高速 PWM 模块

目录

本章包括下列主题：

1.0 简介 2
2.0 特性 3
3.0 控制寄存器 4
4.0 架构概述 33
5.0 模块说明 36
6.0 PWM 发生器 47
7.0 PWM 触发 61
8.0 PWM 中断 68
9.0 PWM 工作模式 69
10.0 PWM 故障引脚 74
11.0 特殊功能 86
12.0 PWM 输出引脚控制 93
13.0 PWM 占空比立即更新 96
14.0 节能模式 97
15.0 独立时基的外部控制 （电流复位模式） 98
16.0 应用信息 99
17.0 PWM 与其他外设互连 111
18.0 相关应用笔记 114
19.0 版本历史 115

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 dsPIC33/PIC24 器件。

请参见当前器件数据手册中**“高速 PWM”**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。

1.0 简介

本章介绍高速 PWM 模块及其相关的工作模式。高速 PWM 模块支持多种 PWM 模式，是电源转换应用的理想选择。高速 PWM 模块支持的一些常见应用有：

- 交流 / 直流转换器
- 功率因数校正 （Power Factor Correction， PFC）
- 交错式功率因数校正 （Interleaved Power Factor Correction， IPFC）
- 逆变器
- 直流 / 直流转换器
- 电池充电器
- 数字照明
- 不间断电源 （Uninterruptable Power Supply， UPS）
- 交流和直流电机
- 谐振转换器

© 2011-2014 Microchip Technology Inc.

2.0 特性

高速 PWM 模块具有以下主要特性：

- 最多 9 个 PWM 发生器
- 每个 PWM 发生器具有两个 PWM 输出
- 每个 PWM 输出具有独立的时基和占空比
- 占空比、死区、相移和频率分辨率为 1.04 ns
- 所有 PWM 输出具有独立的故障输入和限流输入
- 冗余输出
- 真正独立输出
- 中心对齐 PWM 模式
- 输出改写控制
- 特殊事件触发
- 输入时钟预分频器
- 每个 PWM 周期可向模数转换器 （Analog-to-Digital Converter， ADC）产生双触发
- PWMxL 和 PWMxH 输出引脚交换
- 独立的 PWM 频率、占空比和相移更改
- 前沿消隐 （Leading-Edge Blanking， LEB）功能
- PWM 捕捉功能
- 最多两个主控时基
- 死区补偿
- PWM 斩波
- 支持对故障控制寄存器进行 B 类保护

注：	在中心对齐 PWM 模式下，占空比、死区、相移和频率分辨率为 8.32 ns。

3.0 控制寄存器

本节概述控制高速 PWM 模块操作的每个寄存器的具体功能。

- **PTCON：PWMx 时基控制寄存器**
  - 使能或禁止高速 PWM 模块
  - 设置模数转换器 （ADC）的特殊事件触发，允许或禁止主特殊事件触发中断
  - 使能或禁止立即周期更新
  - 选择主控时基的同步源
  - 指定同步设置
- **PTCON2：PWMx 时钟分频比选择寄存器**
  - 提供所有 PWM 时基的时钟预分频比
- **PTPER：PWMx 主控时基周期寄存器**
  - 提供 PWM 时间周期值
- **SEVTCMP：PWMx 特殊事件触发比较寄存器**
  - 提供用于触发 ADC 模块的比较值，产生主特殊事件触发中断
- **STCON：PWMx 辅助主控时基控制寄存器**
  - 设置 ADC 的辅助特殊事件触发，允许或禁止辅助特殊事件触发中断
  - 使能或禁止辅助主控时基的立即周期更新
  - 选择辅助主控时基的同步源
  - 指定辅助主控时基的同步设置
- **STCON2：PWMx 辅助时钟分频比选择寄存器**
  - 提供 PWM 辅助主控时基的时钟预分频比
- **STPER：PWMx 辅助主控时基周期寄存器**
  - 提供辅助主控时基的 PWM 时间周期值
- **SSEVTCMP：PWMx 辅助特殊事件比较寄存器**
  - 提供用于触发 ADC 模块的辅助主控时基的比较值，产生辅助特殊事件触发中断
- **CHOP：PWMx 斩波时钟发生器寄存器**
  - 使能和禁止用于调制 PWM 输出的斩波信号
  - 指定斩波信号的周期
- **MDC：PWMx 主控占空比寄存器**
  - 提供 PWM 主控占空比值
- **PWMCONx：PWMx 控制寄存器**
  - 允许或禁止故障中断、限流中断和主触发中断
  - 提供故障中断、限流中断和主触发中断的中断状态
  - 选择时基类型 （主控时基或独立时基）
  - 选择占空比类型 （主控占空比或独立占空比）
  - 控制死区模式
  - 使能或禁止中心对齐模式
  - 控制外部 PWM 复位操作
  - 使能或禁止立即更新占空比、相位偏移和独立时基周期
- **PDCx：PWMx 发生器占空比寄存器**
  - 在选择主控时基的情况下提供 PWMxH 和 PWMxL 输出的占空比值
  - 在选择独立时基的情况下提供 PWMxH 输出的占空比值

© 2011-2014 Microchip Technology Inc.

- **PHASEx：PWMx 主相移寄存器**
  - 在选择主控时基的情况下提供 PWMxH 和 / 或 PWMxL 输出的相移值
  - 在选择独立时基的情况下提供 PWMxH 和 / 或 PWMxL 输出的独立时基周期
- **DTRx：PWMx 死区寄存器**
  - 在选择正死区的情况下提供 PWMxH 输出的死区值
  - 在选择负死区的情况下提供 PWMxL 输出的死区值
- **ALTDTRx：PWMx 备用死区寄存器**
  - 在选择正死区的情况下提供 PWMxL 输出的死区值
  - 在选择负死区的情况下提供 PWMxH 输出的死区值
- **SDCx：PWMx 辅助占空比寄存器**
  - 在选择独立时基的情况下提供 PWMxL 输出的占空比值
- **SPHASEx：PWMx 辅助相移寄存器**
  - 在选择主控时基和独立输出模式的情况下提供 PWMxL 输出的相移
  - 在选择独立时基和独立输出模式的情况下提供 PWMxL 输出的独立时基周期值
- **TRGCONx：PWMx 触发控制寄存器**
  - 使能 PWMx 触发后分频比开始事件
  - 指定在产生第一次触发之前跳过的 PWM 周期数
  - 使能或禁止主 PWM 触发事件与辅助 PWM 触发事件组合
- **IOCONx：PWMx I/O 控制寄存器**
  - 使能或禁止 PWM 引脚控制功能 （PWM 控制或 GPIO）
  - 控制 PWMxH 和 PWMxL 输出极性
  - 在选择以下任意模式的情况下控制 PWMxH 和 PWMxL 输出：
    - 互补模式
    - 推挽模式
    - 真正独立输出模式
- **FCLCONx：PWMx 故障限流控制寄存器**
  - 选择限流控制信号源
  - 选择限流极性
  - 使能或禁止限流模式
  - 选择故障控制信号源
  - 配置故障极性
  - 使能或禁止故障模式
- **TRIGx：PWMx 主触发比较值寄存器**
  - 提供用于产生主 PWM 触发的比较值
- **STRIGx：PWMx 辅助触发比较值寄存器**
  - 提供用于产生辅助 PWM 触发的比较值
- **LEBCONx：PWMx 前沿消隐控制寄存器 （版本 1）**
  - 选择对 PWM 输出的上升沿或下降沿进行前沿消隐
  - 使能或禁止故障输入和限流输入的前沿消隐
- **LEBCONx：PWMx 前沿消隐控制寄存器 （版本 2）**
  - 选择对 PWM 输出的上升沿或下降沿进行前沿消隐 （LEB）
  - 使能或禁止故障输入和限流输入的前沿消隐
  - 指定在选定消隐信号 （PWMxH、 PWMxL 或 PWMx 附属控制寄存器中的 PWM 状态消隐源选择位 BLANKSEL<3:0>（AUXCONx<11:8>）指定的其他信号）为高电平或低电平时，故障输入和限流信号的消隐状态

- **LEBDLYx：PWMx 前沿消隐延时寄存器**
  - 指定选定故障输入和限流信号的消隐时间
- **AUXCONx：PWMx 附属控制寄存器**
  - 使能或禁止高分辨率 PWM 周期和占空比，以降低系统功耗
  - 选择限流信号和故障输入的状态消隐信号
- **PWMCAPx：PWMx 主时基捕捉寄存器**
  - 提供在限流输入上检测到前沿时捕捉的独立时基值
- **PWMKEY：PWMx 保护锁定 / 解锁密钥寄存器**
  - 使能 PWMx 故障控制寄存器 IOCONx 和 FCLCONx 的写保护，用于提供 B 类故障保护

© 2011-2014 Microchip Technology Inc.

寄存器 3-1：　PTCON：PWMx 时基控制寄存器

R/W-0	U-0	R/W-0	HS/HC-0	R/W-0	R/W-0	R/W-0	R/W-0
PTEN(3)	—	PTSIDL	SESTAT	SEIEN	EIPU(1)	SYNCPOL(1,2)	SYNCOEN(1,2)
bit 15							bit 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
SYNCEN(1,2)	SYNCSRC2(1,2)	SYNCSRC1(1,2)	SYNCSRC0(1,2)	SEVTPS3(1)	SEVTPS2(1)	SEVTPS1(1)	SEVTPS0(1)
bit 7							bit 0

图注：	HC = 硬件清零位	HS = 硬件置 1 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 15　**PTEN：**PWM 模块使能位 (3)
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：**特殊事件触发中断状态位
1 = 特殊事件触发中断处于待处理状态
0 = 特殊事件触发中断不处于待处理状态
该位通过设置 SEIEN = 0 进行清零。

bit 11　**SEIEN：**特殊事件触发中断允许位
1 = 允许特殊事件触发中断
0 = 禁止特殊事件触发中断

bit 10　**EIPU：**使能立即周期更新位 (1)
1 = 立即更新有效周期寄存器
0 = 在 PWM 周期边界处更新有效周期寄存器

bit 9　**SYNCPOL：**同步输入和输出极性位 (1,2)
1 = SYNCIx/SYNCO 极性反相 （低电平有效）
0 = SYNCIx/SYNCO 为高电平有效

bit 8　**SYNCOEN：**主时基同步使能位 (1,2)
1 = 使能 SYNCO 输出
0 = 禁止 SYNCO 输出

bit 7　**SYNCEN：**外部时基同步使能位 (1,2)
1 = 使能主时基的外部同步
0 = 禁止主时基的外部同步

bit 6-4　**SYNCSRC<2:0>：**同步源选择位 (1,2)
011 = SYNCI4
010 = SYNCI3
001 = SYNCI2
000 = SYNCI1

注　**1：**　只有 PTEN = 0 时，才能更改这些位。
2：　PWM 时基同步只能在无相移的主控时基中使用。
3：　通过设置 PTCON<15> = 1 来使能 PWM 模块时，可以在 PWM 输出开始开关之前观察到一定的延时。该延时等于：

PWM 开启延时 = (2/ACLK) + (3 • (PCLKDIV<2:0> 设置)/ACLK) + 15 ns

寄存器 3-1：　PTCON：PWMx 时基控制寄存器 （续）

bit 3-0　**SEVTPS<3:0>：**PWM 特殊事件触发信号输出后分频比选择位 (1)

1111 = 1:16，后分频器在每发生十六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： 只有 PTEN = 0 时，才能更改这些位。

2： PWM 时基同步只能在无相移的主控时基中使用。

3： 通过设置 PTCON<15> = 1 来使能 PWM 模块时，可以在 PWM 输出开始开关之前观察到一定的延时。该延时等于：

PWM 开启延时 = (2/ACLK) + (3 • (PCLKDIV<2:0> 设置)/ACLK) + 15 ns

© 2011-2014 Microchip Technology Inc.

寄存器 3-2：　PTCON2：PWMx 时钟分频比选择寄存器

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	PCLKDIV2(1,2)	PCLKDIV1(1,2)	PCLKDIV0(1,2)
bit 7							bit 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 15-3　**未实现：**读为 0

bit 2-0　**PCLKDIV<2:0>：**PWM 输入时钟预分频比选择位 (1,2)

111 = 保留
110 = 64 分频，最大 PWM 时序分辨率
101 = 32 分频，最大 PWM 时序分辨率
100 = 16 分频，最大 PWM 时序分辨率
011 = 8 分频，最大 PWM 时序分辨率
010 = 4 分频，最大 PWM 时序分辨率
001 = 2 分频，最大 PWM 时序分辨率
000 = 1 分频，最大 PWM 时序分辨率 （上电默认值）

注　1：只有 PTEN = 0 时，才能更改这些位。在工作期间改变时钟选择会产生不可预测的结果。

2：PWM 输入时钟预分频比将会影响 PWM 模块的所有时序参数，包括周期、占空比、相移、死区、触发、前沿消隐和 PWM 捕捉。

寄存器 3-3：　PTPER：PWMx 主控时基周期寄存器

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
PTPER<15:8>(1,2)							
bit 15							bit 8

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-0	R/W-0	R/W-0
PTPER<7:0>(1,2)							
bit 7							bit 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 15-0　**PTPER<15:0>：**PWM 主控时基 （PMTMR）周期值位 (1,2)

注　1： PWM 时基的最小值为 0x0010，最大值为 0xFFF8。

2： 当周期值小于 0x0028 时，低 3 位必须设置为 0。对于这些极短暂的 PWM 周期脉冲，产生的周期分辨率为 8.32 ns （处于最快的附属时钟速率下）。

寄存器 3-4：　SEVTCMP：PWMx 特殊事件触发比较寄存器

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<12:5>(1,2,3)							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0
SEVTCMP<4:0>(1,2,3)					—	—	—
bit 7							bit 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 15-3　**SEVTCMP<12:0>：**主特殊事件触发比较计数值位 (1,2,3)

bit 2-0　**未实现：**读为 0

注　1： 1 LSb = 1.04 ns；因此，在最快的 PWM 时钟分频比设置 （PTCON2<2:0> = 000）下，最小 SEVTCMP 分辨率为 8.32 ns。

2： 与 PWM 主控时基计数器 （PMTMR）发生比较匹配时，将会产生特殊事件触发信号。

3： 这些位 （SEVTCMP<12:0>）与 PTCON<3:0> 位域配合使用。

© 2011-2014 Microchip Technology Inc.

寄存器 3-5：　STCON：PWMx 辅助主控时基控制寄存器

U-0	U-0	U-0	HS/HC-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	SESTAT	SEIEN	EIPU(1)	SYNCPOL(1,2)	SYNCOEN(1,2)
bit 15							bit 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
SYNCEN(1,2)	SYNCSRC2(1)	SYNCSRC1(1)	SYNCSRC0(1)	SEVTPS3(1)	SEVTPS2(1)	SEVTPS1(1)	SEVTPS0(1)
bit 7							bit 0

图注：	HC = 硬件清零位	HS = 硬件置 1 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 15-13　**未实现：**读为 0

bit 12　**SESTAT：**特殊事件触发中断状态位
1 = 辅助特殊事件触发中断处于待处理状态
0 = 辅助特殊事件触发中断不处于待处理状态
该位通过设置 SEIEN = 0 进行清零。

bit 11　**SEIEN：**特殊事件触发中断允许位
1 = 允许辅助特殊事件触发中断
0 = 禁止辅助特殊事件触发中断

bit 10　**EIPU：**使能立即周期更新位 (1)
1 = 立即更新有效辅助周期寄存器
0 = 在 PWM 周期边界处更新有效辅助周期寄存器

bit 9　**SYNCPOL：**同步输入和输出极性位 (1,2)
1 = SYNCEN 的下降沿会复位 SMTMR； SYNCO2 输出为低电平有效
0 = SYNCEN 的上升沿会复位 SMTMR； SYNCO2 输出为高电平有效

bit 8　**SYNCOEN：**辅助主控时基同步使能位 (1,2)
1 = 使能 SYNCO2 输出
0 = 禁止 SYNCO2 输出

bit 7　**SYNCEN：**外部辅助主控时基同步使能位 (1,2)
1 = 使能辅助时基的外部同步
0 = 禁止辅助时基的外部同步

bit 6-4　**SYNCSRC<2:0>：**辅助时基同步源选择位 (1)
011 = SYNCI4
010 = SYNCI3
001 = SYNCI2
000 = SYNCI1

bit 3-0　**SEVTPS<3:0>：**PWM 辅助特殊事件触发信号输出后分频比选择位 (1)
1111 = 1:16 后分频比
•
•
•
0001 = 1:2 后分频比
0000 = 1:1 后分频比

注　1：只有 PTEN = 0 时，才能更改这些位。
2：PWM 时基同步只能在无相移的主控时基中使用。

寄存器 3-6：　STCON2：PWMx 辅助时钟分频比选择寄存器

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	PCLKDIV2(1,2)	PCLKDIV1(1,2)	PCLKDIV0(1,2)
bit 7							bit 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 15-3　**未实现：**读为 0

bit 2-0　**PCLKDIV<2:0>：**PWM 输入时钟预分频比选择位 (1,2)

111 = 保留
110 = 64 分频，最大 PWM 时序分辨率
101 = 32 分频，最大 PWM 时序分辨率
100 = 16 分频，最大 PWM 时序分辨率
011 = 8 分频，最大 PWM 时序分辨率
010 = 4 分频，最大 PWM 时序分辨率
001 = 2 分频，最大 PWM 时序分辨率
000 = 1 分频，最大 PWM 时序分辨率 （上电默认值）

注　1： 只有 PTEN = 0 时，才能更改这些位。在工作期间改变时钟选择会产生不可预测的结果。

2： PWM 输入时钟预分频比将会影响 PWM 模块的所有时序参数，包括周期、占空比、相移、死区、触发、前沿消隐和 PWM 捕捉。

© 2011-2014 Microchip Technology Inc.

寄存器 3-7：　STPER：PWMx 辅助主控时基周期寄存器

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
STPER<15:8>(1,2)							
bit 15							bit 8

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-0	R/W-0	R/W-0
STPER<7:0>(1,2)							
bit 7							bit 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 15-0　**STPER<15:0>：**辅助主控时基 （SMTMR）周期值位 (1,2)

注　1： PWM 时基的最小值为 0x0010，最大值为 0xFFF8。

2： 当周期值小于 0x0028 时，低 3 位必须设置为 0。对于这些极短暂的 PWM 周期脉冲，产生的周期分辨率为 8.32 ns （处于最快的附属时钟速率下）。

寄存器 3-8：　SSEVTCMP：PWMx 辅助特殊事件比较寄存器

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<12:5>(1,2,3)							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0
SSEVTCMP<4:0>(1,2,3)					—	—	—
bit 7							bit 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 15-3　**SSEVTCMP<12:0>：**PWM 辅助特殊事件比较计数值位 (1,2,3)

bit 2-0　**未实现：**读为 0

注　1： 1 LSb = 1.04 ns ；因此，在最快的 PWM 时钟分频比设置 （STCON2<2:0> = 000）下，最小 SSEVTCMP 分辨率为 8.32 ns。

2： 与 PWM 辅助主控时基计数器 （SMTMR）发生比较匹配时，将会产生辅助特殊事件触发信号。

3： 这些位 （SSEVTCMP<12:0>）与 STCON<3:0> 位域配合使用。

© 2011-2014 Microchip Technology Inc.

寄存器 3-9：　CHOP：PWMx 斩波时钟发生器寄存器

R/W-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
CHPCLKEN	—	—	—	—	—	CHOPCLK6	CHOPCLK5
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0
CHOPCLK4	CHOPCLK3	CHOPCLK2	CHOPCLK1	CHOPCLK0	—	—	—
bit 7							bit 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 15　**CHPCLKEN：**使能斩波时钟发生器位
1 = 使能斩波时钟发生器
0 = 禁止斩波时钟发生器

bit 14-10　**未实现：**读为 0

bit 9-3　**CHOPCLK<6:0>：**斩波时钟分频比位
值以 8.32 ns 为单位进行递增。斩波时钟信号频率的计算方法如下：
斩波频率 = 1/(16.64 * (CHOP<6:0> + 1) * 主主控 PWM 输入时钟 /PCLKDIV<2:0>)

bit 2-0　**未实现：**读为 0

寄存器 3-10：　MDC：PWMx 主控占空比寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
MDC<15:8>(1,2,3)							
bit 15							bit 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
MDC<7:0>(1,2,3)							
bit 7							bit 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 15-0　**MDC<15:0>：**主控 PWM 占空比值位 (1,2,3)

注　1：在 PWM 输出上可以产生的最小脉冲宽度对应于值 0x0008，而产生的最大脉冲宽度对应于 （周期 + 0x0008）的值。

2：MDC<15:0> < 0x0008 时，产生的占空比为 0%。MDC<15:0> > 周期 + 0x0008 时，产生的占空比为 100%。

3：当占空比接近 PWM 周期的 0% 或 100% （0 ns 至 40 ns，取决于工作模式）时， PWM 占空比分辨率将从 1 LSb 降至 3 LSb。

© 2011-2014 Microchip Technology Inc.

寄存器 3-11：　PWMCONx：PWMx 控制寄存器

HS/HC-0	HS/HC-0	HS/HC-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
FLTSTAT(1)	CLSTAT(1)	TRGSTAT	FLTIEN	CLIEN	TRGIEN	ITB(3)	MDCS(3)
bit 15							bit 8

R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
DTC1(3)	DTC0(3)	DTCP(3,6)	—	MTBS	CAM(2,3,5)	XPRES(4)	IUE
bit 7							bit 0

图注：	HC = 硬件清零位	HS = 硬件置 1 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 15　**FLTSTAT：**故障中断状态位 (1)
1 = 故障中断处于待处理状态
0 = 故障中断不处于待处理状态
该位通过设置 FLTIEN = 0 进行清零。

bit 14　**CLSTAT：**限流中断状态位 (1)
1 = 限流中断处于待处理状态
0 = 限流中断不处于待处理状态
该位通过设置 CLIEN = 0 进行清零。

bit 13　**TRGSTAT：**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
该位通过设置 TRGIEN = 0 进行清零。

bit 12　**FLTIEN：**故障中断允许位
1 = 允许故障中断
0 = 禁止故障中断，并且 FLTSTAT 位清零

bit 11　**CLIEN：**限流中断允许位
1 = 允许限流中断
0 = 禁止限流中断，并且 CLSTAT 位清零

bit 10　**TRGIEN：**触发中断允许位
1 = 触发事件产生中断请求
0 = 禁止触发事件中断，并且 TRGSTAT 位清零

bit 9　**ITB：**独立时基模式位 (3)
1 = PHASEx/SPHASEx 寄存器为该 PWM 发生器提供时基周期
0 = PTPER/STPER 为该 PWM 发生器提供时序

bit 8　**MDCS：**主控占空比寄存器选择位 (3)
1 = MDC 寄存器为该 PWM 发生器提供占空比信息
0 = PDCx 和 SDCx 寄存器为该 PWM 发生器提供占空比信息

注　1： 必须用软件清零相应的中断状态，以及中断控制器中对应的 IFSx 位。
2： 只有使能独立时基模式 （ITB = 1）时，才能使用中心对齐模式。如果 ITB = 0，则 CAM 位会被忽略。
3： 使能 PWM （PTEN = 1）之后，不应更改这些位。
4： 配置 FCLCONx<8> = 0 且 PWMCONx<9> = 1，以工作于外部周期复位模式。
5： 中心对齐模式会忽略占空比、相位和死区寄存器的低 3 位。在时钟分频比设置为最快时钟时，可用的最高 CAM 分辨率为 8.32 ns。
6： 要使 DTCP 生效，必须设置 DTC<1:0> = 11；否则， DTCP 位会被忽略。

寄存器 3-11： PWMCONx：PWMx 控制寄存器 （续）

bit 7-6 **DTC<1:0>：**死区控制位 (3)
11 = 死区补偿模式
10 = 禁止死区功能
01 = 对于所有输出模式施加负死区
00 = 对于所有输出模式施加正死区

bit 5 **DTCP：**死区补偿极性位 (3,6)
当设置为 1 时：
如果 DTCMPx = 0，则缩短 PWMxL，延长 PWMxH。
如果 DTCMPx = 1，则缩短 PWMxH，延长 PWMxL。
当设置为 0 时：
如果 DTCMPx = 0，则缩短 PWMxH，延长 PWMxL。
如果 DTCMPx = 1，则缩短 PWMxL，延长 PWMxH。

bit 4 **未实现：**读为 0

bit 3 **MTBS：**主控时基选择位
1 = PWM 发生器使用辅助主控时基进行同步，并使用它作为 PWM 发生逻辑的时钟源（如果辅助时基可用）
0 = PWM 发生器使用主主控时基进行同步，并使用它作为 PWM 发生逻辑的时钟源

bit 2 **CAM：**中心对齐模式使能位 (2,3,5)
1 = 使能中心对齐模式
0 = 使能边沿对齐模式

bit 1 **XPRES：**外部 PWM 复位控制位 (4)
1 = 如果 PWM 发生器处于独立时基模式，则限流源复位该 PWM 发生器的时基
0 = 外部引脚不影响 PWM 时基

bit 0 **IUE：**立即更新使能位
1 = 立即更新有效 MDC/PDCx/SDCx/PHASEx/SPHASEx 寄存器
0 = 更新有效 MDC/PDCx/SDCx/PHASEx/SPHASEx 寄存器与本地 PWM 时基同步。

注 1： 必须用软件清零相应的中断状态，以及中断控制器中对应的 IFSx 位。
2： 只有使能独立时基模式 （ITB = 1）时，才能使用中心对齐模式。如果 ITB = 0，则 CAM 位会被忽略。
3： 使能 PWM （PTEN = 1）之后，不应更改这些位。
4： 配置 FCLCONx<8> = 0 且 PWMCONx<9> = 1，以工作于外部周期复位模式。
5： 中心对齐模式会忽略占空比、相位和死区寄存器的低 3 位。在时钟分频比设置为最快时钟时，可用的最高 CAM 分辨率为 8.32 ns。
6： 要使 DTCP 生效，必须设置 DTC<1:0> = 11；否则， DTCP 位会被忽略。

© 2011-2014 Microchip Technology Inc.

寄存器 3-12： PDCx：PWMx 发生器占空比寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PDCx<15:8>(1,2,3,4)							
bit 15							bit 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
PDCx<7:0>(1,2,3,4)							
bit 7							bit 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 15-0 **PDCx<15:0>：**PWMx 发生器占空比值位 (1,2,3,4)

注 1： 在独立输出模式下， PDCx 位仅控制 PWMxH 占空比。在互补、冗余和推挽 PWM 模式下， PDCx 位控制 PWMxH 和 PWMxL 的占空比。

2： 在 PWM 输出上可以产生的最小脉冲宽度对应于值 0x0008，而产生的最大脉冲宽度对应于 （周期 + 0x0008） 的值。

3： PDC<15:0> < 0x0008 时，产生的占空比为 0%。PDC<15:0> > 周期 + 0x0008 时，产生的占空比为 100%。

4： 当占空比接近 PWM 周期的 0% 或 100%（0 ns 至 40 ns，取决于工作模式）时， PWM 占空比分辨率将从 1 LSb 降至 3 LSb。

寄存器 3-13： SDCx：PWMx 辅助占空比寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SDCx<15:8>(1,2,3,4)							
bit 15							bit 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
SDCx<7:0>(1,2,3,4)							
bit 7							bit 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 15-0 **SDCx<15:0>：**PWMxL 输出引脚的辅助占空比位 (1,2,3,4)

注 1： SDCx 位仅在独立输出模式下使用。在独立输出模式下使用时， SDCx 位控制 PWMxL 占空比。在其他 PWM 模式下，这些位将被忽略。

2： 在 PWM 输出上可以产生的最小脉冲宽度对应于值 0x0008，而产生的最大脉冲宽度对应于 （周期 + 0x0008） 的值。

3： SDC<15:0> < 0x0008 时，产生的占空比为 0%。SDC<15:0> > 周期 + 0x0008 时，产生的占空比为 100%。

4： 当占空比接近 PWM 周期的 0% 或 100% （0 ns 至 40 ns，取决于工作模式）时， PWM 占空比分辨率将从 1 LSb 降至 3 LSb。

© 2011-2014 Microchip Technology Inc.

寄存器 3-14： PHASEx：PWMx 主相移寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PHASEx<15:8>(1,2)							
bit 15							bit 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
PHASEx<7:0>(1,2)							
bit 7							bit 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 15-0 **PHASEx<15:0>：** PWM 发生器的 PWM 相移值或独立时基周期位 (1,2)

注 1： 如果 PWMCONx<9> = 0 （主控时基模式），则依工作模式适用以下情况：

- 互补、冗余和推挽 PWM 输出模式 （IOCONx<11:10> = 00、 01 或 10）， PHASEx<15:0> = PWMxH 和 PWMxL 输出的相移值。
- 真正独立 PWM 输出模式 （IOCONx<11:10> = 11）， PHASEx<15:0> = 仅 PWMxH 的相移值。
- 当 PHASEx/SPHASEx 位提供对应于主控时基的相移时，值的有效范围为 0x0000 - 周期。

2： 如果 PWMCONx<9> = 1 （独立时基模式），则依工作模式适用以下情况：

- 互补、冗余和推挽 PWM 输出模式 （IOCONx<11:10> = 00、 01 或 10）， PHASEx<15:0> = PWMxH 和 PWMxL 输出的独立时基周期值。
- 真正独立 PWM 输出模式 （IOCONx<11:10> = 11）， PHASEx<15:0> = 仅 PWMxH 的独立时基周期值。
- 当 PHASEx/SPHASEx 位提供本地周期时，值的有效范围为 0x0010 - 0xFFF8。

寄存器 3-15： SPHASEx：PWMx 辅助相移寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SPHASEx<15:8>(1,2)							
bit 15							bit 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
SPHASEx<7:0>(1,2)							
bit 7							bit 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 15-0 **SPHASEx<15:0>**：PWMxL 输出引脚辅助相位偏移位 (1,2)
（仅在独立 PWM 模式下使用）

注 1： 如果 PWMCONx<9> = 0，则依工作模式适用以下情况：
- 互补、冗余和推挽 PWM 输出模式 （IOCONx<11:10> = 00、 01 或 10）， SPHASEx<15:0> = 不使用。
- 真正独立 PWM 输出模式 （IOCONx<11:10> = 11）， SPHASEx<15:0> = 仅 PWMxL 的相移值。
- 当 PHASEx/SPHASEx 位提供对应于主控时基的相移时，值的有效范围为 0x0000 - 周期。

2： 如果 PWMCONx<9> = 1，则依工作模式适用以下情况：
- 互补、冗余和推挽 PWM 输出模式 （IOCONx<11:10> = 00、 01 或 10）， SPHASEx<15:0> = 不使用。
- 真正独立 PWM 输出模式 （IOCONx<11:10> = 11）， PHASEx<15:0> = 仅 PWMxL 的独立时基周期值。
- 当 PHASEx/SPHASEx 位提供本地周期时，值的有效范围为 0x0010 - 0xFFF8。

© 2011-2014 Microchip Technology Inc.

寄存器 3-16： DTRx：PWMx 死区寄存器

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	DTRx<13:8>					
bit 15							bit 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
DTRx<7:0>							
bit 7							bit 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 15-14 **未实现：**读为 0

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

寄存器 3-17： ALTDTRx：PWMx 备用死区寄存器

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	ALTDTRx<13:8>					
bit 15							bit 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
ALTDTRx<7:0>							
bit 7							bit 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 15-14 **未实现：**读为 0

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

寄存器 3-18： TRGCONx：PWMx 触发控制寄存器

R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0
TRGDIV3	TRGDIV2	TRGDIV1	TRGDIV0	—	—	—	—
bit 15							bit 8

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DTM(1)	—	TRGSTRT5(2)	TRGSTRT4(2)	TRGSTRT3(2)	TRGSTRT2(2)	TRGSTRT1(2)	TRGSTRT0(2)
bit 7							bit 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 15-12 **TRGDIV<3:0>：**触发器 # 输出分频比位
1111 = 每发生 16 个触发事件时触发输出
1110 = 每发生 15 个触发事件时触发输出
1101 = 每发生 14 个触发事件时触发输出
1100 = 每发生 13 个触发事件时触发输出
1011 = 每发生 12 个触发事件时触发输出
1010 = 每发生 11 个触发事件时触发输出
1001 = 每发生 10 个触发事件时触发输出
1000 = 每发生 9 个触发事件时触发输出
0111 = 每发生 8 个触发事件时触发输出
0110 = 每发生 7 个触发事件时触发输出
0101 = 每发生 6 个触发事件时触发输出
0100 = 每发生 5 个触发事件时触发输出
0011 = 每发生 4 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 11-8 **未实现：**读为 0

bit 7 **DTM：**双触发模式位 (1)
1 = 辅助触发事件与主触发事件组合产生 PWM 触发
0 = 辅助触发事件不与主触发事件组合产生 PWM 触发；产生两个独立的 PWM 触发

bit 6 **未实现：**读为 0

bit 5-0 **TRGSTRT<5:0>：**触发后分频比开始使能选择位 (2)
111111 = 使能模块之后，在产生第一个触发事件之前先等待 63 个 PWM 周期
•
•
•
000010 = 使能模块之后，在产生第一个触发事件之前先等待 2 个 PWM 周期
000001 = 使能模块之后，在产生第一个触发事件之前先等待 1 个 PWM 周期
000000 = 使能模块之后，在产生第一个触发事件之前先等待 0 个 PWM 周期

注 1： 辅助触发事件 （STRIGx）无法产生 PWM 触发中断。
2： 触发开始事件与主控 / 辅助主控时基的计满返回事件进行同步。

© 2011-2014 Microchip Technology Inc.

寄存器 3-19： IOCONx：PWMx I/O 控制寄存器

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(1,3)	PENL(1,3)	POLH(1,3)	POLL(1,3)	PMOD1(1,3)	PMOD0(1,3)	OVRENH(3)	OVRENL(3)
bit 15							bit 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
OVRDAT1(2,3)	OVRDAT0(2,3)	FLTDAT1(2,3)	FLTDAT0(2,3)	CLDAT1(2,3)	CLDAT0(2,3)	SWAP(3)	OSYNC(3)
bit 7							bit 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 15 **PENH：** PWMxH 输出引脚所有权位 (1,3)
1 = PWM 模块控制 PWMxH 引脚
0 = GPIO 模块控制 PWMxH 引脚

bit 14 **PENL：** PWMxL 输出引脚所有权位 (1,3)
1 = PWM 模块控制 PWMxL 引脚
0 = GPIO 模块控制 PWMxL 引脚

bit 13 **POLH：** PWMxH 输出引脚极性位 (1,3)
1 = PWMxH 引脚为低电平有效
0 = PWMxH 引脚为高电平有效

bit 12 **POLL：** PWMxL 输出引脚极性位 (1,3)
1 = PWMxL 引脚为低电平有效
0 = PWMxL 引脚为高电平有效

bit 11-10 **PMOD<1:0>：** PWM # I/O 引脚模式位 (1,3)
11 = PWM I/O 引脚对处于真正独立 PWM 输出模式
10 = PWM I/O 引脚对处于推挽 PWM 输出模式
01 = PWM I/O 引脚对处于冗余 PWM 输出模式
00 = PWM I/O 引脚对处于互补 PWM 输出模式

bit 9 **OVRENH：** PWMxH 引脚改写使能位 (3)
1 = OVRDAT1 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 **OVRENL：** PWMxL 引脚改写使能位 (3)
1 = OVRDAT0 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 **OVRDAT<1:0>：** 使能改写时 PWMxH 和 PWMxL 引脚状态位 (2,3)
如果 OVERENH = 1，则 OVRDAT1 提供 PWMxH 的数据。
如果 OVERENL = 1，则 OVRDAT0 提供 PWMxL 的数据。

bit 5-4 **FLTDAT<1:0>：** 使能 FLTMOD 时 PWMxH 和 PWMxL 引脚状态位 (2,3)
FCLCONx<15> = 0：正常故障模式：
如果故障有效，则 FLTDAT1 提供 PWMxH 的状态。
如果故障有效，则 FLTDAT0 提供 PWMxL 的状态。
FCLCONx<15> = 1：独立故障模式：
如果限流有效，则 FLTDAT1 提供 PWMxH 的状态。
如果故障有效，则 FLTDAT0 提供 PWMxL 的状态。

注 1： 使能 PWM 模块 （PTEN = 1）之后，不应更改这些位。
2： 状态代表 PWM 的有效 / 无效状态，具体取决于 POLH 和 POLL 位的设置。
3： 在支持 PWM 解锁功能的器件上，只有向 PWMKEY 寄存器写入正确的位序列之后， IOCONx 寄存器位才可写。关于 PWMKEY 寄存器的可用性，请参见具体器件的数据手册。

寄存器 3-19： IOCONx：PWMx I/O 控制寄存器 （续）

bit 3-2 **CLDAT<1:0>：**使能 CLMOD 时 PWMxH 和 PWMxL 引脚状态位 (2,3)

FCLCONx<15> = 0：正常故障模式：
如果限流有效，则 CLDAT1 提供 PWMxH 的状态。
如果限流有效，则 CLDAT0 提供 PWMxL 的状态。

FCLCONx<15> = 1：独立故障模式：
CLDAT<1:0> 位被忽略。

bit 1 **SWAP：**交换 PWMxH 和 PWMxL 引脚位 (3)

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 引脚映射到它们各自对应的引脚

bit 0 **OSYNC：**输出改写同步位 (3)

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： 使能 PWM 模块 （PTEN = 1）之后，不应更改这些位。
2： 状态代表 PWM 的有效 / 无效状态，具体取决于 POLH 和 POLL 位的设置。
3： 在支持 PWM 解锁功能的器件上，只有向 PWMKEY 寄存器写入正确的位序列之后， IOCONx 寄存器位才可写。关于 PWMKEY 寄存器的可用性，请参见具体器件的数据手册。

寄存器 3-20： TRIGx：PWMx 主触发比较值寄存器

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<12:5>							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0
TRGCMP<4:0>					—	—	—
bit 7							bit 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 15-3 **TRGCMP<12:0>：**触发控制值位
当主 PWM 采用本地时基时，该寄存器包含可触发 ADC 模块并产生 PWM 触发中断请求的比较值。

bit 2-0 **未实现：**读为 0

© 2011-2014 Microchip Technology Inc.

寄存器 3-21： STRIGx：PWMx 辅助触发比较值寄存器

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<12:5>(1)							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0
STRGCMP<4:0>(1)					—	—	—
bit 7							bit 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 15-3 **STRGCMP<12:0>：**辅助触发控制值位 (1)
当辅助 PWM 采用本地时基时，该寄存器包含可触发 ADC 模块的比较值。

bit 2-0 **未实现：**读为 0

注 1： STRIGx 寄存器位无法产生 PWM 触发中断。

寄存器 3-22： FCLCONx：PWMx 故障限流控制寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
IFLTMOD(4)	CLSRC4(2,3,4)	CLSRC3(2,3,4)	CLSRC2(2,3,4)	CLSRC1(2,3,4)	CLSRC0(2,3,4)	CLPOL(1,4)	CLMOD(4)
bit 15							bit 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
FLTSRC4(2,3,4)	FLTSRC3(2,3,4)	FLTSRC2(2,3,4)	FLTSRC1(2,3,4)	FLTSRC0(2,3,4)	FLTPOL(1,4)	FLTMOD1(4)	FLTMOD0(4)
bit 7							bit 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 15 **IFLTMOD：**独立故障模式使能位 (4)
1 = 在独立故障模式下，限流输入会将 FLTDAT1 映射到 PWMxH 输出，故障输入会将 FLTDAT0 映射到 PWMxL 输出； CLDAT<1:0> 位不用于改写功能
0 = 在正常故障模式下，限流模式会将 CLDAT<1:0> 位映射到 PWMxH 和 PWMxL 输出； PWM 故障模式会将 FLTDAT<1:0> 映射到 PWMxH 和 PWMxL 输出

bit 14-10 **CLSRC<4:0>：**PWM 发生器 # 的限流控制信号源选择位 (2,3,4)
这些位也指定死区补偿输入信号 DTCMPx 的来源。关于 CLSRCx 位的更多信息，请参见具体器件的数据手册。

bit 9 **CLPOL：**PWM 发生器 # 的限流极性位 (1,4)
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 8 **CLMOD：**PWM 发生器 # 的限流模式使能位 (4)
1 = 使能限流模式
0 = 禁止限流模式

bit 7-3 **FLTSRC<4:0>：**PWM 发生器 # 的故障控制信号源选择位 (2,3,4)
关于编码 FLTSRCx 位的更多信息，请参见具体器件的数据手册。

bit 2 **FLTPOL：**PWM 发生器 # 的故障极性位 (1,4)
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

bit 1-0 **FLTMOD<1:0>：**PWM 发生器 # 的故障模式位 (4)
11 = 禁止故障输入
10 = 保留
01 = 选定的故障源将 PWMxH 和 PWMxL 引脚强制为 FLTDATx 值 （周期）
00 = 选定的故障源将 PWMxH 和 PWMxL 引脚强制为 FLTDATx 值 （锁定状态）

注 1： 只有 PTEN = 0 时，才能更改这些位。

2： 当使能独立故障模式（IFLTMOD = 1）时，需要确保分别通过 CLSRCx 和 FLTSRCx 位为 PWMxH 和 PWMxL 选择正确的限流和故障源。例如，在某些器件中，CLSRCx 或 FLTSRCx 位的 ‘0b0000 编码指代故障 1 源。在此类器件中，如果为 CLSRCx 选择了故障 1，则必须对 FLTSRCx 使用不同 （或未用）的故障源，以防止故障 1 同时禁止 PWMxL 和 PWMxH 输出。类似地，如果为 FLTSRCx 选择了故障 1，则必须对 CLSRCx 使用不同 （或未用）的故障源，以防止故障 1 同时禁止 PWMxL 和 PWMxH 输出。

3： 关于可用故障引脚数的详细信息，请参见具体器件数据手册中的 **“引脚图”** 章节。

4： 在支持 PWM 解锁功能的器件上，只有向 PWMKEY 寄存器写入正确的位序列之后，FCLCONx 寄存器位才可写。关于 PWMKEY 寄存器的可用性，请参见具体器件的数据手册。

© 2011-2014 Microchip Technology Inc.

寄存器 3-23： LEBCONx：PWMx 前沿消隐控制寄存器 （版本 1）

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PHR(2)	PHF(2)	PLR(2)	PLF(2)	FLTLEBEN(2)	CLLEBEN(2)	LEB6(1,2)	LEB5(1,2)
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0
LEB4(1,2)	LEB3(1,2)	LEB2(1,2)	LEB1(1,2)	LEB0(1,2)	—	—	—
bit 7							bit 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 15 **PHR：** PWMxH 上升沿触发使能位 (2)
1 = PWMxH 的上升沿将触发前沿消隐计数器
0 = 前沿消隐忽略 PWMxH 的上升沿

bit 14 **PHF：** PWMxH 下降沿触发使能位 (2)
1 = PWMxH 的下降沿将触发前沿消隐计数器
0 = 前沿消隐忽略 PWMxH 的下降沿

bit 13 **PLR：** PWMxL 上升沿触发使能位 (2)
1 = PWMxL 的上升沿将触发前沿消隐计数器
0 = 前沿消隐忽略 PWMxL 的上升沿

bit 12 **PLF：** PWMxL 下降沿触发使能位 (2)
1 = PWMxL 的下降沿将触发前沿消隐计数器
0 = 前沿消隐忽略 PWMxL 的下降沿

bit 11 **FLTLEBEN：** 故障输入前沿消隐使能位 (2)
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 (2)
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 9-3 **LEB<6:0>：** 限流输入和故障输入前沿消隐位 (1,2)
消隐可以按照 2^n * 1/(附属时钟频率) ns 的步阶递增；其中，“n”为 PCLKDIV<2:0> 位（PTCON2<2:0>）的设置。

bit 2-0 **未实现：** 读为 0

注 1： 在最高 PWM 分辨率下， LEB<6:0> 位支持在 PWMxH 和 PWMxL 信号任意指定的上升沿或下降沿后，将限流和故障引脚消隐 （忽略） 0 ns 至 1057 ns，消隐周期以 8.32 ns 为单位进行递增。

2： 关于 LEBCONx 寄存器位的相关版本的更多信息，请参见具体器件的数据手册。

寄存器 3-24： LEBCONx：PWMx 前沿消隐控制寄存器 （版本 2）

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0
PHR(1)	PHF(1)	PLR(1)	PLF(1)	FLTLEBEN(1)	CLLEBEN(1)	—	—
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	BCH(1,2)	BCL(1,2)	BPHH(1)	BPHL(1)	BPLH(1)	BPLL(1)
bit 7							bit 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 15 **PHR：** PWMxH 上升沿触发使能位 (1)
1 = PWMxH 的上升沿将触发前沿消隐计数器
0 = 前沿消隐忽略 PWMxH 的上升沿

bit 14 **PHF：** PWMxH 下降沿触发使能位 (1)
1 = PWMxH 的下降沿将触发前沿消隐计数器
0 = 前沿消隐忽略 PWMxH 的下降沿

bit 13 **PLR：** PWMxL 上升沿触发使能位 (1)
1 = PWMxL 的上升沿将触发前沿消隐计数器
0 = 前沿消隐忽略 PWMxL 的上升沿

bit 12 **PLF：** PWMxL 下降沿触发使能位 (1)
1 = PWMxL 的下降沿将触发前沿消隐计数器
0 = 前沿消隐忽略 PWMxL 的下降沿

bit 11 **FLTLEBEN：** 故障输入前沿消隐使能位 (1)
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)
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 9-6 **未实现：** 读为 0

bit 5 **BCH：** 选定消隐信号高电平消隐使能位 (1,2)
1 = 当选定消隐信号为高电平时 （对限流和 / 或故障输入信号）进行状态消隐
0 = 当选定消隐信号为高电平时不进行消隐

bit 4 **BCL：** 选定消隐信号低电平消隐使能位 (1,2)
1 = 当选定消隐信号为低电平时 （对限流和 / 或故障输入信号）进行状态消隐
0 = 当选定消隐信号为低电平时不进行消隐

bit 3 **BPHH：** PWMxH 高电平消隐使能位 (1)
1 = 当 PWMxH 输出为高电平时 （对限流和 / 或故障输入信号）进行状态消隐
0 = 当 PWMxH 输出为高电平时不进行消隐

bit 2 **BPHL：** PWMxH 低电平消隐使能位 (1)
1 = 当 PWMxH 输出为低电平时 （对限流和 / 或故障输入信号）进行状态消隐
0 = 当 PWMxH 输出为低电平时不进行消隐

注 1： 关于 LEBCONx 寄存器位的相关版本的更多信息，请参见具体器件的数据手册。
2： 消隐信号通过 AUXCONx 寄存器中的 BLANKSEL 位进行选择。

© 2011-2014 Microchip Technology Inc.

寄存器 3-24： LEBCONx：PWMx 前沿消隐控制寄存器 （版本 2）（续）

bit 1 **BPLH**：PWMxL 高电平消隐使能位 (1)
1 = 当 PWMxL 输出为高电平时 （对限流和 / 或故障输入信号）进行状态消隐
0 = 当 PWMxL 输出为高电平时不进行消隐

bit 0 **BPLL**：PWMxL 低电平消隐使能位 (1)
1 = 当 PWMxL 输出为低电平时 （对限流和 / 或故障输入信号）进行状态消隐
0 = 当 PWMxL 输出为低电平时不进行消隐

注 1： 关于 LEBCONx 寄存器位的相关版本的更多信息，请参见具体器件的数据手册。
2： 消隐信号通过 AUXCONx 寄存器中的 BLANKSEL 位进行选择。

寄存器 3-25： LEBDLYx：PWMx 前沿消隐延时寄存器

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	LEB<8:5>(1,2)			
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0
LEB<4:0>(1,2)					—	—	—
bit 7							bit 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 15-12 **未实现：** 读为 0

bit 11-3 **LEB<8:0>：** 限流输入和故障输入前沿消隐延时位 (1,2)

值以 8.32 ns 为单位进行递增。

bit 2-0 **未实现：** 读为 0

注 1： 在最高 PWM 分辨率下， LEB<8:0> 位支持在 PWMxH 和 PWMxL 信号任意指定的上升沿或下降沿后，将限流和故障引脚消隐 （忽略） 0 ns 至 4252 ns，消隐周期以 8.32 ns 为单位进行递增。

2： 关于 LEBDLYx 寄存器位可用性的更多信息，请参见具体器件的数据手册。

© 2011-2014 Microchip Technology Inc.

寄存器 3-26： AUXCONx：PWMx 附属控制寄存器

R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
HRPDIS	HRDDIS	—	—	BLANKSEL3	BLANKSEL2	BLANKSEL1	BLANKSEL0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	CHOPSEL3	CHOPSEL2	CHOPSEL1	CHOPSEL0	CHOPHEN	CHOPLEN
bit 7							bit 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 15 **HRPDIS**：高分辨率 PWM 周期禁止位
1 = 禁止高分辨率 PWM 周期以降低功耗
0 = 使能高分辨率 PWM 周期

bit 14 **HRDDIS**：高分辨率 PWM 占空比禁止位
1 = 禁止高分辨率 PWM 占空比以降低功耗
0 = 使能高分辨率 PWM 占空比

bit 13-12 **未实现**：读为 0

bit 11-8 **BLANKSEL<3:0>**：PWM 状态消隐源选择位
选定的状态消隐信号将阻止限流和 / 或故障输入信号（如果通过 LEBCONx 寄存器中的 BCH 和 BCL 位使能）
1001 = 选择 PWM9H 作为状态消隐源
1000 = 选择 PWM8H 作为状态消隐源
0111 = 选择 PWM7H 作为状态消隐源
0110 = 选择 PWM6H 作为状态消隐源
0101 = 选择 PWM5H 作为状态消隐源
0100 = 选择 PWM4H 作为状态消隐源
0011 = 选择 PWM3H 作为状态消隐源
0010 = 选择 PWM2H 作为状态消隐源
0001 = 选择 PWM1H 作为状态消隐源
0000 = 不进行状态消隐

bit 7-6 **未实现**：读为 0

bit 5-2 **CHOPSEL<3:0>**：PWM 斩波时钟源选择位
选定信号将使能和禁止 （斩波）选定的 PWM 输出。
1001 = 选择 PWM9H 作为斩波时钟源
1000 = 选择 PWM8H 作为斩波时钟源
0111 = 选择 PWM7H 作为斩波时钟源
0110 = 选择 PWM6H 作为斩波时钟源
0101 = 选择 PWM5H 作为斩波时钟源
0100 = 选择 PWM4H 作为斩波时钟源
0011 = 选择 PWM3H 作为斩波时钟源
0010 = 选择 PWM2H 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 **CHOPLEN**：PWMxL 输出斩波使能位
1 = 使能 PWMxL 斩波功能
0 = 禁止 PWMxL 斩波功能

寄存器 3-27： PWMCAPx：PWMx 主时基捕捉寄存器

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
PWMCAP<12:5>(1,2,3)							
bit 15							bit 8

R-0	R-0	R-0	R-0	R-0	U-0	U-0	U-0
PWMCAP<4:0>(1,2,3)					—	—	—
bit 7							bit 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 15-3 **PWMCAP<12:0>：**捕捉的 PWM 时基值位 (1,2,3)
该寄存器中的值代表在限流输入上检测到前沿时捕捉的 PWM 时基值。

bit 2-0 **未实现：**读为 0

注 1： 捕捉功能仅在主输出 （PWMxH）上可用，并且仅在对限流输入信号的 LEB 处理完成后有效。
2： 最小捕捉分辨率为 8.32 ns。
3： 只有 XPRES （PWMCONx<1>） = 0 时，才能使用该功能。

寄存器 3-28： PWMKEY：PWMx 保护锁定 / 解锁密钥寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PWMKEY<15:8>(1)							
bit 15							bit 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
PWMKEY<7:0>(1)							
bit 7							bit 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 15-0 **PWMKEY<15:0>：**PWM 保护锁定 / 解锁密钥值位 (1)

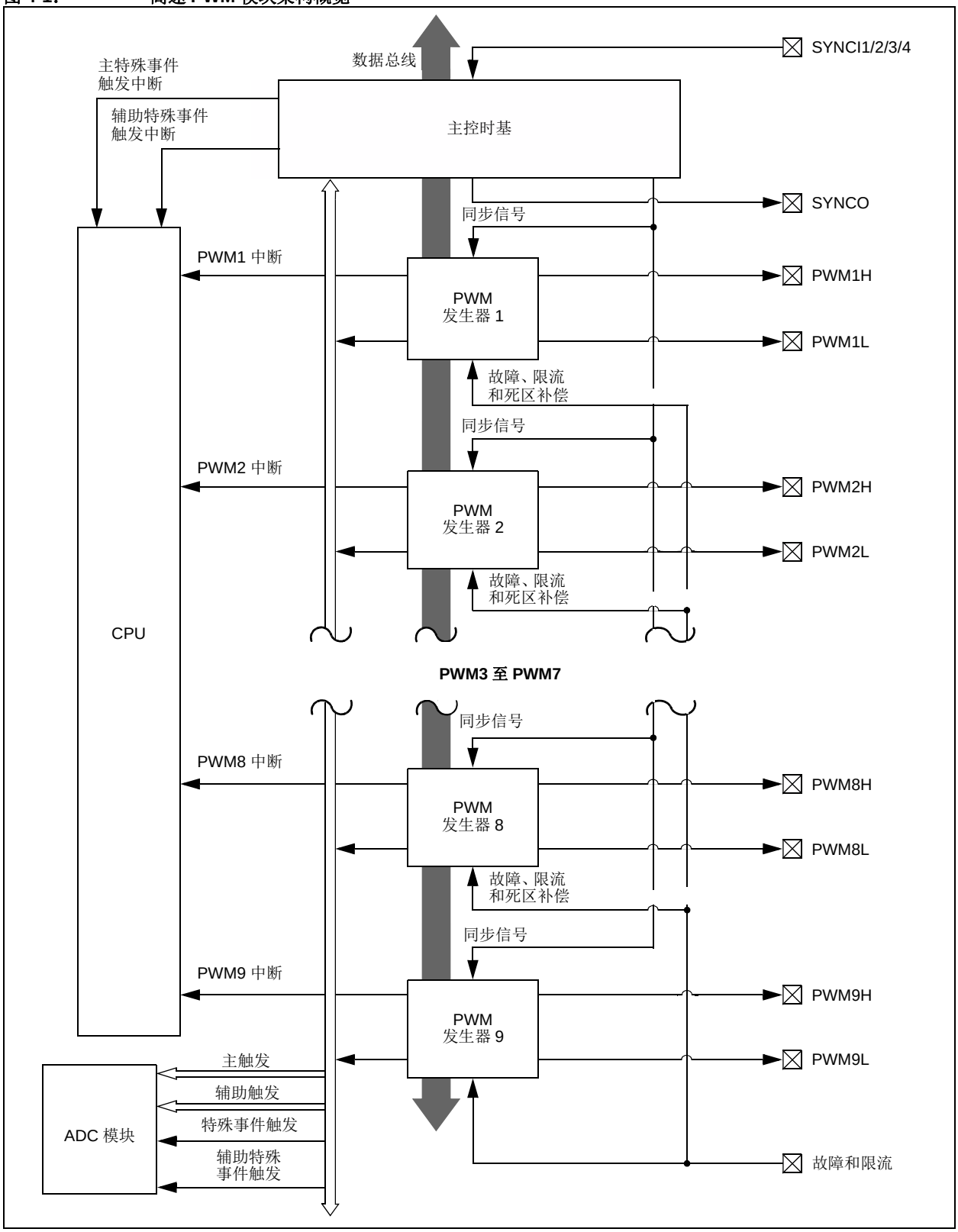
注 1： 关于 PWMKEY 寄存器位的可用性，请参见具体器件的数据手册。

© 2011-2014 Microchip Technology Inc.

4.0 架构概述

图 4-1 显示了高速 PWM 模块的架构概览，以及它与 CPU 和其他外设的相互连接。

图 4-1： 高速 PWM 模块架构概览



高速 PWM 模块最多包含 9 个 PWM 发生器。每个 PWM 发生器都提供了两个 PWM 输出：PWMxH 和 PWMxL。主控时基发生器提供同步信号，作为用于同步各个 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。各个 PWM 输出均可在器件输出引脚上提供。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 都可以向 ADC 模块产生触发信号，使之在 PWM 周期中的特定时刻对模拟信号进行采样。此外，高速 PWM 模块还可以根据主控时基向 ADC 模块产生特殊事件触发信号。

在主控时基模式下，高速 PWM 模块可以将自身与外部信号同步，也可以用作任意外部器件的同步源。SYNCIx 引脚是输入引脚，它们可以将高速 PWM 模块与外部信号同步。SYNCOx 引脚是向外部器件提供同步信号的输出引脚。

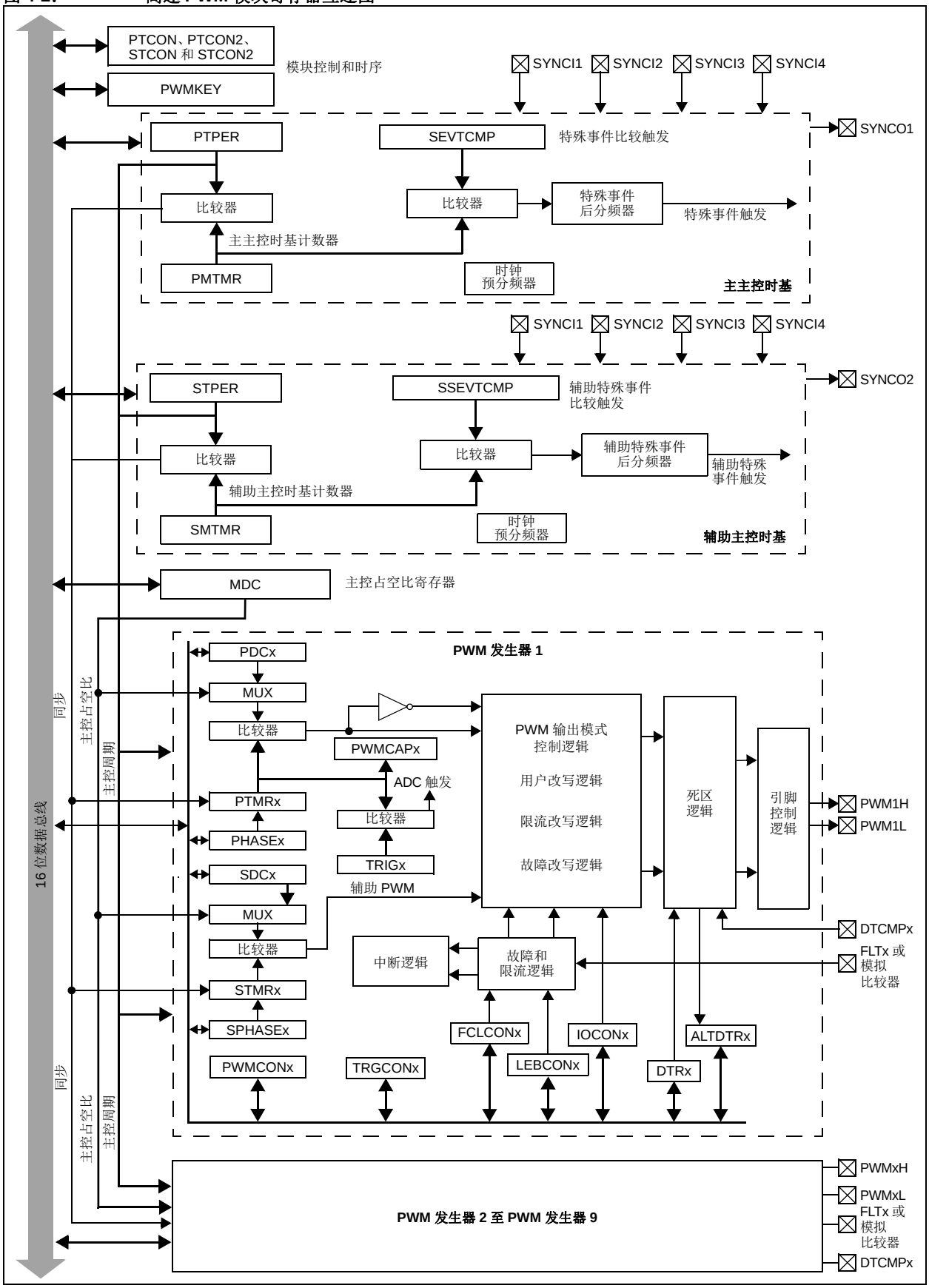
高速 PWM 模块可用于要求以下特性的多种电源转换应用：

- 高工作频率与高分辨率
- 可动态控制 PWM 参数，例如占空比、周期和死区
- 可独立控制每个 PWM
- 可同步控制所有 PWM
- 可为每个 PWM 发生器独立分配资源
- 故障处理能力
- 支持 CPU 负载交错，可执行多个控制环

后面几节介绍了高速 PWM 模块的每个功能。图 4-2 给出了高速 PWM 模块中各个寄存器之间的互连图。

© 2011-2014 Microchip Technology Inc.

图 4-2：　　高速 PWM 模块寄存器互连图



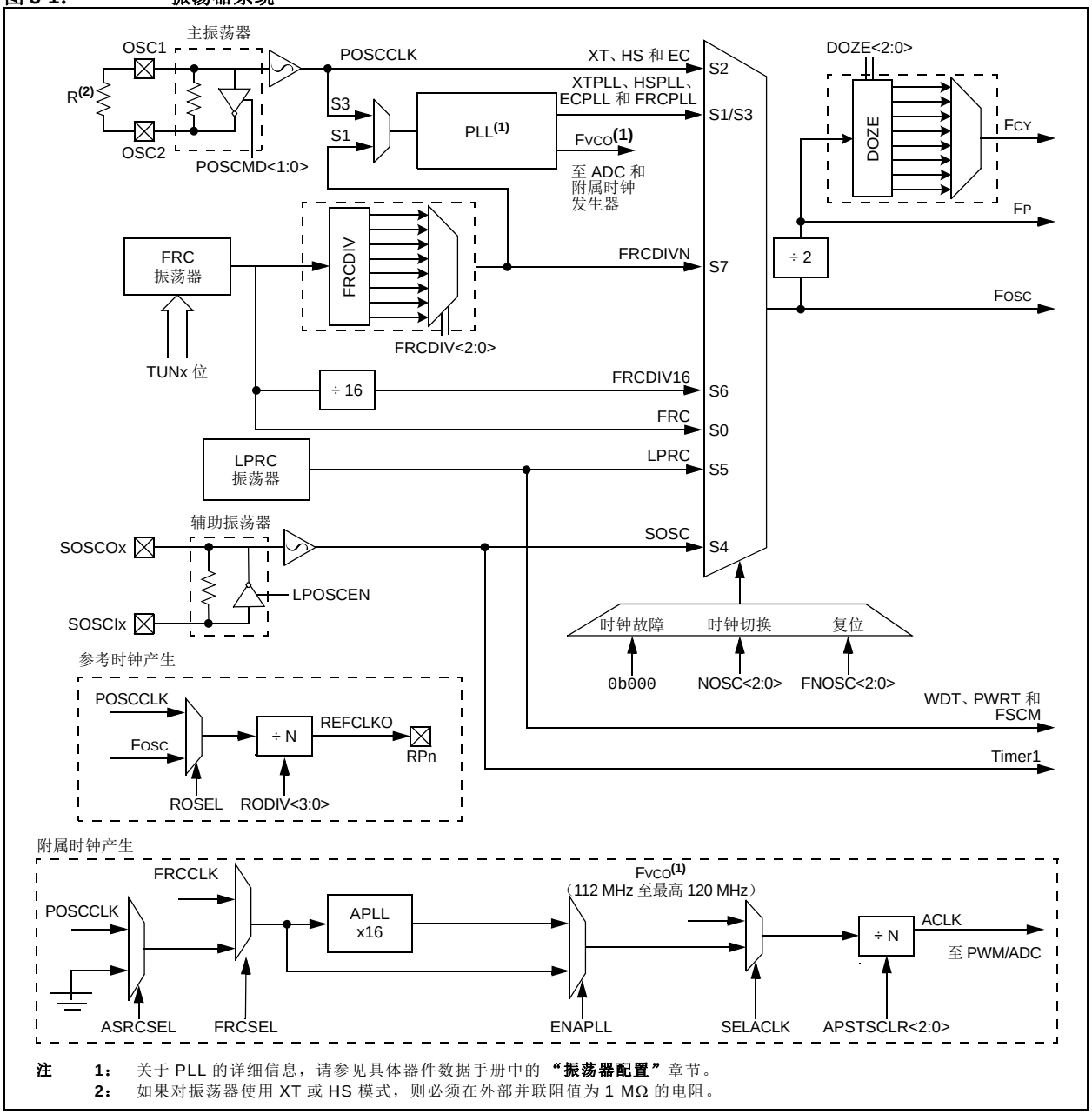
5.0 模块说明

5.1 PWM 时钟选择

必须使用附属时钟发生器生成独立于系统时钟的 PWM 模块时钟。主振荡器时钟（POSCCLK）、主锁相环 （PLL）、主 PLL 输出 （FVCO）和内部 FRC 时钟 （FRCCLK）可以与附属 PLL 一起用于获得附属时钟 （ACLK）。附属 PLL 含有固定的 16x 倍频因子。例 5-1 给出了使用 FRC 的附属时钟的配置。例 5-2 给出了使用主振荡器 （POSC）的附属时钟的配置。

附属时钟控制寄存器 （ACLKCON）用于选择参考时钟、使能附属 PLL 和输出分频器，以获得所需的附属时钟。公式 5-1 给出了参考时钟（REFCLK）输入频率与 ACLK 频率之间的关系。图 5-1 给出了振荡器系统的图示。

图 5-1： 振荡器系统



注 1： 关于 PLL 的详细信息，请参见具体器件数据手册中的 **“振荡器配置”** 章节。

2： 如果对振荡器使用 XT 或 HS 模式，则必须在外部并联阻值为 1 MΩ 的电阻。

© 2011-2014 Microchip Technology Inc.

关于时钟发生器配置的更多信息，请参见具体器件数据手册中的“**振荡器配置**”章节和与**“振荡器”**相关的 《dsPIC33/PIC24 系列参考手册》章节。

公式 5-1： ACLK 频率计算

$$ACLK = \frac{REFCLK \times M1}{N}$$

其中：

REFCLK = 内部 FRC 时钟频率 （7.37 MHz），如果选择内部 FRC 作为时钟源

或

REFCLK = 主振荡器时钟频率 （POSCCLK），如果选择主振荡器 （POSC）作为时钟源

*M*1 = 16，如果通过将 ENAPLL 位 （ACLKCON<15>）置 1 而使能了附属 PLL

或

*M*1 = 1，如果禁止了附属 PLL

N = 通过附属时钟控制寄存器中的附属后分频比位 APSTSCLR<2:0> （ACLKCON<2:0>）选择的后分频比

注 1： PWM 的标称输入时钟频率应为 120 MHz。关于全部工作范围，请参见具体器件数据手册中的 **“电气特性”**章节。

2： 使用 OSCTUN 寄存器的 TUN<5:0> 位来微调 FRC 时钟频率，以获得最高 PWM 分辨率 1.04 ns。更多详细信息，请参见具体器件数据手册中的**“振荡器配置”**章节。

3： 为了使 PWM 模块正常工作，附属时钟后分频比必须配置为 1 分频 （APSTSCLR<2:0> = 111）。

例 5-1： 使用 FRC 来设置 ACLK

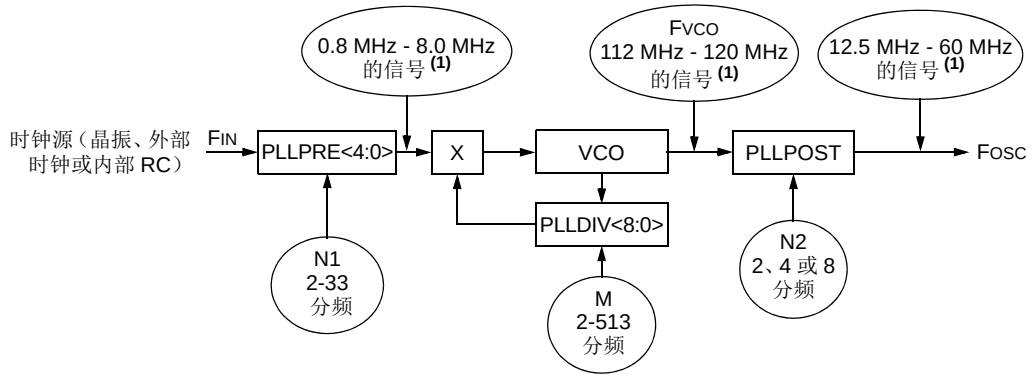
```
/* Setup for the Auxiliary clock to use the FRC as the REFCLK */
/* ((FRC * 16) / APSTSCLR) = (7.37 * 16) / 1 = 117.92 MHz */

ACLKCONbits.FRCSEL = 1;            /* FRC is input to Auxiliary PLL */
ACLKCONbits.SELACLK = 1;           /* Auxiliary Oscillator provides the clock
                                      source*/
ACLKCONbits.APSTSCLR = 7;          /* Divide Auxiliary clock by 1 */
ACLKCONbits.ENAPLL = 1;            /* Enable Auxiliary PLL */
while(ACLKCONbits.APLLCK! = 1)     /* Wait for Auxiliary PLL to Lock */
```

例 5-2： 使用 POSC 来设置 ACLK

```
/* Setup for the Auxiliary clock to use the primary oscillator(7.37 MHz) as
the REFCLK */
/*((primary oscillator* 16) / APSTSCLR) = (7.37 * 16) / 1 = 117.9 MHz */

ACLKCONbits.ARCSEL = 1;            /* Primary Oscillator is the Clock Source */
ACLKCONbits.FRCSEL = 0;            /* Input clock source is determined by
                                      ASRCSEL bit setting */
ACLKCONbits.SELACLK = 1;           /* Auxiliary Oscillator provides the clock
                                      source*/
ACLKCONbits.APSTSCLR = 7;          /* Divide Auxiliary clock by 1 */
ACLKCONbits.ENAPLL = 1;            /* Enable Auxiliary PLL */
while(ACLKCONbits.APLLCK != 1);    /* Wait for Auxiliary PLL to Lock */
```

当器件运行于主 PLL 模式时，可以从系统时钟产生 PWM 模块的 ACLK。公式 5-3 给出了 FVCO 频率和 ACLK 频率之间的关系。图 5-2 给出了 FVCO 作为 ACLK 时钟源的框图。计算 FVCO 的公式如公式 5-2 所示。例 5-3 给出了使用 FVCO 作为附属时钟源的示例。

图 5-2：　　FVCO 作为附属时钟的时钟源



注　　1：　当使用 FVCO 作为附属时钟的时钟源时，必须满足该频率范围。

公式 5-2：　　FVCO 计算

$$FVCO = \frac{FIN \times M}{N1} = FIN \times \left(\frac{PLLDIV<8:0> + 2}{PLLPRE<4:0> + 2}\right)$$

其中：

FVCO = VCO 输出频率

FIN = 时钟源 （晶振、外部时钟或内部 RC）的输入频率

M = 通过 PLLDIV<8:0> 选择的 PLL 反馈分频比

N1 = 通过 PLLPRE<4:0> 选择的 PLL 预分频比

公式 5-3：　　使用 FVCO 的 ACLK 频率计算

$$ACLK = \frac{FVCO}{N}$$

其中：

N = 通过 APSTSCLR<2:0> 位 （ACLKCON<2:0>）选择的后分频比

FVCO = VCO 输出频率

ACLK = 附属时钟频率

注：　如果使用主PLL作为附属时钟的来源，则主PLL的最大工作频率应配置为FCY = 30 MHz或更低，FVCO 必须处于 112 MHz 至 120 MHz 范围内。当 FVCO 作为附属时钟的时钟源时，最小 PWM 分辨率为 8.32 ns。

© 2011-2014 Microchip Technology Inc.

例 5-3：　　使用 Fvco 作为附属时钟源

```
/* Assume Primary Oscillator is 8 MHz and FCY = 30 MHz. */
/* Therefore, Fosc = 60 MHz */
/* Setup for the Auxiliary clock to use Fvco as the source */
/* Fosc = Primary Oscillator * (PLLDIV / PLLPOST * PLLPRE) */
/* Fvco = Fosc * N2 */
/* Fosc = 60 MHz; N2 = 2; Fvco = 120 MHz; M = 30 */
/* Input to the Vco = 4 MHz; N1 = 2; Fin = 8 MHz */

ACLKCONbits.SELACLK = 0;       /* Primary PLL (Fvco) provides the source clock
                                  for the auxiliary clock divider */

/* Configuring PLL prescaler, PLL Post scaler, PLL divider */

PLLFBD = 28;                   /* M = 30 */
CLKDIVbits.PLLPOST = 0;        /* N1 = 2 */
CLKDIVbits.PLLPRE = 0;         /* N2 = 2 */
ACLKCONbits.APSTSCLR = 7;      /* Divide Auxiliary click by 1 */
while (OSCCONbits.LOCK ! = 1); /* Wait for PLL to lock */
```

5.2　时基

PWM 发生器中的每个 PWM 输出可以使用主控时基，也可以使用独立时基（ITB）。高速 PWM 模块的输入时钟具有 1:1 至 1:64 的预分频比选项，可以使用 PWM 时钟分频比选择寄存器中的 PWM 输入时钟预分频比选择位 PCLKDIV<2:0>（PTCON2<2:0>）进行选择。该预分频比会影响所有 PWM 时基。预分频后的值也会反映 PWM 分辨率，可以帮助降低高速 PWM 模块的功耗。预分频后的时钟是 PWM 时钟控制逻辑模块的输入。最大时钟速率可提供 1.04 ns 的占空比和周期分辨率。

例如：

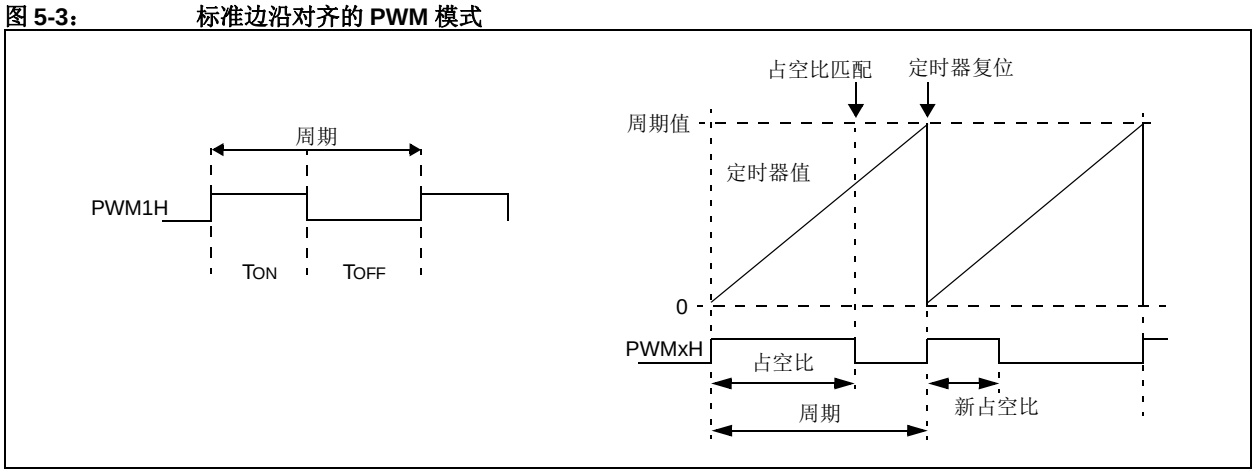
- 如果选择了 1:2 的预分频比选项且 ACLK = 120 MHz，则 PWM 占空比和周期分辨率可以设置为 2.08 ns。因此，与最大速度工作时相比，高速 PWM 模块的功耗可下降约 50%。
- 如果选择了 1:4 的预分频比选项且 ACLK = 120 MHz，则 PWM 占空比和周期分辨率可以设置为 4.16 ns。因此，与最大速度工作时相比，高速 PWM 模块的功耗可下降约 75%。

高速 PWM 模块可工作于标准边沿对齐或中心对齐时基模式。

5.3 标准边沿对齐的 PWM

图 5-3 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 （周期）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。

图 5-3： 标准边沿对齐的 PWM 模式



© 2011-2014 Microchip Technology Inc.

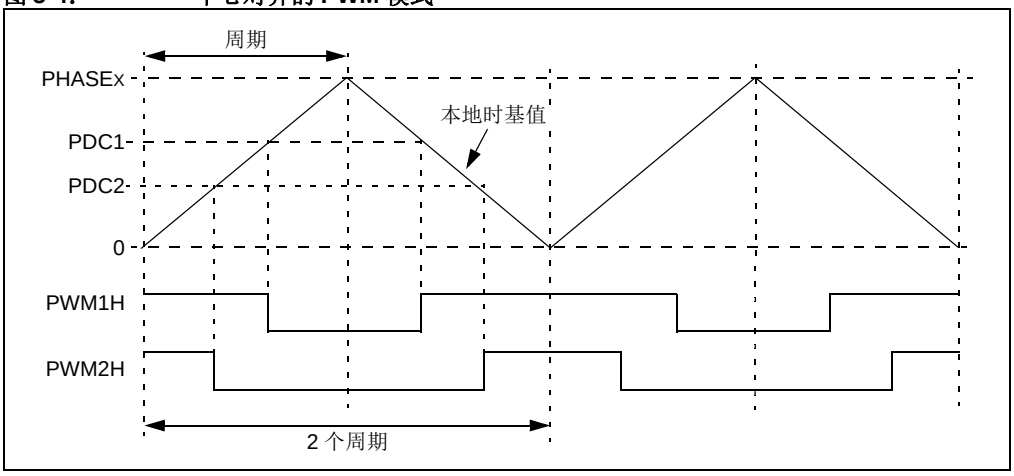
5.4 中心对齐的 PWM

中心对齐的 PWM 波形 （如图 5-4 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 PWMx 控制寄存器中的中心对齐模式使能位 CAM （PWMCONx<2>）置 1 时，将使能中心对齐模式。

工作于中心对齐模式时，实际 PWM 周期是 PWMx 主相移寄存器（PHASEx）指定值的两倍，因为在相应的周期中， PWM 发生器中的独立时基计数器先递增然后再递减。递增和递减计数序列会使实际 PWM 周期加倍。许多电机控制和不间断电源应用中都使用了该模式。例 5-4 给出了边沿对齐或中心对齐模式选择的配置。图5-5给出了UPS应用中的中心对齐PWM模式的典型应用。

注： 只有使能独立时基模式 （ITB = 1）时，才能使用中心对齐模式。如果 ITB = 0，则 CAM 位 （PWMCONx<2>）会被忽略。

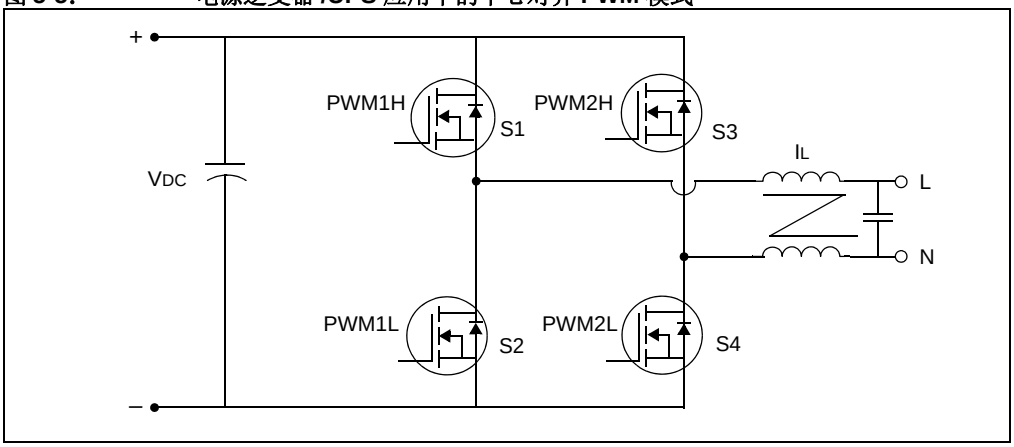
图 5-4： 中心对齐的 PWM 模式



例 5-4： 边沿对齐或中心对齐 PWM 模式选择

```
/* Select Edge-Aligned PWM*/

PWMCON1bits.CAM = 0; /* For Edge-Aligned Mode */

/* Select Center-Aligned PWM*/

PWMCON1bits.CAM = 1; /* For Center-Aligned Mode */
PWMCON1bits.ITB = 1; /* Enable Independent Time Base */
```

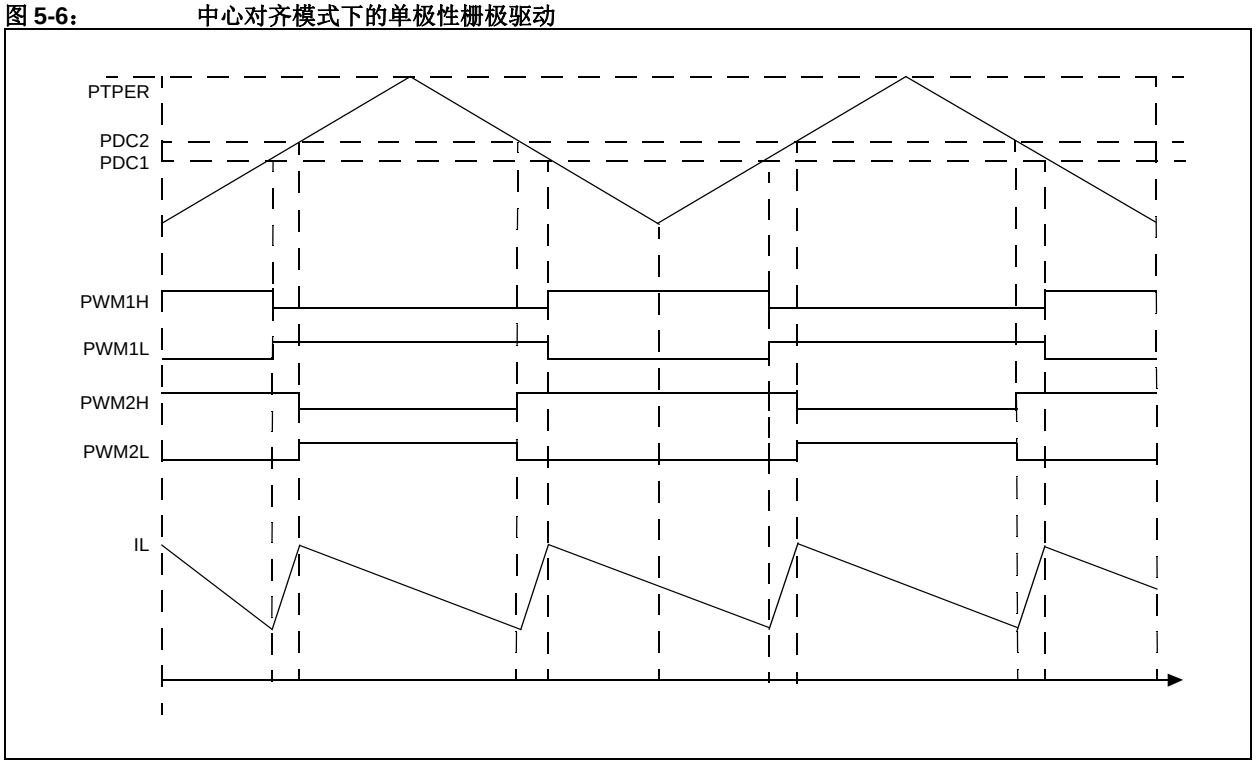
图 5-5： 电源逆变器 /UPS 应用中的中心对齐 PWM 模式



5.4.1 中心对齐模式在 UPS 应用中的优点

电流纹波频率和噪声频率是开关频率的两倍。由于电流纹波的开关频率增倍，所以可以实现幅度较低的电流纹波。电流纹波较低有助于放宽对于直流输入电容和输出滤波电感与电容的要求。电流纹波较低还有助于降低输出电流谐波。图 5-6 给出了配置为在中心对齐模式下进行单极性栅极驱动的 UPS 的典型波形。

图 5-6： 中心对齐模式下的单极性栅极驱动

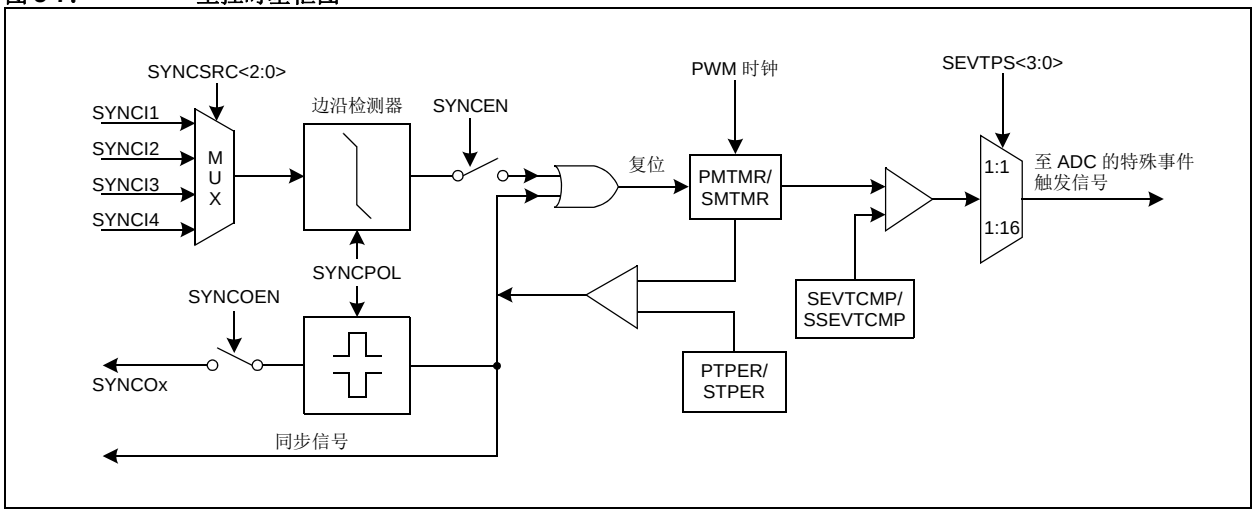


© 2011-2014 Microchip Technology Inc.

5.5 主控时基 / 同步时基

图 5-7 显示了主控时基模式下的 PWM 功能。

图 5-7： 主控时基框图



以下是主控时基的一些常见任务：

- 产生所有 PWM 发生器的时间参考
- 产生 ADC 特殊事件触发和中断
- 支持与外部 SYNCIx 信号 （SYNCI1/SYNCI2/SYNCI3/SYNCI4）进行同步
- 支持使用 SYNCOx 信号与外部器件进行同步

PWM 发生器主控时基的设置方法是向 PWMx 主控时基周期寄存器（PTPER/STPER）中装入一个 16 位值。在主控时基模式下， PHASEx 和 SPHASEx 寄存器中的值提供 PWM 输出之间的相移。PWM 定时器 （PMTMR/SMTMR）的时钟基于系统时钟而产生。

5.6 时基同步

主控时基可以通过主控时基同步信号 （SYNCI1/SYNCI2/SYNCI3/SYNCI4）与外部同步信号进行同步。同步源 （SYNCI1、 SYNCI2、 SYNCI3 和 SYNCI4）可以使用 PWMx 时基控制寄存器中的同步源选择位 SYNCSRC<2:0>（PTCON<6:4>）进行选择。PWMx 时基控制寄存器中的同步输入 / 输出极性位 SYNCPOL （PTCON<9>/STCON<9>）可用于选择同步脉冲的上升沿或下降沿，它会复位定时器 （PMTMR/SMTMR）。外部同步功能可以使用 PWMx 时基控制寄存器中的外部时基同步使能位 SYNCEN （PTCON<7>/STCON<7>）使能或禁止。外部同步信号（SYNCI1/SYNCI2/SYNCI3/SYNCI4）的脉冲宽度应大于 200 ns，以确保主控时基可以可靠地检测到它。

此外，还可以通过使用同步输出 （SYNCOx）信号，将外部器件与主控时基进行同步。SYNCOx 信号在 PTPER/STPER 寄存器复位 PMTMR/SMTMR 定时器时产生。SYNCOx 信号脉冲的宽度为 12 个 TCY 时钟 （40 MIPS 时约为 300 ns， 70 MIPS 时约为 170 ns），以确保其他器件可以检测到该信号。SYNCOx 信号的极性由 PTCON/STCON 寄存器中的 SYNCPOL 位决定。SYNCOx 信号可以通过选择 PTCON/STCON 寄存器中的主时基同步使能位 SYNCOEN （PTCON<8>/STCON<8>）进行使能或禁止。

注 1： SYNCIx 脉冲的周期应小于 PWM 周期值。

2： SYNCIx 脉冲应为连续脉冲，最小脉冲宽度为 200 ns。

3： 对于前两个 SYNCIx 脉冲，预期 PWM 周期会发生失真。

4： 为了让外部同步可以在推挽模式下工作，周期值应为 8 的倍数（低 3 位设置为 0）。

5： 在推挽模式下使用外部同步时，必须以所需 PWM 频率的两倍频率来生成外部同步信号。

6： 从SYNC信号输入到内部时基计数器发生复位之间存在一定的延时；该延时约为30 ns。

7： 外部时基同步不能与相移 PWM 一起使用，因为同步信号无法维持多个 PWM 通道之间的相位关系。

8： 外部时基同步不能在独立时基模式下使用。

进行同步的好处是它可以确保在使用多个电源控制器时，不会产生拍频。例 5-5 给出了将主控时基与外部信号同步的配置。

例 5-5： 将主控时基与外部信号同步

```
/* Synchronizing Master time base with external signal */

PTCONbits.SYNCSRC = 0;     /* Select SYNC1 input as synchronizing source */
PTCONbits.SYNCPOL = 0;     /* Rising edge of SYNC1 resets the PWM Timer */
PTCONbits.SYNCEN  = 1;     /* Enable external synchronization */
```

例 5-6 给出了将外部器件与主控时基同步的配置。

例 5-6： 将外部器件与主控时基同步

```
/* Synchronizing external device with Master time base */

PTCONbits.SYNCPOL = 0;     /* SYNCO output is active-high */
PTCONbits.SYNCOEN = 1;     /* Enable SYNCO output */
```


© 2011-2014 Microchip Technology Inc.

5.7 特殊事件触发器

高速 PWM 模块含有一个主控特殊事件触发器，可用作 CPU 中断源，用于将模数转换与 PWM 时基进行同步。可以将模数采样时间编程为在 PWM 周期内的任意时刻发生。特殊事件触发器使用户应用能够将采集模数转换结果和更新占空比值之间的延时降至最短。特殊事件触发器基于主控时基工作。

主控特殊事件触发值装入 PWMx 特殊事件比较寄存器 （SEVTCMP/SSEVTCMP）中。此外，PWMx 时基控制寄存器或 PWMx 辅助主控时基控制寄存器中的 PWM 特殊事件触发信号输出后分频比选择位 SEVTPS<3:0>（PTCON<3:0>/STCON<3:0>）控制特殊事件触发操作。要向 ADC 模块产生触发信号，模块会将 PWM 主控时基计数器 （PTPER/STPER）中的值与 SEVTCMP/SSEVTCMP 寄存器中的值进行比较。特殊事件触发器含有一个允许后分频比为 1:1 至 1:16 的后分频器。后分频比通过写入 SEVTPS<3:0> 控制位 （PTCON<3:0>）来进行配置。

如果满足以下条件，则会产生特殊事件触发脉冲：

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

特殊事件触发信号输出后分频器在发生以下事件时清零：

- 任何器件复位
- 当 PTEN （PTCON<15>） = 0 时

例 5-7 给出了 ADC 特殊事件触发的配置。

例 5-7： ADC 特殊事件触发配置

```
/* ADC Special Event Trigger configuration */

SEVTCMP = 1248;                    /* Special Event Trigger value set at ~25%
                                      of period value (4999)*/
PTCONbits.SEVTPS = 0;              /* Special Event Trigger output postscaler
                                      set to 1:1 selection (trigger generated
                                      every PWM cycle */
PTCONbits.SEIEN = 0;               /* Special event interrupt is disabled */
while (PTCONbits.SESTAT == 0);     /* Wait for special event status change */
```

除了产生 ADC 触发信号之外，特殊事件触发器还可用于在发生比较匹配事件时产生主和辅助特殊事件触发中断。

5.8 独立 PWM 时基

图 5-8 和图 5-9 显示了独立时基模式下的 PWM 功能。

图 5-8： 独立时基框图 （对于不具有辅助主控时基的器件）

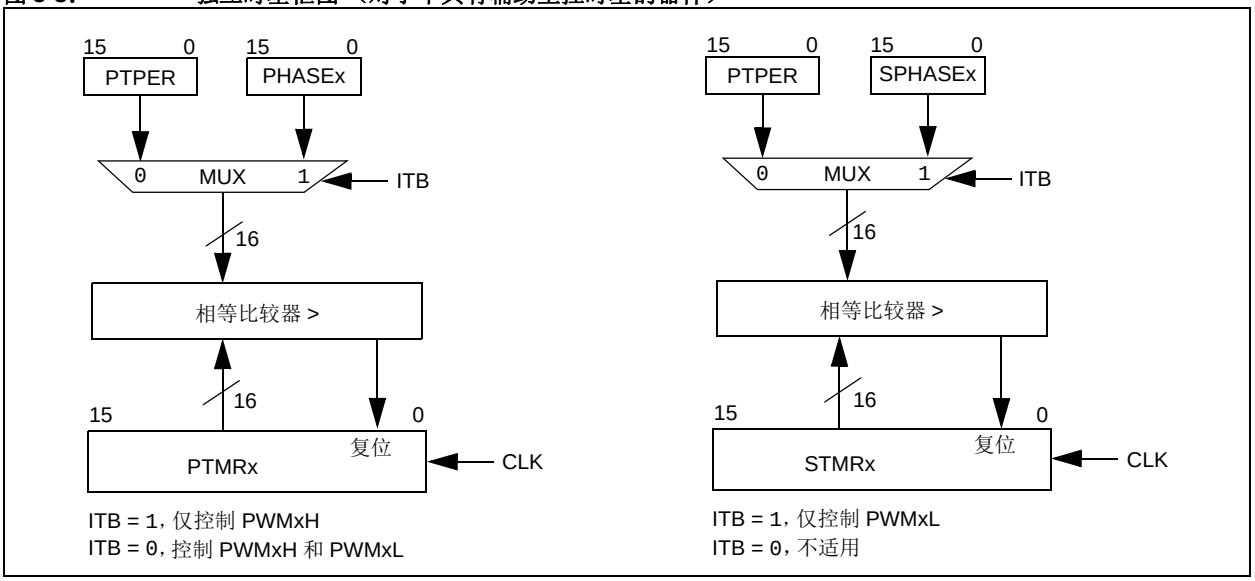
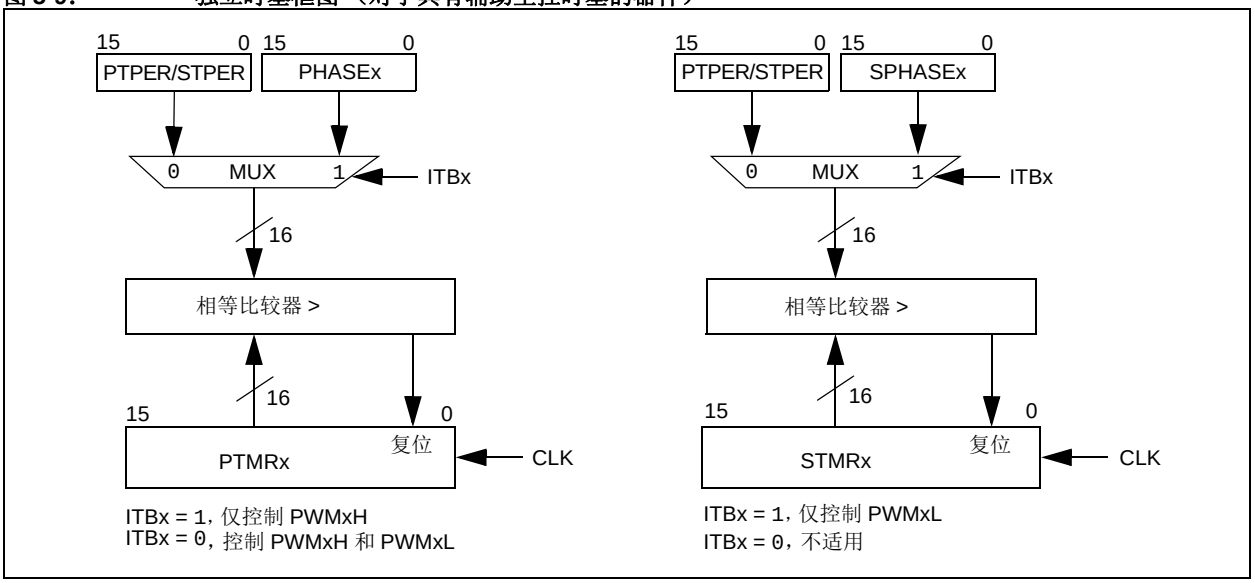


图 5-9： 独立时基框图 （对于具有辅助主控时基的器件）



在独立时基模式下，每个 PWM 发生器可以按以下方式工作：

- 主输出 （PWMxH）和辅助输出 （PWMxL）共用时基
  该操作在互补、冗余或推挽模式下发生。两个 PWM 输出 （PWMxH 和 PWMxL）的独立时基周期通过 PHASEx 寄存器中的值提供。
- 主输出 （PWMxH）和辅助输出 （PWMxL）使用各自的专用时基
  该操作仅在独立输出模式下发生。PWMxH 输出的独立时基周期通过 PHASEx 寄存器中的值提供。PWMxL 输出的独立时基周期通过 PWM 辅助相移寄存器 （SPHASEx）中的值提供。

注： PTMRx 和 STMRx 值不可由用户应用程序读取。

© 2011-2014 Microchip Technology Inc.

6.0 PWM 发生器

本节介绍 PWM 发生器的功能。

6.1 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，也可以通过相移寄存器 PHASEx 和 SPHASEx 控制（分别用于主 PWM 输出和辅助 PWM 输出）。

当高速 PWM 模块工作于独立时基模式（PWMCONx<9> = 1）时，PWM 周期值可以通过两种方式进行控制：

- 在互补、冗余和推挽模式下， PHASEx 寄存器控制 PWM 输出信号 （PWMxH 和 PWMxL）的 PWM 周期。
- 在真正独立 PWM 输出模式下， PHASEx 寄存器控制 PWMxH 输出信号的 PWM 周期，SPHASEx 寄存器控制 PWMxL 输出信号的 PWM 周期。

关于各种 PWM 模式及其特性的详细信息，请参见**第 9.0 节 “PWM 工作模式”**。

当高速PWM工作于主控时基模式时，PTPER/STPER寄存器存放一个16位值，该值指定PMTMR/SMTMR 定时器的计数周期。当高速 PWM 模块工作于独立时基模式时，PHASEx 和 SPHASEx 寄存器各存放一个 16 位值，分别指定 PTMRx 和 STMRx 定时器的计数周期。用户应用程序可以在运行时更新 PWM 周期。可通过公式 6-1 确定 PWM 时间周期。

公式 6-1： PTPER、 STPER、 PHASEx 和 SPHASEx 寄存器值计算

$$PTPER, STPER, PHASEx, SPHASEx = \left(\frac{ACLK \times 8 \times \text{所需的 PWM 周期}}{\text{PWM 输入时钟预分频比（PCLKDIV<2:0>）}} \right) - 8$$

$$ACLK = \frac{REFCLK \times M1}{N}$$ 请参见公式 5-1

（或）

$$ACLK = \frac{FVCO}{N}$$ 请参见公式 5-2

其中：

REFCLK = FRC = 7.49 MHz （ACLKCON<6> = 1）

M1 = 使能 16 倍频附属 PLL （ENAPLL = 1）

N = 通过附属时钟控制寄存器中的附属后分频比位 APTSTSCLR<2:0>（ACLKCON<2:0>）选择的后分频比

注： 使用 OSCTUN 寄存器的 TUN<5:0> 位将 FRC 时钟频率微调为 7.49 MHz，以获得最高 PWM 分辨率 1.04 ns。更多详细信息，请参见具体器件数据手册中的**“振荡器配置”**章节。

根据公式 6-1，例 6-2 给出了要在使用 PTPER 寄存器或 PHASEx 和 SPHASEx 寄存器工作时装入的寄存器值。

公式 6-2：　　PWM 时间周期计算

$$ACLK = \left[\frac{7.49\ MHz * 16}{1}\right] = 119.84\ MHz$$

其中：

REFCLK = 7.49 MHz

*M*1 = 16

N = 1

$$PTPER, STPER, PHASEx, SPHASEx = \left[\frac{119.84\ MHz * 8 * 10\ \mu s}{1}\right] - 8 = 9579$$

其中：

PCLKDIV<2:0> = 1:1

$$所需的\ PWM\ 周期 = \frac{1}{所需的\ PWM\ 开关频率}$$

所需的 PWM 开关频率 = 100 kHz

可获得的最大 PWM 周期分辨率为 1.04 ns。PWM 输入时钟预分频比选择位 PCLKDIV<2:0>（PTCON2<2:0>/STCON2<2:0>）决定 PWM 时钟的类型。可以通过置 1 或清零 PMW 模块使能位 PTEN （PTCON<15>）来使能或禁止定时器 / 计数器。PMTMR/SMTMR 定时器也可以使用 PTEN 位 （PTCON<15>）清零。

如果使能立即周期更新位 EIPU（PTCON<10>/STCON<10>）置 1，则会立即更新所使用的主控周期寄存器（内部影子寄存器），而不是先等待 PWM 周期结束。EIPU 位会影响 PMTMR/SMTMR 主控时基。例 6-1 显示了时钟预分频比选择。例 6-2 显示了 PWM 时间周期选择。例 6-3 显示了 PWM 时间周期初始化。

例 6-1：　　时钟预分频比选择

```
/* Select PWM time base input clock prescaler */
/* Choose divide ratio of 1:2, which affects all PWM timing operations */

PTCON2bits.PCLKDIV = 1;
```

例 6-2：　　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 6-3：　　PWM 时间周期初始化

```
/* Choose PWM time period based on FRC input clock */
/* PWM frequency is 100 kHz */
/* Choose one of the following options */

PTPER = 9579;      /* When PWMCONx<9> = 0 */

PHASEx = 9579;     /* When PWMCONx<9> = 1 */

SPHASEx = 9579;    /* When PWMCONx<9> = 1 */
```


© 2011-2014 Microchip Technology Inc.

6.2 PWM 占空比控制

占空比决定 PWM 输出必须保持有效状态的时间周期。每个占空比寄存器都允许指定一个 16 位的占空比值。通过将立即更新使能位 IUE （PWMCONx<0>）置 1，可以在任意时刻更新占空比值。如果 IUE 位为 0，则所用占空比寄存器（PDCx、SDCx 或 MDC）在下一个 PWM 周期开始时更新。

通过主控占空比寄存器 （MDC），多个 PWM 发生器可以共用一个公共的占空比寄存器。 MDC 寄存器在主控时基模式下具有重要的作用。

此外，每个 PWM 发生器还具有主占空比寄存器（PDCx）和辅助占空比寄存器（SDCx），用于为每个 PWM 提供独立的占空比。

6.2.1 主控占空比 （MDC）

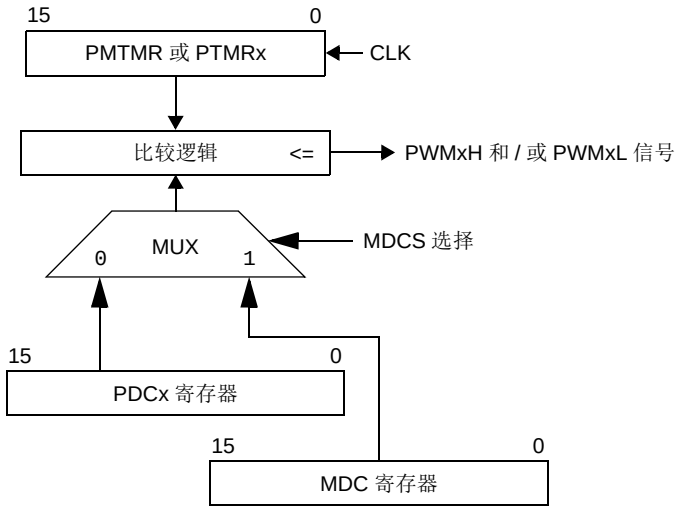
MDC 寄存器可用于向多个 PWM 发生器提供相同的占空比。MDC 寄存器可以在任意 PWM 模式（主控或独立时基）下使用。主控占空比寄存器选择位 MDCS （PWMCONx<8>）决定每个 PWMxH 和 PWMxL 输出的占空比是由 PWM MDC 寄存器还是 PDCx 和 SDCx 寄存器控制。

通过 MDC 寄存器，多个 PWM 发生器之间可以共用公共的占空比寄存器，并节省更新多个占空比寄存器所需的 CPU 开销。

6.2.2 主占空比 （PDCx）

PDCx 寄存器可用于为每个 PWM 发生器产生占空比。在互补、冗余或推挽 PWM 模式下，PDCx 寄存器为 PWMxH 和 PWMxL 输出提供占空比。在独立输出模式下，PDCx 寄存器仅为 PWMxH 输出提供占空比。图 6-1 显示了主占空比比较。

图 6-1： 主占空比比较

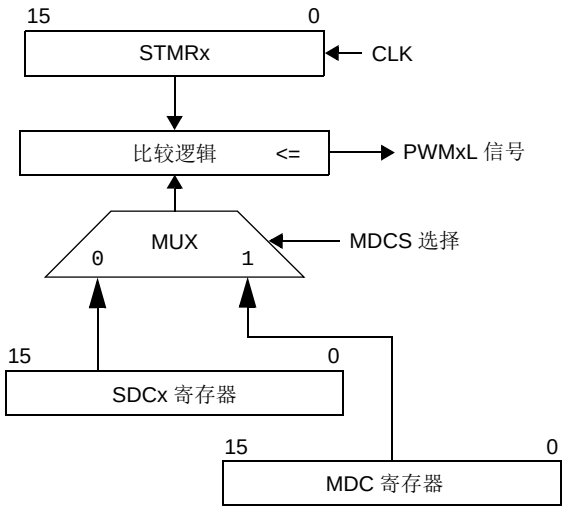


注：在独立输出模式下，PDCx 只会影响 PWMxH。

6.2.3 辅助占空比 （SDCx）

SDCx 寄存器仅在独立输出模式下使用。它在互补、冗余和推挽模式下被忽略。在独立输出模式下，SDCx 寄存器是为辅助 PWM 输出（PWMxL）信号提供占空比值的输入寄存器。图 6-2 显示了辅助占空比比较。

图 6-2： 辅助占空比比较



注： 在独立输出模式下，SDCx 只会影响 PWMxL； SDCx 在所有其他 PWM 输出模式下被忽略。

© 2011-2014 Microchip Technology Inc.

可通过公式 6-3 确定占空比。

公式 6-3：　　MDC、 PDCx 和 SDCx 计算

$$MDC, PDCx, SDCx = \left(\frac{ACLK \times 8 \times 所需的\ PWM\ 占空比}{PWM\ 输入时钟预分频比\ (PCLKDIV<2:0>)} \right)$$

$$ACLK = \frac{REFCLK \times M1}{N}$$ 请参见公式 5-1

（或）

$$ACLK = \frac{FVCO}{N}$$ 请参见公式 5-1

$$ACLK = \frac{7.49\ MHz \times 16}{1} = 119.84\ MHz$$

其中：

REFCLK = 7.49 MHz

*M*1 = 16

N = 1

$$MDC, PDCx, SDCx = \left(\frac{119.84\ MHz \times 8 \times 5\ \mu s}{1} \right) = 4794$$

其中：

最大 PWM 占空比分辨率为 1.04 ns。

所需的 PWM 占空比为 5 µs。

注：　可以使用 OSCTUN 特殊功能寄存器（SFR）的 TUN<5:0> 位来微调 FRC 时钟频率，以获得最高 PWM 分辨率 1.04 ns。更多信息，请参见具体器件数据手册中的 **“振荡器配置”**。

注　1： 如果占空比值小于最小值 （0x0008），则信号的占空比将为 0。值 0x0008 是可以让 PWM 发生器产生输出脉冲的最小可用占空比值。

2： 占空比值大于 （周期 + 0x0008）时，产生的占空比为 100%。

根据公式 6-3，在使用主控占空比、独立主占空比或独立辅助占空比时，寄存器值将分别装入 MDC、PDCx或SDCx寄存器。例6-4显示了PWM占空比选择。例6-5显示了PWM占空比初始化。

例 6-4：　　PWM 占空比选择

```
/* Select either Master Duty cycle or Independent Duty cycle */

PWMCON1bits.MDCS = 0; /* PDCx/SDCx provides duty cycle value */
PWMCON1bits.MDCS = 1; /* MDC provides duty cycle value */
```

例 6-5：　　PWM 占空比初始化

```
/* Initialize PWM Duty cycle value */

PDC1 = 4794;   /* Independent Primary Duty Cycle is 5 µs from Equation 6-3 */
SDC1 = 4794;   /* Independent Secondary Duty Cycle is 5 µs from Equation 6-3 */
MDC  = 4794;   /* Master Duty Cycle is 5 µs from Equation 6-3 */
```

6.2.4 占空比分辨率

当 ACLK = 120 MHz、为 PWM 时钟配置最高的预分频比设置时， PWM 占空比和周期分辨率为每个 LSb 对应 1.04 ns。可通过公式 6-4 确定 PWM 占空比的位分辨率。

公式 6-4： 位分辨率计算

$$\text{位分辨率} = \log_2 \times \left(\frac{\text{ACLK} \times 8 \times \text{所需的 PWM 周期}}{\text{PWM 输入时钟预分频比 （PCLKDIV<2:0>）}} \right)$$

其中：

$$\text{所需的 PWM 周期} = \left(\frac{1}{\text{所需的 PWM 开关频率}} \right)$$

表 6-1 列出了在最高 PWM 时钟频率下，占空比位分辨率与 PWM 频率的关系。

表 6-1： PWM 频率和占空比分辨率

PWM 占空比分辨率	PWM 频率
16 位	14.6 kHz
15 位	29.3 kHz
14 位	58.6 kHz
13 位	117.2 kHz
12 位	234.4 kHz
11 位	468.9 kHz
10 位	937.9 kHz
9 位	1.87 MHz
8 位	3.75 MHz

在最高时钟频率下，时钟周期为 1.04 ns。通过配置其他 PWM 时钟预分频比设置，可以降低 PWM 分辨率。

© 2011-2014 Microchip Technology Inc.

6.3 死区发生

死区指由死区寄存器（DTRx）或备用死区寄存器（ALTDTRx）指定的可编程时间段，死区功能会阻止一个 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。

高速 PWM 模块含有 4 个死区控制单元。每个死区控制单元都有各自的死区值。

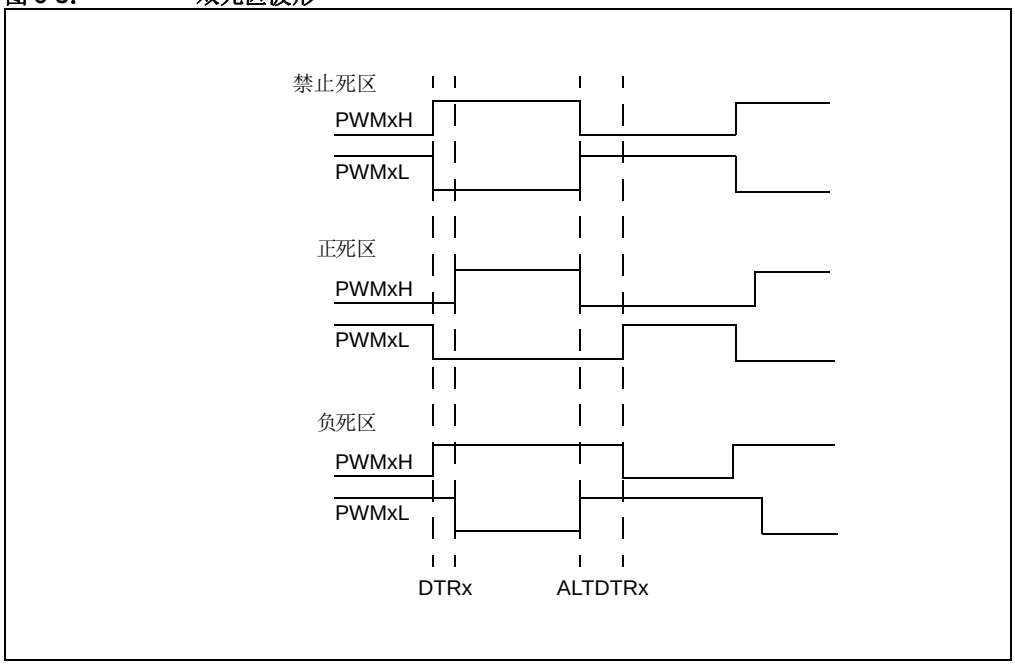
当任一 PWM I/O 引脚对工作于互补 PWM 输出模式时，都可提供死区发生。由于功率晶体管不能瞬间完成开关，因此许多电源转换器电路都需要死区。为了防止电流直通，必须在一个 PWM 输出的关断事件和互补对中另一个 PWM 输出的导通事件之间，或者和另一个晶体管的导通事件之间提供一定的时间。

高速 PWM 模块提供正死区和负死区。正死区用于防止 PWM 输出发生重叠。正死区发生可用于所有输出模式。正死区电路的工作方式是对 PWM 信号前沿进行消隐。负死区用于强制 PWMxH 和 PWMxL 信号发生重叠。负死区的工作方式是延长当前有效 PWM 输出，使延长时间段与刚刚置为有效的 PWM 输出发生重叠。一些转换器技术要求产生一定限度的直通电流。

只有对于互补 PWM 信号，才能指定负死区。负死区不适用于用户改写、限流改写或故障改写。该模式可以通过以下方式实现：使用 PHASEx/SPHASEx 寄存器中的相移值对 PWM 输出进行相移，使输出与来自不同 PMW 输出通道的另一个 PWM 信号重叠。

死区逻辑用作一个门控，让置为有效的 PWM 信号或改写值可以传送到输出上。死区逻辑永远不会主动将 PWM 输出置为有效。图 6-3 给出了禁止死区、正死区和负死区的双死区波形。

图 6-3： 双死区波形



对于每个PWM发生器，可以禁止死区功能。死区功能通过死区控制位DTC<1:0>（PWMCONx<7:6>）进行控制。独立 PWM 输出模式不支持死区。

6.4 死区发生器

高速 PWM 模块的每个互补输出对都有一个 12 位的递减计数器，用于插入死区。每个死区单元都有与占空比比较输出相连的上升沿和下降沿检测器。根据边沿是上升沿还是下降沿，互补输出中的一个电平跳变会被延迟，直到关联的死区定时器产生特定延时周期为止。

死区逻辑会监视 PWM 信号的上升沿和下降沿。当关联的 PWM 信号变为无效时，死区计数器会复位，当 PWM 信号变为有效时，它会开始计数。提供 PWM 输出信号的任意选定信号源都由死区逻辑处理。

死区可以使用公式 6-5 中给出的公式确定：

公式 6-5： 死区计算

$$DTRx, ALTDxTR = \frac{ACLK * 8 * \text{所需的死区}}{\text{PWM 输入时钟预分频比 (PCLKDIV<2:0>)}}$$

注： 最大死区分辨率为 1.04 ns。

例 6-6：

$$ACLK = \left[\frac{7.49\ MHz * 16}{1}\right] = 119.84\ MHz$$ （请参见公式 5-1）

其中：

REFCLK = 7.49 MHz
*M*1 = 16
N = 1
所需的死区 = 100 ns

$$DTRx, ALTDTRx = \left[\frac{119.85\ MHz * 8 * 100\ ns}{1}\right] = 96$$

以下是三种死区控制模式：

- **正死区模式**

  正死区模式定义 PWMxH 和 PWMxL 输出均置为无效的时间段。该模式在以下情况下很有用：应用必须先分配一定时间来禁止一个功率晶体管，这段时间之后再使能其他晶体管。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 输出的正死区， ALTDTRx 寄存器指定 PWMxL 输出的正死区。

- **负死区模式**

  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拓扑中，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 PWMxL 输出的负死区，ALTDTRx 寄存器指定 PWMxH 输出的负死区。只有对于互补 PWM 输出信号，才能指定负死区。

- **死区禁止模式**

  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。

© 2011-2014 Microchip Technology Inc.

6.5 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。在最大工作时钟频率下，占空比分辨率为 1.04 ns，死区分辨率也为 1.04 ns。在最高 PWM 分辨率下，最大死区值为 17.03 µs。

6.6 死区失真

占空比值接近 0% 或 100% 时，如果死区有效，PWM 信号会变为非线性。占空比值小于死区时，PWM 输出为 0。占空比值大于（100% 死区）时，PWM 输出与占空比为（100% 死区）时相同。

6.7 死区分辨率

在最高时钟速率下，正常工作条件下的死区分辨率为 1.04 ns。但存在一些例外情况，例如在最大 CPU 速度和预分频比的情况下，对于故障改写、限流改写或用户改写事件，可能的最高死区分辨率为 8.32 ns （DTRx 和 ALTDTRx 寄存器中的 bit 3）。

注：	当限流或故障改写数据设置为 0 时，将不应用死区，而是立即应用“0”改写数据。

例 6-7 给出了 PWM 死区控制的配置。例 6-8 给出了 PWM 死区初始化的配置。

例 6-7： PWM 死区控制

```
/* Select Dead-Time control */
/* Choose one of these options */

PWMCON1bits.DTC = 0;   /* Positive Dead-Time applied for all modes */
PWMCON1bits.DTC = 1;   /* Negative Dead-Time applied for all modes */
```

例 6-8： PWM 死区初始化

```
/* Dead-Time value for PWM generator */
/* Refer to Equation 6-5 */
DTR1 = 96;     /* Dead-Time value is 100 ns */
ALTDTR1 = 96;  /* Alternate Dead-Time value is 100 ns */
```

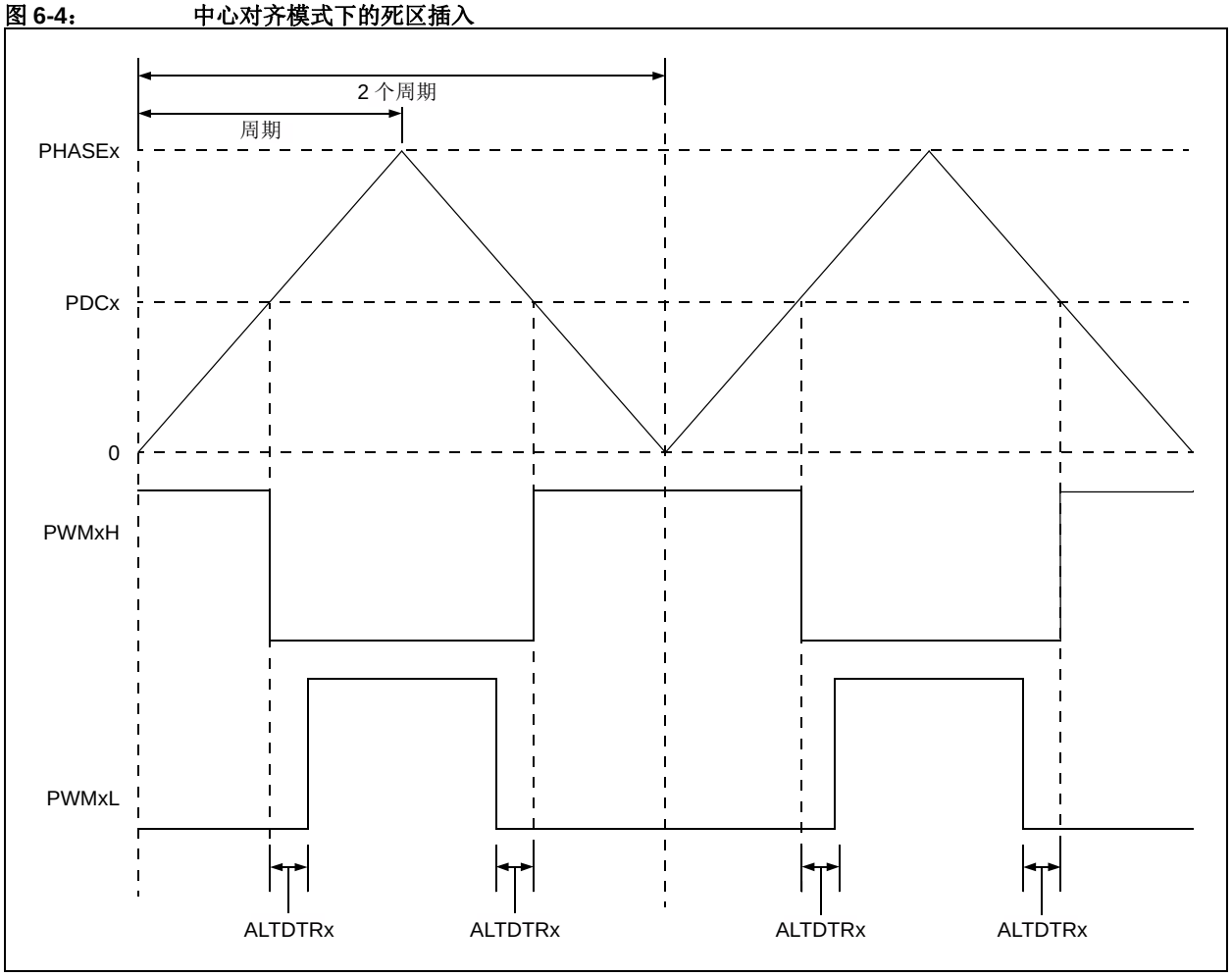
注：	由于占空比值大于（100% 死区），且应用要求占空比为 100%（即， PWM 输出中无死区）；因此，在 PWMCONx 寄存器中配置 DTC<1:0> = 2。

6.8 中心对齐模式下的死区插入

在使用中心对齐模式的互补 PWM 时，必须仅使用 ALTDTRx 寄存器来插入死区。如图 6-4 所示，PWM 波形中插入了死区。

注： IUE = 1 时，全部三种情形（如**第 13.0 节 “PWM 占空比立即更新”**所述）在中心对齐模式下均适用。

图 6-4： 中心对齐模式下的死区插入



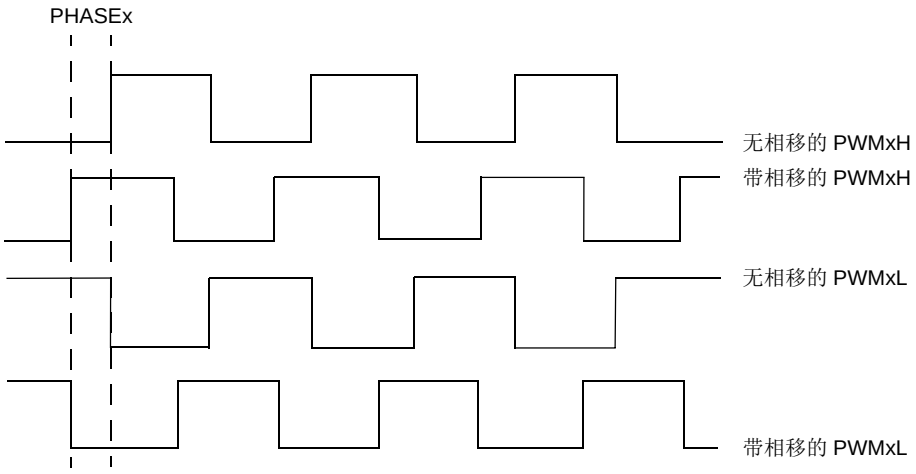
© 2011-2014 Microchip Technology Inc.

6.9 相移

相移是 PWMxH 或 PWMxL 与主控时基之间的相对偏移。在独立输出模式下，PHASEx 寄存器决定 PWMxH 和主控时基之间的相对相移。SPHASEx 寄存器决定 PWMxL 和主控时基之间的相对相移。 PHASEx 寄存器的内容用作 PTMRx 寄存器的初始化值， SPHASEx 寄存器的内容用作 STMRx 寄存器的初始化值。

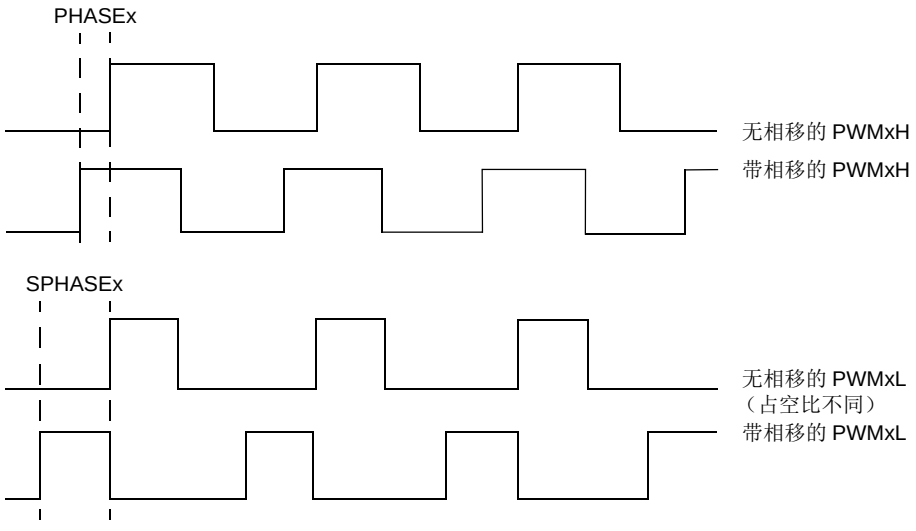
图 6-5 和图 6-6 分别给出了互补模式和独立输出模式下相移的示例波形。

图 6-5： 相移 （互补模式）



注： 在互补、推挽和冗余 PWM 输出模式下， PHASEx 控制 PWMxH 和 PWMxL 输出的相移。

图 6-6： 相移 （独立输出模式）



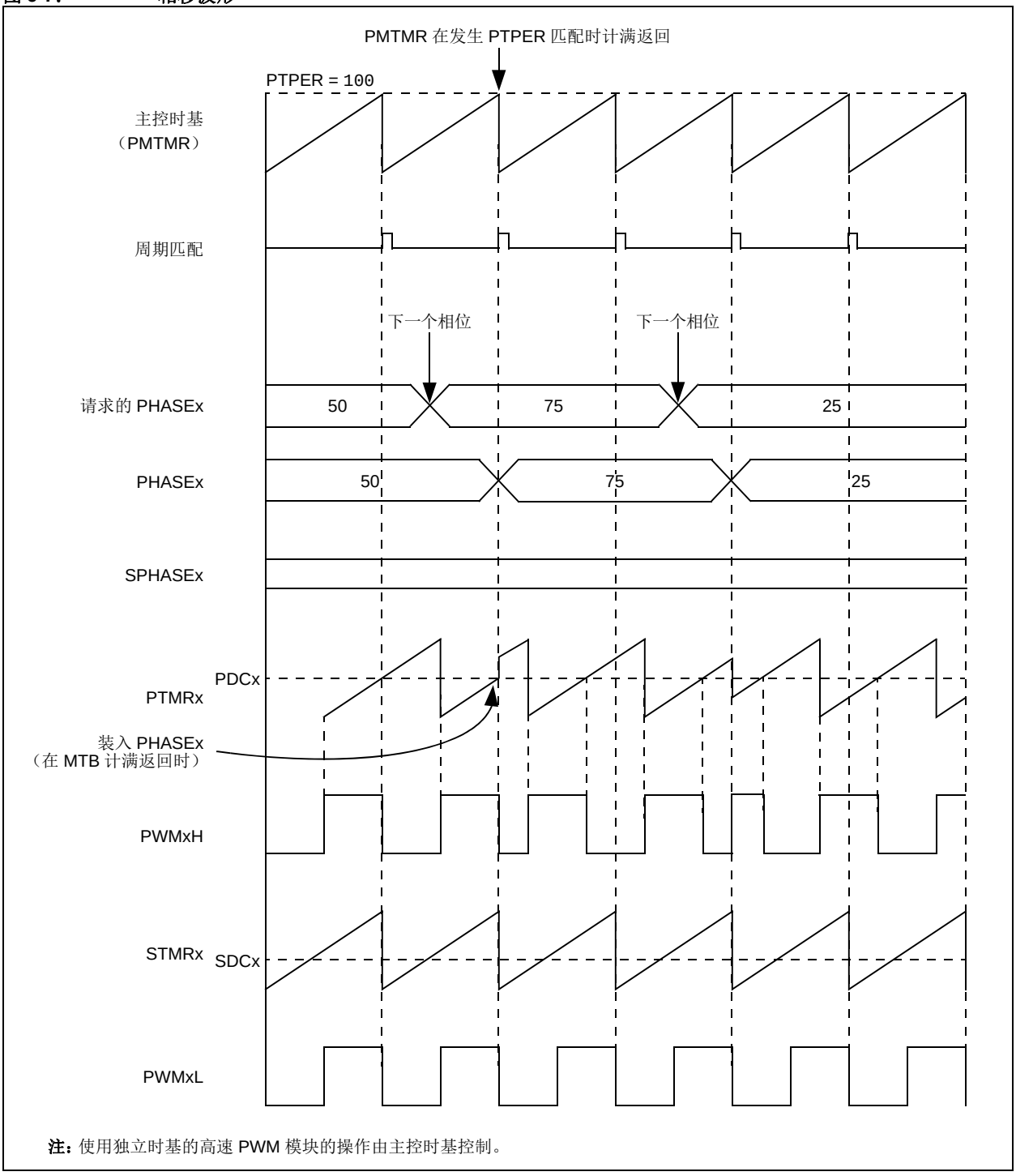
注： 在独立输出模式下， SPHASEx 控制 PWMxL 输出的相移， PHASEx 控制 PWMxH 输出的相移。

此外，还有两个对应于 PHASEx 和 SPHASEx 寄存器的影子寄存器，它们在每次用户应用程序写入新值时更新。在发生主控时基复位时，这些值从影子寄存器传送到 PHASEx 和 SPHASEx 寄存器。对 PWM 输出实际应用这些相位偏移将在发生主控时基复位时进行。图 6-7 给出了说明如何产生这些事件的时序图。

相位偏移值可以为 0 和 PTPER 寄存器值之间的任意值。PHASEx 或 SPHASEx 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。例 6-9 显示了 PWM 相移初始化。

© 2011-2014 Microchip Technology Inc.

图 6-7：　　相移波形



注：使用独立时基的高速 PWM 模块的操作由主控时基控制。

例 6-9：　　PWM 相移初始化

```
/* Initialize phase shift value for the PWM output */
/* Phase shifts are initialized when operating in Master time base */

PHASEx  = 100;  /* Primary phase shift value of 104 ns */

SPHASEx = 100;  /* Secondary phase shift value of 104 ns */
```

表 6-2 中列出了对应于不同输入时钟预分频比选择的 PWM 占空比、相位和死区的位分辨率。

表 6-2： 占空比、相位、死区位分辨率与预分频比选择

PWM 时钟预分频比	位分辨率						
	64 ns	32 ns	16 ns	8 ns	4 ns	2 ns	1 ns
1:1	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
1:2	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	—
1:4	bit 4	bit 3	bit 2	bit 1	bit 0	—	—
1:8	bit 3	bit 2	bit 1	bit 0	—	—	—
1:16	bit 2	bit 1	bit 0	—	—	—	—
1:32	bit 1	bit 0	—	—	—	—	—
1:64	bit 0	—	—	—	—	—	—

© 2011-2014 Microchip Technology Inc.

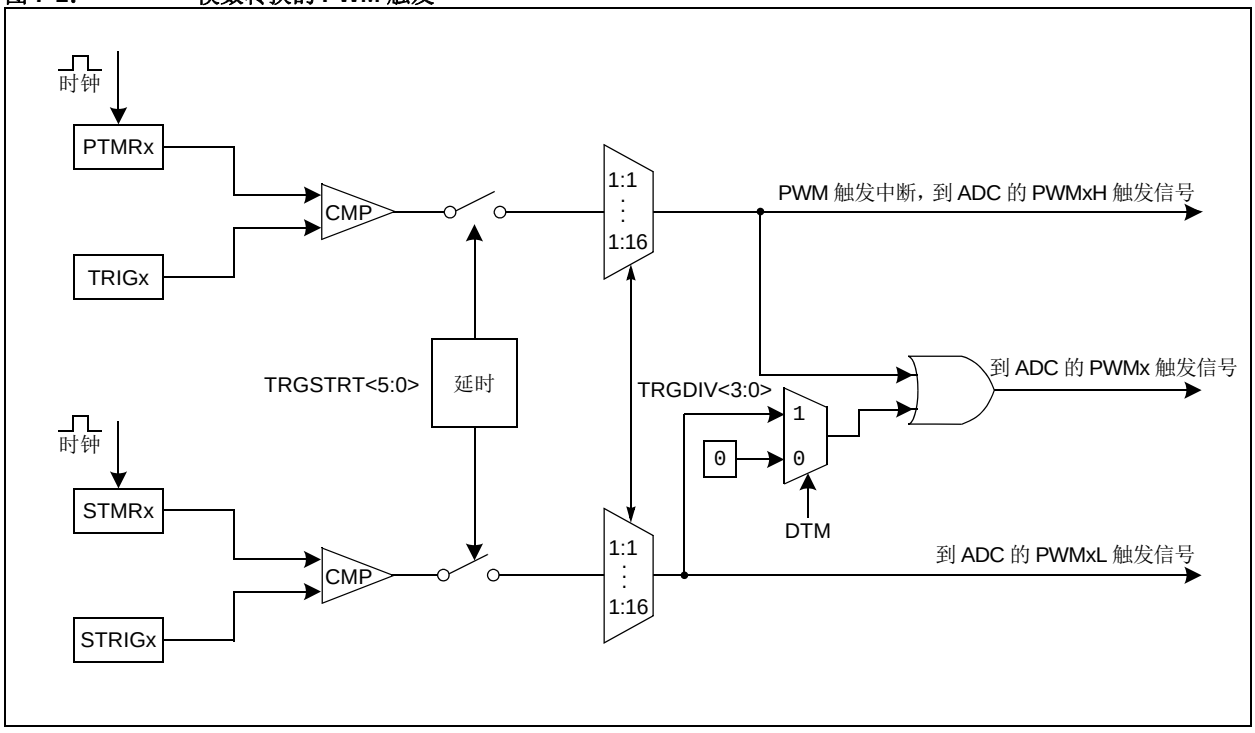
7.0 PWM 触发

对于 ADC 模块，TRIGx 和 STRIGx 寄存器分别指定 PWMxH 和 PWMxL 输出的触发点。当独立时基计数器 （PTMRx 或 STMRx）寄存器值与指定的 TRIGx 或 STRIGx 寄存器值匹配时，将会产生 ADC 触发信号。PWM 触发（TRIGx/STRIGx）的分辨率为 8.32 ns（PWM 分辨率为 1.04 ns 时）。除了由 TRIGx 和 STRIGx 设置产生的触发信号之外，还可以通过各个 PWM 发生器的限流源和特殊事件触发器 （SEVTCMP）来触发 ADC 对。

PWMx 触发控制寄存器中的触发器 # 输出分频比位 TRGDIV<3:0> （TRGCONx<15:12>）用作 TRIGx 寄存器产生 ADC 触发信号的后分频比。这使 ADC 的触发信号可以每隔 1、2、3.... 和 16 个触发事件产生一次。这些位指定产生 ADC 触发信号的频率。

每个 PWM 发生器都含有触发后分频比开始使能选择位 TRGSTRT<5:0> （TRGCONx <5:0>），它们指定在产生第一个 ADC 触发信号之前要等待多少个 PWM 周期。图 7-1 显示了通过高速 PWM 模块触发 ADC 的逻辑。

图 7-1： 模数转换的 PWM 触发



根据 TRGDIV<3:0> 位 （TRGCONx<15:12>）和 TRGSTRT<5:0> 位 （TRGCONx<5:0>）的设置，将以不同的PWM时间间隔产生触发信号，如图 7-2 至图 7-9 所示。触发开始延时（TRGSTRT<5:0>）与主主控定时器和辅助主控定时器的计满返回事件进行同步。因此，需要使用独立时基 （ITB = 1）的应用 （例如，使用中心对齐模式 （CAM = 15）的应用）可能必须根据配置 TRGSTRT<5:0> 的要求来配置 PTPER/STPER 寄存器。

图 7-2： PWM 触发信号与 PWM 输出的关系 （TRGDIV<3:0> = 0000， TRGSTRT<5:0> = 000000）

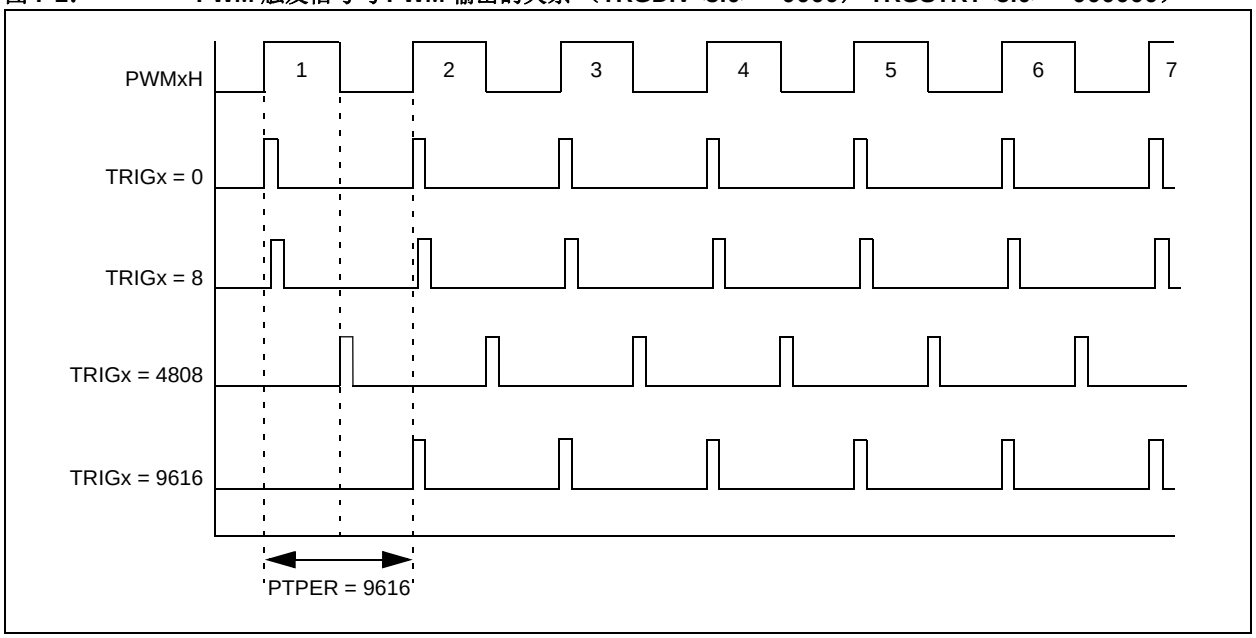
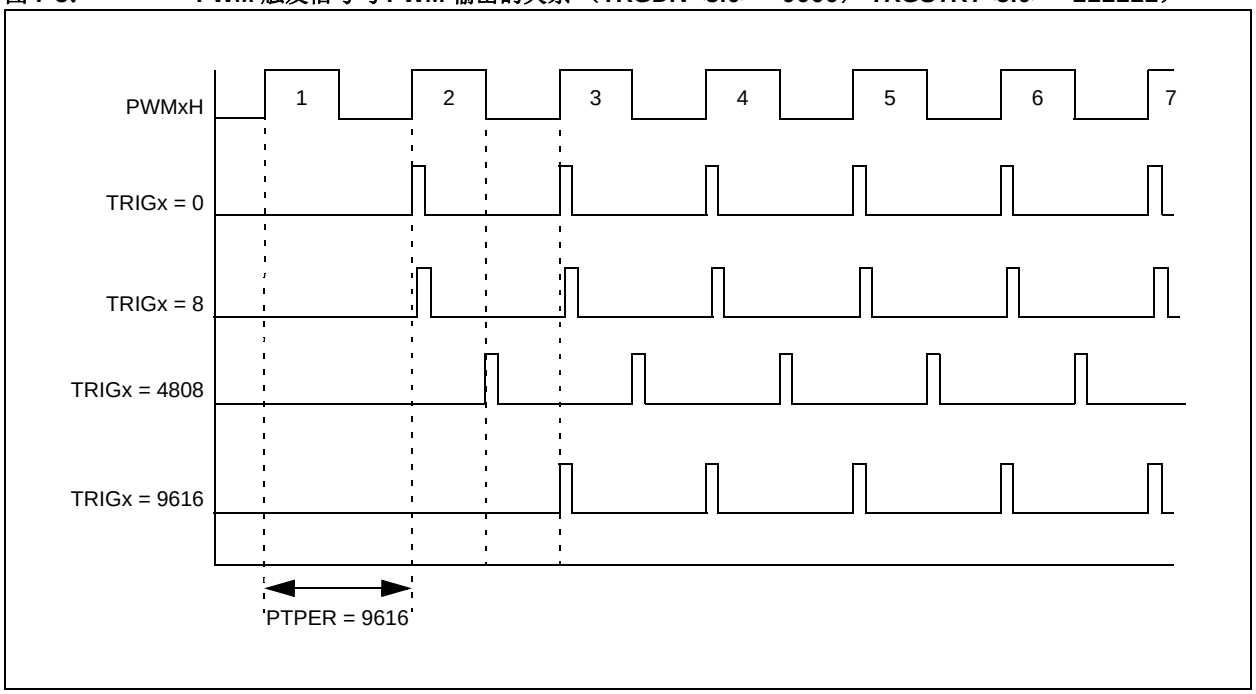


图 7-3： PWM 触发信号与 PWM 输出的关系 （TRGDIV<3:0> = 0000， TRGSTRT<5:0> = 111111）



© 2011-2014 Microchip Technology Inc.

图 7-4：　　PWM 触发信号与 PWM 输出的关系 （TRGDIV<3:0> = 0000， TRGSTRT<5:0> = 000010）

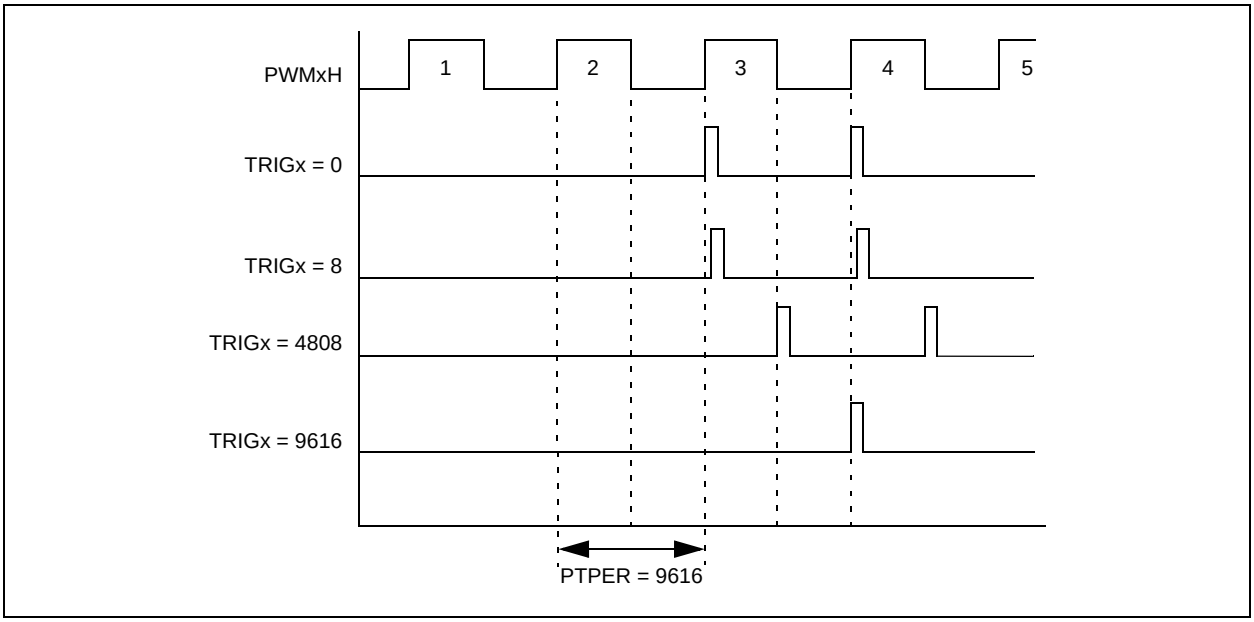


图 7-5：　　PWM 触发信号与 PWM 输出的关系 （TRGDIV<3:0> = 1111， TRGSTRT<5:0> = 000000）

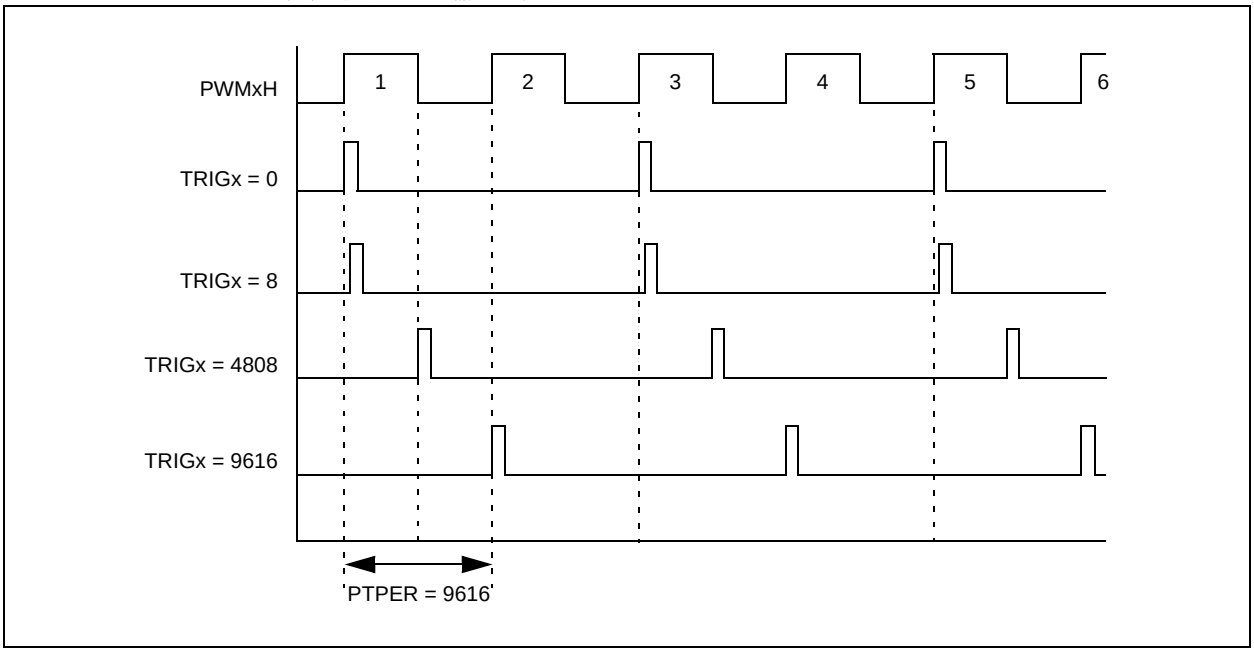


图 7-6：　　PWM 触发信号与 PWM 输出的关系 （TRGDIV<3:0> = 1111， TRGSTRT<5:0> = 111111）

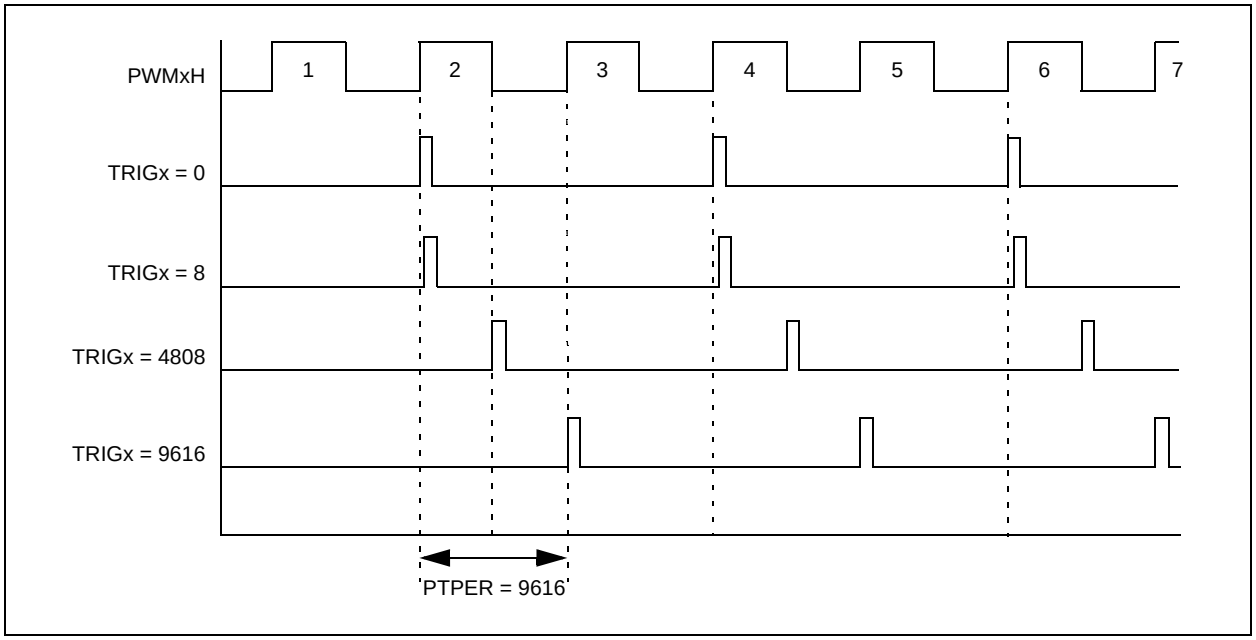
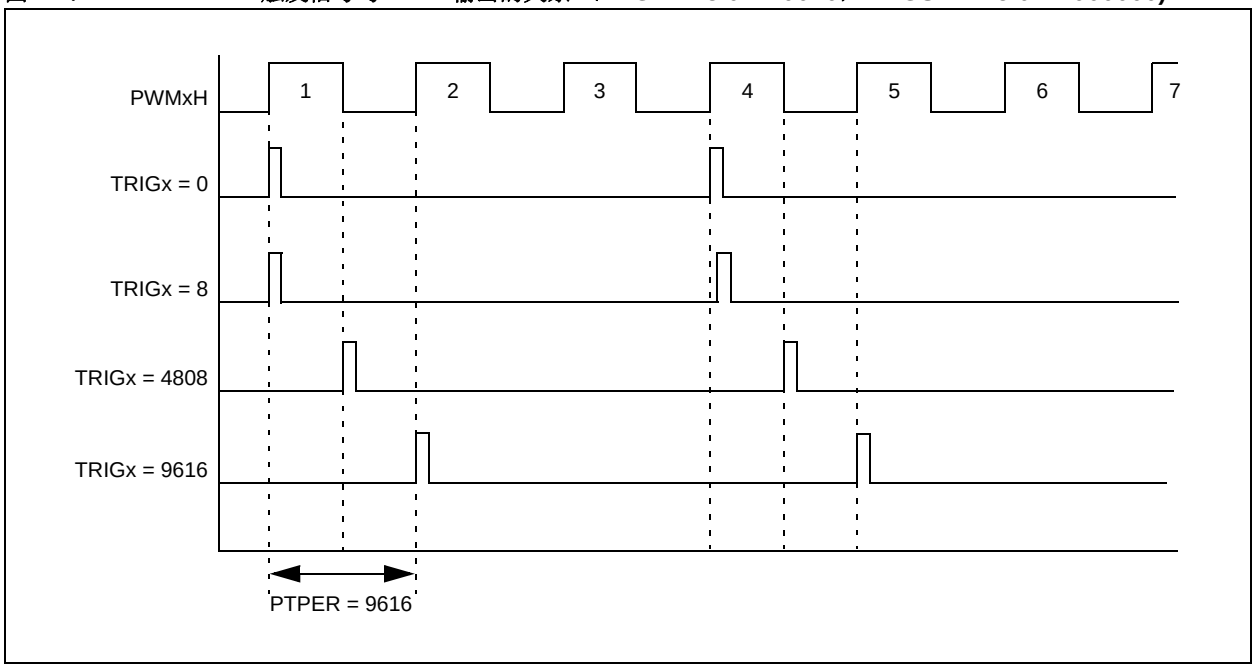


图 7-7：　　PWM 触发信号与 PWM 输出的关系 （TRGDIV<3:0> = 0010， TRGSTRT<5:0> = 000000)



© 2011-2014 Microchip Technology Inc.

图 7-8：　　PWM 触发信号与 PWM 输出的关系 （TRGDIV<3:0> = 0010， TRGSTRT<5:0> = 000010）

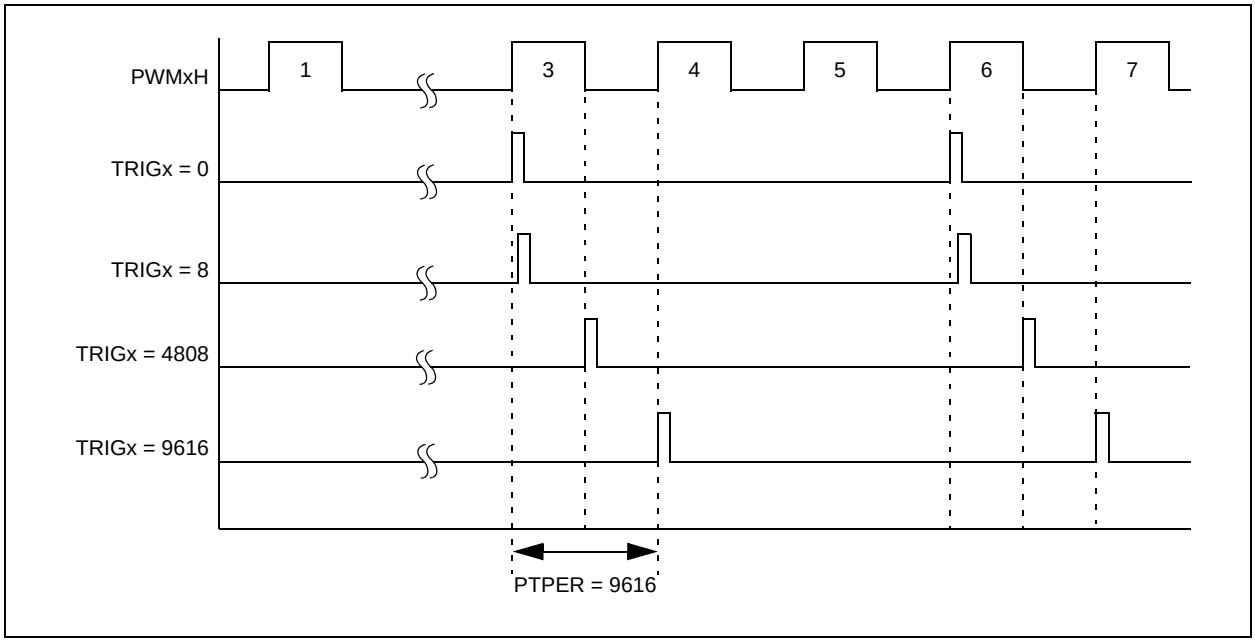
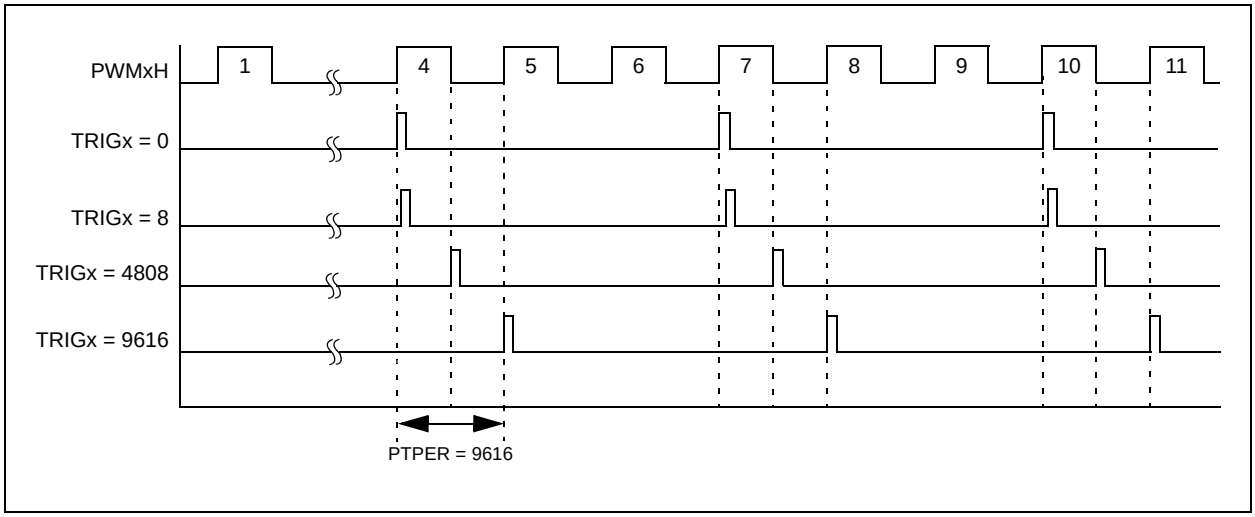


图 7-9：　　PWM 触发信号与 PWM 输出的关系 （TRGDIV<3:0> = 0010， TRGSTRT<5:0> = 000011）



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。

当双触发模式位 DTM（TRGCONx<7>）设置为 1 时，ADC TRIGx 输出为 TRIGx 和 STRIGx 时基比较产生的 ADC 触发脉冲的布尔或。

通过 DTM 工作模式，用户应用程序可以在单个 PWM 周期内在同一引脚上获取两个 ADC 采样。

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 周期内完成两次 ADC 操作。

触发脉冲的产生与触发中断允许位 TRGIEN （PWMCONx<10>）的状态无关。如果 TRGIEN 位设置为 1，则会产生中断请求 （IRQ）。例 7-1 给出了独立 PWM ADC 触发的配置。

注： 无论 DTM 位 （TRGCONx<7>）的状态如何，辅助触发 （STRIGx）比较都不会产生 PWM 中断。

例 7-1： 独立 PWM ADC 触发

```
/* Independent PWM ADC triggering */

TRIG1 = 1248;                       /* Point at which the ADC module is to be
                                       triggered by primary PWM */
STRIG1 = 2496;                      /* Point at which the ADC module is to be
                                       triggered by secondary PWM */
TRGCON1bits.TRGDIV = 0;             /* Trigger output divider set to trigger
                                       ADC on every trigger match event */
TRGCON1bits.DTM = 1;                /* Primary and Secondary triggers combined
                                       to create ADC trigger */
TRGCON1bits.TRGSTRT = 4;            /* First ADC trigger event occurs after
                                       four trigger match events */
PWMCON1bits.TRGIEN = 1;             /* Trigger event generates an interrupt
                                       request */
while (PWMCON1bits.TRGSTAT!= 1);    /* Wait for PWM trigger interrupt status
                                       change */
```

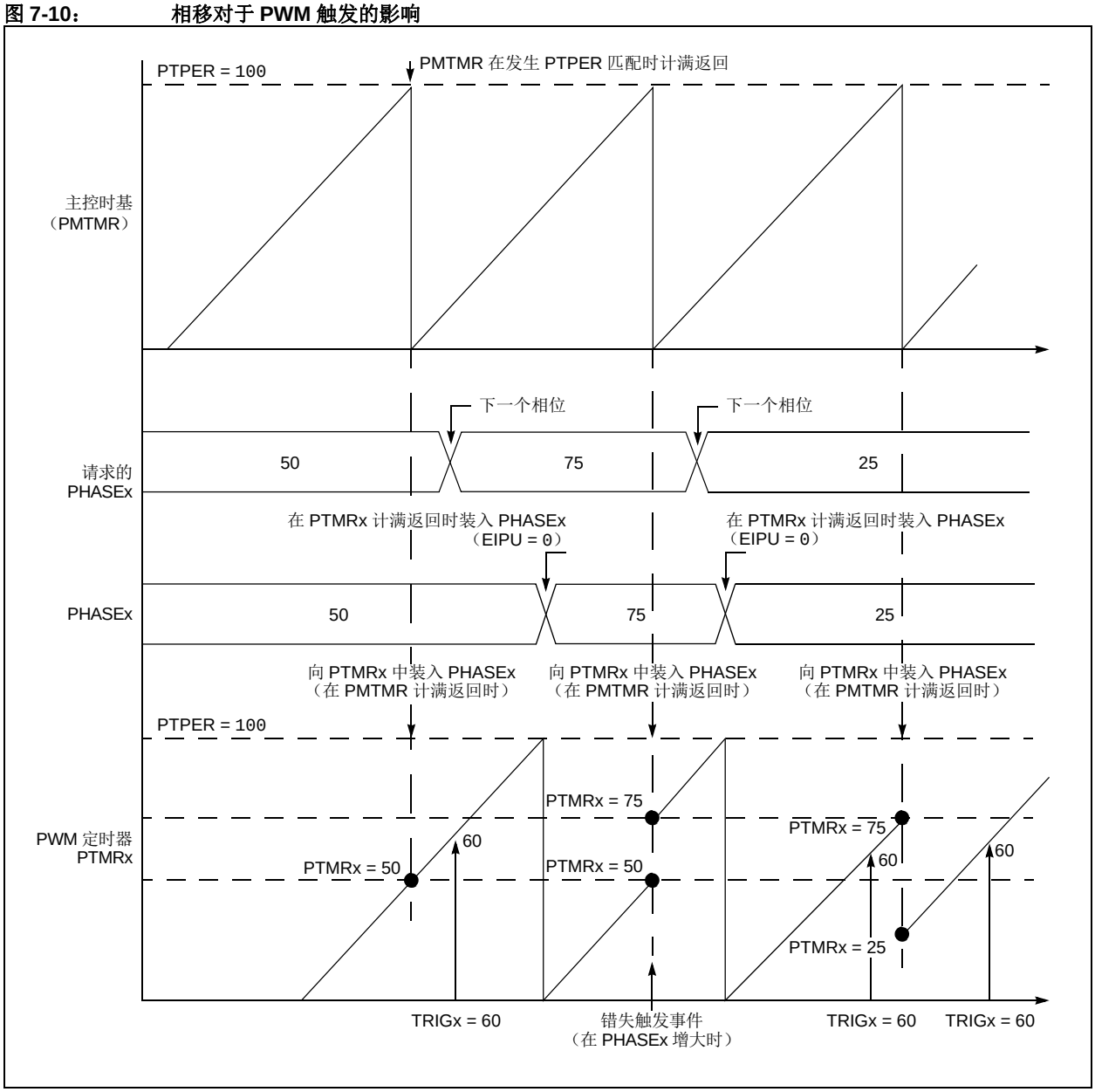
注 1： TRGSTAT 位只能通过清零 TRGIEN 位 （PWMCONx<10>）来清零；它不会自动清零。

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。TRIGx 值可以基于 PWM 周期、占空比和负载电流等参数进行更改。这可用于确保触发点与 PWM 通道的上升和下降时刻相独立。

© 2011-2014 Microchip Technology Inc.

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。在将相位从较小值更改为较大值时，有可能会错失触发事件。用户应用程序必须确保这不会影响任何控制环的执行。图 7-10 说明了相移对于 PWM 触发的影响。

图 7-10： 相移对于 PWM 触发的影响



8.0 PWM 中断

高速 PWM 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。主时基模块可以在发生指定事件时产生中断请求 （IRQ）。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 发生器的中断是模块的触发事件 IRQ、限流输入事件和故障输入事件的布尔或。

除了来自每个 PWM 发生器的各个 PWM IRQ 之外，中断控制器还会在发生特殊事件时接收来自主时基的 IRQ 信号。

来自每个 PWM 发生器的三种 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 故障逻辑或 PWM 限流逻辑。每个 PWM 发生器在 IFSx 寄存器中都含有 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触发中断状态位 TRGSTAT（PWMCONx<13>）、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。

8.1 PWM 时基中断

在每个 PWM 发生器中，高速 PWM 模块可以基于主控时基和 / 或独立时基产生中断。SEVTCMP 寄存器用于为主时基指定基于定时器的中断，TRIGx 寄存器用于为独立时基指定基于定时器的中断。对于具有辅助主控时基的器件， SSEVTCMP 寄存器配置为基于与辅助时基的比较事件来产生中断。

主时基和辅助时基（对于具有辅助主控时基的器件）特殊事件中断分别通过 SEIEN 位 PTCON<11> 和 STCON<11> 允许。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<10>）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。

© 2011-2014 Microchip Technology Inc.

9.0 PWM 工作模式

本节介绍高速 PWM 模块支持的以下工作模式。

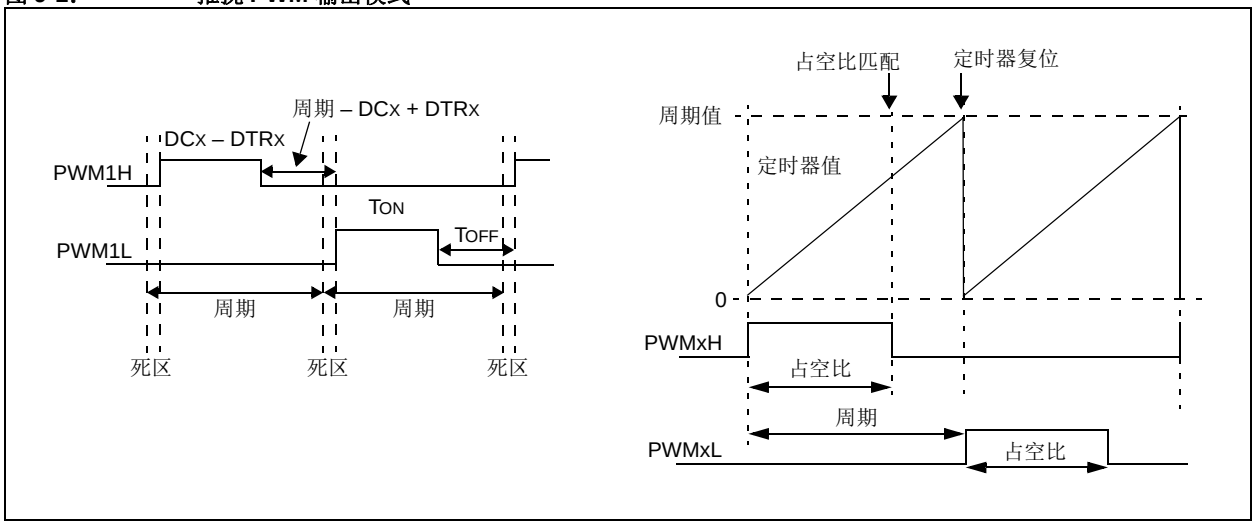
- 推挽 PWM 输出模式
- 互补 PWM 输出模式
- 冗余 PWM 输出模式
- 真正独立 PWM 输出模式

这些工作模式可以使用 PWM I/O 控制寄存器中的 PWM # I/O 引脚模式位 PMOD<1:0>（IOCONx<11:10>）进行选择。

9.1 推挽 PWM 输出模式

在推挽 PWM 输出模式下，PWM 输出在 PWMxH 和 PWMxL 引脚上交替提供。**第 16.0 节 “应用信息”** 中提供了推挽模式的一些典型应用。图 9-1 显示了推挽 PWM 模式下的 PWM 输出。

图 9-1： 推挽 PWM 输出模式

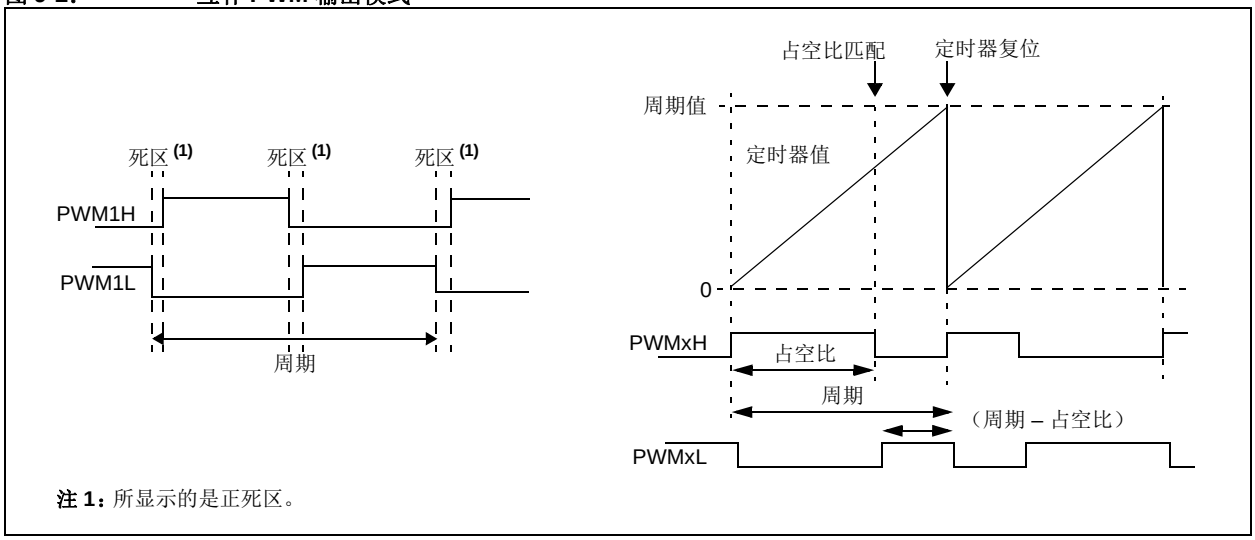


9.2 互补 PWM 输出模式

在互补 PWM 输出模式下， PWM 输出 PWMxL 是 PWMxH 输出的互补信号。**第 16.0 节 “应用信息”** 中提供了互补 PWM 输出模式的一些典型应用。

图 9-2 显示了在模块工作于互补 PWM 输出模式时的 PWM 输出。

图 9-2： 互补 PWM 输出模式

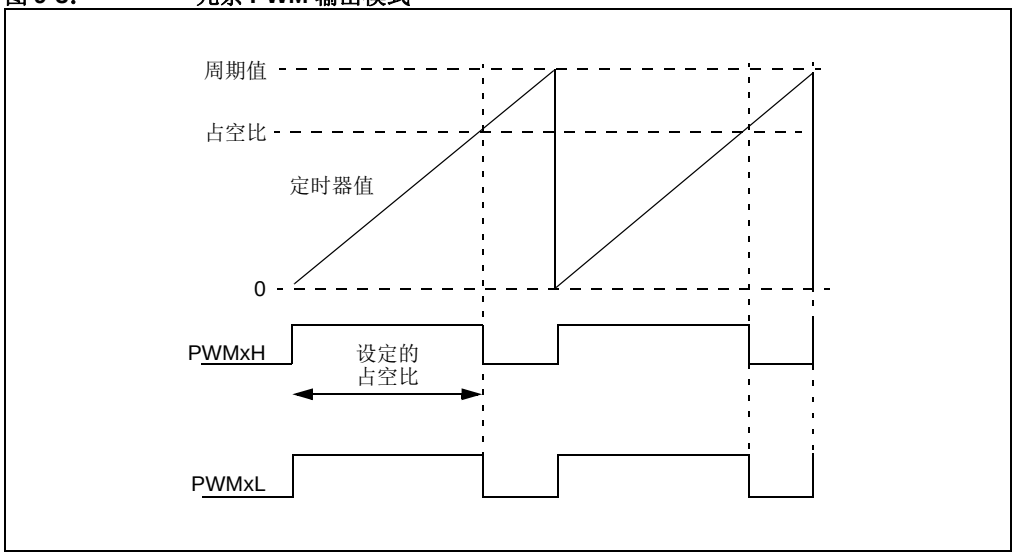


注 1： 所显示的是正死区。

9.3 冗余 PWM 输出模式

在冗余 PWM 输出模式下，对于每个 PWM 引脚对（PWMxH 和 PWMxL），高速 PWM 模块可以提供单端PWM输出信号的两个副本。该模式使用PDCx寄存器来指定占空比。在该输出模式下，除非用户应用程序指定了改写值，否则两个 PWM 输出引脚会提供相同的 PWM 信号。图 9-3 显示了冗余 PWM 输出模式。

图 9-3： 冗余 PWM 输出模式



© 2011-2014 Microchip Technology Inc.

表 9-1 列出了对应于 PWM 模式的 PWM 寄存器功能。

表 9-1： 互补、推挽和冗余 PWM 输出模式的寄存器功能

时基	主主控时基		辅助主控时基 (1)		独立时基	
功能	PWMxH	PWMxL	PWMxH	PWMxL	PWMxH	PWMxL
PWM 周期	PTPER	PTPER	STPER	STPER	PHASEx	PHASEx
PWM 占空比	MDC/PDCx	MDC/PDCx	MCD/PDCx	MDC/PDCx	MDC/PDCx	MDC/PDCx
PWM 相移	PHASEx	PHASEx	PHASEx	PHASEx	N/A	N/A
ADC 触发	SEVTCMP/ TRIGx/ STRIGx	SEVTCMP/ TRIGx/ STRIGx	SSEVTCMP/ TRIGx/ STRIGx	SSEVTCMP/ TRIGx/ STRIGx	SEVTCMP(2)/ SSEVTCMP(2)/ TRIGx/STRIGx	SEVTCMP(2)/ SSEVTCMP(2)/ TRIGx/STRIGx

注 1： 关于辅助主控时基的可用性，请参见具体器件的数据手册。

2： 触发源选择为 SEVTCMP 还是 SSEVTCMP 取决于 MTBS （PWMCONx<3>）位设置。关于 MTBS 位的可用性，请参见具体器件的数据手册。

9.4 真正独立 PWM 输出模式

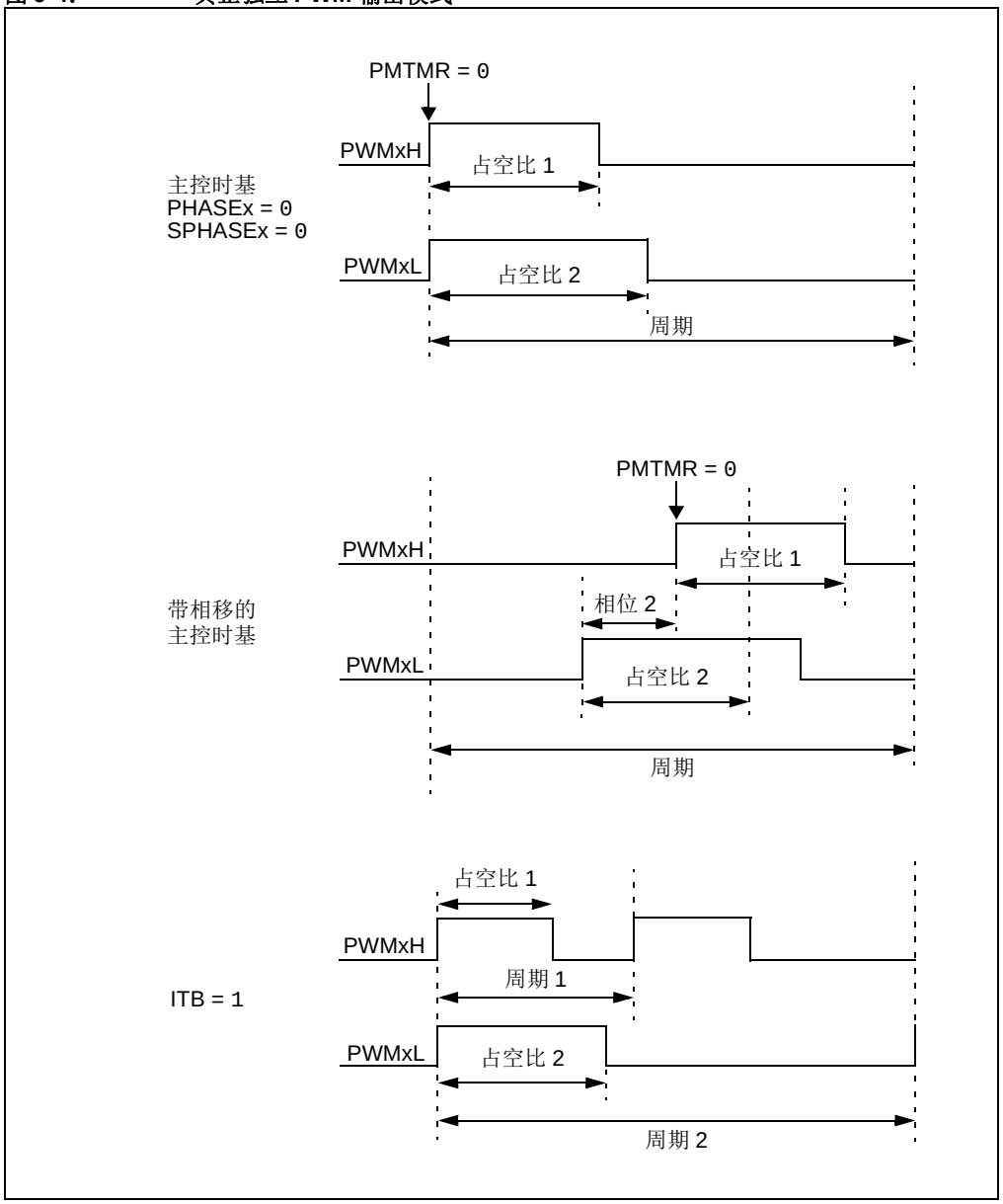
在真正独立 PWM 输出模式 （PMOD<1:0> = 11）下， PWM 输出 （PWMxH 和 PWMxL）可以具有不同的占空比。PDCx 寄存器指定 PWMxH 输出的占空比，而 SDCx 寄存器指定 PWMxL 输出的占空比。此外， PWMxH 和 PWMxL 输出还可以具有不同的周期，或者相互之间可以存在相对相移。

- 当 ITB = 1 时， PHASEx 寄存器指定 PWMxH 输出的 PWM 周期，而 SPHASEx 寄存器指定 PWMxL 输出的 PWM 周期
- 当 ITB = 0 时， PHASEx 寄存器指定 PWMxH 输出的相移，而 SPHASEx 寄存器指定 PWMxL 输出的相移

图 9-4 显示了独立 PWM 输出模式。例 9-1 显示了 PWM 输出引脚模式选择。

注： 在独立时基模式（ITB = 1）下， PWMxH 和 PWMxL 输出之间可能不存在确定的相位关系。

图 9-4：　　真正独立 PWM 输出模式



例 9-1：　　PWM 输出引脚模式选择

```
/* Select PWM I/O pin mode – Choose one of the following output modes */

IOCON1bits.PMOD = 0; /* For Complementary Output mode */

IOCON1bits.PMOD = 1; /* For Redundant Output mode */

IOCON1bits.PMOD = 2; /* For Push-Pull Output mode */

IOCON1bits.PMOD = 3; /* For True Independent Output mode */
```


© 2011-2014 Microchip Technology Inc.

表 9-2 列出了对应于独立输出模式的 PWM 寄存器功能。

表 9-2： 独立输出模式的寄存器功能

时基	主主控时基		辅助主控时基 (1)		独立时基	
功能	PWMxH	PWMxL	PWMxH	PWMxL	PWMxH	PWMxL
PWM 周期	PTPER	PTPER	STPER	STPER	PHASEx	SPHASEx(2)
PWM 占空比	MDC/PDCx	MDC/SDCx	MCD/PDCx	MDC/PDCx	MDC/PDCx	MDC/SDCx
PWM 相移	PHASEx	SPHASEx(2)	PHASEx	SPHASEx(2)	N/A	N/A
ADC 触发	SEVTCMP/ TRIGx/ STRIGx	SEVTCMP/ TRIGx/ STRIGx	SSEVTCMP/ TRIGx/ STRIGx	SSEVTCMP/ TRIGx/ STRIGx	TRIGx	STRIGx

注 1： 关于辅助主控时基的可用性，请参见具体器件的数据手册。
2： SPHASEx 寄存器仅在独立输出模式下使用。

表 9-3 列出了对应于不同拓扑和配置的 PMOD<1:0> 位选择。

表 9-3： 不同拓扑和配置的 PMOD<1:0> 位选择

项	拓扑 (1)	配置	PMOD<1:0> 设置
1	反激转换器	真正独立输出模式 / 冗余输出模式	11 或 01
2	升压 /PFC 转换器	真正独立输出模式 / 冗余输出模式	11 或 01
3	交错式 PFC 转换器	使用主控时基的真正独立输出模式	11
4	正激转换器	真正独立输出模式 / 冗余输出模式	11 或 01
5	双端正激转换器	真正独立输出模式 / 冗余输出模式	11 或 01
6	有源钳位正激转换器	互补 PWM 输出模式	00
7	LLC 半桥串联谐振转换器	互补 PWM 输出模式	00
8	半桥转换器	推挽 PWM 输出模式	10
9	推挽转换器	推挽 PWM 输出模式	10
10	全桥转换器	推挽 PWM 输出模式	10
11	相移全桥转换器	互补 PWM 输出模式	00
12	单相同步降压稳压器	互补 PWM 输出模式	00
13	多相同步降压稳压器	互补 PWM 输出模式，主控时基和相位在每个降压转换器 PWM 栅极驱动之间交错	00

注 1： 所列出的拓扑可以在电压和电流 （即，平均和峰值电流）模式控制下配置。

10.0 PWM 故障引脚

PWM 故障输入引脚的关键功能如下：

- 对于具有可重映射 I/O 的器件，每个 PWM 发生器可以从最多 8 个可重映射故障源中选择其故障输入源。一些器件具有专用的外部故障引脚和可重映射故障源。在一些具有可重映射 I/O 的器件中，模拟比较器的输出可直接用作故障源，而在其他器件中，可以通过虚拟引脚将模拟比较器输出指定为故障（见**第 10.1 节“模拟比较器产生的 PWM 故障信号”**）。关于可用故障源的更多信息，请参见具体器件的数据手册。
- 对于不具有可重映射 I/O 的器件，每个 PWM 发生器可以从最多 23 个故障引脚和最多 4 个模拟比较器输出中选择其故障输入源。
- 每个 PWM 发生器均在 PWM 故障限流控制寄存器中具有故障控制信号源选择位 FLTSRC<4:0>（FCLCONx<7:3>）。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 FLTIEN（PWMCONx<12>）。该位用于允许故障 IRQ 的产生。
- 每个 PWM 发生器都具有关联的故障极性位 FLTPOL（FCLCONx<2>）。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之一：
  - 如果使能了独立故障模式使能位 IFLTMOD（FCLCONx<15>），则 FLTDAT<1:0>（IOCONx<5:4>）位（高 / 低）将提供要分配给 PWMxH 和 PWMxL 输出的数据值。在该模式下，限流源将提供 PWMxH 引脚的故障输入，故障源将提供 PWMxL 引脚的故障输入， CLDAT<1:0> 位将被忽略。
  - 在故障模式下，FLTDAT<1:0>（IOCONx<5:4>）位（高 / 低）将提供要分配给 PWMxH 和 PWMxL 输出的数据值。

以下列表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。故障改写数据位 FLTDAT<1:0>（IOCONx<5:4>）的值可以为 00 或 11。如果 FLTDAT<1:0> 设置为 00，则会对故障进行异步处理，以便可以立即关闭应用电路中的相关功率晶体管。如果 FLTDAT<1:0> 位设置为 11，则死区逻辑会对它进行处理，然后再应用于 PWM 输出。
- 故障信号可以产生中断。FLTIEN 位（PWMCONx<12>）用于控制故障中断信号的产生。即使故障模式位 FLTMOD<1:0>（FCLCONx<1:0>）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。

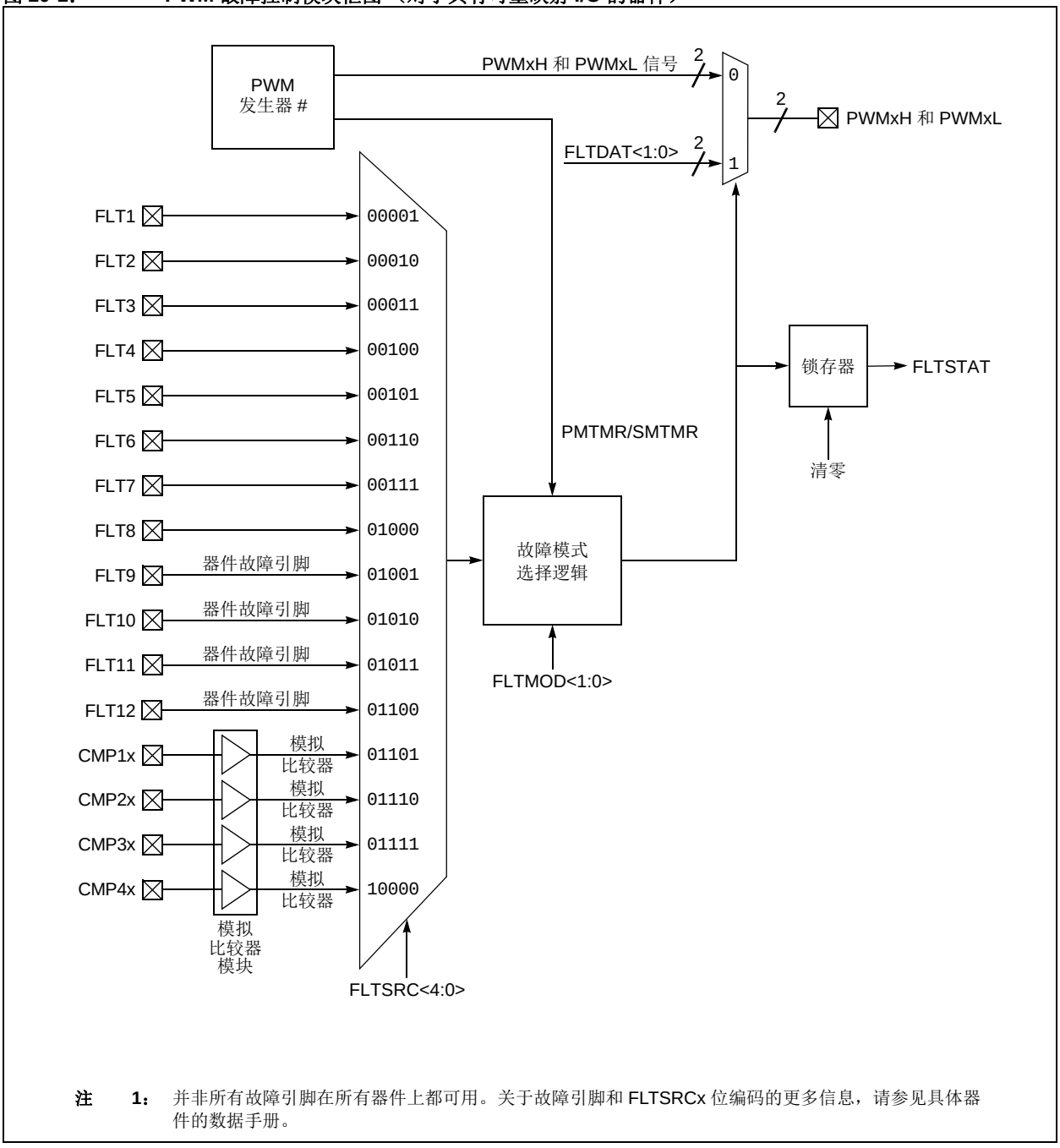
FLTx 引脚通常高电平有效。FLTPOL 位（FCLCONx<2>）设置为 1 时，选定故障输入信号会反相；因此，这些引脚被设置为低电平有效。

在使能高速 PWM 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。

© 2011-2014 Microchip Technology Inc.

图 10-1 给出了对于具有可重映射 I/O 的器件， PWM 故障控制模块的框图。

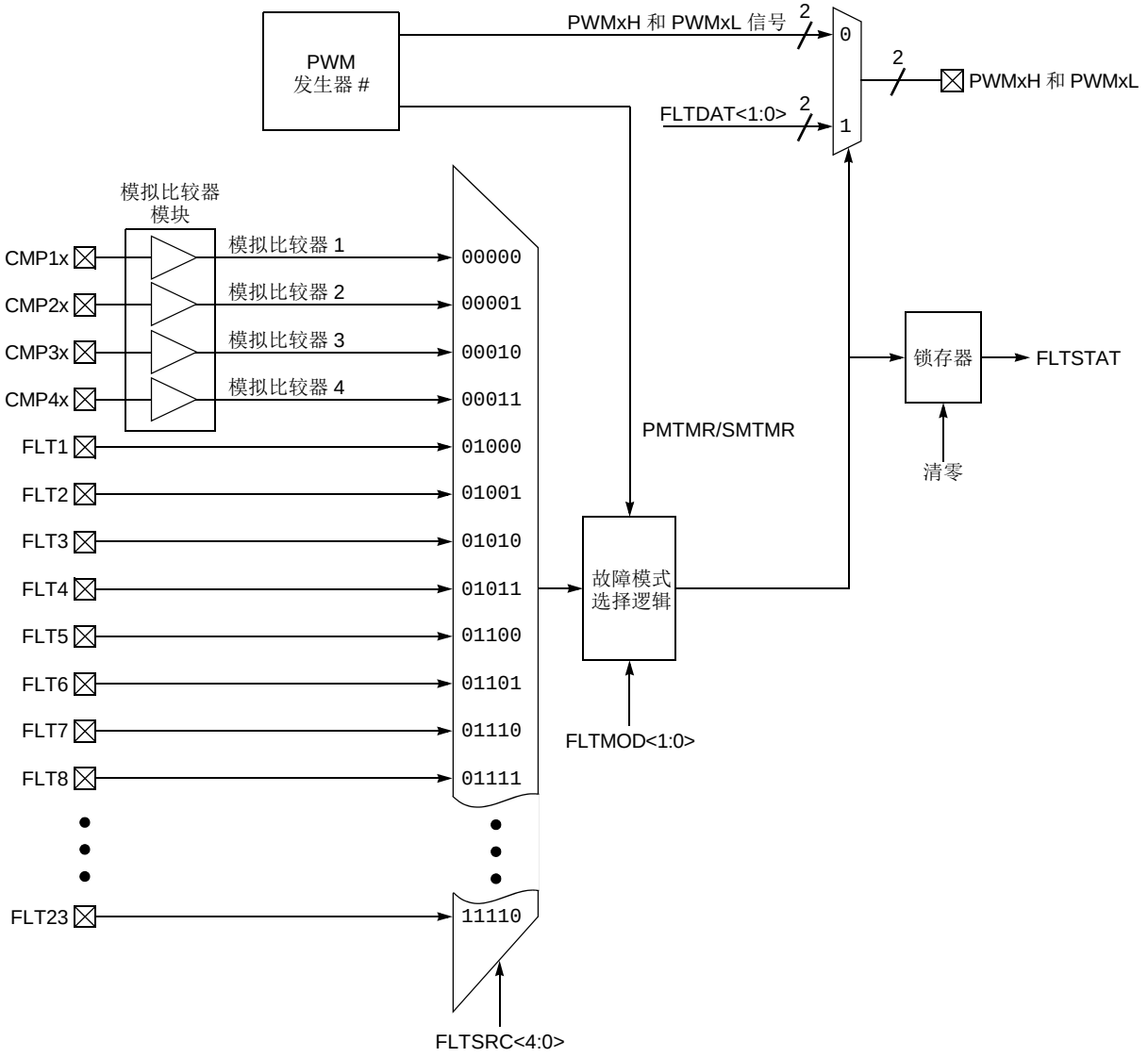
图 10-1： PWM 故障控制模块框图 （对于具有可重映射 I/O 的器件）(1)



注 1： 并非所有故障引脚在所有器件上都可用。关于故障引脚和 FLTSRCx 位编码的更多信息，请参见具体器件的数据手册。

图 10-2 给出了对于不具有可重映射 I/O 的器件， PWM 故障控制模块的框图。

图 10-2： PWM 故障控制模块框图 （对于不具有可重映射 I/O 的器件）(1)



注 1： 关于可用模拟比较器输出和故障引脚的更多信息，请参见具体器件的数据手册。

© 2011-2014 Microchip Technology Inc.

10.1 模拟比较器产生的 PWM 故障信号

注： 本节仅适用于具有可重映射 I/O 但没有将模拟比较器输出作为专用故障源的器件。

要使用比较器输出作为一个故障或限流源，需要将比较器输出重映射到一个可重映射 I/O 引脚，并将一个外部故障源重映射为同一引脚的输入。重映射可以映射到 GPIO 引脚，也可以映射到虚拟引脚。

虚拟引脚的功能与所有其他 RPx 引脚相同，只是引脚排列有所不同。虚拟引脚位于器件内部，它们不与物理器件引脚连接。图 10-3 给出了比较器输出重映射到虚拟引脚的图示。

例如，模拟比较器的输出可以配置为与虚拟引脚 RP32 连接，并将 PWM 故障源配置为 RP32。该配置使模拟比较器无需使用外部器件引脚就可以触发 PWM 故障。关于虚拟引脚的更多信息，请参见具体器件数据手册中的 **“I/O 端口”** 章节。

例 10-1 给出了这样一种配置：模拟比较器 1 （ACMP1）作为 PWM 的一个故障源，连接到故障输入引脚 1。

使用了以下输出和输入功能：

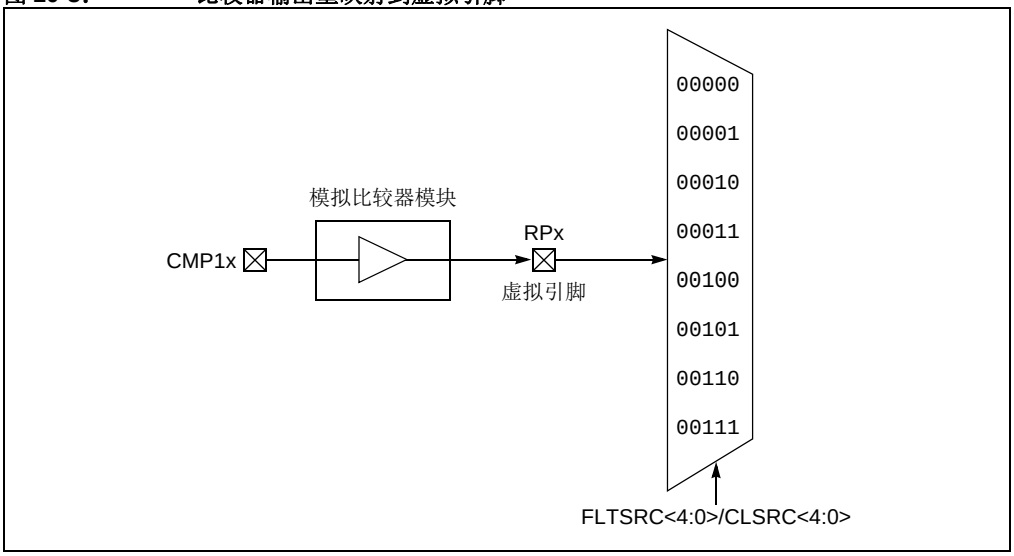
- 输出功能：模拟比较器 1
- 输入功能：PWM 故障引脚 1

例 10-1： 将模拟比较器配置为 PWM 的故障源

```
__builtin_write_OSCCONL(OSCCON & ~(1<<6));  /* Unlock Registers */

/* Configure Comparator Output Function */
RPOR16bits.RP32R = 0b100111;              /* Assign ACMP1 To Pin RP32 */

/* Configure Fault Input Function */
RPINR29bits.FLT1R=32;                     /* Assign Fault1 To Pin RP32 */

__builtin_write_OSCCONL(OSCCON | (1<<6));  /* Lock Registers */
```

注 1： 比较器输出也可以重映射到一个通用 I/O （GPIO）引脚。
2： 例 10-1 给出的是 dsPIC33FJ(16/06)GSXXX 系列器件的示例。

图 10-3： 比较器输出重映射到虚拟引脚



注： 关于虚拟引脚数的更多信息，请参见具体器件的数据手册。

10.2 故障中断

FLTIEN 位 （PWMCONx<12>）用于决定在 FLTx 输入置为高电平时是否产生中断。在发生故障条件时， FLTDAT<1:0> 位 （IOCONx<5:4>）（高 / 低）将提供要分配给 PWMxH 和 PWMxL 引脚的数据值。

PWM 故障状态在故障中断状态位 FLTSTAT （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 FLTx 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。

除了作为 PWM 逻辑工作之外，故障引脚逻辑也可以作为外部中断引脚工作。如果 FCLCONx 寄存器设置不允许故障条件影响 PWM 发生器，则故障引脚可以用作通用中断引脚。

10.2.1 故障输入引脚模式

故障输入引脚具有以下工作模式：

- **锁定模式**

  在锁定模式下，当故障引脚置为有效时， PWM 输出会遵从 IOCONx 寄存器的 FLTDATx 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已用软件清零。当这两个操作均发生且执行了相应的故障退出序列（如**第 10.4 节 “故障退出”**所述）时， PWM 输出会在下一个 PWM 周期边界开始时恢复正常工作。如果 FLTSTAT 位（PWMCONx<15>）在故障条件结束之前清零，高速 PWM 会等到故障引脚不再置为有效为止。软件可以通过向 FLTIEN 位 （PWMCONx<12>）写入 0 来清零 FLTSTAT 位。

- **逐周期模式**

  在逐周期模式下，只要故障引脚保持置为有效，PWM 输出就会遵循由 FLTDATx 位定义的状态。在故障引脚置为无效之后， PWM 输出会在下一个 PWM 周期边界处恢复正常工作。与锁定模式不同，在逐周期故障模式下，不需要执行特定的操作序列来退出该模式。

每个故障输入引脚的工作模式使用 FLTMOD<1:0> 位 （FCLCONx<1:0>）进行选择。

10.3 故障进入

对于故障输入引脚，PWM 引脚总是提供相对于器件时钟信号的异步响应。因此，如果 FLTDATx 位置为无效（设置为 0），PWM 发生器会立即将关联的 PWM 输出置为无效，如果指定的 FLTDAT 位置为有效 （设置为 1），则死区逻辑会先对 FLTDAT<1:0> （IOCONx<5:4>）位 （高 / 低）进行处理，然后再将其输出为 PWM 信号。

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 12.4 节 “故障 / 限流改写和死区逻辑”**。

© 2011-2014 Microchip Technology Inc.

10.4 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。

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。无需编写额外的代码来退出故障条件。

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 PWM 故障源来确定故障信号是否已置为无效。
2. 将 OVRDAT<1:0> （IOCONx<7:6>）位设置为 0。
3. 通过将 OVRENH （IOCONx<9>）和 OVRENL （IOCONx<8>）位设置为高电平，使能 PWMxH 和 PWMxL 改写。
4. 通过设置 FLTMOD<1:0> 位 （FCLCONx <1:0>） = ‘0b11，禁止 PWM 故障。
5. 提供至少 1 个 PWM 周期的延时。
6. 通过设置 FLTMOD<1:0> 位 （FCLCONx <1:0>） = ‘0b00，使能 PWM 故障。
7. 如果允许了 PWM 故障中断，则执行以下子步骤，然后执行步骤 8；如果未允许，则跳过该步骤并执行步骤 8。
   - 完成 PWM 故障中断服务程序 （Interrupt Service Routine， ISR）。
   - 通过设置 FLTIEN 位 （PWMCONx<12>）= 0，禁止 PWM 故障中断。
   - 通过设置 FLTIEN 位 （PWMCONx<12>）= 1，使能 PWM 故障中断。
8. 通过清零 OVRENH 和 OVRENL 位禁止改写。

10.5 PMTMR 被禁止时的故障退出

存在一种特殊的情形：在 PWM 时基被禁止 （PTEN = 0）时退出故障条件。当故障输入设定为逐周期模式时，PWM 输出会在故障输入引脚置为无效时立即恢复为正常操作。PWM 输出应恢复为它们的默认设定值。当故障输入设定为锁定模式时， PWM 输出会在故障输入引脚置为无效，并且 FLTSTAT 位 （PWMCONx<15>）在软件中清零时立即恢复。

10.6 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 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。

10.7 PWM 限流引脚

PWM 限流引脚的关键功能如下：

- 对于具有可重映射 I/O 的器件，每个 PWM 发生器可以从最多 8 个可重映射故障源中选择其故障输入源。少数器件具有专用的外部故障引脚和可重映射故障源。在一些具有可重映射 I/O 的器件中，模拟比较器的输出可直接用作故障源，而在其他器件中，可以通过虚拟引脚将模拟比较器输出指定为故障 （见**第 10.1 节 “模拟比较器产生的 PWM 故障信号”**）。关于可用故障源的更多信息，请参见具体器件的数据手册。
- 对于不具有可重映射 I/O 的器件，每个 PWM 发生器可以从最多 23 个故障引脚和最多 4 个模拟比较器输出中选择其限流输入源。
- 每个 PWM 发生器都具有限流控制信号源选择位 CLSRC<4:0> （FCLCONx<14:10>）。这些位指定其限流信号的来源。
- 每个 PWM 发生器都具有相应的限流中断允许位 CLIEN （PWMCONx<11>）。该位用于允许限流 IRQ 的产生。
- 每个 PWM 发生器都具有关联的限流极性位 CLPOL （FCLCONx<9>）。
- 在发生限流条件时， PWMxH 和 PWMxL 发生器的输出可以变为以下状态之一：
  - 如果独立故障模式使能位 IFLTMOD （FCLCONx<15>）置 1， CLDAT<1:0> （IOCONx<3:2>）位不用于改写功能。
  - 如果 IFLTMOD 位清零且 CLMOD 位 （FCLCONx<8>）置 1，使能了限流功能，则 CLDAT<1:0> 位将提供要在限流有效时分配给 PWMxH 和 PWMxL 输出的数据值。

限流引脚的主要功能如下：

- 限流可以改写 PWM 输出。CLDAT<1:0> 位的值可以为 00 或 11。如果 CLDATx 位设置为 00，则会对它进行异步处理，以便可以立即关闭应用电路中的相关功率晶体管。如果 CLDAT 位设置为 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<11>）用于控制限流中断信号的产生。即使 CLMOD 位 （FCLCONx<8>）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 ADC 转换过程。ADC 触发信号总是有效，无论高速 PWM 模块、FLTMOD<1:0> 位（FCLCONx<1:0>）或 FLTIEN 位（PWMCONx<12>）的状态如何。

10.7.1 电流复位模式的配置

进行以下配置时，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 = 1）。

例 10-2 给出了电流复位模式的配置。更多信息，请参见**第 16.5 节 “电流复位 PWM 模式”**。

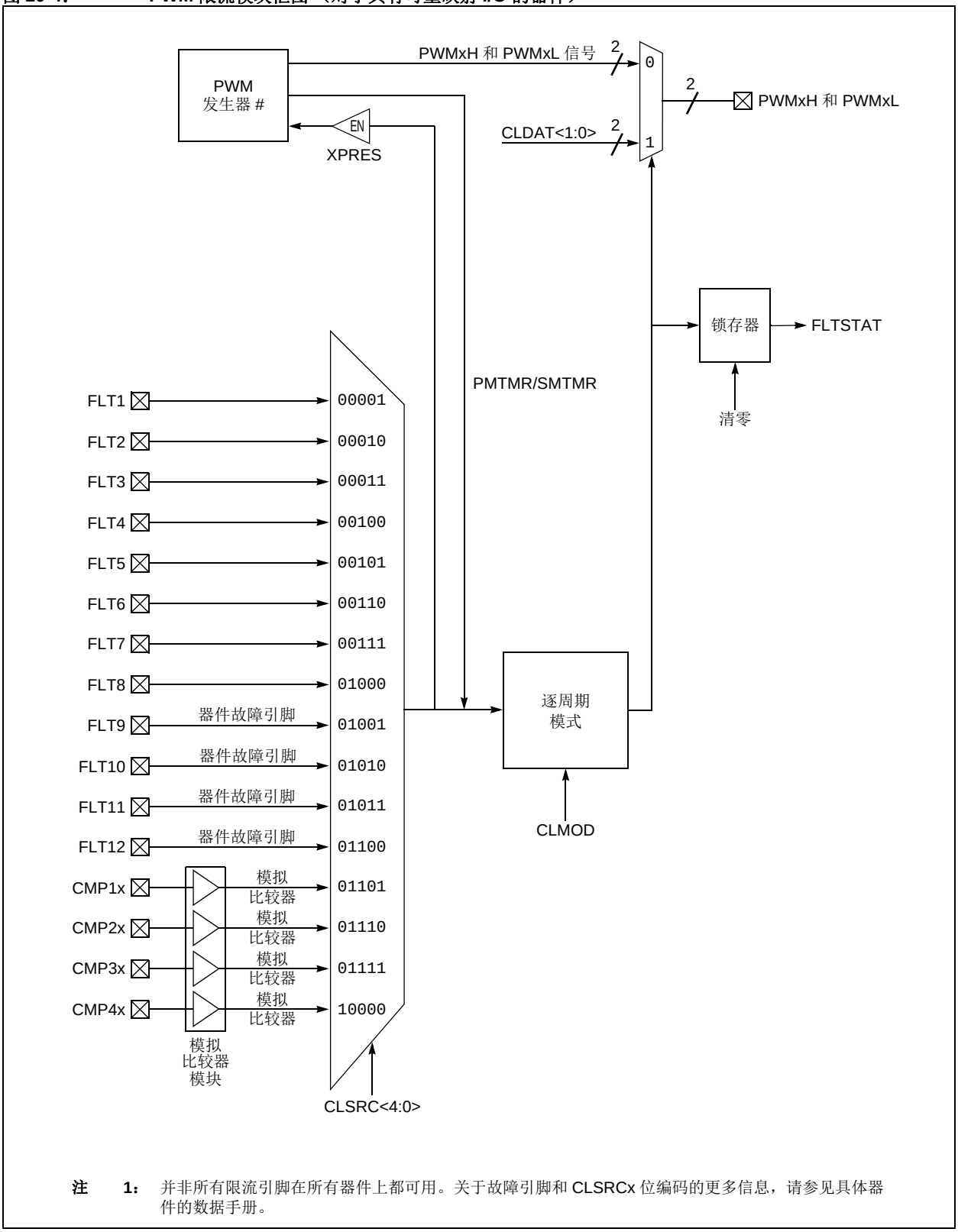
例 10-2： 电流复位模式的配置

```
/* Configuration of Current Reset mode */
FCLCONxbits.CLMOD = 0;     /* Current-limit mode is disabled */
PWMCONxbits.XPRES = 1;     /* External PWM Reset mode is enabled */
PWMCONxbits.ITB = 1;       /* Independent Time Base mode is enabled */
```


© 2011-2014 Microchip Technology Inc.

图 10-4 给出了对于具有可重映射 I/O 的器件， PWM 限流模块的框图。

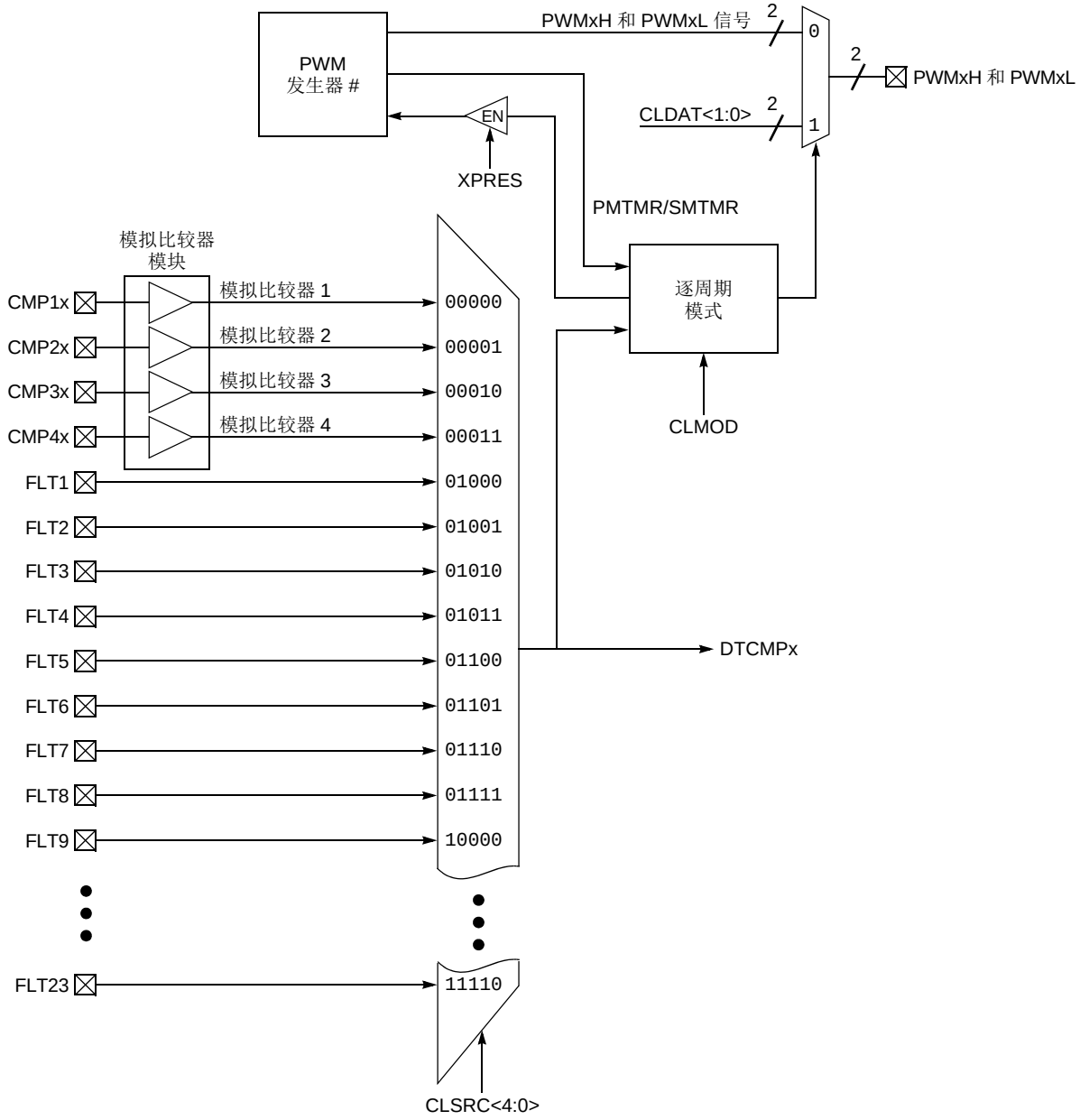
图 10-4： PWM 限流模块框图 （对于具有可重映射 I/O 的器件）(1)



注 1： 并非所有限流引脚在所有器件上都可用。关于故障引脚和 CLSRCx 位编码的更多信息，请参见具体器件的数据手册。

图 10-5 给出了对于不具有可重映射 I/O 的器件， PWM 限流模块的框图。

图 10-5： PWM 限流控制模块框图 （对于不具有可重映射 I/O 的器件）(1)



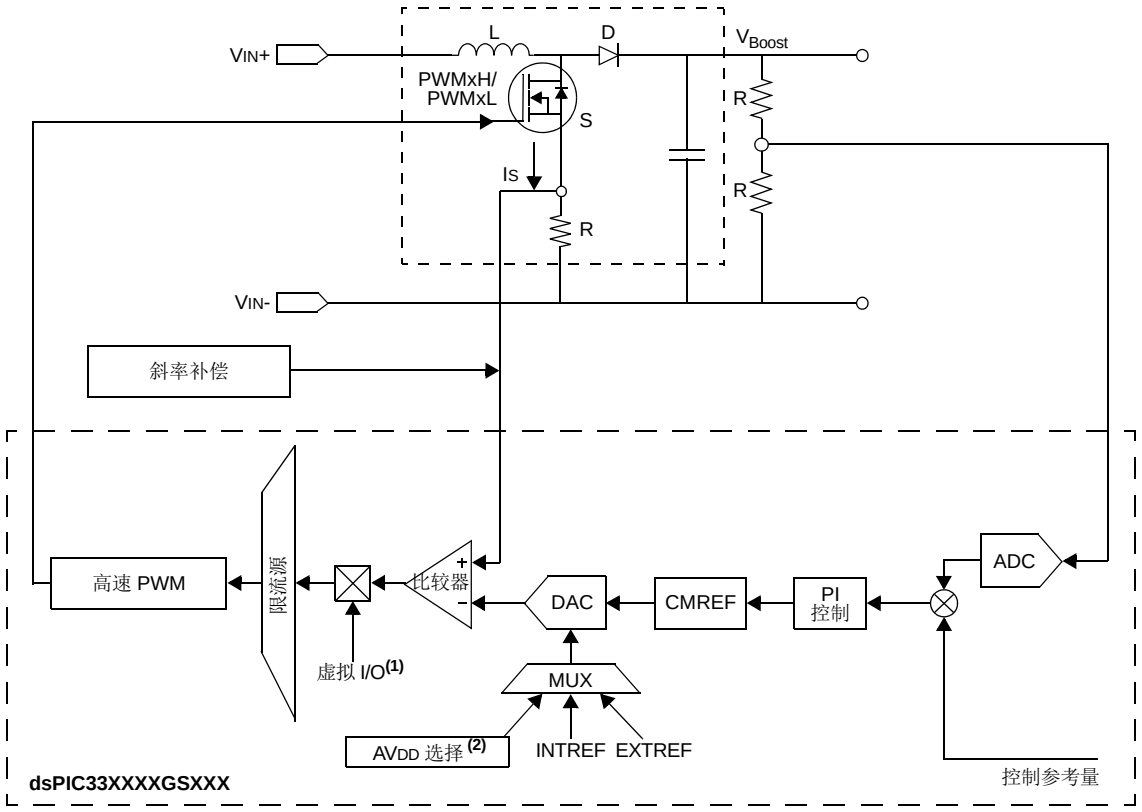
注 1： 关于可用模拟比较器输出和故障引脚的更多信息，请参见具体器件的数据手册。

© 2011-2014 Microchip Technology Inc.

10.7.2 在逐周期模式下配置模拟比较器

内置的高速模拟比较器可以通过配置来设置逐周期模式。图 10-6 给出了模拟比较器在逐周期模式下的典型配置。

图 10-6： 数字峰值电流模式升压转换器



注 1： 仅适用于具有可重映射 I/O 的器件。

2： 该电压可以为 AVDD 或 AVDD/2，具体取决于器件型号。关于模拟比较器 DAC 参考电压的更多信息，请参见具体器件的数据手册。

模拟比较器提供高速操作，典型延时为 15–20 ns。比较器的同相输入与 CMPCONx 寄存器中的一个模拟多路开关 （INSEL<1:0>）连接。比较器的同相输入用于测量电流信号 （电压信号）。

比较器的反相输入端总是连接到 DAC 电路。

根据器件型号，高速模拟比较器的 DAC 可能为 10 位或 12 位。关于 DAC 的更多信息，请参见具体器件的数据手册。

DAC 范围可以使用比较器控制寄存器（CMPCONx）中的 RANGE 位选择。具有可重映射 I/O 的 dsPIC33F/dsPIC33E“GS”系列器件最多支持 6 个虚拟 RPn 引脚，它们的功能与所有其他 RPn 引脚相同，只是引脚排列有所不同。更多信息，请参见具体器件数据手册中的**“I/O 端口”**章节。

这些引脚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 10-3 显示了在具有 FCLCONx 寄存器 （版本 1）位 FLTSRC<4:0>/CLSRC<4:0> 的器件中，使用虚拟引脚 RP32 将故障 1 配置为模拟比较器 1 的输出。

例 10-3： 虚拟引脚配置 （对于具有可重映射 I/O 的器件）

```
RPINR29bits.FLT1R = 0b100000; /* Fault Source(FLT1) connected to RP32 */

RPOR16bits.RP32R = 0b100111;    /* Output of the analog comparator 1 connected
                                   to RP32 */
```

10.8 限流中断

PWM 限流条件的状态在 CLSTAT 位 （PWMCONx<14>）中提供。CLSTAT 位指示限流 IRQ 标志状态 （如果 CLIEN 位 （PWMCONx<11>）置 1）。如果不允许限流中断，则 CLSTAT 位以正逻辑形式指示选定限流输入的状态。当不使用与 PWM 发生器关联的限流输入引脚时，该引脚可以用作通用 I/O 或中断输入引脚。

限流引脚通常高电平有效。如果 CLPOL 位 （FCLCONx<9>）设置为 1，则模块会将选定限流输入信号极性反相，将信号驱动为低电平有效状态。

选定限流信号产生的中断将组合产生单个 IRQ 信号。该信号将发送到中断控制器，它具有与自己关联的中断向量、中断标志位、中断允许位和中断优先级位。

在使能高速 PWM 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。

10.9 并发 PWM 故障和限流

除非故障功能使能并有效，否则限流改写功能（如果使能并有效）会强制 PWMxH 和 PWMxL 引脚读为由 CLDAT<1:0> 位 （IOCONx<3:2>）指定的值。如果选定的故障输入有效， PWMxH 和 PWMxL 输出会读为由 FLTDAT<1:0> 位 （IOCONx<5:4>）指定的值。

© 2011-2014 Microchip Technology Inc.

10.10 到 ADC 的 PWM 限流触发输出

CLSRC<4:0> 位 （FCLCONx<14:10>）和 FLTSRC<4:0> 位 （FCLCONx<7:3>）用于控制每个 PWM 发生器模块的故障选择。控制多路开关会为相应的模块选择所需的故障和限流信号。选定的限流信号 （它也可以用作 ADC 模块的触发信号）会启动 ADC 采样和转换操作。例 10-4 给出了 PWM 故障、限流和前沿消隐 （LEB）的配置。

例 10-4： PWM 故障、限流和前沿消隐配置

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

//FCLCON1bits.IFLTMOD = 0;  /* CLDAT bits and FLTDAT bits control PWMxH/PWMxL pins on occurrence
                               of current limit and Fault inputs respectively */
//FCLCON1bits.CLSRC = 0;     /* Fault 1 is selected as source for the Current Limit Control signal */
//FCLCON1bits.FLTSRC = 3;    /* Fault 4 is selected as source for the Fault Control Signal source */
//FCLCON1bits.CLPOL = 1;     /* Current-limit source is active-low */
//FCLCON1bits.FLTPOL = 1;    /* Fault Input source is active-low */
//FCLCON1bits.CLMOD = 1;     /* Enable current-limit function */
//FCLCON1bits.FLTMOD = 1;    /* Enable Cycle-by-Cycle Fault mode */

FCLCON1 = 0x031D;

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 */
//LEBCON1bits.LEB = 8;      /* Blanking period of 8.32 ns */

LEBCON1 = 0xAC40;

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

注： 例 10-4 中的代码只适用于具有 LEBCONx （版本 1）寄存器的器件。

11.0 特殊功能

高速 PWM 模块具有以下特殊功能：

- 前沿消隐 （LEB）
- 独立时基捕捉
- 死区补偿
- 斩波模式
- PWM 引脚交换
- PWM 保护锁定 / 解锁密钥寄存器

11.1 前沿消隐 （LEB）

每个 PWM 发生器通过 LEB 控制寄存器中的 LEB<6:0> 位 （LEBCONx<9:3> 或 LEBDLYx 寄存器中的 LEB<8:0> 位 （LEBDLYx<11:3>），具体取决于器件型号）、 PHR （LEBCONx<15>）、 PHF （LEBCONx<14>）、 PLR （LEBCONx<13>）、 PLF （LEBCONx<12>）、 FLTLEBEN （LEBCONx<11>）和 CLLEBEN（LEBCONx<10>）位来支持限流和故障输入的前沿消隐。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。

各 LEB 位是边沿敏感的，支持根据器件型号在 PWMxH 和 PWMxL 引脚任意指定的上升沿或下降沿后，将限流和故障输入消隐 （忽略） 0 ns 到最高 4252 ns，消隐周期以 8.32 ns 为单位进行递增。

公式 11-1： 具有 LEB （版本 2）寄存器的器件的 LEB 计算

$$\textit{最大时钟速率下的 LEB 持续时间} = (LEBDLYx\langle 8{:}0\rangle) * 8.32\ ns$$

公式 11-2： 具有 LEB （版本 1）寄存器的器件的 LEB 计算

$$\textit{最大时钟速率下的 LEB 持续时间} = (LEBCONx\langle 6{:}0\rangle) * 8.32\ ns$$

在高速开关应用中，开关（如 MOSFET/IGBT）通常会产生极大的瞬变。这些瞬变可能会导致会产生问题的测量误差。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。

PHR 位 （LEBCONx<15>）、 PHF 位 （LEBCONx<14>）、 PLR 位 （LEBCONx<13>）和 PLF 位（LEBCONx<12>）用于选择 PWMxH 和 PWMxL 信号的边沿类型，该边沿类型将启动消隐定时器。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。请注意，PHR、PHF、PLR 和 PLF 位在由于 IOCONx 寄存器中的 POLH 或 POLL 设置而应用 PWM 极性之前，基于 PWMxH/PWMxL 信号来控制前沿消隐（LEB）的应用。例如，设置 PHR = 1 和 POLH = 1 时，将在 PWMxH 信号的下降沿在器件输出引脚上应用 PWMxH 前沿消隐。

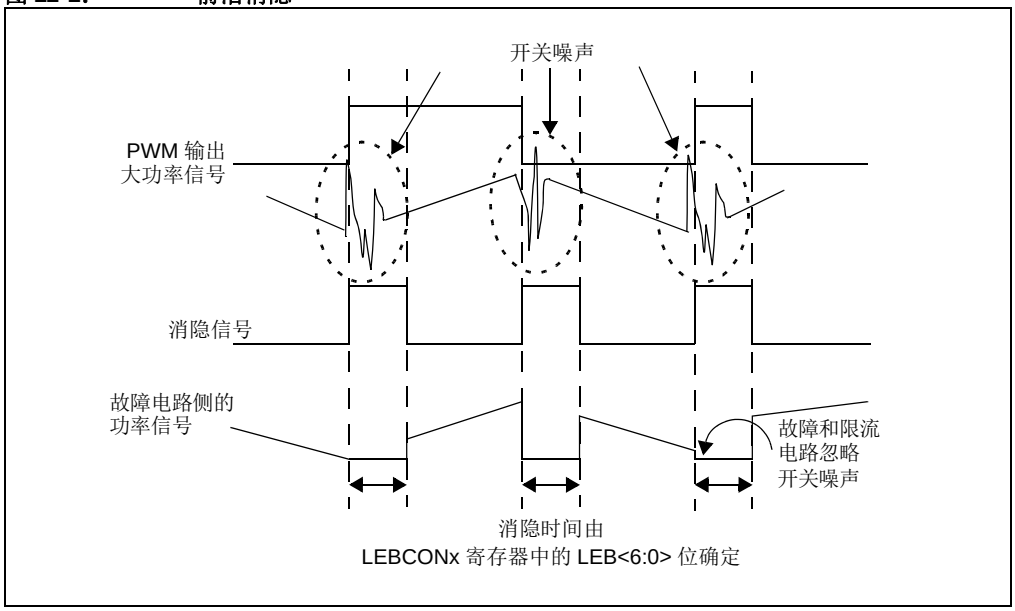
© 2011-2014 Microchip Technology Inc.

FLTLEBEN 位（LEBCONx<11>）和 CLLEBEN 位（LEBCONx<10>）用于使能对选定的故障和限流输入应用消隐周期。图 11-1 说明了应用如何忽略指定消隐周期中的故障信号。

在具有 LEB 版本 2 寄存器的器件上，可以指定完全忽略限流和 / 或故障信号的时间段。LEBCONx 寄存器中的 BCH、BCL、BPHH、BPHL、BPLH 和 BPLL 位用于选择 PWMxH、PWMxL 和 / 或斩波时钟信号作为状态消隐功能的信号源。此外，还可以在 PWMxH 输出为高电平和 / 或低电平，以及 PWMxL 为高电平和 / 或低电平时对选定故障或限流信号进行消隐。PWM 附属控制寄存器中的 PWM 状态消隐源选择位 BLANKSEL<3:0> （AUXCONx<11:8>）用于选择用作消隐信号源的 PWM 发生器。

注：	要确定您所用器件可用的 LEB 版本，请参见具体器件数据手册中的 **“高速 PWM”** 章节。

图 11-1： 前沿消隐



11.2 独立时基捕捉

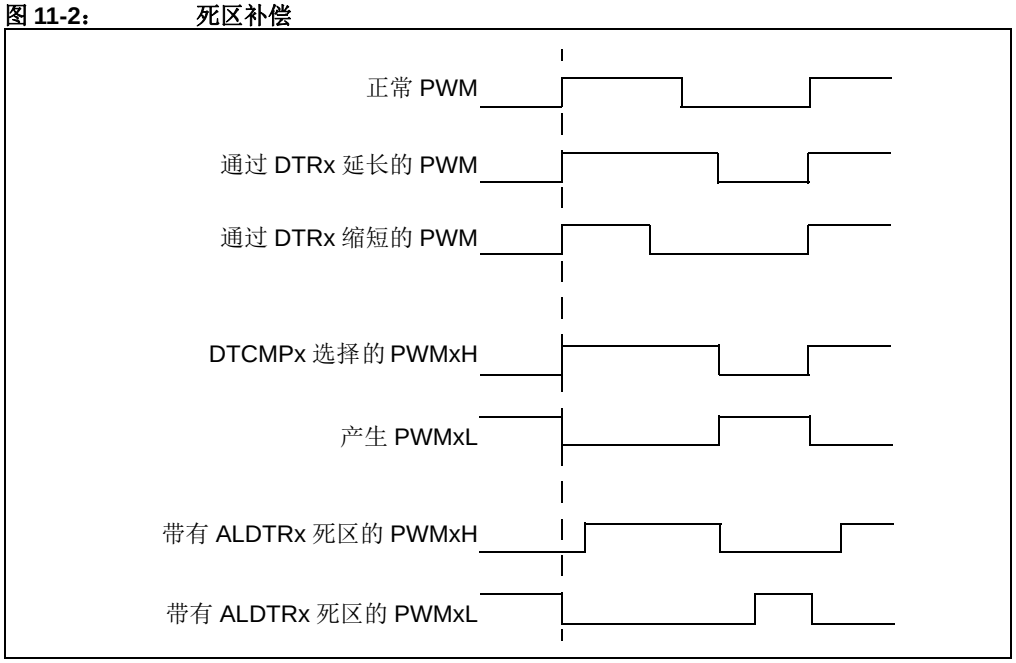
每个 PWM 发生器都具有一个主 PWM 时基捕捉寄存器 （PWMCAPx），该寄存器会在检测到限流信号上升沿时自动捕捉独立时基计数器值。该功能仅在应用 LEB 功能之后有效。用户应用程序应在下一个 PWM 周期导致 PWMCAPx 寄存器再次更新之前读取寄存器。

在使用模拟比较器或外部电路来终止 PWM 占空比或周期的限流 PWM 控制应用中，需要使用 PWMCAPx 寄存器。通过读取达到电流门限值时的独立时基值，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主 PWM 时基捕捉寄存器。

11.3 死区补偿

在交流电机控制应用中，对 PWM 信号应用死区时，晶体管会被禁止。在死区期间，电机电流会继续流过续流二极管，但施加的电压为 0。死区期间施加零电压会导致所需电压波形失真，继而导致电机电流失真。该失真会导致可影响控制系统稳定性和电机性能的转矩变化。当通过 DTC<1:0> 位（PWMCONx<7:6>）选择死区补偿模式时，外部输入信号 DTCMPx 将导致 DTRx 寄存器中的值与 MDC/PDCx 寄存器指定的占空比相加或相减。 ALTDTRx 寄存器将指定 PWMxH 和 PWMxL 输出信号的死区周期。死区补偿只能用于正死区模式。死区补偿不支持负死区。图 11-2 给出了死区补偿时序图。

图 11-2： 死区补偿



注： 死区补偿模式仅适用于互补 PWM 输出模式。在任何其他 PWM 输出模式中指定死区补偿会产生不可预测的结果。关于死区补偿功能的可用性，请参见具体器件的数据手册。

11.4 斩波模式

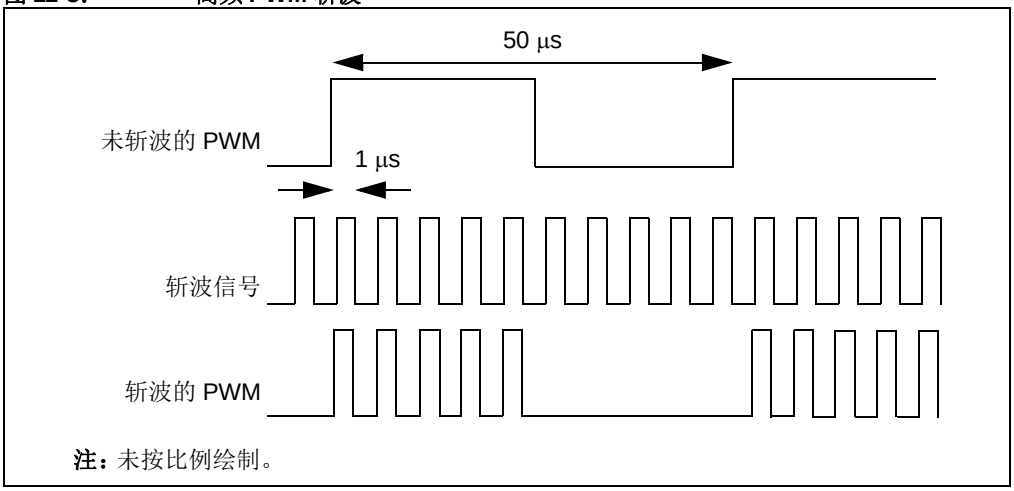
许多电源控制应用都使用需要隔离晶体管栅极驱动的晶体管配置。一个示例就是三相“H 桥”配置，在该配置中，上桥臂晶体管处于抬高电势。

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 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 11-3 给出了一个高速 PWM 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。

© 2011-2014 Microchip Technology Inc.

图 11-3：　　高频 PWM 斩波



注：未按比例绘制。

斩波功能会对 PWM 输出执行逻辑与操作。由于斩波时钟的周期有限，所以所产生的 PWM 占空比分辨率会被限制为斩波时钟周期的一半。

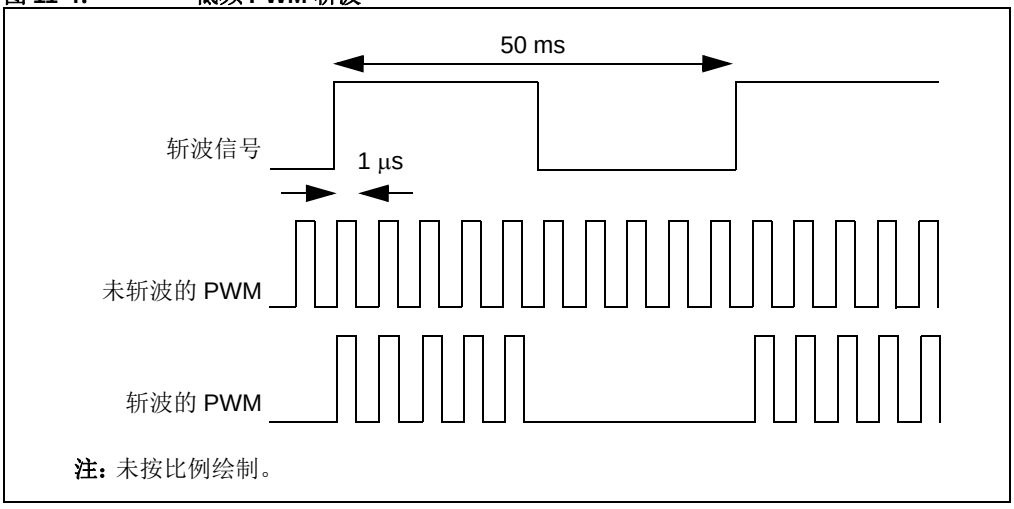
用户可以通过 PWM 斩波时钟发生器寄存器（CHOP）指定斩波时钟频率。斩波值用于指定 PWM 时钟分频比。斩波时钟分频器以 PCLKDIV<2:0> 位（PTCON2<2:0>）指定的 PWM 时钟频率工作。 CHOP 寄存器中的 CHPCLKEN 位用于使能斩波时钟发生器。

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 （AUXCONx<1>）或 CHOPLEN 位 （AUXCONx<0>）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。CHPCLK 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 11-4 给出了一个低频 PWM 斩波波形。在该图中，另一个以较低频率工作的 PWM 发生器对 PWM 信号进行斩波或 “消隐”。

图 11-4：　　低频 PWM 斩波



注：未按比例绘制。

11.5 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（IOCONx<1>）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 （IOCONx<0>）清零时，用户应用程序更改了 SWAP 位的状态，交换功能会尝试在 PWM 周期中途执行，此操作将产生不可预测的结果。

交换功能必须在应用死区之前执行。由于执行切换功能可能会使能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：**在动态交换中，可以根据系统响应动态更改 SWAP 位的状态 （如， SMPS 电源控制）。
- **静态交换：**在静态交换中， 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）。

11.5.1 例 1：SMPS 电源控制的引脚交换

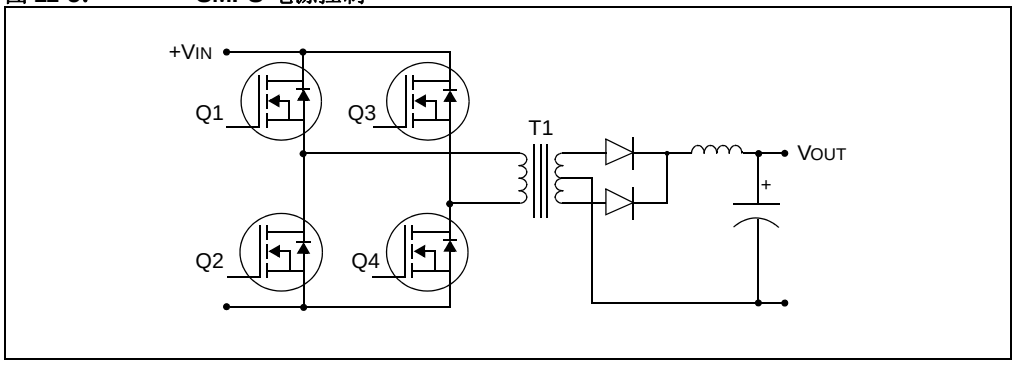
SMPS 电源控制示例介绍动态交换。在电源转换应用中，晶体管调制技术可以在全桥零电压转换（Zero Voltage Transition， ZVT）和标准的全桥“动态”转换之间切换，以满足不同的负载和效率要求。如图 11-5 所示，通用全桥转换器可以工作于推挽模式。晶体管配置如下：

- 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 11-5： SMPS 电源控制



© 2011-2014 Microchip Technology Inc.

11.5.2 例 2：电机控制的引脚交换

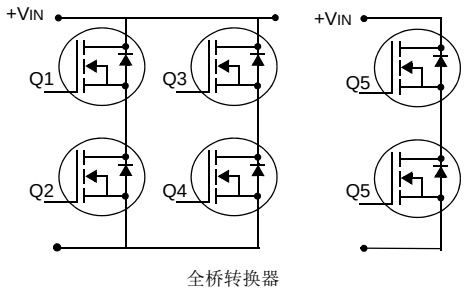
电机控制示例介绍静态交换。假设有一个通用电机控制系统，它可以驱动两种不同类型的电机，例如直流电机和三相交流电感电机。

直流有刷电机通常使用全桥晶体管配置，如图 11-6 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”。直流有刷电机中不使用 Q5 和 Q6。

晶体管配置如下：

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

图 11-6： 电机控制



与直流电机不同，交流电感电机会使用全桥配置中的所有晶体管。但是，明显的一点区别是对于后者晶体管以三个半桥形式驱动，其中，上桥臂晶体管由 PWMxH 输出驱动，而下桥臂晶体管则由 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

例 11-1 显示了 PWM 引脚交换。

例 11-1： PWM 引脚交换

```
/* PWM Pin Swapping feature */
IOCONxbits.SWAP = 1;
/* PWMxH output signal is connected to the PWMxL pin and vice versa */
```

11.6 PWM 保护锁定 / 解锁密钥寄存器

FCLCONx 和 IOCONx 寄存器包含用于控制 PWM 发生器输出引脚状态的位。PWMKEY 寄存器用于为这些寄存器提供 B 类故障保护。

要写入 FCLCONx 和 IOCONx 寄存器，用户必须向 PWMKEY 寄存器连续写入两个字（先写 0xABCD，再写 0x4321）来执行解锁操作。对 IOCONx 和 FCLCONx 的写访问必须是紧接在解锁过程之后的下一个 SFR 访问。在解锁过程和后续寄存器（IOCONx 或 FCLCONx）写访问之间不能有任何其他 SFR 访问。写入寄存器 IOCONx 或 FCLCONx 可能需要许多解锁操作。

PWMKEY 寄存器是只写寄存器。对该寄存器的读访问将产生：0x0000。

例 11-2 给出了用于 PWM 保护锁定 / 解锁密钥寄存器配置的代码段。

例 11-2： PWM 保护锁定 / 解锁配置

```
/* To enable writing of FCLCON1 register */

asm volatile ("push.s");                  /* Context save w0-w3 */
asm volatile ("mov #0xABCD, w0");
asm volatile ("mov #0x4321, w1");
asm volatile ("mov #0x0603, w2");
asm volatile ("mov w0,PWMKEY");           /* Perform Unlock Sequence */
asm volatile ("mov w1,PWMKEY");
asm volatile ("mov w2,FCLCON1");          /* Write FCLCONx register */
asm volatile ("pop.s");                   /* Restore context for w0-w3*/

/* To enable writing of IOCON1 register  */
asm volatile ("push.s");                  /* Context save w0-w3 */
asm volatile ("mov #0xABCD, w0");
asm volatile ("mov #0x4321, w1");
asm volatile ("mov w0,PWMKEY");           /* Perform Unlock Sequence */
asm volatile ("mov w1,PWMKEY");
asm volatile ("bset IOCON1, #9");         /* Set OVRENH bit */
asm volatile ("pop.s");                   /* Restore context for w0-w3 */
```

注： PWM 锁定 / 解锁功能可以通过更改配置设置而禁止。关于该功能的可用性和更多信息，请参见具体器件的数据手册。

© 2011-2014 Microchip Technology Inc.

12.0 PWM 输出引脚控制

如果使能了高速 PWM 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
- 限流改写逻辑
- 故障改写逻辑
- PENH/L （GPIO/PWM）所有权 （最高优先级）

如果禁止了高速 PWM 模块，则 GPIO 模块将控制 PWMx 引脚。

例 12-1： PWM 输出引脚分配

```
/* PWM Output pin control assigned to PWM generator */
IOCON1bits.PENH = 1;
IOCON1bits.PENL = 1;
```

例 12-2： PWM 输出引脚状态选择

```
/* High and Low switches set to active-high state */
IOCON1bits.POLH = 0;
IOCON1bits.POLL = 0;
```

例 12-3： 使能高速 PWM 模块

```
/* Enable High-Speed PWM module */
PTCONbits.PTEN = 1;
```

12.1 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。高速 PWM 模块改写功能具有上面列表中分配的优先级。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 输出引脚所有权位 PENH （IOCONx<15>）和 PWMxL 输出引脚所有权位 PENL （IOCONx<14>）置 1，则高速 PWM 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。

当 PWMxH 引脚改写使能位 OVRENH （IOCONx<9>）和 PWMxL 引脚改写使能位 OVRENL 位（IOCONx<8>）改写特定输出时，使能改写时 PWMxH 和 PWMxL 引脚状态位 OVRDAT<1:0>（IOCONx<7:6>）将决定 PWM I/O 引脚的状态。

OVRENH 位 （IOCONx<9>）和 OVRENL 位 （IOCONx<8>）是高电平有效的控制位。当这两位置 1 时，相应的 OVRDATx 位会改写 PWM 发生器的 PWM 输出。

当 PWM 处于互补 PWM 输出模式时，死区发生器在改写时仍然有效。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。当手动改写 PWM 通道时，仍能执行死区插入。

注 1： 当 PWM 的分辨率配置为非 1.04 ns 时（即，PTCON2<2:0> = 1、2、3、...7，或 STCON2<2:0> = 1、2、3、...7），必须在对 OVRENH 位（IOCONx<9>）和 OVRENL 位（IOCONx<8>）的连续位写操作之间插入一条 NOP 指令。

2： 在使能了 PWM 锁定 / 解锁功能的器件上，只有通过向 PWMKEY 寄存器写入适当的字序列，才能写入 IOCONx 和 FCLCONx 寄存器。更多信息，请参见**第 11.6 节“PWM 保护锁定 / 解锁密钥寄存器”**。

例 12-4： PWM 输出改写控制

```
/* Define override state of the PWM outputs. PWMxH and PWMxL outputs will be
at logic level  ‘0’when overridden. */
IOCON1bits.OVRDAT = 0;

/* Override PWMxH and PWMxL outputs */
IOCON1bits.OVRENH = 1;
__builtin_nop();
IOCON1bits.OVRENL = 1;
.
.
.
/* Clear overrides of PWMxH and PWMxL outputs */

IOCON1bits.OVRENH = 0;
__builtin_nop();
IOCON1bits.OVRENL = 0;
```

12.2 改写优先级

当 PENH 位（IOCONx<15>）和 PENL 位（IOCONx<14>）置 1 时，以下优先级应用于 PWMx 输出：

1. 如果故障有效，FLTDAT<1:0> 位（IOCONx<5:4>）会优先于所有其他潜在信号源，由它设置 PWM 输出。
2. 如果故障无效，但限流事件有效，则会选择 CLDAT<1:0> 位（IOCONx<3:2>）作为信号源来设置 PWM 输出。
3. 如果故障和限流事件均无效，则用户改写使能位设置为 OVRENH 和 OVRENL，关联的 OVRDAT<1:0> 位（IOCONx<7:6>）将设置 PWM 输出。
4. 如果没有任何改写条件有效，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。

12.3 改写同步

如果 OSYNC 位（IOCONx<0>）置 1，则通过 OVRENH、OVRENL 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 = 0（表示 PWM 定时器不在运行），对 IOCONx 的写操作会在下一个 TCY 边界处生效。

© 2011-2014 Microchip Technology Inc.

12.4 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（IOCONx<5:4>）或 CLDAT<1:0> 位（IOCONx<3:2>）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> 位中的任意位为 0，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应用有帮助。

如果 FLTDAT<1:0> 位或 CLDAT<1:0> 位中的任意位为 1，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。

12.5 通过限流将输出置为有效

在响应限流事件时，CLDATx 位（IOCONx<3:2>）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）。当手动改写 PWM 通道时，仍能执行死区插入。

12.6 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 引脚分别与 RE0 和 RE1 I/O 端口复用，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 12-5： GPIO 引脚配置代码示例

```
/* PWM outputs will be pulled low when the device is halted by the
debugger */

TRISEbits.TRISE0 = 0;         /* Configure RE0 as output */
TRISEbits.TRISE1 = 0;         /* Configure RE1 as output */

LATEbits.LATE0 = 0;           /* Configure RE0 as low output */
LATEbits.LATE1 = 0;           /* Configure RE1 as low output */

/* PWM outputs will be pulled high when the device is halted by the
debugger */

TRISEbits.TRISE0 = 0;         /* Configure RE0 as output */
TRISEbits.TRISE1 = 0;         /* Configure RE1 as output */

LATEbits.LATE0 = 1;           /* Configure RE0 as low output */
LATEbits.LATE1 = 1;           /* Configure RE1 as low output */

/* PWM outputs will be in tristate (high impedence) when the device is halted
by the debugger */

TRISEbits.TRISE0 = 1;         /* Configure RE0 as input */
TRISEbits.TRISE1 = 1;         /* Configure RE1 as input */
```

13.0 PWM 占空比立即更新

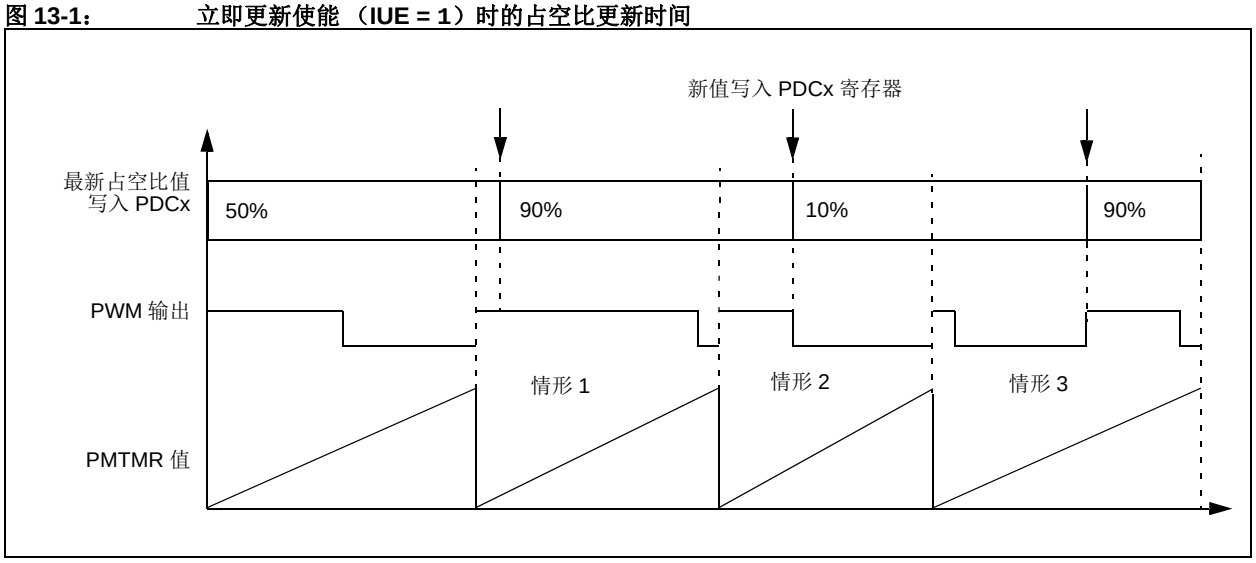
高性能 PWM 控制环应用需要最大的占空比更新速率。将 IUE 位 （PWMCONx<0>）置 1 会使能该功能。在闭环控制应用中，检测系统状态和驱动应用的 PWM 控制信号后续输出之间的任何延时都会使控制环稳定性下降。将 IUE 位置 1 可以最大程度降低写入占空比寄存器和 PWM 发生器响应该变化之间的延时。

通过 IUE 位，用户应用程序可以在写入占空比寄存器之后立即更新占空比值，而不是等待时基周期结束。如果 IUE 位置 1，使能占空比立即更新。如果该位清零，禁止占空比立即更新。使能立即更新功能时，可能会有以下三种情形：

- **情形 1：**如果新占空比写入时 PWM 输出有效，且新占空比大于当前时基值，则 PWM 脉冲宽度将变宽。
- **情形 2：**如果新占空比写入时 PWM 输出有效，且新占空比小于当前时基值，则 PWM 脉冲宽度将变窄。
- **情形 3：**如果新占空比写入时 PWM 输出无效，且新占空比大于当前时基值，则 PWM 输出将立即变为有效，并对新写入的占空比值保持有效。

图 13-1 显示了立即更新使能 （IUE = 1）时的占空比更新时间。例 13-1 给出了立即更新选择的配置。

图 13-1： 立即更新使能 （IUE = 1）时的占空比更新时间



例 13-1： 立即更新选择

```
/* Enable Immediate update of PWM */
PTCONbits.EIPU = 1;    /* Update Active period register immediately */
PWMCON1bits.IUE = 1;   /* Update active duty cycle, phase offset, and
                          independent time period registers immediately */
```


© 2011-2014 Microchip Technology Inc.

14.0 节能模式

本节讨论高速 PWM 模块在休眠模式和空闲模式下的操作。

14.1 休眠模式下的高速 PWM 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 （TCY）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。如果高速 PWM 模块用于控制电源应用中的负载，在执行 PWRSAV 指令之前，必须先将高速 PWM 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。

如果为高速 PWM 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。当器件处于休眠模式时，如果其中一个故障引脚被驱动为低电平， PWM 输出将被驱动为设定的故障状态。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 PWRSAV 指令后的下一条指令开始继续执行程序。

14.2 空闲模式下的高速 PWM 操作

PWM 模块含有一个 PWM 时基空闲模式停止位 PTSIDL （PTCON<13>）。PTSIDL 位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，系统无法访问 SFR。这是模块的最低功耗模式。停止空闲模式的行为与休眠模式一样，并且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：	关于节能模式和看门狗定时器的更多信息，请参见具体器件的数据手册。

14.3 低速模式

在该模式下，可使用两种方法来降低功耗：

1. 通过 PCLKDIV<2:0> 位 （PTCON2<2:0>）和 （STCON2<2:0>）位选择的 PWM 时钟预分频比会将 PWM 模块配置为以较低速度工作，从而降低功耗。该模式牺牲 PWM 分辨率来实现功耗降低。
2. 高分辨率 PWM 周期禁止位 HRPDIS（AUXCONx<15>）和高分辨率 PWM 占空比禁止位 HRDDIS （AUXCONx<14>）用于禁止与高分辨率占空比和 PWM 周期关联的电路。如果 HRDDIS 位置 1，则与相应 PWM 发生器的高分辨率占空比、相位偏移和死区关联的电路会被禁止。如果 HRPDIS 位置 1，则与相应 PWM 发生器的高分辨率 PWM 周期关联的电路会被禁止。通常，许多应用要么需要高分辨率占空比或相位偏移（对于固定频率工作模式），要么需要高分辨率 PWM 周期 （对于可变频率工作模式，例如谐振模式）。极少有应用同时需要这两种高分辨率模式。可降低工作电流的能力始终都是一个优势。当 HRPDIS 位置 1 时， PWM 周期的最小度量单位为 8 ns。如果 HRDDIS 位置 1，则 PWM 占空比、相位偏移和死区的最小度量单位为 8 ns。

15.0 独立时基的外部控制 （电流复位模式）

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独立时基模式位 ITB（PWMCONx<9>）置 1，PWM 发生器将以独立时基模式工作。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<4:0> 位 （FCLCONx<14:10>）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 （FCLCONx<9>）指定。

在主独立时基模式下，以及迟滞和临界导通模式下，PFC 应用必须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。

注：	XPRES = 1 且 SWAP = 1 时，PWM 发生器仍然要求到达 PWMxH 引脚的信号无效，才会复位 PWM 计数器。

© 2011-2014 Microchip Technology Inc.

16.0 应用信息

使用不同 PWM 工作模式和功能的典型应用如下：

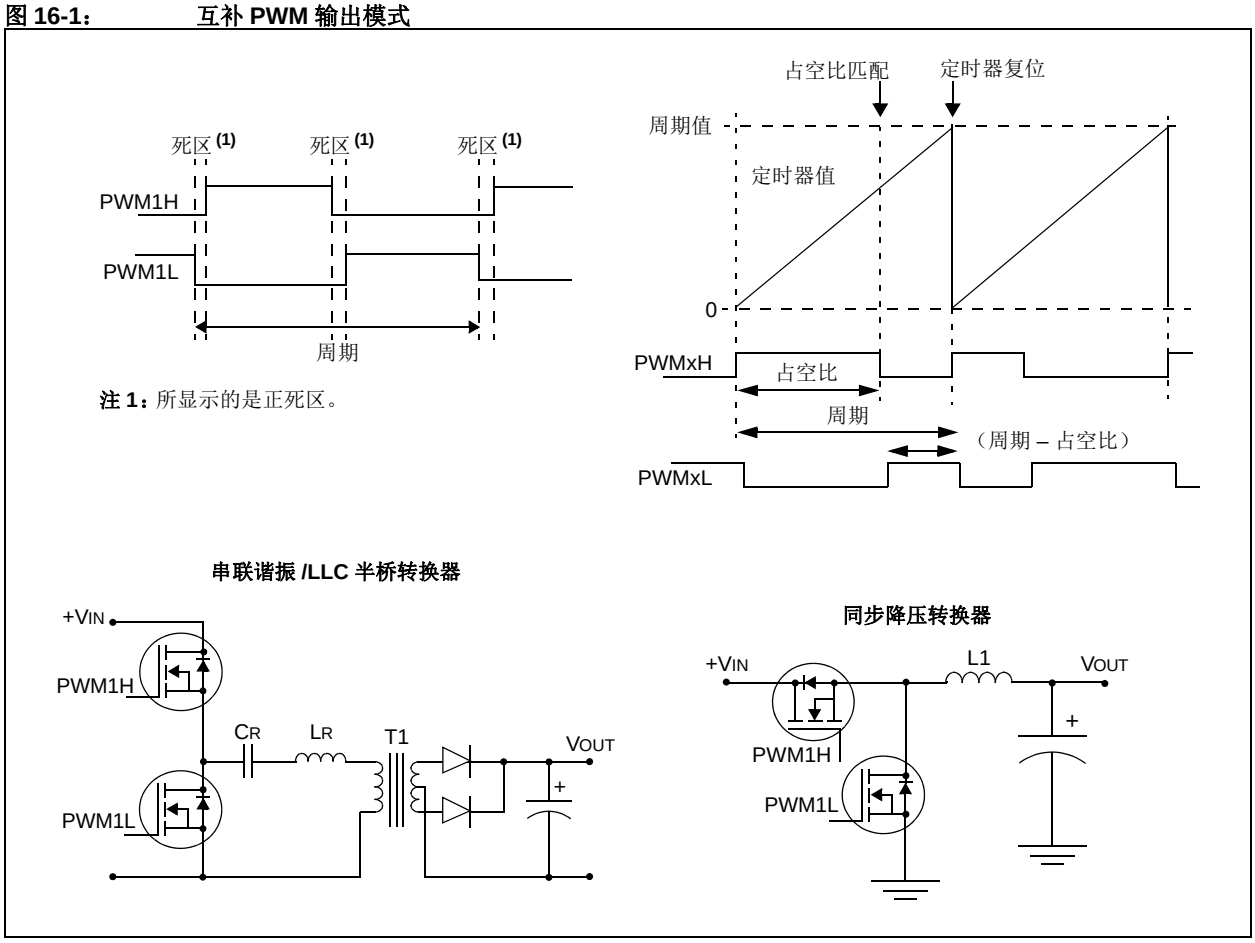
- 互补 PWM 输出模式
- 推挽 PWM 输出模式
- 多相 PWM 模式
- 可变相位 PWM 模式
- 电流复位 PWM 模式
- 恒定关断时间 PWM 模式
- 限流 PWM 模式
- 多调制方案实现模式
- 迟滞电流控制模式
- 突发模式实现

以下几节对每种应用进行了介绍。

16.1 互补 PWM 输出模式

互补 PWM 输出模式（如图 16-1 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 16-1： 互补 PWM 输出模式

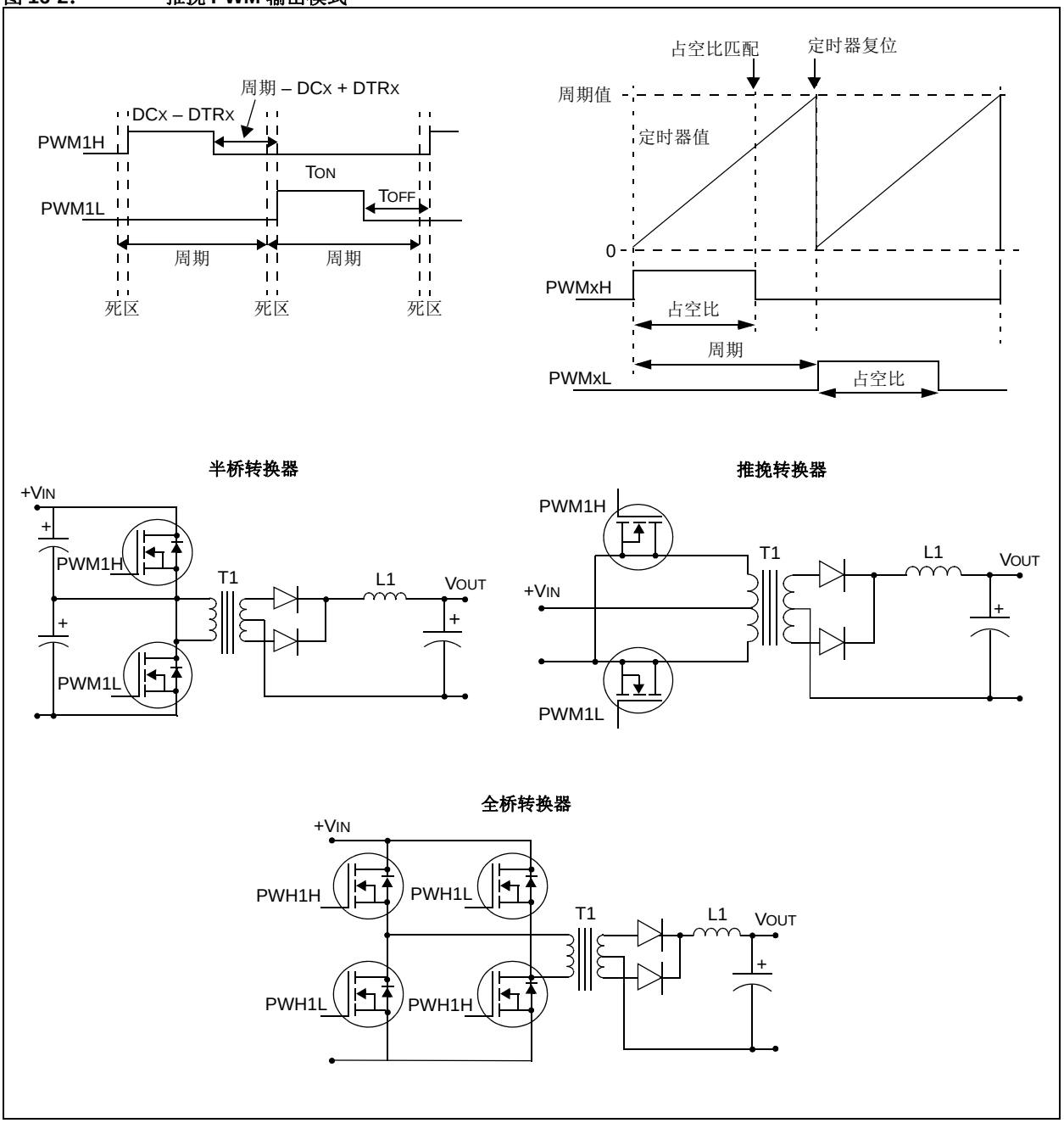


注 1： 所显示的是正死区。

16.2 推挽 PWM 输出模式

推挽 PWM 输出模式（如图 16-2 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 16-2： 推挽 PWM 输出模式



© 2011-2014 Microchip Technology Inc.

16.3 多相 PWM 模式

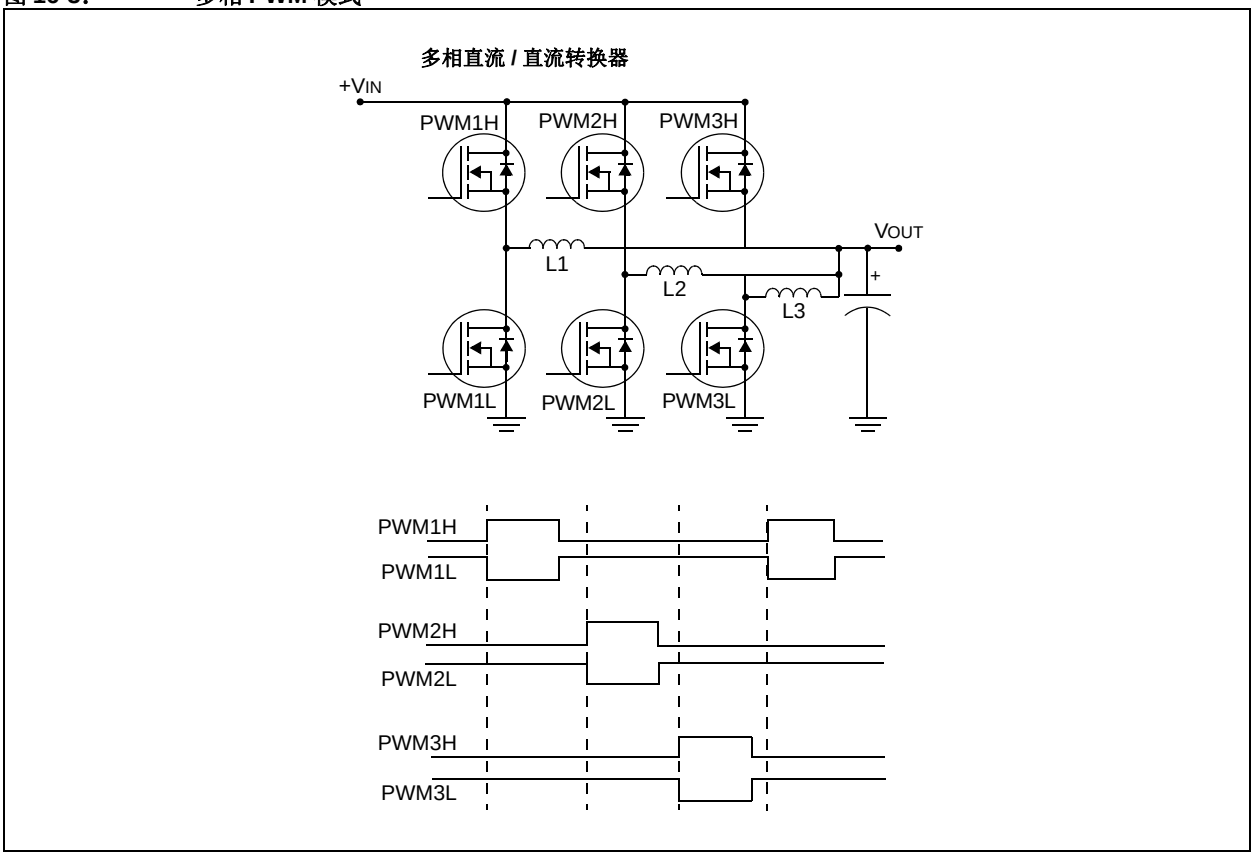
多相 PWM 模式（如图 16-3 所示）使用 PHASEx 寄存器中的相移值来相对于主时基对 PWM 输出进行偏移。由于相移值与主时基相加，所以相移后的输出会比指定零相移的PWM信号早出现。在多相模式下，指定的相移通过应用设计固定。在所有使用主控时基的 PWM 模式下，都可以使用相移。

16.3.1 多相降压稳压器

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 模式。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 16-3 所示）为 1 MHz。这种高开关频率可以极大地降低输入和输出电容尺寸要求；它还可以改善负载瞬态响应和纹波指标。

图 16-3： 多相 PWM 模式



16.3.2 交错式功率因数校正 （IPFC）

在 PFC 电路中交错使用多个升压转换器在近来的应用中变得非常流行。图 16-4 给出了典型的交错式 PFC 电路配置。图 16-5 给出了交错式 PFC 工作波形。

多通道交错式 PFC 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电流纹波。输入电流纹波降低意味着可以采用低差模 （Differential Mode， DM）噪声滤波器。一般认为， DM 噪声幅度降低时， DM 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 16-4： 交错式 PFC 图

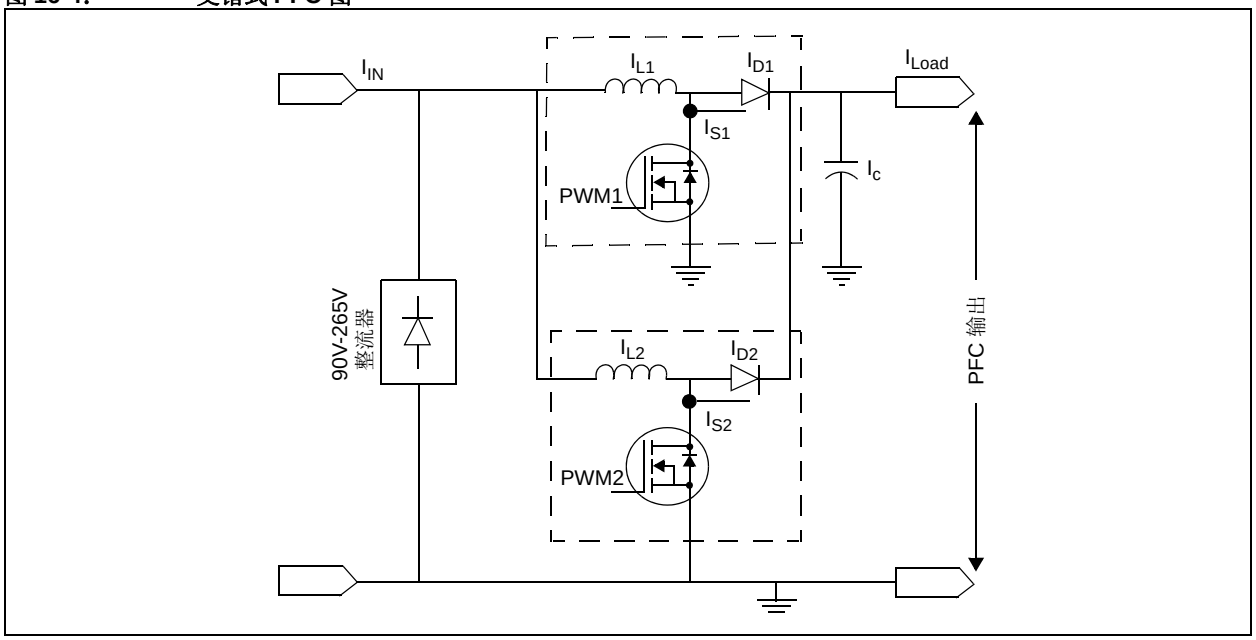
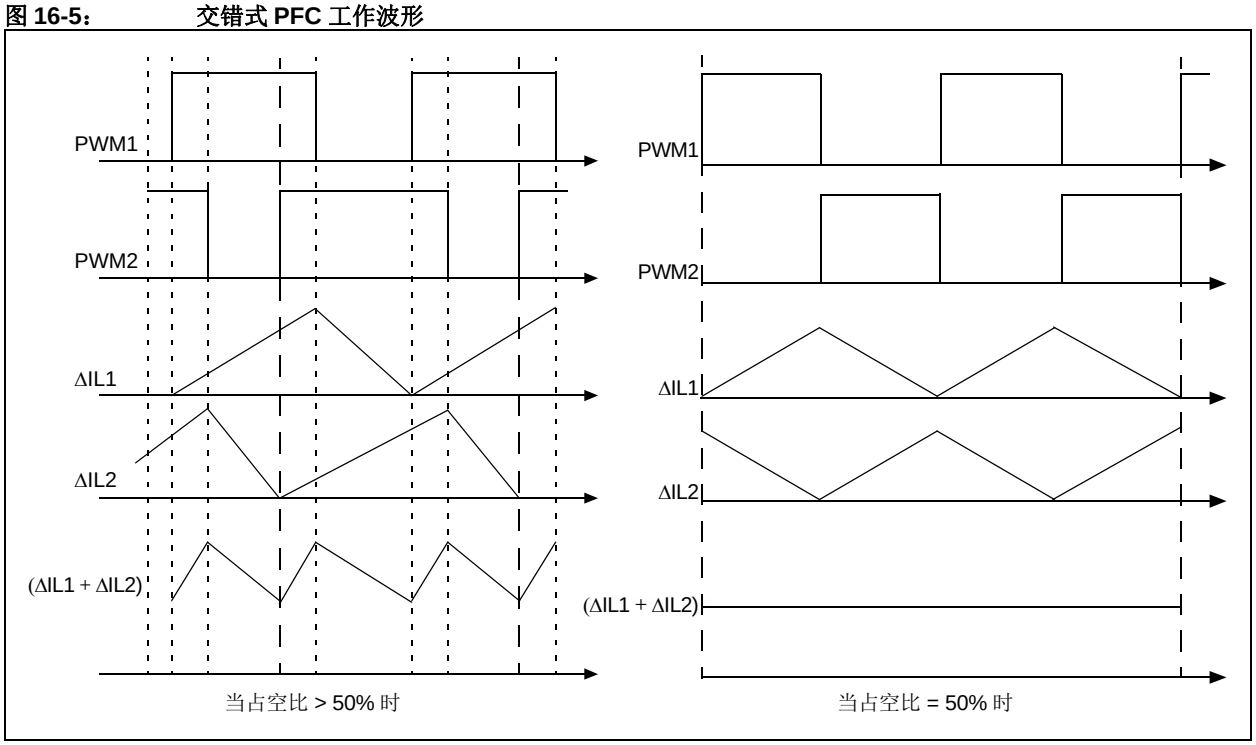


图 16-5： 交错式 PFC 工作波形



© 2011-2014 Microchip Technology Inc.

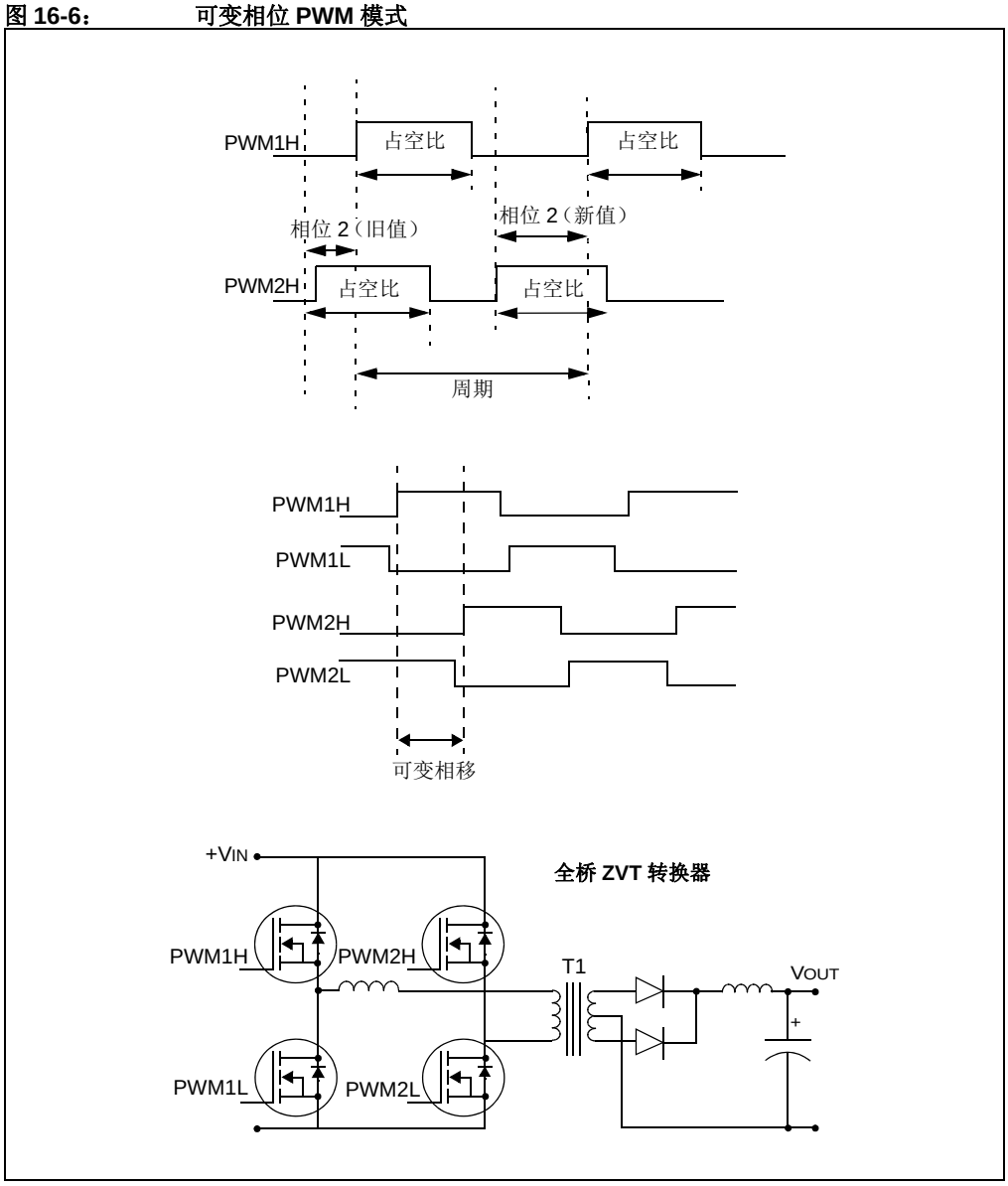
16.4 可变相位 PWM 模式

可变相位 PWM 模式 （如图 16-6 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拓扑中，使用了可变相位 PWM 模式。在标准 PWM 方法中，当晶体管在导通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 * TSW * FPWM*）是可观的。

零电压开关（ZVS）和零电流开关（ZVC）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 16-6： 可变相位 PWM 模式



16.5 电流复位 PWM 模式

电流复位 PWM 模式（如图 16-7 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。要在电流复位 PWM 模式下工作， PWM 发生器必须处于独立时基模式。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 PWM 模式下，本地时基复位将在 PWMxH/L 占空比完成之后基于限流输入信号的前沿产生。

在电流复位 PWM 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 模式输出，在 PFC 应用中在临界导通模式下使用。

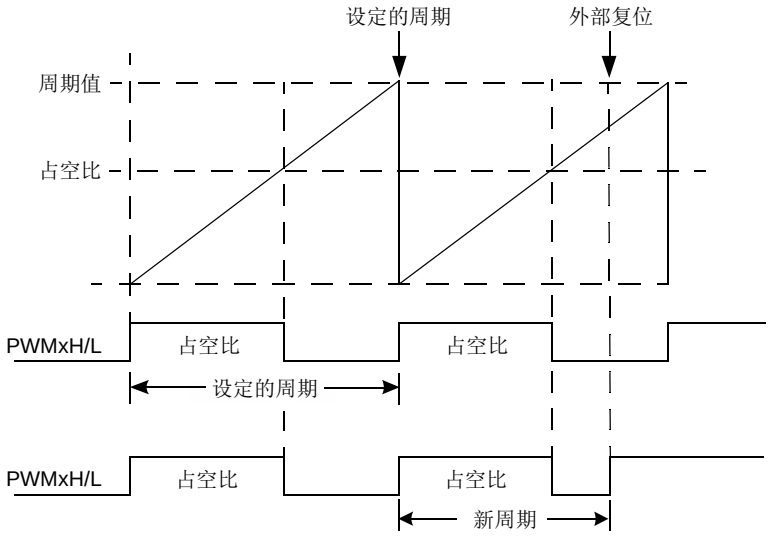
用户不应将它与逐周期限流 PWM 输出混淆，在后者中 PWM 输出置为有效，外部电路产生电流故障，PWM 信号在设定的占空比正常关断它之前关断。此时，PWM 频率对于给定时基周期是固定的。

电流复位 PWM 模式在 PFC 应用中的优点如下：

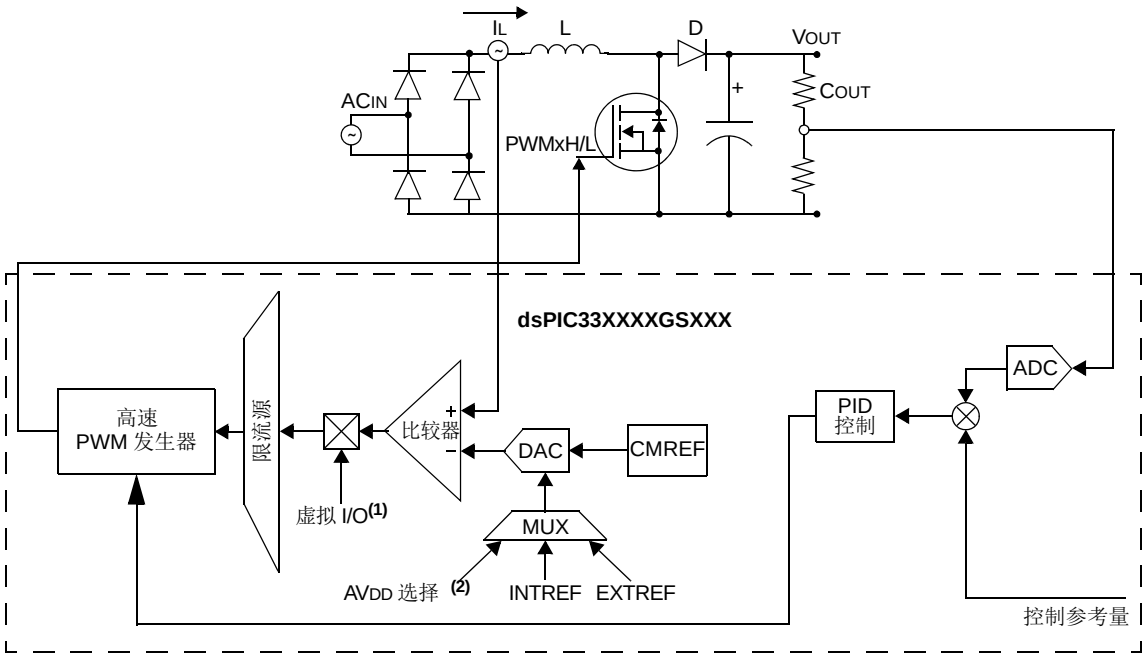
- 由于 PFC 升压电感不需要在每个开关周期结束时存储能量，所以可以使用较小的电感。使用较小的电感可以降低成本。
- 升压二极管从导通到关断的换流在电流为零时发生。可以通过使用速度较慢的二极管来降低成本。
- 由于每个周期都接收到反馈，所以内部电流反馈环的速度要快得多。

© 2011-2014 Microchip Technology Inc.

图 16-7：　　电流复位 PWM 模式



注：　外部电流比较器复位 PWM 计数器。PWM 周期早于设定的周期重新开始。这是恒定开启时间可变频率 PWM 模式。



注　1：　仅适用于具有可重映射 I/O 的器件。

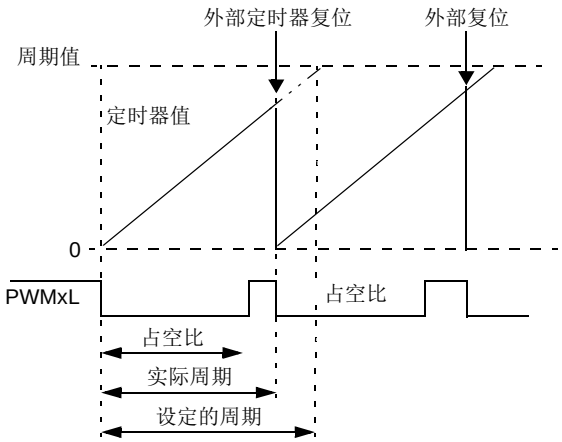
2：　该电压可以为 AVDD 或 AVDD/2，具体取决于器件型号。关于模拟比较器 DAC 参考电压的更多信息，请参见具体器件的数据手册。

16.6 恒定关断时间 PWM 模式

恒定关断时间 PWM 模式（如图 16-8 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 模式的开启时间 PWM 模式输出，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 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 寄存器中指定的值。

图 16-8： 恒定关断时间 PWM 模式



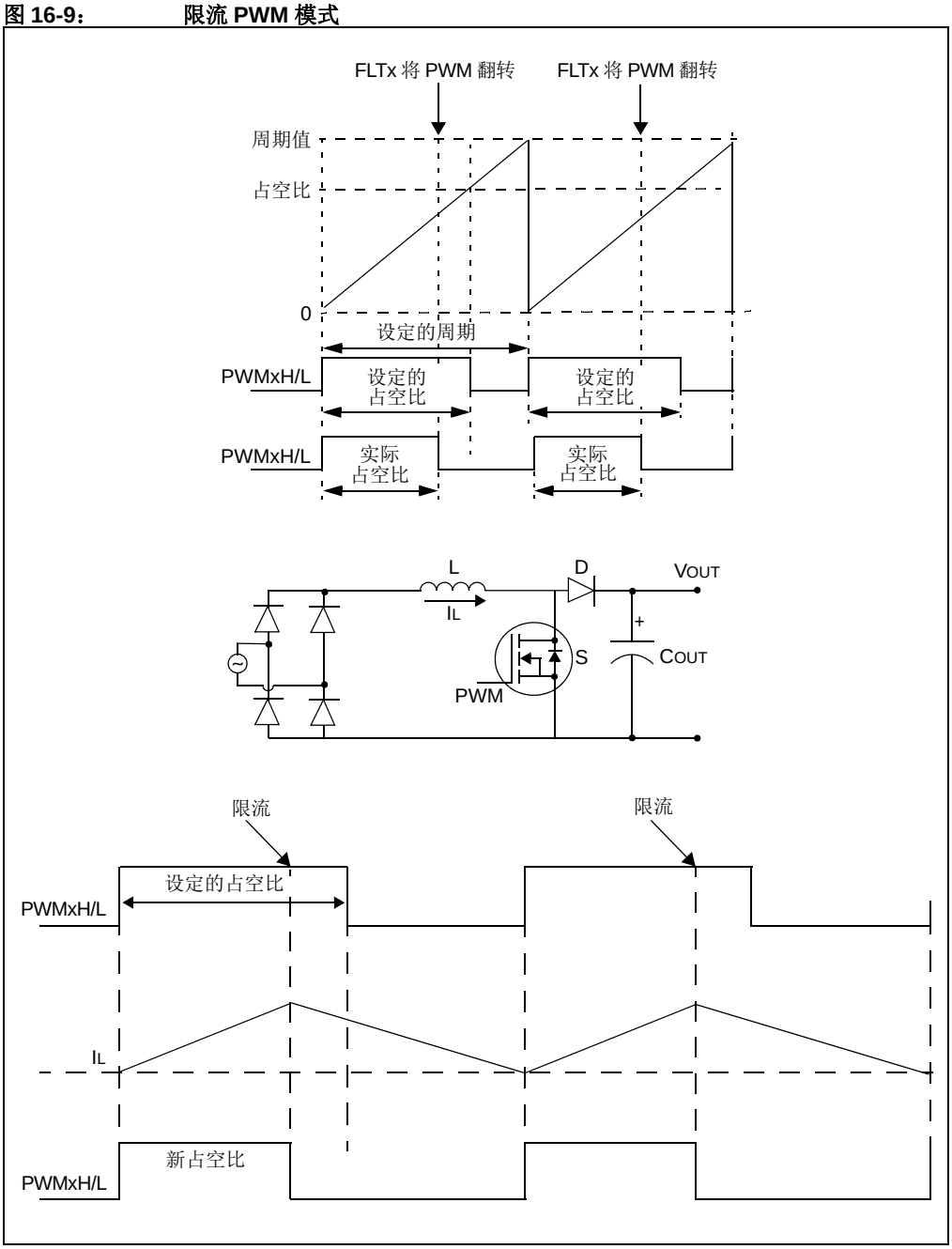
注： 占空比代表关断时间。

© 2011-2014 Microchip Technology Inc.

16.7 限流 PWM 模式

逐周期限流（如图 16-9 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 16-9： 限流 PWM 模式



16.8 多调制方案实现模式 （PWM + PFM）

具有主主控时基和辅助主控时基 （PTPER/STPER）的器件支持同时实现 PWM 转换器和脉冲频率调制 （PFM）转换器。因此，可以使用单个器件实现对两个涉及不同调制策略的转换级的控制。这种功能非常有用，特别是在所有电源转换级中都使用交错式转换器时。在其他情况下，仍然可以通过独立时基设置 （PWMCONx 寄存器中的 ITB 位）仅使用主时基实现 PWM 和 PFM。

一些拓扑（如谐振转换器）通常使用 PFM 进行控制。PFM 可以通过调节 PWM 时基周期寄存器（对于主时基为 PTPER/STPER，对于本地时基为 PHASEx/SPHASEx）来实现。在许多应用中，通常需要在输入和 / 或输出上耦合一个或多个电源转换级，它们使用脉宽（占空比）调制（PWM/PDM）进行控制。很多时候，这种应用要求使用单个控制器器件来控制包含 PWM 转换器和 PFM 转换器的多个转换级。例如，可以使用主主控时基为脉宽调制的固定频率转换级提供时钟，使用辅助时基来驱动 PFM 调制的谐振转换器级。

举例来说，假设这样一种情形：交错式 PFC 转换器（IPFC 级）连接到交流电源，IPFC 级的输出用作一个隔离的交错式谐振转换器级的输入。例 16-1 给出了 PWM-IPFC 级和 PFM 交错式隔离半桥谐振转换器级的 PWM 配置的代码示例。在例 16-1 中，假定 IPFC 级包含两个升压转换器，以 180 度的相移并行工作。PFM 交错式隔离半桥转换器级包含两个以 90 度相移工作的交错式谐振转换器 （因为交错操作是为了减少隔离变压器次级侧中二极管整流级后的输出电容处的纹波）。除了使用单个器件控制 PFM 转换器和 PWM 转换器之外，多个主控时基还可以支持需要两个以不同开关频率运行的 PWM 转换器的应用。

例 16-1： IPFC+ 交错式谐振转换器实现的初始化软件

```
/* Interleaved PFC Stage controlled by PWM Generator#1 */

PWMCON1bits.MTBS = 0;       /* Primary Master time base selected */
IOCON1BITS.PMOD = 3;        /* True Independent Time Base mode selected, PWM1H Controlling IPFC-Boost1
                               MOSFET and PWM1L Controlling IPFC-Boost2 */

PTPER = 9615;               /* Select 100 kHz switching frequency */
PHASE1 = (PTPER>>1);        /* Provide 180 deg Phase shift between the interleaved Converters */
SPHASE1 = 0;

PDC1 = (PTPER>>2);          /* Initialize duty cycle of IPFC-Boost1 to 25% */
SDC1 = (PTPER>>2);          /* Initialize duty cycle of IPFC-Boost2 to 25% */

/* Interleaved Half bridge resonant converter controlled by PWM Generator#2 and PWM Generator#3 */

PWMCON2bits.MTBS = 1;       /* Secondary Master time base selected */
IOCON2bits.PMOD = 0;        /* Complementary mode selected for PWM2H and PWM2L */

STPER = 3000;               /* Initialize to 320 kHz switching frequency */
PDC2 = (STPER>>1);          /* Set 50% duty cycle for symmetric voltage of transformer primary */
DTR2 = 200;                 /* Provide dead-time between complementary switches */
ALTDTR2 = 200;              /* Provide dead-time between complementary switches */

PWMCON3bits.MTBS = 1;       /* Secondary Master time base selected */
IOCON3bits.PMOD = 0;        /* Complementary mode selected for PWM3H and PWM3L */

PHASE3 = (STPER>>2);        /* Provide a Phase shift of 90 deg for interleaving action at output of
                               secondary side rectifier circuit */
PDC3 = (STPER>>1);          /* Set 50% duty cycle for symmetric voltage of transformer primary */

DTR3 = 200;                 /* Provide dead-time between complementary switches */
ALTDTR3 = 200;              /* Provide dead-time between complementary switches */
```


© 2011-2014 Microchip Technology Inc.

16.9 迟滞电流控制模式

在低功耗应用中（如功率因数校正），通过将电感电流控制在限流上限和限流下限内来实现连续导通工作模式。这种应用会产生可变频率工作模式，这种控制方案称为迟滞电流控制模式。可以使用两个高速模拟比较器来实现迟滞电流控制模式。为了实现迟滞电流控制模式， PWM 模块需要同时使用逐周期故障限制模式和电流复位模式。

举例来说，假设一个升压转换器需要使用迟滞电流控制模式来控制其电感电流，如图16-10所示。

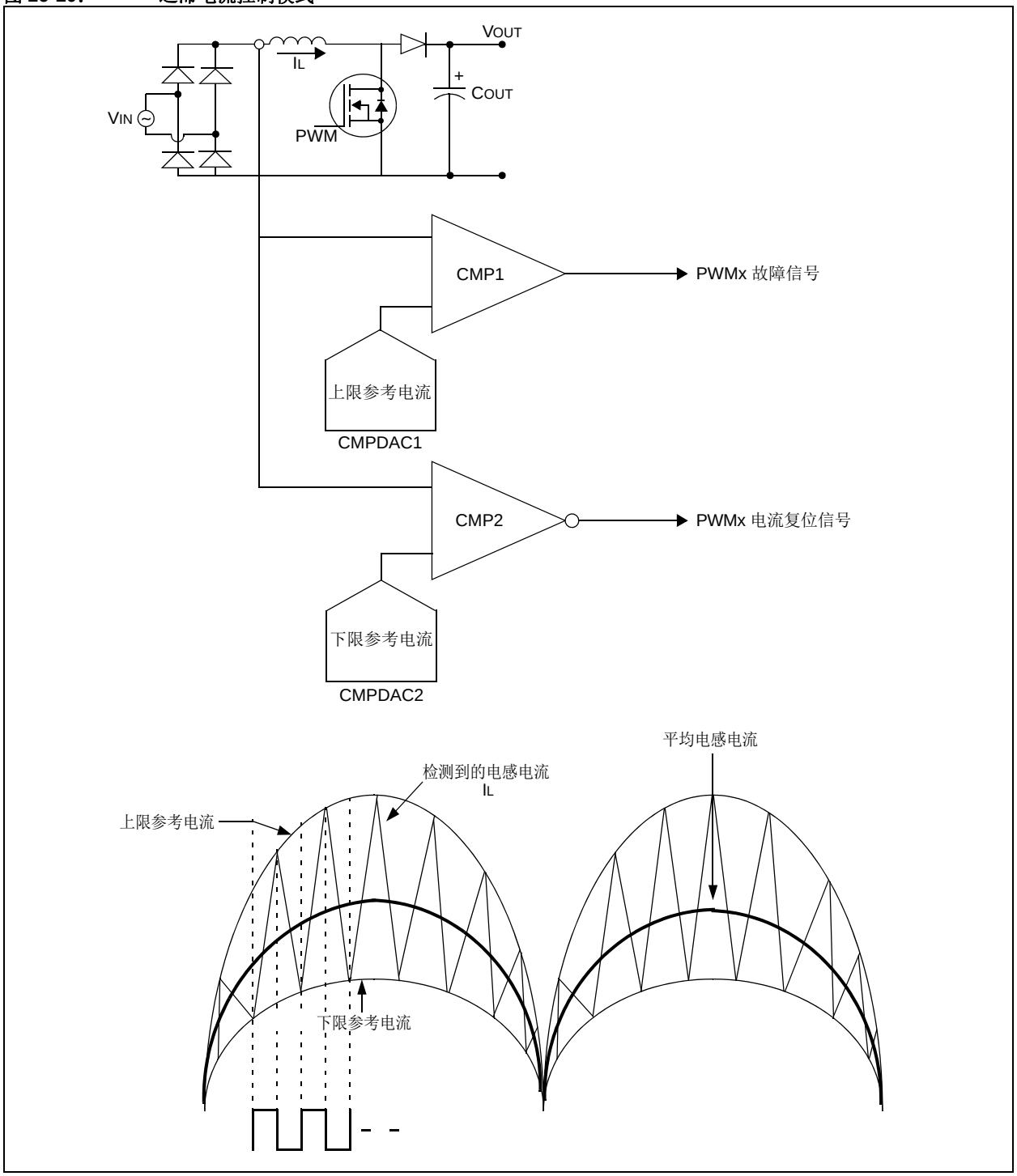
当 MOSFET 导通时，电感电流会上升。当电流达到上限（在第一个比较器的 DAC 中配置）时，PWM 输出会被设为低电平， MOSFET 会被关闭 （PWM 故障源配置为第一个比较器的输出），并且通过电感的电流会开始下降。当电流达到下限时，使用第二个比较器 （以反相极性模式配置）检测这种情况，并且第二个比较器的输出用作复位 PWM 周期的信号。例 16-2 给出了对于具有可重映射 I/O 的器件， PWM 和比较器模块的初始化代码段。

例 16-2： 迟滞电流控制模式的初始化软件

```
/* Initializing PWM1 Generator for controlling MOSFET */

PWMCON1bits.ITB = 1;            /* Select independent time base for enabling XPRES */
PWMCON1bits.XPRES = 1;          /* Select Current Reset mode */

IOCON1bits.PMOD = 1;            /* Select Redundant mode since only PWM1H is being used for MOSFET */
IOCON1bits.FLTDAT = 0;          /* To make the PWM signals low during Fault condition */

FCLCON1bits.FLTSRC = 0b01101;   /* Select Analog Comparator1 as Fault Source for PWM1 */
FCLCON1bits.FLMOD = 1;          /* Select Cycle-by-cycle Fault mode for upper limit cut-off */
FCLCON1bits.CLSRC = 0b01110;    /* Select Analog Comparator2 as Current Limit Source for Current Reset
                                       of PWM1 */

/* Configuring ACMP1 for Upper Current Limit and ACMP2 for Lower Current Limit */

CMPCON1bits.Range = 1;          /* Set Maximum DAC output voltage to AVDD */
CMPDAC1 = 900;                  /* Configure to turn OFF MOSFET at 2.9V of comparator input (upper
                                   current reference) */
CMPCON1bits.CMPON = 1;          /* Turn ON Analog Comparator1 */


CMOCON2bits.Range = 1;          /* Set Maximum DAC output voltage to AVDD */
CMPCON2bits.CMPPOL = 1;         /* Invert output polarity of Analog Comparator 2 for lower limit
                                       current detection and PWM Reset */
CMPDAC2 = 100;                  /* Configure to reset PWM at 0.322V of comparator input (lower current
                                       reference */

CMPCON2bits.CMPON = 1;          /* Turn ON Analog Comparator2 */
```

注： 例 16-2 给出的是 dsPIC33EPXXGSXXX 系列器件的示例。

图 16-10：　　迟滞电流控制模式



16.10　突发模式实现

在从转换器汲取的负载电流远小于标称电流 / 转换器空载工作的应用中，可以通过强制转换器进入断续模式来降低从电源吸收的功率。它的实现方式是使用手动改写功能将PWM输出置为无效，并保持一段特定的时间。

通常情况下，可以基于输出电压稳压而将 PWM 转换器输出关闭一段时间，这可以显著降低空载功耗要求。

© 2011-2014 Microchip Technology Inc.

17.0 PWM 与其他外设互连

本节介绍 PWM 与其他外设（如 ADC、模拟比较器和中断控制器）的互连。大多数电源转换应用都需要将 PWM 模块与诸如高速 ADC（10 位或 12 位，取决于器件）和高速模拟比较器之类的其他外设进行紧密同步。由于电源转换应用的苛刻时序要求，这种互连必须在极低 CPU 开销或无 CPU 开销的情况下实现。这种互连还需要确保快速响应时间，通常需要达到纳秒量级。

高速 PWM 模块包含了许多增强功能，用于支持与高速 ADC 和高速模拟比较器模块直接互连。本节将介绍其中每种增强功能，并给出一些说明这些增强功能如何有益于电源转换应用的示例。

17.1 PWM 与 ADC 互连

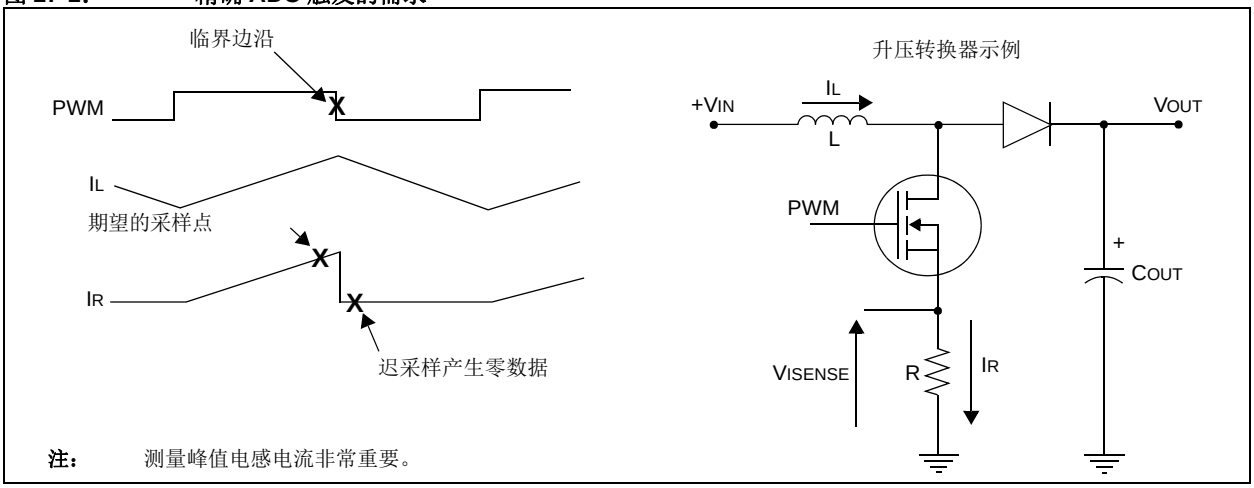
17.1.1 ADC 的精确触发

在数字电源中，ADC 用于反馈信号的测量。这些反馈信号可能具有复杂的波形或高噪声含量。因此， ADC 的精确触发是非常重要的。

错误的 ADC 触发可能会对电源转换器的工作产生很大影响。作为示例，图 17-1 给出了一个直流 / 直流升压转换器的图示，其中将电流传感器与功率 MOSFET 的源极引脚串联。这种配置可以无需使用具有高共模电压的差分放大器，因此提供了一种低成本的检测方案。这种方案的缺点是，ADC 只能检测 MOSFET 电流。

如果数字控制系统配置为测量峰值电流，则只要 ADC 触发存在很小的延时，得到的结果就会是 0x0000。产生该延时的原因可能是软件开销或 ADC 在采样时刻处于忙状态。

图 17-1： 精确 ADC 触发的需求



注： 测量峰值电感电流非常重要。

上述情形可以通过使用高速 PWM 模块灵活的 ADC 触发功能来防止。特殊事件触发器、主 PWM 触发器和辅助 PWM 触发器可以用于产生 ADC 转换请求，无需任何软件开销。该功能可以保证恰好在电路需要时精确地触发 ADC 转换。由于触发信号是在硬件中直接从 PWM 发送到 ADC 模块，所以该功能可以防止由于软件而导致的任何触发延时。

产生触发信号的确切时刻由用于特殊事件触发的 SEVTCMP 寄存器，或用于 PWM 主触发和辅助触发的 TRIGx 和 STRIGx 寄存器决定。关于 PWM 触发生成的更多信息，请参见**第 7.0 节“PWM 触发”**。

高速 ADC（10 位或 12 位，取决于不同的器件）提供了多个 S&H 电路来支持并行采样。该功能可以克服 ADC 在采样时刻处于忙状态的问题。关于配置 ADC 触发源的更多信息，请参见相应器件的数据手册和 《dsPIC33/PIC24 系列参考手册》中的 ADC 章节。

17.1.2 ADC 的 PWM 限流触发

图 17-1 中的示例也可以使用峰值电流模式控制来实现。在该方法中，通过限流功能来自动截断 PWM。虽然限流功能可以紧密地控制电流，但 PWM 下降沿的位置是无法预测的。因此，无法使用特殊事件触发器、主 PWM 触发器和辅助 PWM 触发器来有效地触发 ADC 转换。

这个问题可以通过使用 PWM 限流源直接产生 ADC 触发信号来解决。通过使用该功能，可以在 PWM 脉冲下降沿的精确时刻触发 ADC 转换。因此，可以在 PWM 信号的每个下降沿可靠地进行峰值电流测量。

17.2 PWM 与高速模拟比较器互连

17.2.1 比较器限流和故障

限流和故障功能可以用于逐个 PWM 周期地限制任意系统参数，包括电流、电压、功率或温度。模拟比较器提供了一种直接在硬件中截断 PWM 输出的独特方式。

PWM 脉冲截断的实现无需软件干预，并且可以设定为响应一个可变的门限值。模拟比较器还可以设定为选择反相极性。例如，在检测欠压条件或是否存在系统负载时，反相极性会很有用。

逐周期限流或故障与模拟比较器配合使用，还可以用于峰值电流模式控制。图 10-6 给出了在升压转换器应用中实现峰值电流模式控制的图示。

在某些情况下，需要使用锁定故障模式来保护系统硬件。高速 PWM 模块提供了锁定故障模式，该模式可以关闭 PWM 输出，直到用软件清除故障为止。模拟比较器可用于在比较器输入超出故障门限值时闭锁 PWM 输出。

锁定故障模式一个很好的使用示例就是进行短路保护。短路事件可能导致电源转换器发生灾难性损坏，因此，逐周期故障并不是最好的选择。此时可以选择将 PWM 输出无限期地闭锁，直到软件检测到故障清除为止。

关于如何将模拟比较器配置为 PWM 模块的限流或故障源的更多信息，请参见**第 10.1 节 “模拟比较器产生的 PWM 故障信号”**。

17.2.2 外部周期复位模式

外部周期复位模式类似于故障 / 限流操作，但其作用完全相反。该模式实际上会复位 PWM 周期，而不是关闭 PWM 输出，因此，会比所设定周期更早地重新启动 PWM。

第 16.5 节 “电流复位 PWM 模式”介绍了一个对于模拟比较器使用外部周期复位模式的示例。

© 2011-2014 Microchip Technology Inc.

17.3 PWM 与中断控制器互连

17.3.1 PWM 中断

PWM 中断可以基于 PWM 故障、限流或触发事件而产生。该功能对于需要在每次发生此类事件时执行特定软件的情况非常有用。

例如， PWM ISR 中可以包含应在 PWM 关闭之后执行的故障处理程序。诸如数据记录、故障的外部通信，或者故障恢复程序之类的任务可以在此处执行。

PWM 中断还可以用于执行控制算法，或者更新系统变量或控制参考量。

17.3.2 ADC 中断和 CPU 负载交错

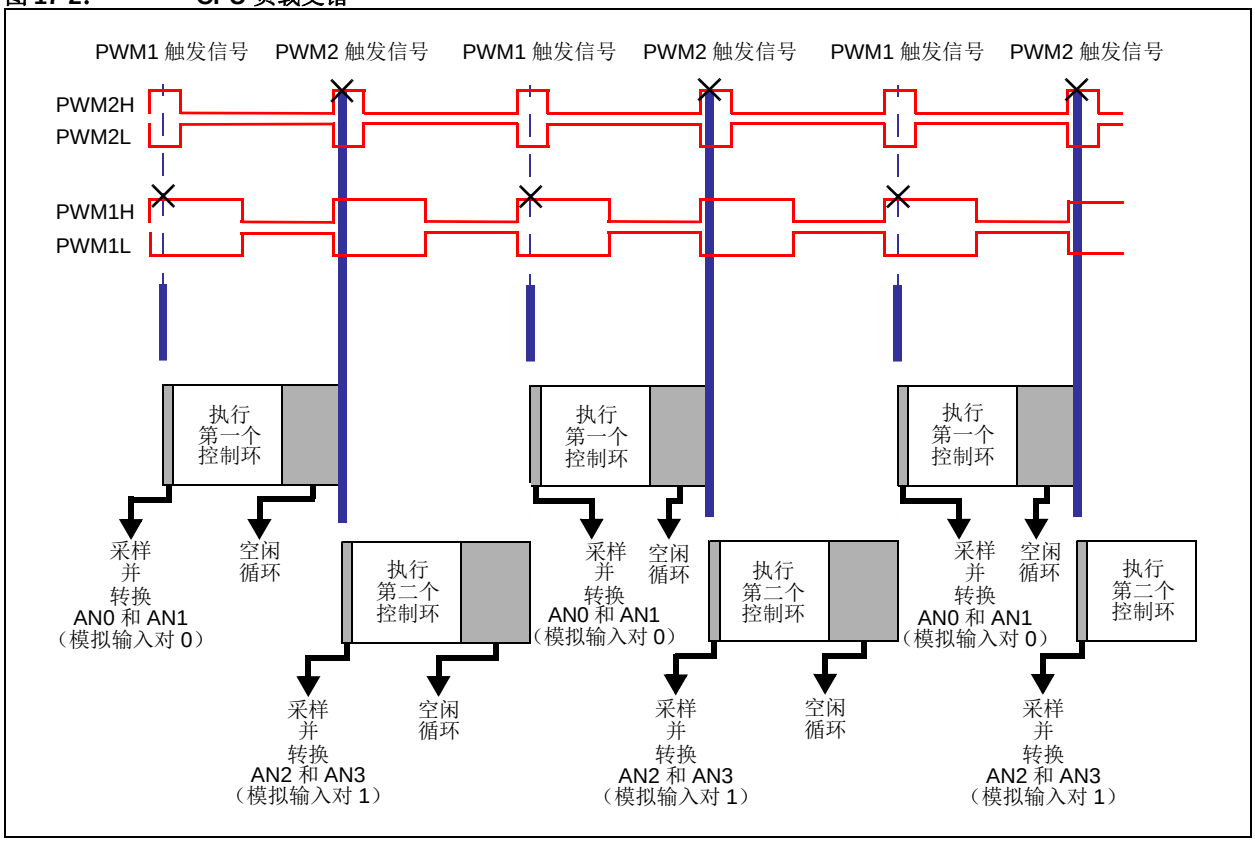
使用数字信号控制器 （DSC）进行电源转换的独特优点之一就是能够使用单个控制器控制多级。在同一器件上执行多个控制环时，必须小心地安排每个控制环的执行顺序，以避免在处理来自 ADC 数据时产生任何延时。

PWM 模块提供了一个触发分频器，可以每隔几个 PWM 周期产生 ADC 触发信号。除了该功能之外，还可以对第一个触发信号的生成进行延时，以错开可用 CPU 时间中的控制环。

图 17-2 显示了某个系统中控制环的执行顺序，该系统通过单个 dsPIC® DSC 同时控制两个电源转换器。

如图 17-2 所示， ADC 模拟输入对中断用于执行每个电源转换器级的控制算法。每个 ADC 模拟输入对的转换使用 PWM 触发信号进行触发。通过使用 TRGDIV<3:0> 位 （TRGCONx<15:12>），每个 PWM 触发信号将每隔 1 个 PWM 周期产生。来自 PWM2 的第一个触发信号通过使用 TRGSTRT<5:0> 位 （TRGCONx<5:0>）来延迟 1 个 PWM 周期产生。使用该配置时，每个电源转换器的控制环都间隔一个 PWM 周期交替执行，从而有效地利用 CPU 带宽。

图 17-2： CPU 负载交错



18.0 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 dsPIC33/PIC24 产品系列而编写的，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。当前与高速 PWM 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 dsPIC33/PIC24 系列器件的应用笔记和代码示例，请访问 Microchip 网站 （www.microchip.com）。

© 2011-2014 Microchip Technology Inc.

19.0 版本历史

版本 A （2008 年 2 月）

这是本文档的初始版本。

版本 B （2008 年 9 月）

该版本包括以下更新：

- 公式：
  - 更新了**第 6.0 节 “PWM 发生器”**中的公式 6-1
  - 更新了**第 6.2.3 节 “辅助占空比 （SDCx）”**中的公式 6-3
- 示例：
  - 在**第 5.1 节 “PWM 时钟选择”**中增加了一个关于 PWM 时钟代码的示例
- 图片：
  - 更新了图 5-4 中的标注
  - 在**第 6.7 节 “死区分辨率”**中包含了新图 （见图 6-4）
  - 更新了图 10-1 和图 10-4 中的故障源值
- 标题：
  - 在**第 5.0 节 “模块说明”**中增加了“附属 PLL”新章节（见**第 5.1 节 “PWM 时钟选择”**）
  - 在**第 6.6 节 “死区失真”**中更正了关于 “死区失真”的描述
  - 在 “中心对齐模式”中增加了一个关于 “死区插入”的新章节 （见**第 6.8 节 “中心对齐模式下的死区插入”**）
  - 在**第 10.0 节 “PWM 故障引脚”**中增加了一个关于 “PWM 故障发生器”的新小节（见**第 10.1 节 “模拟比较器产生的 PWM 故障信号”**）
- 注：
  - 在**第 5.1 节 “PWM 时钟选择”**中增加了关于 PWM 标称输入时钟的 “注”？
  - 在以下寄存器中增加了关于 PWM 分辨率边界条件的 “注”：
    - MDC：PWM 主控占空比寄存器 （见寄存器 3-10 中的 “注 2”）
    - PDCx：PWM 发生器占空比寄存器 （见寄存器 3-12 中的 “注 2”）
    - SDCx：PWM 辅助占空比寄存器 （见寄存器 3-13 中的 “注 2”）
  - 在寄存器 3-22 中增加了关于对限流模式使用故障 1 的 “注”（CLSRC<4:0> = b0000）（见 “注 2”）
  - 在**第 5.1 节 “PWM 时钟选择”**中增加了关于配置附属时钟的 “注”
  - 在**第 16.5 节 “电流复位 PWM 模式”**中增加了关于复位本地时基的 “注”
- 寄存器：
  - 更正了 **PDCx：PWMx 发生器占空比寄存器**和 **SDCx：PWMx 辅助占空比寄存器**的寄存器说明
  - 更正了寄存器 3-22 中 bit 14-10 和 bit 7-3 的位说明
  - 在 “LEBCONx：前沿消隐控制寄存器”中，将 LEB 的位域值更新为 LEB<4:0> 和 LEB<6:5> （见寄存器 3-23）
  - 在 “PWMCAPx：主 PWM 时基捕捉寄存器”中更正了 bit 3 至 bit 15 的读 / 写状态（见寄存器 3-27）)
- 章节：
  - 在整篇文档中，将 “互补 PWM 输出模式”和 “互补 PWM 模式”这两个用词更正为 “互补模式”
  - 在整篇文档中，将 “推挽 PWM 输出模式”和 “推挽 PWM 模式”这两个用词更正为 “推挽模式”
- 对整篇文档的文字和格式进行了更改

版本 C （2010 年 3 月）

- 公式：
  - 更新了以下公式：公式 5-1 以及公式 5-3 至公式 6-5
  - 增加了以下公式：公式 5-2 和公式 11-2
- 示例：
  - 更新了以下示例：
    - 例 5-1、例 5-4、例 6-2、例 6-3 和 例 10-4
    - 在例 5-2 中进行了以下更新：更新了示例，并将示例的位置调整到例 5-1 之后
    - 在例 6-4 和例 6-5 中进行了以下更新：更新了示例，并将示例的位置从**第 6.2.4 节 “占空比分辨率”**调整到**第 6.2.3 节 “辅助占空比 （SDCx）”**
  - 增加了以下示例：例 5-3、例 6-3、例 6-6、例 10-2、例 10-3 和 例 12-5
- 图片：
  - 更新了以下图：图 5-5、图 5-7、图 5-8、图 7-2 至图 7-9、图 10-1、图 10-4 以及图 11-1 至图 16-9
  - 增加了以下图：图 5-1、图 5-2、图 5-6、图 10-6、图 16-4 和 图 16-5
- 注：
  - 增加了关于指导用户将系列参考手册章节和数据手册用作共同参考的信息的 “注”（见**第 1.0 节 “简介”**上方的 “注”）
  - 在寄存器 3-2 和寄存器 3-3 中增加了 “注 2”
  - 在寄存器 3-4 中增加了 “注 1”
  - 在寄存器 3-11 中增加了 “注 5”
  - 在寄存器 3-14 中进行了以下更新：
    - 在 “注 1”和 “注 2”中增加了一条小注
    - 删除了 “注 2”中的一条小注
  - 在寄存器 3-15 中进行了以下更新：
    - 更新了 “注 1”中的第二条小注
    - 更新了 “注 2”中的小注
  - 在寄存器 3-16 和寄存器 3-17 中，更新了 bit 13-0 的位文字说明
  - 在寄存器 3-18 中，删除了 bit 7 的注释引用，并删除了以下注释：辅助 PWM 发生器无法产生 PWM 触发中断
  - 在寄存器 3-19 中增加了 “注 2”
  - 在寄存器 3-21 中增加了 “注”
  - 更新了寄存器 3-22 中的 “注 1”
  - 在寄存器 3-23 中增加了 “注 1”
  - 在寄存器 3-27 中增加了 “注 3”和 “注 4”
  - 在**第 5.1 节 “PWM 时钟选择”**中更新了以下 “注”：如果使用主 PLL 作为附属时钟的来源，则主 PLL 的最大工作频率应配置为 FCY = 30 MHz 或更低， FVCO 必须处于 112 MHz - 120 MHz 范围内
  - 在**第 5.6 节 “时基同步”**中增加了 “注 1”至 “注 7”
  - 在**第 6.7 节 “死区分辨率”**中增加了关于占空比值的 “注”
  - 在**第 7.0 节 “PWM 触发”**中增加了关于动态触发的 “注”
  - 在表 9-1 中删除了以下 “注”：在独立时基模式下， PWMxH 占空比通过 MDC 或 PDCx 进行控制， PWMxL 占空比通过 MDC 或 SDCx 进行控制
  - 在表 9-2 中删除了以下 “注”：在独立输出时基模式下， PWMxH 占空比通过 MDC 或 PDCx 进行控制， PWMxL 占空比通过 MDC 或 SDCx 进行控制

© 2011-2014 Microchip Technology Inc.

版本 C （2010 年 3 月）（续）

- 在**第 14.2 节 “空闲模式下的高速 PWM 操作”**中增加了关于节能模式的 “注”
- 更新了**第 16.5 节 “电流复位 PWM 模式”**中的 “注”

• 寄存器：
- 在**第 3.0 节 “控制寄存器”**中，更新了 “PWMCAPx：主 PWM 时基捕捉寄存器”的寄存器说明
- 在寄存器 3-11 中，将 bit 0 对应于 “0”的文字说明中的 “PDCx”一词更正为 “MDC/PDCx/SDCx/PHASEx/SPHASEx”
- 在寄存器 3-19 中，对于 bit 3-2、 bit 5-4 和 bit 7-6，将 “数据”一词更正为 “状态”
- 将寄存器 43-17：“STRIGx：PWM 辅助触发比较值寄存器”调整到寄存器 3-20 之后，作为寄存器 3-21
- 在寄存器 3-23 中，将 bit 9-3 的位文字说明更正为 “消隐可以按照 8.32 ns 的步阶递增”

• 章节：
- 在**第 1.0 节 “简介”**中，向高速 PWM 的常见应用中增加了 “交错式功率因数校正（IPFC）”
- 在**第 2.0 节 “特性”**中，对高速 PWM 主要功能列表进行了以下更新：
  • 删除了 “PWM 捕捉功能”
  • 将 “每个 PWM 周期可从 PWM 向模数转换器 （ADC）产生双触发”更新为 “每个 PWM 周期可向模数转换器 （ADC）产生双触发”
  • 将 “可重映射 PWMxH 和 PWMxL 引脚”更新为 “可重映射 PWM4H 和 PWM4L 引脚”
- 在**第 5.1 节 “PWM 时钟选择”**中进行了以下更新：
  • 在第一段中增加了 “主 PLL 输出 （FVCO）”
  • 将以下说明中的 “PLLCLK”一词更正为 “FVCO”：当器件运行于主 PLL 模式时，可以从系统时钟产生 PWM 模块的附属时钟 公式 5-3 给出了主 PLL 时钟 （FVCO）频率与附属时钟 （ACLK）频率之间的关系
- 增加了**第 5.4.1 节 “中心对齐模式在 UPS 应用中的优点”**
- 在**第 5.6 节 “时基同步”**中进行了以下更新：
  • 将脉冲宽度 “130 ns”更正为 “200 ns”
  • 增加了以下说明：SYNCO 信号会驱动 200 ns，以确保其他器件可靠地检测到信号
- 在**第 5.7 节 “特殊事件触发器”**中，将事件 “当 PTEN = 0 时”更新为 “当 PTCON<PTEN> = 0 时”
- 在**第 5.8 节 “独立 PWM 时基”**中删除了以下说明：在独立时基模式下， PHASEx 和 SPHASEx 寄存器提供 PWMx 输出 （PWMxH 和 PWMxL）的时间周期值
- 在**第 6.3 节 “死区发生”**中进行了以下更新：
  • 增加了以下说明：独立 PWM 输出模式不支持死区
  • 从说明中删除了 **“门控”**
- 在**第 6.4 节 “死区发生器”**的 **“负死区”**小节中增加了以下说明：只有对于互补 PWM 输出信号，才能指定负死区
- 在**第 6.7 节 “死区分辨率”**中删除了以下说明：如果器件不实现高分辨率 PWM 选项，且 PWM 时钟预分频器分辨率为 1.04 ns、 2.08 ns 或 4.16 ns，则可能的最高死区分辨率为 8.32 ns

© 2011-2014 Microchip Technology Inc.

版本 C （2010 年 3 月）（续）

- 在**第 7.0 节 “PWM 触发”**中，将 “TRGCONx 寄存器中的双触发模式位 （DTM7）” 更新为 “PWM 触发控制寄存器中的双触发模式位 DTM （TRGCONx<DTM> = 7）”
- 在**第 10.4 节 “故障退出”**中进行了以下更新：
  - 删除了以下说明：下一个 PWM 周期在 PTMR 值为 0 时开始
  - 更新了步骤 “c)”
- 在**第 10.5 节 “PMTMR 被禁止时的故障退出”**中，将 **“FSTAT”**一词更新为 **“FLTSTAT”**
- 在**第 10.7 节 “PWM 限流引脚”**中进行了以下更新：
  - 将 **“该行为称为电流复位模式，一些功率因数校正 （PFC）应用中会使用该模式”这一说明**更新为 **“更多详细信息，请参见第 16.5 节 “电流复位 PWM 模式””**
  - 增加了**第 10.7.2 节 “在逐周期模式下配置模拟比较器”**
- 在**第 11.1 节 “前沿消隐 （LEB）”**中进行了以下更新：
  - 将 **“8.4 ns”**更新为 **“8.32 ns”**
  - 增加了以下说明：在高速开关应用中，开关 （如 MOSFET/IGBT）通常会产生极大的瞬变。这些瞬变可能会导致会产生问题的测量误差。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晶体管开关所导致的瞬变。
- 在**第 11.2 节 “独立时基捕捉”**中，将 **“电流模式控制”一词**更正为 **“限流 PWM 控制”**
- 在**第 12.4 节 “故障 / 限流改写和死区逻辑”**中进行了以下更新：
  - 更正了说明中的以下几个用词：将 **“低电平”**更新为 **“无效”**，并将 **“影响”**更新为 **“帮助”**
  - 在说明中增加了 **“驱动为有效状态”**
- 增加了**第 12.6 节 “PENx （GPIO/PWM）所有权”**
- 在**第 15.0 节 “独立时基的外部控制 （电流复位模式）”**中进行了以下更新：
  - 将标题 **“独立时基的外部控制”**更新为 **“独立时基的外部控制 （电流复位模式）”**
  - 在说明中增加了 **“迟滞和临界导通模式”**
- 将**第 16.3 节 “多相 PWM 模式”**中的第二段调整为新的小节**第 16.3.1 节 “多相降压稳压器”**
- 增加了**第 16.3.2 节 “交错式功率因数校正 （IPFC）”**
- 在**第 16.5 节 “电流复位 PWM 模式”**中，增加了 “电流复位模式在 PFC 应用中的优点”
- 增加了**第 16.10 节 “突发模式实现”**

- 表格：
  - 更新了以下表格：表 9-1 和 表 9-2
  - 增加了以下表格：表 9-3 至 表 10.7.2
- 在本系列参考手册中，将对 **“dsPIC33F”**的具体引用更新为 **“dsPIC33F/PIC24H”**
- 将系列参考手册名称 **“dsPIC33F 第 43 章 高速 PWM”**重命名为 **“dsPIC33F/PIC24H 第 43 章 高速 PWM”**
- 对整篇文档的文字和格式进行了更改

© 2011-2014 Microchip Technology Inc.

版本 D （2011 年 3 月）

该版本包括以下更新：

- 在**第 3.0 节 “控制寄存器”**中，更新了 PTCON2、 PHASEx 和 SPHASEx 寄存器的定义
- 在 PTCON 寄存器中增加了 “注 2”和 “注 3”（见寄存器 3-1）
- 在 SEVTCMP 寄存器下的阴影注释中增加了 “注 2”和 “注 3”（寄存器 3-4）
- 从 STCON 寄存器中删除了 “注 1”（寄存器 3-5）
- 在 SSEVTCMP 寄存器中增加了 “注 1”、“注 2”和 “注 3”（寄存器 3-8）
- 在 MDC 寄存器下的阴影注释中增加了新的 “注 2”（寄存器 3-10）
- 在 PDCx 寄存器下的阴影注释中，更新了 “注 1”，增加了新的 “注 3”（寄存器 3-12）
- 在 SDCx 寄存器下的阴影注释中，更新了 “注 1”，增加了新的 “注 3”（寄存器 3-13）
- 在 PHASEx 寄存器下的阴影注释中，更新了 “注 1”和 “注 2”（寄存器 3-15）
- 在 IOCONx 寄存器的 CLDAT<1:0> 位中增加了对于 “注 2”的引用 （寄存器 3-19）
- 在 LEBCONx 寄存器中，更新了 “注 1”，并更新了 LEB<6:0> 位的位定义 （寄存器 3-23）
- 更新了 PWMCAPx 寄存器阴影注释中的 “注 4”（寄存器 3-27）
- 更新了**第 4.0 节 “架构概述”**中第四段的第一句
- 更新了高速 PWM 模块架构概览图 （见图 4-1）
- 在振荡器系统图的 “附属时钟产生”模块中，向 FVCO 引用中增加了 “最高 120 MHz” （图 5-1）
- 更新了 “使用 FVCO 作为附属时钟源”中的代码 （见例 5-3）
- 更新了**第 5.2 节 “时基”**中的预分频比选项
- 在 “边沿对齐或中心对齐模式选择”中，更新了注释，并增加了一行用于使能独立时基的代码 （见例 5-4）
- 更新了**第 5.7 节 “特殊事件触发器”**。
- 更新了 “ADC 特殊事件触发配置”中的代码 （见例 5-7）
- 更新了**第 5.8 节 “独立 PWM 时基”**
- 更新了**第 6.1 节 “PWM 周期”**中的第二段、第四段和第六段，以及第一个项目符号项
- 更新了 “时钟预分频比选择”中的第二个注释 （见例 6-1）
- 在 “PWM 时间周期初始化”的三行代码中增加了注释 （见例 6-3）
- 更新了**第 6.2 节 “PWM 占空比控制”**中的第一段
- 更新了**第 6.2.1 节 “主控占空比 （MDC）”**中的第一段
- 更新了**第 6.2.2 节 “主占空比 （PDCx）”**中的第一段
- 在 “主占空比比较”中，将 PWMx 信号引用更改为 PWMxH 和 / 或 PWMxL （见图 6-1）
- 更新了**第 6.2.3 节 “辅助占空比 （SDCx）”**中的第一段
- 在 “辅助占空比比较”中，将 PWMx 信号引用更改为 PWMxL，并更新了 “注”（见图 6-2）
- 更新了**第 6.2.4 节 “占空比分辨率”**中第一段的第一句
- 更新了 “PWM 触发模数转换”，向 DTM 多路开关中增加了一个零值输入 （见图 7-1）
- 增加了新的章节**第 17.0 节 “PWM 与其他外设互连”**

版本 E（2012 年 7 月）

该版本包括以下更新：

- 示例：
  - 在例 5-2 中，将 8 MHz 更新为 7.37 MHz，将 120 MHz 更新为 117.9 MHz
- 公式：
  - 增加了公式 5-1 至公式 5-3 的公式标题
  - 在公式 6-3 中，将 “最大 PWM 占空比分辨率为 1.04 ns” 中的 1.04 ns 更新为 1.06 ns
- 图片：
  - 在图 5-1 中，将标签 “FVCO(1)（最高 120 MHz）” 更新为 “FVCO(1)（80 MHz 至最高 120 MHz）”
  - 更新了图 5-6
  - 在图 6-1 和图 6-2 中，将标签 “clk” 更新为 “CLK”
  - 在图 7-4、图 7-7 至图 7-9 中，将图片标题中十进制数字的字体更新为 Computer 文本
  - 在图 7-10 中，将 PTMTMR 更新为 PMTMR
- 注：
  - 在寄存器 3-8、寄存器 3-10、寄存器 3-12 和寄存器 3-13 中，将 LSB 的所有引用更新为 LSb
  - 在寄存器 3-10 中，将 “注 1” 和 “注 2” 中的 “周期 - 0x0008” 更新为 “周期 + 0x0008”
  - 在寄存器 3-12 和寄存器 3-13 中，将 “注 2” 和 “注 3” 中的 “周期 - 0x0008” 更新为 “周期 + 0x0008”
  - 在寄存器 3-23 中，将 “注 1” 中的 1023 ns 更新为 1058 ns
  - 在寄存器 3-24 中进行了以下更新：
    - 删除了 “注 1”，并删除了寄存器标题中对注 1 的引用
    - 更新了 bit 5 和 bit 4 的注释引用
  - 在寄存器 3-25 中，将 “注 1” 中的 1023 ns 更新为 4258 ns
  - 在**第 5.1 节 “PWM 时钟选择”** 中，在公式 5-1 下注释框中增加了 “注 2”
  - 在**第 5.6 节 “时基同步”** 中进行了以下更新：
    - 将 “注 1：SYNCI 脉冲的周期应小于 PWM 周期值” 更换为 “注 1：SYNCI 脉冲的周期应大于 PWM 周期值”
    - 增加了 “注 5”
  - 在图 6-2 中，将 “注” 中的 SCDx 更新为 SDCx
  - 在**第 6.2.3 节 “辅助占空比（SDCx）”** 中，将 “注 1” 中的 “- 0x0008” 更新为 “+ 0x0008”（公式 6-3 上方）
  - 在**第 6.8 节 “中心对齐模式下的死区插入”** 中增加了一条注释
  - 更新了**第 10.1 节 “模拟比较器产生的 PWM 故障信号”** 中的注释
  - 在**第 13.0 节 “PWM 占空比立即更新”** 中增加了一条注释
  - 在**第 15.0 节 “独立时基的外部控制（电流复位模式）”** 中增加了一条注释
- 寄存器：
  - 在寄存器 3-7 中，将 PMTMR 更新为 SMTMR
  - 在寄存器 3-8 中，更新了 bit 15-3 的名称
  - 在寄存器 3-11 中，将 PWMLx 和 PWMHx 的所有引用更新为 PWMxL 和 PWMxH
  - 更新了寄存器 3-11 中 bit 0 位值 0 的说明
  - 在寄存器 3-19 中，将 bit 0 的 “OVDDAT<1:0> 位” 更新为 “OVRDAT<1:0> 位”
  - 在寄存器 3-23 中，将 “2^n * 8.32 ns” 更新为 “2^n * [1/(附属时钟频率)] ns”

© 2011-2014 Microchip Technology Inc.

版本 E （2012 年 7 月）（续）

- 章节：
  - 在**第 5.6 节 “时基同步”**中，将 “SYNCO 信号的脉冲宽度为 200 ns，以确保其他器件可靠地检测到信号”更新为 “SYNCO 信号脉冲的宽度为 12 个 TCY 时钟 （40 MIPS 时约 300 ns），以确保其他器件可以检测到该信号”
  - 在**第 6.2.4 节 “占空比分辨率”**中，将 “最低有效字节 （LSB）”更换为 LSb
  - 在**第 10.6 节 “故障引脚软件控制”**中，将在以下句子中的 “引脚”一词更新为 “GPIO 引脚”：“当引脚的端口位置 1 时，故障输入将被激活”
  - 在**第 10.7 节 “PWM 限流引脚”**中进行了以下更新：
    - 更新了第一个项目符号项
    - 将在第二个项目符号项中的 “故障输入信号”一词更新为 “限流信号”
    - 将项目符号子项 “在 IFLTMOD 位的独立故障模式下， CLDAT<1:0> 位不用于改写功能”更新为 “如果独立故障使能位 IFLTMOD （FCLCONx<15>）置 1，则 CLDAT<1:0> 位 （IOCONx<3:2>）不用于改写功能”
    - 将项目符号子项 “在限流模式使能位 （CLMOD）下，将会使能限流功能。CLDAT<1:0> 位 （高 / 低）将提供要分配给 PWMxH 和 PWMxL 输出的数据值”更新为 “如果 IFLTMOD 位 （FCLCONx<15>）清零且 CLMOD 位 （FCLCONx<8>）置 1，使能了限流功能，则 CLDAT<1:0> 位 （高 / 低）（IOCONx<3:2>）将提供要在限流有效时分配给 PWMxH 和 PWMxL 输出的数据值”
- 表格：
  - 在表 6-2 中，对于 PWM 时钟预分频比 1:1，将 **16 ns** 列中的 bit 6 更新为 bit 4
  - 对整篇文档的文字和格式进行了少量更新

版本 F （2014 年 3 月）

更新了 FRM 标题和器件名称。

- 章节：
  - 在**第 2.0 节 “特性”**中包含了新的项目符号列表，
  - 在**第 3.0 节 “控制寄存器”**中包含了新的寄存器，
  - 在**第 10.0 节 “PWM 故障引脚”**中包含了新的项目符号列表，
  - 更新了**第 10.1 节 “模拟比较器产生的 PWM 故障信号”**中的 “注”，
  - 在**第 10.7 节 “PWM 限流引脚”**中包含了新的项目符号列表，
  - 更新了**第 10.7.1 节 “电流复位模式的配置”**，
  - 在**第 11.0 节 “特殊功能”**中包含了新的项目符号列表，
  - 更新了**第 11.1 节 “前沿消隐 （LEB）”**，
  - 更新了**第 12.1 节 “PWM 输出改写逻辑”**，
  - 更新了**第 14.2 节 “空闲模式下的高速 PWM 操作”**中的 “注”，
  - 在**第 16.0 节 “应用信息”**中包含了新的项目符号列表，
  - 更新了**第17.0节“PWM与其他外设互连”**和**第17.2节“PWM 与高速模拟比较器互连”**。
  - 包含了新的章节：**第 11.6 节 “PWM 保护锁定 / 解锁密钥寄存器”**、**第 16.8 节 “多调制方案实现模式 （PWM + PFM）”**和**第 16.9 节 “迟滞电流控制模式”**。
- 寄存器
  - 更新了寄存器 3-11
  - 在寄存器 3-19 中包含了 “注 3”
  - 在寄存器 3-22 中，更新了 bit 14-10 的寄存器说明，并包含了 “注 5”和 “注 6”
  - 插入了新的寄存器 3-28
- 图片
  - 修改了图 10-1、图 10-2、图 10-3、图 10-4、图 10-5、图 10-6、图 11-1、图 16-7 和图 16-8
  - 插入了新的图 10-2、图 10-4 和图 16-10
- 表
  - 在**第 10.7.2 节 “在逐周期模式下配置模拟比较器”**的图 10-6 中增加了 “注”
- 公式
  - 更新了公式 11-1 和公式 11-2 的标题
  - 对整篇文档的文字和格式进行了少量更新

© 2011-2014 Microchip Technology Inc.

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。在 Microchip 知识产权保护下，不得暗中或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、 dsPIC、 FlashFlex、 flexPWR、 JukeBlox、 KEELOQ、 KEELOQ 徽标、 Kleer、 LANCheck、 MediaLB、 MOST、 MOST 徽标、 MPLAB、 OptoLyzer、 PIC、 PICSTART、 PIC32 徽标、 RightTouch、 SpyNIC、 SST、 SST 徽标、 SuperFlash 及 UNI/O 均为 Microchip Technology Inc. 在美国和其他国家或地区的注册商标。

The Embedded Control Solutions Company 和 mTouch 为 Microchip Technology Inc. 在美国的注册商标。

Analog-for-the-Digital Age、BodyCom、chipKIT、chipKIT 徽标、 CodeGuard、 dsPICDEM、 dsPICDEM.net、 ECAN、 In-Circuit Serial Programming、 ICSP、 Inter-Chip Connectivity、 KleerNet、 KleerNet 徽标、 MiWi、 MPASM、 MPF、 MPLAB Certified 徽标、 MPLIB、 MPLINK、 MultiTRAK、 NetDetach、 Omniscient Code Generation、 PICDEM、 PICDEM.net、 PICkit、 PICtail、 RightTouch 徽标、 REAL ICE、 SQI、 Serial Quad I/O、 Total Endurance、 TSHARC、 USBCheck、 VariSense、 ViewSpan、 WiperLock、 Wireless DNA 和 ZENA 均为 Microchip Technology Inc. 在美国和其他国家或地区的商标。

SQTP 为 Microchip Technology Inc. 在美国的服务标记。

Silicon Storage Technology 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。

© 2011-2014, Microchip Technology Inc. 版权所有。

ISBN：978-1-63276-244-3

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
═ ISO/TS 16949═

Microchip 位于美国亚利桑那州 Chandler 和 Tempe 与位于俄勒冈州 Gresham 的全球总部、设计和晶圆生产厂及位于美国加利福尼亚州和印度的设计中心均通过了 ISO/TS-16949:2009 认证。 Microchip 的 PIC® MCU 与 dsPIC® DSC、KEELOQ® 跳码器件、串行 EEPROM、单片机外设、非易失性存储器和模拟产品严格遵守公司的质量体系流程。此外， Microchip 在开发系统的设计和生产方面的质量体系也已通过了 ISO 9001:2000 认证。

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

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

克里夫兰 Cleveland
Independence, OH
Tel: 1-216-447-0464
Fax: 1-216-447-0643

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

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

纽约 New York, NY
Tel: 1-631-435-6000

圣何塞 San Jose, CA
Tel: 1-408-735-9110

加拿大多伦多 Toronto
Tel: 1-905-673-0699
Fax: 1-905-673-6509

亚太地区

亚太总部 Asia Pacific Office
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon
Hong Kong
Tel: 852-2943-5100
Fax: 852-2401-3431

中国 - 北京
Tel: 86-10-8569-7000
Fax: 86-10-8528-2104

中国 - 成都
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

中国 - 重庆
Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

中国 - 杭州
Tel: 86-571-8792-8115
Fax: 86-571-8792-8116

中国 - 香港特别行政区
Tel: 852-2943-5100
Fax: 852-2401-3431

中国 - 南京
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

中国 - 青岛
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

中国 - 上海
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

中国 - 沈阳
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

中国 - 深圳
Tel: 86-755-8864-2200
Fax: 86-755-8203-1760

中国 - 武汉
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

中国 - 西安
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

中国 - 厦门
Tel: 86-592-238-8138
Fax: 86-592-238-8130

中国 - 珠海
Tel: 86-756-321-0040
Fax: 86-756-321-0049

亚太地区

台湾地区 - 高雄
Tel: 886-7-213-7830

台湾地区 - 台北
Tel: 886-2-2508-8600
Fax: 886-2-2508-0102

台湾地区 - 新竹
Tel: 886-3-5778-366
Fax: 886-3-5770-955

澳大利亚 Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

印度 India - Bangalore
Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

印度 India - New Delhi
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

印度 India - Pune
Tel: 91-20-3019-1500

日本 Japan - Osaka
Tel: 81-6-6152-7160
Fax: 81-6-6152-9310

日本 Japan - Tokyo
Tel: 81-3-6880-3770
Fax: 81-3-6880-3771

韩国 Korea - Daegu
Tel: 82-53-744-4301
Fax: 82-53-744-4302

韩国 Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 或
82-2-558-5934

马来西亚 Malaysia - Kuala Lumpur
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

马来西亚 Malaysia - Penang
Tel: 60-4-227-8870
Fax: 60-4-227-4068

菲律宾 Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

新加坡 Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

泰国 Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

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

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

德国 Germany - Dusseldorf
Tel: 49-2129-3766400

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

德国 Germany - Pforzheim
Tel: 49-7231-424750

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

意大利 Italy - Venice
Tel: 39-049-7625286

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

波兰 Poland - Warsaw
Tel: 48-22-3325737

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

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

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

03/25/14

© 2011-2014 Microchip Technology Inc.