有的配置
5	EPWM5_EN	0	RW	EPWM5 的使能位。 0: 不使能 1: 使能, 必须先完成所有的配置
4	EPWM4_EN	0	RW	EPWM4 的使能位。 0: 不使能 1: 使能, 必须先完成所有的配置
3	EPWM3_EN	0	RW	EPWM3 的使能位。 0: 不使能 1: 使能, 必须先完成所有的配置
2	EPWM2_EN	0	RW	EPWM2 的使能位。 0: 不使能 1: 使能, 必须先完成所有的配置
1	EPWM1_EN	0	RW	EPWM1 的使能位。 0: 不使能 1: 使能, 必须先完成所有的配置
0	EPWM0_EN	0	RW	EPWM0 的使能位。 0: 不使能 1: 使能, 必须先完成所有的配置

3-2 EPWM_ADCSEL0

地址偏移: 0x250

Bit	名称	复位值	读/写	功能
31:30	保留	00	R	-
29:27	ADC9_SEL	000	RW	启动 ADC9 采样的 EPWM SOC 信号选择位。 000: 不启动 ADC9 采样 001: epwm0_socb 010: epwm1_socb

3 EPWM

				011: epwm2_socb 100: epwm3_socb 101: epwm4_socb 110: epwm5_socb 111: epwm6_socb
26:24	ADC8_SEL	000	RW	启动 ADC8 采样的 EPWM SOC 信号选择位。 000: 不启动 ADC8 采样 001: epwm0_socb 010: epwm1_socb 011: epwm2_socb 100: epwm3_socb 101: epwm4_socb 110: epwm5_socb 111: epwm6_socb
23:21	ADC7_SEL	000	RW	启动 ADC7 采样的 EPWM SOC 信号选择位。 000: 不启动 ADC7 采样 001: epwm0_socb 010: epwm1_socb 011: epwm2_socb 100: epwm3_socb 101: epwm4_socb 110: epwm5_socb 111: epwm6_socb
20:18	ADC6_SEL	000	RW	启动 ADC6 采样的 EPWM SOC 信号选择位。 000: 不启动 ADC6 采样 001: epwm0_soca 010: epwm1_soca 011: epwm2_soca 100: epwm3_soca 101: epwm4_soca 110: epwm5_soca 111: epwm6_soca
17:15	ADC5_SEL	000	RW	启动 ADC5 采样的 EPWM SOC 信号选择位。 000: 不启动 ADC5 采样 001: epwm0_soca

3 EPWM

				010: epwm1_soca 011: epwm2_soca 100: epwm3_soca 101: epwm4_soca 110: epwm5_soca 111: epwm6_soca
14:12	ADC4_SEL	000	RW	启动 ADC4 采样的 EPWM SOC 信号选择位。 000: 不启动 ADC4 采样 001: epwm0_soca 010: epwm1_soca 011: epwm2_soca 100: epwm3_soca 101: epwm4_soca 110: epwm5_soca 111: epwm6_soca
11:9	ADC3_SEL	000	RW	启动 ADC3 采样的 EPWM SOC 信号选择位。 000: 不启动 ADC3 001: epwm0_soca 010: epwm1_soca 011: epwm2_soca 100: epwm3_soca 101: epwm4_soca 110: epwm5_soca 111: epwm6_soca
8:6	ADC2_SEL	000	RW	启动 ADC2 采样的 EPWM SOC 信号选择位。 000: 不启动 ADC2 采样 001: epwm0_soca 010: epwm1_soca 011: epwm2_soca 100: epwm3_soca 101: epwm4_soca 110: epwm5_soca 111: epwm6_soca
5:3	ADC1_SEL	000	RW	启动 ADC1 采样的 EPWM SOC 信号的选择位。 000: 不启动 ADC1 采样

3 EPWM

				001: epwm0_soca 010: epwm1_soca 011: epwm2_soca 100: epwm3_soca 101: epwm4_soca 110: epwm5_soca 111: epwm6_soca
2:0	ADC0_SEL	000	RW	启动 ADC0 采样的 EPWM SOC 信号选择位。 000: 不启动 ADC0 采样 001: epwm0_soca 010: epwm1_soca 011: epwm2_soca 100: epwm3_soca 101: epwm4_soca 110: epwm5_soca 111: epwm6_soca

3-3 EPWM_ADCSEL1

地址偏移: 0x254

Bit	名称	复位值	读/写	功能
31:12	保留	20'b0	R	-
11:9	ADC13_SEL	000	RW	启动 ADC13 采样的 EPWM SOC 信号选择位。 000: 不启动 ADC13 采样 001: epwm0_socb 010: epwm1_socb 011: epwm2_socb 100: epwm3_socb 101: epwm4_socb 110: epwm5_socb 111: epwm6_socb
8:6	ADC12_SEL	000	RW	启动 ADC12 采样的 EPWM SOC 信号选择位。 000: 不启动 ADC12 采样 001: epwm0_socb 010: epwm1_socb 011: epwm2_socb 100: epwm3_socb

3 EPWM

				101: epwm4_socb 110: epwm5_socb 111: epwm6_socb
5:3	ADC11_SEL	000	RW	启动 ADC11 采样的 EPWM SOC 信号选择位。 000: 不启动 ADC11 采样 001: epwm0_socb 010: epwm1_socb 011: epwm2_socb 100: epwm3_socb 101: epwm4_socb 110: epwm5_socb 111: epwm6_socb
2:0	ADC10_SEL	000	RW	启动 ADC10 采样的 EPWM SOC 信号选择位。 000: 不启动 ADC10 采样 001: epwm0_socb 010: epwm1_socb 011: epwm2_socb 100: epwm3_socb 101: epwm4_socb 110: epwm5_socb 111: epwm6_socb

3-4 EPWMx_TBCTL

地址偏移: 0x00/0x54/0xa8/0xfc/0x150/0x1a4/0x1f8

Bit	名称	复位值	读/写	功能
31:16	TBCTR	16'b0	R	时基计数器当前值读取位。必须给 READ_TBCTR 写 1 并等待 READ_TBCTR 变为 0 才有效。
15:11	保留	5'b0	R	-
10	READ_TBCTR	0	RW	TBCTR 读操作启动控制位。 写 0: 不启动 TBCTR 的读操作 写 1: 启动 TBCTR 的读操作, 当 TBCTR 正在进行读操作, READ_TBCTR 保持为 1; 当 TBCTR 完成读操作, 此位自动清 0
9	TBCTR_DIR	0	R	时基计数器方向状态位。复位时计数器被冻结, 因此该位无意义。为了让该位有意义, 需要先将 CTRMODE (EPWMx_TBCTL[2:1]) 设置为适

3 EPWM

				当的模式。 0: 当前时基计数器递减计数 1: 当前时基计数器递增计数
8	SYNCI	0	RW	输入同步锁存状态位。 0: 无任何效果, 当该位为 0 说明没有外部同步事件发生 1: 该位为 1 说明有外部同步事件发生 (EPWMxSYNCI)。向该位写 1 将清除锁存事件。
7:5	SYNCO_SEL	3'd0	RW	SyncOut 信号选择控制位。 000: SyncIn 001: CTR = 0, 时基计数器的值为 0 010: CTR = CMPA, 时基计数器等于 CMPA 的值 011: CTR = CMPB, 时基计数器等于 CMPB 的值 1xx: SyncOut 输出 0
4	SWF_SYNC	0	W	软件同步控制位。 0: 无任何操作 1: 强制发生软件同步
3	PHSEN	0	RW	相位寄存器的相位同步使能位。 0: 同步脉冲发生时, 时基计数器 (EPWMx_TBCTR) 不加载相位寄存器的值 1: 同步脉冲发生时, 时基计数器 (EPWMx_TBCTR) 加载相位寄存器的值
2:1	CTRMODE	00	RW	时基计数器模式选择位。时基计数器的模式一旦选定就不能再更改, 如果更改了计数器的模式, 更改将在下一个时基计数器时钟周期来临时生效。 00: 递增模式, 时基计数器的值将不断增加至周期指 (TBPRD), 然后复位为 0 01: 递减模式, 时基计数器将从周期值开始递减, 直到 0 再回周期值 10: 增-减模式, 时基计数器将从 0 开始递增至周期值, 再递减至 0 11: 时基计数器不发生变化
0	PRDLD	0	RW	TBPRD (EPWMx_TBPRD_SD[15:0]) 从影子

3 EPWM

				寄存器（Shadow Register）处加载数据的控制位。 0: 当时基计数器为 0 时 TBPRD（激活寄存器）从影子寄存器处加载数据 1: TBPRD（激活寄存器）可直接更新，不经过影子寄存器，对 TBPRD 的读写操作直接影响激活寄存器，立即发生改变
--	--	--	--	---

3-5 EPWMx_TBPRD_SD（保留）

地址偏移：0x04/0x58/0xac/0x100/0x154/0x1a8/0x1fc

3-6 EPWMx_TBPHASE

地址偏移：0x08/0x5c/0xb0/0x104/0x158/0x1ac/0x200

Bit	名称	复位值	读/写	功能
31:17	保留	15'b0	RW	-
16	TBDIR	0	RW	相对于提供同步输入信号的时基，所选的 EPWM 的时基计数器的方向选择位。 当 PHSEN（EPWMx_TBCTL[3]）= 0 时，如果忽略同步事件，时基计数器不加载方向 当 PHSEN（EPWMx_TBCTL[3]）= 1 时，时基计数器加载该方向。同步事件由输入同步信号（EPWMx_SYNCI）激励产生，也可以由软件强制产生 注意：该位仅当增-减模式下有效。
15:0	TBPHS	16'b0	RW	相对于提供同步输入信号的时基，所选的 EPWM 的时基计数器的相位选择位。 ➤ 当 PHSEN（EPWMx_TBCTL[3]）= 0 时，同步事件会被忽略，时基计数器不加载相位 ➤ 当 PHSEN（EPWMx_TBCTL[3]）= 1 时，当同步事件发生时时基计数器加载相位。同步事件由输入同步信号（EPWMx_SYNCI）激励产生，也可以由软件强制产生

3-7 EPWMx_CMPCTL

地址偏移：0x0c/0x60/0xb4/0x108/0x15c/0x1b0/0x204

Bit	名称	复位值	读/写	功能
31:12	保留	20'b0	RW	-
11	SHDWCFULL	0	R	时基计数比较器 C 影子寄存器已满的状态标志位。每发生一次 load-strobe，该位就自动清 0。 0: 比较器 C 的影子 FIFO 还未满

				1: 比较器 C 的影子 FIFO 已满, 执行 CPU 写操作时会覆盖当前影子 FIFO 的值
10	SHDWCMODE	0	RW	时基计数比较器 C 寄存器工作模式选择位。 0: 影子模式 (shadow mode), 运行双 buffer。通所有的写操作都会通过 CPU 访问影子寄存器 1: 立即模式 (immediate mode), 只使用了比较器 C 激活寄存器。所有的读写操作都可以直接访问比较器 C 激活寄存器本身执行比较操作
9:8	LOADCMODE	0	RW	时基计数比较器 C 的激活比较寄存器从时基计数比较器 C 的影子寄存器加载数据的模式选择位。当在立即模式 (EPWMx_CMPCTL[10] = 1) 下, 本位无效。 00: 当时基计数器等于 0 时加载数据 (TBCTR = 0x0000) 01: 当时基计数器等于周期值时加载数据 (TBCTR = TBPRD) 10: 当时基计数器等于 0 或等于周期值时加载数据 (TBCTR = 0x0000 或 TBCTR = TBPRD) 11: 时基计数器被冻结, 不可能进行数据加载
7	SHDWBFULL	0	R	时基计数比较器 B 影子寄存器已满的状态标志位, 当 load-strobe 发生时本位自动清 0。 0: 比较器 B 影子 FIFO 未滿 1: 比较器 B 影子 FIFO 已滿; 执行 CPU 写操作将覆盖当前影子 FIFO 的值
6	SHDWAFULL	0	R	时基计数比较器 A 影子寄存器已满的状态标志位, 每发生一次 load-strobe, 该标志位会自动清 0。 0: CMPA 影子 FIFO 未滿 1: CMPA 影子 FIFO 已滿, 执行 CPU 写操作将覆盖当前影子 FIFO 的值
5	SHDWBMODE	0	RW	时基计数比较器 B 寄存器工作模式选择位。 0: 影子模式 (shadow mode), 运行双 buffer。通所有的写操作都会通过 CPU 访问影子寄存器 1: 立即模式 (immediate mode), 只使用了比较器 B 激活寄存器。所有的读写操作都可以直接访问比较器 B 激活寄存器本身执行比较操作
4	SHDWAMODE	0	RW	时基计数比较器 A 寄存器工作模式选择位。 0: 影子模式 (shadow mode), 运行双 buffer。

3 EPWM

				所有的写操作都会通过 CPU 访问影子寄存器 1: 立即模式 (immediate mode), 仅使用比较 A 激活寄存器。所有的读写操作都可以直接访问比较器 A 激活寄存器本身立即进行比较
3:2	LOADBMODE	00	RW	时基计数比较器 B 的激活比较器从时基计数比较器 B 的影子寄存器加载数据的选择位。当 CMPB 工作在立即模式下 (EPWNx_CMPCTL[5] = 1) 时, 本位无效。 00: 时基计数器等于 0 时加载数据 (TBCTR = 0x0000) 01: 当时基计数器等于周期值时加载数据 (TBCTR = TBPRD) 10: 当时基计数器等于 0 或等于周期值时加载数据 (TBCTR = 0x0000 或 TBCTR = TBPRD), 即在一个周期内加载两次数据 11 时基计数器被冻结, 无法加载数据
1:0	LOADAMODE	00	RW	时基计数比较器 A 的比较器从时基计数比较器 A 的影子寄存器加载数据的选择位。当 CMPA 工作在立即模式 (EPWMx_CMPCTL[4] = 1) 下时, 本位无效。 00: 当时基计数器等于 0 时加载数据 (TBCTR = 0x0000) 01: 当时基计数器等于周期值时加载数据 (TBCTR = TBPRD) 10: 当时基计数器等于 0 或等于周期值时加载数据 (TBCTR = 0x0000 或 TBCTR = TBPRD), 即一个周期内加载两次数据 11: 时基计数器被冻结, 无法加载数据

3-8 EPWMx_CMPA_SD

地址偏移: 0x10/0x64/0xb8/0x10c/0x160/0x1b4/0x208

Bit	名称	复位值	读/写	功能
31:16	保留	16'b0	RW	-
15:0	CMPA	16'b0	RW	时基计数比较器 A, 默认写入此寄存器的数据保存在影子寄存器中。 由 EPWMx_CMPCTL[4] 决定影子寄存器的打开和关闭。 ➤ 当 SHDWAMODE (EPWMx_CMPCTL[4]) 为 0 时, 使能影子寄存器。对本寄存器的数

3 EPWM

				据写入将保存在影子寄存器中。 LOADAMODE (EPWMx_CMPCTL[1:0]) 位决定影子寄存器中哪一个事件会被加载进激活寄存器中。 ➤ 当 SHDWAMODE (EPWMx_CMPCTL[4]) 为 1 时, 关闭影子寄存器。写入的数据将直接进入激活寄存器, 即寄存器主动控制硬件 注意: 在执行写操作之前, 会读取 SHDWAFULL (EPWMx_CMPCTL[6]) 位的值判断影子 FIFO 是否已满。
--	--	--	--	--

3-9 EPWMx_CMPB_SD

地址偏移: 0x14/0x68/0xbc/0x110/0x164/0x1b8/0x20c

Bit	名称	复位值	读/写	功能
31:16	保留	16'b0	RW	-
15:0	CMPB	16'b0	RW	时基计数比较器 B, 默认写入的数据保存在影子寄存器中。 由 SHDWBMODE (EPWMx_CMPCTL[5]) 决定影子寄存器的打开与关闭 ➤ 当为 0 时, 使能影子寄存器。写入的数据会自动保存在影子寄存器中。LOADBMODE (EPWMx_CMPCTL[3:2]) 决定了哪一个事件从影子寄存器加载进激活寄存器中 ➤ 当 SHDWBMODE (EPWMx_CMPCTL[5]) 为 1 时, 关闭影子寄存器。写入的数据直接保存在激活寄存器中, 由该寄存器主动控制硬件 注意: 执行写操作之前, 先读取 SHDWBFULL (EPWMx_CMPCTL[7]) 的值判断影子 FIFO 是否已满。

3-10 EPWMx_CMPC_SD

地址偏移: 0x18/0x6c/0xc0/0x114/0x168/0x1bc/0x210

Bit	名称	复位值	读/写	功能
31:16	保留	16'b0	RW	-
15:0	CMPC	16'b0	RW	时基计数比较器 C。此特殊功能寄存器 (SFR) 只用于启动 ADC 转换。 默认写入的数据保存在影子寄存器中, 由 SHDWCMODE (EPWMx_CMPCTL[10]) 决定影子寄存器的打开与关闭。

3 EPWM

				➤ 当 SHDWCMODE (EPWMx_CMPCTL[10]) 为 0 时, 使能影子寄存器。写入的数据将自动保存在影子寄存器中。LOADCMODE (EPWMx_CMPCTL[9:8]) 决定哪一个事件从影子寄存器中加载进激活寄存器 ➤ 当 SHDWCMODE (EPWMx_CMPCTL[10]) 为 1 时, 关闭影子寄存器。写入的数据直接保存在激活寄存器中, 由该寄存器主动控制硬件 注意: 执行写操作之前, 可以先读取 SHDWCFULL (EPWMx_CMPCTL[11]) 的值判断当前影子 FIFO 是否已满。
--	--	--	--	---

3-11 EPWMx_AQCTLAB

地址偏移: 0x1C/0x70/0xC4/0x118/0x16C/0x1C0/0x214

Bit	名称	复位值	读/写	功能
31:12	保留	20'b0	RW	-
27:26	CBDB	00	RW	当时基计数器递减计数并且等于 CMPB 的激活寄存器时, 产生动作。 00: 不动作 01: 清 0, 强制 EPWMxB 输出低电平 10: 置 1, 强制 EPWMxB 输出高电平 11: 翻转, EPWMxB 输出低电平时会被强制翻转为高电平; 输出高电平时会被强制翻转为低电平
25:24	CBUB	00	RW	当时基计数器递增计数并且等于 CMPB 的激活寄存器时, 产生动作。 00: 不动作 01: 清 0, 强制 EPWMxB 输出低电平 10: 置 1, 强制 EPWMxB 输出高电平 11: 翻转, EPWMxB 输出低电平时会被强制翻转为高电平; 输出高电平时会被强制翻转为低电平
23:22	CADB	00	RW	当时基计数器递减计数且等于 CMPA 的激活寄存器时, 产生动作。 00: 不动作 01: 清 0, 强制 EPWMxB 输出低电平 10: 置 1, 强制 EPWMxB 输出高电平

3 EPWM

				11: 翻转, EPWMxB 输出低电平时会被强制翻转为高电平; 输出高电平时会被强制翻转为低电平
21:20	CAUB	00	RW	当时基计数器递增计数且等于 CMPA 的激活寄存器时, 产生动作。 00: 不动作 01: 清 0, 强制 EPWMxB 输出低电平 10: 置 1, 强制 EPWMxB 输出高电平 11: 翻转, EPWMxB 输出低电平时会被强制翻转为高电平; 输出高电平时会被强制翻转为低电平
19:18	PRDB	00	RW	当时基计数器等于周期值时 (TBCTR = TBPRD), 产生动作。 00: 不动作 01: 清 0, 强制 EPWMxB 输出低电平 10: 置 1, 强制 EPWMxB 输出高电平 11: 翻转, EPWMxB 输出低电平时会被强制翻转为高电平; 输出高电平时会被强制翻转为低电平 注意: 在默认情况下, 当计数器的值等于周期值且在计数器增-减模式下时, 计数器方向定义为 0, 即向下计数。
17:16	ZROB	00	RW	当计数值等于 0 时, 产生动作。 00: 不动作 01: 清 0, 强制 EPWMxB 输出低电平 10: 置 1, 强制 EPWMxB 输出高电平 11: 翻转, EPWMxB 输出低电平时强制翻转为高电平; EPWMxB 输出高电平时强制翻转为低电平 注意: 在默认情况下, 当计数值等于 0 且计数器处于增-减计数模式下, 方向定义为 1, 即向上计数。
15:12	保留	4'b0	RW	-
11:10	CBDA	00	RW	当时基计数器递减计数且等于 CMPB 激活寄存器的值时, 产生动作。 00: 不动作 01: 清 0, 强制 EPWMxA 输出低电平

3 EPWM

				10: 置 1, 强制 EPWMxA 输出高电平 11: 翻转, EPWMxA 说错话低电平强制翻转为高电平; 输出高电平时强制翻转为低电平
9:8	CBUA	00	RW	当时基计数器递增计数且等于 CMPB 激活寄存器的值时, 产生动作。 00: 不动作 01: 清 0, 强制 EPWMxA 输出低电平 10: 置 1, 强制 EPWMxA 输出高电平 11: 翻转, EPWMxA 输出低电平时强制翻转为高电平; 输出高电平时强制翻转为低电平
7:6	CADA	00	RW	当时基计数器递减计数且等于 CMPA 激活寄存器的值时, 产生动作。 00: 不动作 01: 清 0, 强制 EPWMxA 输出低电平 10: 置 1, 强制 EPWMxA 输出高电平 11: 翻转, EPWMxA 输出低电平时强制翻转为高电平; 输出高电平时强制翻转为低电平
5:4	CAUA	00	RW	当时基计数器递增计数且等于 CMPA 激活寄存器的值时, 产生动作。 00: 不动作 01: 清 0, 强制 EPWMxA 输出低电平 10: 置 1, 强制 EPWMxA 输出高电平 11: 翻转, EPWMxA 输出低电平时强制翻转为高电平; 输出高电平时强制翻转为低电平
3:2	PRDA	00	RW	当时基计数器等于周期值时, 产生动作。 注意: 在默认情况下, 当计数值等于周期值且计数器处于增-减计数模式下, 方向定义为 0 或向下计数 00: 不动作 01: 清 0, 强制 EPWMxA 输出低电平 10: 置 1, 强制 EPWMxA 输出高电平 11: 翻转, EPWMxA 输出低电平时强制翻转为高电平; 输出高电平时强制翻转为低电平
1:0	ZROA	00	RW	当时基计数器等于 0 时, 产生动作。 注意: 在默认情况下, 当计数值为 0 且处于增-减计数模式下时, 方向定义为 1 或向上计数。

3 EPWM

				00: 不动作 01: 清 0, 强制 EPWMxA 输出低电平 10: 置 1, 强制 EPWMxA 输出高电平 11: 翻转, EPWMxA 输出低电平时强制翻转为高电平; 输出高电平时强制翻转为低电平
--	--	--	--	--

3-12 EPWMX_AQSFRC

地址偏移: 0x20/0x74/0xC8/0x11C/0x170/0x1C4/0x218

Bit	名称	复位值	读/写	功能
31:12	保留	20'b0	RW	-
7:6	RLDCSF	00	R	AQSFRC 和 AQCSFRC 的激活寄存器从对应的影子寄存器处再次加载数据的时机选择位。 00: 当时基计数器等于 0 时加载数据 (TBCTR = 0x0000) 01: 时基计数器等于周期值时加载数据 (TBCTR = TBPRD) 10: 当时基计数器等于 0 或等于周期值时加载数据 (TBCTR = 0x0000 或 TBPRD, 即一个周期内加载两次数据) 11: 立即加载, 执行读写操作时, 由 CPU 直接访问 AQSFRC 和 AQCSFRC 激活寄存器无需经过其影子寄存器
5	OTSFB	0	RW	输出 B 上的单次软件强制事件使能位。 0: 无任何效果, 读返回 0 1: 激发单个软件强制事件 每激发一次强制事件, 该位会自动清 0。这是一个一次性的强制事件, 可以被输出 B 上的后来发生的事件所覆盖。
4:3	ACTSFB	0	RW	当单次软件强制事件 B 被调用时, EPWMxB 产生动作。 00: 不动作 01: 清 0, 强制输出低电平 10: 置 1, 强制输出高电平 11: 翻转, 将输出的低电平翻转为高电平, 输出的高电平翻转为低电平 注意: 此动作不受计数器的方向限制。
2	OTSFA	0	RW	输出 A 上的单次软件强制事件使能位。 0: 没有任何效果, 读返回 0。

3 EPWM

				1: 激发一个单次软件强制事件 每激发一个单次强制事件, 该位就会自动清 0。
1:0	ACTSFA	0	RW	当单次软件强制事件 A 被调用时, EPWMxA 产生动作。 00: 不动作 01: 清 0, 输出低电平 10: 置 1, 输出高电平 11: 翻转, 将输出的高电平翻转为低电平, 输出的低电平翻转为高电平 注意: 此动作不受计数器方向限制。

3-13 EPWMX_AQCSFRC

地址偏移: 0x24/0x78/0xCC/0x120/0x174/0x1C8/0x21C

Bit	名称	复位值	读/写	功能
31:4	保留	28'b0	RW	-
3:2	CSFB	00	RW	输出 B 上的连续软件强制事件选择位。 在立即模式下, 一个连续强制事件作用于下一个 TBCLK 边沿。 在影子模式下, 通过设置 RLDCSF (EPWMx_AQSFRC[7:6]) 位配置影子模式, 加载最新的配置到激活寄存器后, 一个连续强制事件作用于下一个 TBCLK 边沿。 00: 不使能, 无任何效果 01: 在输出 B 上强制输出持续低电平 10: 在输出 B 上强制输出持续高电平 11: 不使能软件强制, 没有任何效果
1:0	CSFA	00	RW	输出 A 上的连续软件强制事件选择位。 在立即模式下, 一个连续强制事件作用于下一个 TBCLK 边沿。 在影子模式下, 通过设置 RLDCSF (EPWMx_AQSFRC[7:6]) 位配置影子模式, 加载最新的配置到激活寄存器后, 一个连续强制事件作用于下一个 TBCLK 边沿。 00: 不使能强制, 无任何效果 01: 在输出 A 上强制输出持续低电平 10: 在输出 A 上强制输出持续高电平 11: 不使能软件强制, 没有任何效果

3-14 EPWMX_DBCTL

3 EPWM

地址偏移: 0x28/0x7C/0xD0/0x124/0x178/0x1CC/0x220

Bit	名称	复位值	读/写	功能
31:7	保留	25'b0	RW	-
6	HALFCYCLE	0	RW	半/全周期时钟使能位。 0: 使能全周期时钟。每得到一个 TBCLK 的正边沿, 死区计数器就会加 1 1: 使能半周期时钟。每得到一个 TBCLK 正边沿, 死区计数器就会加 2
5:4	IN_MODE	00	RW	死区输入模式控制位。IN_MODE 允许用户为下降沿和上升沿延时选择输入源。为了产生典型的死区波形, 默认由 EPWMxA In 作为上升沿和下降沿延时的来源。 00: EPWMxA In (来自动作产生子模块)是上升沿和下降沿延时的来源 01: EPWMxB In (来自动作产生子模块) 是上升沿延时的信号源; EPWMxA In (来自动作产生子模块)是下降沿延时的信号源 10: EPWMxA In 来自动作产生子模块)是上升沿延时的信号源; EPWMxB In (来自动作产生子模块) 是下降沿延时的信号源 11: EPWMxB In (来自动作产生子模块)是上升沿延时和下降沿的信号源
3:2	POLSEL	00	RW	极性选择控制位。POLSEL 允许用户在延时信号输出到死区子模块之前可以有选择地转换其中一个延时信号。 00: 高电平有效 (AH) 模式, EPWMxA 和 EPWMxB 都不翻转 (默认情况) 01: 激活低电平互补 (ALC) 模式, EPWMxA 翻转后输出 10: 激活高电平互补 (AHC) 模式, 翻转 EPWMxB 翻转后输出 11: 低电平有效 (AL) 模式, EPWMxA 和 EPWMxB 翻转后输出
1:0	OUT_MODE	00	RW	死区输出模式控制位。OUT_MODE 允许用户选择使能或忽略上升沿或下降沿延时的死区生成。 00: 两个输出信号都忽略死区产生。在此模式下, 来自动作产生子模块的 EPWMxA 和 EPWMxB 的输出可以直接传输到 Event-trigger

3 EPWM

				子模块。在此模式下，POLSEL（EPWMx_DBCTL[3:2]）和IN_MODE（EPWMx_DBCTL[5:4]）位都没有任何效果。 01：不使能上升沿延迟，使能下降沿延迟。动作产生子模块的EPWMxA信号可以直接传输到Event-trigger子模块的EPWMxA输入。下降沿延时信号在EPWMxB的输出处可见。延时信号的输出则由IN_MODE（EPWMx_DBCTL[5:4]）所决定。 10：使能上升沿延迟，不使能下降沿延迟。上升沿延时信号在EPWMxA输出处可见。由IN_MODE（EPWMx_DBCTL[5:4]）位决定输入信号的延时；动作产生子模块的EPWMxB信号可以直接传输到Event-trigger子模块的EPWMxB的输入。 11：使能双边（上升沿和下降沿）延迟。EPWMxA输出的上升沿延时和EPWMxBD输出的下降沿延时的死区全使能。由IN_MODE（EPWMx_DBCTL[5:4]）决定输入信号的延时。
--	--	--	--	---

3-15 EPWMX_DBDELAY

地址偏移：0x2C/0x80/0xD4/0x128/0x17C/0x1D0/0x224

Bit	名称	复位值	读/写	功能
31:26	保留	6'b0	RW	-
25:16	FED	10'd0	RW	下降沿延时计数器，10位计数器（下降沿死区）
15:10	保留	6'b0	RW	-
9:0	RED	10'b0	RW	上升沿延时计数器，10位计数器（上升沿死区）

3-16 EPWMx_ETCTL

地址偏移：0x30/0x84/0xD8/0x12C/0x180/0x1D4/0x228

Bit	名称	复位值	读/写	功能
31	保留	0	R	-
30	SOCBEN	0	RW	ADC启动脉冲B（EPWMxSOCB）的使能位。 0：不使能EPWMxSOCB 1：使能EPWMxSOCB脉冲
29:26	SOCBSEL	4'b0	RW	EPWMxSOCB选择控制位。SOCBSEL决定EPWMxSOCB脉冲信号产生的时间。 0000：使能DCBEVT1启动的事件（DCBEVT1 SOC）

				0001: 使能时基计数器等于 0 的事件 (TBCTR = 0x0000) 0010: 使能时基计数器等于周期值的事件 (TBCTR = TBPRD) 0011: 使能时基计数器等于 0 或等于周期值的事件 (TBCTR = 0x0000 和 TBCTR = TBPRD), 该模式仅在增减模式下有效 0100: 当定时器增加时, 使能时基计数器等于 CMPA 的事件 0101: 当定时器递减时, 使能时基计数器等于 CMPA 的事件 0110: 当定时器递增时, 使能时基计数器等于 CMPB 的事件 0111: 当定时器递减时, 使能时基计数器等于 CMPB 的事件 1000: 当定时器递增时, 使能时基计数器等于 CMPC 的事件 1001: 当定时器递减时, 使能时基计数器等于 CMPC 的事件
25	SOCAEN	0	RW	ADC 启动脉冲 A (EPWMxSOCA) 的使能位。 0: 不使能 EPWMxSOCA 1: 使能 EPWMxSOCA
24:21	SOCASEL	4'b0	RW	EPWMxSOCA 选择控制位。SOCASEL 决定 EPWMxSOCA 脉冲产生的时间。 0000: 使能 DCAEVT1 启动的事件 (DCAEVT1 SOC) 0001: 使能时基计数器等于 0 的事件 (TBCTR = 0x0000) 0010: 使能时基计数器等于周期值的事件 (TBCTR = TBPRD) 0011: 使能时基计数器等于 0 或等于周期值的事件 (TBCTR = 0x0000 和 TBCTR = TBPRD), 该模式仅在增-减计数的模式下有效 0100: 当定时器递增时, 使能时基计数器等于 CMPA 的事件 0101: 当定时器递减时, 使能时基计数器等于 CMPA 的事件 0110: 当定时器递增时, 使能时基计数器等于

3 EPWM

				CMPB 的事件 0111: 当定时器递减时, 使能时基计数器等于 CMPB 的事件 1000: 当定时器递增时, 使能时基计数器等于 CMPC 的事件 1001: 当定时器递减时, 使能时基计数器等于 CMPC 的事件
20	INTEN	0	RW	EPWM 中断 (EPWMx_INT) 使能位。 0: 不使能 EPWMx_INT 1: 使能 EPWMx_INT
19:16	INTSEL	4'b0	RW	EPWM 中断(EPWMx_INT) 选择控制位。 0000: 保留 0001: 使能时基计数器等于 0 的事件 (TBCTR = 0x0000) 0010: 使能时基计数器等于周期值的事件 (TBCTR = TBPRD) 0011: 使能时基计数器等于 0 或等于周期值的事件 (TBCTR = 0x0000 和 TBCTR = TBPRD) 该模式仅在增减模式下有效 0100: 当定时器递增时, 使能时基计数器等于 CMPA 的事件 0101: 当定时器递减时, 使能时基计数器等于 CMPA 的事件 0110: 当定时器递增时, 使能时基计数器等于 CMPB 的事件 0111: 当定时器递减时, 使能时基计数器等于 CMPB 的事件 1000: 当定时器递增时, 使能时基计数器等于 CMPC 的事件 1001: 当定时器递减时, 使能时基计数器等于 CMPC 的事件
15:14	SOCBCNT	00	R	EPWM 的 ADC 启动事件 B (EPWMxSOCB) 的计数寄存器。SOCBCNT 位表示发生了多少个由 SOCBSSEL (EPWMx_ETCTL[29:26]) 选择的事件。 00: 没有事件发生 01: 发生了 1 个事件 10: 发生了 2 个事件

				11: 发生了 3 个事件
13:12	SOCBPRD	00	RW	EPWM 的 ADC 启动事件 B (EPWMxSOCB) 的周期选择位。SOCBPRD 决定了在产生一个 EPWMxSOCB 脉冲之前需要发生了多少个在 SOCBSEL (EPWMx_ETCTL[29:26]) 选择的事件。为了产生脉冲, SOCBEN (EPWMx_ETCTL[30]) 要设为 1。甚至当 SOCB(EPWMx_ETFLAG[3])等于 1 时, SOCB 脉冲也会产生。一旦产生 SOCB 脉冲, SOCBCNT (EPWMx_[15:14]) 会自动清 0。 00: 不使能 SOCB 事件计数器, 没有 EPWMxSOCB 脉冲产生 01: 在第一个事件内产生 EPWMxSOCB 脉冲: SOCBCNT (EPWMx_ETCTL[15:14]) = 0,1 10: 在第二个事件内产生 EPWMxSOCB 脉冲: SOCBCNT (EPWMx_ETCTL[15:14]) = 1,0 11 在第三个事件内产生 EPWMxSOCB 脉冲: SOCBCNT (EPWMx_ETCTL[15:14]) = 1,1
11:10	SOCACNT	00	R	EPWM 的 ADC 启动事件 A (EPWMxSOCA) 计数器寄存器。本位表示发生了多少个在 SOCASEL (EPWMxETCTL[24:21]) 中选择的事件。 00: 没有事件发生 01: 发生了 1 个事件 10: 发生了 2 个事件 11: 发生了 3 个事件
9:8	SOCAPRD	00	RW	EPWM 的 ADC 启动事件 A (EPWMxSOCA) 的周期选择位。该位决定了在产生一个 EPWMxSOCA 信号之前, 需要发生多少个 SOCASEL (EPWMx_ETCTL[24:21]) 中选择的事件。为了成功产生 EPWMxSOCA 脉冲, 要设置 SOCAEN (EPWMx_ETCTL25[]) = 1。即使当前启动的转换设置了一个状态脉冲 (EPWMx_ETFLAG[SOCA] = 1), 也会产生 SOCA 脉冲信号。一旦产生 SOCA 脉冲信号, SOCACNT (EPWMx_ETCTL[11:10]) 就会自动清 0。 00: 不使能 SOCA 事件计数器。没有 EPWMxSOCA 脉冲信号产生

3 EPWM

				01: 在第一个事件中产生 EPWMxSOCA 脉冲事件: SOCACNT (EPWMx_ETCTL[11:10]) = 01 10: 在第二个事件中产生 EPWMxSOCA 脉冲信号: SOCACNT (EPWMx_ETCTL[11:10]) = 10 11: 在第三个事件中产生 EPWMxSOCA 脉冲信号: SOCACNT (EPWMx_ETCTL[11:10]) = 11
7:4	保留	00	R	-
3:2	INTCNT	00	R	EPWM 中断事件 (EPWMx_INT) 计数器寄存器。本位指示发生了多少个 INTSEL (EPWMx_ETCTL[19:16]) 中选择的事件, 当产生一个中断时, 该位会自动清 0。如果没有中断发生 (EPWMx_ETFLAG[0] = 0) 或者当 ETFLAG[0] = 1 时, 如果达到周期值 (INTCNT = EPWMx_ETCTL[1:0]) 计数器会停止计数。 00: 没有事件发生 01: 发生 1 个事件 10: 发生 2 个事件 11: 发生 3 个事件
1:0	INTPRD	00	RW	EPWM 中断周期选择位。该位决定了在产生中断之前, 需要发生多少个 INTSEL (EPWMx_ETCTL[19:16]) 选择的事件。为了成功打开中断, 必须打开中断使能位 INT (EPWMx_FLAG[0])。如果当前的中断产生了中断标志 INT (EPWMx_FLAG[0]) = 1, 那么等到清除中断标志位 (CLRINT (EPWMx_ETFLAG[4]) 置 1) 后才会再次产生中断。当一个中断正在执行时, 允许挂起另一个中断。一旦产生了中断, INTCNT (EPWMx_ETCTL[3:2]) 会自动清 0。 如果 INTPRD 不为 0 并清除中断标志位 INT (EPWMx_ETFLAG[0]), 则向 INTPRD 写入一个与当前计数器相同的值可以触发中断。 向 INTPRD 写入一个小于当前计数器的值将进入一个未定义的状态。 如果同时发生计数器事件和向 INTPRD 写入一个 0 或非 0 的值时, 该计数器递增计数。

3 EPWM

				00: 不打开中断事件计数器。没有中断发生并且忽略 ETFRC (EPWMxETFLAG[8]) 01: 在第一个事件中产生一个中断, INTCNT = 01 10: 当 INTCNT = 10 时产生中断 (第二个事件) 11: 当 INTCNT = 11 时产生中断 (第三个事件)
--	--	--	--	--

3-17 EPWMx_ETFLAG

地址偏移: 0x34/0x88/0xDC/0x130/0x184/0x1D8/0x22C

Bit	名称	复位值	读/写	功能
31:12	保留	20'b0	R	-
11	FRCSOCB	0	RW	SOCB 强制控制位。如果 SOCB 使能 (EPWMx_ETCTL[30] = 1), 就会产生一个 SOCB 脉冲, 则必须设置 SOCB (EPWMx_ETFLAG[3]) 为 1。 0: 没有任何效果, 读返回 0 1: 在 EPWMxSOCB 处产生脉冲并设置 SOCB (EPWMx_ETFLAG[3]), 本位通常用于测试
10	FRCSOCA	0	RW	SOCA 强制控制位。如果 SOCA 使能 (EPWMx_ETCTL[25] = 1), 就会产生一个 SOCA 脉冲, 则必须设置 SOCA (EPWMx_ETFLAG[2]) 为 1。 0: 没有任何效果, 读返回 0 1: 在 EPWMxSOCA 处产脉冲并设置 SOCA (EPWMx_ETFLAG[2]), 本位通常用于测试
9	保留	0	RW	-
8	FRCINT	0	RW	INT 强制控制位。如果中断使能 (EPWMx_ETCTL[20] = 1), 将产生中断。那么必须设置中断标志 INT (EPWMx_ETFLAG[0]) 为 1。 0: 无任何效果, 读返回 0 1: 在 EPWMxINT 产生一个中断并设置中断标志位 INT (EPWMx_ETFLAG[0])。该位通常用于测试
7	CLRSOCB	0	RW	EPWM 的 ADC 启动事件 B (EPWMxSOCB) 标志清除控制位。 0: 无任何效果, 读返回 0 1: SOCB (EPWMx_FLAG[3]) 标志位清 0

3 EPWM

6	CLRSOCA	0	RW	EPWM 的 ADC 启动 A (EPWMxSOCA) 的标志清除控制位。 0: 无任何效果, 读返回 0 1: 清除 SOCA (EPWMx_ETFLAG[2]) 标志位
5	保留	0	R	-
4	CLRINT	0	RW	EPWM 中断标志位 (EPWMx_INT) 清除控制位。 0: 无任何效果, 读返回 0 1: 清除中断标志位 INT (EPWMx_FLAG[0])
3	SOCB	0	R	EPWM 的 ADC 启动 B (EPWMxSOCB) 的状态标志位。 0: 没有 EPWMxSOCB 发生 1: 指示在 EPWMxSOCB 发生一个启动转换的脉冲。即使标志位 SOCB 被设为 1, EPWMxSOCB 也会持续产生输出
2	SOCA	0	RW	EPWM 的 ADC 启动 A (EPWMxSOCA) 的状态标志位。与 INT (EPWMx_FLAG[0]) 不同, 即使标志位 SOCA 被设为 1, EPWMxSOCA 也会持续产生输出 0: 没有 EPWMxSOCA 发生 1: 指示在 EPWMxSOCA 发生一个启动转换的脉冲。即使标志位 SOCA 被设为 1, EPWMxSOCA 也会持续产生输出
1	保留	0	R	-
0	INT	0	RW	EPWM 的中断状态标志位 (EPWMx_INT)。 0: 没有中断发生 1: 有一个 EPWM 的中断发生 注意: 只有当此标志位清 0 后才会产生下一个标志位。

3-18 EPWMx_DCCTL

地址偏移: 0x38/0x8C/0xE0/0x134/0x188/0x1DC/0x230

Bit	名称	复位值	读/写	功能
31:20	保留	22'b0	R	-
19	AEVT2FRCSYNCSEL	0	RW	DCAEVT2 强制同步信号选择控制位。 0: 选择同步信号源 1: 选择非同步信号源

3 EPWM

18	AEVT2SRCSEL	0	RW	DCAEVT2 源信号选择控制位。 0: 信号源为 DCAEVT2 1: 信号源为 DCEVTFILT
17	AEVT1SYNCE	0	RW	DCAEVT1 同步信号使能位。 0: 不产生.SYNC 同步信号 1: 产生.SYNC 同步信号
16	AEVT1SOCE	0	RW	DCAEVT1 的 SOC 信号使能位。 0: 不产生.SOC 信号 1: 产生.SOC 信号
15	AEVT1FRCSYNCSSEL	0	RW	DCAEVT1 强制同步信号选择控制位。 0: 选择同步信号源 1: 选择非同步信号源
14	AEVT1SRCSEL	0	RW	DCAEVT1 信号源选择位。 0: 信号源为 DCAEVT1 1: 信号源为 DCEVTFILT
13	BEVT2FRCSYNCSSEL	0	RW	DCBEVT2 强制同步信号选择位。 0: 信号源为同步信号 1: 信号源为非同步信号, 在 EPWMxSOCB 产生脉冲并设置 SOCB 标志位 (EPWMx_FLAG[3])。此位仅用于测试
12	BEVT2SRCSEL	0	RW	DCBEVT2 源信号选择控制位。 0: 信号源为 DCBEVT2 1: 信号源为 DCEVTFILT
11	BEVT1SYNCE	0	RW	DCBEVT1 同步信号使能位。 0: 不产生同步信号 1: 产生同步信号
10	BEVT1SOCE	0	RW	DCBEVT1 的 SOC 使能位。 0: 不产生 SOC 信号 1: 产生 SOC 信号
9	BEVT1FRCSYNCSSEL	0	RW	DCBEVT1 强制同步信号选择位。 0: 信号源为同步信号 1: 信号源为非同步信号
8	BEVT1SRCSEL	0	RW	DCBEVT1 源信号选择位。 0: 源信号为 DCBEVT1

3 EPWM

				1: 源信号为 DCEVTFILT
7:6	PULSESEL	00	R	Blanking 和 Capture Alignment 的脉冲选择控制位。 00: 时基计数器等于周期值 (TBCTR = TBPRD) 01: 时基计数器等于 0 (TBCTR = 0x0000) 10: 保留 11: 保留
5	BLANKINV	0	RW	Blanking Window 反转 0: Blanking window 不反转 1: Blanking window 反转
4	BLANKE	0	R	Blanking Window 使能位。 0: 不使能 Blanking window 1: 使能 Blanking window
3:2	FILTSRCSEL	00	RW	滤波块 (Filter Block Signal) 信号源选择位。 00: 信号源为 DCAEVT1 01: 信号源为 DCAEVT2 10: 信号源为 DCBEVT1 11: 信号源为 DCBEVT2
1	SHDWMODE	0	RW	时基计数器 capture 影子模式选择位。 0: 使能影子模式。当 TBCTR = TBPRD 或 TBCTR = 0 时 (由 PULSESEL (EPWMx_DCCTL[7:6]) 决定), DCCAP 激活寄存器从 DCCAP 影子寄存器处复制数据。CPU 读取 DCCAP 时返回的是 DCCAP 影子寄存器的内容。 1: 使能激活模式, 在此模式下, 影子寄存器模式被关闭。CPU 读取 DCCAP 时返回的是 DCCAP 激活寄存器中的值。
0	CAPE	0	RW	时基计数器 capture 功能使能位。 0: 不使能时基计数器的 capture 功能 1: 使能时基计数器的 capture 功能

3-19 EPWMx_DCTRISEL

地址偏移: 0x3C/0x90/0xE4/0x138/0x18C/0x1E0/0x234

Bit	名称	复位值	读/写	功能
31:28	保留	4'b0	R	-
27:25	DCBEVT2	000	RW	数字比较器输出 B 事件 2 的选择位。

				000: 没有事件发生 001: DCBH = 低电平, DCBL = 无效 010: DCBH = 高电平, DCBL = 无效 011: DCBL = 低电平, DCBH = 无效 100: DCBL = 高电平, DCBH = 无效 101: DCBL = 高电平, DCBH = 低电平 110: 保留 111: 保留
24:22	DCBEVT1	000	RW	数字比较器输出 B 事件 1 的选择位。 000: 没有事件发生 001: DCBH = 低电平, DCBL = 无效 010: DCBH = 高电平, DCBL = 无效 011: DCBL = 低电平, DCBH = 无效 100: DCBL = 高电平, DCBH = 无效 101: DCBL = 高电平, DCBH = 低电平 110: 保留 111: 保留
21:19	DCAEVT2	000	R	数字比较器输出 A 事件 2 的选择位。 000: 没有事件发生 001: DCAH = 低电平, DCAL = 无效 010: DCAH = 高电平, DCAL = 无效 011: DCAL = 低电平, DCAH = 无效 100: DCAL = 高电平, DCAH = 无效 101: DCAL = 高电平, DCAH = 低电平 110: 保留 111: 保留
18:16	DCAEVT1	000	RW	数字比较器输出 A 事件 1 的选择位。 000: 没有事件发生 001: DCAH = 低电平, DCAL = 无效 010: DCAH = 高电平, DCAL = 无效 011: DCAL = 低电平, DCAH = 无效 100: DCAL = 高电平, DCAH = 无效 101: DCAL = 高电平, DCAH = 低电平 110: 保留 111: 保留

15:12	DCBLCOMPSEL	4'b0	R	数字比较器 B 低电平 (DCBL) 输入信号选择位, 定义了 DCBL 的源信号。当 TZ 信号用作故障信号时, 作为正常输入来处理并且定义为高电平有效或 低电平有效。 0000: TZ1 输入信号 0001: TZ2 输入信号 0010: TZ3 输入信号 0011: TZ4 输入信号 1000: COMP1OUT 信号 1001: COMP2OUT 信号 1010: COMP3OUT 信号 1011: COMP4OUT 信号 1100: COMP5OUT 信号 注意: 没有显示的值作为保留。
11:8	DCBHCOMPSEL	4'b0	RW	数字比较器 B 高电平 (DCBH) 输入选择位。定义 DCBH 输入的信号源。当 TZ 信号用作故障信号时, 作为正常输入处理并且定义为 active high 或 active low。 0000: TZ1 输入信号 0001: TZ2 输入信号 0010: TZ3 输入信号 0011: TZ4 输入信号 1000: COMP1OUT 信号 1001: COMP2OUT 信号 1010: COMP3OUT 信号 1011: COMP4OUT 信号 1100: COMP5OUT 信号 注意: 没有显示的值作为保留。
7:4	DCALCOMPSEL	4'b0	RW	数字比较 A 低电平 (DCAL) 输入选择位, 定义 DCAL 输入信号的信号源。当 TZ 信号用作故障信号时, 作为正常输入处理并且定义为 active high 或 active low。 0000: TZ1 输入信号 0001: TZ2 输入信号 0010: TZ3 输入信号 0011: TZ4 输入信号

3 EPWM

				1000: COMP1OUT 信号 1001: COMP2OUT 信号 1010: COMP3OUT 信号 1011: COMP4OUT 信号 1100: COMP5OUT 信号 注意: 没有显示的值作为保留。
3:0	DCAHCOMPSEL	4'b0	RW	数字比较 A 高电平 (DCAH) 输入信号选择位, 定义 DCAH 输入的信号源。当 TZ 信号用作故障信号时, 作为正常输入处理并且定义为 active high 或 active low . 0000: TZ1 输入信号 0001: TZ2 输入信号 0010: TZ3 输入信号 0011: TZ4 输入信号 1000: COMP1OUT 信号 1001: COMP2OUT 信号 1010: COMP3OUT 信号 1011: COMP4OUT 信号 1100: COMP5OUT 信号 注意: 没有显示的值作为保留。

3-20 EPWMx_DCCAP

地址偏移: 0x40/0x94/0xE8/0x13C/0x190/0x1E4/0x238

Bit	名称	复位值	读/写	功能
31:16	保留	16'b0	RW	-
15:0	DCCAP	16'b0	RW	数字比较时基计数器 capture 位。设置 CAPE (EPWMx_DCCTL[0]) 为 1, 使能时基计数器 capture 功能。 如果使能时基计数器的 capture , 在 dcevt filter 的输出出现上升沿时记录时基计数器的值。直到时基计数器值等于 0 或周期值时 (在 PULSESEL (EPWMx_DCCTL[7:6]) 设置) 才会产生下一个 capture 。 在 SHDWMODE (EPWMx_DCCTL[1]) 位设置 DCCAP 的影子是否使能, 默认 DCCAP 为影子模式。 ➤ 如果 SHDWMODE (EPWMx_DCCTL[1]) = 0, 使能影子模式。在此模式下, 当 TBCTR

3 EPWM

				= TBPRD 或 TBCTR = 0 时(由 PULSESEL (EPWMx_DCCTL[7:6]) 设置), DCCAP 激活寄存器会将数据复制到 DCCAP 影子寄存器。CPU 读取 DCCAP 时将返回影子寄存器中的值。 ➤ 如果 SHDWMODE (EPWMx_DCCTL[1]) = 1, 关闭影子模式。CPU 读取 DCCAP 时返回 DCCAP 激活寄存器中的值。 注意: DCCAP 激活寄存器和影子寄存器共享同一个内存映射地址。
--	--	--	--	---

3-21 EPWMx_BLANKOFFSET

地址偏移: 0x44/0x98/0xEC/0x140/0x194/0x1E8/0x23C

Bit	名称	复位值	读/写	功能
31:16	保留	16'b0	RW	-
15:0	OFFSET	16'b0	RW	Blanking Window 偏移量设置。此位指定从 blanking window 引用到应用 blanking window 的时基计数器时钟的周期个数。blanking window 引用是由 PULSESEL (EPWMx_DCCTL[7:6]) 位定义的时基计数器等于 0 或周期值。 该偏移寄存器是影子寄存器并且激活寄存器被加载在由 PULSESEL (EPWMx_DCCTL[7:6]) 定义的参考点处。当激活寄存器被加载时, 偏移计数器也被初始化并且开始向下计数。当计数器满时, 应用 blanking window。如果当前 blanking window 是激活模式, 则重启 blanking window 计数器。

3-22 EPWMx_WINWIDTH

地址偏移: 0x48/0x9C/0xF0/0x144/0x198/0x1EC/0x240

Bit	名称	复位值	读/写	功能
31:12	保留	20'b0	RW	-
11:0	WINDOW	12'b0	RW	Blanking Window 的宽度设置。 0x000: 不产生 blanking window 0x001-0xFFFF: 指定在时基计数器周期内的 blanking window 的宽度。当偏移计数器满时启动 blanking window。接着, window 计数器加载完成并开始向下计数。如果当前的 blanking window 是 active 的并且偏移计数器满, 重启 blanking window 计数器。

3 EPWM

				Blanking window 可以通过一个 PWM 周期边界，如果当前 blanking window 是激活的，重启 blanking window。
--	--	--	--	---

3-23 EPWMx_TZCTL

地址偏移：0x4C/0xA0/0xF4/0x148/0x19C/0x1F0/0x244

Bit	名称	复位值	读/写	功能
31	DCBEVT1_OSTEN	0	RW	数字比较输出事件 B1 的使能位。 0: 不使能 DCBEVT1 作为 EPWM 模块的单次触发源(DCBEVT1 从 DC 模块获取) 1: 使能 DCBEVT1 作为 EPWM 模块的单次触发源
30	DCAEVT1_OSTEN	0	RW	数字比较输出事件 A1 使能位。 0: 不使能 DCAEVT1 作为 EPWM 模块的单次触发源 1: 使能 DCAEVT1 作为 EPWM 模块的单次触发源
29	OSHT6	0	RW	EMUSTOP 的使能选择位。 0: 不使能 EMUSTOP 作为 EPWM 模块的单次触发源 1: 使能 EMUSTOP 作为 EPWM 模块的单次触发源
28	OSHT5	0	RW	SYSTEM CLOCK 的使能选择位。 0: 不使能 SYSTEM CLOCK 作为 EPWM 模块的单次触发源 1: 使能 SYSTEM CLOCK 作为 EPWM 模块的单次触发源
27	OSHT4	0	RW	故障区 4 (TZ4) 的使能选择位。 0: 不使能 TZ4 作为 EPWM 的模块单次触发源 1: 使能 TZ4 作为 EPWM 的模块单次触发源
26	OSHT3	0	RW	故障区 3 (TZ3) 的使能选择位。 0: 不使能 TZ3 作为 EPWM 的模块单次触发源 1: 使能 TZ3 作为 EPWM 的模块单次触发源
25	OSHT2	0	RW	故障区 2 (TZ2) 的使能选择位。 0: 不使能 TZ2 作为 EPWM 模块的单次触发源 1: 使能 TZ2 作为 EPWM 模块的单次触发源
24	OSHT1	0	RW	故障区 1 (TZ1) 的使能选择位。

3 EPWM

				0: 不使能 TZ1 作为 EPWM 的模块单次触发源 1: 使能 TZ1 作为 EPWM 的模块单次触发源
23	DCBEVT2_CBCEN	0	RW	数字比较输出事件 B2 选择位。 0: 不使能 DCBEVT2 作为 EPWM 模块的周期循环触发源 1: 使能 DCBEVT2 作为 EPWM 模块的周期循环触发源
21	DCAEVT2_CBCEN	0	RW	数字比较输出事件 A2 选择位。 0: 不使能 DCAEVT2 作为 EPWM 模块的周期循环触发源 1: 使能 DCAEVT2 作为 EPWM 模块的周期循环触发源
21	CBC6	0	RW	EMUSTOP 的使能选择位。 0: 不使能 EMUSTOP 作为 EPWM 模块的周期循环触发源 1: 使能 EMUSTOP 作为 EPWM 模块的周期循环触发源
20	CBC5	0	RW	SYSTEM CLOCK 的使能选择位。 0: 不使能 SYSTEM CLOCK 作为 EPWM 模块的周期循环触发源 1: 使能 SYSTEM CLOCK 作为 EPWM 模块的周期循环触发源
19	CBC4	0	RW	故障区 4 (TZ4) 的使能选择位。 0: 不使能 TZ4 作为 EPWM 模块的周期循环触发源 1: 使能 TZ4 作为 EPWM 模块的周期循环触发源
19	CBC3	0	RW	故障区 3 (TZ3) 的使能选择位。 0: 不使能 TZ3 作为 EPWM 模块的周期循环触发源 1: 使能 TZ3 作为 EPWM 模块的周期循环触发源
17	CBC2	0	RW	故障区 2 (TZ2) 的使能选择位。 0: 不使能 TZ2 作为 EPWM 模块的周期循环触发源 1: 使能 TZ2 作为 EPWM 模块的周期循环触发源

3 EPWM

				源
16	CBC1	0	RW	故障区 1 (TZ1) 的使能选择位。 0: 不使能 TZ1 作为 EPWM 模块的周期循环触发源 1: 使能 TZ1 作为 EPWM 模块的周期循环触发源
15:12	保留	4'b0	R	-
11:10	DCBEVT2_FRCEN	00	R	数字比较输出事件 B2 在 EPWMxB 的动作选择位。 00: 高阻抗 (EPWMxB = 高阻抗状态) 01: 强制 EPWMxB 输出为高 10: 强制 EPWMxB 输出为低 11: 无任何效果, 不使能 trip
9:8	DCAEVT2_FRCEN	00	RW	数字比较输出事件 A2 在 EPWMxA 的动作选择位。 00: 高阻抗 (EPWMxA = 高阻抗状态) 01: 强制 EPWMxA 输出为高 10: 强制 EPWMxA 输出为低 11: 无任何效果, 不使能 trip
7:6	DCBEVT1_FRCEN	00	R	数字比较输出事件 B1 在 EPWMxB 的动作选择位。 00: 高阻抗 (EPWMxB = 高阻抗状态) 01: 强制 EPWMxB 输出为高 10: 强制 EPWMxB 输出为低 11: 无任何效果, 不使能 trip
5:4	DCAEVT1_FRCEN	00	RW	数字比较输出事件 A1 在 EPWMxA 的动作选择位。 00: 高阻抗 (EPWMxA = 高阻抗状态) 01: 强制 EPWMxA 输出为高 10: 强制 EPWMxA 输出为低 11: 无任何效果, 不使能 trip
3:2	TZB	00	RW	EPWMxB 事件发生选择位。当发生 trip 事件时, EPWMxB 会发生下列动作。Tripo_zone 引脚可以使得在 TZSEL 寄存器中定义事件。 00: 高阻抗 (EPWMxB = 高阻抗状态)

3 EPWM

				01: 强制 EPWMxB 输出为高 10: 强制 EPWMxB 输出为低 11: 无任何效果, EPWMxB 无任何动作发生
1:0	TZA	00	RW	EPWMxA 事件发生选择位。当发生 trip 事件时, EPWMxA 会发生下列动作。Tripo_zone 引脚可以使得在 TZSEL 寄存器中定义事件。 00: 高阻抗 (EPWMxA = 高阻抗状态) 01: 强制 EPWMxA 输出为高 10: 强制 EPWMxA 输出为低 11: 无任何效果, EPWMxA 无任何动作发生

3-24 EPWMx_TZFLAG

地址偏移: 0x50/0xA4/0xF8/0x14C/0x1A0/0x1F4/0x248

Bit	名称	复位值	读/写	功能
31	保留	0	RW	-
30	DCBEVT2INTE	0	RW	数字比较器输出事件 B2 的中断使能位。 0: 不使能 1: 使能
29	DCBEVT1INTE	0	RW	数字比较器输出事件 B1 的中断使能位。 0: 不使能 1: 使能
28	DCAEVT2INTE	0	RW	数字比较器输出事件 A2 的中断使能位。 0: 不使能 1: 使能
27	DCAEVT1INTE	0	RW	数字比较器输出事件 A1 的中断使能位。 0: 不使能 1: 使能
26	OSTINTE	0	RW	故障区单次中断使能位。 0: 不使能单次中断 1: 使能单次中断, 一个单次触发事件可以触发一个 EPWMx_TZINT 的中断
25	CBCINTE	0	RW	故障区周期循环 (CBC) 中断使能位。 0: 不使能周期循环 (CBC) 中断 1: 使能周期循环中断, 一个周期循环事件可以触发一个 EPWMx_TZINT 中断

3 EPWM

24:23	保留	00	R	-
22	CLRDCBEVT2	0	W	写 1 清 0
21	CLRDCBEVT1	0	W	写 1 清 0
20	CLRDCAEVT2	0	W	写 1 清 0
19	CLRDCAEVT1	0	W	写 1 清 0
18	CLROST	0	W	写 1 清 0
17	CLRCBC	0	W	写 1 清 0
16	CLRINT	0	W	写 1 清 0
15	保留	0	R	-
14	FRDCBEVT2	0	W	数字比较输出事件 B2 的强制标志位。 0: 无任何效果, 读返回 0 1: 强制 DCBEVT2 事件 trip, 并设置 DCBEVT2_FLAG (EPWMx_TZFLAG[6]) 为 1
13	FRDCBEVT1	0	W	数字比较输出事件 B1 的强制标志位。 0: 无任何效果, 读返回 0 1: 强制 DCBEVT1 事件 trip, 并设置 DCBEVT1_FLAG (EPWMx_TZFLAG[5]) 为 1
12	FRDCAEVT2	0	W	数字比较输出事件 A2 的强制标志位。 0: 无任何效果, 读返回 0 1: 强制 DCAEVT2 事件 trip, 并设置 DCAEVT2_FLAG (EPWMx_TZFLAG[4]) 为 1
11	FRDCAEVT1	0	W	数字比较输出事件 A1 的强制标志位。 0: 无任何效果, 读返回 0 1: 强制 DCAEVT1 事件 trip, 并设置 DCAEVT1_FLAG (EPWMx_TZFLAG[3]) 为 1
10	FR COST	0	W	单次 trip 事件的软件强制标志位。 0: 无任何效果, 读返回 0 1: 强制发生单次 trip 事件并设置 OST (EPWMx_TZFLAF[2]) 为 1
9	FRCCBC	0	W	周期循环 (CBC) trip 事件的软件强制标志位。 0: 无任何效果, 读返回 0 1: 强制发生一个周期循环事件并设置 CBC 位 (EPWMx_TZFLAG[1]) 为 1

3 EPWM

8:7	保留	00	R	-
6	DCBEVT2_FLAG	0	RW	数字比较输出事件 B2 的锁存状态标志位。 0: DCBEVT2 没有发生 trip 事件 1: DCBEVT2 发生了一个 trip 事件
5	DCBEVT1_FLAG	0	RW	数字比较输出事件 B1 的锁存状态位。 0: DCBEVT1 没有发生 trip 事件 1: DCBEVT1 发生了一个 trip 事件
4	DCAEVT2_FLAG	0	RW	数字比较输出事件 A2 的锁存状态标志位。 0: DCAEVT2 没有发生 trip 事件 1: DCAEVT2 发生了一个 trip 事件
3	DCAEVT1_FLAG	0	RW	数字比较输出事件 A1 的锁存状态位。 0: DCAEVT1 没有发生 trip 事件 1: DCAEVT1 发生了一个 trip 事件
2	OST	0	RW	单次 trip 事件的锁存状态标志位。 0: 没有发生单次 trip 事件 1: 单次 trip 信号源引脚上发生了一个单次 trip 事件 注意: 当在 CLROST (EWPMx_TZINT[18]) 上写入适当的值时该位清 0。
1	CBC	0	RW	周期循环 trip 事件的锁存状态标志位。 0: 不发生周期循环 (CB) trip 事件 1: 在周期循环 (CBC) trip 信号源的信号上发生一个 trip 事件。该位需由用户手动清 0。当该位清 0 后, 如果周期循环 trip 事件仍然存在, 那么该位将立即再次置 1。当 EPWM 的时基计数器等于 0 时 (TBCTR = 0x0000), 如果周期循环 trip 事件不再存在, 那么将自动清除信号上的 trip 事件。无论在循环的什么时候 CBC 清 0, 只有当 TBCTR = 0x0000 时, 信号上的 trip 事件才会被清 0。当向 CLROST (EWPMx_TZINT[18]) 中写入适当的值, 该位会被清 0。
0	INT	0	RW	Trip 锁存中断状态标志位。 0: 没有中断发生 1: EPWMx_TZINT PIE 中断发生, 因为发生了 Trip 事件

3 EPWM

				注意：只有当该位清 0 之后才会产生下一个 EPWMx_TZINT PIE 中断。当 CBC 或 OST 置 1 时，如果中断标志位清 0，那么就会产生其他中断脉冲。清除所有的中断位可以防止产生更多中断。向 CLROST (EPWMx_TZINT[18]) 写入适当的值可以清除该位。
--	--	--	--	---

4 历史

版本号	日期	说明
V1.0	2019/04/18	Official version
V1.1	2020/06/11	修改部分寄存器的功能描述