

---

## 第 44 章 电机控制 PWM（MCPWM）

---

本章包括下列主题：

44.1 简介 .....	44-2
44.2 特性 .....	44-2
44.3 控制寄存器 .....	44-3
44.4 架构概述 .....	44-37
44.5 模块说明 .....	44-39
44.6 PWM 输出状态控制 .....	44-43
44.7 PWM 工作模式 .....	44-46
44.8 PWM 发生 .....	44-61
44.9 写保护 .....	44-79
44.10 PWM 输出模式 .....	44-80
44.11 PWM 发生器触发 .....	44-84
44.12 PWM 中断 .....	44-90
44.13 PWM 故障 .....	44-91
44.14 PWM 限流 .....	44-96
44.15 并发 PWM 故障和限流 .....	44-100
44.16 到 ADC 的 PWM 故障和限流触发输出 .....	44-100
44.17 特殊功能 .....	44-102
44.18 节能模式 .....	44-117
44.19 独立时基的外部控制 .....	44-118
44.20 应用信息 .....	44-118
44.21 相关应用笔记 .....	44-130
44.22 版本历史 .....	44-131

**注：** 本系列参考手册章节旨在用作对器件数据手册的补充。根据不同的器件型号，本手册章节可能并不适用于所有 PIC32 器件。

请参见当前器件数据手册中“**电机控制 PWM (Motor Control PWM, MCPWM)**”章节开头部分的注，以检查本文档是否支持您所使用的器件。

器件数据手册和系列参考手册章节可从 Microchip 网站下载：

<http://www.microchip.com>。

## 44.1 简介

本章介绍电机控制脉宽调制器（MCPWM）模块及其相关的工作模式。PIC32 器件系列中的 MCPWM 模块支持多种 PWM 模式，是电源转换 / 电机控制应用的理想选择。一些常见应用包括：

- SMPS 应用：
  - 交流 / 直流转换器
  - 直流 / 直流转换器
- 交流和直流电机（例如，BDC、BLDC、PMSM、ACIM、SRM）
- 逆变器
- 电池充电器
- 数字照明
- 不间断电源（UPS）
- 功率因数校正（PFC）

## 44.2 特性

MCPWM 模块具有以下主要特性：

- 两个主控时基
- 最多 12 个 PWM 发生器，每个发生器具有独立时基：
  - 8 个 PWM 发生器，带互补输出
  - 4 个额外的 PWM 发生器，带单端输出
- 单个周期、占空比和相移寄存器，具有每个发生器的动态更新
- 从系统时钟（SYSCLK）产生的占空比、死区、相移和频率分辨率
- 全部 12 个 PWM 发生器的独立故障输入和限流输入
- 冗余输出模式
- 辅助占空比寄存器支持异步 PWM 模式
- 推挽输出模式
- 互补输出模式
- 中心对齐 PWM 模式
- 输出改写控制
- 用于同步模数转换的特殊事件触发器
- PWM 捕捉功能
- 输入时钟预分频器
- 模数转换器（Analog-to-Digital Converter, ADC）触发功能，带有 PWM
- 前沿消隐（Leading-Edge Blanking, LEB）功能
- 死区补偿
- 输出时钟斩波
- 每个与 PWM 模块相关的输出引脚都可以被单独使能
- PWM 输出引脚的手动改写 SFR 位

### 44.3 控制寄存器

**注：** 并非所有寄存器在所有器件上都可用。关于可用性，请参见具体器件数据手册中的“电机控制 PWM (MCPWM)”章节。

以下寄存器用于控制 MCPWM 模块的工作。

#### 44.3.1 主控时基控制寄存器

- **PTCON: PWM 主时基控制寄存器**

此寄存器可控制主控时基的工作，包括使能或禁止 MCPWM 模块、提供模块状态、设置特殊事件触发、触发后分频器以及中断。此外，此寄存器可用于控制 PWM 输出改写逻辑电平，并对主 PWM 输入时钟进行预分频。

- **PTPER: 主主控时基周期寄存器**

此寄存器可存储从主主控时基产生时钟源的发生器的同步周期。

- **SEVTCMP: PWM 主特殊事件比较寄存器**

此寄存器可存储用于触发 ADC 模块的主主控时基的比较值。

- **PMTMR: 主主控时基定时器寄存器**

此寄存器可实现从主主控时基产生时钟的发生器定时器的重载同步。

- **STCON 辅助主控时基控制寄存器**

此寄存器可对辅助 PWM 输入时钟进行预分频、选择辅助主控时基的同步源，并指定辅助主控时基控制的同步设置。

- **STPER: 辅助主控时基周期寄存器**

此寄存器可存储从辅助主控时基产生时钟源的发生器的同步周期。

- **SSEVTCMP: PWM 辅助特殊事件比较寄存器**

此寄存器可提供用于触发 ADC 模块的辅助主控时基的比较值。

- **SMTMR: 辅助主控时基定时器寄存器**

此寄存器可实现从辅助主控时基产生时钟的发生器定时器的重载同步。

- **CHOP: PWM 斩波时钟发生器寄存器**

此寄存器可使能 / 禁止斩波时钟发生器并提供斩波时钟频率。

- **PWMKEY: PWM 解锁寄存器**

此寄存器接受解锁序列，允许写入 IOCONx 寄存器。

## 44.3.2 PWM 发生器控制寄存器

每个 PWM 发生器具有以下寄存器组合：

- **PWMCONx: PWM 控制寄存器 x (x = 1 至 12)**

此寄存器可控制 PWM 模块的基本操作，包括外部 PWM 复位操作、死区补偿模式和极性、使能 / 禁止和提供故障中断、限流中断和主目标中断的状态以及选择同步类型、时钟源和对齐模式。

- **IOCONx: PWMx I/O 控制寄存器 x (x = 1 至 12)**

此寄存器可控制 PWM 模块的 I/O 功能，包括取决于所选模式、交换和输出极性的 PWMxH/PWMxL 引脚输出，以及使能 / 禁止 PWM 引脚控制功能和故障 / 限流改写值。

- **PDCx: PWM 发生器占空比寄存器 x (x = 1 至 12)**

当 MCPWM 模块处于边沿对齐模式时，此寄存器可指定开启时间的下降沿实例并直接控制占空比。

当 MCPWM 模块处于对称中心对齐模式时，此寄存器可控制前沿跳变状态时的实例。SDCx 寄存器将在更新时自动复制 PDCx 寄存器的值。

当 MCPWM 模块处于非对称中心对齐模式时，此寄存器只会存储前沿跳变比较实例。

- **SDCx: PWM 辅助占空比寄存器 x (x = 1 至 12)**

当 MCPWM 模块处于边沿对齐模式时，此寄存器未使用。

当 MCPWM 模块处于对称中心对齐模式时，此寄存器虽然可在此模式下工作，但会透明地更新到软件。装入 PDCx 寄存器的数据会自动复制到 SDCx 寄存器。

当 MCPWM 模块处于非对称中心对齐模式时，下降沿比较实例将存储在此寄存器中。

- **PHASEx: PWM 主相移寄存器 x (x = 1 至 12)**

如果选择了主控时基，此寄存器将提供 PWMxH 和 PWMxL 输出的相移值。如果选择了独立时基，此寄存器将提供 PWMxH 和 PWMxL 输出的独立时基周期。

- **DTRx: PWM 死区寄存器 x (x = 1 至 12)**

当 MCPWM 模块处于正死区模式时，此寄存器将从 PWMxL 的下降沿延迟 PWMxH 的前沿。

当 MCPWM 模块处于负死区模式时，此寄存器将从 PWMxL 的前沿延迟 PWMxH 的下降沿。

- **ALTDTRx: PWM 备用死区寄存器 x (x = 1 至 12)**

当 MCPWM 模块处于正死区模式时，此寄存器将从 PWMxH 的下降沿延迟 PWMxL 的前沿。

当 MCPWM 模块处于负死区模式时，此寄存器将从 PWMxH 的前沿延迟 PWMxL 的下降沿。

- **DTCOMPx: 死区补偿寄存器 x (x = 1 至 12)**

此寄存器可根据 DTCMPy 引脚和 DTCP 极性位的状态，延长或缩短 PWMxH/PWMxL 的开启时间。关于死区补偿的详细信息，请参见第 44.8.6 节“死区补偿”。

- **TRIGx: PWM 主触发比较值寄存器 x (x = 1 至 12)**

此寄存器提供的比较值与本地定时器 PTMRx 时基进行比较可产生 ADC 转换或中断的主触发。

- **TRGCONx: PWM 触发控制寄存器 x (x = 1 至 12)**

此寄存器使能 PWMx 触发后分频比开始事件，并指定在产生第一次触发之前跳过的 PWM 周期数。

- **STRIGx: 辅助 PWM 触发比较寄存器 x (x = 1 至 12)**

此寄存器提供的比较值与本地定时器 PTMRx 时基进行比较可产生 ADC 转换或中断的辅助触发。

- **CAPx: PWM 定时器捕捉寄存器 x (x = 1 至 12)**

此寄存器提供在以下情况下捕捉的本地时基定时器 (PTMRx) 的时基值: 在限流输入上检测到前沿, 并且对限流输入信号的 LEB 处理已完成。

- **LEBCONx: 前沿消隐控制寄存器 x (x = 1 至 12)**

此寄存器选择前沿消隐 (Leading-edge Blanking, LEB) 的 PWMxH 和 PWMxL 输出的上升和 / 或下降沿, 并使能或禁止故障和限流输入的 LEB。

- **LEBDLYx: 前沿消隐延时寄存器 x (x = 1 至 12)**

此寄存器提供故障输入和限流输入的前沿消隐延时。

- **AUXCONx: PWM 附属控制寄存器 x (x = 1 至 12)**

此寄存器选择 PWM 状态消隐和斩波时钟源以及 PWMxH 和 PWMxL 输出斩波功能。

- **PTMRx: PWM 定时器寄存器 x (x = 1 至 12)**

此寄存器包含每个发生器的本地 PWM 时基定时器的当前值。

表 44-1 简要汇总了所有与电机控制 PWM（MCPWM）模块相关的寄存器。该汇总表之后列出了相应的寄存器，并且每个寄存器均附有详细的说明。

表 44-1: MCPWM 特殊功能寄存器汇总

寄存器名称	位范围	Bit 31/15	Bit 30/14	Bit 29/13	Bit 28/12	Bit 27/11	Bit 26/10	Bit 25/9	Bit 24/8	Bit 23/7	Bit 22/6	Bit 21/5	Bit 20/4	Bit 19/3	Bit 18/2	Bit 17/1	Bit 16/0
PTCON	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	15:0	PTEN	—	PTSIDL	SESTAT	SEIEN	PWMRDY	—	—	—	PCLKDIV<2:0>			SEVTPS<3:0>			
PTPER	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	15:0	PTPER<15:0>															
SEVTCMP	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	15:0	SEVTCMP<15:0>															
PMTMR	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	15:0	PMTMR<15:0>															
STCON	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	15:0	—	—	—	SSESTAT	SSEIEN	—	—	—	—	SCLKDIV<2:0>			SEVTPS<3:0>			
STPER	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	15:0	STPER<15:0>															
SSEVTCMP	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	15:0	SSEVTCMP<15:0>															
SMTMR	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	15:0	SMTMR<15:0>															
CHOP	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	15:0	CHPCLKEN	—	—	—	—	—	CHOPCLK<9:0>									
PWMKEY	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	15:0	PWMKEY<15:0>															
PWMCONx (x = 1-12)	31:16	FLTIF	CLIF	TRGIF	PWMLIF	PWMHIF	—	—	—	—	FLTIE	CLIE	TRGIE	PWMLIE	PWMHIE	—	—
	15:0	FLTSTAT	CLTSTAT	—	—	ECAM<1:0>			ITB	—	DTC<1:0>		DTCP	PTDIR	MTBS	—	XPRES
IOCONx (x = 1-12)	31:16	—	—	CLSRC<3:0>				CLPOL	CLMOD	—	FLTSRC<3:0>				FLTPOL	FLTMOD<1:0>	
	15:0	PENH	PENL	POLH	POLL	PMOD<1:0>			OVRENH	OVRENL	OVRDAT<1:0>		FLTDAT<1:0>		CLDAT<1:0>		SWAP
PDCx (x = 1-12)	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	15:0	PDC<15:0>															
SDCx (x = 1-12)	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	15:0	SDC<15:0>															
PHASEx (x = 1-12)	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	15:0	PHASE<15:0>															
DTRx (x = 1-12)	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	15:0	DTR<15:0>															
ALTDTRx (x = 1-12)	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	15:0	ALTDTR<15:0>															
DTCOMPx (x = 1-12)	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	15:0	—	—	DTCOMP<13:0>													
TRIGx (x = 1-12)	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	15:0	TRGCMP<15:0>															
TRGCONx (x = 1-12)	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	15:0	TRGDIV<3:0>				TRGSEL<1:0>			STRGSEL<1:0>			DTM	STRGIS	—	—	—	—

图注: — = 未实现, 读为 0。

表 44-1: MCPWM 特殊功能寄存器汇总 (续)

寄存器名称	位范围	Bit 31/15	Bit 30/14	Bit 29/13	Bit 28/12	Bit 27/11	Bit 26/10	Bit 25/9	Bit 24/8	Bit 23/7	Bit 22/6	Bit 21/5	Bit 20/4	Bit 19/3	Bit 18/2	Bit 17/1	Bit 16/0
STRIGx (x = 1-12)	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	15:0	STRGCMP<15:0>															
CAPx (x = 1-12)	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	15:0	CAP<15:0>															
LEBCONx (x = 1-12)	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	15:0	PHR	PHF	PLR	PLF	FLTLEBEN	CLLEBEN	—	—	—	—	—	—	—	—	—	—
LEBDLYx (x = 1-12)	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	15:0	—	—	—	—	LEB<11:0>											
AUXCONx (x = 1-12)	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	15:0	—	—	—	—	—	—	—	—	—	CHOPSEL<3:0>					CHOPHEN	CHOPLEN
PTMRx (x = 1-12)	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	15:0	TMR<15:0>															

图注: — = 未实现, 读为 0。

寄存器 44-1: PTCN: PWM 主时基控制寄存器

位范围	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	R/W-0	U-0	R/W-0	HS/HC-0	R/W-0	HS/HC-0	U-0	U-0
	PTEN	—	PTSIDL	SESTAT <sup>(1)</sup>	SEIEN <sup>(3)</sup>	PWMRDY	—	—
7:0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	—	PCLKDIV<2:0> <sup>(2)</sup>			SEVTPS<3:0> <sup>(2)</sup>			

图注:	HS = 硬件置 1 位	HC = 硬件清零位
R = 可读位	W = 可写位	U = 未实现位, 读为 0
-n = POR 时的值	1 = 置 1	0 = 清零
		x = 未知

- bit 31-16 **未实现:** 读为 0
- bit 15 **PTEN:** PWM 模块使能位  
1 = 使能 PWM 模块  
0 = 禁止 PWM 模块
- bit 14 **未实现:** 读为 0
- bit 13 **PTSIDL:** PWM 时基空闲模式停止位  
1 = PWM 时基在 CPU 空闲模式下暂停  
0 = PWM 时基在 CPU 空闲模式下运行
- bit 12 **SESTAT:** 特殊事件中断状态位 <sup>(1)</sup>  
1 = 特殊事件中断处于待处理状态  
0 = 特殊事件中断不处于待处理状态
- bit 11 **SEIEN:** 特殊事件中断允许位 <sup>(3)</sup>  
1 = 允许特殊事件中断  
0 = 禁止特殊事件中断
- bit 10 **PWMRDY:** PWM 模块状态位  
1 = PWM 模块已就绪, 操作已开始  
0 = PWM 模块未就绪
- bit 9-7 **未实现:** 读为 0
- bit 6-4 **PCLKDIV<2:0>:** 主 PWM 输入时钟预分频比位 <sup>(2)</sup>  
111 = 128 分频, PWM 分辨率 = 128/FSYSCLK  
110 = 64 分频, PWM 分辨率 = 64/FSYSCLK  
.  
.  
.  
000 = 1 分频, PWM 分辨率 = 1/FSYSCLK (上电默认值)

- 注 1: 通过将 SEIEN 位和 IFSx 寄存器中的相应位清零, 可将 SESTAT 位清零。
- 2: 只有 PTEN 位 (PTCN<15>) = 0 时, 才能更改 SEVTPS<3:0> 位。
- 3: 要清除主特殊事件中断, 用户应用程序必须执行以下操作:  
a) 通过将 SEIEN 位清零 (也就是将该位设置为 0), 禁止主特殊事件中断。  
b) 通过将 IFSx 寄存器中的相应位清零, 清零主特殊事件中断标志。  
c) 如果需要, 通过将 SEIEN 位置 1, 重新使能主特殊事件中断。只要 SEIEN 位等于 1, 用户应用程序就无法清零主特殊事件中断标志。



### 寄存器 44-1: PTCON: PWM 主时基控制寄存器 (续)

bit 3-0      **SEVTPS<3:0>**: PWM 特殊事件触发输出后分频比选择位 <sup>(2)</sup>

1111 = 1:16, 后分频器在每发生 16 个比较匹配事件时产生特殊事件触发信号

•

•

•

0001 = 1:2, 后分频器在每发生两个比较匹配事件时产生特殊事件触发信号

0000 = 1:1, 后分频器在每次发生比较匹配事件时产生特殊事件触发信号

**注 1:** 通过将 SEIEN 位和 IFSx 寄存器中的相应位清零, 可将 SESTAT 位清零。

**2:** 只有 PTEN 位 (PTCON<15>) = 0 时, 才能更改 SEVTPS<3:0> 位。

**3:** 要清除主特殊事件中断, 用户应用程序必须执行以下操作:

a) 通过将 SEIEN 位清零 (也就是将该位设置为 0), 禁止主特殊事件中断。

b) 通过将 IFSx 寄存器中的相应位清零, 清零主特殊事件中断标志。

c) 如果需要, 通过将 SEIEN 位置 1, 重新使能主特殊事件中断。只要 SEIEN 位等于 1, 用户应用程序就无法清零主特殊事件中断标志。

寄存器 44-2: PTPER: 主主控时基周期寄存器

位范围	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	PTPER<15:8> <sup>(1,2)</sup>							
7:0	R/W-0	R/W-0	R/W-1	R/W-0	R/W-0	R/W-0 <sup>(3)</sup>	R/W-0 <sup>(3)</sup>	R/W-0 <sup>(3)</sup>
	PTPER<7:0> <sup>(1,2)</sup>							

图注:			
R = 可读位	W = 可写位	U = 未实现位, 读为 0	
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

bit 31-16 未实现: 读为 0  
bit 15-0 PTPER<15:0>: 主主控时基周期值位 <sup>(1,2)</sup>

- 注 1: 1 LSb = 1/FSYSCLK (最小值)。  
2: 最小值为 0x0008。  
3: 如果选择了小于 0x0008 的周期值, 内部硬件会将周期强制设置为最小值 0x0008。  
4:  $PTPER = (FSYSCLK / (FPWM * (2^{PTCON<PCLKDIV>})))$ ; FPWM = 用户所需的 PWM 频率。

## 第 44 章 电机控制 PWM（MCPWM）

寄存器 44-3: SEVTCMP: PWM 主特殊事件比较寄存器

位范围	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	SEVTCMP<15:8>							
7:0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	SEVTCMP<7:0>							

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 31-16 未实现: 读为 0

bit 15-0 SEVTCMP<15:0>: 特殊事件比较计数位

当此寄存器的内容与 PMTMR 值相匹配时, 将产生特殊事件。

注 1: 1 LSb = 1/FSYSCLK (最小值)

2: 设置 SEVTCMP = 0, 以在 PWM 周期边界产生主特殊事件触发。SEVTCMP >= PTPER 将不会产生任何触发。

寄存器 44-4: PMTMR: 主主控时基定时器寄存器

位范围	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
	PMTMR<15:8>							
7:0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
	PMTMR<7:0>							

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 31-16 未实现: 读为 0

bit 15-0 PMTMR<15:0>: 主主控时基定时器值位

此定时器随每个 PWM 时钟递增, 直到达到 PTPER 值。

**寄存器 44-5: STCON 辅助主控时基控制寄存器**

位范围	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	U-0	U-0	U-0	HS/HC-0	R/W-0	U-0	U-0	U-0
	—	—	—	SSESTAT <sup>(1)</sup>	SSEIEN	—	—	—
7:0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	—	SCLKDIV<2:0> <sup>(2)</sup>			SEVTPS<3:0> <sup>(2)</sup>			

**图注:**

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 31-13 **未实现:** 读为 0

bit 12 **SSESTAT:** 辅助特殊事件中断状态位 <sup>(1)</sup>

1 = 辅助特殊事件中断处于待处理状态

0 = 辅助特殊事件中断不处于待处理状态

bit 11 **SSEIEN:** 辅助特殊事件中断允许位

1 = 允许辅助特殊事件中断

0 = 禁止辅助特殊事件中断

bit 10-7 **未实现:** 读为 0

bit 6-4 **SCLKDIV<2:0>:** 辅助 PWM 输入时钟预分频比 <sup>(2)</sup>

111 = 128 分频, PWM 分辨率 = 128/FSYSCLK

110 = 64 分频, PWM 分辨率 = 64/FSYSCLK

•

•

•

000 = 1 分频, PWM 分辨率 = 1/FSYSCLK (上电默认值)

bit 3-0 **SEVTPS<3:0>:** PWM 辅助特殊事件触发输出后分频比选择位 <sup>(2)</sup>

1111 = 1:16 后分频比

•

•

•

0001 = 1:2 后分频比

0000 = 1:1 后分频比

**注 1:** 通过将 SSEIEN 位和 IFSx 寄存器中的相应位清零, 可将 SSESTAT 位清零。

**2:** 只有 PTEN 位 (PTCON<15>) = 0 时, 才能更改 SEVTPS<3:0> 位。

**3:** 要清除辅助特殊事件中断, 用户应用程序必须执行以下操作:

a) 通过将 SEIEN 位清零 (也就是将该位设置为 0), 禁止辅助特殊事件中断。

b) 通过将 IFSx 寄存器中的相应位清零, 清零辅助特殊事件中断标志。

c) 如果需要, 通过将 SSEIEN 位置 1, 重新使能辅助特殊事件中断。只要 SSEIEN 位等于 1, 用户应用程序就无法清零辅助特殊事件中断标志。

## 第 44 章 电机控制 PWM（MCPWM）

寄存器 44-6: STPER: 辅助主控时基周期寄存器

位范围	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0 <sup>(3)</sup>	R/W-0 <sup>(3)</sup>	R/W-0 <sup>(3)</sup>
	STPER<15:8> <sup>(1,2)</sup>							
7:0	R/W-0	R/W-0	R/W-1	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	STPER<7:0> <sup>(1,2)</sup>							

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 31-16 未实现: 读为 0

bit 15-0 STPER<15:0>: 主主控时基周期值位<sup>(1,2)</sup>

注 1: 1 LSb = 1/FSYSCLK (最小值)

2: 最小值为 0x0008。

3: 如果选择了小于 0x0008 的周期值, 内部硬件会将周期强制设置为最小值 0x0008。

4:  $STPER = (FSYSCLK / (FPWM * (2^{PTCON<SCLKDIV>})))$ ; FPWM = 用户所需的 PWM 频率。

寄存器 44-7: SSEVTCMP: PWM 辅助特殊事件比较寄存器

位范围	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	SSEVTCMP<15:8>							
7:0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	SSEVTCMP<7:0>							

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 31-16 未实现: 读为 0

bit 15-0 SSEVTCMP<15:0>: 辅助特殊事件比较值位

当此寄存器的内容与 SMTMR 值相匹配时, 将产生辅助特殊事件。

注 1: 设置 SSEVTCMP = 0, 以在 PWM 周期边界产生辅助特殊事件触发。SEVTCMP>= STPER 将不会产生任何触发。

寄存器 44-8: SMTMR: 辅助主控时基定时器寄存器

位范围	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
	SMTMR<15:8>							
7:0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
	SMTMR<7:0>							

图注:			
R = 可读位	W = 可写位	U = 未实现位, 读为 0	
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

bit 31-16 未实现: 读为 0

bit 15-0 **SMTMR<15:0>**: 辅助主控时基定时器值位  
此定时器随每个 PWM 时钟递增, 直到达到 STPER 值。

寄存器 44-9: CHOP: PWM 斩波时钟发生器寄存器

位范围	Bit 31/2 /15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	R/W-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
	CHPCLKEN	—	—	—	—	—	CHOPCLK<9:8>(2,3)	
7:0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	CHOPCLK<7:0>(2,3)							

**图注:**

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 31-16 未实现: 读为 0

bit 15 **CHPCLKEN**: 使能斩波时钟发生器位

1 = 使能斩波时钟发生器 (1)

0 = 禁止斩波时钟发生器

bit 14-10 未实现: 读为 0

bit 9-0 **CHOPCLK<9:0>**: 斩波时钟分频比位 (2,3)

斩波频率 = (FSYSCLK/(2<sup>PTCON<PCLKDIV></sup>)) / (CHOPCLK<9:0>)

**注 1:** 斩波时钟发生器通过 PCLKDIV<2:0> 位 (PTCON<6:4>) 工作。

**2:** 最小值为 0x0002。值为 0x0000 或 0x0001 时, 将无法产生斩波时钟。

**3:** 只有在 PTEN 位 (PTCON<15>) 清零时才能更改这些位。

**注:** 斩波时钟是一种连续高频信号 (相对于 PWM 周期), 它可选择用 PWM 输出信号进行门控, 以允许 PWM 信号通过外部隔离层, 比如脉冲变压器或电容。[CHOP<9:0> \* PWM 时钟持续时间] 的值定义斩波时钟的高电平和低电平时间。CHOP 寄存器中的值为 8 时, 会产生周期为 16 个 PWM 时钟周期的斩波时钟信号, 如主 PWM 时钟预分频比 PCLKDIV<2:0> 所定义。值为 0x0000 或 0x0001 时, 不会产生任何斩波时钟。

**寄存器 44-10: PWMKEY: PWM 解锁寄存器**

位范围	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0
	PWMKEY<15:8>							
7:0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0
	PWMKEY<7:0>							

**图注:**

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 31-16 **未实现:** 读为 0

bit 15-0 **PWMKEY<15:0>:** PWM 解锁位

如果 PWMLOCK 配置位置为有效 (PWMLOCK = 1), 则只有向 PWMKEY 寄存器中写入正确序列之后, IOCONx 寄存器才可写。如果 PWMLOCK 配置位置为无效 (PWMLOCK = 0), 则 IOCONx 寄存器总是可写。关于解锁序列的更多信息, 请参见第 44.9 章“写保护”。

只有在 FOSCSEL 配置寄存器中存在 PWMLOCK 配置位的器件中才能实现该寄存器。

**注:** 如果 PWMLOCK = 0, 用户必须向 PWMKEY 寄存器连续写入两个值 0xABCD 和 0x4321, 才能执行解锁操作。对任何后续安全寄存器的写访问必须是紧接在解锁过程之后的下一个访问。这并不是原子操作, 而且解锁序列期间或紧接其之后的任何 CPU 中断都可能导致无法写入任何 PWM 安全寄存器。



寄存器 44-11: PWMCONx: PWM 控制寄存器 x (x = 1 至 12)

位范围	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0
	FLTIF	CLIF	TRGIF	PWMLIF	PWMHIF	—	—	—
23:16	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0
	FLTIE	CLIE	TRGIE	PWMLIE	PWMHIE	—	—	—
15:8	HS/HC-0	HS/HC-0	U-0	U-0	R/W-0	R/W-0	R/W-0	U-0
	FLTSTAT <sup>(1)</sup>	CLTSTAT <sup>(1)</sup>	—	—	ECAM<1:0> <sup>(1)</sup>		ITB <sup>(2)</sup>	—
7:0	R/W-0	R/W-0	R/W-0	HS/HC/R-0	R/W-0	U-0	R/W-0	U-0
	DTC<1:0>		DTCP <sup>(4)</sup>	PTDIR <sup>(6)</sup>	MTBS <sup>(7)</sup>	—	XPRES <sup>(3)</sup>	—

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 31 **FLTIF**: 故障中断标志位

1 = 发生了故障中断

0 = 未发生故障中断

bit 30 **CLIF**: 限流状态位

1 = 发生了限流

0 = 未发生限流

bit 29 **TRGIF**: 触发中断状态位

1 = 触发中断处于待处理状态

0 = 触发中断不处于待处理状态

bit 28 **PWMLIF**: PWML 中断状态位

1 = 发生了 PWM 周期复位中断

0 = 未发生 PWM 周期复位中断

bit 27 **PWMHIF**: PWMH 中断状态位

1 = 发生了 PWM 周期匹配中断

0 = 未发生 PWM 周期匹配中断

bit 26-24 **未实现**: 读为 0

bit 23 **FLTIE**: 故障中断允许位

1 = 允许故障中断。如果 FLTIF = 1, 将产生中断事件。

0 = 禁止故障中断

bit 22 **CLIE**: 限流中断允许位

1 = 允许限流中断。如果 CLIF = 1, 将产生中断事件。

0 = 禁止限流中断

bit 21 **TRGIE**: 主触发中断允许位

1 = 主触发事件产生中断请求

0 = 禁止产生主触发事件中断请求

**注 1:** 如果允许 PWM 中断, 则软件必须先清零 PWMCONx 中断标志, 然后将中断控制器中的相应 IFSx 位清零。如果未先将这些本地 PWMCON 中断位清零, 将无法清零相应的 PWM IFSx 中断标志。因此, 将产生无限的中断循环。

**2:** 使能 PWM (PTEN (PTCON<15>) = 1) 之后, 不应更改该位。

**3:** 要在外部周期复位模式下工作, ITB 位必须设置为 1, IOCONx 寄存器中的 CLMOD 位必须设置为 0。

**4:** 要使 DTCP 生效, DTC<1:0> 必须设置为 11; 否则, DTCP 会被忽略。

**5:** 负死区仅对于边沿对齐模式实现。

**6:** 该位只有在中心对齐模式期间才有效并得到更新。

**7:** 即使选择了 ITB = 1, 时钟源也是其中一个主控时基。

## 寄存器 44-11: PWMCONx: PWM 控制寄存器 x (x = 1 至 12) (续)

bit 20	<b>PWMLIEN:</b> PWM 低相位中断允许位 1 = 当发生 PWM 周期复位 (即 PWM 定时器 = 0x0) 时, PWMLIF 标志 = 1 并产生中断请求 0 = 禁止 PWM 周期事件中断请求
bit 19	<b>PWMHIEN:</b> PWM 高相位中断允许位 1 = 当 PWM 周期与 PWM 定时器中的值相匹配时, 将产生中断请求 0 = 禁止 PWM 周期事件中断请求, 并将 PWMHIF 位清零
bit 18-16	<b>未实现:</b> 读为 0
bit 15	<b>FLTSTAT:</b> 故障中断状态位 <sup>(1)</sup> 1 = 故障中断处于待处理状态 0 = 没有故障中断处于待处理状态 该位通过设置 <b>FLTIE</b> = 0 进行清零。
bit 14	<b>CLSTAT:</b> 限流中断状态位 <sup>(1)</sup> 1 = 限流中断处于待处理状态 0 = 没有限流中断处于待处理状态 该位通过设置 <b>CLIE</b> = 0 进行清零。
bit 13-12	<b>未实现:</b> 读为 0
bit 11-10	<b>ECAM&lt;1:0&gt;:</b> 边沿 / 中心对齐模式使能位 <sup>(1)</sup> 11 = 非对称中心对齐模式, 可同时更新 10 = 非对称中心对齐模式 01 = 对称中心对齐模式 00 = 边沿对齐模式
bit 9	<b>ITB:</b> 独立时基模式位 <sup>(2)</sup> 1 = PHASEx 寄存器为该 PWM 发生器提供时基周期 0 = PTPER/STPER 寄存器基于 MTBS 位为该 PWM 发生器提供时序
bit 8	<b>未实现:</b> 读为 0
bit 7-6	<b>DTC&lt;1:0&gt;:</b> 死区控制位 11 = 使能死区补偿模式 10 = 禁止死区功能 01 = 对于互补输出模式施加负死区 <sup>(5)</sup> 00 = 对于所有输出模式施加正死区
bit 5	<b>DTCP:</b> 死区补偿极性位 <sup>(5)</sup> 1 = 如果 DTCMPx 引脚 = 0, 则缩短 PWMxL, 延长 PWMxH 如果 DTCMPx 引脚 = 1, 则缩短 PWMxH, 延长 PWMxL 0 = 如果 DTCMPx 引脚 = 0, 则缩短 PWMxH, 延长 PWMxL 如果 DTCMPx 引脚 = 1, 则缩短 PWMxL, 延长 PWMxH
bit 4	<b>PTDIR:</b> PWM 定时器方向位 <sup>(6)</sup> 1 = PWM 定时器正在递减 0 = PWM 定时器正在递增
bit 3	<b>MTBS:</b> 主控时基选择位 <sup>(7)</sup> 1 = 辅助主控时基是 MCPWM 模块的时钟源 0 = 主主控时基是 MCPWM 模块的时钟源

**注 1:** 如果允许 PWM 中断, 则软件必须先清零 PWMCONx 中断标志, 然后将中断控制器中的相应 IFSx 位清零。如果未先将这些本地 PWMCON 中断位清零, 将无法清零相应的 PWM IFSx 中断标志。因此, 将产生无限的中断循环。

**2:** 使能 PWM (PTEN (PTCON<15>) = 1) 之后, 不应更改该位。

**3:** 要在外部周期复位模式下工作, ITB 位必须设置为 1, IOCONx 寄存器中的 CLMOD 位必须设置为 0。

**4:** 要使 DTCP 生效, DTC<1:0> 必须设置为 11; 否则, DTCP 会被忽略。

**5:** 负死区仅对于边沿对齐模式实现。

**6:** 该位只有在中心对齐模式期间才有效并得到更新。

**7:** 即使选择了 ITB = 1, 时钟源也是其中一个主控时基。

### 寄存器 44-11: PWMCONx: PWM 控制寄存器 x (x = 1 至 12) (续)

bit 2      未实现: 读为 0

bit 1      **XPRES:** 外部 PWM 复位控制位 <sup>(3)</sup>

1 = 如果该 PWM 发生器处于独立时基模式, 则限流源复位该 PWM 发生器的主本地时基  
0 = 外部引脚不影响 PWM 时基

bit 0      未实现: 读为 0

**注 1:** 如果允许 PWM 中断, 则软件必须先清零 PWMCONx 中断标志, 然后将中断控制器中的相应 IFSx 位清零。如果未先将这些本地 PWMCON 中断位清零, 将无法清零相应的 PWM IFSx 中断标志。因此, 将产生无限的中断循环。

**2:** 使能 PWM (PTEN (PTCON<15>) = 1) 之后, 不应更改该位。

**3:** 要在外部周期复位模式下工作, ITB 位必须设置为 1, IOCONx 寄存器中的 CLMOD 位必须设置为 0。

**4:** 要使 DTCP 生效, DTC<1:0> 必须设置为 11; 否则, DTCP 会被忽略。

**5:** 负死区仅对于边沿对齐模式实现。

**6:** 该位只有在中心对齐模式期间才有效并得到更新。

**7:** 即使选择了 ITB = 1, 时钟源也是其中一个主控时基。

**寄存器 44-12: IOCONx: PWMx I/O 控制寄存器 x (x = 1 至 12)**

位范围	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	—	—	CLSRC<3:0> <sup>(2,4)</sup>				CLPOL <sup>(2,4)</sup>	CLMOD <sup>(2,4)</sup>
23:16	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-0	R/W-0	R/W-0
	—	FLTSRC<3:0> <sup>(2,4)</sup>				FLTPOL <sup>(2)</sup>	FLTMOD<1:0> <sup>(4)</sup>	
15:8	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	PENH <sup>(1)</sup>	PENL <sup>(1)</sup>	POLH <sup>(2)</sup>	POLL <sup>(2)</sup>	PMOD<1:0> <sup>(2)</sup>		OVRENH	OVRENL
7:0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	OVRDAT<1:0> <sup>(3)</sup>		FLTDAT<1:0> <sup>(2,3)</sup>		CLDAT<1:0>		SWAP	OSYNC

**图注:**

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 31-30 **未实现:** 读为 0

bit 29-26 **CLSRC<3:0>:** PWM 发生器 x 的限流控制信号源选择位 <sup>(2,4)</sup>

这些位指定限流控制信号源。关于可用选择, 请参见具体器件数据手册中的“**电机控制 PWM (MCPWM)**”章节。

bit 25 **CLPOL:** PWM 发生器 x 的限流极性位 <sup>(2,4)</sup>

1 = 选定的限流源为低电平有效

0 = 选定的限流源为高电平有效

bit 24 **CLMOD:** PWM 发生器 x 的限流模式使能位 <sup>(2,4)</sup>

1 = 使能限流功能

0 = 禁止限流功能, 并禁止限流改写 (可能仍会产生限流中断)

更改将在使能 PWM 之后的下个 PWM 周期边界以及后续每个 PWM 周期边界上生效。将 CLMOD 从 1 更新到 0 时, 如果限流输入仍处于有效状态, 将不会删除限流改写条件。

bit 23 **未实现:** 读为 0

bit 22-19 **FLTSRC<3:0>:** PWM 发生器 x 的故障控制信号源选择位 <sup>(2,4)</sup>

这些位指定故障控制源。关于可用选择, 请参见具体器件数据手册中的“**电机控制 PWM (MCPWM)**”章节。

bit 18 **FLTPOL:** PWM 发生器 x 的故障极性位 <sup>(2)</sup>

1 = 选定的故障源为低电平有效

0 = 选定的故障源为高电平有效

**注 1:** 在 PWM 初始化期间, 如果 PWMLOCK 熔丝位处于已使能状态 (逻辑 0), 则 PWMxL/PWMxH 输出引脚状态仅取决于 PENH 和 PENL 位。但是, 这些位为 0, 这使引脚由 I/O 模块控制。必须注意不要无意间将 TRIS 位设置为输出, 这会使 PWMxH/PWMxL 引脚即使在存在外部上拉和下拉电阻时也输出不正确。引脚的数据方向必须设置为输入 (如果需要三态行为) 或驱动为相应的逻辑状态。必须总是在使能 MCPWM 模块 (PTEN 位 = 1) 前初始化 PENH 和 PENL 位。

**2:** 使能 MCPWM 模块 (PTEN 位 = 1) 之后, 不得更改这些位。

**3:** 状态代表 PWM 的有效/无效状态, 具体取决于 POLH 和 POLL 位的设置。例如, 如果 FLTDAT<1> 设置为 1, POLH 设置为 1, 则在发生故障时 PWMxH 引脚将为逻辑电平 0 (有效电平)。

**4:** 如果 PWMLOCK = 0, 则只有向 PWMKEY 写入正确序列之后, 这些位才可写。如果 PWMLOCK = 1, 则这些位始终可写。

## 寄存器 44-12: IOCONx: PWMx I/O 控制寄存器 x (x = 1 至 12) (续)

bit 17-16	<b>FLTMOD&lt;1:0&gt;</b> : PWM 发生器 x 的故障模式位 <sup>(4)</sup> 11 = 禁止故障输入, 无法进行故障改写。(可能仍会发生故障中断) 10 = 保留 01 = 选定的故障源将 PWMxH 和 PWMxL 引脚强制为 FLTDAT<1:0> 值 (逐个周期) 00 = 选定的故障源将 PWMxH 和 PWMxL 引脚强制为 FLTDAT<1:0> 值 (锁定状态) 更改将在使能 PWM 之后的下个 PWM 周期边界以及后续每个 PWM 周期边界上生效。将 FLTMOD<1:0> 从 00 或 01 更新到 11 (禁止) 时, 如果故障输入仍处于有效状态, 将不会删除故障改写条件。
bit 15	<b>PENH</b> : PWMxH 输出引脚所有权位 <sup>(1)</sup> 1 = PWM 模块控制 PWMxH 引脚 0 = GPIO 模块控制 PWMxH 引脚
bit 14	<b>PENL</b> : PWMxL 输出引脚所有权位 <sup>(1)</sup> 1 = PWM 模块控制 PWMxL 引脚 0 = GPIO 模块控制 PWMxL 引脚
bit 13	<b>POLH</b> : PWMxH 输出引脚极性位 <sup>(2)</sup> 1 = PWMxH 引脚为低电平有效 0 = PWMxH 引脚为高电平有效
bit 12	<b>POLL</b> : PWMxL 输出引脚极性位 <sup>(2)</sup> 1 = PWMxL 引脚为低电平有效 0 = PWMxL 引脚为高电平有效
bit 11-10	<b>PMOD&lt;1:0&gt;</b> : PWM x I/O 引脚模式位 <sup>(2)</sup> 11 = PWMxL 输出保持为逻辑 0 (通过 POLL 位调整) 10 = PWM I/O 引脚对处于推挽输出模式 01 = PWM I/O 引脚对处于冗余输出模式 00 = PWM I/O 引脚对处于互补输出模式
bit 9	<b>OVRENH</b> : PWMxH 引脚改写使能位 1 = OVRDAT<1> 为 PWMxH 引脚上的输出提供数据 0 = PWM 发生器为 PWMxH 引脚提供数据
bit 8	<b>OVRENL</b> : PWMxL 引脚改写使能位 1 = OVRDAT<0> 为 PWMxL 引脚上的输出提供数据 0 = PWM 发生器为 PWMxL 引脚提供数据
bit 7-6	<b>OVRDAT&lt;1:0&gt;</b> : 使能改写时 PWMxH 和 PWMxL 引脚状态 <sup>(3)</sup> 位 如果 OVRENH = 1, 则 OVRDAT<1> 提供 PWMxH 的数据 如果 OVRENL = 1, 则 OVRDAT<0> 提供 PWMxL 的数据
bit 5-4	<b>FLTDAT&lt;1:0&gt;</b> : 使能 FLTMOD 时 PWMxH 和 PWMxL 引脚状态 <sup>(3)</sup> 位 <sup>(2)</sup> 如果 FLTMOD<1:0> (IOCONx<17:16>) = 00 或 01, 将使能以下其中一个故障模式: 如果故障有效, 则 FLTDAT<1> 提供 PWMxH 的状态 如果故障有效, 则 FLTDAT<0> 提供 PWMxL 的状态 如果故障无效, 则 FLTDAT<1:0> 位被忽略

**注 1:** 在 PWM 初始化期间, 如果 PWMLOCK 熔丝位处于已使能状态 (逻辑 0), 则 PWMxL/PWMxH 输出引脚状态仅取决于 PENH 和 PENL 位。但是, 这些位为 0, 这使引脚由 I/O 模块控制。必须注意不要无意间将 TRIS 位设置为输出, 这会使 PWMxH/PWMxL 引脚即使在存在外部上拉和下拉电阻时也输出不正确。引脚的数据方向必须设置为输入 (如果需要三态行为) 或驱动为相应的逻辑状态。必须总是在使能 MCPWM 模块 (PTEN 位 = 1) 前初始化 PENH 和 PENL 位。

**2:** 使能 MCPWM 模块 (PTEN 位 = 1) 之后, 不得更改这些位。

**3:** 状态代表 PWM 的有效 / 无效状态, 具体取决于 POLH 和 POLL 位的设置。例如, 如果 FLTDAT<1> 设置为 1, POLH 设置为 1, 则在发生故障时 PWMxH 引脚将为逻辑电平 0 (有效电平)。

**4:** 如果 PWMLOCK = 0, 则只有向 PWMKEY 写入正确序列之后, 这些位才可写。如果 PWMLOCK = 1, 则这些位始终可写。

## 寄存器 44-12: IOCONx: PWMx I/O 控制寄存器 x (x = 1 至 12) (续)

- bit 3-2     **CLDAT<1:0>**: 使能 CLMOD 时 PWMxH 和 PWMxL 引脚状态<sup>(3)</sup> 位  
如果 CLMOD (IOCONx<24>) = 1, 将使能限流模式, 如下所示:  
如果限流有效, 则 CLTDAT<1> 提供 PWMxH 的状态  
如果限流有效, 则 CLTDAT<0> 提供 PWMxL 的状态  
如果限流无效, 则 CLTDAT<1:0> 位被忽略
- bit 1     **SWAP**: 交换 PWMxH 和 PWMxL 引脚位  
1 = PWMxH 输出信号连接到 PWMxL 引脚; PWMxL 输出信号连接到 PWMxH 引脚  
0 = PWMxH 和 PWMxL 输出信号引脚映射到它们各自对应的引脚
- bit 0     **OSYNC**: 输出改写同步位  
1 = 通过 OVRDAT<1:0> 位进行的输出改写与 PWM 时基同步  
0 = 通过 OVRDAT<1:0> 位进行的输出改写在下一个 CPU 时钟边界发生

- 注 1:** 在 PWM 初始化期间, 如果 PWMLOCK 熔丝位处于已使能状态 (逻辑 0), 则 PWMxL/PWMxH 输出引脚状态仅取决于 PENH 和 PENL 位。但是, 这些位为 0, 这使引脚由 I/O 模块控制。必须注意不要无意间将 TRIS 位设置为输出, 这会使 PWMxH/PWMxL 引脚即使在存在外部上拉和下拉电阻时也输出不正确。引脚的数据方向必须设置为输入 (如果需要三态行为) 或驱动为相应的逻辑状态。必须总是在使能 MCPWM 模块 (PTEN 位 = 1) 前初始化 PENH 和 PENL 位。
- 2:** 使能 MCPWM 模块 (PTEN 位 = 1) 之后, 不得更改这些位。
- 3:** 状态代表 PWM 的有效 / 无效状态, 具体取决于 POLH 和 POLL 位的设置。例如, 如果 FLTDAT<1> 设置为 1, POLH 设置为 1, 则在发生故障时 PWMxH 引脚将为逻辑电平 0 (有效电平)。
- 4:** 如果 PWMLOCK = 0, 则只有向 PWMKEY 写入正确序列之后, 这些位才可写。如果 PWMLOCK = 1, 则这些位始终可写。

寄存器 44-13: PDCx: PWM 发生器占空比寄存器 x (x = 1 至 12)

位范围	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	PDC<15:8>							
7:0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	PDC<7:0>							

**图注:**

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 31-16 未实现: 读为 0

bit 15-0 **PDC<15:0>**: PWM 发生器占空比值位

如果使能边沿对齐模式 (ECAM<1:0> (PWMCONx<11:10>) = 00), 则这些位将指定开启时间实例下降沿实例, 并直接控制占空比。

最小 PWM 分辨率 = 1 / FSYSCLK。

如果使能其中一个中心对齐模式 (ECAM<1:0> (PWMCONx<11:10>) = 01、10 或 11), 则这些位将指定前沿电平跳变的比较实例。

最小 PWM 分辨率 = 2 / FSYSCLK。

**注 1:** 在独立 PWM 模式 (PMOD<1:0> (IOCONx<11:10>) = 11) 下, PDCx 寄存器仅控制 PWMxH 占空比。在互补、冗余和推挽 PWM 模式 (PMOD<1:0> = 00、01 或 10) 下, PDCx 寄存器控制 PWMxH 和 PWMxL 的占空比。

寄存器 44-14: SDCx: PWM 辅助占空比寄存器 x (x = 1 至 12)

位范围	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	SDC<15:8>							
7:0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	SDC<7:0>							

**图注:**

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 31-16 **未实现:** 读为 0

bit 15-0 **SDC<15:0>:** PWMxL 输出引脚的辅助占空比位

如果使能边沿对齐模式 (ECAM<1:0> (PWMCONx<11:10>) = 00), 将不会使用这些位。

如果使能对称中心对齐模式 (ECAM<1:0> (PWMCONx<11:10>) = 01), 将对用户透明地更新这些位。装入 PDCx 寄存器的数据会自动复制到 SDCx 寄存器。

如果使能非对称中心对齐模式 (ECAM<1:0> (PWMCONx<11:10>) = 10 或 11), 则这些位将指定下降沿电平跳变的比较实例。



寄存器 44-15: PHASEx: PWM 主相移寄存器 x (x = 1 至 12)

位范围	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	PHASE<15:8>							
7:0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	PHASE<7:0>							

**图注:**

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 31-16 未实现: 读为 0

bit 15-0 PHASE<15:0>: PWM 发生器的 PWM 相移值或独立时基周期<sup>(5)</sup> 位

**注 1:** 如果 ITB 位 (PWMCONx<9>) = 0, 则依工作模式适用以下情况:

互补、冗余和推挽输出模式 (PMOD<1:0> (IOCONx<11:10>) = 00、01 或 10), PHASE<15:0> = PWMxH 和 PWMxL 输出的相移值

**2:** 如果 ITB 位 = 1, 则依工作模式适用以下情况:

互补、冗余和推挽输出模式 (PMOD<1:0> = 00、01 或 10), PHASE<15:0> = PTMRx 的本地时基周期值

**3:** 超出 PWM 周期值的相位偏移将导致无法预测的结果。

**4:** ITB = 1 时的最小周期值为 0x0008。

**5:** 关于边沿对齐和中心对齐模式下的周期计算, 请分别参见公式 44-1 和公式 44-3。

**寄存器 44-16: DTRx: PWM 死区寄存器 x (x = 1 至 12)**

位范围	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	—	—	DTR<13:8>					
7:0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	DTR<7:0>							

**图注:**

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 31-16 **未实现:** 读为 0

bit 15-0 **DTR<13:0>:** PWMxH 死区单元的无符号 14 位死区值位

这些位指定 PWMxH 和 PWMxL 之间的前沿死区计数。该计数的时基与 PWM 发生器的相同。

**注:** 死区周期通常设置为应用电路中功率晶体管的开关时间。它专用于互补输出模式。任何其他模式下的死区使用都可能发生意外或无法预测的结果。如果 PDCx/SDCx 寄存器中的占空比值等于 0, 或者大于或等于周期, 将忽略死区补偿。占空比 + 死区 + 死区补偿的值不得超过周期寄存器减 1 的值。如果总和超过周期寄存器 - 1, 则可能会产生无法预料的结果。占空比 + 死区 + 死区补偿的值不能大于 0, 否则可能产生无法预料的结果。

寄存器 44-17: ALTDTRx: PWM 备用死区寄存器 x (x = 1 至 12)

位范围	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	—	—	ALTDTR<13:8>					
7:0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	ALTDTR<7:0>							

**图注:**

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 31-16 未实现: 读为 0

bit 15-0 ALTDTR<13:0>: PWMxL 死区单元的无符号 14 位死区值位

这些位指定 PWMxH 和 PWMxL 之间的下降沿死区计数。该计数的时基与 PWM 发生器的相同。

**注:** 备用死区周期通常设置为应用电路中功率晶体管的开关时间。它专用于互补输出模式。任何其他模式下的死区使用都可能发生意外或无法预测的结果。如果 PDCx/SDCx 寄存器中的占空比值等于 0, 或者大于或等于周期, 将忽略备用死区补偿。占空比 + 死区 + 备用死区补偿的值不能超过周期寄存器 - 1 的值。如果总和超过周期寄存器 - 1, 则可能会产生无法预料的结果。占空比 + 死区 - 备用死区补偿的值不得大于 0, 否则可能产生无法预料的结果。

寄存器 44-18: DTCOMPx: 死区补偿寄存器 x (x = 1 至 12)

位范围	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	—	—	DTCOMP<13:8>					
7:0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
	DTCOMP<7:0>							

**图注:**

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 31-16 未实现: 读为 0

bit 15-0 **DTCOMP<13:0>**: 死区补偿值位  
使能死区补偿模式时的死区补偿值。

**注 1:** 最小死区补偿分辨率:  $1 \text{ LSb} = 1 / \text{FSYSCLK}$  (边沿对齐模式)。

$1 \text{ LSb} = 1 / (2 * \text{FSYSCLK})$  (中心对齐模式)。

**2:** 当通过 PWMCONx 寄存器中的 DTC<1:0> 位选择死区补偿模式时, 连接到死区补偿模块输入信号的外部引脚 DTCMPx (即 FLT<sub>x</sub>) 会将 DTCOMPx 寄存器中的值加上 PWMx 占空比, 或从该占空比减去该值。将在 PWM 周期结束时对死区补偿输入信号进行采样, 以用于下一个 PWM 周期。在占空比结束 (下降沿) 时将通过 DTCMPx 寄存器对占空比持续时间进行修改。死区补偿只能用于正死区模式。DTCMPx 值必须小于占空比寄存器 PDCx/SDCx 的值的一半, 否则可能产生无法预测的行为。死区补偿不适用于值为零的占空比。在这种情况下, PWM 输出将保持为零, 而不管 DTCMPx 输入引脚处于什么状态。

寄存器 44-19: TRIGx: PWM 主触发比较值寄存器 x (x = 1 至 12)

位范围	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	TRGCMP<15:8>							
7:0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	TRGCMP<7:0>							

**图注:**

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 31-16 未实现: 读为 0

bit 15-0 **TRGCMP<15:0>**: 触发比较值位

这些位指定与本地时基寄存器 PTMRx 相匹配的值, 以产生 ADC 模块的触发和中断 (如果 TRGIEN 位 (PWMCONx<21>) 置 1)。

**注 1:** 设置 TRIGx = 0, 以在 PWM 周期边界产生触发。TRIGx >= PWM 周期将不会产生任何触发。

寄存器 44-20: TRGCONx: PWM 触发控制寄存器 x (x = 1 至 12)

位范围	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	TRGDIV<3:0>				TRGSEL<1:0> <sup>(1)</sup>		STRGSEL<1:0> <sup>(1)</sup>	
7:0	R/W-0	R/W-0	U-0	U-0	U-0	U-0	U-0	U-0
	DTM <sup>(1,2)</sup>	STRGIS <sup>(1)</sup>	—	—	—	—	—	—

**图注:**

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 31-16 **未实现:** 读为 0

bit 15-12 **TRGDIV<3:0>:** 触发器 x 输出分频比位

1111 = 每发生 16 个触发事件时触发输出

•  
•  
•

0010 = 每发生 3 个触发事件时触发输出

0001 = 每发生 2 个触发事件时触发输出

0000 = 每次发生触发事件时触发输出

bit 11-10 **TRGSEL<1:0>:** 双周期 PWM 周期的触发周期选择 (中心对齐和推挽) <sup>(1)</sup>

此位域对单周期 PWM 模式 (比如边沿对齐 PWM) 的原始触发生成没有影响。每次发生原始比较事件时, 原始事件将由触发分频器进行处理。

11 = 保留, 默认设为与 TRGSEL<1:0> = 00 相同的行为。

10 = 当在双周期 PWM 模式 (PTDIR = 0) 的递增相位中发生触发比较匹配事件时, 如果触发分频器对相应的触发事件进行了计数, 将产生触发事件输出。

01 = 当在双周期 PWM 模式 (PTDIR = 1) 的递减相位中发生触发比较匹配事件时, 如果触发分频器对相应的触发事件进行了计数, 将产生触发事件输出。

00 = 当发生触发比较事件时, 如果触发分频器对相应的原始触发事件进行了计数, 将产生触发事件输出。对于双周期 PWM 模式 (比如中心对齐模式和推挽模式), 将在每个周期产生两次原始触发事件。当 TRGSEL<1:0> = 00 时, 建议避免 TRIGx/STRIGx 比较值为 0 或等于 “周期 - 1”, 这会导致双周期 PWM 模式 (比如中心对齐模式和推挽模式) 的错误双触发。

**注 1:** 使能 MCPWM 模块 (PTEN 位 (PTCON<15>) = 1) 之后, 不得更改这些位。

**注 2:** 无论 DTM 位的设置如何, 都将产生辅助触发事件。

## 寄存器 44-20: TRGCONx: PWM 触发控制寄存器 x (x = 1 至 12) (续)

- bit 9-8     **STRGSEL<1:0>**: 双周期 PWM 周期的辅助触发周期选择位 (中心对齐和推挽) <sup>(1)</sup>
- 这些位对单周期 PWM 模式 (比如边沿对齐 PWM) 的原始辅助 PWM 触发生成没有影响。每次发生原始比较事件时, 原始事件将由辅助 PWM 触发分频器进行处理。
- 11 = 保留, 默认设为与 STRGSEL<1:0> = 00 相同的行为
- 10 = 当在双周期 PWM 模式 (PTDIR = 0) 的后半部分发生辅助 PWM 触发比较匹配事件时, 如果辅助 PWM 触发分频器对相应的辅助 PWM 触发事件进行了计数, 将产生辅助 PWM 触发事件输出。
- 01 = 当在双周期 PWM 模式 (PTDIR = 1) 的前半部分发生辅助 PWM 触发比较匹配事件时, 如果辅助 PWM 触发分频器对相应的辅助 PWM 触发事件进行了计数, 将产生触发事件输出。
- 00 = 当发生辅助 PWM 触发比较匹配事件时, 如果触发分频器对相应的原始辅助 PWM 触发事件进行了计数, 将产生辅助 PWM 触发事件输出。对于双周期 PWM 模式 (比如中心对齐模式和推挽模式), 将产生两次原始辅助 PWM 触发事件。当 TRGSEL<1:0> = 00 时, 建议避免 STRIGx 比较值为 0 或等于“周期 - 1”, 这会导致双周期 PWM 模式 (比如中心对齐模式和推挽模式) 的错误双触发。
- bit 7     **DTM**: 双 ADC 触发模式 <sup>(1, 2)</sup>
- 1 = 辅助触发事件与主触发事件组合产生组合 ADC 触发
- 0 = 辅助触发事件不与主触发事件组合产生组合 ADC 触发
- bit 6     **STRGIS**: 辅助触发中断选择
- 只有 PTEN = 0 时, 用户才能更改此位。
- 1 = 为中断选择基于辅助触发寄存器 (STRIGx) 的事件
- 0 = 当 DTM 位 (TRGCONx<7>) 清零 (= 0) 时, 将为中断选择基于 TRIGx 的事件。当 DTM 位置 1 (= 1) 时, 将为中断选择基于 STRIGx 和 TRIGx 的触发的逻辑或。
- bit 5-0     **未实现**: 读为 0

注 1: 使能 MCPWM 模块 (PTEN 位 (PTCON<15>) = 1) 之后, 不得更改这些位。

2: 无论 DTM 位的设置如何, 都将产生辅助触发事件。

**寄存器 44-21: STRIGx: 辅助 PWM 触发比较寄存器 x (x = 1 至 12)**

位范围	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	STRGCMP<15:8>							
7:0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	STRGCMP<7:0>							

**图注:**

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 31-16 **未实现:** 读为 0

bit 15-0 **STRGCMP<15:0>:** 辅助触发值

这些位存储的 16 位值与本地定时器 PTMRx 进行比较时, 如果使能 TRGIEN 位 (PWMCONx<21>) 和 DTM 位 (TRIGCONx<7>), 将产生 ADC 模块的触发以启动转换, 并产生中断的触发。

**注 1:** 最小分辨率: 1 LSB = 1/FSYSCLK

**2:** 设置 STRIGx = 0, 以在 PWM 周期边界产生触发。STRIGx >= PWM 周期将不会产生任何触发。

**寄存器 44-22: CAPx: PWM 定时器捕捉寄存器 x (x = 1 至 12)**

位范围	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
	CAP<15:8> <sup>(1)</sup>							
7:0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	CAP<7:0> <sup>(1)</sup>							

**图注:**

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 31-16 **未实现:** 读为 0

bit 15-0 **CAP<15:0>:** 捕捉的本地 PWM 定时器值位 <sup>(1)</sup>

该寄存器中的值代表在限流输入上检测到前沿时捕捉的本地 PWM 定时器 (PTMRx) 值。

**注 1:** 只有限流输入信号的 LEB 处理完成之后, 该功能才有效。



寄存器 44-23: LEBCONx: 前沿消隐控制寄存器 x (x = 1 至 12)

位范围	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0
	PHR	PHF	PLR	PLF	FLTLEBEN	CLLEBEN	—	—
7:0	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—

**图注:**

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 31-16 未实现: 读为 0

bit 15 **PHR:** PWMxH 上升沿触发使能位

1 = PWMxH 的上升沿将触发 / 重新触发前沿消隐计数器

0 = PWMxH 的上升沿将不会触发 / 重新触发前沿消隐计数器

bit 14 **PHF:** PWMxH 下降沿触发使能位

1 = PWMxH 的下降沿将触发 / 重新触发前沿消隐计数器

0 = PWMxH 的下降沿将不会触发 / 重新触发前沿消隐计数器

bit 13 **PLR:** PWMxL 上升沿触发使能位

1 = PWMxL 的上升沿将触发 / 重新触发前沿消隐计数器

0 = PWMxL 的上升沿将不会触发 / 重新触发前沿消隐计数器

bit 12 **PLF:** PWMxL 下降沿触发使能位

1 = PWMxL 的下降沿将触发 / 重新触发前沿消隐计数器

0 = PWMxL 的下降沿将不会触发 / 重新触发前沿消隐计数器

bit 11 **FLTLEBEN:** 故障输入前沿消隐使能位

1 = 对选定故障输入应用前沿消隐

0 = 不对选定故障输入应用前沿消隐

bit 10 **CLLEBEN:** 限流前沿消隐使能位

1 = 对选定限流输入应用前沿消隐

0 = 不对选定限流输入应用前沿消隐

bit 9-0 未实现: 读为 0

寄存器 44-24: LEBDLYx: 前沿消隐延时寄存器 x (x = 1 至 12)

位范围	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
	—	—	—	—	LEB<11:8>			
7:0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	LEB<7:0>							

图注:			
R = 可读位	W = 可写位	U = 未实现位, 读为 0	
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

bit 31-12   **未实现:** 读为 0

bit 11-0   **LEB<11:0>:** 限流输入和故障输入前沿消隐延时位

这些位用于指定在 PWM 信号的选定边沿跳变之后对选定限流和故障信号进行消隐或延时的时间周期。这种可重新触发的计数器以 PWM 模块时钟源 (FSYSCLK) 作为时基。

寄存器 44-25: AUXCONx: PWM 附属控制寄存器 x (x = 1 至 12)

位范围	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0 —	U-0 —	U-0 —	U-0 —	U-0 —	U-0 —	U-0 —	U-0 —
23:16	U-0 —	U-0 —	U-0 —	U-0 —	U-0 —	U-0 —	U-0 —	U-0 —
15:8	U-0 —	U-0 —	U-0 —	U-0 —	U-0 —	U-0 —	U-0 —	U-0 —
7:0	U-0 —	U-0 —	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CHOPSEL<3:0> <sup>(1)</sup>							CHOPHEN	CHOPLN

**图注:**

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 31-6 未实现: 读为 0

bit 5-2 **CHOPSEL<3:0>**: PWM 斩波时钟源选择位 <sup>(1)</sup>

选定信号将使能和禁止（斩波）选定的 PWM 输出。

1111 = 保留。不要使用

1110 = 保留。不要使用

1101 = 保留。不要使用

1100 = 选择 PWM12H 作为斩波时钟源

•

•

•

0111 = 选择 PWM7H 作为斩波时钟源

•

•

•

0001 = 选择 PWM1H 作为斩波时钟源

0000 = 选择斩波时钟发生器作为斩波时钟源

bit 1 **CHOPHEN**: PWMxH 输出斩波使能位

1 = 使能 PWMxH 斩波功能

0 = 禁止 PWMxH 斩波功能

bit 0 **CHOPLN**: PWMxL 输出斩波使能位

1 = 使能 PWMxL 斩波功能

0 = 禁止 PWMxL 斩波功能

**注 1:** 只有 PTEN 位 (PTCON<15>) = 0 时, 才能更改此位。

寄存器 44-26: PTMRx: PWM 定时器寄存器 x (x = 1 至 12)

位范围	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	TMR<15:8>							
7:0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	TMR<7:0>							

图注:			
R = 可读位	W = 可写位	U = 未实现位, 读为 0	
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

- bit 31-16 未实现: 读为 0
- bit 15-0 **TMR<15:0>**: PWM 定时器位

当 ECAM<1:0> 位 (PWMCONx<11:10>) = 00 时, 计数器将递增计数, 直到周期匹配强制计满返回。

当 ECAM<1:0> 位 (PWMCONx<11:10>) ≠ 00 时, 计数器将递减计数, 从主控时基同步信号开始到 0, 然后递增计数, 直到下一次同步。

## 44.4 架构概述

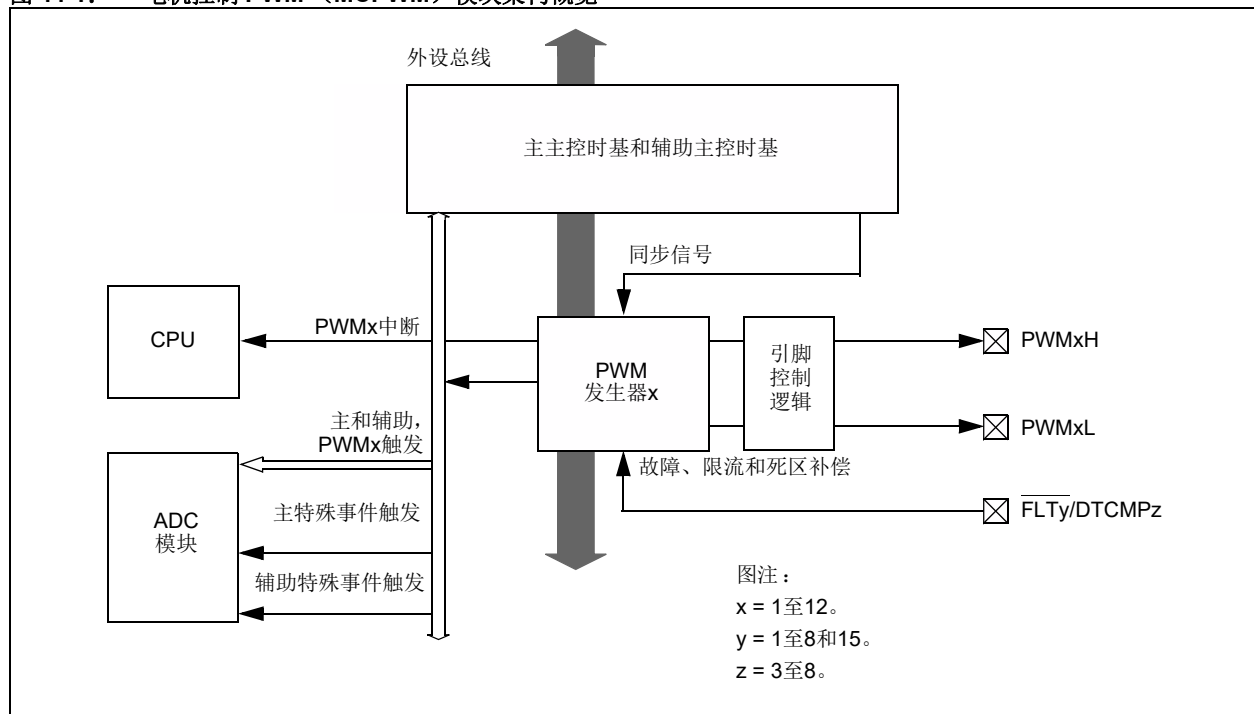
图 44-1 显示了 MCPWM 模块的架构概览，以及它与 CPU 和其他外设的相互连接。

MCPWM 模块包含两个具有附加监控功能（比如休眠模式）并使能 MCPWM 模块的控制和工作状态的主控时基。每个主控时基都提供同步信号，作为用于同步各种 PWM 输出的公共时基。每个发生器可以独立工作，也可以与主控时基同步。不管是什么同步类型，每个 PWM 发生器都会以其中一个主控时基作为其时基。输入故障信号和限流信号（在使能时）可以通过将 PWM 输出置为已知的“安全”状态来监视并保护系统。

每个 PWM 发生器都可以向 ADC 模块产生两个触发，使之可以使用 TRIGx 和 STRIGx 比较寄存器，在 PWM 周期中的特定时刻对模拟信号进行采样。而且，每个主控时基都可以向 ADC 模块发送一个特殊事件触发以启动转换，并在发生比较匹配事件时通过 SEVTCMP 和 SSEVTCMP 寄存器中断 CPU。

每个 PWM 发生器还具有一个专用的捕捉寄存器 CAPx，它可在发生故障或限流事件时通过数字输入 FLT<sub>x</sub> 引脚用于获取 PTMR<sub>x</sub> 值的快照。相对于及时发生的事件而言，在需要执行延时操作时，此功能会很有用。然后可以使用新的比较实例（CAP<sub>x</sub> 加上所需偏移量）来更新比较寄存器 TRIGx/STRIGx。当捕捉 / 比较事件不需要 ADC 触发时，可以相应地配置 ADC 模块。

图 44-1： 电机控制 PWM (MCPWM) 模块架构概览

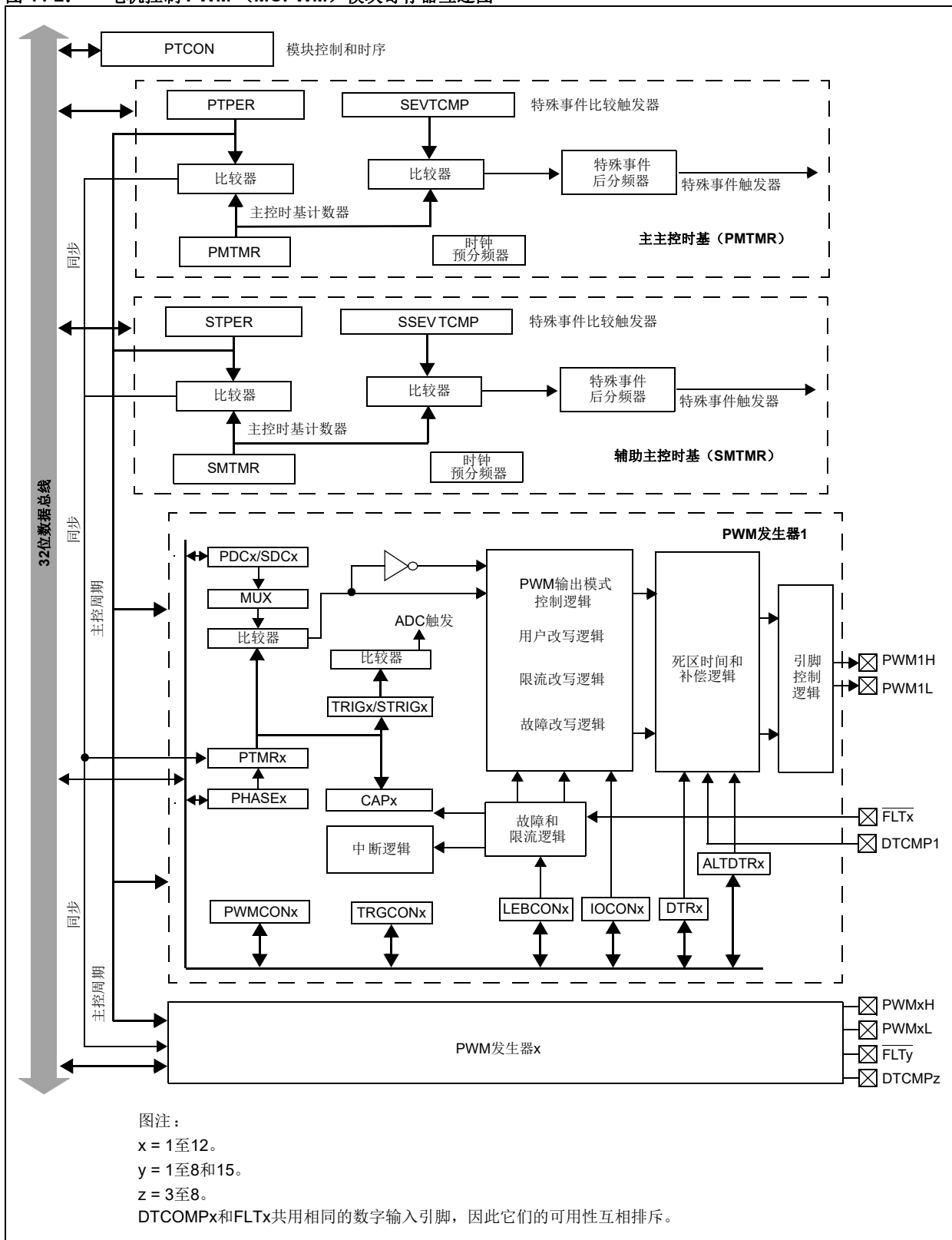


MCPWM 模块可用于要求以下特性的多种电源转换 / 电机控制应用：

- 高工作频率与高分辨率
- 占空比、周期和死区的动态更新
- 可同步或独立控制每个 PWM 发生器
- 故障处理能力
- 支持 CPU 负载交错，可执行多个控制环

后续章节将对 MCPWM 模块的每个功能进行详细描述。图 44-2 给出了 MCPWM 模块中各个寄存器之间的互连图。

图 44-2: 电机控制 PWM (MCPWM) 模块寄存器互连图



## 44.5 模块说明

### 44.5.1 PWM 时钟选择

系统时钟 (SYSCLK) 用于在内部产生 MCPWM 模块的时钟。当  $PTCON<PCLKDIV>/STCON<SCLKDIV> = 0$  时, 此模块的工作频率为最大。

### 44.5.2 主控时基

每个 PWM 发生器都使用 MTBS 位 ( $PWMCONx<3>$ ), 以两个主控时基的其中一个作为其时钟。即使 PWM 发生器周期未同步到时基也是如此。

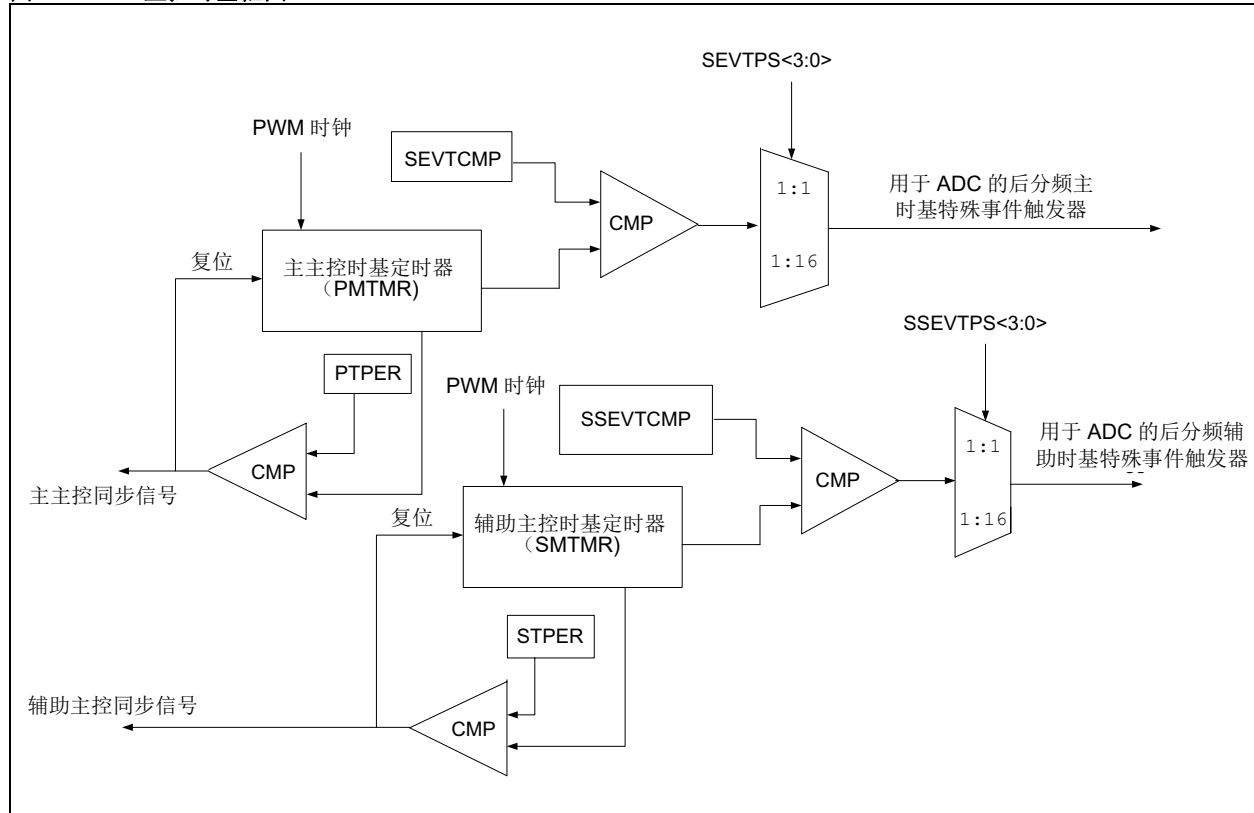
每个主控时基发生器的输入时钟都可以使用分频器选择位  $PCLKDIV<2:0>$  ( $PTCON<6:4>$ ) 和  $SCLKDIV<2:0>$  ( $STCON<6:4>$ ) 预分频到  $1/128$ 。预分频后的时钟是 PWM 时钟控制逻辑模块的输入。最大时钟速率提供 SYSCLK 的占空比和周期分辨率。

MCPWM 模块的功耗与其时基时钟速率具有线性关系。例如, 如果选择了 1:2 的预分频比选项, 则 PWM 占空比和周期分辨率可以设置为  $(1/FSYSCLK) * 2$ 。因此, 与最大速度工作时相比, MCPWM 模块的功耗可下降约 50%。

主控时基由主主控时基和辅助主控时基组成。

图 44-3 说明了主控时基的功能。

图 44-3: 主控时基框图



主控时基的一些常见任务包括：

- 向 PWM 发生器提供时钟
- 在周期匹配时产生同步触发以复位本地定时器 PTMRx
- 产生特殊事件 ADC 触发和中断

各个 PWM 发生器的时间周期选择是基于 MTBS 位（PWMCONx<3>）的设置。

PWM 发生器的 PWM 周期可能产生自两个源中的其中一个：

- 同步模式下两个主控时基发生器的其中一个。例如，使用 PTPER（MTBS = 0）和 STPER（MTBS = 1）寄存器的中心对齐模式（ITB 位（PWMCONx<9>）= 0）。
- 当 ITB = 1 时使用 PHASEx 寄存器进行周期匹配 / 复位

## 44.5.2.1 特殊事件触发器

MCPWM 模块具有两个基于每个主控时基的特殊事件触发器，可用于将 ADC 采样和转换同步到 PWM 周期内的任意特定时间。

触发器是使用比较寄存器 SEVTCMP/SSEVTCMP 产生的，它们将从 PWM 周期开始时装入所需的实例或计时器计数偏移。与任意主控时基定时器寄存器发生比较匹配将启动可进行后分频（如果需要）的触发，以降低中断频率。原始触发可以使用 SEVTP<3:0> 位（PTCON<3:0>）进行分频，它可以将中断线性缩小到 1:16。在双周期 PWM 模式下，由于每个周期产生两次原始触发，因此附加屏蔽允许在预分频前选择所需的触发。

定时器寄存器无论何时发生比较匹配，都将产生触发。为了选择性地禁止转换触发，可以将写入 SEVTCMP/SSEVTCMP SFR 的比较值设为大于 PTPER/STPER 的值，也可以在 12 位 HS SAR ADC 模块中禁止触发。请参见《PIC32 系列参考手册》中的第 22 章“12 位高速逐次逼近寄存器（SAR）模数转换器（ADC）”（DS60001344）。

存在可以通过触发同步使能的各个相关中断。

在以下情况下，总是会产生特殊事件触发脉冲：

- 在发生匹配条件时，无论特殊事件中断允许位 SEIEN（PTCON<11>）的状态如何
- 如果 SEVTCMP 寄存器中的比较值介于 0 和 PTPER 寄存器最大值之间

特殊事件触发输出后分频器在发生以下事件时清零：

- 任何器件复位
- 当 PTEN = 0 时

例 44-1 提供了 ADC 特殊事件触发配置的代码。关于特殊事件触发的更多信息，请参见第 44.11 章“PWM 发生器触发”。

### 例 44-1: ADC 特殊事件触发配置

```
/* Primary Master time base chosen */
/* ADC Special Event Trigger Configuration */
PTCONbits.SEVTPS = 0; // Special Event Trigger output postscaler set to 1:1 selection
PTCONbits.SEIEN = 0; // Special event interrupt is disabled

/* Choose ADC1 dedicated core to start conversion asynchronously */
ADCCON1bits.STRGSRC = 0x8;

SEVTCMP = 1248; // Special Event Trigger value set at 25% of the period value (4999)
while (PTCONbits.SESTAT == 1); // Wait for special event interrupt status change
```



### 44.5.3 PWM 发生器

MCPWM 模块最多有 12 个 PWM 发生器。每个发生器都具有以下功能：

- 边沿对齐、对称和非对称 PWM 模式
- 死区发生
- 通过外部引脚 (DTCMPx) 进行死区补偿
- 使用两个比较寄存器的触发机制
- 可控制前沿消隐的限流和故障改写功能
- 手动改写 PWM 输出功能

#### 44.5.3.1 工作模式

每个 PWM 发生器都可以在同步或独立模式下工作。

两种模式下的时钟源是通过 MTBS 位 (PWMCONx<3>) 选择的主主控时基或辅助主控时基。

##### 44.5.3.1.1 独立 PWM 操作

在独立 PWM 周期模式下，主控定时器周期匹配不会影响 PTMRx 值。这意味着 PTMRx 会与主控时基定时器 PMTMR 或 SMTMR 异步。PHASEx 寄存器用途发生了改变，可保存 PWM 的周期。因此，PWM 波形与其他发生器的波形不存在相位关系。通过将 ITB 位 (PWMCONx<9>) 置 1 选择独立时基模式。

##### 44.5.3.1.2 同步 PWM 操作

多个发生器的 PWM 周期可以通过 MTBS 位 (PWMCONx<3>) 同步到其中一个主控时基。通过将 ITB 位 (PWMCONx<9>) 设置为 0 选择同步时基模式。所有使用多个发生器的中心对齐模式或双周期模式都采用此方法。边沿对齐 PWM 周期也可使用这种方法进行同步。

详情请参见第 44.7 章“PWM 工作模式”。

##### 44.5.3.2 死区寄存器和补偿

每个 PWM 发生器具有两个死区寄存器，以支持具有非对称导通和关断时间的功率器件。通过延迟正在变为有效的边沿插入死区。死区会将非线性失真问题引入产生的波形，因此需要补偿才能重新获得损耗的“有效”时间。补偿的实现是通过关联外部引脚 DTCMPx (FLT<sub>x</sub>) 以及使用其中一个死区寄存器值写入 DTCOMPx 寄存器。将根据此数字输入引脚提高或降低产生的波形的占空比以实现补偿。为获得完全补偿，在 DTCMPx 引脚指示电路存在负和正电流情形时，将向占空比调整 DTCOMPx 寄存器写入 DTRx 和 ALTDTRx 中的值。详情请参见第 44.8.3 章“死区发生”和第 44.17.7 章“死区补偿”章节。

##### 44.5.3.3 占空比寄存器

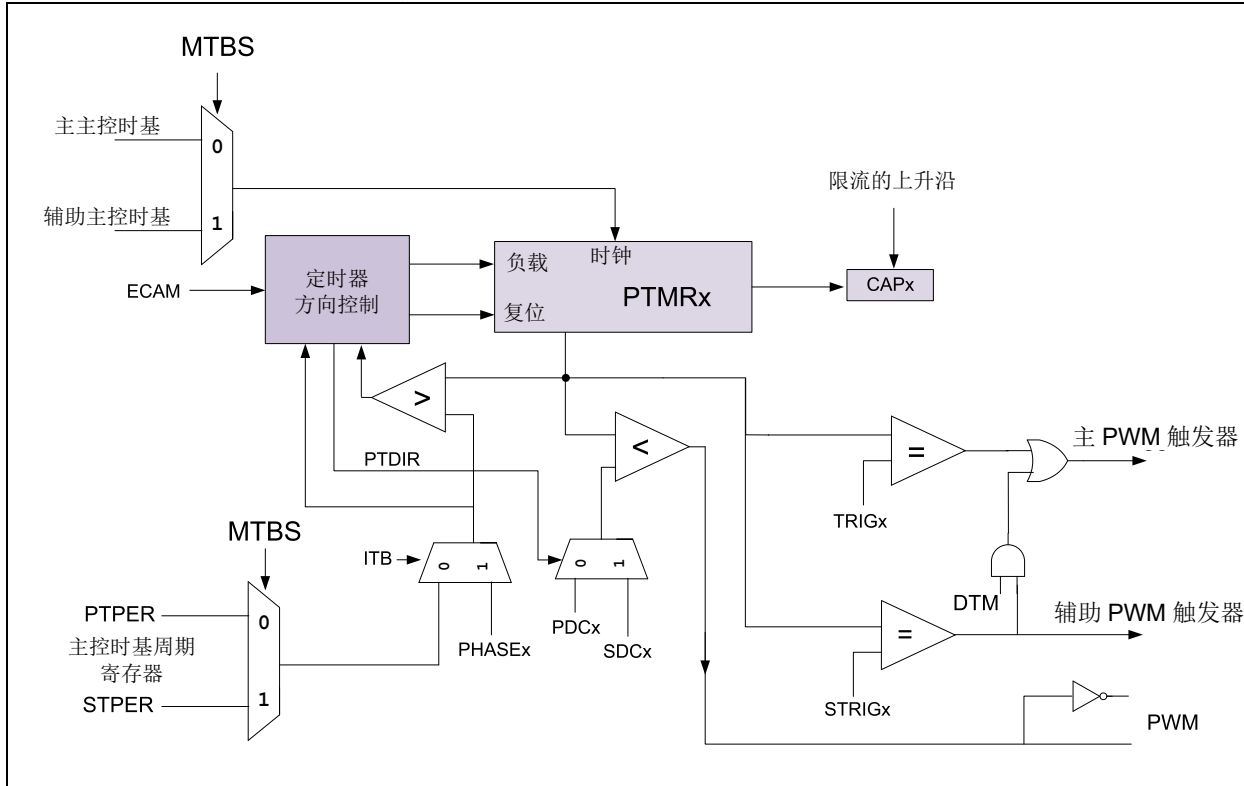
每个 PWM 发生器都具有两个支持不同 PWM 模式的占空比寄存器。主占空比寄存器可指定大多数模式（如 EAM 和 CAM）下的占空比。在非对称 CAM 模式下，占空比由 SDCx 寄存器与 PDCx 寄存器决定。为提高控制带宽，可同时更新或逐个更新这两个寄存器。

##### 44.5.3.4 触发比较寄存器。

每个 PWM 发生器具有两个触发比较寄存器，主触发寄存器 (TRIGx) 和辅助触发寄存器 (STRIGx)，它们可用在每个 PWM 周期内的特定点产生触发。触发可在双触发模式下进一步组合在一起。详情请参见第 44.11 章“PWM 发生器触发”。

图 44-4 说明了每个 PWM 发生器内的触发控制和相关逻辑。

图 44-4: PWM 发生器框图和触发机制



## 44.5.3.5 限流和故障改写

每个发生器都可以与可供选择的 16 个不同源中的一个限流引脚和一个故障引脚相关联。可选择比较器输出作为故障和限流的源。可以选择 PWM 输出的所有边沿，以便在可重新触发的延时计数器（LEBDLYx）的帮助下，屏蔽故障和限流的触发。当检测到故障或过流条件时，之前存储在 FLTDAT<1:0> 位（IOCONx<5:4>）和 CLDAT<1:0> 位（IOCONx<3:2>）的改写数据将自动强制开启 PWM 输出引脚。详情请参见第 44.13 章“PWM 故障”和第 44.14 章“PWM 限流”章节。

## 44.5.3.6 手动改写

每个发生器都具有手动改写功能，该功能可在软件控制下，通过写入 OVRENH 位（IOCONx<8>）和 OVRENH 位（IOCONx<9>），将引脚上的逻辑电平保持为之前写入 OVRDAT<1:0> 位（IOCONx<7:6>）的状态。PWMxH 和 PWMxL 引脚可用软件单独改写。在采用制动或 PWM 转向进行 BLDC 电机的梯形换相的电机控制应用中，此功能很有用。

### 44.5.3.6.1 改写同步

如果 OSYNC 位（IOCONx<0>）置 1，则通过 OVRENH 位、OVRENH 位和 OVRDAT<1:0> 位执行的输出改写将与 PWM 时基进行同步。同步输出改写在时基为 0 时发生。如果 PTEN 位 = 0（表示 PWM 定时器不在运行），对 IOCONx 寄存器的写操作会在下一个 Tcy 边界处生效。

## 44.6 PWM输出状态控制

### 44.6.1 输出使能控制

PWM 高电平和低电平使能位、IOCONx 寄存器中的 PENH 和 PENL 将使能由 MCPWM 模块使用的每个 PWM 输出引脚。

将 B 类熔丝设置为 0（使能）时，引脚的所有权在复位后各不相同。当使能 MCPWM 模块以及引脚的 PWM 功能时，PORTx 和 TRISx 寄存器不再控制引脚。关于 B 类熔丝选择下的引脚行为，请参见表 44-2。

表 44-2: B 类合规性和引脚状态

B 类熔丝设置 PWMLOCK (DEVCFG3<20>)	PTEN 位 (PTCON<15>)	PENH (IOCONx<15>、 PENL (IOCONx<14>)	PWMxH/PWMxL 引脚所有权
1（禁止）	0	0	I/O 控制（复位时输入）
1	0	1	I/O 控制
1	1	0	I/O 控制
1	1	1	PWM 控制
0（使能）	0	0	I/O 控制（复位时输入）
0	0	1	PWM 控制
0	1	0	I/O 控制
0	1	1	PWM 控制

## 44.6.2 输出极性控制

可以通过 POLH 位 (IOCONx<13>) 和 POLL 位 (IOCONx<12>) 将 PWM 输出有效状态选择为高电平有效 (默认设置) 或低电平有效。

触发、死区和边沿敏感消隐总是在引脚上使用极性校正的有效电平。

POLH 和 POLL 位会影响写入 OVRDAT<1:0> (IOCONx<7:6>)、FLTDAT<1:0> (IOCONx<5:4>) 和 CLDAT<1:0> (IOCONx<3:2>) 位的改写数据。无论 PWMxH 和 PWMxL 引脚的极性 (如 POLH 位 (IOCONx<13>) 和 POLL 位 (IOCONx<12>) 位所选择) 如何, MCPWM 模块的内部逻辑状态总是遵循正逻辑 (即高电平有效为级别 1, 反之亦然)。因此, 所有变为有效的状态都将被延时或受死区影响。POLH 和 POLL 的极性翻转可能会在此阶段之后发生。在基于 POLH 和 POLL 位的引脚极性选择写入 FLTDAT<1:0> 和 CLDAT<1:0> 位之前, 所有改写数据都应进行极性校正。例如, 如果 POLL 或 POLH 位为任何引脚定义了低电平有效, 则写入的改写数据应该成为引脚上所需实际改写逻辑状态的补码。

表 44-3 提供了针对不同组合的改写数据和极性选择的值。

表 44-3: 改写数据极性校正

引脚上的改写逻辑状态		输出引脚极性选择		改写数据	
PWMxH	PWMxL	POLH	POLL	OVRDAT<1>、 FLTDAT<1>、 CLDAT<1>	OVRDAT<0>、 FLTDAT<0>、 CLDAT<0>
0	0	0	0	0	0
0	0	0	1	0	1
0	0	1	0	1	0
0	0	1	1	1	1
0	1	0	0	0	1
0	1	0	1	0	0
0	1	1	0	1	1
0	1	1	1	1	0
1	0	0	0	1	0
1	0	0	1	1	1
1	0	1	0	0	0
1	0	1	1	0	1
1	1	0	0	1	1
1	1	0	1	1	0
1	1	1	0	0	1
1	1	1	1	0	0

### 44.6.3 PWM 输出引脚复位状态

对 B 类 (PWMLOCK 位 (DEVCFG3<20>) = 1) 禁止端口引脚时, 退出复位的端口引脚在处于三态的情况下, 由默认设置为输入的 I/O 模块端口 PORT 和 TRIS 控制。用户硬件用于将引脚驱动为无效状态, 并通过相应的上拉和下拉来防止功率器件意外开启。

对 B 类 (PWMLOCK 位 (DEVCFG3<20>) = 0) 使能端口引脚时, 退出复位的端口引脚 (默认的复位状态配置为禁止) 由 MCPWM 模块的 PENH 和 PENL 位控制。因此, 虽然用户硬件可能具有防止产生直通条件的相应上拉和下拉, 但在将 PENH 和 PENL 位置 1 前更改 TRIS 位可能会带来不良后果。

关于复位时 PWM 引脚状态的所有权, 请参见[表 44-2](#)。

44.7 PWM工 作模式

MCPWM 模块支持以下工作模式：

- 推挽输出模式
- 互补输出模式
- 冗余输出模式

这些工作模式可以使用 PWM x I/O 引脚模式位 PMOD<1:0>（IOCONx<11:10>）进行选择。

**注：** 所有 PWMxL 引脚都具有 PWMxH 备用功能。如需要，可通过使能 PWMxL 引脚上的 PWMxH 备用功能，同时禁止 PWMxL 功能，实现独立输出。关于 PWMxL 引脚上的备用 PWM 输出功能，请参见具体器件数据手册中的“电机控制 PWM（MCPWM）”章节。

以下几节介绍了多个操作模式下的 PWM 输出。表 44-4 提供了可用模式和设置的列表，并通过编号引用图形。

表 44-4： 模式和编码交叉引用表

PWM 模式	模式设置	相关 图片
推挽	独立占空比和相位，固定周期，边沿对齐	<a href="#">44-5</a>
	独立占空比和周期，无相移，边沿对齐	<a href="#">44-6</a>
	独立占空比和周期，无相移，中心对齐模式	<a href="#">44-7</a>
互补	独立占空比和相位，固定周期，边沿对齐	<a href="#">44-8</a>
	独立占空比和周期，无相移，边沿对齐	<a href="#">44-9</a>
	独立占空比和周期，无相移，中心对齐	<a href="#">44-10</a>
冗余	独立占空比和相位，固定周期，边沿对齐	<a href="#">44-11</a>
	独立占空比和周期，无相移，边沿对齐	<a href="#">44-12</a>

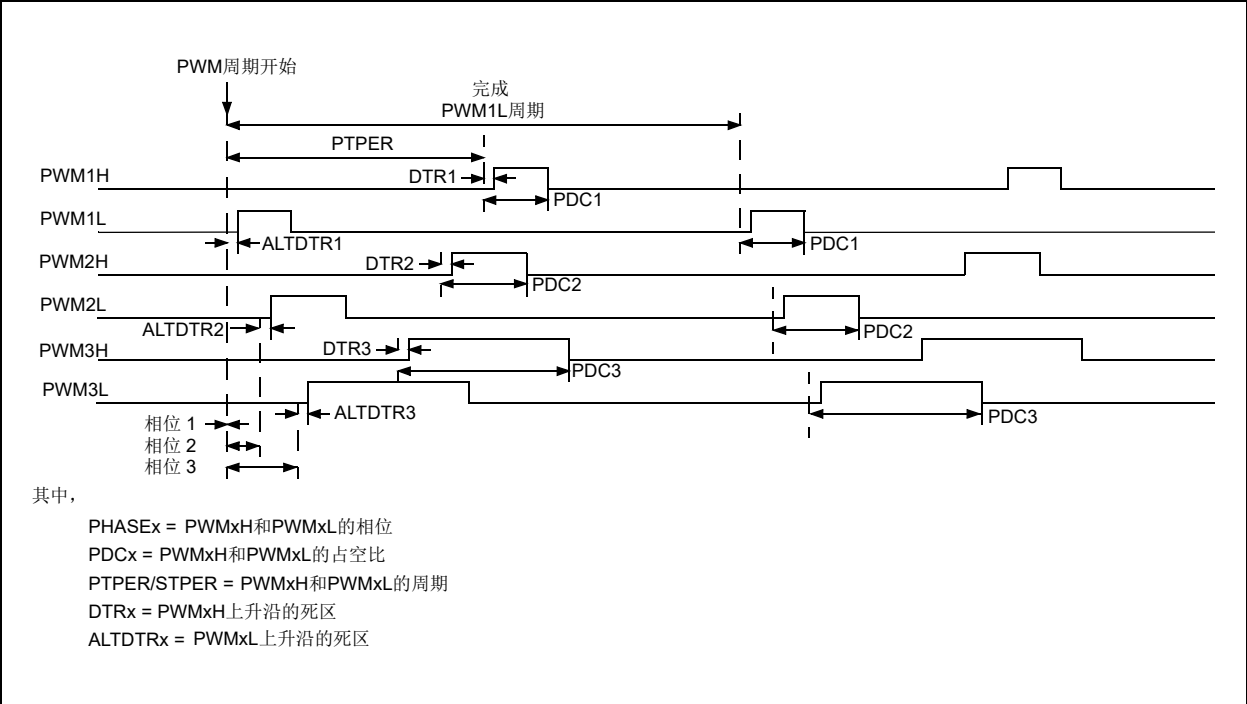
44.7.1 推挽 PWM 模式

**注：** 本章节的“图形和代码示例”中列出的所有功能和寄存器并非在所有器件上均可用。关于可用性，请参见具体器件数据手册中的“电机控制 PWM（MCPWM）”章节。

在推挽模式下，PWM 输出在 PWMxH 和 PWMxL 引脚上交替提供。第 44.20 章“应用信息”中提供了推挽模式的一些典型应用。

图 44-5 至图 44-7 以及例 44-2 至例 44-4 说明了最常用配置中推挽 PWM 模式的 PWM 输出。

图 44-5： 推挽 PWM 模式 —— 独立占空比和相位，固定周期，边沿对齐



例 44-2: 推挽 PWM 模式 —— 独立占空比和相位, 固定主周期, 边沿对齐

```
/* Set PWM Period on Primary Time Base */
PTPER = 1000;

/* Set Phase Shift */
PHASE1 = 0;
PHASE2 = 100;
PHASE3 = 200;

/* Set Duty Cycles */
PDC1 = 150;
PDC2 = 200;
PDC3 = 400;

/* Set Dead Time Values */
DTR1 = DTR2 = DTR3 = 25;
ALTDTR1 = ALTDTR2 = ALTDTR3 = 25;

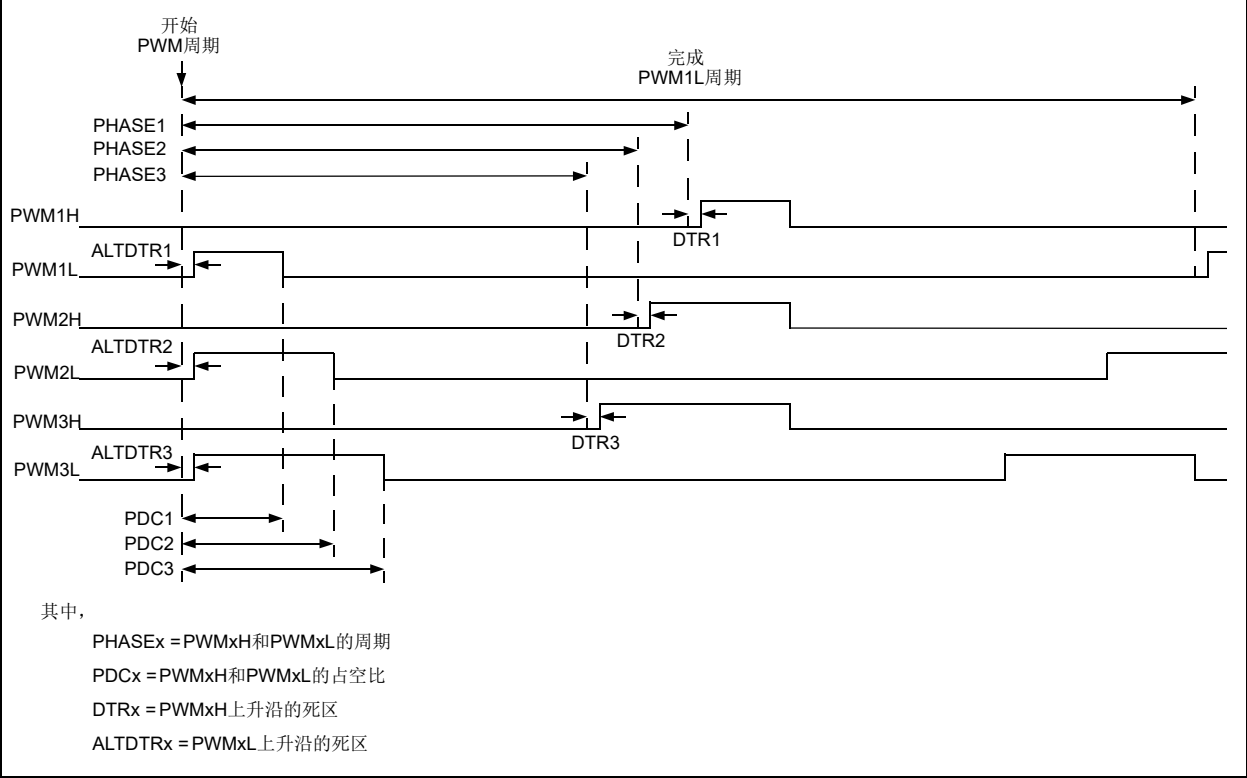
/* Set Primary Time Base, Edge-Aligned mode and Independent Duty Cycles */
PWMCON1 = PWMCON2 = PWMCON3 = 0x00000000;

/* Set PWM Mode to Push-Pull and Fault mode to disabled */
IOCON1 = IOCON2 = IOCON3 = 0x0003C800;

/* SYSCLK is the clock source...1:1 Prescaler */
PTCON = 0x0000;

/* Enable PWM Module */
PTCON = 0x8000
```

图 44-6: 推挽 PWM 模式 —— 独立占空比和周期, 无相移, 边沿对齐





### 例 44-3: 推挽 PWM 模式 —— 独立占空比和独立周期, 无相移, 边沿对齐

```
/* Set PWM Periods on PHASEx Registers */
PHASE1 = 1000;
PHASE2 = 900;
PHASE3 = 800;

/* Set Duty Cycles */
PDC1 = 200;
PDC2 = 300;
PDC3 = 400;

/* Set Dead Time Values */
DTR1 = DTR2 = DTR3 = 25;
ALTDTR1 = ALTDTR2 = ALTDTR3 = 25;

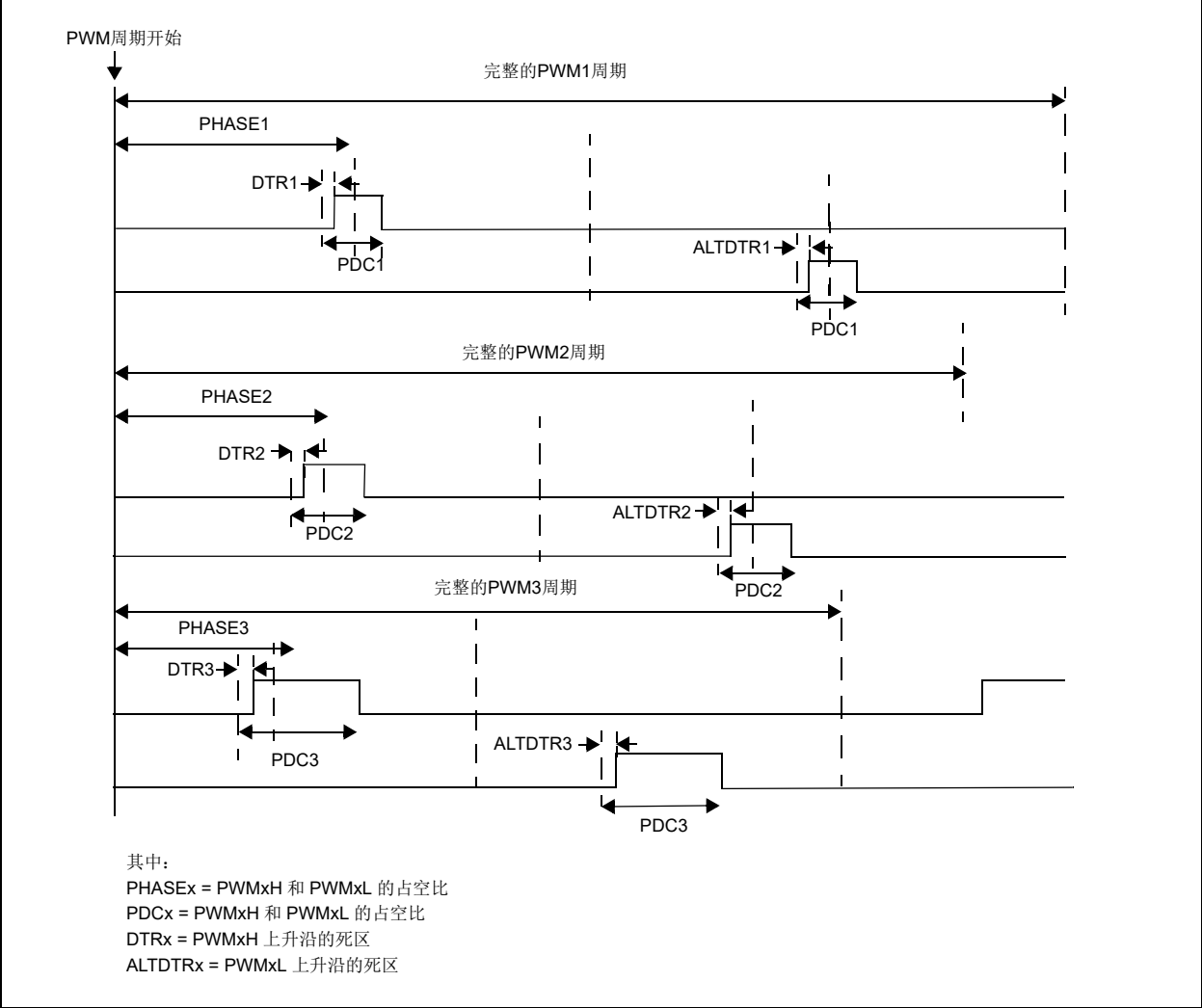
/* Set Independent Time Base, Edge-Aligned mode and Independent Duty Cycles */
PWMCON1 = PWMCON2 = PWMCON3 = 0x0200;

* Set PWM Mode to Push-Pull and Fault mode to disabled */
IOCON1 = IOCON2 = IOCON3 = 0x3C800;

/* 1:1 Prescaler */
PTCON = 0x0000;

/* Enable PWM Module */
PTCON = 0x8000;
```

图 44-7： 推挽 PWM 模式 —— 独立占空比和周期，无相移，中心对齐模式



### 例 44-4: 推挽 PWM 模式——独立占空比和独立周期，无相移，中心对齐模式

```
/* Set PWM Periods on PHASEx Registers */
PHASE1 = 1000;
PHASE2 = 900;
PHASE3 = 800;

/* Set Duty Cycles */
PDC1 = 200;
PDC2 = 300;
PDC3 = 400;

/* Set Dead Time Values */
DTR1 = DTR2 = DTR3 = 25;
ALTDTR1 = ALTDTR2 = ALTDTR3 = 25;

/* Set Independent Time Base, Center-Aligned mode and Independent Duty Cycles */
PWMCON1 = PWMCON2 = PWMCON3 = 0x00000600;

* Set PWM Mode to Push-Pull and Fault mode to disabled */
IOCON1 = IOCON2 = IOCON3 = 0x0003C800;

/* 1:1 Prescaler */
PTCON = 0x00000000;

/* Enable PWM Module */
PTCON = 0x8000;
```

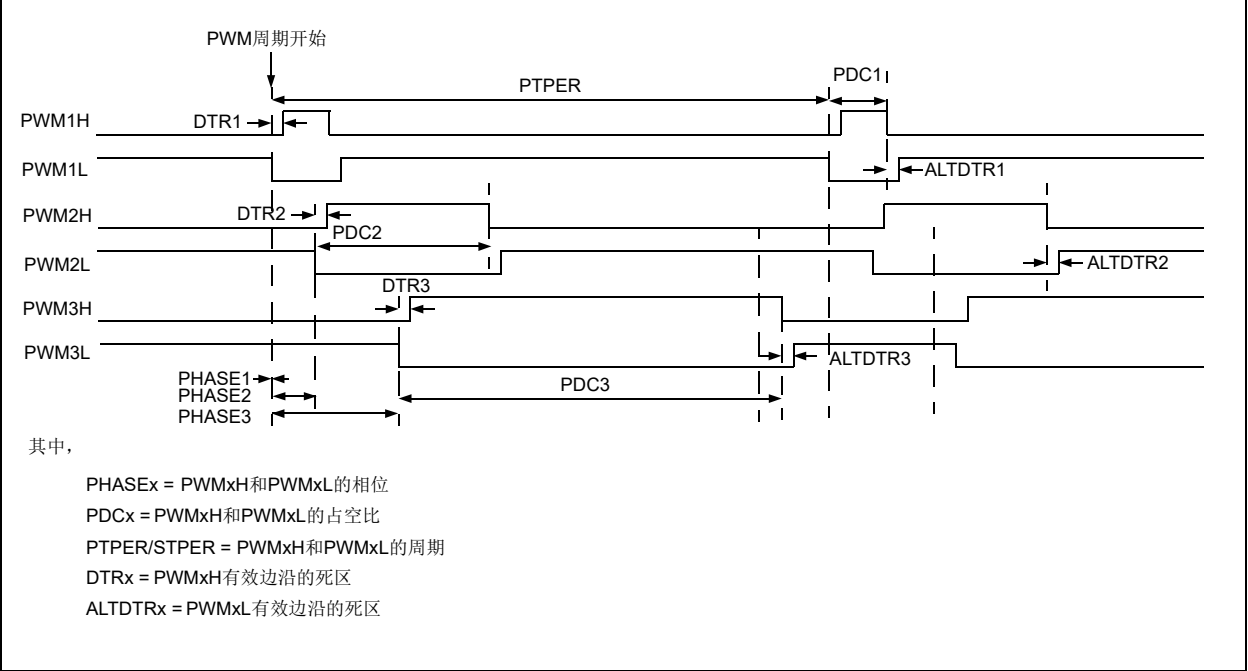
44.7.2 互补 PWM 模式

注： 本章节的“图形和代码示例”中列出的所有功能和寄存器并非在所有器件上均可用。关于可用性，请参见具体器件数据手册中的“电机控制 PWM（MCPWM）”章节。

在互补 PWM 模式下，PWM 输出 PWMxH 是 PWMxL 输出的互补信号。第 44.20 章“应用信息”中提供了互补 PWM 模式的一些典型应用。

图 44-8 至图 44-10 以及例 44-5 至例 44-7 说明了互补 PWM 模式下不同常用配置中的 PWM 输出。

图 44-8： 互补 PWM 模式 —— 独立占空比和相位，固定周期，边沿对齐



例 44-5： 互补 PWM 模式 —— 独立占空比和相位，主控时基（主和辅助），边沿对齐

```
/* Set PWM Period on Primary Time Base */
PTPER = 1000;

/* Set Phase Shift */
PHASE1 = 0;
PHASE2 = 100;
PHASE3 = 200;

/* Set Duty Cycles */
PDC1 = 150;
PDC2 = 200;
PDC3 = 400;

/* Set Dead Time Values */
DTR1 = DTR2 = DTR3 = 25;
ALTDTR1 = ALTDTR2 = ALTDTR3 = 25;

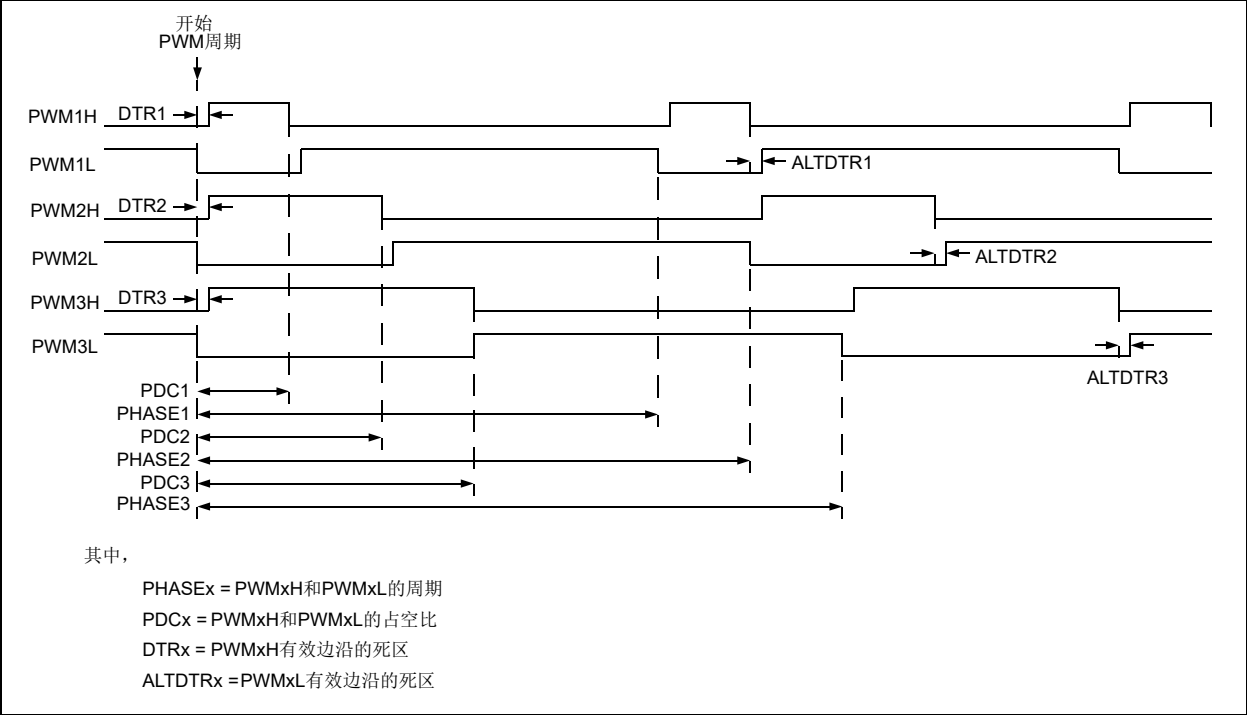
/* Set Primary Time Base, Edge-Aligned mode and Independent Duty Cycles */
PWMCON1 = PWMCON2 = PWMCON3 = 0x00000000;

* Set PWM Mode to Complimentary and Fault mode to disabled */
IOCON1 = IOCON2 = IOCON3 = 0x0003C000;

/* 1:1 Prescaler */
PTCON = 0x0000;

/* Enable PWM Module */
PTCON = 0x8000;
```

图 44-9： 互补 PWM 模式 —— 独立占空比和周期，无相移，边沿对齐



例 44-6: 互补 PWM 模式 —— 独立占空比和独立周期，无相移，边沿对齐

```
/* Set PWM Periods on PHASEx Registers */
PHASE1 = 800;
PHASE2 = 900;
PHASE3 = 1000;

/* Set Duty Cycles */
PDC1 = 200;
PDC2 = 300;
PDC3 = 400;

/* Set Dead Time Values */
DTR1 = DTR2 = DTR3 = 25;
ALTDTR1 = ALTDTR2 = ALTDTR3 = 25;

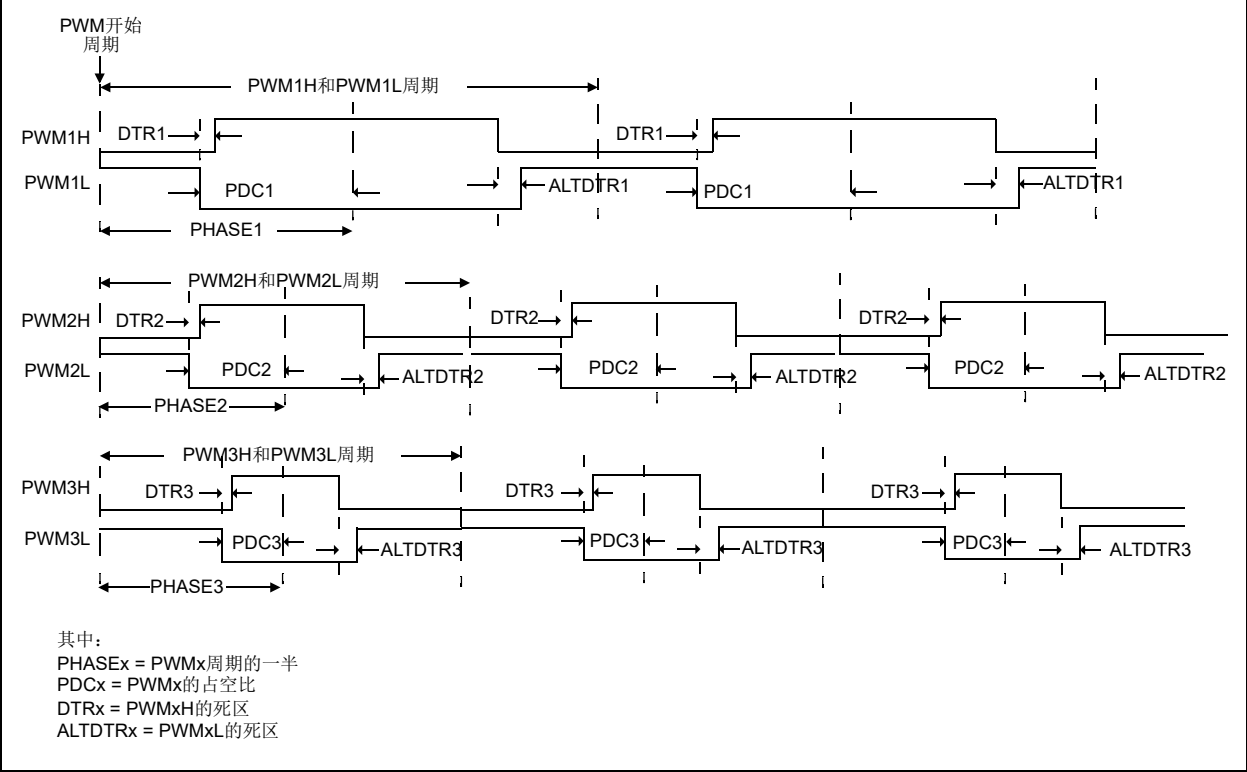
/* Set Independent Time Base, Edge-Aligned mode and Independent Duty Cycles */
PWMCON1 = PWMCON2 = PWMCON3 = 0x00000200;

* Set PWM Mode to Complimentary and Fault mode to disabled */
IOCON1 = IOCON2 = IOCON3 = 0x0003C000;

/* 1:1 Prescaler */
PTCON = 0x0000;

/* Enable PWM Module */
PTCON = 0x8000;
```

图 44-10: 互补 PWM 模式 —— 独立占空比和周期，无相移，中心对齐



### 例 44-7: 互补 PWM 模式 —— 独立占空比和独立周期, 无相移, 中心对齐

```
/* Set PWM Periods on PHASEx Registers */
PHASE1 = 1000;
PHASE2 = 900;
PHASE3 = 800;

/* Set Duty Cycles */
PDC1 = 400;
PDC2 = 300;
PDC3 = 200;

/* Set Dead Time Values */
DTR1 = DTR2 = DTR3 = 25;
ALTDTR1 = ALTDTR2 = ALTDTR3 = 25;

/* Set Independent Base, Center-Aligned mode and Independent Duty Cycles */
PWMCON1 = PWMCON2 = PWMCON3 = 0x00000600;

* Set PWM Mode to Complementary and Fault mode to disabled */
IOCON1 = IOCON2 = IOCON3 = 0x0003C000;

/* 1:1 Prescaler */
PTCON = 0x0000;

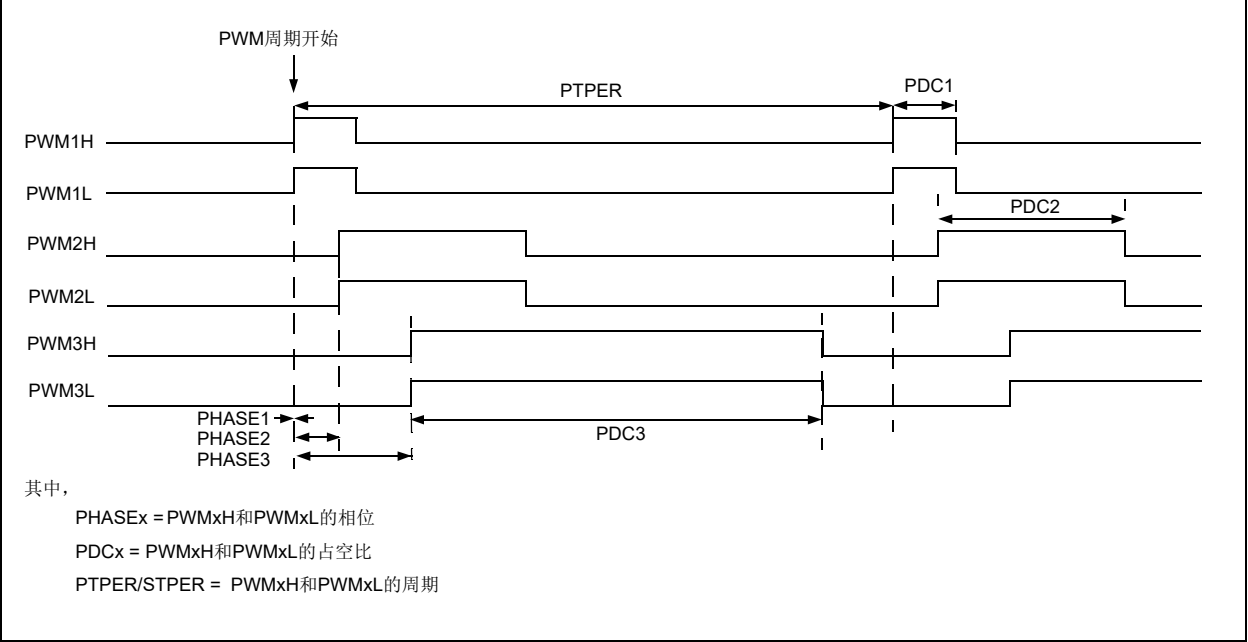
/* Enable PWM Module */
PTCON = 0x8000;
```

44.7.3 冗余 PWM 输出模式

**注：** 本章节的“图形和代码示例”中列出的所有功能和寄存器并非在所有器件上均可用。关于可用性，请参见具体器件数据手册中的“电机控制 PWM（MCPWM）”章节。

在冗余 PWM 输出模式下，对于每个 PWM 引脚对（PWMxH 和 PWMxL），MCPWM 模块可以提供单端 PWM 输出信号的两个副本。此模式使用 PWM 生成的占空比寄存器（PDCx）来指定占空比。在此输出模式下，除非用户应用程序通过 IOCONx 寄存器中的 OVRENH、OVRENL 和 OVRDAT<1:0> 位指定了改写值，以产生针对特定应用（比如开关磁阻电机）的波形，否则两个 PWM 输出引脚将提供相同的 PWM 信号。死区发生和补偿在此模式下没有任何影响。图 44-11、例 44-8、图 44-12 和例 44-9 说明了一些常用配置中的冗余 PWM 输出模式。

图 44-11： 冗余 PWM 模式 —— 独立占空比和相位，固定周期，边沿对齐





例 44-8： 冗余 PWM 模式 —— 独立占空比和相位，边沿对齐

```
/* Set PWM Period on Primary Time Base */
PTPER = 1000;

/* Set Phase Shift */
PHASE1 = 0;
PHASE2 = 100;
PHASE3 = 200;

/* Set Duty Cycles */
PDC1 = 150;
PDC2 = 200;
PDC3 = 400;

/* Set Dead Time Values */
DTR1 = DTR2 = DTR3 = 0;
ALTDTR1 = ALTDTR2 = ALTDTR3 = 0;

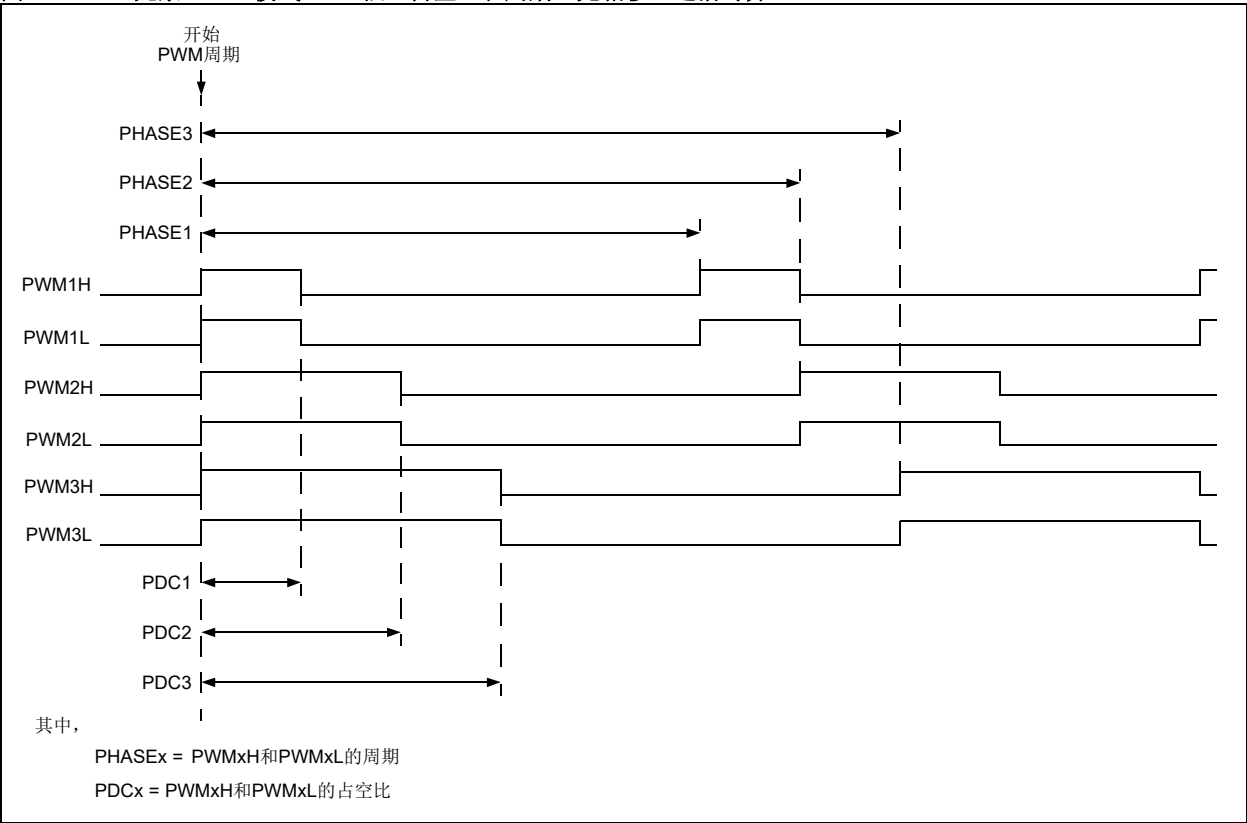
/* Set Primary Time Base, Edge-Aligned mode and Independent Duty Cycles */
PWMCON1 = PWMCON2 = PWMCON3 = 0x00000000;

* Set PWM Mode to Redundant and Fault mode to disabled */
IOCON1 = IOCON2 = IOCON3 = 0x0003C400;

/* 1:1 Prescaler */
PTCON = 0x0000;

/* Enable PWM Module */
PTCON = 0x8000;
```

图 44-12： 冗余 PWM 模式 —— 独立占空比和周期，无相移，边沿对齐



例 44-9: 冗余 PWM 模式 —— 独立占空比和独立周期，无相移

```
/* Set PWM Periods on PHASEx Registers */
PHASE1 = 800;
PHASE2 = 900;
PHASE3 = 1000;

/* Set Duty Cycles */
PDC1 = 200;
PDC2 = 300;
PDC3 = 400;

/* Set Dead Time Values */
DTR1 = DTR2 = DTR3 = 0;
ALTDTR1 = ALTDTR2 = ALTDTR3 = 0;

/* Set Independent Time Base, Edge-Aligned mode and Independent Duty Cycles */
PWMCON1 = PWMCON2 = PWMCON3 = 0x00000200;

* Set PWM Mode to Redundant and Fault mode to disabled */
IOCON1 = IOCON2 = IOCON3 = 0x0003C400;

/* 1:1 Prescaler */
PTCON = 0x0000;

/* Enable PWM Module */
PTCON = 0x8000;
```

表 44-5 列出了对应于 PWM 模式的 PWM 寄存器功能。

表 44-5: 互补、推挽和冗余模式的寄存器功能

函数	PWMCONx 中的配置			寄存器
	ITB	MTBS	ECAM<1:0>	
占空比	x	x	00 和 01	PDCx
	x	x	10 和 11	PDCx/SDCx
相移	0	x	xx	PHASEx
周期	0	0	xx	PTPER
	0	1	xx	STPER
	1	x	xx	PHASEx

图注: x = 无关

## 44.7.4 真正独立输出模式

MCPWM 模块在硬件中不支持真正独立输出模式。但是，两个单独发生器的两个 PWMxH 引脚可以使用相同的结果通过共同的主控时基进行同步（请参见注）。

**注：** 只有不同发生器的 PWMxH 和 PWMxL 在相同引脚上进行复用的器件才支持此功能。这是一个特定于器件的功能，无法始终用于所有器件。与此模式相关的支持的详细信息，请参见具体器件数据手册中的“电机控制 PWM (MCPWM)”章节。

如果使用 PWM 备用 I/O 引脚选择位 CFGCON<23:18> 执行真正独立模式功能，可通过可重新编程逻辑使能两个 PWMxH 引脚。

例 44-10 说明了在死区被禁止并同步到相同主控时基，并且 PHASEx 寄存器提供波形之间的相对偏移的情况下，通过独立发生器的两个 PWMxH 引脚实现的真正独立功能。

## 例 44-10: 为真正独立操作配置的 PWM 通道

```
/* Set PWM Period on Primary Time Base */
PTPER = 1000;

/* Set relative offsets between the PWM waveforms to PERIOD/2 */
PHASE1 = 0;
PHASE7 = 500; /* One-half of PWM period */

/* Set Symmetric Duty Cycles */
PDC1 = 300;
PDC7 = 300;

/* Set Dead time control to disabled and Edge-Aligned mode enabled */
PWMCON1 = PWMCON7 = 0x00000080;

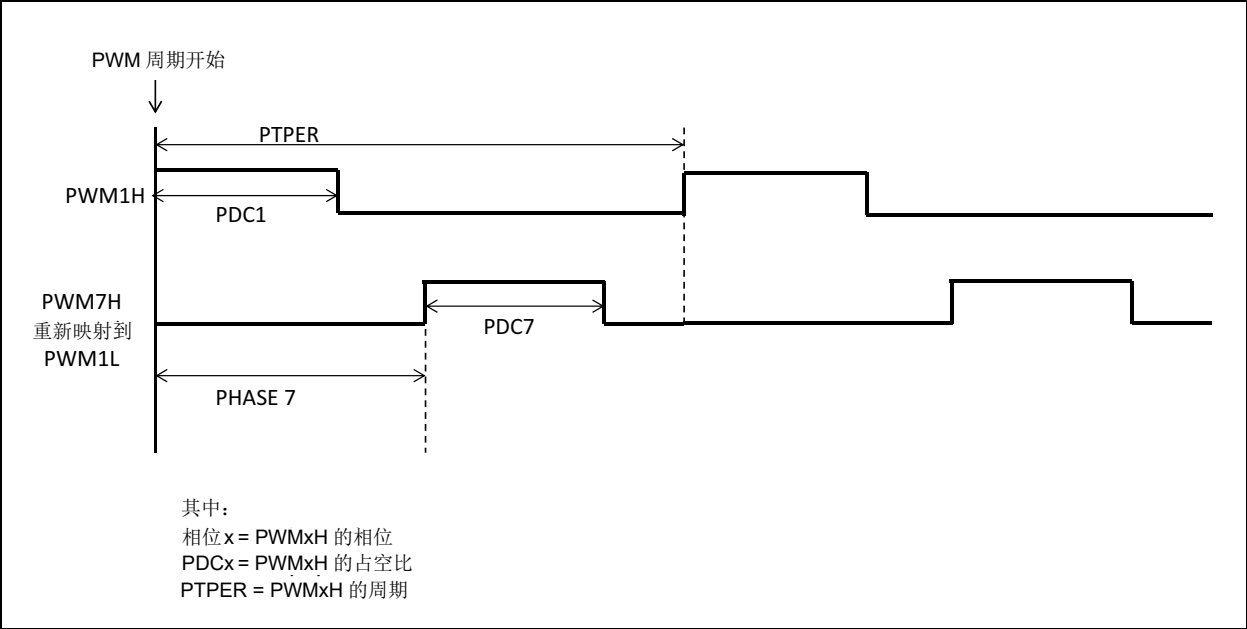
/* Set PWM mode to Redundant output mode and fault mode to disabled */
/* Set PWMxL to I/O Function */
IOCON1 = IOCON7 = 0x00038400;

/* Write Unlock Sequence to allow write access to CFGCON register */
SYSKEY = 0xAA996655;
SYSKEY = 0x556699AA;
/*Remap PWM7H to PWM1L*/
CFGCONbits.PWMAPIN1=1;
/*Lock Write access to CFGCON register */
SYSKEY = 0;

/* Set Primary Time Base */
/* 1:1 Prescaler */
PTCON = 0x0000;

/* Enable PWM Module */
PTCON = 0x8000;
```

图 44-13： 真正独立 PWM 模式 —— 独立占空比，固定周期，相移，边沿对齐



## 44.8 PWM 发生

**注：** 本章节中列出的所有寄存器和功能并非在所有器件上均可用。关于可用性，请参见具体器件数据手册中的“电机控制 PWM (MCPWM)”章节。

本章节介绍 PWM 发生器的功能。

### 44.8.1 PWM 周期

PWM 周期值定义 PWM 脉冲的开关频率。PWM 周期值可以通过 PTPER/STPER 寄存器控制，也可以通过独立时间周期寄存器 PHASEx 控制（分别用于主和辅助 PWM 输出）。

PWM 周期值可以用两种方式控制

- 在 PWM 主控时基模式（ITB 位（PWMCONx<9>）= 0）下运行 PWM 发生器时，周期取决于 PTPER（MTBS 位（PWMCONx<3>）= 0）或 STPER 寄存器（MTBS = 1）
- 在独立时基模式（ITB = 1）下时，周期取决于 PHASEx 寄存器

关于各种 PWM 模式及其特性的详细信息，请参见第 44.7 章“PWM 工作模式”。

当 MCPWM 模块在主控时基模式下工作时，PTPER 寄存器存放一个 16 位值，该值可指定主控时基定时器的计数周期。当 MCPWM 模块在独立时基模式下工作时，PHASEx 寄存器存放一个 16 位值，该值可指定 PTMRx 定时器的计数周期。用户应用程序可以在任意时刻更新定时器周期。边沿对齐 PWM 模式（ECAM 位 PWMCONx<11:10> = 0b00）设置为 0 下的 PWM 时间周期（PTPER/STPER/PHASEx）可以使用公式 44-1 确定。

**公式 44-1： 边沿对齐模式的周期寄存器值计算**

$$PTPER, STPER, PHASEx = \frac{FSYSCLK}{F_{PWM} \times PWM \text{ 输入时钟预分频器}}$$

其中，

$F_{PWM}$  = 所需的 PWM 频率

$FSYSCLK$  = 系统时钟频率

$PWM \text{ 输入时钟预分频比}$  = PCLKDIV<2:0> 位（PTCON<6:4>）和 SCLKDIV<2:0> 位（STCON<6:4>）中定义的值

在主 / 辅助主控时基（即 ITB 位（PWMCONx<9>）= 0）下工作时，周期值将分别装入 PTPER/STPER 寄存器中。在独立时基（PHASEx 寄存器）（即 ITB 位（PWMCONx<9>）= 1）下工作时，周期值将装入 PHASEx 寄存器中。基于公式 44-1，将装入的值如公式 44-2 所示。

**公式 44-2： 边沿对齐模式的周期寄存器值计算**

$$PTPER, STPER, PHASEx = \frac{120MHz}{20kHz \times 1} = 6000$$

其中，

$20 \text{ kHz}$  = 所需的 PWM 开关频率

$1:1$  = PWM 输入时钟预分频比

$120 \text{ MHz}$  = 系统时钟（FSYSCLK）

中心对齐模式（ECAM 位（PWMCONx<11:10> = 01、10、11））下的 PWM 时间周期（PTPER/STPER/PHASEx）可以通过公式 44-3 确定。

**公式 44-3： 中心对齐模式下的周期寄存器值计算**

$$PTPER, STPER, PHASEx = \frac{FSYSCLK}{FPWM \times PWM \text{ 输入时钟预分频器} \times 2}$$

其中，

$FSYSCLK$  = 系统时钟频率

$FPWM$  = 所需的 PWM 频率

在主 / 辅助主控时基（即 ITB 位（PWMCONx<9>）= 0）下工作时，周期值将分别装入 PTPER/STPER 寄存器中。在独立时基（PHASEx 寄存器）（即 ITB 位（PWMCONx<9>）= 1）下工作时，周期值将装入 PHASEx 寄存器中。基于公式 44-3，将装入的值如公式 44-4 所示。

**公式 44-4： 中心对齐模式下的周期寄存器值计算**

$$PTPER, STPER, PHASEx = \frac{120MHz}{20 \text{ kHz} \times 1 \times 2} = 3000$$

其中，

$PWM$  频率（ $FPWM$ ）= 20 kHz

$PWM$  输入时钟预分频比 = 1:1

系统时钟频率（ $FSYSCLK$ ）= 120 MHz

对于边沿对齐模式，可获得的最大 PWM 周期分辨率为  $1/FSYSCLK$ ；对于中心对齐模式，则为  $1/2 * FSYSCLK$ 。PCLKDIV<2:0> 位（PTCON<6:4>）和 SCLKDIV<2:0> 位（STCON<6:4>）对 PWM 时钟进行预分频。可以通过置 1 或清零 PWM 模块使能位 PTEN（PTCON<15>）来使能或禁止定时器或计数器。主主控时基定时器也可以通过清零 PTEN 位进行清零。例 44-11 提供了用于时钟预分频比选择的代码。例 44-12 提供了用于 PWM 时间周期选择的代码。例 44-13 提供了用于 PWM 时间周期初始化的代码。

**例 44-11： 时钟预分频比选择**

```
/* Select PWM time base input clock prescaler */
/* Choose divide ratio of 1:2 */
```

```
PTCONbits.PCLKDIV = 1;
```

**例 44-12： PWM 时间周期选择**

```
/* Select Time Base Period Control */
/* Choose one of these options */
```

```
PWMCON1bits.ITB = 0; /* PTPER provides the PWM time period value */
PWMCON1bits.ITB = 1; /* PHASEx/SPHASEx provides the PWM time period value */
```

**例 44-13： PWM 时间周期初始化**

```
/* PWM frequency is 20 kHz */
/* Choose one of the following options */
```

```
PTPER = 2000;
```

```
PHASEx = 2000;
```

## 44.8.2 PWM 占空比

PWMxH 和 PWMxL 输出的占空比由 PDCx 和 SDCx 寄存器中的值决定。这些寄存器的功能随 PWM 工作模式而变化。

- 边沿对齐模式：PDCx 寄存器决定 PWM 占空比
- 中心对齐模式：在此模式下，写入 PDCx 寄存器的值将被复制到 SDCx 寄存器。PDCx 和 SDCx 寄存器会保存与本地定时器 (PTMRx) 相匹配的比较值。
- 非对称中心对齐模式：在此模式下，不同的值将分别写入 PDCx 和 SDCx 寄存器。当 PTDIR = 1 时，PDCx 值决定边沿跳变，而当 PTDIR = 0 时，SDCx 值决定边沿跳变。

对于不同的对齐模式，占空比可以通过公式 44-5 至公式 44-7 确定。

**公式 44-5: 边沿对齐模式的 PDCx 计算**

$$PDCx = \frac{FSYSCLK}{F_{PWM} \times PWM \text{ 输入时钟预分频器}} \times \text{所需占空比}$$

其中，

$F_{PWM}$  = PWM 频率

PWM 输入时钟预分频比 = PCLKDIV<2:0> 位 (PTCON<6:4>) 和 SCLKDIV<2:0> 位 (STCON<6:4>) 中定义的值

所需占空比 = 介于 0 和 1 之间对应于所需占空比的值

**公式 44-6: 中心对齐模式的 PDCx 计算**

$$PDCx = FSYSCLK \times \frac{(\text{所需占空比})}{(F_{PWM} \cdot 2 \cdot PWM \text{ 输入时钟预分频器})}$$

其中，

$F_{PWM}$  = PWM 频率

PWM 输入时钟预分频比 = PCLKDIV<2:0> 位 (PTCON<6:4>) 和 SCLKDIV<2:0> 位 (STCON<6:4>) 中定义的值

所需占空比 = 介于 0 和 1 之间对应于所需占空比的值

**公式 44-7: 非对称中心对齐模式的 PDCx 和 SDCx 计算**

$$PDCx = FSYSCLK \times \frac{(\text{占空比 1})}{(F_{PWM} \cdot 2 \cdot PWM \text{ 输入时钟预分频器})}$$

$$SDCx = FSYSCLK \times \frac{(\text{占空比 2})}{(F_{PWM} \cdot 2 \cdot PWM \text{ 输入时钟预分频器})}$$

其中，

$F_{PWM}$  = PWM 频率

PWM 输入时钟预分频比 = PCLKDIV<2:0> 位 (PTCON<6:4>) 和 SCLKDIV<2:0> 位 (STCON<6:4>) 中定义的值

占空比 x = 介于 0 和 1 之间的值

**注 1:** 如果占空比值大于或等于周期值，则 PWMxH 的占空比将为 100%，PWMxL 的占空比将为 0%。

**2:** 如果将占空比值指定为零，则 PWMxH 的占空比将为 0%，PWMxL 的占空比将为 100%。

**3:** 如果不满足  $PDCx > DTRx$  的条件，这可能使 PWMxH 始终保持低电平。

## 44.8.2.1 占空比分辨率

边沿对齐模式的 PWM 占空比分辨率可通过[公式 44-8](#) 确定，而中心对齐模式的 PWM 占空比分辨率则可以通过[公式 44-9](#) 确定。[例 44-14](#) 提供了用于 PWM 占空比初始化的代码。

### 公式 44-8: 边沿对齐模式的位分辨率计算

$$\text{位分辨率} = \log_2 \left[ \frac{FSYSCLK}{F_{PWM} \times PWM \text{ 输入时钟预分频器}} \right]$$

### 公式 44-9: 中心对齐模式的位分辨率计算

未经测试的代码——仅供参考

$$\text{位分辨率} = \log_2 \left[ \frac{FSYSCLK}{F_{PWM} \times PWM \text{ 输入时钟预分频器} \times 2} \right]$$

### 例 44-14: PWM 占空比初始化

```
/*Initialize PWM Period Value*/  
PTPER = 2000;  
/* Initialize PWM Duty Cycle Value */  
  
PDC1 = 1000; /* Duty Cycle is 50% of the period */
```

## 44.8.2.2 占空比限制

在中心对齐和边沿对齐模式下，如果 PDCx 值为 0，则输出将保持无效状态，这对应于 0% 的占空比。同样，如果 PDCx 值等于或大于周期（PTPER、STPER 或 PHASEx）寄存器，则输出将保持有效状态，这对应于 100% 的占空比。

在非对称中心对齐模式下，PDCx 和 SDCx 寄存器都必须写入 0 或周期值，才能产生 0% 和 100% 的占空比。



### 44.8.3 死区发生

死区指由 PWM 死区寄存器（DTRx）或 PWM 备用死区寄存器（ALTDTRx）指定的可编程时间段，死区寄存器会阻止一个 PWM 输出变为有效，直到其互补 PWM 信号已保持无效一段指定的时间。

死区寄存器可以指定 16384 乘以 FSYSCLOCK 的总持续时间。功率器件通常具有非对称导通和关断时间，因而要求进入和退出导通的不同器件具有单独的死区寄存器。

#### 44.8.3.1 死区模式

以下是三种死区控制模式：

- 正死区
- 负死区
- 禁止死区

死区可以使用公式 44-10 中给出的公式确定。

#### 公式 44-10：死区计算

$$ALTDTRx, DTRx = FSYSCLOCK \times \frac{\text{所需死区}}{\text{PWM 输入时钟预分频器}}$$

#### 44.8.3.2 正死区

正死区模式定义 PWMxH 和 PWMxL 输出均置为无效的时间段。当需要通过有效或同步整流来负载再循环电流时，此模式很有用。这类似于“先断后合”开关。当指定正死区模式时，DTRx 寄存器指定 PWMxH 输出的死区，ALTDTRx 寄存器指定 PWMxL 输出的死区。图 44-14 和图 44-15 说明了此模式。

正死区的工作原理是使用预编程的死区延时值延迟正在变为有效的 PWM 输出的边沿。可以为所有互补模式指定正死区，而且即使在故障、限流和手动改写状态期间，正死区也处于有效状态。作为一般规则，正死区将保持最小值，以避免过热和谐波失真。

#### 44.8.3.3 负死区

负死区模式定义 PWMxH 和 PWMxL 输出均置为有效的时间段。对于需要在功率晶体管进行开关时为电流提供通路的电流馈入电源转换拓扑，该模式很有用。这类似于“先合后断”开关。当指定负死区模式时，DTRx 寄存器指定 PWMxH 输出的负死区，ALTDTRx 寄存器指定 PWMxL 输出的负死区。边沿对齐模式下仅支持负死区。在中心对齐模式下，负死区选择会使死区发生器失效。

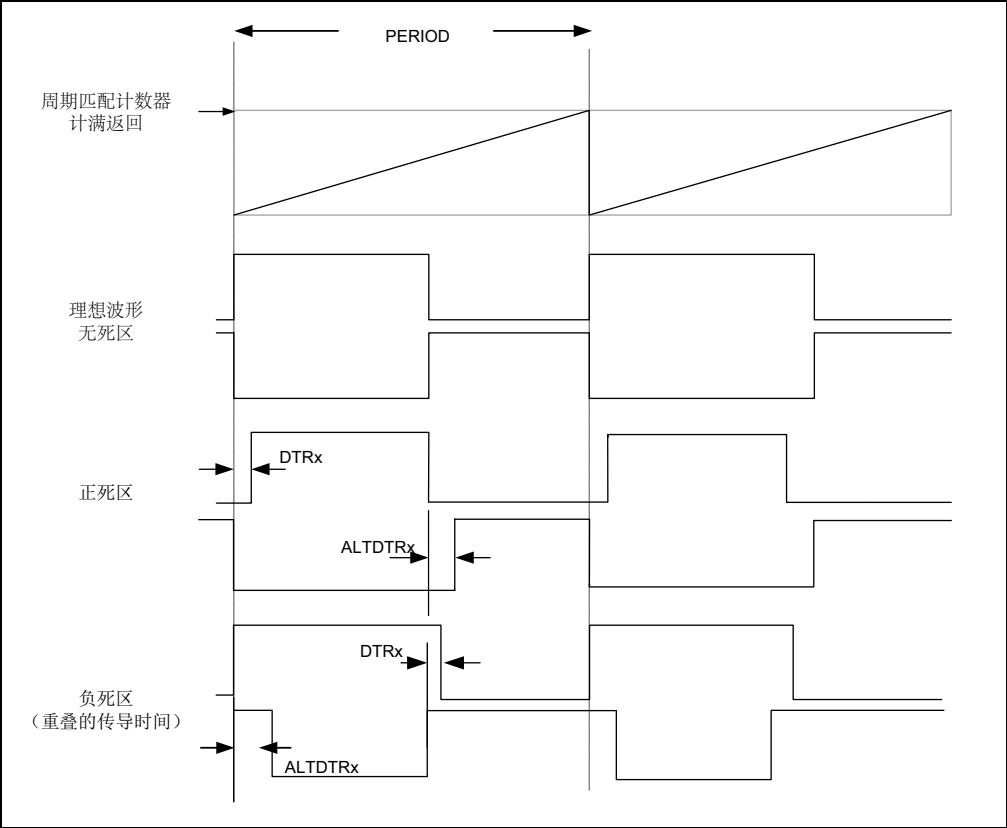
负死区通过延迟将 PWM 置为无效状态的边沿发生。一些电源转换技术（比如电流源转换器）需要一定限度的直通电流。只能为边沿对齐 PWM 模式指定负死区，负死区不会影响 PWMxH 和 PWMxL 引脚的改写输出状态。图 44-14 说明了此模式。

44.8.3.4 禁止死区

对于每个 PWM 发生器，可以禁止死区逻辑。死区功能通过  $DTC<1:0>$  位（ $PWMCONx<7:6>$ ）进行控制。

图 44-14 说明了边沿对齐模式下 PWM 输出时的正死区和负死区发生。

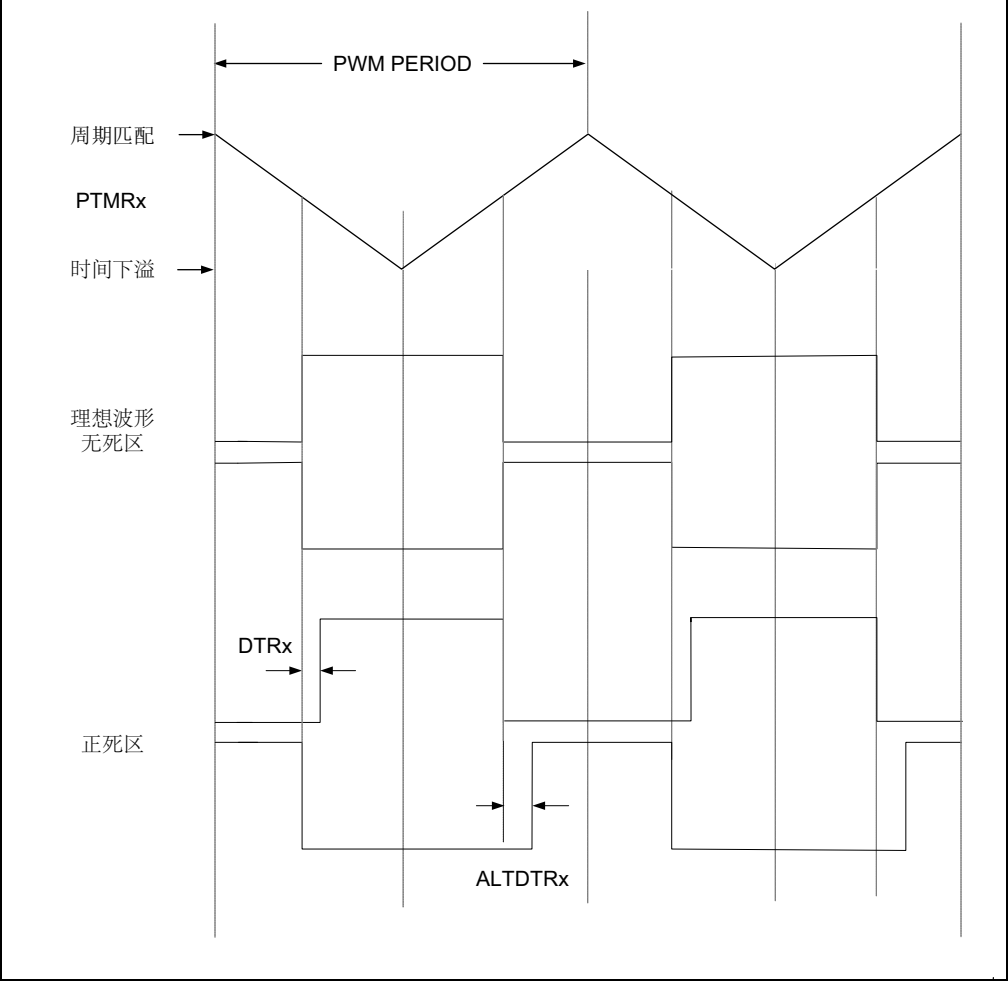
图 44-14： 边沿对齐模式的死区波形



对于每个 PWM 发生器，可以禁止死区功能。死区功能通过  $DTC<1:0>$  位（ $PWMCONx<7:6>$ ）进行控制。

图 44-15 说明了中心对齐模式的正死区发生。

图 44-15: 中心对齐模式的死区波形



## 44.8.4 死区范围

每个死区单元提供的死区持续时间通过在  $DTRx$  和  $ALTDTRx$  寄存器中指定一个无符号值来设置。在使用  $SYSClk$  占空比分辨率的最高工作时钟频率下，死分辨率率为  $SYSClk$ （ $SYSClk$  为 120 MHz 时为 8.33 ns）。当在 120 MHz  $SYSClk$  工作时，最大死区值为  $(2^{14} - 1) SYSClk = 136.52 \mu s$ 。

## 44.8.5 死区失真

对于小于或等于  $DTRx$  寄存器值的  $PDCx$  占空比值， $PWMxH$  占空比将保持在 0%；但是， $PWMxL$  将钳位到周期值 -  $PDCx$  -  $ALTDTRx$ （周期值是  $PTPER$ 、 $STPER$  或  $PHASEx$  寄存器给定的值）的占空比。当  $PWMxH$  显示的占空比为 0%，且  $PWMxL$  显示的占空比为 100% 时，此条件不同于  $PDCx = 0$  的规范。当  $PDCx$  占空比大于或等于周期值减去  $ALTDTRx$  寄存器值，但不等于或大于周期值时，也存在类似的情况。此条件会将  $PWMxH$  占空比钳位到  $PDCx - DTRx$ ，并将  $PWMxL$  钳位到 0% 占空比。当  $PWMxH$  的占空比保持在 100%，且  $PWMxL$  的占空比保持在 0% 时，此条件仍不同于大于或等于周期值的  $PDCx$ 。占空比极值下的此行为，虽然仅是受控制百分比的一小部分，但也可能导致其他线性占空比控制发生失真或不连续。此非线性会给控制环带来一些挑战，并且会使一些不连续 PWM 方案的发生变得困难。

由于死区而产生的失真不仅仅在占空比极值时发生。由于死区会减少或增加导通时间，因此完整的调制波形（逆变器输出基频，取决于电流方向）导通周期会造成伏秒积（伏特 \* 秒）失衡。关于因死区而产生的失真影响，请参见图 44-56。死区失真影响在占空比接近死区值时更加显著，因为损耗的导通时间在总导通时间中的占比变大。

死区在实际的逆变器中不可避免，但是，可以在很大程度上减少因此产生的失真。死区补偿是一种要求在硬件中检测电流方向信息的常见技术。PWM 控制中的非线性表现为一种控制失衡，而且控制算法或控制环尝试将这种失真影响降低到一定程度，使是否转而采取其他方法变得无关紧要。为在这些应用中保持低失真度，并仍然将受控参数（比如上升时间和稳定时间）保持在规范值范围内，如可能，将避免使控制占空比贸然接近这些极值情况。并非所有应用都可使用此解决方法。通过对死区寄存器进行运行时更新的死区补偿是恢复完全线性的唯一方法，如第 44.8.6 章“死区补偿”中所述。

图 44-16 说明了在调制电压导通周期不同时间的两个死区的影响，以及所产生的电流失真。所示功率器件通过基于电流方向（如箭头所示）的不同通路传输电流。如果通过 H 桥的电流方向为负，则在理想情况下，在持续时间 A 和 B 期间电流会流过上桥臂，而不必通过下桥臂 FET 发生任何导通。下桥臂应在时间 D 开始导通，该时间是一个死区，会使电流通过上桥臂，并施加  $V_{BUS}$ ，而非  $V_{SS}$ ，从而造成波形的伏秒失真。类似失真还会出现在带正方向电流的导通前沿上。为清晰起见，放大了电压和电流波形的失真影响。

图 44-16: 死区失真的影响

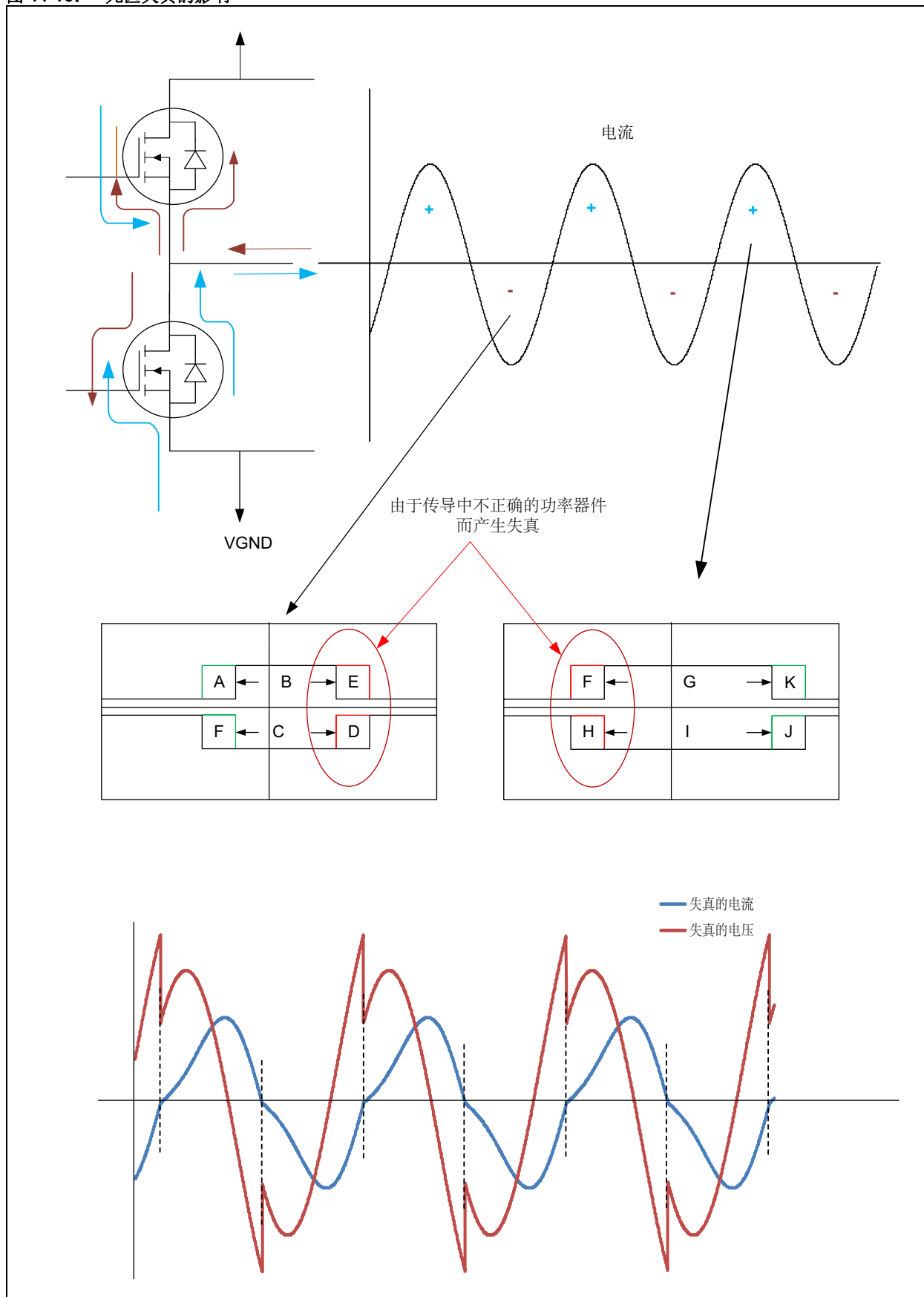


表 44-6 说明了不同占空比（PDCx）规范的 PWM 占空比的非线性或突然转换。如需要，死区寄存器可逐渐写入越来越小的值，以重新恢复线性。这样做不同于死区补偿，后者依赖于基于电流方向的调整。更多信息，请参见第 44.8.6 章“死区补偿”。

表 44-6: 占空比失真以及与死区的关系

PDCx 规范	占空比	
	PWMxH	PWMxL
0	0%	100%
≥ 周期	100%	0%
< DTRx	0%	周期 – PDCx – ALTDTRx
> 周期 – ALTDTRx	PDCx – DTRx	0%

## 44.8.6 死区补偿

当 H 桥将电流切换到通过其功率晶体管时，将插入死区，以避免直通。需要产生此死区，并附带有关通过半桥的电流方向的信息。产生死区的传统方法是延迟上升沿，以避免导通重叠。此方法对电压波形有负面影响，会在死区成为总周期重要一部分时导致电流失真 / 谐波。这是因为续流二极管只会以单一方向导通，因此，死区损耗的导通时间取决于电流方向。

MCPWM 模块中的死区补偿会通过加上或减去 DTRx 和 ALTDTRx 值（取决于通过 DTCMPy 引脚获取的电流方向）恢复损耗的导通时间（见注）。每个发生器都有一个与其关联的 DTCMPy 引脚。死区失真会导致电机应用中产生的扭矩发生变化，这会影响控制系统的稳定性以及电机的性能，尤其是在低速和低扭矩工作区。向总谐波失真添加谐波内容会导致高压机中的功率因数降低。

**注：** DTCMPy 引脚与 FLTx 输入引脚相关联，且与 GPIO 引脚共用。关于 x-y 关联，请参见具体器件的数据手册。因此所有 FLTx 引脚都可以用软件进行查询或控制。当 TRISx 引脚置 1 时，可用软件查询状态。当 TRISx 引脚清零时，软件可按需将状态置为有效 / 无效。当通过用户软件控制故障输入时，应特别注意。如果故障引脚的 TRISx 寄存器的相应位清零，就无法从外部驱动故障输入。

死区补偿模式会使能外部信号（DTCMPx）修改占空比，以克服由于死区而引起的电机电流失真。

当通过死区控制位 DTC<1:0>（PWMCONx<7:6>）选择死区补偿模式时，外部输入信号 DTCMPx 将导致 DTCOMPx 寄存器中的值与 PDCx/SDCx 寄存器指定的占空比相加或相减。

仅提供了单一死区补偿寄存器（DTCOMPx）。可向 DTCOMPx 寄存器装入等于 DTRx 或 ALTDTRx 的固定值，但在两个死区差异非常大时，将无法完全补偿。要进行完全补偿，软件可以监控软件中的 DTCMPx 行，并相应地改变 DTCOMPx 寄存器值，如表 44-7 中所提供。

**表 44-7： 更新 DTCOMPx 寄存器值，以进行完全死区补偿**

DTCMPx 引脚的极性	DTCMPx 引脚上的逻辑电平	DTCOMPx 寄存器的补偿值
0	0	DTCOMPx = ALTDTRx
0	1	DTCOMPx = DTRx
1	0	DTCOMPx = DTRx
1	1	DTCOMPx = ALTDTRx

图 44-17 说明了具有正死区配置的边沿对齐模式中 PWM 输出的死区补偿。

**图 44-17: 使用死区补偿的边沿对齐正死区**

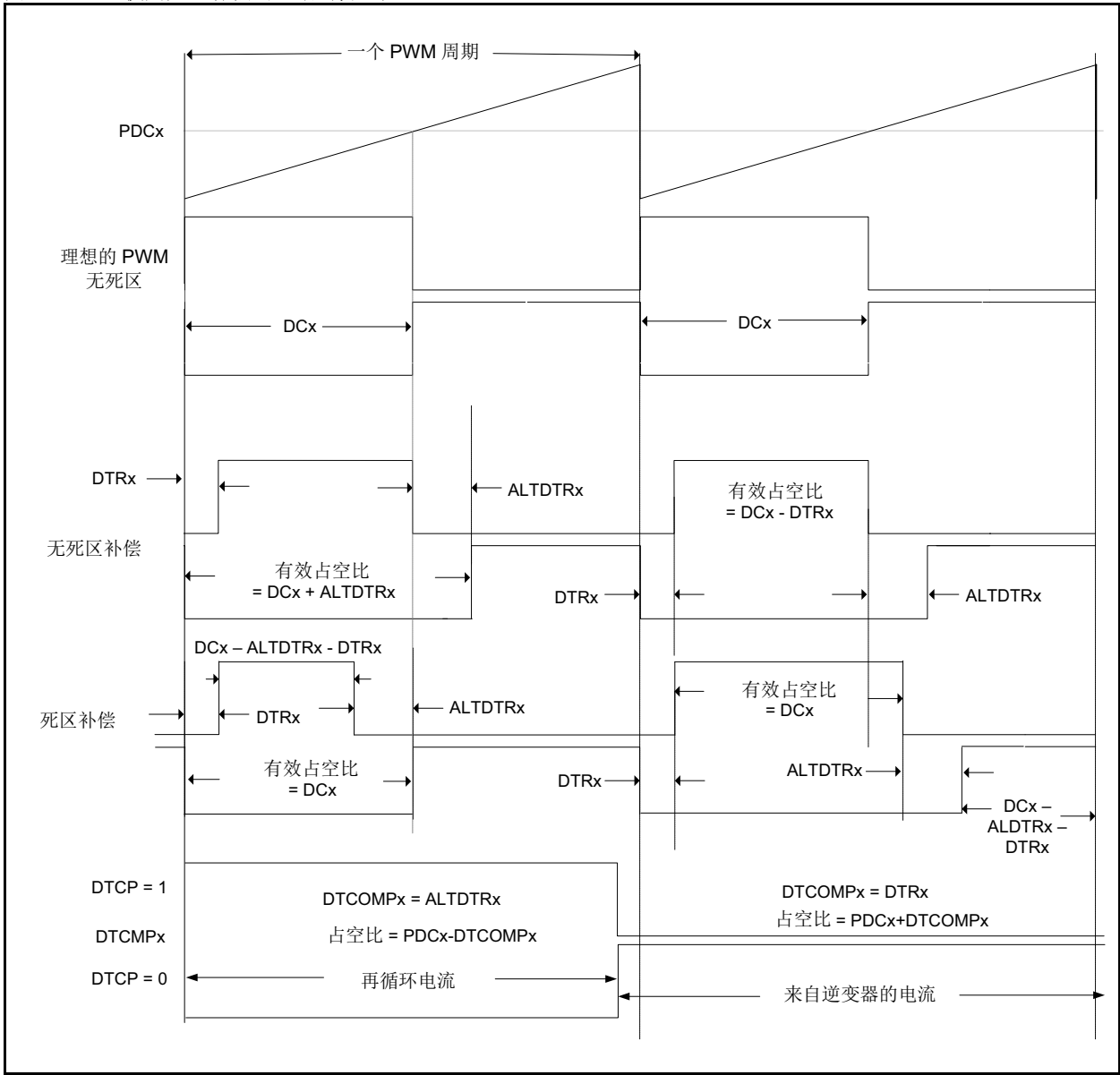




图 44-18 说明了中心对齐模式的死区补偿。

图 44-18: 使用死区补偿的中心对齐正死区

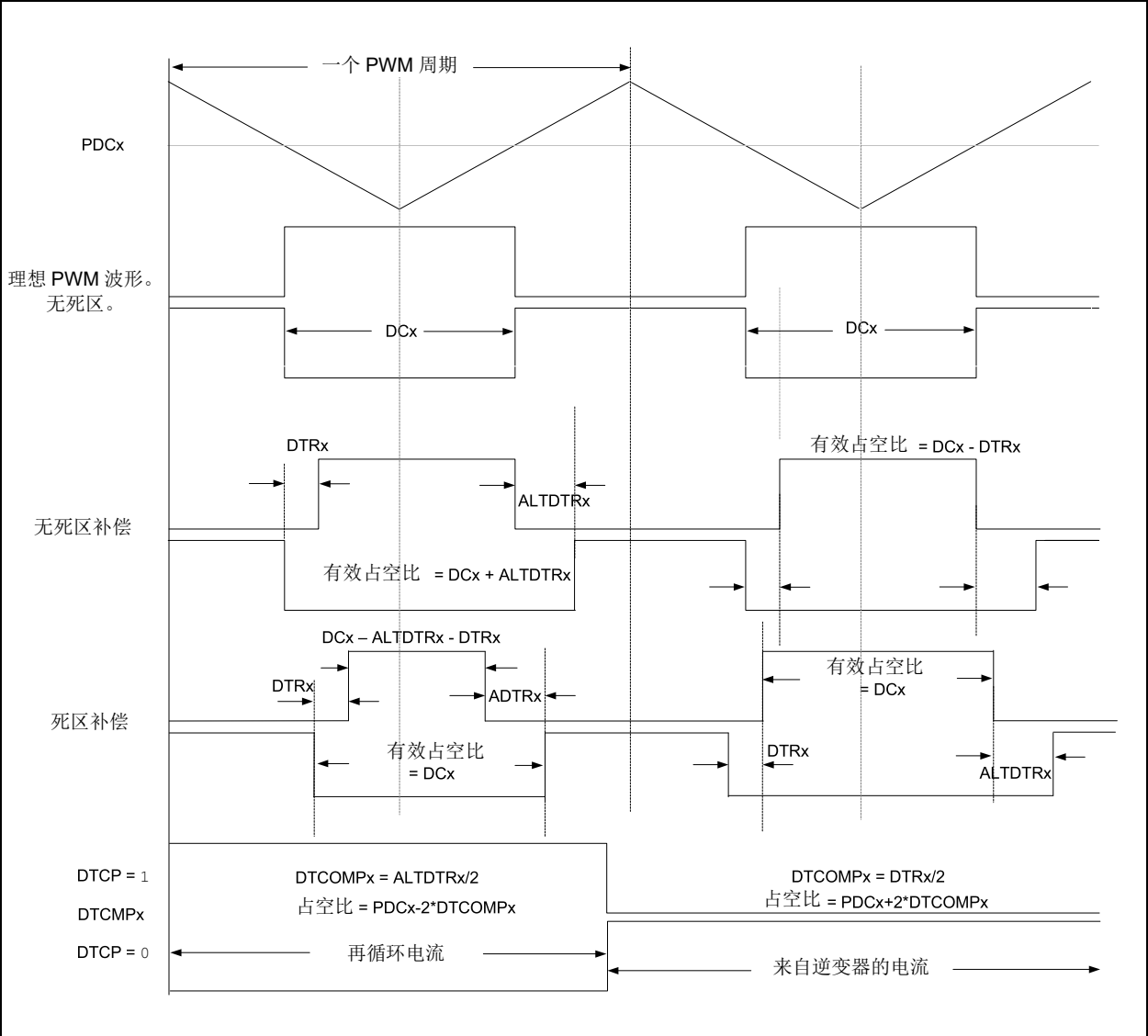


表 44-8 提供了边沿对齐和中心对齐模式中的正死区补偿规则。补偿规则不适用于使用负死区或推挽模式的任何模式。

表 44-8: 死区补偿规则 (1,2,3,4)

边沿对齐模式中的引脚	$DTCMPx = DTPOL$ , $DTCOMPx = ALTDTRx$	$DTCMPx \neq DTPOL$ , $DTCOMPx = DTRx$
PWMxH 占空比	$PDCx - DTRx - DTCOMPx$	$PDCx - DTRx + DTCOMPx$
PWMxL 占空比	周期 - ( $PDCx - DTCOMPx + ALTDTRx$ )	周期 - ( $PDCx + DTCOMPx + ALTDTRx$ )
对 PWMxH 的实际效果	减小	已补偿 = $PDCx$
对 PWMxL 的实际效果	已补偿 = 周期 - $PDCx$	减小

中心对齐模式中的引脚	$DTCMPx = DTPOL$ $DTCOMPx = ALDTRx / 2$	$DTCMPx \neq DTPOL$ , $DTCOMPx = DTRx / 2$
PWMxH 占空比	$2 * PDCx - DTRx - 2 * DTCOMPx$	$2 * PDCx - DTRx + 2 * DTCOMPx$
PWMxL 占空比	周期 - ( $2 * PDCx - 2 * DTCOMPx + ALTDTRx$ )	周期 - ( $2 * PDCx + 2 * DTCOMPx + ALTDTRx$ )
对 PWMxH 的实际效果	减小	已补偿 = $2 * PDCx$
对 PWMxL 的实际效果	已补偿 = 周期 - $2 * PDCx$	减小

- 注 1: 死区补偿可不适用于负死区模式和推挽模式。
- 2:  $DTCMPx$  是一个数字输入引脚。
- 3:  $DTPOL$  是一个 SFR 位。
- 4:  $DTCOMPx$  是死区补偿 SFR。

死区补偿外部输入控制引脚  $DTCMPx$  在每个死区周期的起始点进行采样。这可以尽可能确保补偿是准确的。 $DTCMPx$  引脚将决定占空比是加上还是减去  $DTCOMPx$  寄存器中指定的数量。

- 注 1:  $DTCOMPx$  值必须小于占空比 ( $PDCx$ ) 寄存器的值的一半, 否则可能产生无法预测的行为。
- 2: 死区补偿不适用于为 0% 的占空比。在这种情况下, PWM 输出将保持为 0%, 而不管  $DTCMPx$  输入引脚处于什么状态。

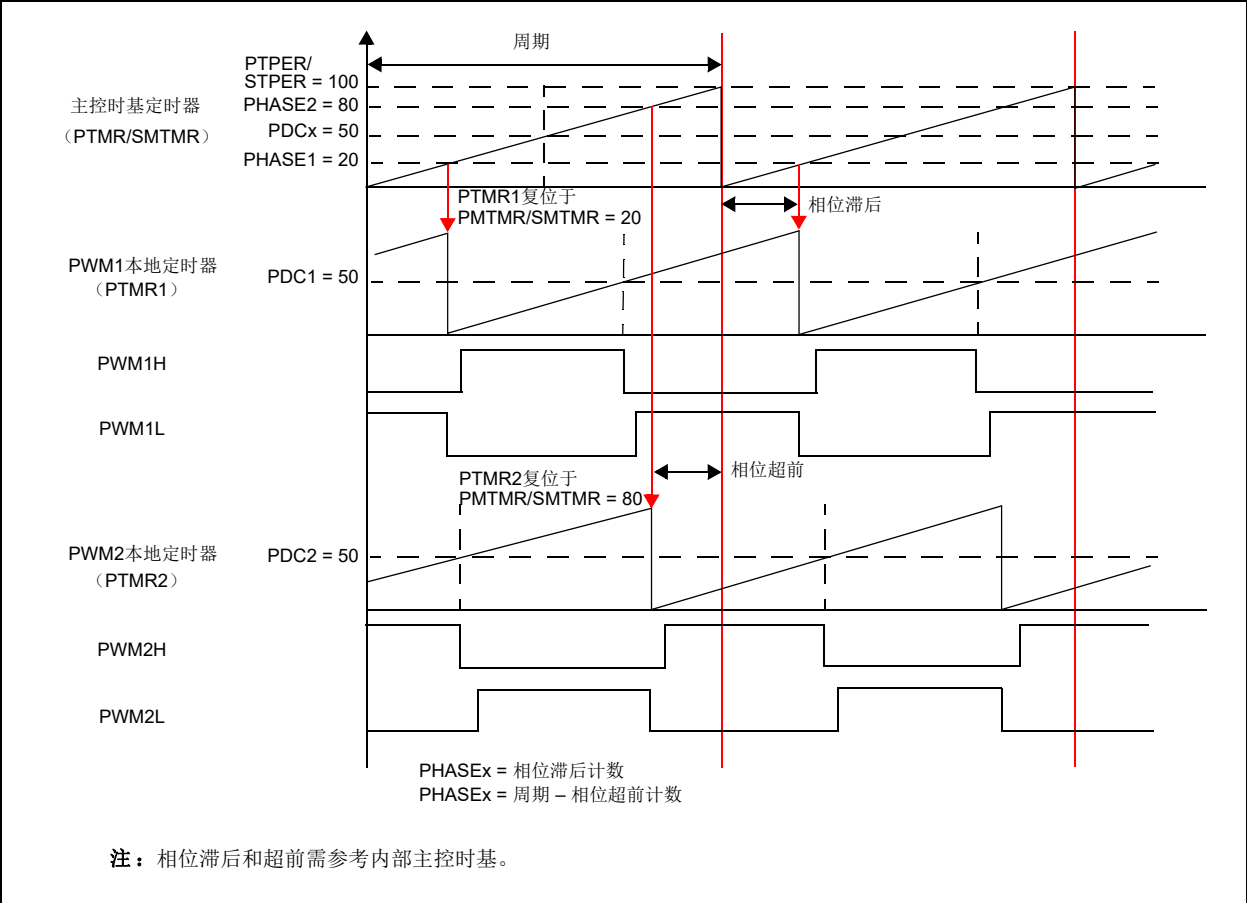
44.8.7 相移

当PWM发生器进行了同步操作配置时，写入PHASEx寄存器的相移值会按照主控时基对PWMxH和PWMxL 引脚进行移位。

为实现多个发生器输出之间的完美同步，需要在使能 MCPWM 模块（即，当 PTEN 位（PTCON<15>）= 0 时）将相移寄存器（PHASEx）更新为零。

图 44-19 和图 44-20 说明了分别在边沿对齐和中心对齐模式中相对于彼此进行相移的两个不同 PWM 通道 / 发生器输出。

图 44-19： 相移（边沿对齐模式）



相位偏移值可以为 0 与 PTPER 和 STPER 寄存器值之间的任意值。PHASEx 值大于周期值时，它们会被视为等于周期的值。要产生大于周期的相移是不可能的。

图 44-20： 中心对齐模式下的相移

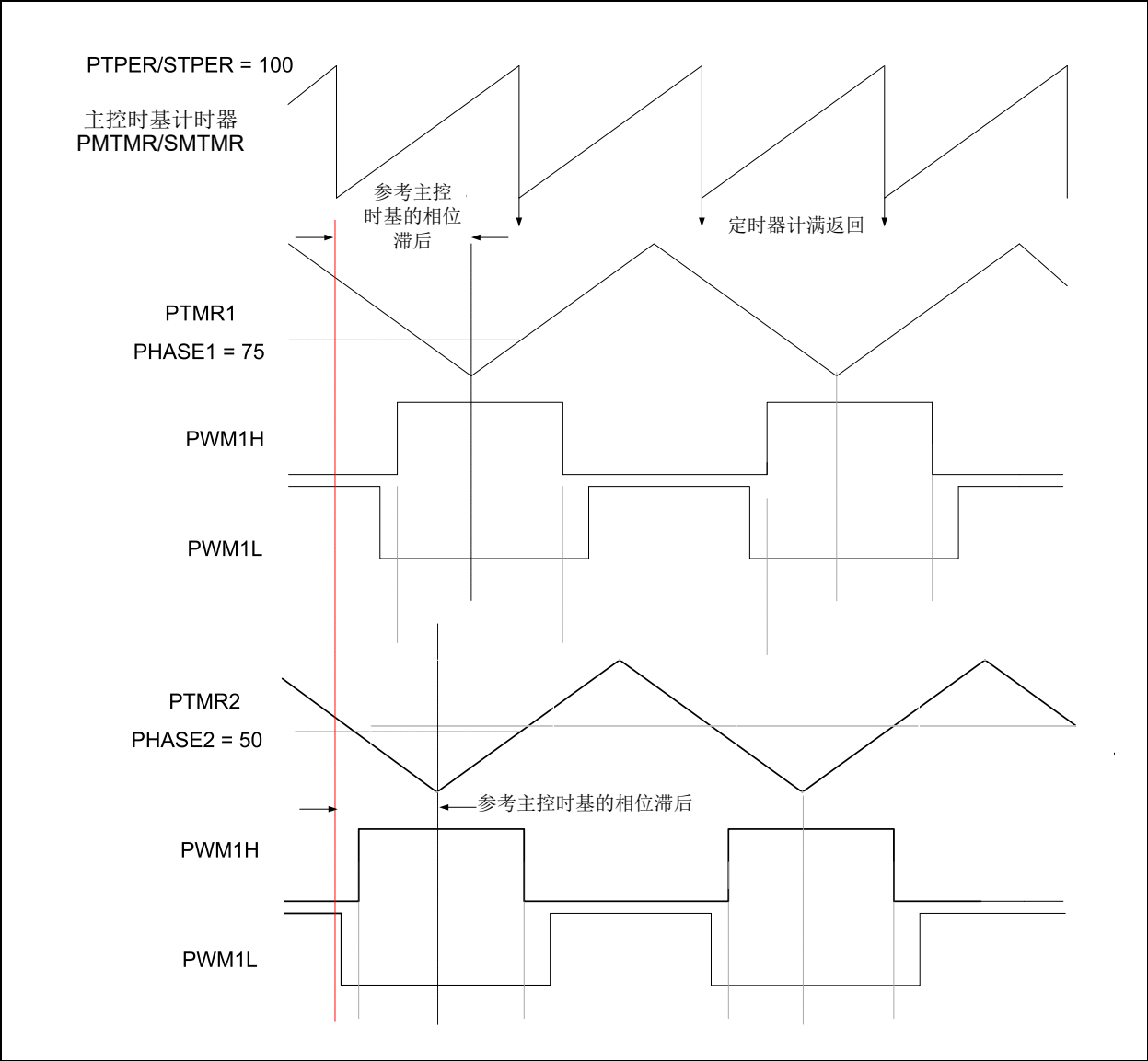
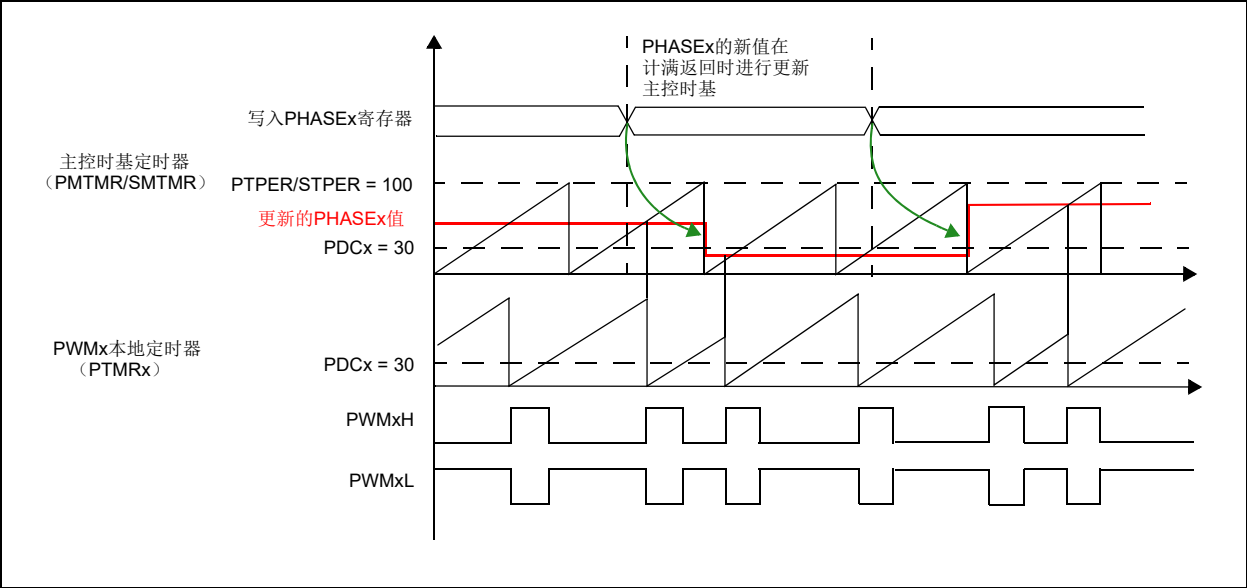


图 44-21 说明了在同步模式中对通道 / 发生器定时器的 PHASEx 寄存器进行运行时更新的影响。

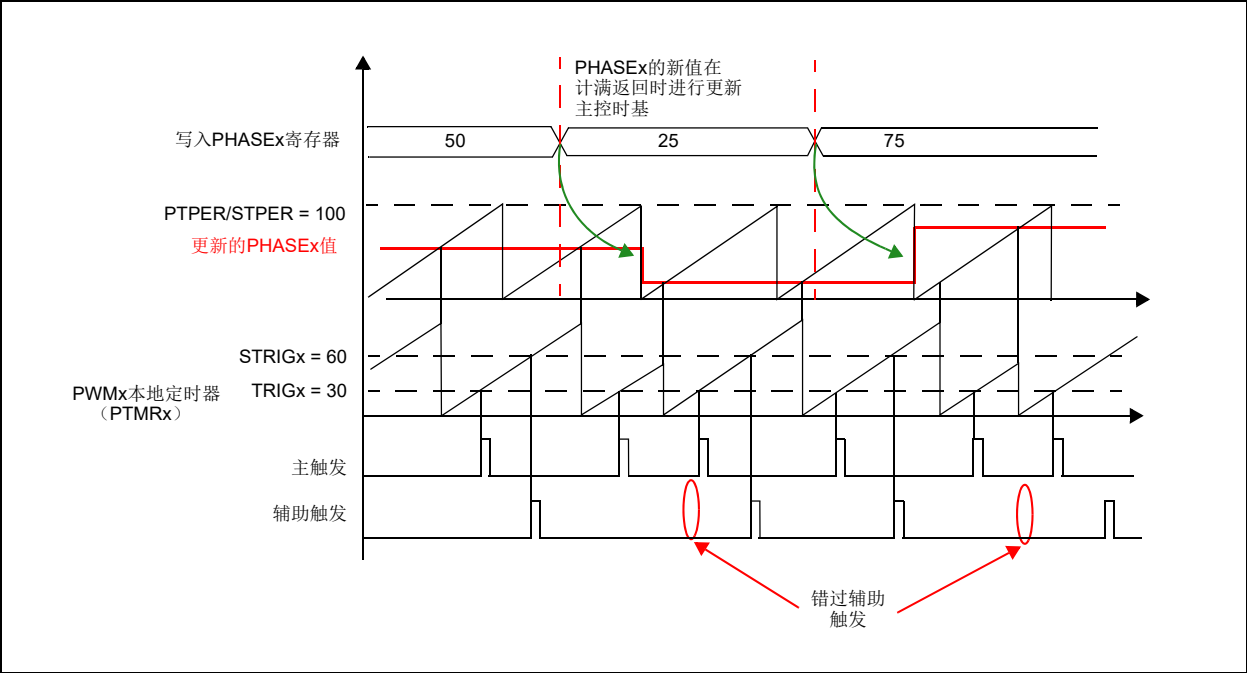
图 44-21： 相移对通道 / 发生器定时器和输出波形的影响



对 PWM 信号进行相移时，PWM 定时器值会被更新，以反映新的相位值。更改相位时可能会错过触发事件。用户应用程序必须确保这不会对任何控制环的执行产生影响。

图 44-22 给出了相移对于 PWM 触发的影响的波形图。

图 44-22: 相移对通道 / 发生器触发的影响



## 44.9 写保护

一些器件为 PWM I/O 控制寄存器 (IOCONx) 实现了写保护功能, 可以防止对这些寄存器的任何意外写操作。该功能可以通过 PWMLOCK 配置位 (DEVCFG3<20>) 进行控制。写保护功能的默认状态是禁止 (PWMLOCK = 1)。关于闪存配置字节的更多信息, 请参见具体器件数据手册中的“特殊功能”章节。

要对这些锁定的寄存器进行写访问, 用户应用程序必须向 PWM 解锁寄存器 (PWMKEY) 中连续写入两个值 0xABCD 和 0x4321。对 IOCONx 寄存器的写访问必须是紧接在解锁序列之后的下一个特殊功能寄存器 (Special Function Register, SFR) 访问; 在解锁过程和后续写访问期间不能有任何其他 SFR 访问。

正确的解锁序列如例 44-15 所示。

**例 44-15: PWM 写保护寄存器的解锁序列**

```
; FLT15 pin must be pulled high externally to clear and disable the fault
; Writing to IOCONx register requires unlock sequence
mov #0xabcd,r1          ;Load first unlock key to r1 register
mov #0x4321,r2          ;Load second unlock key to r2 register
mov r1,PWMKEY           ;Write first unlock key to PWMKEY register
mov r2,PWMKEY           ;Write second unlock key to PWMKEY register
mov r3,IOCON4           ;Write desired value to IOCON SFR for channel 4
```

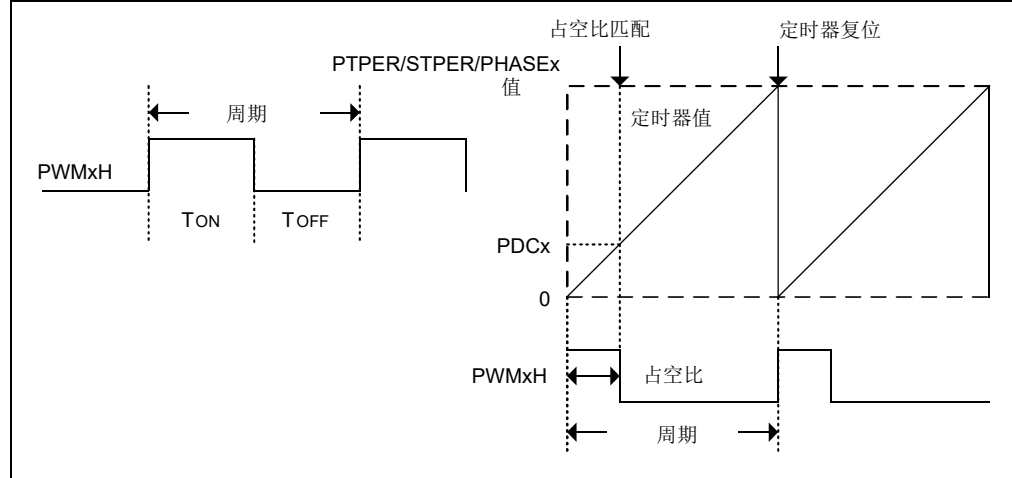
## 44.10 PWM输出模式

### 44.10.1 标准边沿对齐的 PWM

图 44-23 给出了标准边沿对齐的 PWM 波形。要产生边沿对齐的 PWM，定时器或计数器电路需要从 0 开始递增计数至某个指定的最大值（称为周期）。另一个寄存器中包含占空比值，该值不断与定时器值进行比较。当定时器或计数器值小于等于占空比值时，PWM 输出信号置为有效。当定时器值超出占空比值时，PWM 信号置为无效。当定时器值大于等于周期值时，定时器会复位自身，并且该过程会一直重复。

多个 PWM 发生器可以使用其中一个主控时基发生器进行同步。在此模式中不同发生器的波形可以使用 PHASEx 寄存器相对于彼此进行移位。

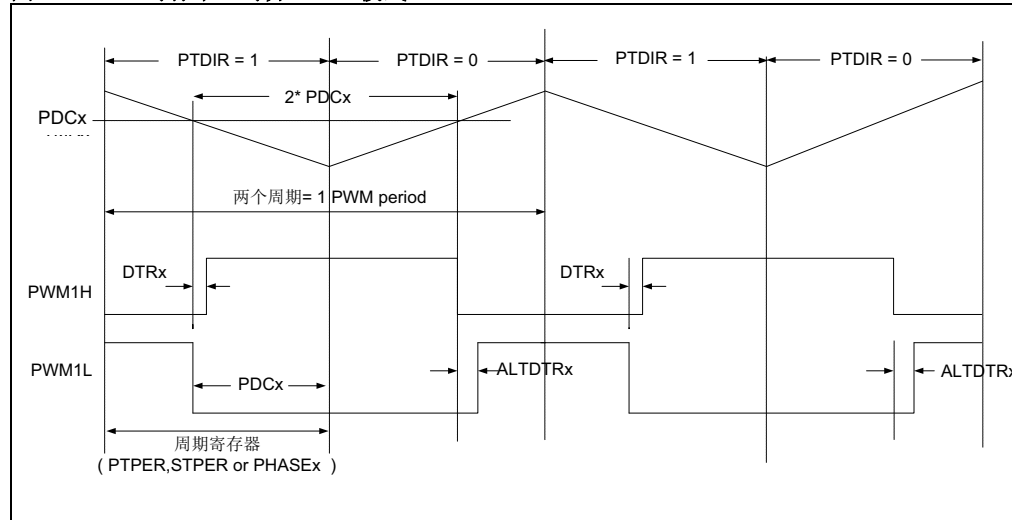
图 44-23: 标准边沿对齐的 PWM 模式



### 44.10.2 对称中心对齐 PWM

中心对齐的 PWM 波形（如图 44-24 所示）根据一个参考点来对齐 PWM 信号，使 PWM 信号的一半出现在参考点之前，信号的另一半出现在参考点之后。当 PWM 控制寄存器中的 ECAM 位（PWMCONx<11:10>）为 01、10 或 11 时，使能中心对齐模式。

图 44-24: 对称中心对齐 PWM 模式



在中心对齐模式下工作时，实际 PWM 周期是周期寄存器（PTPER/STPER/PHASEx）指定值的两倍，因为在相应的周期中，PWM 发生器中的本地定时器（PTMRx）计数器先递增然后再递减。递增 / 递减计数序列会使实际 PWM 周期加倍。许多电机控制应用中都使用了该模式。



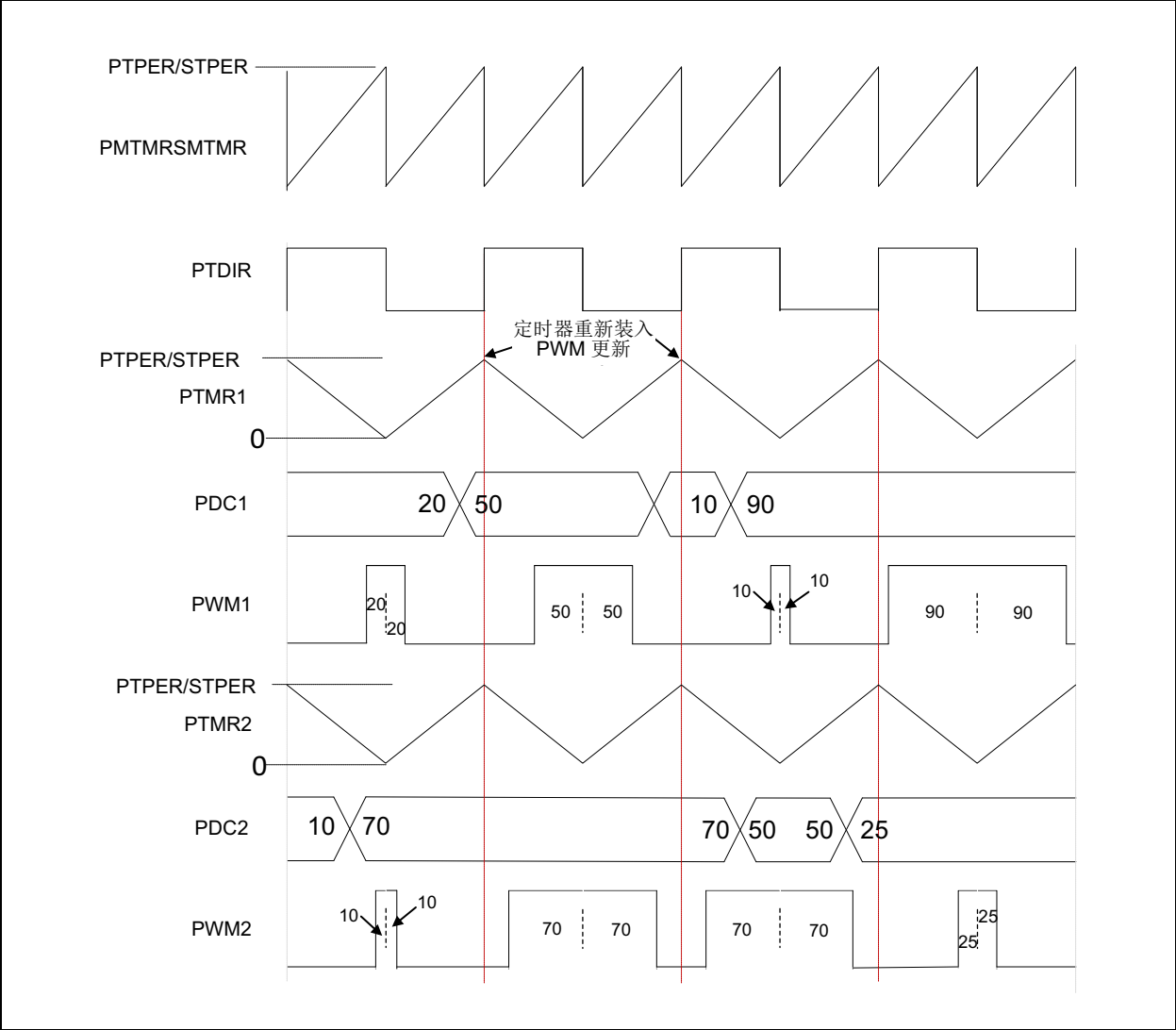
中心对齐模式中 PWM 的分辨率是边沿对齐模式中的一半。当需要具有全分辨率的中心对齐模式时，建议使用非对称中心对齐模式（ECAM<1:0> 位（PWMCONx<11:10>）= 10）。占空比寄存器 PDCx 和 SDCx 按以下方式写入，以获得全分辨率：

- PDCx = 占空比 / 2
- SDCx = 占空比 - PDCx

在周期匹配（PTMRx = PTPER/STPER/PHASEx）时也可能发生中断。在对称中心对齐 PWM 模式下只会使用 PDCx 寄存器。SDCx 寄存器会自动将写操作反映到 PDCx。

图 44-25 说明了对 PDCx 寄存器的软件更新（在此模式中 SDCx 的更新对软件保持透明）与波形或引脚更新的实际发生时间之间的时序关系。

图 44-25： 对称模式占空比更新时序图



44.10.3 非对称中心对齐 PWM 模式

此模式通过将 ECAM<1:0> 位（PWMCONx<11:10>）设置为 10 进行选择。此模式中有两个占空比寄存器，即 PDCx 和 SDCx，它们提供 PWM 前沿和下降沿的跳变实例。实际上，通过这两个寄存器，用户可产生同步多相波形，其中的占空比相对于内部同步参考来说是非对称的。此模式主要在利用非对称 PWM 方案且要求较高更新带宽的情况中使用。

图 44-26 说明了有关不同于中心对齐模式的中心参考的非对称前沿和下降沿跳变。

图 44-26： 非对称中心对齐 PWM 模式

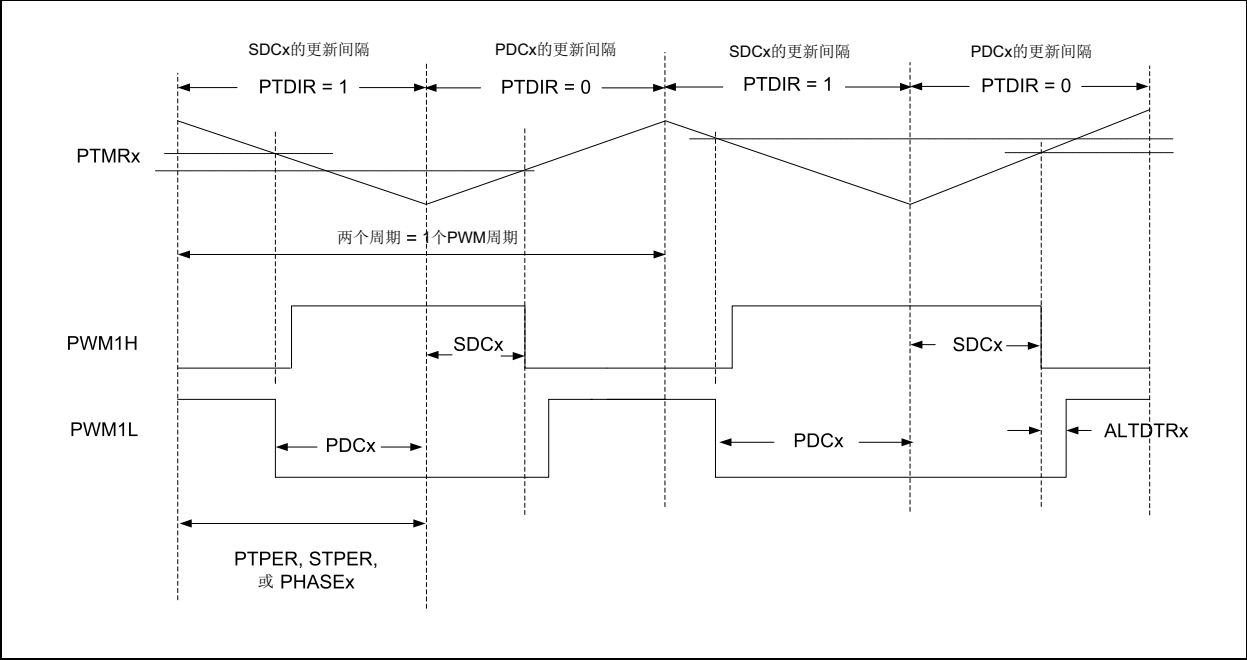
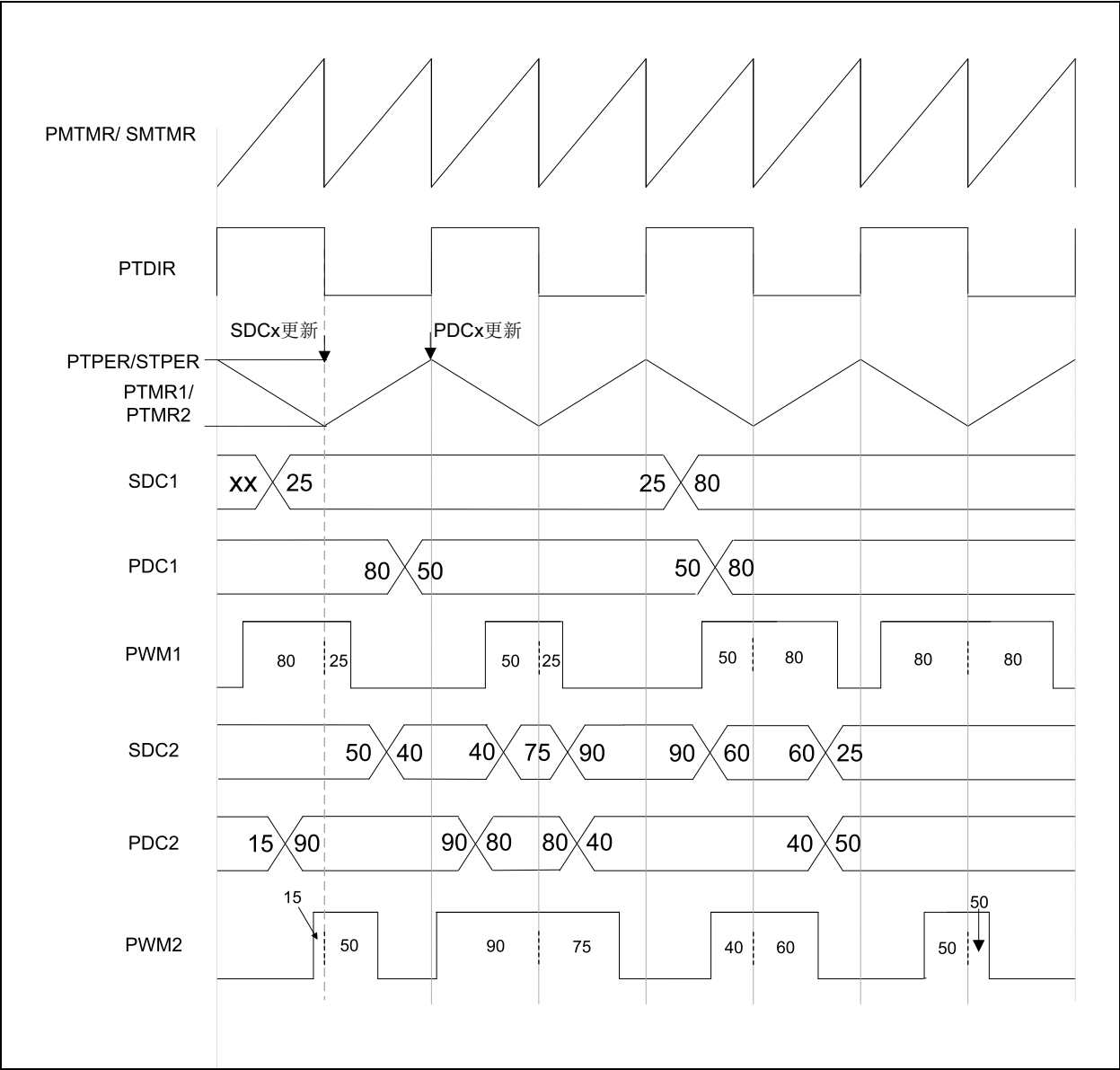


图 44-27 说明了对 PDCx 和 SDCx 寄存器的软件更新与波形 / 引脚更新的实际发生时间之间的时序关系。

图 44-27： 非对称模式双重更新速率时序图



例 44-16 说明了如何在不同对齐模式中配置通道 / 发生器。

例 44-16： 边沿对齐或中心对齐模式选择

```
/* Select Edge Aligned PWM mode */
PWMCON1bits.ECAM = 0x00;

/* Select Center Aligned PWM mode */
PWMCON1bits.ECAM = 0x01;

/* Select Asymmetric Center Aligned PWM mode with double update rate */
PWMCON1bits.ECAM = 0x02;

/* Select Asymmetric Center Aligned PWM mode with simultaneous update */
PWMCON1bits.ECAM = 0x03;
```

## 44.11 PWM发生器触发

PWM 模块具有来自各个 PWM 发生器的 ADC 触发输出。可使用触发脉冲来启动 ADC 的采样和转换操作或产生中断。

为便于在产生 PWMxH/PWMxL 波形期间在指定点及时触发，提供了 TRIGx 和 STRIGx 两个比较寄存器。用户在每个寄存器中指定一个匹配值，在值与本地时基定时器 PTMRx 相匹配时，在周期内波形中的不同点上产生两个不同的触发。

有些 PWM 模式，比如中心对齐模式和推挽模式，属于双周期 PWM 周期。默认的触发产生行为是在双周期 PWM 周期中创建两个原始触发比较事件。通过 TRGSEL<1:0> 位 (TRGCONx<11:0>)，用户可以同时选择两个触发，或仅从双周期 PWM 周期的前半部分或后半部分选择原始比较事件。

触发脉冲共用一个 PWM 周期分频器，即 TRGDIV<3:0> 位 (TRGCONx<15:12>)。这使 ADC 的触发信号可以每隔 1、2、3-16 个触发事件产生一次。

用户可以通过触发分频器来调整 ADC 采样速率，使之满足控制环的要求。

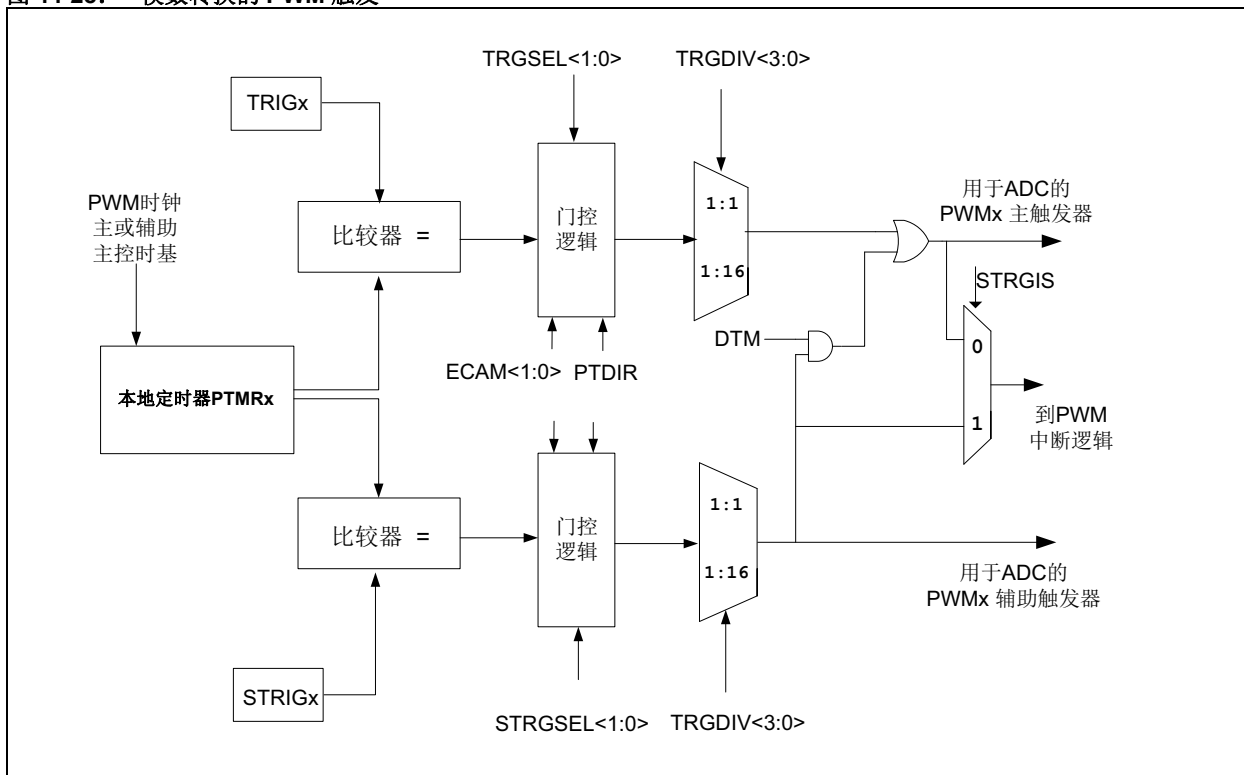
如果 PWMCONx 寄存器中的 TRGIEN 位 (PWMCONx<21>) 设置为 1，则会产生中断请求。

**注 1:** 用作 ADC 触发的触发脉冲的产生与 TRGIEN 位的状态无关。

**注 2:** 对于双周期 PWM 模式，应注意避免可能在周期值的较高端或较低端发生的连续中断。

图 44-28 说明了结合关联的中断功能，来自 MCPWM 模块的 ADC 模块的三种不同触发可能。

图 44-28: 模数转换的 PWM 触发



**注:** 当 TRGDIV<3:0> 位设置为 0 时，触发只能在第一个 PWM 间隔产生。

根据 TRGDIV<3:0> 位的设置，触发将按不同 PWM 间隔产生，如图 44-29 至图 44-32 所示。

图 44-29： 边沿对齐模式下 PWM 触发信号与 PWM 输出的关系（TRGDIV = 0）

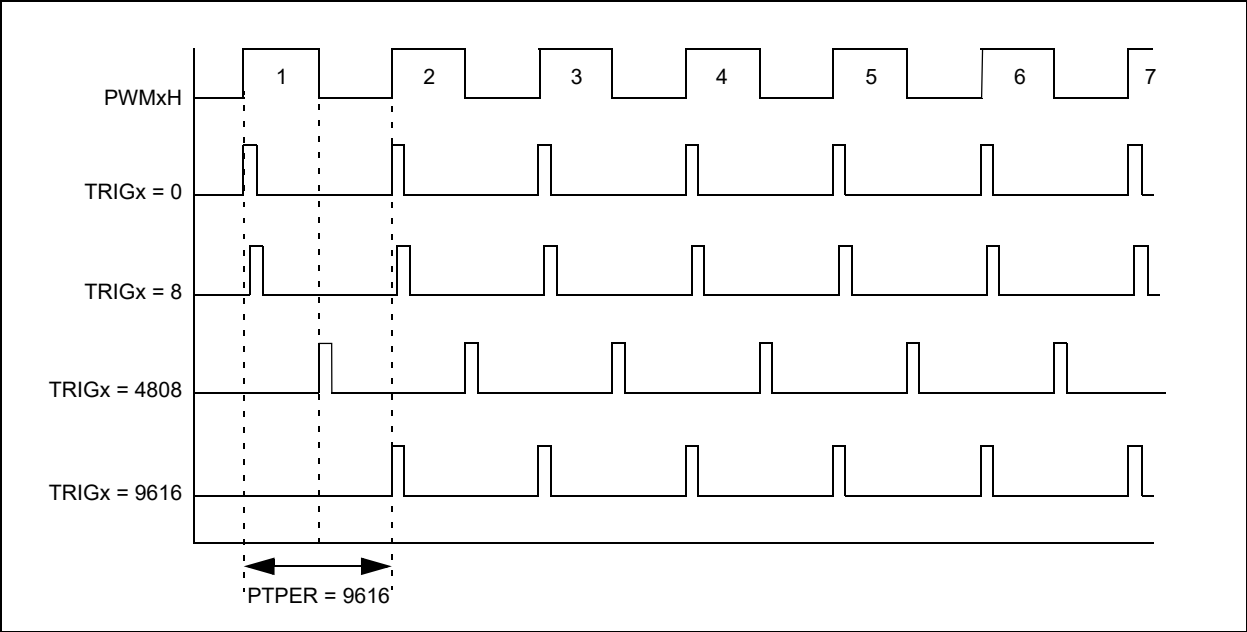


图 44-30： 边沿对齐模式下 PWM 触发信号与 PWM 输出的关系（TRGDIV = 1）

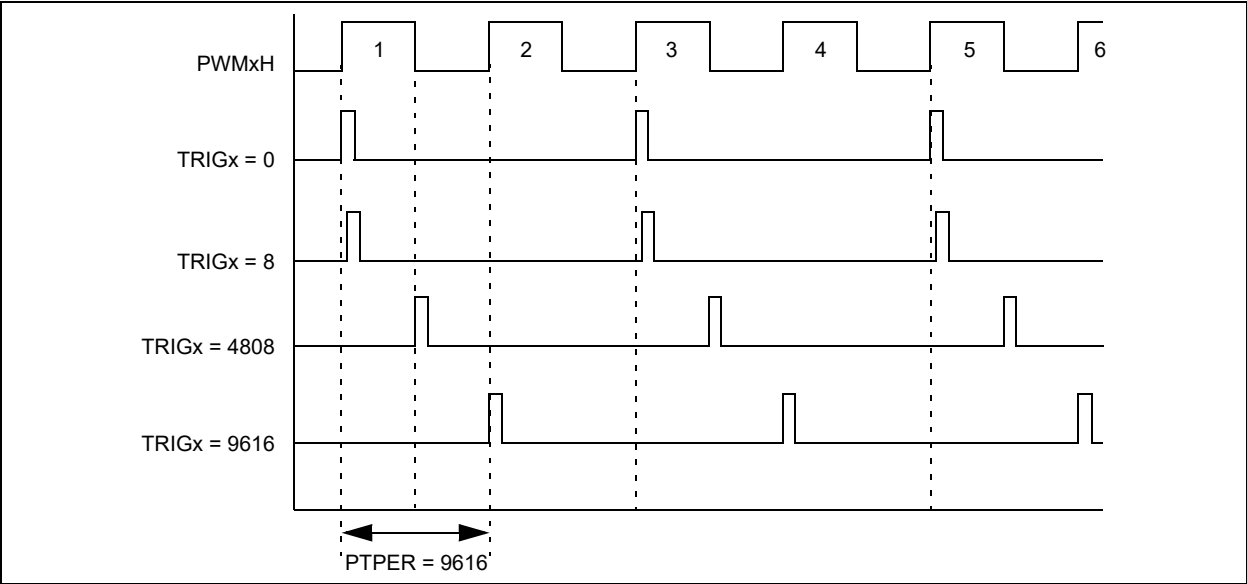


图 44-31： 边沿对齐模式下 PWM 触发信号与 PWM 输出的关系（TRGDIV = 2）

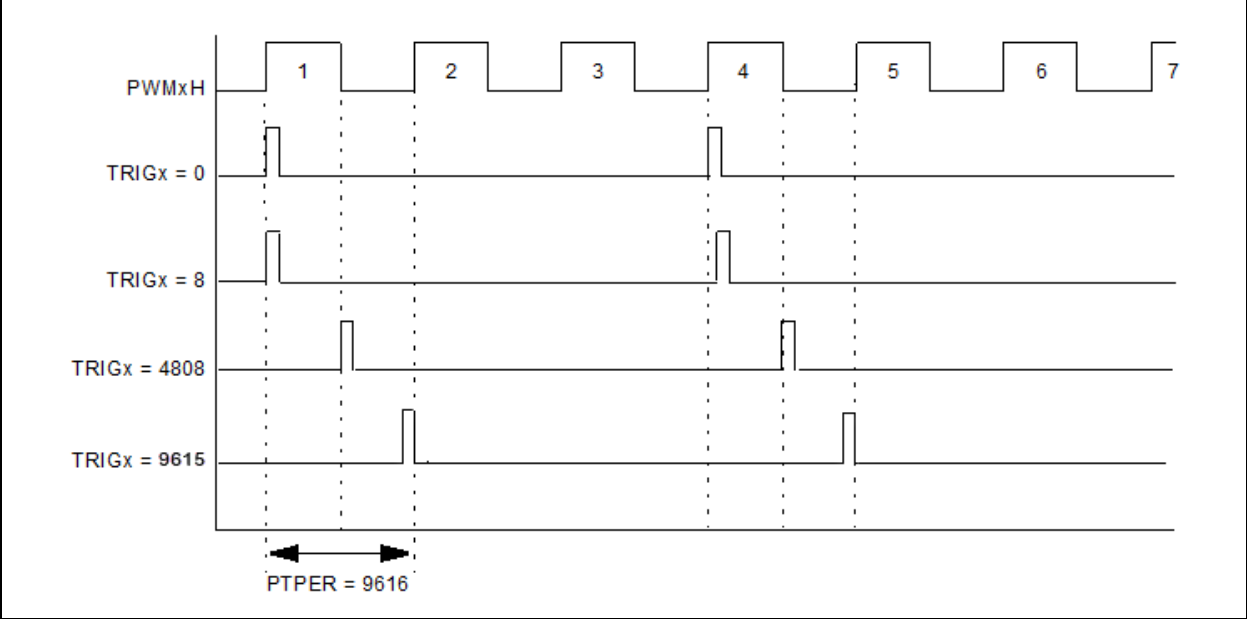


图 44-32： 中心对齐模式下触发信号与 PWM 输出的关系（TRGDIV = 0， TRGSEL = 0 或 3）

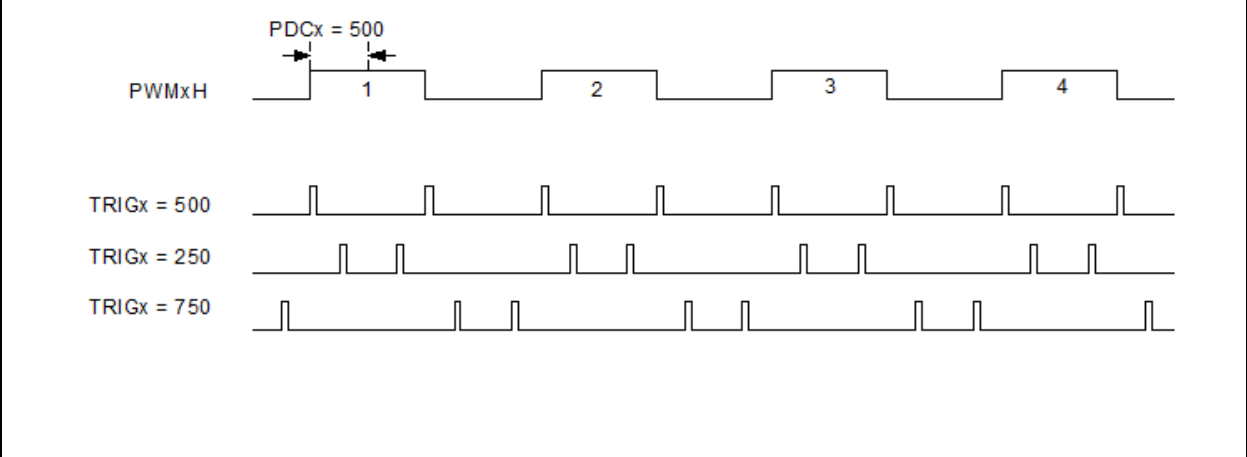


图 44-33 说明了触发信号波形和分频与中心对齐模式（TRGDIV = 1）中的 PWM 输出的关系。

图 44-33： 中心对齐模式下触发信号与 PWM 输出的关系（TRGDIV = 1， TRGSEL = 0 或 3）

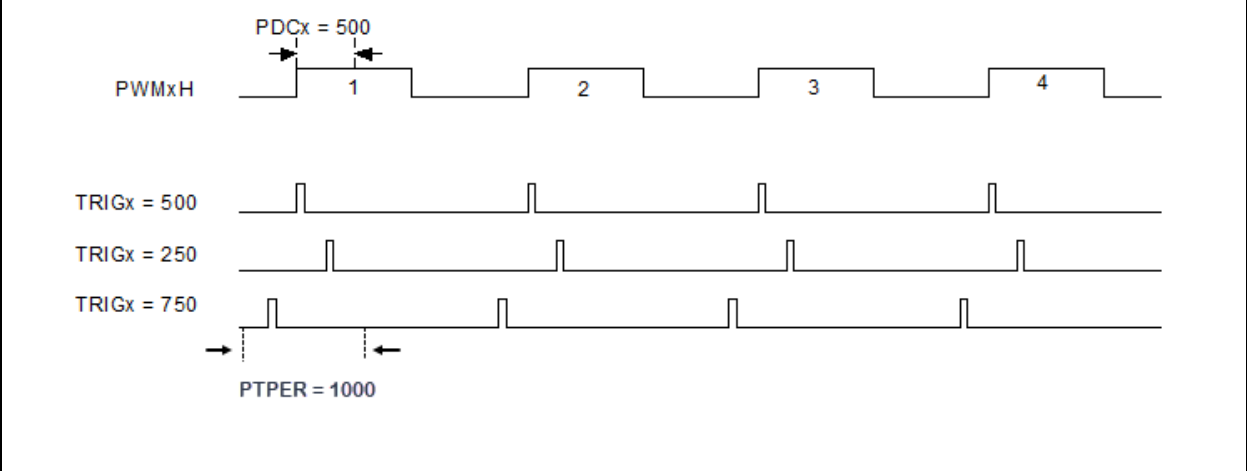


图 44-34: 中心对齐模式下触发信号与 PWM 输出的关系 (TRGDIV = 0, TRGSEL = 1)

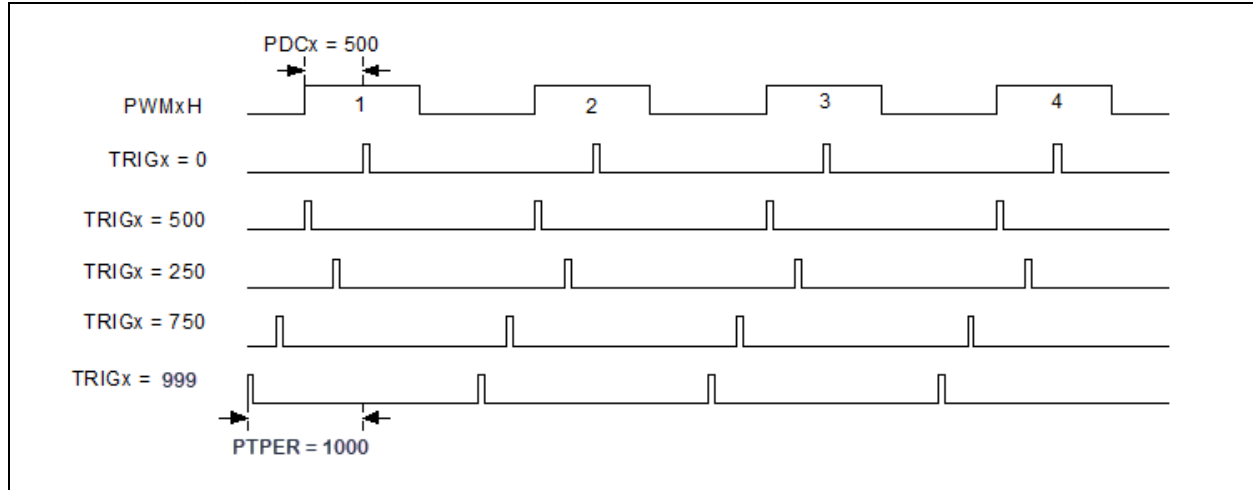


图 44-35: 中心对齐模式下触发信号与 PWM 输出的关系 (TRGDIV = 0, TRGSEL = 2)

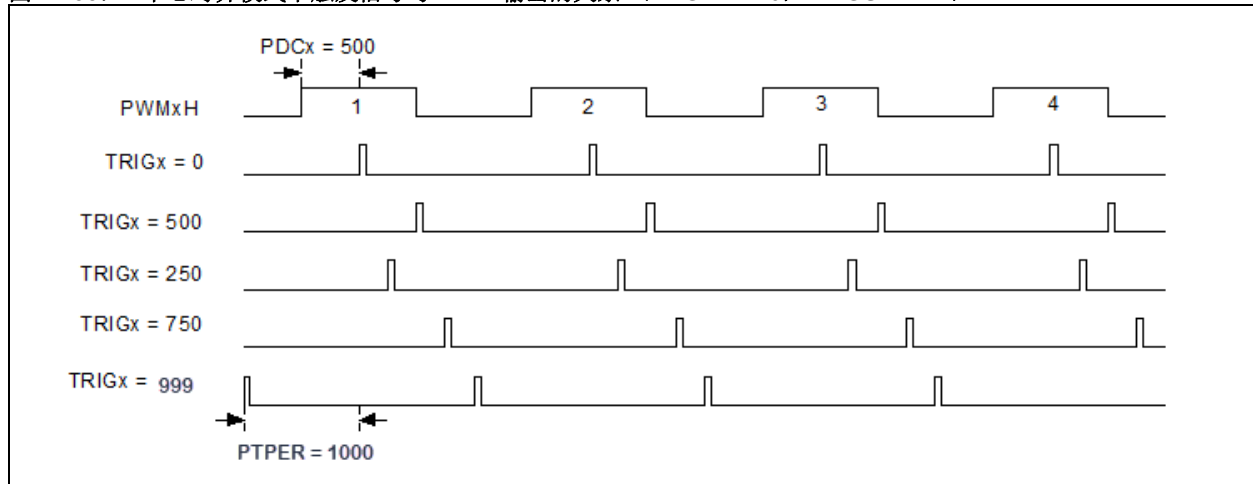


图 44-36: 中心对齐模式下触发信号与 PWM 输出的关系 (TRGDIV = 1, TRGSEL = 2)

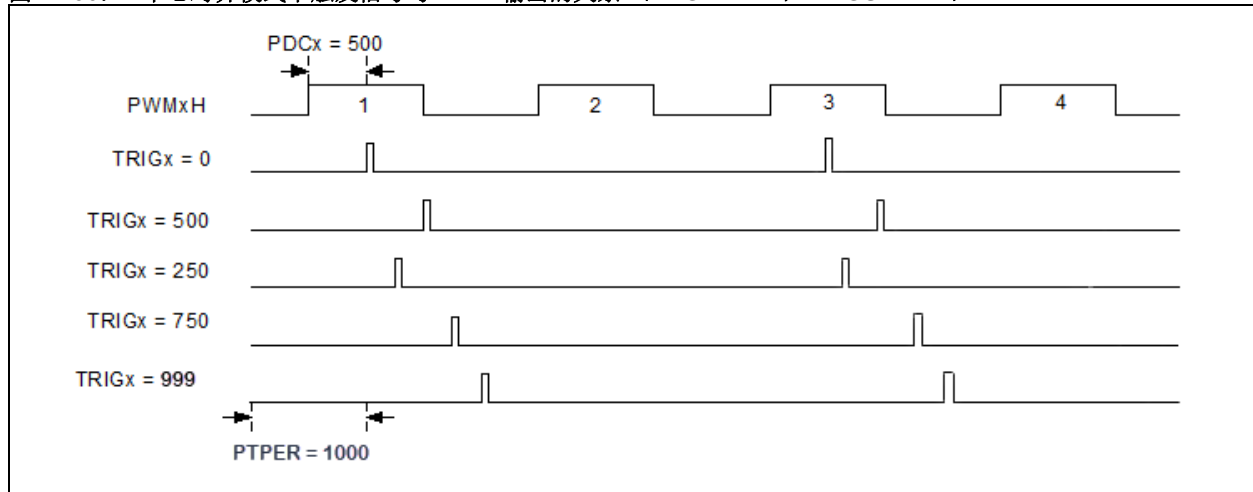
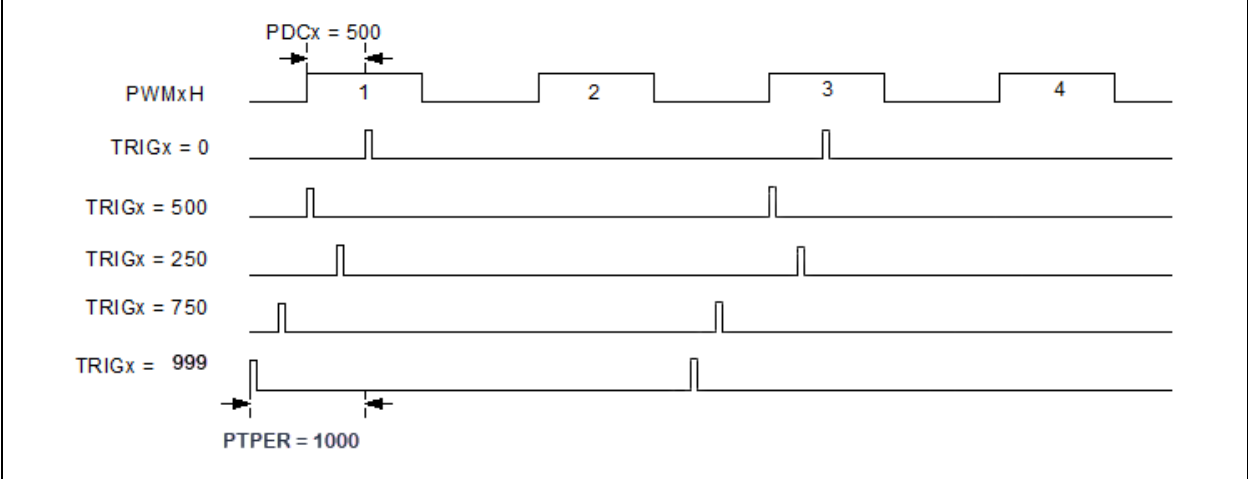


图 44-37： 中心对齐模式下触发信号与 PWM 输出的关系（TRGDIV = 1， TRGSEL = 1）



用户应用程序可以通过触发分频器来调整 ADC 采样速率，使之满足控制环的要求。

如果 ADC 触发信号的产生速率高于 ADC 的处理速率，则操作可能导致一些采样丢失。但是，用户应用程序可以确保它提供的时间足以在单个 PWM 周期内完成一次 ADC 操作。

触发脉冲的产生与触发中断允许位 TRGIEN（PWMCONx<21>）的状态无关。如果 TRGIEN 位设置为 1，则会产生中断请求（IRQ）。第 44.5.2.1 章“特殊事件触发器”中的例 44-1 提供了独立 PWM ADC 触发的代码。

- 注 1：** 触发中断状态位 TRGSTAT（PWMCONx<13>）只能通过清零 TRGIEN 位进行清零。它不会自动清零。

**2：** 在使用多个 PWM 通道的应用（如 IPFC 和多相降压稳压器）中，动态触发具有一些优点。可基于 PWM 周期、占空比、负载电流等更改 TRIGx 的值。这是为了确保触发点与 PWM 通道的上升和下降时刻相独立。



表 44-9 汇总了用作 ADC 触发的周期和比较值的寄存器以及源自 MCPWM 模块的关联中断。

表 44-9: 使用关联中断的 ADC 触发和同步源

ADC 触发事件 <sup>(1)</sup>	触发中断 <sup>(3)</sup>			触发源		
DTM = 无关	STRGIS = 0 DTM = 0	STRGIS = 0 DTM = 1	STRGIS = 1 DTM = 无关	ITB = 0 MTBS = 0	ITB = 0 MTBS = 1	ITB = 1 MTBS = 无关
SEVT	SEVT	SEVT	SEVT	PTPER	PTPER	PTPER
SSEVT	SSEVT	SSEVT	SSEVT	STPER	STPER	STPER
TRIGx、 STRIGx	TRIGx	TRIGx、 STRIGx	STRIGx	PTPER	STPER	PHASEx
PWM 发生器 x 限流	限流中断			基于异步外部数字输入事件或 FLTy <sup>(2)</sup> 引脚		

注 1: 从 MCPWM 模块选择的 ADC 触发事件在 ADC 模块（STRGSRC<4:0> 位（ADCCON1<20:16>））中进行指定，并且特定于器件。关于 ADC 触发关联，请参见具体器件数据手册中的“ADC”章节。

2: x 和 y = 1，最大为 6。如需确定可用性和引脚分配，请参见具体器件的数据手册。

3: TRIGx 和 STRIGx 中断源中仅有单个中断向量。需要使用软件来对中断的源进行仲裁。

例 44-17 说明了在发生器 1 的 PWM 模块的每个周期中用于向 ADC 模块产生两个触发信号的代码。主触发信号（TRIGx）在前半部分触发，辅助触发信号（STRIGx）则在后半部分触发。在允许两个触发信号和相同中断向量中断的情况下，使能双触发模式。由于事件在时间上交错，因而单个中断将减少一个额外的中断源，而无需担心发生嵌套。

例 44-17: 双交错触发和中断产生

```

TRGCON1bits.DTM = 1;           // Logically OR the interrupt sources to the
                                // same vector
TRGCON1bits.STRGIS = 0;        // Enable both interrupts
TRGCON1bits.TRGDIV = 0;        // No postscaler divider; trigger and interrupt
                                // every cycle
TRGCON1bits.TRGSEL = 1;        // Trigger only during the first half of the cycle
TRGCON1bits.STRGDIV = 0;       // No postscaler divider; trigger and interrupt
                                // every cycle
TRGCON1bits.STRGSEL = 2;       // Trigger only during the second half of the cycle
    
```

## 44.12 PWM 中断

MCPWM 模块可以基于内部时序信号或通过限流和故障输入基于外部信号产生中断。每个 PWM 发生器模块都可以向中断控制器提供它自己的 IRQ 信号。每个 PWM 发生器的中断是该模块的触发事件 IRQ、周期匹配、周期复位、限流输入事件或故障输入事件的布尔或。

除了每个 PWM 发生器的一个 IRQ 信号之外，中断控制器还可以基于两个特殊事件的每一个特殊事件，从主控时基接收 IRQ 信号。

来自每个 PWM 发生器的 IRQ 称为独立 PWM 中断。每个独立中断的 IRQ 可以来自 PWM 独立触发器、PWM 周期匹配 / 复位、PWM 故障逻辑或 PWM 限流逻辑。每个 PWM 发生器在 IFSx 寄存器中都具有 PWM 中断标志。当以上任意中断源产生 IRQ 时，与选定 PWM 发生器关联的 PWM 中断标志会置 1。

如果使能了多个 IRQ 源，则使用用户应用程序通过检查 TRGIF 位（PWMCONx<29>）、PWML 中断状态位（PWMCONx<28>）、PWMH 中断状态位（PWMCONx<27>）、故障中断状态位 FLTSTAT（PWMCONx<15>）和限流中断状态位 CLSTAT（PWMCONx<14>）来确定中断源。

为配置中断，应用软件应先清零中断标志，允许中断（请参见[寄存器 44-1](#)、[寄存器 44-5](#)和[寄存器 44-11](#)），并在最后使能 MCPWM 模块。

### 44.12.1 PWM 时基中断

在每个 PWM 发生器中，MCPWM 模块可以基于主控时基和 / 或独立时基产生中断。SEVTCMP/ SSEVTCMP 寄存器用于为主时基指定基于定时器的中断，TRIGx/STRIGx 寄存器用于为独立时基指定基于定时器的中断。

通过 SEIEN 位（PTCON<11>）允许主时基特殊事件中断，并通过 SSEIEN 位（STCON<11>）允许辅助时基特殊事件中断。在每个 PWM 发生器中，由触发逻辑产生的独立时基中断通过 TRGIEN 位（PWMCONx<21>）进行控制。

**注：** 当发生相应的匹配条件时，无论相应中断允许位的设置如何，总是会产生送到 ADC 的特殊事件触发信号和独立 PWM 触发脉冲。

### 44.13 PWM 故障

PWM 故障输入引脚的关键功能如下：

- 每个 PWM 发生器可以从最多 16 个故障和限流引脚中选择其自己的故障输入源
- 每个 PWM 发生器都具有故障控制信号源选择位 FLT SRC<3:0> (IOCONx<22:19>)。这些位指定其故障输入信号的来源。
- 每个 PWM 发生器都具有故障中断允许位 FLT IEN (PWMCONx<23>)。该位用于使能故障 IRQ 的产生。
- 每个 PWM 发生器都具有 PWM 发生器 x 的故障极性位，即 FLTPOL 位 (IOCONx<18>)。该位用于选择选定故障输入的有效状态。
- 在发生故障条件时，可以将 PWMxH 和 PWMxL 输出强制为以下状态：
  - 在故障模式下，FLT DAT<1:0> 位 (IOCONx<5:4>) 将提供要赋给 PWMxH 和 PWMxL 输出的数据值

下面介绍了故障输入引脚的主要功能：

- 故障条件可以改写 PWM 输出。FLT DAT<1:0> 位的值可以为 0 或 1。如果 FLT DAT<1:0> 设置为 00，则它将进行异步处理，以便立即关断应用电路中的相关功率晶体管。如果 FLT DAT<1:0> 设置为 11，则死区逻辑会对它进行处理，然后再应用于 PWM 输出。
- 故障输入信号可以用作 ADC 的触发信号，触发信号可以启动 ADC 转换过程。无论 MCPWM 模块、FLTMOD<1:0> 位或 FLT IEN 位的状态如何，ADC 触发信号总是有效。关于 ADC 触发的更多信息，请参见具体器件数据手册中的“12 位高速逐次逼近寄存器 (SAR) 模数转换器 (ADC)”章节。
- 故障信号可以产生中断。FLT IEN 位用于控制故障中断信号的产生。即使 PWM 发生器 x 的故障模式位 FLTMOD<1:0> (IOCONx<17:16>) 禁止了故障改写功能，用户应用程序也可以指定产生中断信号。这使故障输入信号可以用作通用外部 IRQ 信号。

FLT x 引脚通常高电平有效。FLTPOL 位设置为 1 时，选定故障输入信号会反相；因此，这些引脚被设置为低电平有效。

在使能 MCPWM 模块时，故障引脚还可以通过端口 I/O 逻辑读取。这使用户应用程序可以在软件中查询故障引脚的状态。

#### 44.13.1 B 类故障

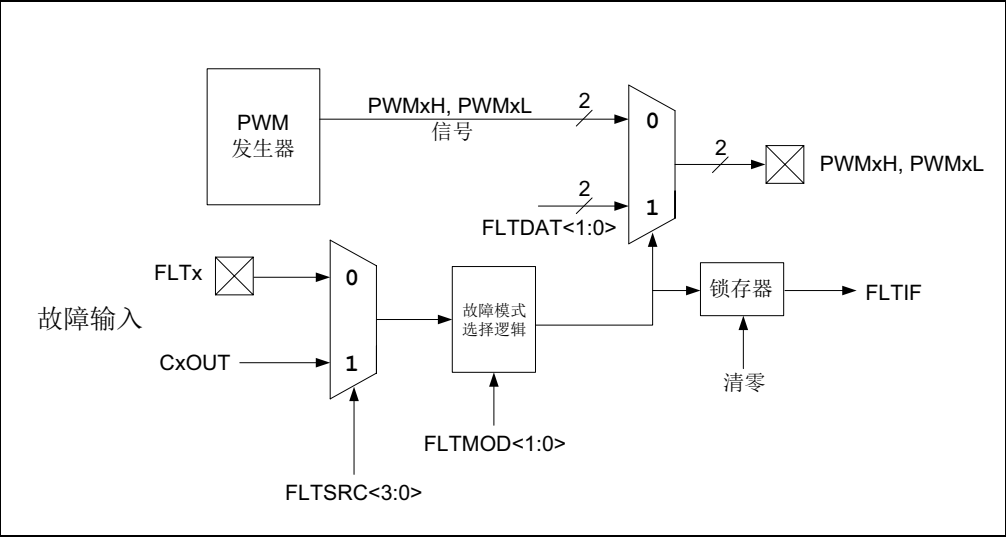
一些器件具有一种实现了 B 类安全特性的故障，称为 B 类故障。这种故障的工作方式与其他故障类似，只是在发生任何类型的复位时，MCPWM 模块会保持该引脚的所有权。关于具有该特性的故障的更多信息，请参见具体器件数据手册中的“电机控制 PWM (MCPWM)”章节。

发生复位时，该故障均以锁定模式使能，以保证应用安全上电。应用软件必须先清除该故障，然后才能使能 PWM 模块。要清除故障条件，必须先将该故障引脚从外部拉为低电平，或者在 CNPDx 寄存器中使能内部下拉电阻。清除故障条件之后，可以使能 PWM 模块，如果需要的话，可以禁止该故障。

**注：** 如果器件上提供，则 B 类故障会在发生任何类型的复位时使能。用户应用程序必须先清除该故障，然后才能使能 MCPWM 模块。

图 44-38 给出了 PWM 故障改写机制的框图。

图 44-38: PWM 故障控制模块框图



44.13.2 模拟比较器产生的 PWM 故障信号

PIC32 器件支持与比较器模块 CxOUT 的输出的虚拟（内部）连接（请参见第 39 章“运放 / 比较器寄存器”（DS60001178）中的图 26-1）。

虚拟连接提供了一种简单方式来进行外设间的连接，而无需使用物理引脚。例如，通过将 FLT SRC<3:0> 位（IOCONx<22:19>）设置为 0b1000，比较器 1 输出可用作故障源，这允许模拟比较器触发 PWM 故障，而无需使用器件上的实际物理引脚。

例 44-18 给出了这样一种配置：模拟比较器 1 作为 PWM 的一个故障源，连接到故障输入引脚 1。

例 44-18: 将模拟比较器 1 配置为高速 PWM 模块的故障源

```
// Unlock IOCON Register
PWMKEY=0xABCD;
PWMKEY=0x4321;
//Set C1OUT as fault source for PWM1
IOCON1bits.FLT SRC=0b1000;
//Lock IOCON Register
PWMKEY=0x0000;
```

## 44.13.3 故障中断

FLTIEN 位 (PWMCONx<23>) 用于决定在 FLT<sub>x</sub> 输入置为高电平时是否产生中断。在发生故障时, FLTDAT<1:0> 位 (IOCONx<5:4>) 将提供要赋给 PWM<sub>x</sub>H 和 PWM<sub>x</sub>L 引脚的数据值。

PWM 故障状态在 FLTSTAT 位 (PWMCONx<15>) 中提供。FLTSTAT 位指示故障 IRQ 锁定状态。如果不允许故障中断, FLTSTAT 位将以正逻辑形式指示选定 FLT<sub>x</sub> 输入的状态。当故障输入引脚不与 PWM 发生器关联使用时, 这些引脚可以用作通用 I/O 或中断输入引脚。

## 44.13.3.1 故障输入模式

选定故障和限流输入信号将被锁存, 以捕捉限流和故障输入信号。捕捉功能在识别限流或故障信号是否置为有效时是异步功能。这可以确保在信号置为有效的时间较短或出现时钟系统故障, 或者检测到故障或限流事件时, 可以关断 (即, 置为无效) PWM 输出。

对于间歇性的故障或限流, 如果限流和故障信号已置为无效, 则锁存器将在 PWM 周期结束时自动清零。

选定故障输入引脚具有两种可使用 FLTMOD<1:0> 位 (IOCONx<17:16>) 选择的工作模式:

- **锁定模式:** 在锁定模式下, 当 FLT<sub>x</sub> 引脚置为有效时, PWM 输出会遵从 FLTDAT<1:0> 位中定义的状态。PWM 输出将保持在此状态, 直到故障引脚置为无效, 并且相应的中断标志 (FLTIF) 已用软件清零。在这两个操作都发生后, PWM 输出将在下一个 PWM 周期边界开始时返回到正常工作状态。如果 FLTIF 位在故障条件结束之前清零, 输出将保持改写状态, 直到故障引脚不再置为有效为止。由于 FLTSTAT 位反映 FLT<sub>x</sub> 输入引脚置为有效的状态, 因而只要存在故障, 将持续地保持改写状态。
- 因此在锁定模式下时, 用户软件必须定期查询 FLTSTAT 位, 以便确定故障的“恢复”条件, 然后清零 FLTIF/CLIF 标志, 以使能正常功能。
- **逐周期模式:** 在逐周期模式下, 事件 (限流和 / 或故障) 将保持至当前 PWM 周期结束为止。在下个 PWM 周期开始时, 将基于所选 FLT<sub>x</sub> 引脚上外部硬件置为有效的逻辑电平, 对事件的状态进行重新求值。如果限流或故障输入仍置为有效, 则在发生故障条件之前, PWM 输出将保持在故障或限流改写状态 (如 FLTDAT 和 CLDAT 位所设定)。此逐周期重新求值的目的是: 在故障置为无效时为切换出改写状态提供最快速和最可靠的机制, 同时防止用户软件查询故障, 这可能会导致 PWM 输出在故障改写状态和工作状态之间快速交替。此 PWM 输出的快速交替可能会损坏应用的输出晶体管。

故障模式 FLTMOD<1:0> 位可以防止设置为 0 的 PWMLOCK 位和所需写序列的意外写入。

**注 1:** 更改将在使能 PWM 之后的下个 PWM 周期边界以及后续每个 PWM 周期边界上生效。将 FLTMOD<1:0> 位从 00 或 01 更新到 11 (禁止) 时, 如果故障输入仍处于有效状态, 将不会删除故障改写条件。

**2:** 在使能 PWM 模块时, 故障引脚还可以通过 GPIO I/O 逻辑读取。这使用户可以在软件中查询故障引脚的状态。

图 44-39： 故障时序周期恢复模式

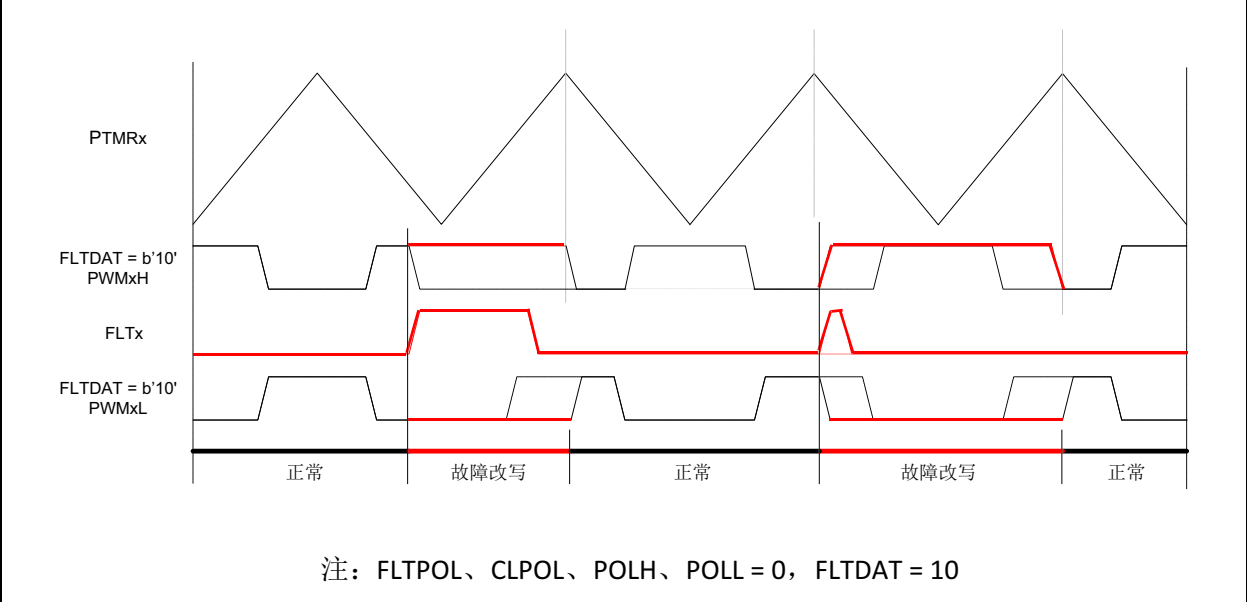
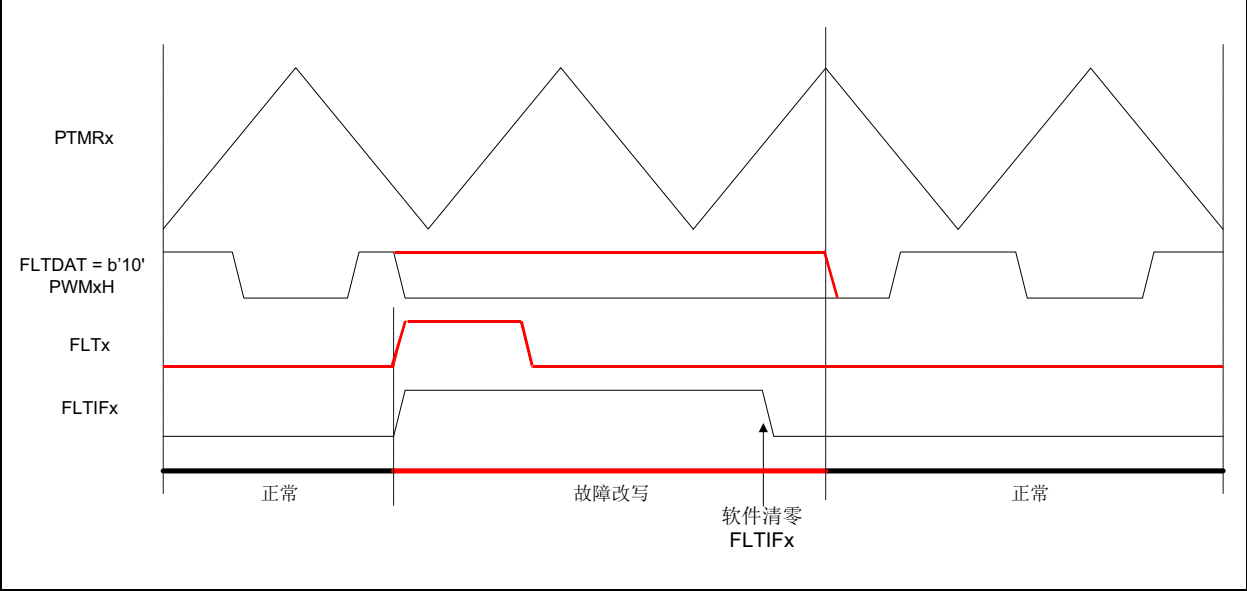


图 44-40： 故障时序锁定模式



### 44.13.4 故障进入

将按照 FLTDAT<1:0> 位的设置，将 PWMxH 和 PWMxL 引脚的故障条件置为有效。如果适用的位是逻辑 0，将发生引脚的异步复位。将在 1 SYSCLK 持续时间内发生立即关断。如果适用的位是逻辑 1，则在置为有效时将延迟相应的死区（DTRx, ALTDTRx）。

关于在响应限流或故障事件时的数据敏感性和行为的更多信息，请参见第 44.14.3 章“故障 / 限流改写和死区逻辑”。

### 44.13.5 故障退出

在故障条件结束之后，必须在 PWM 周期边界处恢复 PWM 信号，以确保正确地同步 PWM 信号边沿和手动信号改写。对于边沿对齐 PWM 模式，下个 PWM 周期在本地时基（PTMRx）值为 0 时开始。对于中心对齐 PWM 模式，下个 PWM 周期在本地时基（PTMRx）值为周期值减 1 时开始。

如果选择了逐周期故障模式，故障条件会在每个 PWM 周期自动复位。无需使用额外的软件来退出故障条件。

对于锁定故障模式，要退出故障条件，必须遵循以下序列：

1. 查询 FLT<sub>x</sub> 引脚的 FLTSTAT 或关联 GPIO 引脚，以确定故障信号是否已置为无效。
2. 如果 PWM 故障中断未使能，请跳过以下子步骤，继续执行步骤 3。如果 PWM 故障中断已使能，请执行以下子步骤，然后继续执行步骤 4。
  - a) 完成 PWM 故障中断服务程序（Interrupt Service Routine，ISR）。
  - b) 通过设置 FLTMOD<1:0> = 11，禁止 PWM 故障。
  - c) 通过设置 FLTMOD<1:0> 位（IOCON<sub>x</sub><17:16>）= 0b00，使能锁定 PWM 故障模式。
  - d) 通过清零 FLTIF 位（PWMCON<sub>x</sub><31>），清零故障中断标志。
  - e) 通过清零对应的 IFS 位，清零 PWM 中断标志。
3. 通过设置 FLTMOD<1:0> = 11，禁止 PWM 故障。
4. 通过设置 FLTMOD<1:0> 位 = 00，使能锁定 PWM 故障模式。

### 44.13.6 故障引脚软件控制

可以用软件对故障引脚进行手动控制。由于故障输入与 GPIO 端口引脚共用，所以可以通过清零相应的 TRIS<sub>x</sub> 位将该引脚配置为输出。当 GPIO 引脚的端口位置 1 时，故障输入将被激活。输入的极性或有效电平取决于在 FLTPOL 中做出的选择。

**注：** 来自 FLT<sub>x</sub> 引脚的所有数字输入都可以使用软件进行查询或控制，如本章节中所述。当 TRIS<sub>x</sub> 引脚置 1 时，可用软件查询状态。当 TRIS<sub>x</sub> 清零时，软件可以将故障、限流或死区补偿置为有效。  
当通过用户软件控制故障输入时，应特别注意。如果故障引脚的 TRIS<sub>x</sub> 寄存器的相应位清零，就无法从外部驱动故障输入。

## 44.14 PWM 限流

PWM 限流引脚的关键功能如下：

- 每个 PWM 发生器可以从最多 16 个故障和限流引脚和 / 或模拟比较器中选择自己的限流输入源。要将模拟比较器配置为一个限流源，请参见第 44.13.2 章“模拟比较器产生的 PWM 故障信号”。
- 每个 PWM 发生器都具有控制位，PWM 发生器 x 的限流控制信号源选择位 CLSRC<3:0> (IOCONx<29:26>)。这些位指定其限流输入信号的来源。
- 每个 PWM 发生器都具有相应的限流中断允许位 CLIEN (PWMCONx<22>)。该位用于允许限流 IRQ 的产生。
- 每个 PWM 发生器都具有 PWM 发生器的关联限流极性位 CLPOL (IOCONx<25>)。限流引脚通常高电平有效。如果 CLPOL 位设置为 1，将反转所选限流输入信号，使其变为低电平有效。
- 在发生限流条件时，如果限流改写功能已使能并处于有效状态，会将 PWMxH/PWMxL 引脚强制设置为 CLDAT<1:0> 位 (IOCONx<3:2>) 指定的值

CLDAT<1:0> 位 (IOCONx<3:2>) 的值可以为 0 或 1。如果 CLDAT<1:0> 位设置为 00，则它将进行异步处理，以便能够立即关断应用电路中的相关功率晶体管。如果 CLDAT<1:0> 设置为 11，则死区逻辑会对它进行处理，然后再应用于 PWM 输出。

限流信号可以产生中断。CLIEN 位 (PWMCONx<22>) 用于控制限流中断信号的产生。即使 CLMOD 位 (IOCONx<24>) 禁止限流改写功能，用户应用程序也可以指定产生中断。这使限流输入信号可以用作通用外部 IRQ 信号。

限流输入信号可以用作 ADC 的触发信号，用于启动模数转换过程。ADC 触发信号总是有效，无论 MCPWM 模块、CLMOD 位 (IOCONx<24>) 或 CLIEN 位 (PWMCONx<22>) 的状态如何。关于 ADC 触发源的更多信息，请参见具体器件数据手册中的“12 位高速逐次逼近寄存器 (SAR) 模数转换器 (ADC)”章节。

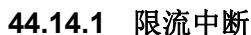
发生以下条件时，限流信号会复位受影响 PWM 发生器的时基：

- PWM 发生器的 CLMOD 位为 0
- 外部 PWM 复位控制位 XPRES (PWMCONx<1>) 为 1
- PWM 发生器处于独立时基模式 (ITB 位 (PWMCONx<9>) = 1)

该行为称为电流复位模式，一些 PFC 应用中会使用该模式。

图 44-41 说明了 PWM 限流控制电路逻辑。



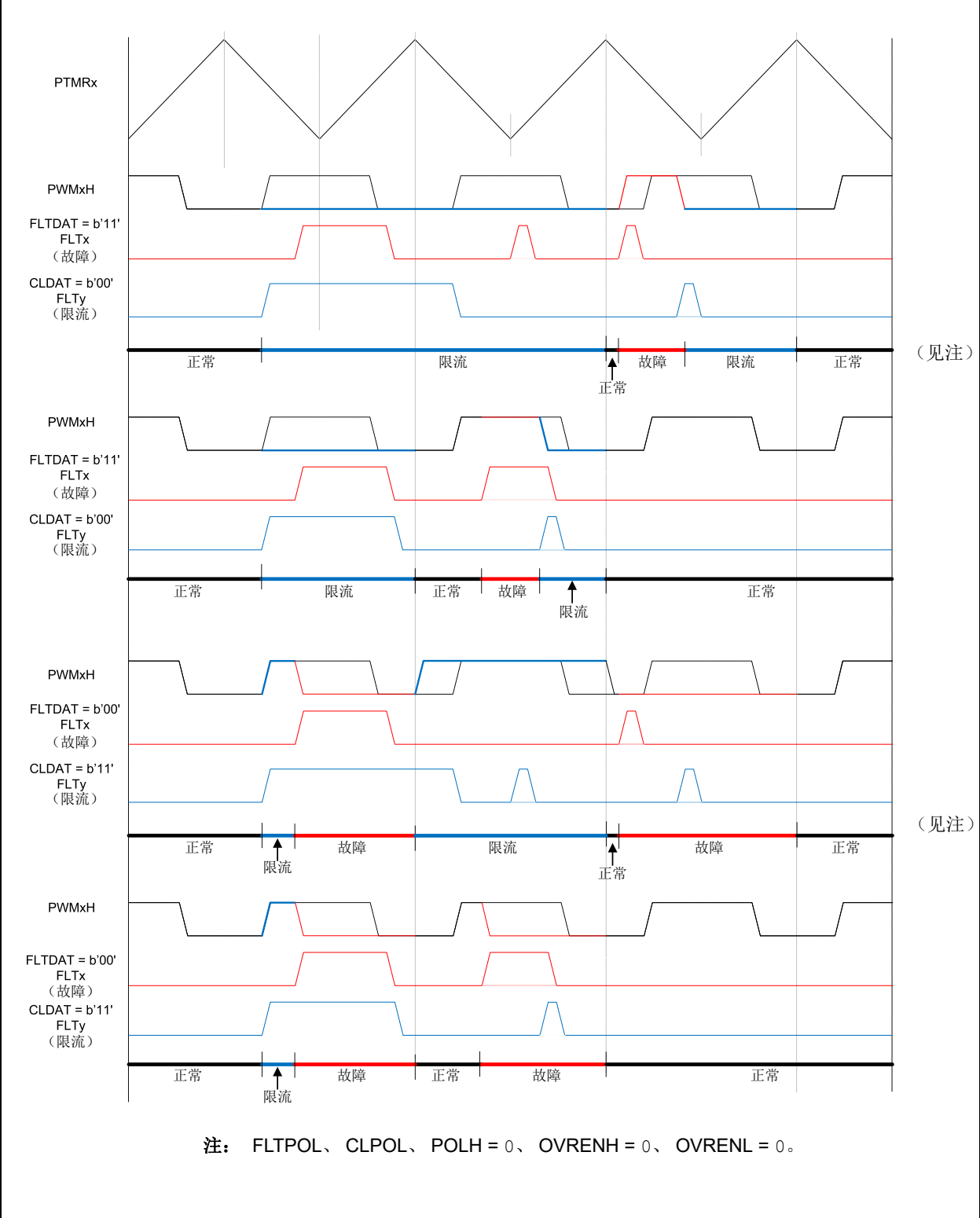


在使能 MCPWM 模块时，故障引脚还可以通过端口 I/O 逻辑读取。该功能使用户应用程序可以在软件中查询故障引脚的状态。

### 44.14.2 改写优先级

- 故障和限流处于相同的优先级，改写数据保存在 PWM 引脚上。
- 只要故障有效，FLTDAT<1:0>位（IOCONx<5:4>）就会改写手动改写数据并设置PWM输出。
- 只要限流有效，CLTDAT<1:0>位（IOCONx<3:2>）就会改写手动改写数据并设置 PWM 输出。
- 如果同时存在故障和限流，无效或已置于无效状态的改写数据将优先。这意味着只有在特定引脚的改写数据（CLDAT 和 FLTDAT）写入有效或已置为有效的数据时，引脚才会在适用的死区延时之后进入置为有效的状态。否则，它将异步进入置为无效的状态，而不会有任何死区延时。
- 如果故障和限流事件均无效，且向 OVRENH 位和 OVRENL 位设置用户改写使能，则关联的 OVRDAT<1:0>位（IOCONx<7:6>）将设置 PWM 输出。
- 如果改写条件都无效，则由时基和占空比比较器逻辑产生的 PWM 信号将作为用于设置 PWM 输出的信号源。

图 44-42： 改写优先级和极性对 PWM 引脚的影响



### 44.14.3 故障 / 限流改写和死区逻辑

在发生故障和限流条件时，FLTDAT<1:0> 位或 CLDAT<1:0> 位中的数据将决定 PWM I/O 引脚的状态。

如果 FLTDAT<1:0> 位或 CLDAT<1:0> 位包含无效或已置为无效的逻辑电平，PWMxH 和 / 或 PWMxL 输出会立即驱动为无效状态，将死区逻辑旁路。该行为会立即关断 PWM 输出，没有任何额外的延时。这可能会对许多需要快速响应故障关断信号以限制电路损坏和控制系统精度的电源转换 / 电机控制应用有帮助。

如果 FLTDAT<1:0> 位或 CLDAT<1:0> 位包含有效或已置为有效的逻辑电平，PWMxH 和 / 或 PWMxL 输出会立即驱动为有效状态，并通过死区逻辑，因而会被延迟一个指定的死区值。这种情况下，即使发生故障或限流条件，也会插入死区。

### 44.14.4 通过限流将输出置为有效

在响应限流事件时，CLDAT<1:0> 位可用于将 PWMxH 和 PWMxL 输出置为有效。这种行为可以用作一种电流强制功能，用于响应指示以下情况的外部电流或电压测量：电源转换器输出上的负载突然急剧上升。强制将 PWM 设为开启状态可以视为一种前馈操作，让系统可以快速响应意外的负载上升，而无需等待数字控制环进行响应。

**注：** 在互补 PWM 输出模式下，死区发生器会在所有改写条件下保持有效。输出改写和故障改写会产生由死区单元使用的控制信号，用于按请求设置输出（包括死区）。

## 44.15 并发 PWM 故障和限流

如果限流和故障事件已使能，则在 FLTDAT<1:0> 位（IOCONx<5:4>）或 CLDAT<1:0>（IOCONx<3:2>）位包含给定 PWM 通道置为无效的状态时，PWMxH/PWMxL 输出将置为无效。按命令置于无效的状态将被锁定，且优先于置于有效状态的限流或故障改写。

关于改写输入并行激活的影响的更多信息，请参见第 44.14.2 章“改写优先级”。

## 44.16 到 ADC 的 PWM 故障和限流触发输出

故障和限流源选择位 FLTSRC<3:0>（IOCONx<22:19>）和 CLSRC<3:0>（IOCONx<29:26>）可控制每个 PWM 发生器模块中的多路开关，该多路开关可选择对其各个模块可用的所需故障和限流信号。这些选定的故障和限流信号也提供给 ADC 模块，用作可用于启动 ADC 采样和转换操作的触发信号。

无论 PTEN 位（PTCON<15>）的状态如何，这些发送到 ADC 的故障和限流触发信号总是保持有效。

多路开关选择位 CLSRC<3:0> 和 FLTSRC<3:0> 以及极性控制位 CLPOL 和 FLTPOL（IOCONx<18>）总能有效控制发送到 ADC 的信号的选择和极性。

如果 PTEN 位为 1，将向这些信号应用前沿消隐（Leading Edge Blanking, LEB）功能。如果 PTEN 位为 0，将不会向限流和故障 ADC 触发信号应用 LEB 功能，故障和限流信号将直接连接到 ADC 模块。

发送到 ADC 的限流和故障触发信号也独立于 CLMOD 位（IOCONx<24>）和 FLTMOD<1:0> 位（IOCONx<17:16>）。

关于 ADC 触发源的更多信息，请参见具体器件数据手册中的“12 位高速逐次逼近寄存器（SAR）模数转换器（ADC）”章节。

例 44-19 提供了 PWM 故障、限流和 LEB 配置的代码。

**例 44-19: PWM 故障、限流和前沿消隐配置**

```
/* PWM Fault, Current-Limit and Leading-Edge Blanking Configuration */

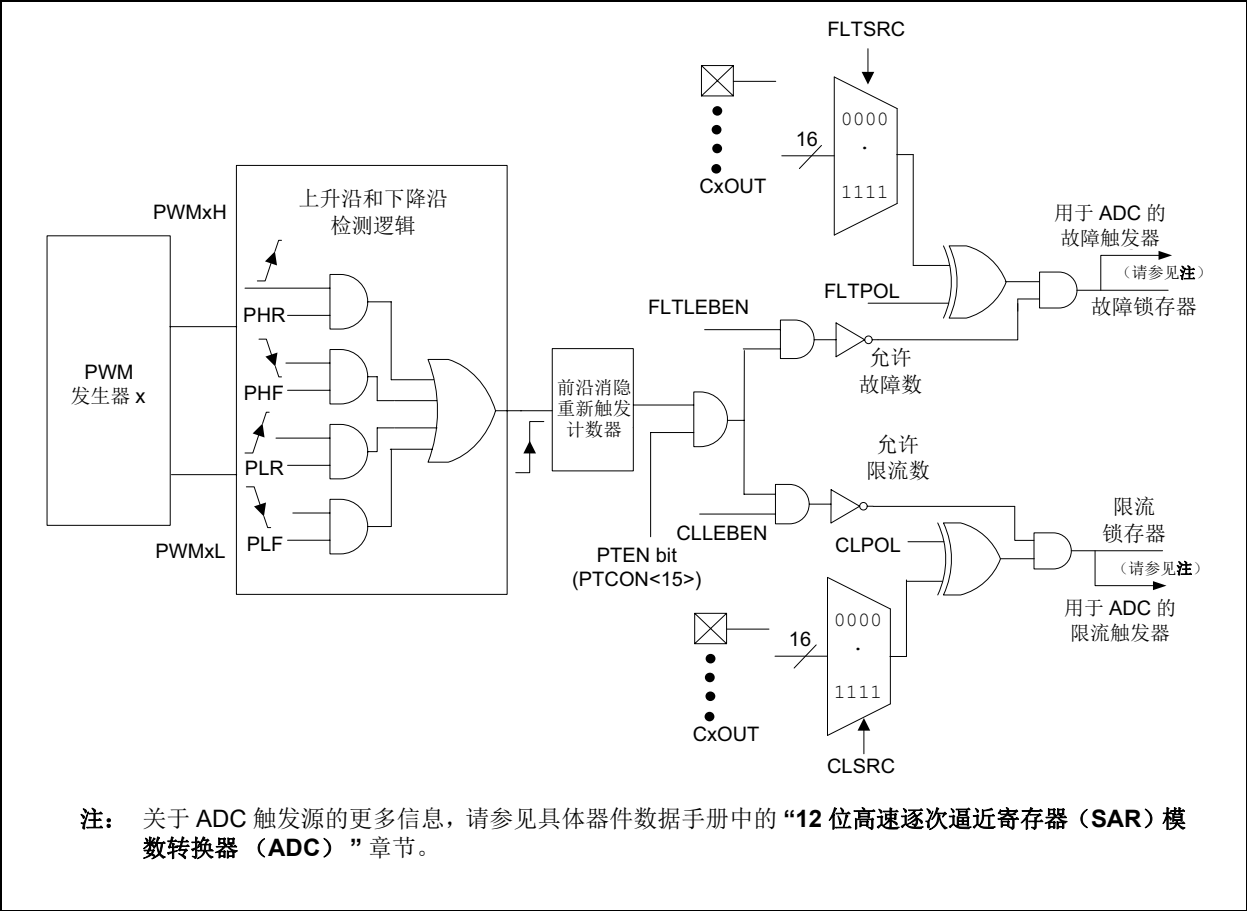
IOCON1bits.FLTMOD = 0;      /* CLDAT<1:0> bits control PWMxH; FLTDAT<1:0> bits control PWMxL */
IOCON1bits.CLSRC = 8;       /* Current-limit input source is Analog Comparator 1 */
IOCON1bits.FLTSRC = 9;      /* Fault input source is Analog Comparator 2 */
IOCON1bits.CLPOL = 1;       /* Current-limit source is active-low */
IOCON1bits.FLTPOL = 1;      /* Fault Input source is active-low */
IOCON1bits.CLMOD = 1;       /* Enable current-limit function */
IOCON1bits.FLTMOD = 1;      /* Enable Cycle-by-Cycle Fault mode */
IOCON1bits.FLTDAT = 0;      /* PWMxH and PWMxL are driven inactive on occurrence of fault */
IOCON1bits.CLDAT = 0;       /* PWMxH and PWMxL are driven inactive on occurrence of current-limit */

LEBCON1bits.PHR = 1;        /* Rising edge of PWMxH will trigger LEB counter */
LEBCON1bits.PHF = 0;        /* Falling edge of PWMxH is ignored by LEB counter */
LEBCON1bits.PLR = 1;        /* Rising edge of PWMxL will trigger LEB counter */
LEBCON1bits.PLF = 0;        /* Falling edge of PWMxL is ignored by LEB counter */
LEBCON1bits.FLTLEBEN = 1;   /* Enable fault LEB for selected source */
LEBCON1bits.CLLEBEN = 1;    /* Enable current-limit LEB for selected source */
LEBDLY1bits.LEB = 8;        /* Blank for a period (8 x TOSC) */

PWMCON1bits.XPRES = 0;      /* External pins do not affect PWM time base reset */
PWMCON1bits.FLTEN = 1;      /* Enable fault interrupt */
PWMCON1bits.CLIEN = 1;      /* Enable current-limit interrupt */

while (PWMCON1bits.FLTSTAT == 1); /* Wait when fault interrupt is pending */
while (PWMCON1bits.CLSTAT == 1);  /* Wait when current-limit interrupt is pending */
```

图 44-43： 限流和故障触发机制



## 44.17 特殊功能

MCPWM 模块具有以下特殊功能：

- 前沿消隐（LEB）
- 斩波模式
- 独立时基捕捉
- PWM 引脚交换
- PWM 输出引脚控制和改写
- 双重更新 PWM 模式
- 死区补偿

### 44.17.1 前沿消隐（LEB）

每个 PWM 发生器都支持使用前沿消隐控制寄存器 **LEBCONx** 和 **LEBDLYx** 中的位，对配置的限流和故障输入执行 LEB 功能。LEB 的用途是屏蔽在功率晶体管导通和关断时，在应用印刷电路板上产生的瞬变。

LEB 位是边沿敏感的。这些位支持在 **PWMxH** 和 **PWMxL** 信号任意指定的上升沿或下降沿后，将限流输入和故障输入消隐（忽略）0 至  $(4095 * TSYSCLK)$ 。公式 44-11 给出了 LEB 的计算。

#### 公式 44-11： 前沿消隐计算

$$LEB\ Duration = (LEBDLYx<11:0>) \cdot \frac{1}{FSYSCLK}$$

其中，  
 $LEBDLYx$  = 所需前沿消隐时间

在高速开关应用中，开关（例如 MOSFET 和 IGBT）通常会产生极大的瞬变。这些瞬变可能会导致测量误差。利用 LEB 功能，用户应用程序可以忽略预期会由靠近 PWM 输出信号边沿发生的 MOSFET/IGBT 开关所导致的瞬变。

前沿消隐控制寄存器（**LEBCONx<15>**）中的 **PWMxH** 上升沿触发使能位（**PHR**）、**PWMxH** 下降沿触发使能位 **PHF**（**LEBCONx<14>**）、**PWMxL** 上升沿触发使能位 **PLR**（**LEBCONx<14>**）以及 **PWMxL** 下降沿触发使能位 **PLF**（**LEBCONx<12>**）将选择启动消隐定时器的 **PWMxH** 和 **PWMxL** 信号的边沿类型。如果新选定的边沿在先前选定的 PWM 边沿对于 LEB 定时器仍然有效时触发了定时器，定时器会重新初始化并继续计数。

故障输入前沿消隐使能位 **FLTLEBEN**（**LEBCONx<11>**）和限流前沿消隐使能位 **CLLEBEN**（**LEBCONx<11>**）都支持向选定故障和限流输入应用消隐周期。

图 44-44 说明了如何屏蔽故障和限流信号，以避免启动错误的改写条件。图中展示了可以用来重新触发延时计数器的不同边沿和关联配置位。

图 44-44： 故障和限流处理的前沿消隐

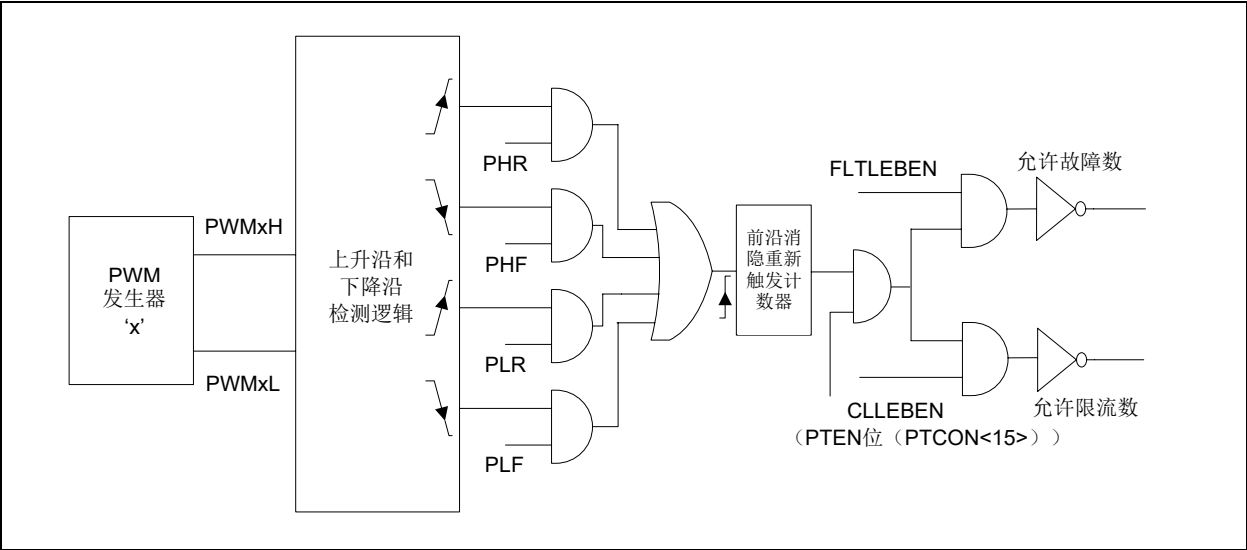
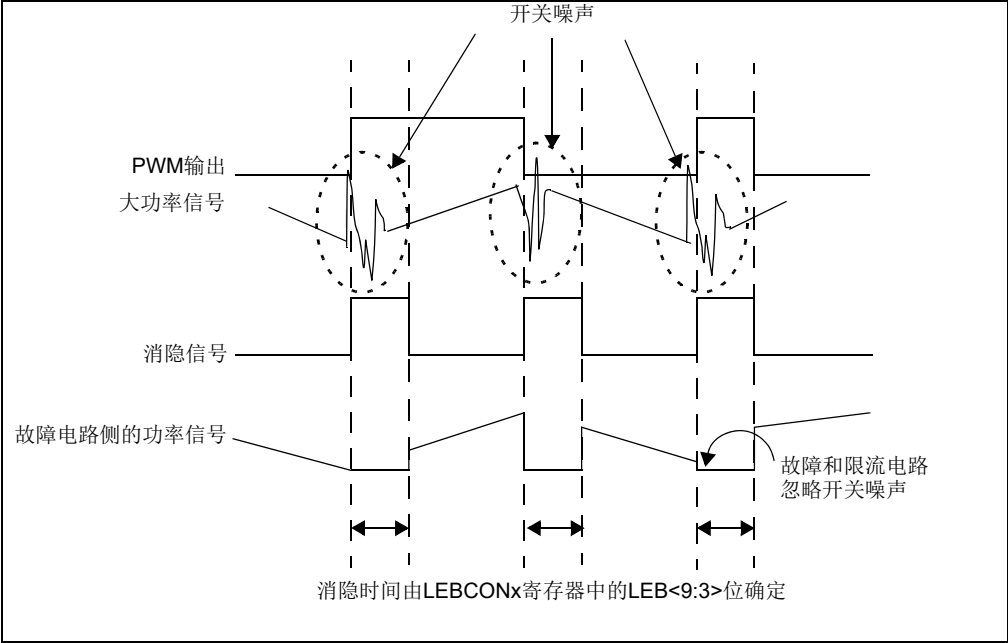


图 44-45： 前沿消隐



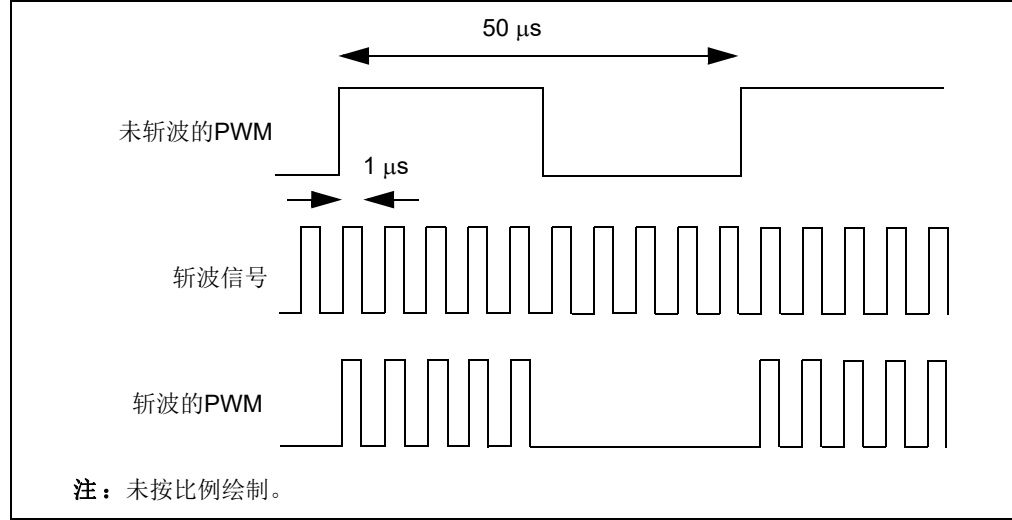
## 44.17.2 斩波模式

许多电源控制应用都使用需要隔离晶体管栅极驱动的晶体管配置。例如，在一个三相“H 桥”配置中，上桥臂晶体管处于抬高电势。

实现隔离栅极驱动电路的一个方法是使用脉冲变压器，使 PWM 信号通过电流隔离势垒耦合到晶体管。遗憾的是，在使用大占空比或低 PWM 频率的应用中，变压器的低频响应很差。脉冲变压器无法将长时的 PWM 信号传递给隔离的晶体管。如果通过高频时钟信号对 PWM 信号进行“斩波”或门控，高频交变信号可以很容易地通过脉冲变压器。与 PWM 频率相比，斩波频率通常会高几百或几千倍。斩波（载波）频率相对于 PWM 频率越高，可获得的 PWM 占空比分辨率就越高。

图 44-46 给出了一个 MCPWM 斩波的示例波形。在该示例中，20 kHz PWM 信号使用通过斩波时钟产生的 500 kHz 载波进行斩波。

图 44-46： 高频 PWM 斩波



斩波功能会对 PWM 输出执行逻辑与操作。

用户可以通过 PWM 斩波时钟发生器寄存器 (CHOP) 指定斩波时钟频率。斩波值用于指定 PWM 时钟分频比。斩波时钟分频器以 PCLKDIV<2:0> 位 (PTCON<6:4>) 和 SCLKDIV<2:0> 位 (STCON<6:4>) 指定的 PWM 时钟频率工作。使能斩波时钟发生器位 CHPCLKEN (CHOP<15>) 可使能斩波时钟发生器。

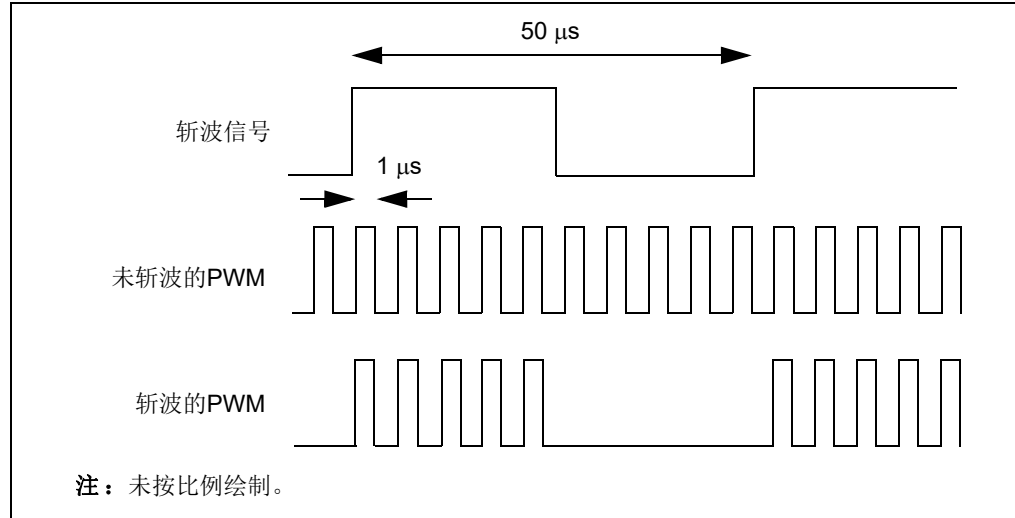
PWMxH 输出斩波使能位 CHOPHEN (AUXCONx<1>) 和 PWMxL 输出斩波使能位 CHOPLEN (AUXCONx<0>) 用于使能要应用于 PWM 输出的斩波时钟。PWM 斩波时钟源选择位 CHOPSEL<3:0> (AUXCONx<5:2>) 用于选择所需的斩波时钟源。默认选择是由 CHOP 寄存器控制的斩波时钟发生器。通过 CHOPSEL<3:0> 位 (AUXCONx<5:2>)，用户可以选择其他 PWM 发生器作为斩波时钟源。

如果 CHOPHEN 位或 CHOPLEN 位置 1，则在对 PWM 信号应用限流和故障功能之后，会对 PWM 输出信号应用斩波功能。斩波时钟信号可从模块中输出，用作器件的输出信号。

通常，斩波时钟频率高于 PWM 周期频率，但新应用可以使用远低于 PWM 周期频率的斩波时钟频率。图 44-47 给出了一个低频 PWM 斩波波形，它还说明，另一个在较低频率工作的 PWM 发生器对 PWM 信号进行斩波或“消隐”。



图 44-47: 低频 PWM 斩波



## 44.17.3 独立时基捕捉

每个 PWM 发生器都具有一个主 PWM 时基捕捉寄存器 (CAPx)，该寄存器会在检测到限流信号上升沿时自动捕捉独立时基计数器值。该功能仅在应用 LEB 功能之后有效。用户应用程序应在下一个 PWM 周期导致捕捉寄存器再次更新之前读取寄存器。

在使用模拟比较器或外部电路来终止 PWM 占空比或周期的电流模式控制应用中，需要使用捕捉寄存器。通过读取达到电流阈值时的独立时基值，用户应用程序可以计算电感中的电流上升斜率。辅助独立时基不具有关联的捕捉寄存器。

## 44.17.4 PWM 引脚交换

通过交换 PWMxH 和 PWMxL 引脚位 SWAP (IOCONx<1>) (如果设置为 1)，用户应用程序可以将 PWMxH 信号与 PWMxL 引脚连接，将 PWMxL 信号与 PWMxH 引脚连接。如果 SWAP 位设置为 0，PWM 信号将与它们各自相应的引脚连接。

要在 PWM 周期边界处执行交换功能，输出改写同步位 OSYNC (IOCONx<0>) 必须置 1。如果在模块正在工作，并且 OSYNC 位清零时，用户应用程序更改了 SWAP 位的状态，SWAP 功能会尝试在 PWM 周期中途执行操作，操作将产生不可预测的结果。

SWAP 功能应在应用死区之前执行。由于执行切换功能可能会使用户应用中先前处于禁止状态的晶体管，可能导致产生电流直通，所以需要进行死区处理。

对于通过单个应用电路板支持多种开关拓扑的应用，交换功能很有用。它也使用户应用可以通过改变晶体管调制方案来响应不断变化的条件。

交换功能可以通过以下方法之一实现：

- 动态交换：在此方法中，可以根据系统响应动态更改 SWAP 位的状态（例如，开关电源 (Switch Mode Power Supply, SMPS) 控制）
- 静态交换：在此方法中，SWAP 位在启动配置期间置 1，在程序执行或运行期间保持不变（如电机控制）

### 44.17.4.1 例 1：SMPS 电源控制的引脚交换

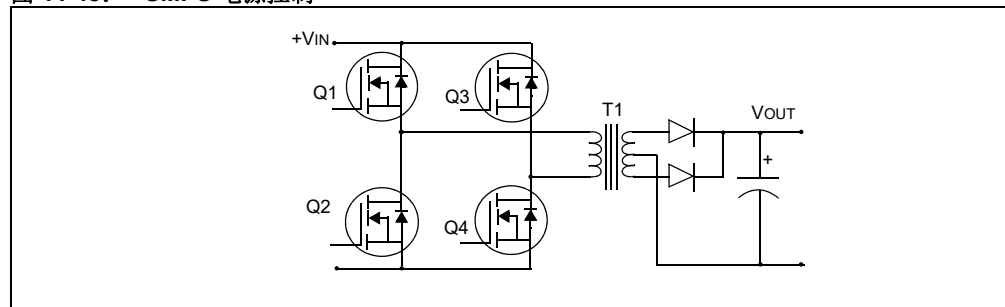
SMPS 电源控制示例介绍动态交换。在电源转换 / 电机控制应用中，晶体管调制技术可以在全桥零电压转换 (Zero Voltage Transition, ZVT) 和标准的全桥“动态”转换之间切换，以满足不同的负载和效率要求。如图 44-48 所示，通用全桥转换器可以工作于推挽模式。晶体管配置如下：

- Q1 = Q4
- Q2 = Q3

通用全桥转换器也可以工作于 ZVT 模式。晶体管配置如下：

- Q1 = PWM1H
- Q2 = PWM1L
- Q3 = PWM2H
- Q4 = PWM2L

图 44-48： SMPS 电源控制



## 44.17.4.2 例 2：电机控制的引脚交换

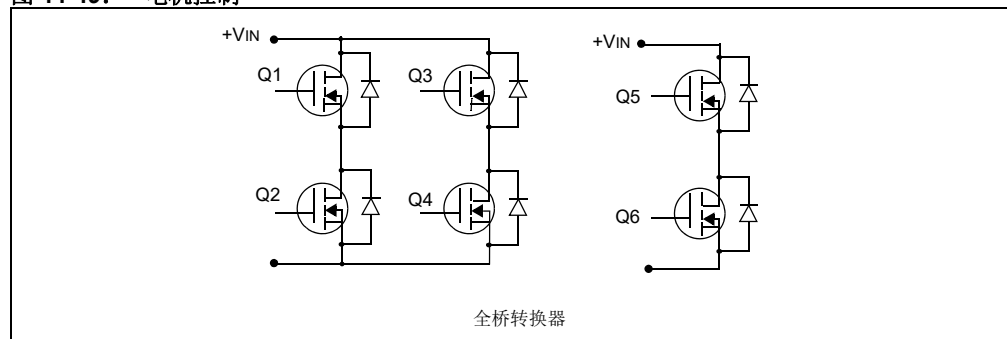
电机控制示例介绍静态交换。假设有一个通用电机控制系统，它可以驱动两种不同类型的电机，例如直流电机和三相交流电感电机。

有刷直流电机通常使用全桥晶体管配置，如图 44-49 所示。Q1 和 Q4 晶体管使用类似波形驱动，而 Q2 和 Q3 晶体管使用互补波形驱动。这也称为“对角线驱动”。请注意，有刷直流电机中不使用 Q5 和 Q6 晶体管。

晶体管配置如下：

- Q1 = PWM1H
- Q2 = PWM1L
- Q3 = PWM2L
- Q4 = PWM2H

图 44-49： 电机控制



与直流电机不同，交流电感电机使用全桥配置中的所有晶体管。但是，明显的一点区别是现在晶体管以三个半桥形式驱动，其中，高端晶体管由 PWMxH 输出驱动，而低端晶体管则由 PWMxL 输出驱动。

晶体管配置如下：

- Q1 = PWM1H
- Q2 = PWM1L
- Q3 = PWM2H（请注意与直流电机的区别）
- Q4 = PWM2L（请注意与直流电机的区别）
- Q5 = PWM3H
- Q6 = PWM3L

## 44.17.5 PWM 输出引脚控制和改写

如果使能了 MCPWM 模块，则从最低优先级到最高优先级，PWMxH/PWMxL 引脚所有权的优先级如下：

- PWM 发生器（最低优先级）
- 交换功能
- PWM 输出改写逻辑
- 限流和故障改写逻辑
- GPIO/PWM 所有权（最高优先级）

如果禁止了 MCPWM 模块，则 GPIO 模块将控制 PWMx 引脚。

例 44-20 提供了 PWM 输出引脚分配的代码，例 44-21 提供了 PWM 输出引脚状态选择的代码，例 44-22 则提供了用于使能 MCPWM 模块的代码。

### 例 44-20: PWM 输出引脚分配

```
/* PWM Output pin control assigned to PWM generator */
IOCONbits.PENH = 1;
IOCONbits.PENL = 1;
```

### 例 44-21: PWM 输出引脚状态选择

```
/* High and Low switches set to active-high state */
IOCONbits.POLH = 0;
IOCONbits.POLL = 0;
```

### 例 44-22: 使能电机控制 PWM (MCPWM) 模块

```
/* Enable Motor Control PWM module */
PTCONbits.PTEN = 1;
```

#### 44.17.5.1 PWM 输出改写逻辑

PWM 输出改写功能用于根据系统要求，将各个 PWM 输出驱动为所需的状态。输出可以驱动为有效状态，也可以驱动为无效状态。

改写功能具有以下降序排列的优先级：

- 故障和限流改写
- 手动改写

所有与 PWM 输出改写功能相关的控制位都包含在 IOCONx 寄存器中。如果 PWMxH 输出引脚所有权 PENH (IOCONx<15>) 和 PWMxL 输出引脚所有权 PENL (IOCONx<14>) 置 1，则 MCPWM 模块将控制 PWMx 输出引脚。PWM 输出改写位允许用户应用程序手动将 PWM I/O 引脚驱动为指定逻辑状态，而不受占空比比较单元的影响。

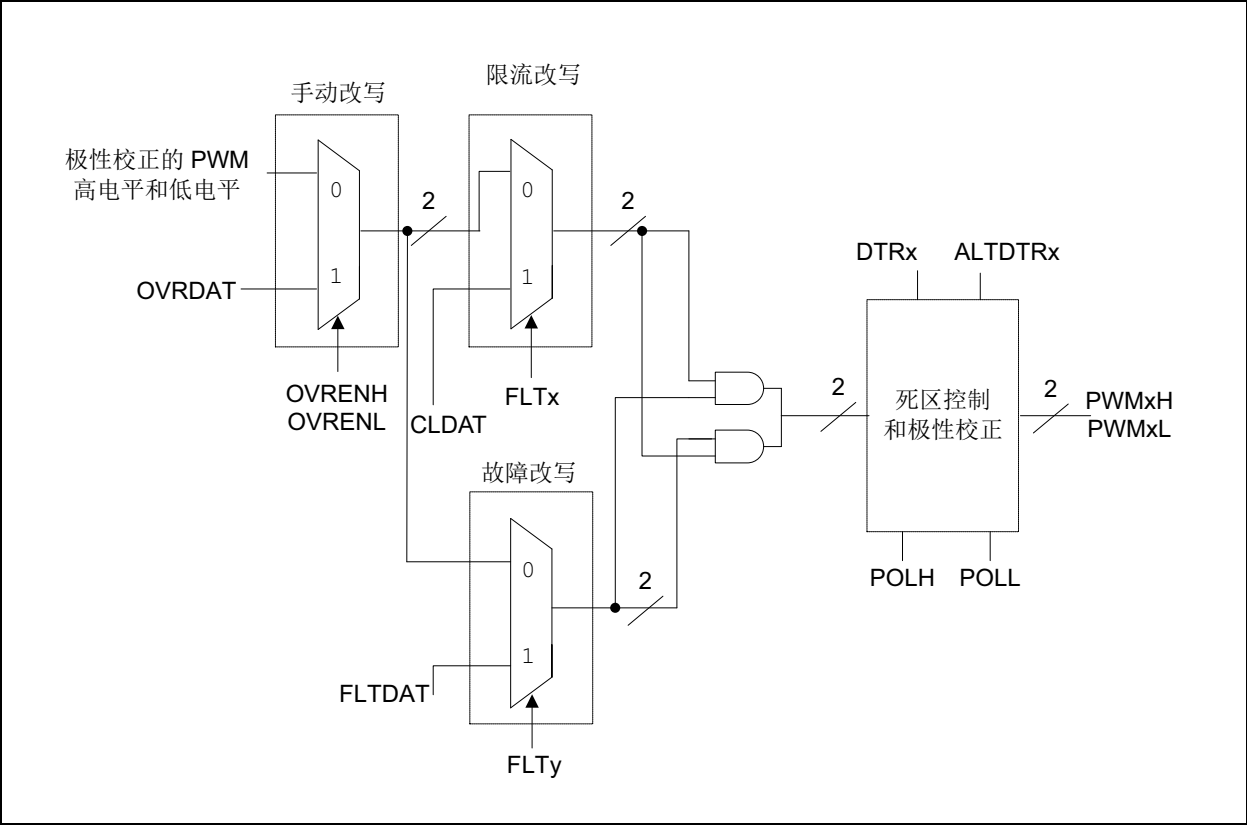
手动改写处于有效状态时 PWMxH 和 PWMxL 引脚的状态由改写数据位 OVRDAT<1:0> (IOCONx<7:6>) 决定。

手动改写可以使用属于高电平有效控制位的改写使能位 OVERNH 位 (IOCONx<9>) 和 OVERNL 位 (IOCONx<8>)，在 PWMxH 和 PWMxL 引脚上单独执行。

无论是什么原因，所有改写都将由已使能的死区控制逻辑进行处理。这意味着低电平有效改写数据将被异步反映在引脚上，而高电平有效改写数据将在根据引脚在死区持续时间之后变为有效，如同正常死区发生中一样。详细信息，请参见第 44.8.3 章“死区发生”章节。

图 44-50 说明了故障和限流条件同时发生时改写的优先级及其相对于 PWM 极性的仲裁。

图 44-50: PWM 引脚上的故障和限流改写优先级



44.17.5.2 PENX（GPIO/PWM）所有权

大多数 PWM 输出引脚通常都与其他 GPIO 引脚复用。当调试器暂停器件时，PWM 引脚将具有在该引脚上进行复用的 GPIO 的特性。例如，如果 PWM1L 和 PWM1H 引脚与 RE0 和 RE1 复用，则 GPIO 引脚的配置将决定调试器暂停器件时的 PWM 输出状态。

例 44-23 给出了 GPIO 配置的代码。

例 44-23: GPIO 配置代码示例

```
/* PWM output will be pulled to low when the device is halted by the
   debugger */
TRISE = 0x0000; RE0 and RE1 configured for an output
LATE = 0x0000; RE0 and RE1 configured as Low output

/* PWM output will be pulled to high when the device is halted by the
   debugger */
TRISE = 0x0000; RE0 and RE1 configured for an output
LATE = 0x0003; RE0 and RE1 configured as High output

/* PWM output will be in tri-state when the device is halted by the
   debugger */
TRISE = 0x0003; RE0 and RE1 configured for an input
```

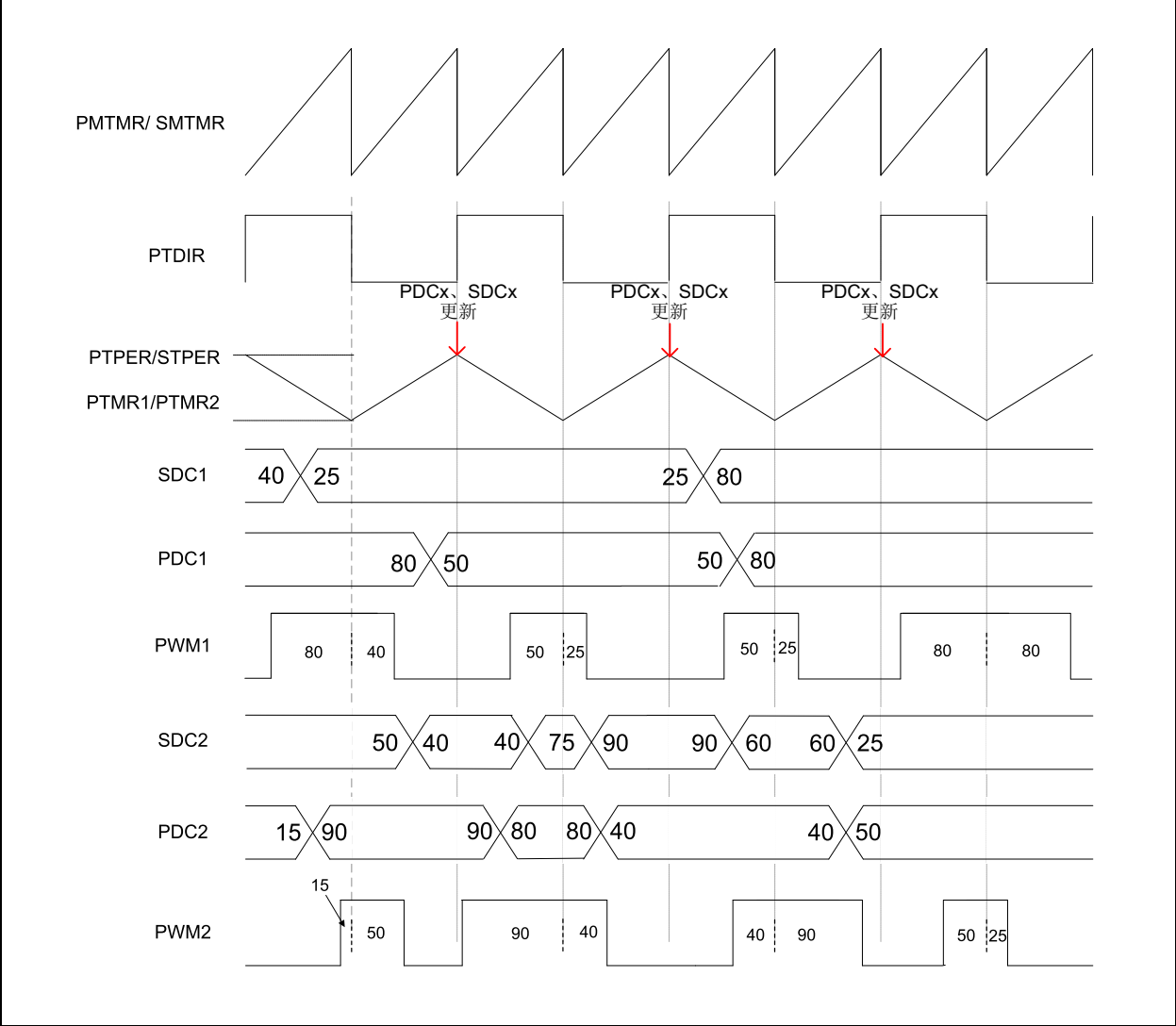
44.17.6 双重更新速率和同时更新模式

双周期模式提供了两个在单个 PWM 周期中更新占空比的机会。因此每组同步的发生器都可以在周期匹配和计数器计满返回时使用新的占空比进行更新，这实际上可以将更新速率加倍（与之前的发生器件相比）。只有非对称中心对齐模式才能使用双重更新速率。

在电机速度较快时，逆变器会由于有限的处理和定时器功能而产生不那么理想的波形，从而导致向 THD 添加谐波失真。在双重更新速率下，谐波失真会减少，因为波形更接近于理想状态。在正常双周期模式中，定时器分辨率将以 2 为比率因数降低。双重更新速率将重新获得损失的分辨率。

图 44-51 说明了对 PWM 波形的更新，其中对 PDCx 和 SDCx 的更新在软件中同时完成。关于双重更新速率模式，请参见图 44-27。

图 44-51： 非对称中心对齐模式中的同时占空比更新模式



双重更新速率模式通过设置 `ECAM<1:0>` 位 = `0b10` 进行配置。在此模式中，如果需要快速更新，则需要单独写入 PDCx 和 SDCx 寄存器。这两个寄存器一旦写入，就会产生预定波形，而无需更新每个周期。PDCx 寄存器在 PTMRx 的递增计数（PTDIR = 0）期间写入，SDCx 则在计数器递减计数（PTDIR = 1）时写入。这将确保边沿跳变的更新，因此占空比会尽早发生。

同时更新模式通过设置  $ECAM<1:0>$  位 = 0b11 进行配置。在此模式中，写入  $PDCx$  和  $SDCx$  寄存器将在双周期中的计数器重新装入时同时生效。此模式主要用于通过额外写入  $SDCx$  寄存器（除了中心对齐模式正常要求的  $PDCx$  寄存器外），在对称中心对齐模式中重新获得损失的分辨率。关于如何在双重更新速率和同时非对称中心对齐模式中配置通道 / 发生器，请参见图 44-16。

### 44.17.6.1 非对称同步 PWM 模式

对于电机应用，为三相星形或三角连接绕线电机产生三相正弦波的传统方法是使用三角调制方案实现的。较新的现代方法向波形添加了一个零序列。对于三接线端（没有电流通过星形连接电机中的零线），比如在电机中，则用来增强调制 PWM 的零序列非正弦将提供额外的波形产生自由度，因为零线和进行 2 分频的 VDC 中的直流链路参考之间的电压可以形成任何波形。为改变三相中每一相上的占空比，添加了一个精心设计的零序列波形，这不会影响每个 PWM 周期的平均线间电压。

仔细选择的零序列可实现一个或多个以下所需属性。

- 更低的开关损耗。
- 更低的总谐波失真
- 扩展的调制深度 / 系数
- 更低的公共模式驱动电压

通过具有双重和同时更新功能的异步 PWM 模式，可以产生同步多相波形，这些波形相对于彼此不连续且对称或不对称。

当增强零序列波形为连续时，将产生连续的 PWM 方案，如图 44-52 所示。

当零序列为不连续，且具有使 PWM 达到正或负轨道（VDC 或 0）的足够高幅值时，将会产生不连续的 PWM，如图 44-53 所示。在针对降低开关损耗进行设计时，对于相同的输送功率，不连续的 PWM 方案会大幅降低开关损耗。此方案在高环境温度应用中为首选。

图 44-54 给出了电压波形的不连续和非对称载波调制如何产生更低失真度的三相电流。此方法在必须降低开关损耗的情况下为首选。

所有前述方案可根据环境温度和负载变化，应用于不同的时间。

在有意降低调制频率以保持低开关损耗的应用中，双重更新速率很有用。当三相波形的基频接近调制频率的十分之一时，由于相对输出采样速率降低，波形完整性开始偏离理想状态。双重更新通过使输出采样速率有效倍增，可帮助恢复波形。

图 44-52： 使用常规 SVPWM 的对称连续电压调制

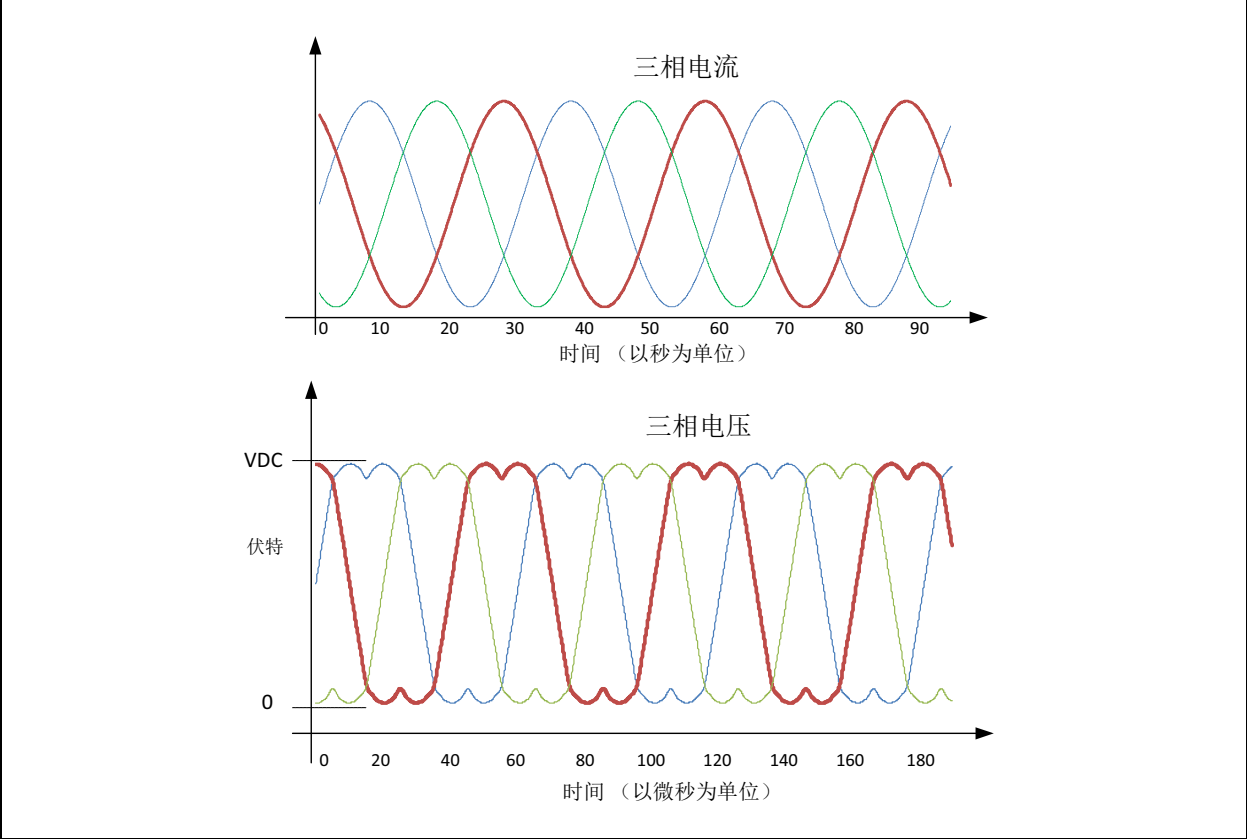


图 44-53： 对称不连续电压调制

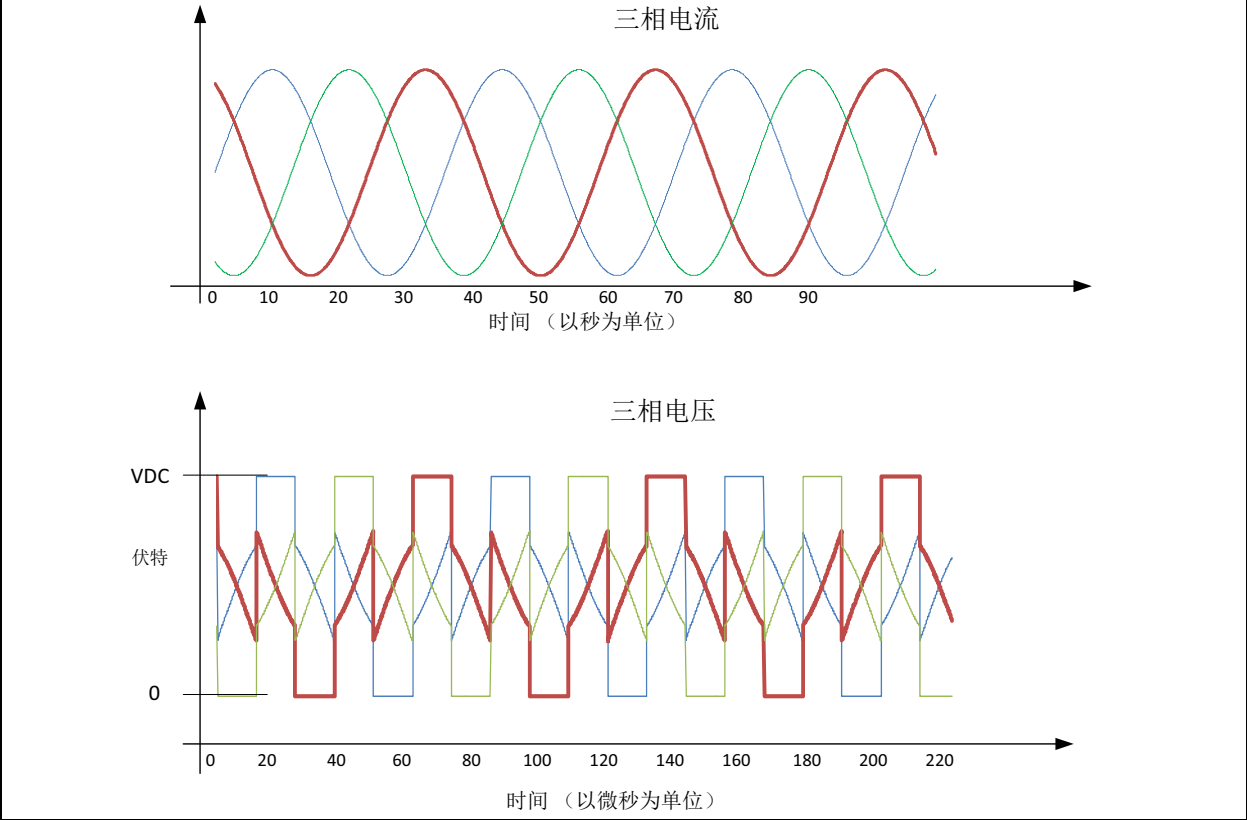
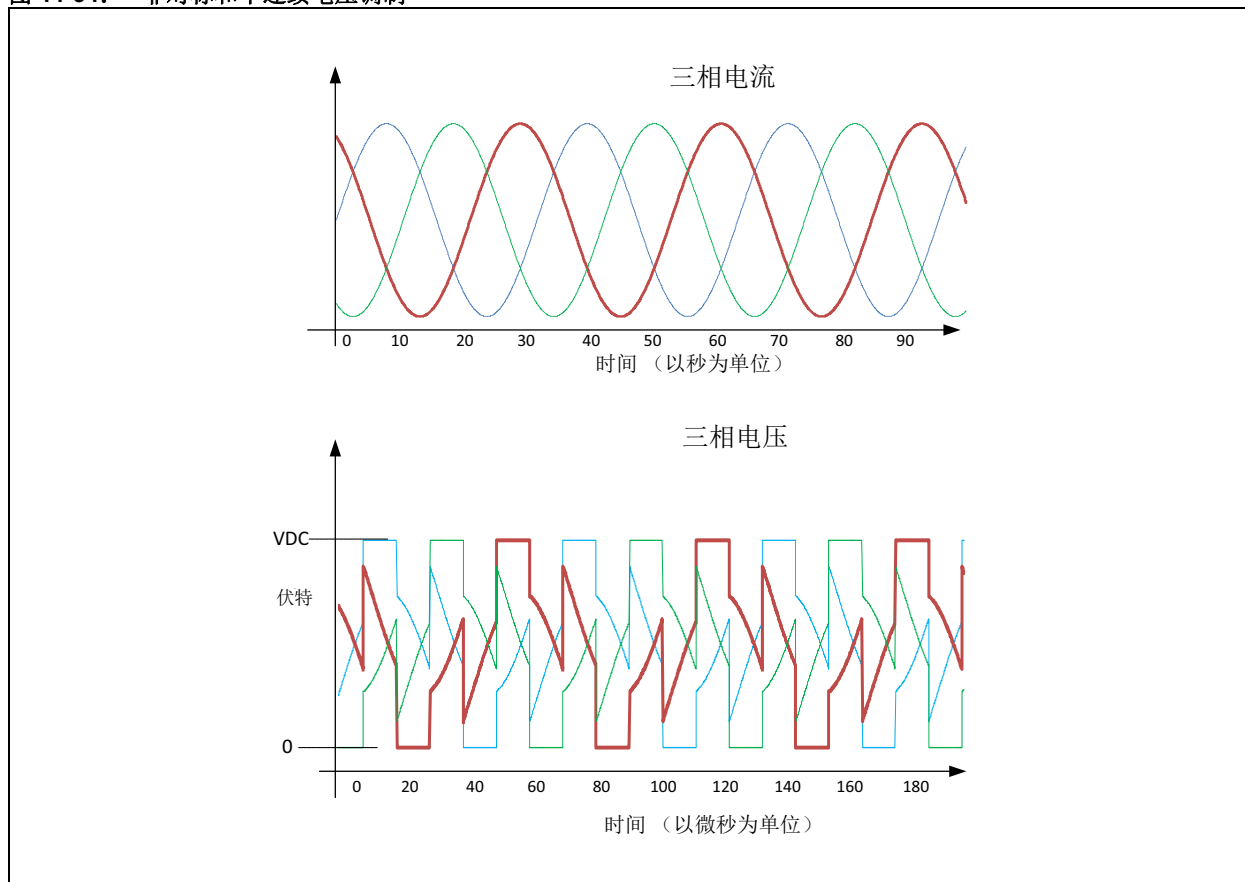




图 44-54: 非对称和不连续电压调制



44.17.7 死区补偿

死区补偿大多用于基于 PWM 的三相逆变器，以进行电机控制。

补偿的目的基本上是为降低相位电流中的谐波失真。失真的重要原因是“传导死区”，它会降低或提高软件预定的相位电压波形伏秒，这反过来会影响上桥臂和下桥臂功率器件的电流导通时间。

死区（DTRx，ALTDTRx）是在低速和低扭矩（即，低占空比）时导致相位电压波形的伏秒失真的重要因素。

在已知相位中电流方向的情况下，可以对死区进行补偿。电流通常使用比较器进行运行时过零比较，并使用数字输入 DTCMPx 来自动补偿硬件中较高和较低的占空比。专用模拟输入上可用的数字比较器可用于监控电流幅值 / 方向变化，以改善死区补偿。

图 44-55 给出了使用单相常规空间矢量调制的所需理想线间电压和电流波形及其相应的实际或非理想波形。为清晰起见，对失真进行了放大。完全补偿的逆变器可实现接近理想的电流波形，几乎无谐波失真。

图 44-55： 理想 / 补偿的波形与未补偿的波形

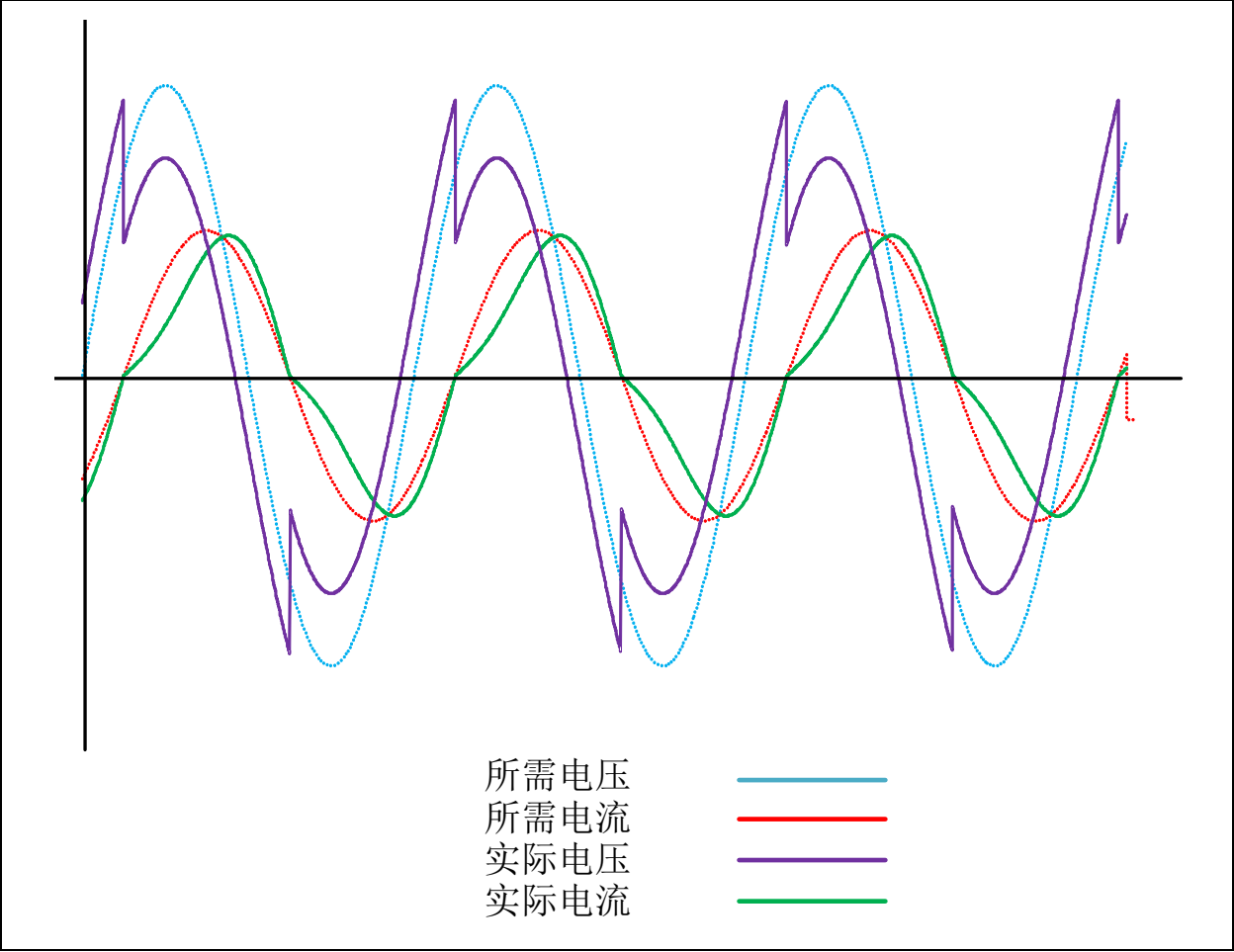
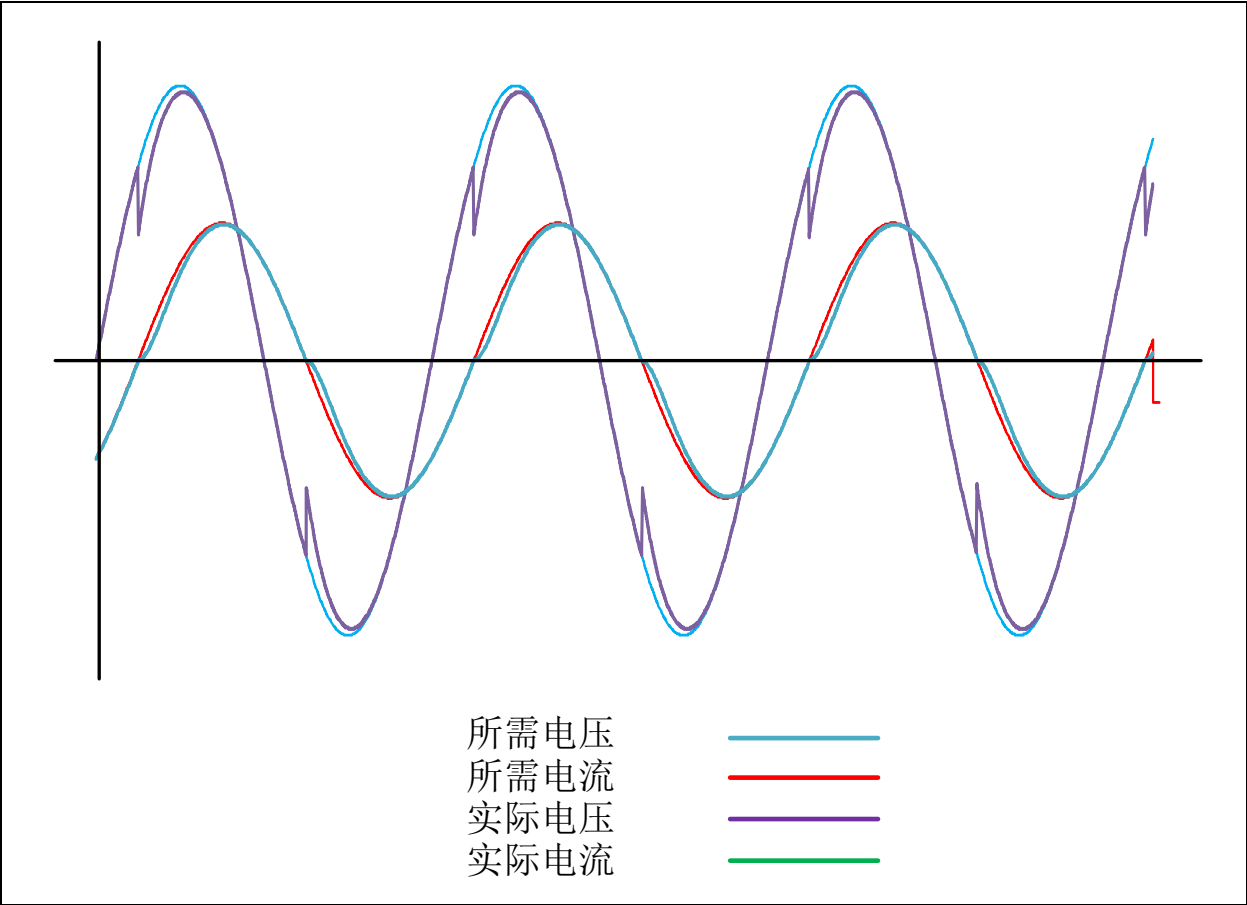


图 44-56 给出了当占空比降低到死区以下或超过“周期死区”值时补偿不足的线间电压和电流波形。此外，该图还展示了死区对电流谐波失真的影响以及调整死区值后使用补偿执行的校正。电压以及最终的电流波形表现出严重的失真，且此失真在较低频率下将会加重。在幅值变小（较低占空比）时失真会变得更加严重。有关详细信息，请参见第 44.8.5 章“死区失真”。为支持所有占空比下的完全补偿，MCPWM 模块能够对死区寄存器进行运行时更新。DTCOMPx 值仅会在所产生的实际死区（DTRx/ALTDTRx 中的值）范围内进行补偿，因此在上述条件下仅仅向 DTCOMPx 寄存器写入更高的值并不会有效果，并且可能导致波形失真。

图 44-56： 占空比极值下的谐波失真和缓解



当死区寄存器进行动态更新时（如公式 44-12 所示），可以进一步减少谐波失真。

公式 44-12: 为减少谐波失真进行死区调整

当所需占空比超过最大占空比 ( $MaxDuty$ ) 时 :

$$DTRx = HWLDT - \frac{(Period - MaxDuty)}{2}$$

$$ALTDTRx = HWTDT - \frac{(Period - MaxDuty)}{2}$$

当所需占空比小于最小占空比 ( $MinDuty$ ) 时 :

$$DTRx = HWLDT - \frac{(MinDuty - Period)}{2}$$

$$ALTDTRx = HWTDT - \frac{(MinDuty - Period)}{2}$$

当所需占空比介于最小占空比和最大占空比 ( $MinDuty$ 和 $MaxDuty$ ) 之间时 :

$$DTRx = HWLDT$$

$$ALTDTRx = HWTDT$$

其中 :

$周期 = STPER/PTPER/PHASEx$   
 $HWLDT$  = 硬件所需的前沿死区计数  
 $HWTDT$  = 硬件所需的下降沿死区计数  
 $MaxDuty$  = 周期 -  $ALTDTRx$  -  $DTRx$   
 $MinDuty$  =  $ALTDTRx$  +  $DTRx$

当占空比超过周期减去死区之和所限定的范围时, 将出现波形失真。发生这种情况是因为占空比饱和且更新没有效果。单单向  $DTCOMPx$  寄存器写入无法纠正这种情况。需要减少该周期的死区寄存器, 因此需要向  $DTRx$  和  $ALTDTRx$  进行运行时写入。当死区值降低时, 可以通过同时从两个死区寄存器减去相应值, 以对称方式完成写入。无论 PWM 占空比是否超过限定范围, 每个周期都必须完成这个策略。当占空比未超过限定范围时, 将使用硬件施加的实际死区值。

当占空比超过正常限定范围时, 其中一个 PWM 线路的占空比为 0%, 因此无需死区发生。因此, 向死区寄存器 ( $DTRx$  和  $ALTDTRx$ ) 写入较小值不会在功率级中产生直通效应。

### 44.18 节能模式

本章节讨论 MCPWM 模块在休眠模式和空闲模式下的操作。

#### 44.18.1 电机控制 PWM（MCPWM）休眠模式下的操作

当器件进入休眠模式时，系统时钟被禁止。由于 PWM 时基的时钟是基于系统时钟源（TSYSCLK）产生的，所以该时钟也会被禁止，并且在进入休眠模式之前处于有效状态的所有使能 PWM 输出引脚将被冻结在输出状态。如果 MCPWM 模块用于控制电源应用中的负载，在将器件置于休眠模式之前，必须先将 MCPWM 模块输出置于安全状态。根据不同的应用，当 PWM 输出被冻结在特定输出状态下时，负载可能会开始消耗过大的电流。这种情况下，可以使用改写功能将 PWM 输出引脚驱动为无效状态。

如果为 MCPWM 模块配置了故障输入，则在器件进入休眠模式时，故障输入引脚会继续正常工作。如果通过 FLT SRC<3:0> 位与 PWM 发生器关联的故障引脚置为有效，则 PWM 输出将被驱动到在 FLTDAT<1:0> 中保持的设定故障状态。故障输入引脚还能将 CPU 从休眠模式唤醒。如果故障引脚中断优先级大于当前的 CPU 优先级，则当器件被唤醒时，将从故障引脚中断向量地址处开始执行程序。否则，将从 PWSAV 指令后的下一条指令开始继续执行程序。

#### 44.18.2 空闲模式下的 MCPWM 操作

PWM 模块含有一个 PWM 时基空闲模式停止控制位 PTSIDL（PTCON<13>）。PTSIDL 位（PTCON<13>）用于确定当器件进入空闲模式时 PWM 模块是继续工作还是停止工作。如果 PTSIDL = 0，则模块将继续正常工作。如果 PTSIDL = 1，模块将关闭，并停止其内部时钟。在该模式下，系统将无法访问 SFR。这是模块的最低功耗模式。停止空闲模式功能（比如休眠模式和故障引脚）将异步工作。如果 PWM 模块进入空闲状态，PWM 引脚的控制权将归还给与 PWM 引脚关联的 GPIO 位。

建议用户应用程序在进入空闲模式之前先禁止 PWM 输出。如果 PWM 模块正在控制电源转换 / 电机控制应用，将器件置为空闲模式的操作会导致所有控制环被禁止，所以除非明确设计为可在开环模式下工作，否则大多数应用都可能遇到问题。

**注：** 详情请参见第 10 章“节能模式”（DS60001130）。

## 44.19 独立时基的外部控制

如果 XPRES 位 (PWMCONx<1>) 置 1, 外部信号可以复位主专用时基。该工作模式称为电流复位 PWM 模式。如果用户应用程序将 ITB 位 (PWMCONx<9>) 置 1, 则 PWM 发生器工作于独立时基模式。如果用户应用程序将 XPRES 位置 1, 并且 PWM 发生器以主控时基模式工作, 则结果可能不可预测。

CLSRC<3:0> 位 (IOCONx<29:26>) 指定的限流源信号会导致独立时基复位。选定限流信号的有效边沿由 CLPOL 位 (IOCONx<25>) 指定。

在独立时基模式下, 一些 PFC 应用需要将电感电流值维持在高于所需最小电流。这些应用会使用外部复位功能。如果电感电流降低至低于所需值, PWM 周期会被提前终止, 从而可以将 PWM 输出置为有效, 以增大电感电流。PWM 周期会因应用需求而异。此类应用是可变频率 PWM 模式。

## 44.20 应用信息

使用不同 PWM 工作模式和功能的典型应用如下:

- 互补输出模式
- 推挽输出模式
- 多相 PWM
- 可变相位 PWM
- 电流复位 PWM
- 恒定关断时间 PWM
- 限流 PWM

以下几节对每种应用进行了介绍。

44.20.1 互补输出模式

互补 PWM 模式（如图 44-57 所示）通过一种类似于标准边沿对齐模式的方式产生。该模式会在 PWMxL 引脚上提供第二个 PWM 输出信号，该信号是主 PWM 信号（PWMxH）的互补信号。图 44-58 说明了电机控制的互补 PWM 输出模式。

图 44-57： 用于 SMPS 的互补 PWM 输出模式

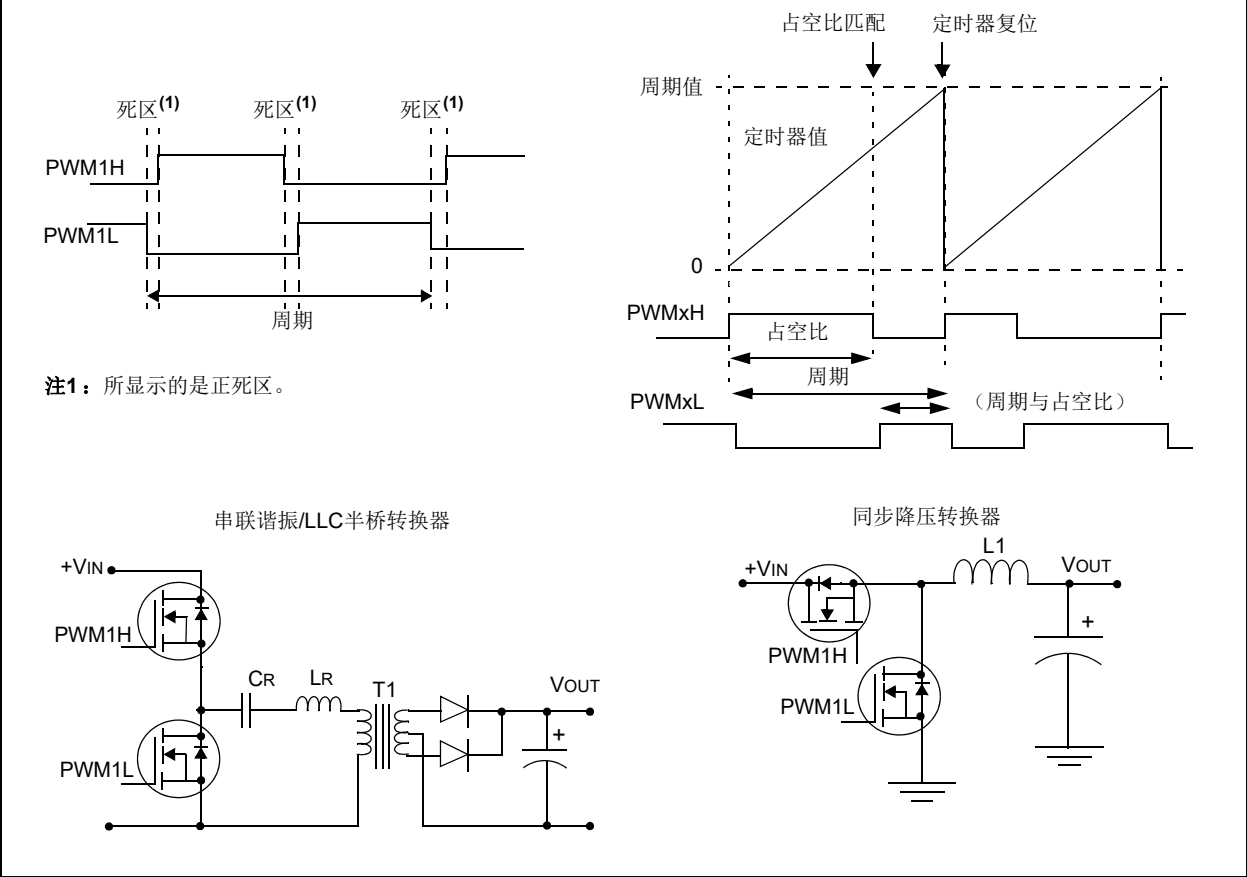
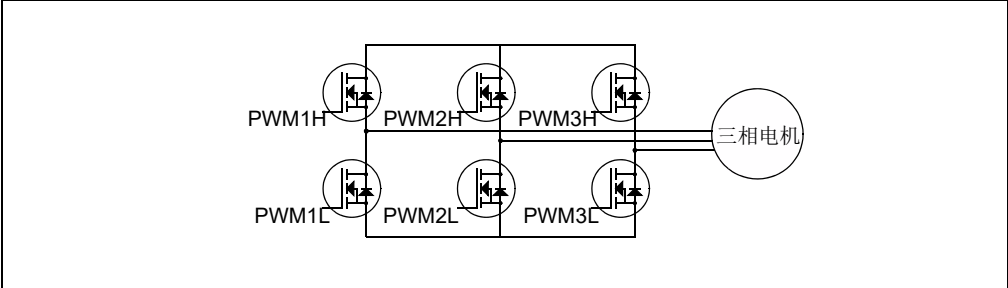


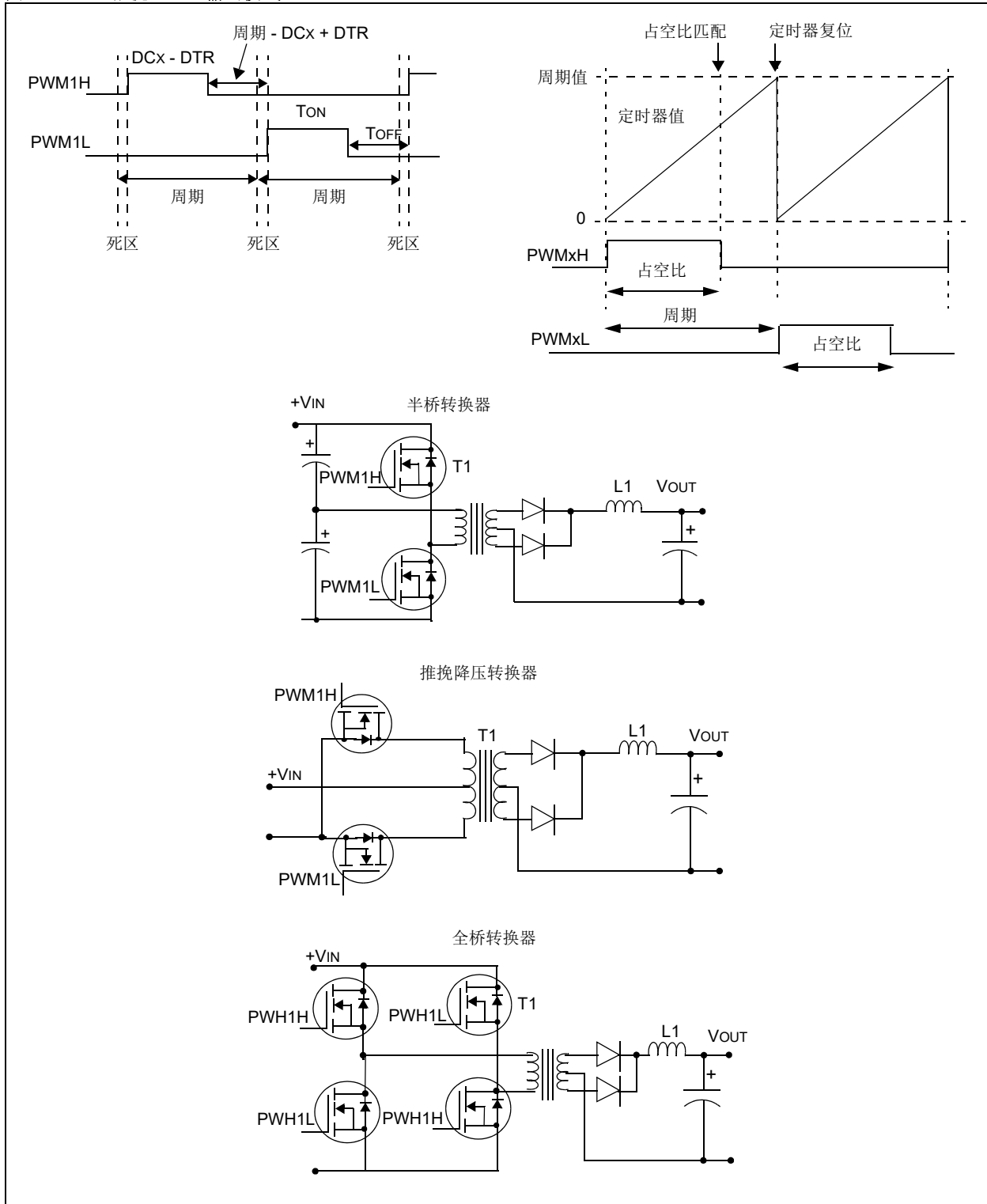
图 44-58： 用于电机控制的互补 PWM 输出模式



## 44.20.2 推挽输出模式

推挽 PWM 输出模式（如图 44-59 所示）会在两个 PWM 引脚之一上交替输出 PWM 信号。在该模式下，互补 PWM 输出不可用。在需要避免可使铁芯饱和的直流电流、基于变压器的电源转换器电路中，该模式很有用。推挽模式可确保两相的占空比相同，从而产生的净直流偏置为 0。

图 44-59： 推挽 PWM 输出模式





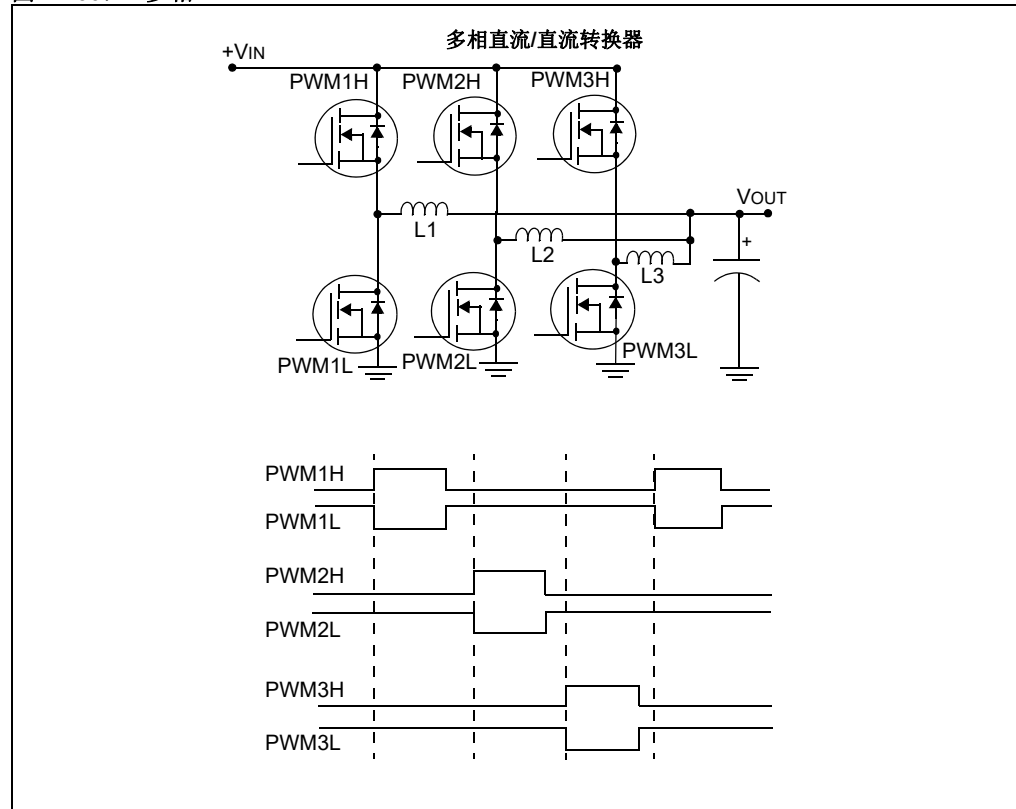
## 44.20.3 多相 PWM

多相 PWM（如图 44-60 所示）使用 PHASEx 寄存器中的相移值来相对于主时基对 PWM 输出进行偏移。由于相移值与主时基相加，所以相移后的输出会比指定零相移的 PWM 信号早出现。在多相模式下，指定的相移通过应用设计固定。在所有使用主控时基的 PWM 模式下，都可以使用相移。

在需要处理快速负载电流瞬变，并且需要满足较小空间要求的直流 / 直流转换器中，经常会使用多相 PWM。多相转换器本质上是一个由降压转换器组成的阵列，这些降压转换器以相互之间稍有相移的方式工作。多个相产生的实际开关速度等于各个转换器的和。

如果单个相以 333 kHz 的 PWM 频率工作，则电路的实际开关频率（如图 44-61 所示）为 1 MHz。这种高开关频率可以降低输入和输出电容尺寸要求。此外，它还可以改善负载瞬态响应和纹波指标。

图 44-60： 多相 PWM



44.20.3.1 交错式功率因数校正（INTERLEAVED POWER FACTOR CORRECTION, IPFC）

在 PFC 电路中交错使用多个升压转换器在近期的应用中变得很流行。图 44-61 和图 44-62 给出了典型的 IPFC 电路配置和工作波形。

图 44-61： 交错式 PFC 图

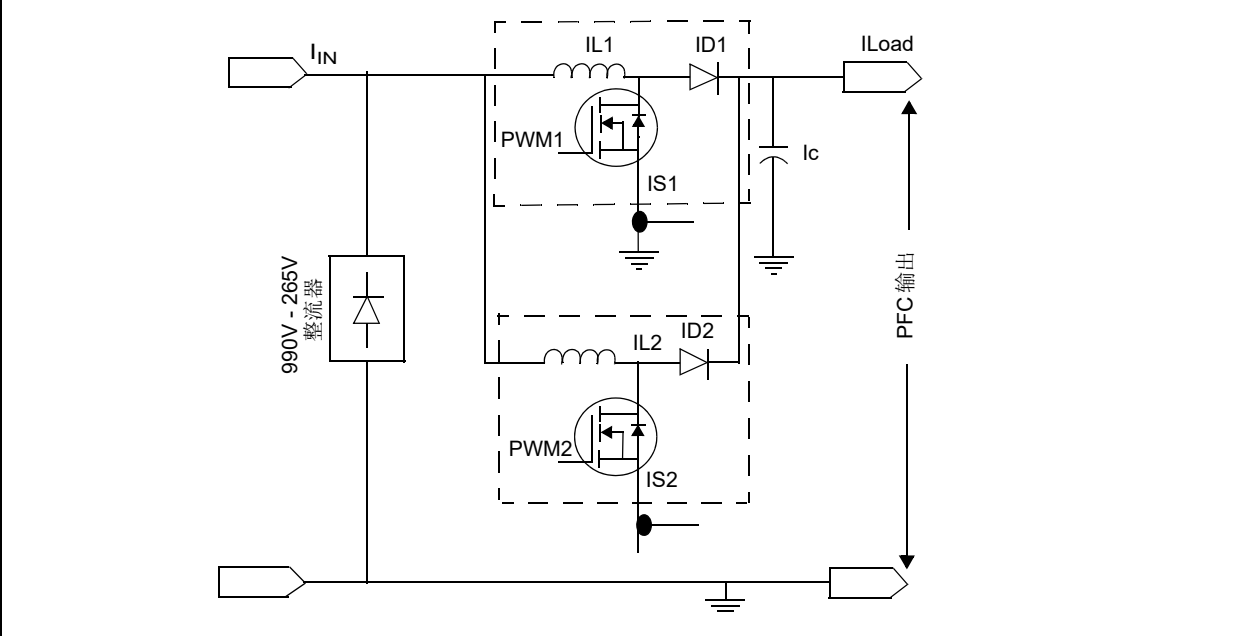
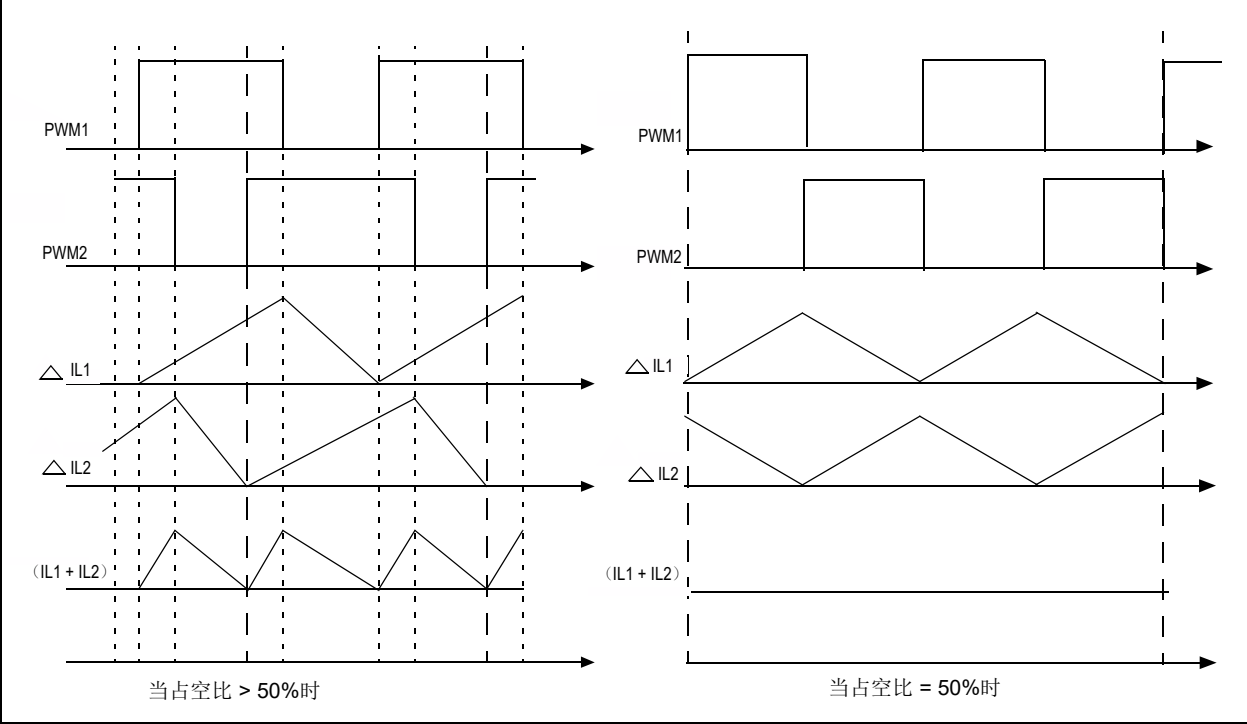


图 44-62： 交错式 PFC 工作波形



多通道 IPFC 按照一致的时间间隔交错地使用通道，由于纹波抵消效应，这可以显著地降低输入纹波电流。输入纹波电流降低意味着可以采用低差模（Differential Mode, DM）噪声滤波器。一般认为，差模噪声幅度降低时，差模滤波器的尺寸也可以减小。作为占空比的函数，输出电容电压纹波也会显著下降。

图 44-63 说明了使用交错式 PFC 的三相电机控制。图 44-64 说明了使用交错式 PFC 的三相电机控制工作波形。

图 44-63： 使用交错式 PFC 的三相电机控制

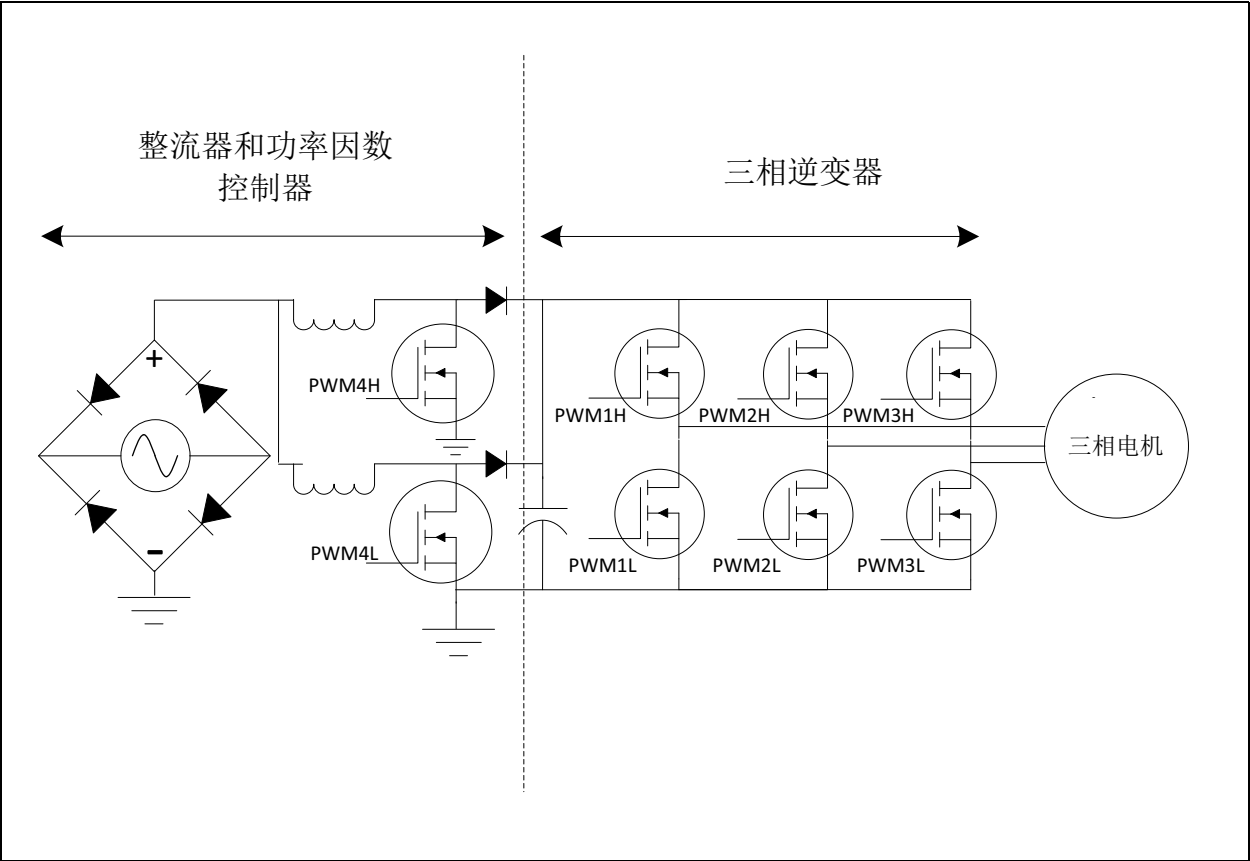
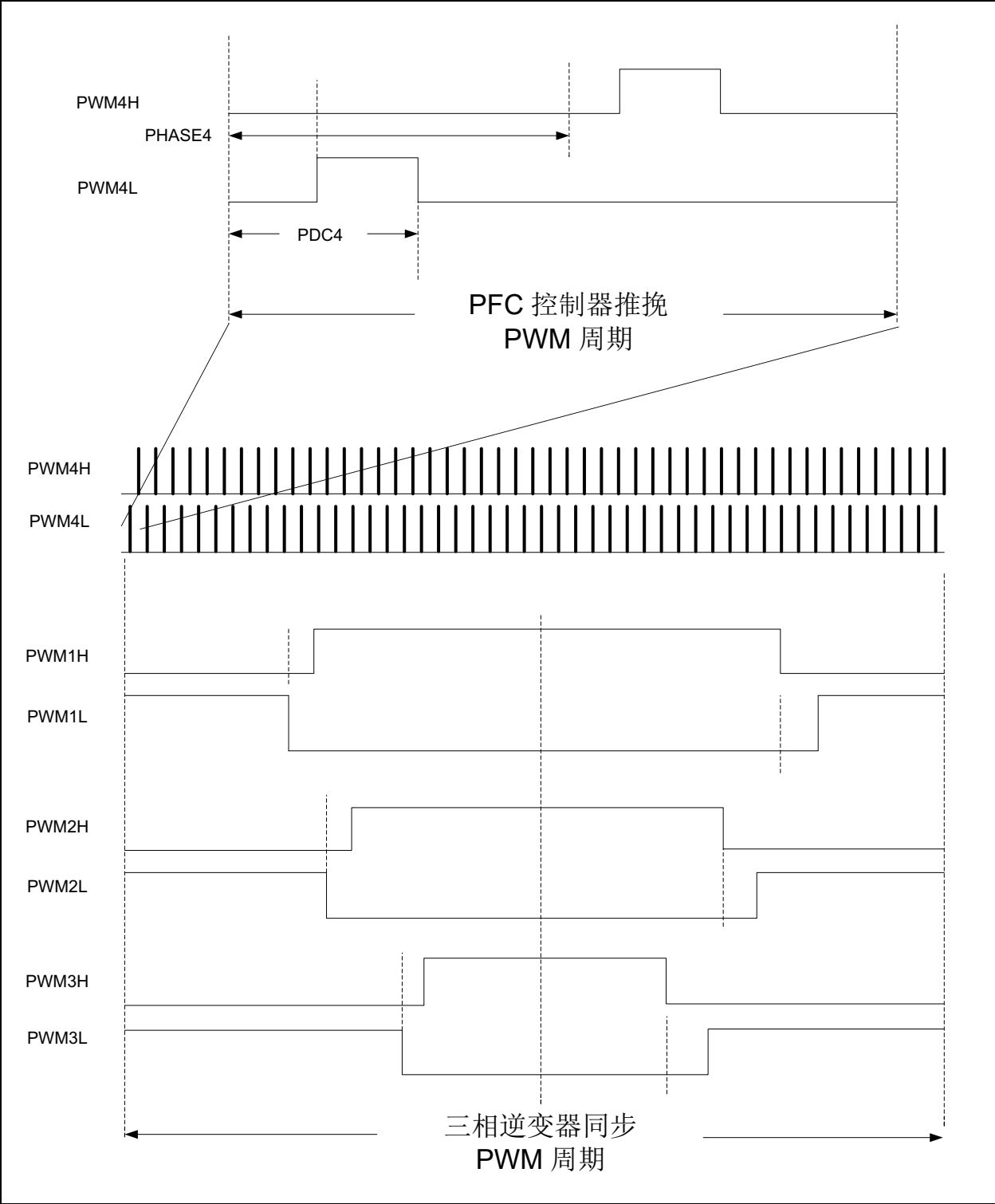


图 44-64： 使用交错式 PFC 的三相电机控制工作波形



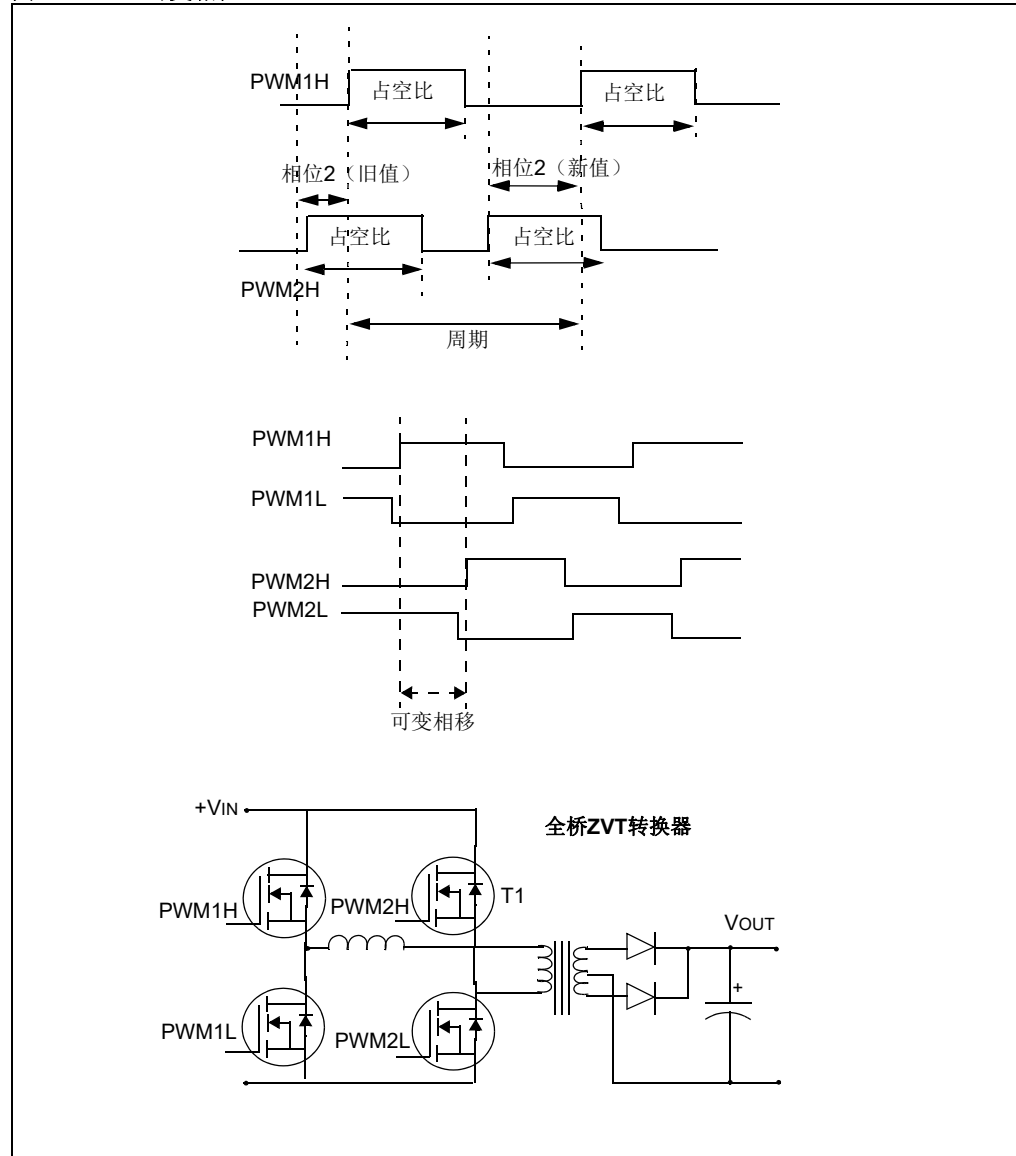
## 44.20.4 可变相位 PWM

可变相位 PWM（如图 44-65 所示）通过不断改变 PWM 通道之间的相移来控制功率流，这一点不同于大多数通过改变 PWM 信号占空比来控制功率流的 PWM 电路。在可变相位应用中，PWM 占空比通常维持在 50%。在所有使用主控时基的 PWM 模式下，都可以使用相移值。

在旨在降低开关损耗的较新型电源转换 / 电机控制拓扑中，使用了可变相位 PWM。在标准 PWM 方法中，当晶体管在导通状态和非导通状态之间切换时（以及反之），在晶体管导通或关断期间晶体管会暴露在全电流和电压条件下，高频时的功率损耗（ $V * I * T_{sw} * F_{PWM}$ ）是可观的。

零电压开关（ZVS）和零电流开关（ZCS）电路拓扑会尝试使用准谐振技术，该技术会对电压或电流波形进行相对偏移，以便在晶体管导通或关断时将电压或电流值变为零。如果电流或电压为零，则不产生开关损耗。

图 44-65： 可变相位 PWM



## 44.20.5 电流复位 PWM

电流复位 PWM（如图 44-66 所示）是一种可变频率模式，在该模式下，实际 PWM 周期小于等于指定的周期值。独立时基在 PWM 信号置为无效之后被外部复位。电流复位 PWM 模式可在恒定 PWM 开启时间模式下使用。要在 PWM 电流复位模式下工作，PWM 发生器应处于独立时基模式。如果未接收到外部复位信号，则默认情况下，PWM 周期会使用 PHASEx 寄存器值。

**注：** 在电流复位模式下，本地时基复位将在 PWMxH/PWMxL 占空比完成之后基于限流输入信号的消息有效电平产生。

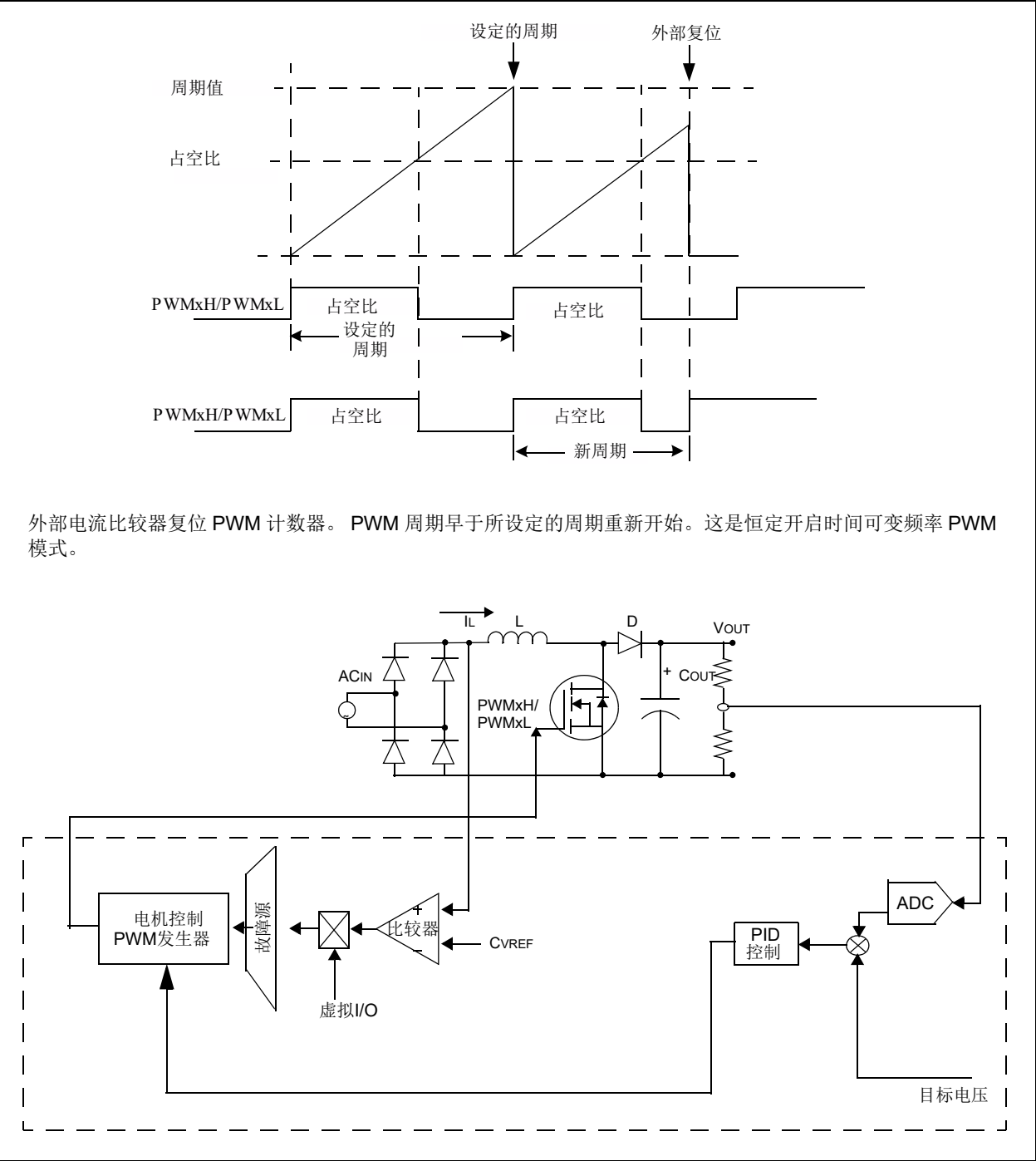
在电流复位模式下，PWM 频率会随负载电流而改变。这一点与大多数 PWM 模式不同，因为用户应用程序会设置最大 PWM 周期，并通过外部电路测量电感电流。当电感电流降至低于指定值时，外部电流比较器电路会产生复位 PWM 时基计数器的信号。用户应用程序会指定一个 PWM 开启时间，然后在 PWM 信号变为无效的一定时间之后，电感电流会降至低于指定值，PWM 计数器会在早于所设定 PWM 周期的时间复位。它被称为恒定开启时间变频 PWM 输出，并在临界导通模式 PFC 应用中使用。

用户不应将它与逐周期限流 PWM 输出混淆，在后者中 PWM 输出置为有效，外部电路产生电流故障，PWM 信号在所设定的占空比正常关断它之前关断。此时，PWM 频率对于给定时基周期是固定的。

电流复位 PWM 模式在 PFC 应用中的优点如下：

- 由于 PFC 升压电感不需要在每个开关周期结束时存储能量，所以可以使用较小的电感。使用较小的电感可以降低成本。
- 升压二极管从导通到关断的换流在电流为零时发生。可以通过使用速度较慢的二极管来降低成本。
- 由于每个周期都接收到反馈，所以内部电流反馈环的速度要快得多

图 44-66: 电流复位 PWM

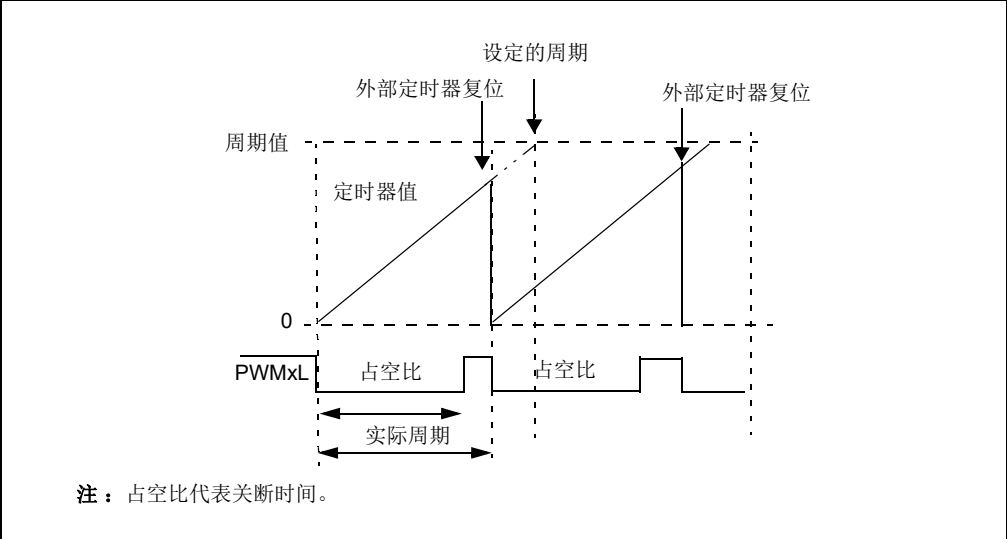


44.20.6 恒定关断时间 PWM

恒定关断时间 PWM（如图 44-67 所示）是一种可变频率 PWM 输出，在该模式下，实际 PWM 周期小于等于指定的周期值。在达到 PWM 信号占空比值，并且 PWM 信号置为无效之后，PWM 时基会被外部复位。它的实现方式是使能称为电流复位 PWM 的开启时间 PWM 输出，并使用互补 PWM 输出（PWMxL）。

只有在 PWM 发生器以独立时基模式工作时，才能使能恒定关断时间 PWM。如果未接收到外部复位信号，则默认情况下，PWM 周期会使用 PHASEx 寄存器中指定的值。

图 44-67： 恒定关断时间 PWM

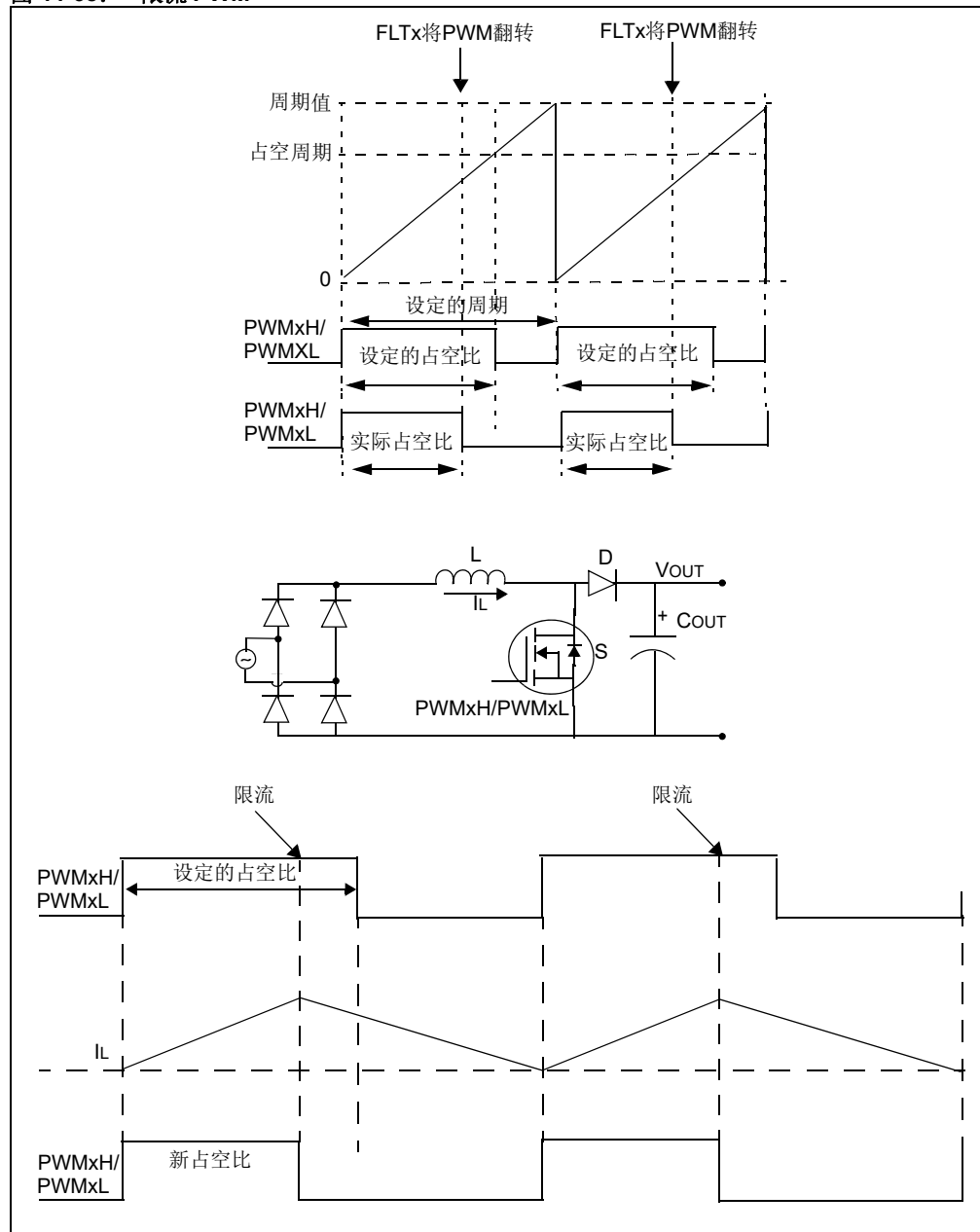




## 44.20.7 限流 PWM

逐周期限流（如图 44-68 所示）会在选定外部故障信号置为有效时截断置为有效的 PWM 信号。PWM 输出值通过 CLDAT<1:0> 位（IOCONx<3:2>）指定。改写输出将保持有效，直到下一个 PWM 周期开始。这有时会在 PFC 电路中使用，在这种电路中，电感电流控制 PWM 开启时间。这是一种恒定频率 PWM。

图 44-68： 限流 PWM



## 44.20.8 断续或突发模式实现

在从转换器汲取的负载电流小于标称电流 / 转换器空载工作的应用中，可以通过使用手动改写强制转换器将 PWM 输出置为无效来降低从电源吸收的功率。通常情况下，可以基于输出电压稳压而将转换器 PWM 输出关闭一段时间，这可以显著降低空载功耗要求。

44.21 相关应用笔记

本节列出了与手册本章内容相关的应用笔记。这些应用笔记可能并不是专为 PIC32 器件系列而编写的，但其概念是相近的，通过适当修改并受到一定限制即可使用。当前与电机控制 PWM（MCPWM）模块相关的应用笔记有：

标题	应用笔记编号
目前没有相关的应用笔记。	N/A

**注：** 如需获取更多 PIC32 系列器件的应用笔记和代码示例，请访问 Microchip 网站（[www.microchip.com](http://www.microchip.com)）。

### 44.22 版本历史

#### 版本 A（2017 年 5 月）

这是本文档的初始版本。

#### 版本 B（2018 年 10 月）

- 添加关于在周期边界产生触发的触发比较寄存器值的注释
- 向TRGCONx寄存器添加注释，以提醒谨慎处理在双周期PWM模式下的双触发产生
- 对时序图图44-31、图44-36和图44-37作了修正
- 对图44-32、图44-33、图44-34、图44-35、图44-36和图44-37作了修正

注:

---

请注意以下有关 **Microchip** 器件代码保护功能的要点：

- **Microchip** 的产品均达到 **Microchip** 数据手册中所述的技术指标。
- **Microchip** 确信：在正常使用的情况下，**Microchip** 系列产品是当今市场上同类产品中最安全的产品之一。
- 目前，仍存在着恶意、甚至是非法破坏代码保护功能的行为。就我们所知，所有这些行为都不是以 **Microchip** 数据手册中规定的操作规范来使用 **Microchip** 产品的。这样做的人极可能侵犯了知识产权。
- **Microchip** 愿与那些注重代码完整性的客户合作。
- **Microchip** 或任何其他半导体厂商均无法保证其代码的安全性。代码保护并不意味着我们保证产品是“牢不可破”的。

代码保护功能处于持续发展中。**Microchip** 承诺将不断改进产品的代码保护功能。任何试图破坏 **Microchip** 代码保护功能的行为均可视为违反了《数字器件千年版权法案（Digital Millennium Copyright Act）》。如果这种行为导致他人在未经授权的情况下，能访问您的软件或其他受版权保护的成果，您有权依据该法案提起诉讼，从而制止这种行为。

---

提供本文档的中文版本仅为了便于理解。请勿忽视文档中包含的英文部分，因为其中提供了有关 **Microchip** 产品性能和使用情况的有用信息。**Microchip Technology Inc.** 及其分公司和相关公司、各级主管与员工及事务代理机构对译文中可能存在的任何差错不承担任何责任。建议参考 **Microchip Technology Inc.** 的英文原版文档。

本出版物中所述的器件应用信息及其他类似内容仅为您提供便利，它们可能由更新之信息所替代。确保应用符合技术规范，是您自身应负的责任。**Microchip** 对这些信息不作任何明示或暗示、书面或口头、法定或其他形式的声明或担保，包括但不限于针对其使用情况、质量、性能、适销性或特定用途的适用性的声明或担保。**Microchip** 对因这些信息及使用这些信息而引起的后果不承担任何责任。如果将 **Microchip** 器件用于生命维持和/或生命安全应用，一切风险由买方自负。买方同意在由此引发任何一切伤害、索赔、诉讼或费用时，会维护和保障 **Microchip** 免于承担法律责任，并加以赔偿。除非另外声明，在 **Microchip** 知识产权保护下，不得暗中以其他方式转让任何许可证。

## 商标

**Microchip** 的名称和徽标组合、**Microchip** 徽标、**Adaptec**、**AnyRate**、**AVR**、**AVR** 徽标、**AVR Freaks**、**BesTime**、**BitCloud**、**chipKIT**、**chipKIT** 徽标、**CryptoMemory**、**CryptoRF**、**dsPIC**、**FlashFlex**、**flexPWR**、**HELDO**、**IGLOO**、**JukeBlox**、**KeeLoq**、**Kleer**、**LANCheck**、**LinkMD**、**maXStylus**、**maXTouch**、**MediaLB**、**megaAVR**、**Microsemi**、**Microsemi** 徽标、**MOST**、**MOST** 徽标、**MPLAB**、**OptoLyzer**、**PackTime**、**PIC**、**picoPower**、**PICSTART**、**PIC32** 徽标、**PolarFire**、**Prochip Designer**、**QTouch**、**SAM-BA**、**SenGenuity**、**SpyNIC**、**SST**、**SST** 徽标、**SuperFlash**、**Symmetricom**、**SyncServer**、**Tachyon**、**TempTrackr**、**TimeSource**、**tinyAVR**、**UNI/O**、**Vectron** 及 **XMEGA** 均为 **Microchip Technology Incorporated** 在美国和其他国家或地区的注册商标。

**APT**、**ClockWorks**、**The Embedded Control Solutions Company**、**EtherSynch**、**FlashTec**、**Hyper Speed Control**、**HyperLight Load**、**IntelliMOS**、**Libero**、**motorBench**、**mTouch**、**Powermite 3**、**Precision Edge**、**ProASIC**、**ProASIC Plus**、**ProASIC Plus** 徽标、**Quiet-Wire**、**SmartFusion**、**SyncWorld**、**Temux**、**TimeCesium**、**TimeHub**、**TimePictra**、**TimeProvider**、**Vite**、**WinPath** 和 **ZL** 均为 **Microchip Technology Incorporated** 在美国的注册商标。

**Adjacent Key Suppression**、**AKS**、**Analog-for-the-Digital Age**、**Any Capacitor**、**AnyIn**、**AnyOut**、**BlueSky**、**BodyCom**、**CodeGuard**、**CryptoAuthentication**、**CryptoAutomotive**、**CryptoCompanion**、**CryptoController**、**dsPICDEM**、**dsPICDEM.net**、**Dynamic Average Matching**、**DAM**、**ECAN**、**EtherGREEN**、**In-Circuit Serial Programming**、**ICSP**、**INICNet**、**Inter-Chip Connectivity**、**JitterBlocker**、**KleerNet**、**KleerNet** 徽标、**memBrain**、**Mindi**、**MiWi**、**MPASM**、**MPF**、**MPLAB Certified** 徽标、**MPLIB**、**MPLINK**、**MultiTRAK**、**NetDetach**、**Omniscient Code Generation**、**PICDEM**、**PICDEM.net**、**PICKit**、**PICtail**、**PowerSmart**、**PureSilicon**、**QMatrix**、**REAL ICE**、**Ripple Blocker**、**SAM-ICE**、**Serial Quad I/O**、**SMART-I.S.**、**SQL**、**SuperSwitcher**、**SuperSwitcher II**、**Total Endurance**、**TSHARC**、**USBCheck**、**VariSense**、**ViewSpan**、**WiperLock**、**Wireless DNA** 和 **ZENA** 均为 **Microchip Technology Incorporated** 在美国和其他国家或地区的商标。

**SQTP** 为 **Microchip Technology Incorporated** 在美国的服务标记。

**Adaptec** 徽标、**Frequency on Demand**、**Silicon Storage Technology** 和 **Symmcom** 均为 **Microchip Technology Inc.** 在除美国外的国家或地区的注册商标。

**GestIC** 为 **Microchip Technology Inc.** 的子公司 **Microchip Technology Germany II GmbH & Co. KG** 在除美国外的国家或地区的注册商标。

在此提及的所有其他商标均为各持有公司所有。

© 2017-2020, **Microchip Technology Inc.** 版权所有。

ISBN: 978-1-5224-5426-7

有关 **Microchip** 质量管理体系的更多信息，请访问 [www.microchip.com/quality](http://www.microchip.com/quality)。

## 全球销售及服务网点

### 美洲

公司总部 **Corporate Office**  
2355 West Chandler Blvd.  
Chandler, AZ 85224-6199  
Tel: 1-480-792-7200  
Fax: 1-480-792-7277

技术支持:  
<http://www.microchip.com/support>

网址: [www.microchip.com](http://www.microchip.com)

**亚特兰大 Atlanta**  
Duluth, GA  
Tel: 1-678-957-9614  
Fax: 1-678-957-1455

**奥斯汀 Austin, TX**  
Tel: 1-512-257-3370

**波士顿 Boston**  
Westborough, MA  
Tel: 1-774-760-0087  
Fax: 1-774-760-0088

**芝加哥 Chicago**  
Itasca, IL  
Tel: 1-630-285-0071  
Fax: 1-630-285-0075

**达拉斯 Dallas**  
Addison, TX  
Tel: 1-972-818-7423  
Fax: 1-972-818-2924

**底特律 Detroit**  
Novi, MI  
Tel: 1-248-848-4000

**休斯敦 Houston, TX**  
Tel: 1-281-894-5983

**印第安纳波利斯 Indianapolis**  
Noblesville, IN  
Tel: 1-317-773-8323  
Fax: 1-317-773-5453  
Tel: 1-317-536-2380

**洛杉矶 Los Angeles**  
Mission Viejo, CA  
Tel: 1-949-462-9523  
Fax: 1-949-462-9608  
Tel: 1-951-273-7800

**罗利 Raleigh, NC**  
Tel: 1-919-844-7510

**纽约 New York, NY**  
Tel: 1-631-435-6000

**圣何塞 San Jose, CA**  
Tel: 1-408-735-9110  
Tel: 1-408-436-4270

**加拿大多伦多 Toronto**  
Tel: 1-905-695-1980  
Fax: 1-905-695-2078

### 亚太地区

**中国 - 北京**  
Tel: 86-10-8569-7000

**中国 - 成都**  
Tel: 86-28-8665-5511

**中国 - 重庆**  
Tel: 86-23-8980-9588

**中国 - 东莞**  
Tel: 86-769-8702-9880

**中国 - 广州**  
Tel: 86-20-8755-8029

**中国 - 杭州**  
Tel: 86-571-8792-8115

**中国 - 南京**  
Tel: 86-25-8473-2460

**中国 - 青岛**  
Tel: 86-532-8502-7355

**中国 - 上海**  
Tel: 86-21-3326-8000

**中国 - 沈阳**  
Tel: 86-24-2334-2829

**中国 - 深圳**  
Tel: 86-755-8864-2200

**中国 - 苏州**  
Tel: 86-186-6233-1526

**中国 - 武汉**  
Tel: 86-27-5980-5300

**中国 - 西安**  
Tel: 86-29-8833-7252

**中国 - 厦门**  
Tel: 86-592-238-8138

**中国 - 香港特别行政区**  
Tel: 852-2943-5100

**中国 - 珠海**  
Tel: 86-756-321-0040

**台湾地区 - 高雄**  
Tel: 886-7-213-7830

**台湾地区 - 台北**  
Tel: 886-2-2508-8600

**台湾地区 - 新竹**  
Tel: 886-3-577-8366

### 亚太地区

**澳大利亚 Australia - Sydney**  
Tel: 61-2-9868-6733

**印度 India - Bangalore**  
Tel: 91-80-3090-4444

**印度 India - New Delhi**  
Tel: 91-11-4160-8631

**印度 India - Pune**  
Tel: 91-20-4121-0141

**日本 Japan - Osaka**  
Tel: 81-6-6152-7160

**日本 Japan - Tokyo**  
Tel: 81-3-6880-3770

**韩国 Korea - Daegu**  
Tel: 82-53-744-4301

**韩国 Korea - Seoul**  
Tel: 82-2-554-7200

**马来西亚 Malaysia - Kuala Lumpur**  
Tel: 60-3-7651-7906

**马来西亚 Malaysia - Penang**  
Tel: 60-4-227-8870

**菲律宾 Philippines - Manila**  
Tel: 63-2-634-9065

**新加坡 Singapore**  
Tel: 65-6334-8870

**泰国 Thailand - Bangkok**  
Tel: 66-2-694-1351

**越南 Vietnam - Ho Chi Minh**  
Tel: 84-28-5448-2100

### 欧洲

**奥地利 Austria - Wels**  
Tel: 43-7242-2244-39  
Fax: 43-7242-2244-393

**丹麦 Denmark - Copenhagen**  
Tel: 45-4450-2828  
Fax: 45-4485-2829

**芬兰 Finland - Espoo**  
Tel: 358-9-4520-820

**法国 France - Paris**  
Tel: 33-1-69-53-63-20  
Fax: 33-1-69-30-90-79

**德国 Germany - Garching**  
Tel: 49-8931-9700

**德国 Germany - Haan**  
Tel: 49-2129-3766400

**德国 Germany - Heilbronn**  
Tel: 49-7131-72400

**德国 Germany - Karlsruhe**  
Tel: 49-721-625370

**德国 Germany - Munich**  
Tel: 49-89-627-144-0  
Fax: 49-89-627-144-44

**德国 Germany - Rosenheim**  
Tel: 49-8031-354-560

**以色列 Israel - Ra'anana**  
Tel: 972-9-744-7705

**意大利 Italy - Milan**  
Tel: 39-0331-742611  
Fax: 39-0331-466781

**意大利 Italy - Padova**  
Tel: 39-049-7625286

**荷兰 Netherlands - Drunen**  
Tel: 31-416-690399  
Fax: 31-416-690340

**挪威 Norway - Trondheim**  
Tel: 47-7288-4388

**波兰 Poland - Warsaw**  
Tel: 48-22-3325737

**罗马尼亚 Romania - Bucharest**  
Tel: 40-21-407-87-50

**西班牙 Spain - Madrid**  
Tel: 34-91-708-08-90  
Fax: 34-91-708-08-91

**瑞典 Sweden - Gothenberg**  
Tel: 46-31-704-60-40

**瑞典 Sweden - Stockholm**  
Tel: 46-8-5090-4654

**英国 UK - Wokingham**  
Tel: 44-118-921-5800  
Fax: 44-118-921-5820