
外设触发信号发生器（PTG）

目录

本章包括下列主题：

1.0	简介	2
2.0	寄存器映射	4
3.0	步阶命令和格式	14
4.0	模块工作原理	19
5.0	应用示例	32
6.0	节能模式	39
7.0	相关应用笔记	40
8.0	版本历史	41

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。

请参见您所使用器件的数据手册中“**外设触发信号发生器 (PTG)**”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。

Microchip 器件数据手册和系列参考手册的相关章节可从 Microchip 网站下载，网址为：<http://www.microchip.com>。

1.0 简介

外设触发信号发生器 (Peripheral Trigger Generator, PTG) 模块是一个用户可编程序列发生器，它能够生成复杂的触发信号序列来协调其他外设的操作。

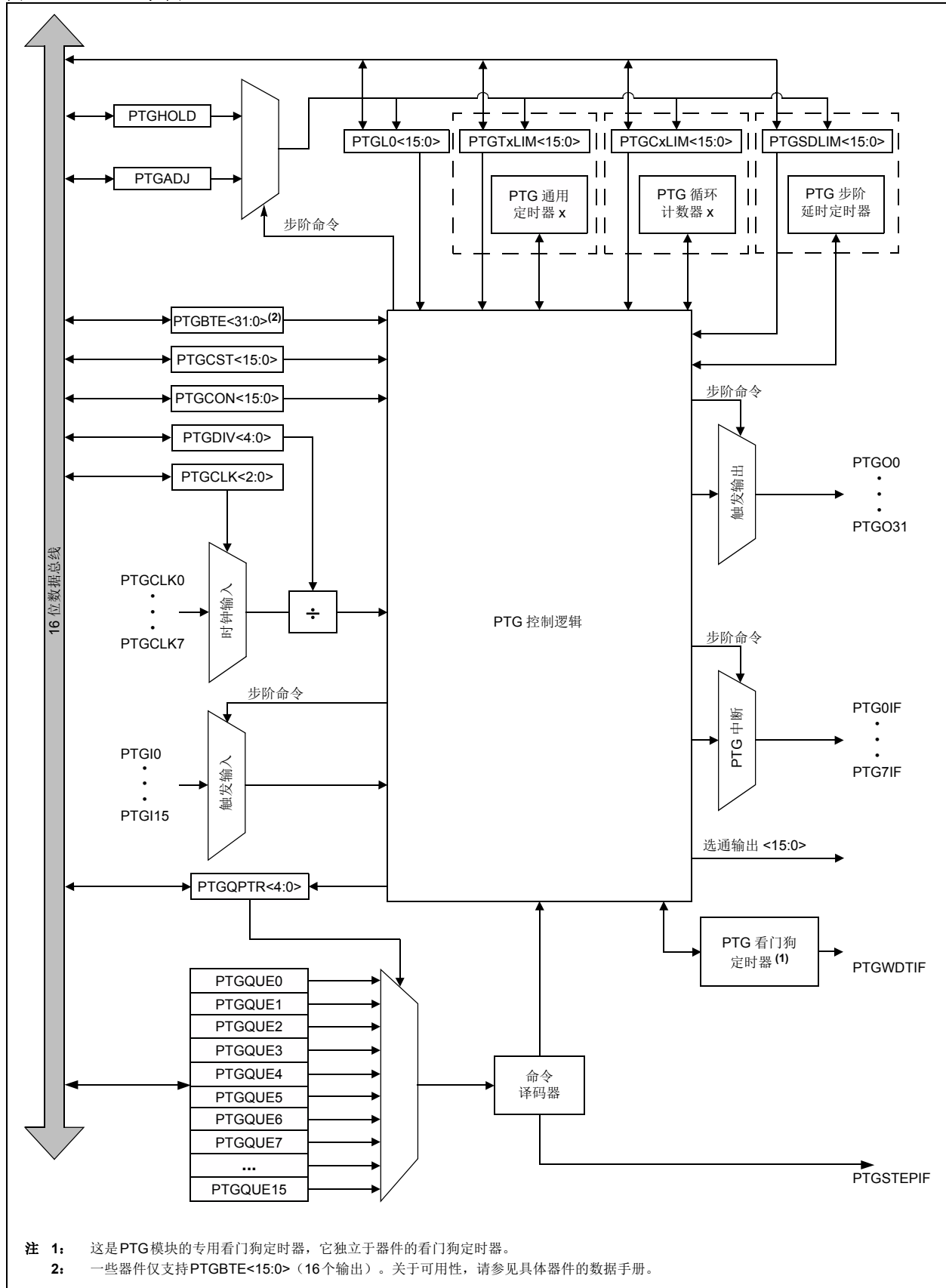
PTG 模块设计为可以与诸如模数转换器 (Analog-to-Digital Converter, ADC)、输出比较和 PWM 模块、定时器以及中断控制器等其他模块接口。

1.1 特性

- 行为通过步阶命令来驱动：
 - 步阶命令的宽度为 8 位
- 命令存储在一个步阶队列中：
 - 队列深度最大为 32
 - 可编程步阶执行时间 (步阶延时)
- 支持命令序列循环：
 - 可以嵌套 1 级深
 - 有条件或无条件循环
 - 两个 16 位循环计数器
- 16 个硬件输入触发信号：
 - 正边沿或负边沿敏感，或者高电平或低电平敏感
- 1 个软件输入触发信号
- 最多可生成 32 个不同的输出触发信号
- 可生成两种类型的触发输出：
 - 独立
 - 广播
- 立即数数据值的选通输出端口：
 - 5 位立即数写操作 (命令的立即数部分)
 - 16 位立即数写操作 (立即数保存在 PTGL0 寄存器中)
- 最多可生成 10 个独特的中断信号
- 两个 16 位通用定时器
- 灵活的独立看门狗定时器 (Watchdog Timer, WDT)，用于设置触发等待时间的上限
- 调试模式下支持单步命令
- 可选时钟 (系统、脉宽调制器 (Pulse-Width Modulator, PWM) 或 ADC)
- 可编程时钟分频器

外设触发信号发生器 (PTG)

图 1-1: PTG 框图



2.0 寄存器映射

表2-1: PTG 寄存器映射

寄存器名称	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PTGCST	PTGEN	r	PTGSIDL	PTGTOGL	—	PTGSWT	PTGSSEN	PTGIVIS	PTGSTRT	PTGWDTO	PTGBUSY	—	—	—	PTGITM<1:0>	
PTGCON	PTGCLK<2:0>			PTGDIV<4:0>					PTGPWD<3:0>				—	PTGWDT<2:0>		
PTGBTE	PTGBTE<15:0>															
PTGBTEH ⁽¹⁾	PTGBTE<31:16>															
PTGHOLD	PTGHOLD<15:0>															
PTGTOLIM	PTGTOLIM<15:0>															
PTGT1LIM	PTGT1LIM<15:0>															
PTGSDLIM	PTGSDLIM<15:0>															
PTGC0LIM	PTGC0LIM<15:0>															
PTGC1LIM	PTGC1LIM<15:0>															
PTGADJ	PTGADJ<15:0>															
PTGLO	PTGLO<15:0>															
PTGQPTR	—	—	—	—	—	—	—	—	—	—	—	PTGQPTR<4:0>				
PTGQUEn ⁽²⁾	STEP2n+1<7:0> ^(3,4)								STEP2n<7:0> ^(3,4)							

图注: — = 未实现, 读为0; r = 保留位。

注 1: 并非在所有器件上均提供该寄存器。关于可用性, 请参见具体器件的数据手册。

2: n = 0-15

3: 当模块执行步阶命令时, 这些位是只读的。

4: 有关步阶命令编码的信息, 请参见表3-1。

外设触发信号发生器 (PTG)

寄存器 2-1: **PTGCST: PTG 控制/状态低位字寄存器**

R/W-0	r-0	R/W-0	R/W-0	U-0	R/W-0, HC	R/W-0	R/W-0
PTGEN	—	PTGSIDL	PTGTOGL	—	PTGSWT ⁽²⁾	PTGSSEN ⁽³⁾	PTGIVIS
bit 15							bit 8
R/W-0, HC	R/W-0, HS	R/W-0, HS/HC	U-0	U-0	U-0	R/W-0	R/W-0
PTGSTRT	PTGWDTO	PTGBUSY ⁽⁴⁾	—	—	—	PTGITM1 ⁽¹⁾	PTGITM0 ⁽¹⁾
bit 7							bit 0

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

- bit 15 **PTGEN:** PTG 使能位
1 = 使能 PTG
0 = 禁止 PTG
- bit 14 **保留:** 必须写为 0
- bit 13 **PTGSIDL:** PTG 调试模式冻结位
1 = 器件空闲时, PTG 暂停工作
0 = 器件空闲时, PTG 继续工作
- bit 12 **PTGTOGL:** PTG 触发输出翻转位
1 = 每次执行 PTGTRIG 后翻转触发输出的状态
0 = 每次执行 PTGTRIG 后生成一个触发脉冲
- bit 11 **未实现:** 读为 0
- bit 10 **PTGSWT:** 软件触发位⁽²⁾
1 = 如果 PTG 状态机在执行“等待软件触发”步阶命令 (OPTION<3:0> = 1010 或 1011), 将完成命令执行并继续执行。
0 = 除清零该位外不执行任何其他操作
- bit 9 **PTGSSEN:** PTG 单步位⁽³⁾
1 = 在调试模式下使能单步
0 = 禁止单步
- bit 8 **PTGIVIS:** PTG 计数器/定时器可视性位
1 = 读取 PTGSDLIM、PTGCxLIM 或 PTGTxLIM 寄存器时返回它们相应的计数器/定时器寄存器 (PTGSD、PTGCx 和 PTGTx) 的当前值
0 = 读取 PTGSDLIM、PTGCxLIM 或 PTGTxLIM 寄存器时返回这些限制寄存器的值
- bit 7 **PTGSTRT:** 启动 PTG 序列发生器位
1 = 开始按顺序执行命令 (连续模式)
0 = 停止执行命令
- bit 6 **PTGWDTO:** PTG 看门狗定时器超时状态位
1 = PTG 看门狗定时器已超时
0 = PTG 看门狗定时器未超时
- bit 5 **PTGBUSY:** PTG 状态机忙位⁽⁴⁾
1 = PTG 基于所选的时钟源运行; 不允许对 PTGCLK<2:0> 和 PTGDIV<4:0> 执行 SFR 写操作
0 = PTG 状态机不在运行
- bit 4-2 **未实现:** 读为 0

注 1: 这些位仅适用于 PTGWHI 和 PTGWLO 命令。

2: 该位仅用于 PTGCTRL 步阶命令软件触发选项。

3: 只能在调试模式下写入 PTGSSEN 位。

4: 并非在所有器件上都提供该位。关于可用性, 请参见具体器件的数据手册。

dsPIC33/PIC24 系列参考手册

寄存器 2-1: **PTGCST: PTG 控制/状态低位字寄存器 (续)**

bit 1-0 **PTGITM<1:0>**: PTG 输入触发操作选择位⁽¹⁾

- 11 = 在退出命令时不执行带步阶延时的单电平检测 (无论 PTGCTRL 命令如何) (模式 3)
- 10 = 在退出命令时执行带步阶延时的单电平检测 (模式 2)
- 01 = 在退出命令时不执行带步阶延时的连续边沿检测 (无论 PTGCTRL 命令如何) (模式 1)
- 00 = 在退出命令时执行带步阶延时的连续边沿检测 (模式 0)

- 注 1:** 这些位仅适用于 PTGWHI 和 PTGWLO 命令。
- 2:** 该位仅用于 PTGCTRL 步阶命令软件触发选项。
 - 3:** 只能在调试模式下写入 PTGSSEN 位。
 - 4:** 并非在所有器件上均提供该位。关于可用性, 请参见具体器件的数据手册。

外设触发信号发生器 (PTG)

寄存器 2-2: **PTGCONH: PTG控制/状态高位字寄存器**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PTGCLK<2:0>				PTGDIV<4:0>			
bit 15				bit 8			

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
PTGPWD<3:0>				—	PTGWDT<2:0>		
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 15-13 **PTGCLK<2:0>**: PTG 模块时钟源选择位
有关时钟源选择的信息, 请参见具体器件数据手册。

bit 12-8 **PTGDIV<4:0>**: PTG 模块时钟预分频比位
11111 = 32 分频
11110 = 31 分频
•
•
•
00001 = 2 分频
00000 = 1 分频

bit 7-4 **PTGPWD<3:0>**: PTG 触发输出脉冲宽度 (用 PTG 时钟周期表示) 位
1111 = 所有触发输出的宽度为 16 个 PTG 时钟周期
1110 = 所有触发输出的宽度为 15 个 PTG 时钟周期
•
•
•
0001 = 所有触发输出的宽度为 2 个 PTG 时钟周期
0000 = 所有触发输出的宽度为 1 个 PTG 时钟周期

bit 3 **未实现**: 读为 0

bit 2-0 **PTGWDT<2:0>**: PTG 看门狗定时器超时选择位
111 = 看门狗定时器将在 512 个 PTG 时钟后超时
110 = 看门狗定时器将在 256 个 PTG 时钟后超时
101 = 看门狗定时器将在 128 个 PTG 时钟后超时
100 = 看门狗定时器将在 64 个 PTG 时钟后超时
011 = 看门狗定时器将在 32 个 PTG 时钟后超时
010 = 看门狗定时器将在 16 个 PTG 时钟后超时
001 = 看门狗定时器将在 8 个 PTG 时钟后超时
000 = 禁止看门狗定时器

dsPIC33/PIC24 系列参考手册

寄存器2-3: PTGBTE: PTG广播触发使能低位字寄存器^(1,2)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PTGBTE<15:8>							
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
PTGBTE<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 15-0 **PTGBTE<15:0>**: PTG广播触发使能位
1 = 执行广播命令时生成触发信号
0 = 执行广播命令时不生成触发信号

- 注 1: 当模块执行步阶命令时, 这些位是只读的。
2: 关于可用位及位分配, 请参见具体器件的数据手册。

寄存器2-4: PTGBTEH: PTG广播触发使能高位字寄存器^(1,2,3)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PTGBTE<31:24>							
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
PTGBTE<23:16>							
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 **PTGBTE<31:16>**: PTG广播触发使能位
1 = 执行广播命令时生成触发信号
0 = 执行广播命令时不生成触发信号

- 注 1: 并非在所有器件上均提供该寄存器。关于可用性, 请参见具体器件的数据手册。
2: 当模块执行步阶命令时, 这些位是只读的。
3: 关于可用位及位分配, 请参见具体器件的数据手册。

外设触发信号发生器 (PTG)

寄存器2-5: PTGHOLD: PTG保持寄存器⁽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
PTGHOLD<15:8>							
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
PTGHOLD<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 15-0 **PTGHOLD<15:0>**: PTG通用保持寄存器位
 该寄存器用于保存用户提供的、要使用PTGCOPY命令复制到PTGTxLIM、PTGCxLIM、PTGSDLIM或PTGL0寄存器的数据。

注 1: 当模块执行步阶命令时, 这些位是只读的。

寄存器2-6: PTGT0LIM: PTG Timer0限制寄存器⁽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
PTGT0LIM<15:8>							
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
PTGT0LIM<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 15-0 **PTGT0LIM<15:0>**: PTG Timer0限制寄存器位
 通用Timer0限制寄存器。

注 1: 当模块执行步阶命令时, 这些位是只读的。

dsPIC33/PIC24 系列参考手册

寄存器 2-7: PTGT1LIM: PTG Timer1 限制寄存器⁽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
PTGT1LIM<15:8>							
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
PTGT1LIM<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 15-0 **PTGT1LIM<15:0>**: PTG Timer1 限制寄存器位
 通用Timer1 限制寄存器。

注 1: 当模块执行步阶命令时, 这些位是只读的。

寄存器 2-8: PTGSDLIM: PTG 步阶延时限制寄存器⁽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
PTGSDLIM<15:8>							
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
PTGSDLIM<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 15-0 **PTGSDLIM<15:0>**: PTG 步阶延时限制寄存器位
 该寄存器用于保存一个PTG 步阶延时值, 该值代表在步阶命令启动和步阶命令完成之间的额外PTG 时钟数量。

注 1: 当模块执行步阶命令时, 这些位是只读的。

外设触发信号发生器 (PTG)

寄存器 2-9: PTGC0LIM: PTG 计数器 0 限制寄存器⁽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
PTGC0LIM<15:8>							
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
PTGC0LIM<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 15-0 **PTGC0LIM<15:0>**: PTG 计数器 0 限制寄存器位
 该寄存器用于指定 PTGJMPC0 步阶命令的循环计数, 或用作通用计数器 0 的限制寄存器。

注 1: 当模块执行步阶命令时, 这些位是只读的。

寄存器 2-10: PTGC1LIM: PTG 计数器 1 限制寄存器⁽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
PTGC1LIM<15:8>							
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
PTGC1LIM<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 15-0 **PTGC1LIM<15:0>**: PTG 计数器 1 限制寄存器位
 该寄存器用于指定 PTGJMPC1 步阶命令的循环计数, 或作为通用计数器 1 的限制寄存器。

注 1: 当模块执行步阶命令时, 这些位是只读的。

dsPIC33/PIC24 系列参考手册

寄存器2-11: PTGADJ: PTG 调节寄存器⁽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
PTGADJ<15:8>							
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
PTGADJ<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 15-0 **PTGADJ<15:0>**: PTG 调节寄存器位
该寄存器用于保存用户提供的、要使用PTGADD命令加到PTGTxLIM、PTGCxLIM、PTGSDLIM或PTGL0寄存器的数据。

注 1: 当模块执行步阶命令时, 这些位是只读的。

寄存器2-12: PTGL0: PTG 立即数0寄存器^(1,2)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PTGL0<15:8>							
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
PTGL0<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 15-0 **PTGL0<15:0>**: PTG 立即数0寄存器位
该寄存器用于保存要使用PTGCTRL命令写入AD1CHS0寄存器的16位值。

注 1: 当模块执行步阶命令时, 这些位是只读的。

2: PTG选通输出通常连接到ADC通道选择寄存器。这样, PTG便可直接控制ADC通道切换。有关PTG输出连接的信息, 请参见具体器件数据手册。

外设触发信号发生器 (PTG)

寄存器 2-13: PTGQPTR: PTG 步阶队列指针寄存器⁽¹⁾

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	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	
—	—	—	PTGQPTR<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 15-5 **未实现:** 读为0

bit 4-0 **PTGQPTR<4:0>:** PTG 步阶队列指针寄存器位
 该寄存器用于指向步阶队列中当前处于活动状态的步阶命令。

注 1: 当模块执行步阶命令时, 这些位是只读的。

寄存器 2-14: PTGQEn: PTG 步阶队列指针 n 寄存器 (n = 0-15)^(1,2)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
STEP2n+1<7:0>							
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
STEP2n<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 15-8 **STEP2n+1<7:0>:** PTG 命令 4n+1 位
 用于存储 STEP2n+1 命令字节的队列位置, 其中 “n” 来自 PTGQEn。

bit 7-0 **STEP2n<7:0>:** PTG 命令 4n 位
 用于存储 STEP2n 命令字节的队列位置, 其中 “n” 等于奇编号步阶队列指针。

注 1: 当模块执行步阶命令时, 这些位是只读的。

2: 有关步阶命令编码的信息, 请参见表 3-1。

3.0 步阶命令和格式

PTG 使用 11 个 8 位步阶命令来执行更高级任务。步阶命令分为四种类型：

- 输入事件控制
- 控制功能
- 流控制
- 输出生成

通过组合使用这些命令并按照不同的顺序执行，可以在没有 CPU 干预的情况下做出决策并采取相应操作。

3.1 输入事件控制

输入事件控制包含两条命令：PTGWHI 和 PTGWLO。这两条命令分别等待至 16 个 PTG_{Ix} 输入之一上出现电平或边沿变化时触发。这两条命令与指定使用哪个 PTG_{Ix} 的 4 位 OPTION 位域配合使用。当指定输入按预期方向翻转后，将使步阶队列指针递增，并评估下一个步阶命令。更多信息，请参见第 4.7.2 节“等待触发输入”。

3.2 控制功能

控制功能包含三条命令：PTGCTRL、PTGADD 和 PTGCOPY。PTGCTRL 命令用于控制延时定时器、软件触发和选通输出的操作。PTGADD 命令用于将 PTGHOLD 寄存器的内容加到其他 PTG 寄存器，包括计数器、定时器、步阶延时和立即数寄存器。PTGCOPY 命令用于将 PTGHOLD 寄存器的内容复制到其他 PTG 寄存器，与 PTGADD 命令类似。

3.3 流控制

流控制通过以下 3 个跳转命令来实现：PTGJMP、PTGJMPC0 和 PTGJMPC1。这些跳转命令均为 3 位命令，允许 5 位 OPTION 位域与队列指针 PTGQPTR 的对应项进行匹配。PTGJMP 命令仅跳转到指定队列位置，而 PTGJMPC_x 命令则根据计数器和 PTG 计数器限制寄存器的比较结果进行条件跳转（PTGC_xLIM）。

3.4 输出生成

输出生成通过以下 3 条命令来实现：PTGTRIG、PTGIRQ 和 PTGSTRB。PTGTRIG 用于选择和生成输出触发信号（PTGO_x）。当与 5 位 OPTION 位域搭配使用时，PTGTRIG 还支持从 32 个 PTGO 中进行选择。PTGIRQ 命令用于生成中断请求（由 OPTION 位域指定中断（PTG_xIF））。有关触发信号和中断的更多信息，请参见第 4.9 节“PTG 模块输出”。PTGSTRB 命令用于生成选通输出，从而将 16 位数据输出到另一个外设。更多信息，请参见第 4.10 节“选通输出”。

外设触发信号发生器 (PTG)

表3-1提供了PTG命令的概述，表3-2详细说明了每条命令的可用选项。例3-1、例3-2和例3-3提供了命令定义及其选项的C语言代码示例。本FRM中的后续示例将会引用这些示例。

表3-1: PTG步阶命令格式和说明

步阶命令字节			
STEPx<7:0>			
CMD<3:0>		OPTION<3:0>	
bit 7	bit 4	bit 3	bit 0

bit 7-4	步阶命令	CMD<3:0>	命令说明
	PTGCTRL	0000	根据 OPTION<3:0> 位的描述执行控制命令。
	PTGADD	0001	根据 OPTION<3:0> 位的描述，将 PTGADJ 寄存器的内容加到目标寄存器中。
	PTGCOPY		根据 OPTION<3:0> 位的描述，将 PTGHOLD 寄存器的内容复制到目标寄存器中。
	PTGSTRB	001x	将 CMD<0>:OPTION<3:0> 位中包含的值复制到选通输出位 <4:0>。
	PTGWHI	0100	根据 OPTION<3:0> 位的描述，等待来自选定 PTG 触发输入的从低电平变为高电平的边沿输入。
	PTGWLO	0101	根据 OPTION<3:0> 位的描述，等待来自选定 PTG 触发输入的从高电平变为低电平的边沿输入。
	—	0110	保留；不要使用。(1)
	PTGIRQ	0111	根据 OPTION<3:0> 位的描述，产生独立的中断请求。
	PTGTRIG	100x	根据 CMD<0>:OPTION<3:0> 5 位位域的描述，产生独立的触发输出。
	PTGJMP	101x	将 CMD<0>:OPTION<3:0> 位中包含的值复制到 PTGQPTR 寄存器，然后跳转到相应的步阶队列。
	PTGJMPC0	110x	PTGC0 = PTGC0LIM: 递增 PTGQPTR 寄存器。
			PTGC0 ≠ PTGC0LIM: 递增计数器 0 (PTGC0) 并将 CMD<0>:OPTION<3:0> 位中包含的值复制到 PTGQPTR 寄存器，然后跳转到相应的步阶队列。
	PTGJMPC1	111x	PTGC1 = PTGC1LIM: 递增 PTGQPTR 寄存器。
			PTGC1 ≠ PTGC1LIM: 递增计数器 1 (PTGC1)，并将 CMD<0>:OPTION<3:0> 位中包含的值复制到 PTGQPTR 寄存器，然后跳转到该步阶队列。

注 1: 所有保留的命令或选项都将执行，但它们不起任何作用（即，它们作为一条 NOP 指令执行）。

例3-1: PTG命令定义

```

/* PTG Commands */
#define PTGCTRL(x)      ((0x0 << 4) | ((x) & 0x0F))
#define PTGADD(x)       ((0x2 << 3) | ((x) & 0x07))
#define PTGCOPY(x)      ((0x3 << 3) | ((x) & 0x07))
#define PTGSTRB(x)      ((0x1 << 5) | ((x) & 0x1F))
#define PTGWHI(x)       ((0x4 << 4) | ((x) & 0x0F))
#define PTGWLO(x)       ((0x5 << 4) | ((x) & 0x0F))
#define PTGIRQ(x)       ((0x7 << 4) | ((x) & 0x0F))
#define PTGTRIG(x)      ((0x4 << 5) | ((x) & 0x1F))
#define PTGJMP(x)       ((0x5 << 5) | ((x) & 0x1F))
#define PTGJMPC0(x)     ((0x6 << 5) | ((x) & 0x1F))
#define PTGJMPC1(x)     ((0x7 << 5) | ((x) & 0x1F))

```

dsPIC33/PIC24 系列参考手册

表 3-2: PTG 命令选项

bit 3-0	步阶命令	OPTION<3:0>	命令说明
	PTGCTRL ⁽¹⁾	0000	NOP
		0001	保留；不要使用。
		0010	禁止步阶延时定时器（PTGSD）。
		0011	保留；不要使用。
		0100	保留；不要使用。
		0101	保留；不要使用。
		0110	使能步阶延时定时器（PTGSD）。
		0111	保留；不要使用。
		1000	启动并等待 PTGT0 与 PTGT0LIM 寄存器发生匹配。
		1001	启动并等待 PTGT1 与 PTGT1LIM 寄存器发生匹配。
		1010	等待软件触发信号（电平触发，PTGSWT = 1）。
		1011	等待软件触发信号（正边沿触发，PTGSWT = 0 翻转为 1）。
		1100	将 PTGC0LIM 寄存器的内容复制到选通输出。
		1101	将 PTGC1LIM 寄存器的内容复制到选通输出。
		1110	将 PTGL0 寄存器的内容复制到选通输出。
		1111	产生 PTGBTE 寄存器中指示的触发信号。
		PTGADD ⁽¹⁾	0000
	0001		将 PTGADJ 寄存器的内容加到 PTGC1LIM 寄存器。
	0010		将 PTGADJ 寄存器的内容加到 PTGT0LIM 寄存器。
	0011		将 PTGADJ 寄存器的内容加到 PTGT1LIM 寄存器。
	0100		将 PTGADJ 寄存器的内容加到 PTGSDLIM 寄存器。
	0101		将 PTGADJ 寄存器的内容加到 PTGL0 寄存器。
	0110		保留；不要使用。
	0111		保留；不要使用。
	PTGCOPY ⁽¹⁾	1000	将 PTGHOLD 寄存器的内容复制到 PTGC0LIM 寄存器。
		1001	将 PTGHOLD 寄存器的内容复制到 PTGC1LIM 寄存器。
		1010	将 PTGHOLD 寄存器的内容复制到 PTGT0LIM 寄存器。
		1011	将 PTGHOLD 寄存器的内容复制到 PTGT1LIM 寄存器。
		1100	将 PTGHOLD 寄存器的内容复制到 PTGSDLIM 寄存器。
		1101	将 PTGHOLD 寄存器的内容复制到 PTGL0 寄存器。
		1110	保留；不要使用。
		1111	保留；不要使用。

注 1: 所有保留的命令或选项都将执行，但它们不起任何作用（即，它们作为一条 NOP 指令执行）。

外设触发信号发生器 (PTG)

表 3-2: PTG 命令选项 (续)

bit 3-0	步阶命令	OPTION<3:0>	选项说明	
	PTGWHI ⁽¹⁾ 或 PTGWLO ⁽¹⁾	0000	PTGI0 (关于输入分配, 请参见具体器件数据手册)。	
		.	.	
		.	.	
			1111	PTGI15 (关于输入分配, 请参见具体器件数据手册)。
	PTGIRQ ⁽¹⁾		0000	产生PTG中断0 (关于中断分配, 请参见具体器件数据手册)。
			.	.
			.	.
			.	.
			0111	产生PTG中断7 (关于中断分配, 请参见具体器件数据手册)。
			1000	保留; 不要使用。
			.	.
		.	.	
		.	.	
		1111	保留; 不要使用。	
	PTGTRIG		00000	PTGO0 (关于分配, 请参见具体器件数据手册)。
		00001	PTGO1 (关于分配, 请参见具体器件数据手册)。	
		.	.	
		.	.	
		.	.	
		11110	PTGO30 (关于分配, 请参见具体器件数据手册)。	
		11111	PTGO31 (关于分配, 请参见具体器件数据手册)。	

注 1: 所有保留的命令或选项都将执行, 但它们不起任何作用 (即, 它们作为一条NOP指令执行)。

例 3-2: PTGCTRL 选项

```
/ Used with PTGCTRL command
typedef enum
{
    stepDelayDisable      = 0x2,
    stepDelayEnable      = 0x6,
    t0Wait                = 0x8,
    t1Wait                = 0x9,
    softTriggerLevelWait = 0xA,
    softTriggerEdgeWait  = 0xB,
    c0Strobe              = 0xC,
    c1Strobe              = 0xD,
    l0Strobe              = 0xE,
    triggerGenerate      = 0xF,
} CTRL_T;
```

例 3-3: PTGADD 和 PTGCOPY 命令选项

```
/ Used with PTGADD and PTGCOPY commands
typedef enum
{
    c0Limit = 0x0,
    c1Limit = 0x1,
    t0Limit = 0x2,
    t1Limit = 0x3,
    stepDelay = 0x4,
    literal0 = 0x5,
} ADD_COPY_T;
```

4.0 模块工作原理

4.1 PTG 说明

PTG 模块是一个用于生成复杂外设触发信号序列的用户可编程序列发生器。PTG 模块能够调度复杂的外设操作，这是通过软件解决方案难以实现或无法实现的。

用户需要将8位命令（称为步阶命令）写入PTG队列寄存器（PTGQUE0-PTGQUE15）。每个8位步阶命令均由一条命令代码和一个选项字段组成。表3-1列出了步阶命令的格式和编码。基于这些命令，PTG可以与其他外设（如PWM、ADC、MCCP、输入捕捉和输出比较）进行交互。有关是否支持这些外设的信息，请参见具体器件数据手册。

4.2 PTG 时钟选择

PTG 模块具有多种时钟选项，以及一个可选的预分频器，它可以对PTG时钟输入进行1至32分频。

4.2.1 时钟源选择

PTGCLK<2:0>位（PTGCON<15:13>）用于指定PTG时钟发生逻辑的时钟源。这些时钟源特定于器件。关于可用时钟源，请参见具体器件的数据手册。

4.2.2 时钟预分频比选择

PTGDIV<4:0>位（PTGCON<12:8>）用于指定PTG时钟发生逻辑的预分频值。只有在PTG模块被禁止（PTGEN = 0）时，才能写入这些位。

注： 在PTGEN = 1时，尝试写入PTGDIV<4:0>位或PTGCLK<2:0>位不会产生任何作用。

4.2.3 模块使能延时

使能PTG模块（PTGEN = 1）之后，在PTG开始执行命令之前存在一定的延时。该延时可以使用公式4-1表示。PTG时钟周期是预分频器后的有效时钟。

公式4-1:

$$TDLYEN = 4 \cdot PTG \text{ 时钟周期 (最大值)}$$

用户必须确保在该延时期间不修改任何控制位。此外，在PTG状态机开始执行之前不能有任何外部触发信号置为有效，否则将会丢失触发信号。

4.3 基本操作

用户将步阶命令（8位值）装入PTG队列寄存器。这些命令用于定义一个向外设产生触发输出信号的事件序列。步阶命令还可以用于向处理器产生中断请求。

当PTGEN位（PTGCST<15>）= 1时，将使能PTG模块，并向其提供时钟。当PTGSTRT位（PTGCST<7>）= 0时，PTG模块将处于暂停状态。

- 注 1:** 当PTG模块处于暂停状态时，将无法修改控制寄存器。
- 2:** 只有先使能PTG模块（PTGEN = 1）之后，才能将PTGSTRT位置1。
- 3:** 用户不应尝试在同一数据写周期中将PTGEN和PTGSTRT位置1。

随后设置PTGSTRT = 1，这将使能模块的步阶命令队列连续执行模式。PTG序列发生器将从队列指针（PTGQPTR）中保存的地址处开始读取步阶队列。每条命令字节将按顺序进行读取、译码和执行。所有步阶命令的最小持续时间为1个PTG时钟，如第4.2节“PTG时钟选择”中所述。

步阶命令将顺序执行，直到发生以下任意情况为止：

- 执行了PTGJMP、PTGJMPC0或PTGJMPC1（流改变）步阶命令。
- 用户通过清零PTGSTRT位来停止PTG序列发生器。模块将不再进一步读取/译码步阶命令，并暂停执行。
- 内部看门狗定时器发生溢出，将PTGSTRT位清零，并停止PTG序列发生器。模块将不再进一步读取/译码步阶命令，并暂停执行。
- PTG模块被禁止（PTGEN = 0）。

此外，还可以将步阶命令设置为等待某种条件（如输入触发边沿、软件触发或定时器匹配），之后再继续执行。有关详细信息，请参见第4.11节“停止序列发生器”。

4.4 控制寄存器访问

当使能PTG模块 (PTGEN = 1) 时, 将禁止写入除PTGCST寄存器 (该寄存器可以正常读写) 之外的所有其他控制寄存器。

当使能PTG模块 (PTGEN = 1) 时, 可以在任意时刻对任意控制寄存器执行读操作; 但是, 读取的数据 (控制寄存器或关联的定时器/计数器值) 将取决于PTGIVIS位 (PTGCST<8>) 的状态。

注: 只有一些寄存器会受PTGIVIS位状态影响。请参见PTGIVIS位的说明。

当禁止PTG模块 (PTGEN = 0) 时, 所有控制寄存器都可以正常读写。当PTGEN = 0时, PTGIVIS位 (PTGCST<8>) 没有任何作用, 因为所有定时器/计数器都会清零; 但是, 限制寄存器中的值将保持不变。

4.5 步阶队列指针

PTG步阶队列指针寄存器 (PTGQPTR) 用于保存指向PTG队列的指针。当PTGEN = 1且PTGSTRT位置1时, 将从PTGQPTR寄存器装入指针, 并寻址步阶队列中当前活动的步阶命令。当模块被禁止 (PTGEN = 0) 或处于复位状态时, PTGQPTR寄存器会被清零。这是一个一次性事件, 它在PTGEN位从1变为0时发生。

注 1: 当PTG模块暂停 (PTGSTRT = 0) 时, PTGQPTR寄存器不会清零。

2: PTGQPTR寄存器在PTGEN位清零时清零, 但无论PTGEN位的状态如何, 都可以向该寄存器中写入任意值。

用户可以在任意时刻读取PTGQPTR寄存器。在禁止状态 (PTGEN = 0) 下, 读操作将返回PTGQPTR寄存器的内容。在空闲或活动状态 (PTGEN = 1) 下, 读操作将返回指针的内容。

PTGQPTR寄存器通常在每条命令的第一个周期进行递增。该规则的例外情况有:

- 如果执行的是PTGJMP命令: 步阶队列指针中将装入目标队列地址
- 如果执行的是PTGJMPCx命令, 且PTGCx不为0: 步阶队列指针中将装入目标队列地址

4.6 命令循环控制

模块提供了两个16位循环计数器 (PTGC0和PTGC1), PTGJMPCx命令可以使用它们作为块循环计数器或延时发生器。

每个循环计数器都包含一个递增计数器 (PTGCx) 和一个限制寄存器 (PTGCxLIM)。限制寄存器的值可以通过直接写入寄存器进行更改 (禁止模块时), 也可以通过PTG序列发生器进行更改 (使能模块时)。从限制寄存器中读取的数据取决于PTG计数器/定时器可视性位 (PTGIVIS) 的状态。当模块处于复位状态或PTG模块被禁止 (PTGEN = 0) 时, 计数器会清零。

4.6.1 使用循环计数器作为块循环计数器

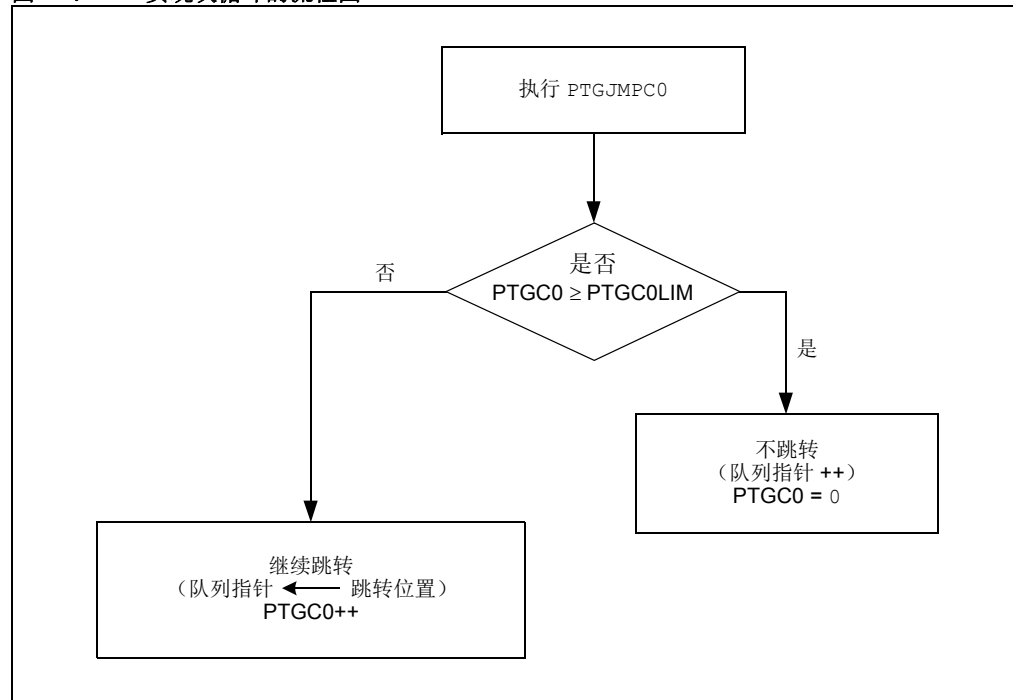
PTGJMPCx（条件跳转）命令使用一个循环计数器来跟踪PTGJMPCx命令的执行次数，因而可以用于构造代码块循环。这在需要若干次重复某个外设事件序列的应用中非常有用。通过PTGJMPCx命令，用户可以构造代码循环，并使用更少的步阶命令。

每次执行PTGJMPCx命令时，相应的内部循环计数器都会与其限制值进行比较。如果循环计数器尚未达到限制值，则将跳转位置装入PTGQPTR寄存器，并将循环计数器递增1。之后将从新的队列位置获取下一条命令。如果计数器达到限制值，则序列发生器会执行下一条命令（即，递增队列指针）。在准备执行下一个PTGJMPCx命令循环时，相应的循环计数器会被清零（见图4-1）。

注： 在执行命令循环的第一次迭代之前，可以（通过PTGADD或PTGCOPY命令）修改循环计数器的值。

模块提供了两个独立的循环计数器和关联的PTGJMPCx命令，从而支持循环嵌套（1级深）。对于哪条PTGJMPCx命令放在内循环或外循环中，并不存在任何限制。

图4-1： 实现块循环的流程图



4.7 序列发生器操作

除流改变命令和等待外部输入的命令外，所有其他命令的执行时间都是单个周期。

4.7.1 步阶命令持续时间

默认情况下，每条步阶命令的执行时间都为1个PTG时钟周期。有几种方法可以降低步阶命令的执行速率：

- 等待触发输入
- 等待GP定时器 (PTGTxLIM)
- 使用PTGJMP和PTGJMPCx命令插入一个延时循环
- 使能在执行每条命令之后插入步阶延时 (PTGSDLIM)

4.7.2 等待触发输入

PTG模块最多可以支持16个独立的触发输入。PTG输入PTGI0至PTGI15特定于器件；关于可用性，请参见器件数据手册。用户可以指定一条步阶命令等待选定的输入信号出现正边沿或负边沿，或者出现高电平或低电平。工作模式通过PTGCST寄存器中的PTGITM<1:0>位进行选择。

PTGWHI命令会等待选定触发输入上出现正边沿或高电平状态。PTGWLO命令会等待选定触发输入上出现负边沿或低电平状态。PTG会重复触发输入命令（即，实际上进行等待），直到选定信号变为有效，然后再继续执行步阶命令。

等待触发信号的最短执行时间为1个PTG时钟。除了看门狗定时器所产生的限制之外，对于PTG等待触发输入的时间不存在任何其他限制。更多信息，请参见第4.8节“PTG看门狗定时器”。

PTG模块支持4种输入触发命令工作模式（模式0-模式3），它们通过PTGCST寄存器中的PTGITM<1:0>位进行选择。

注： 如果禁止步阶延时，则模式0和模式1在操作上是等价的，模式2和模式3在操作上是等价的。

4.7.2.1 模式0: PTGITM<1:0> = 0x00（在退出时进行带步阶延时的连续边沿检测）

在该模式下，将在执行PTGWHI或PTGWLO命令之后立即开始对选定触发输入进行连续测试。当检测到触发边沿时，即完成命令执行。

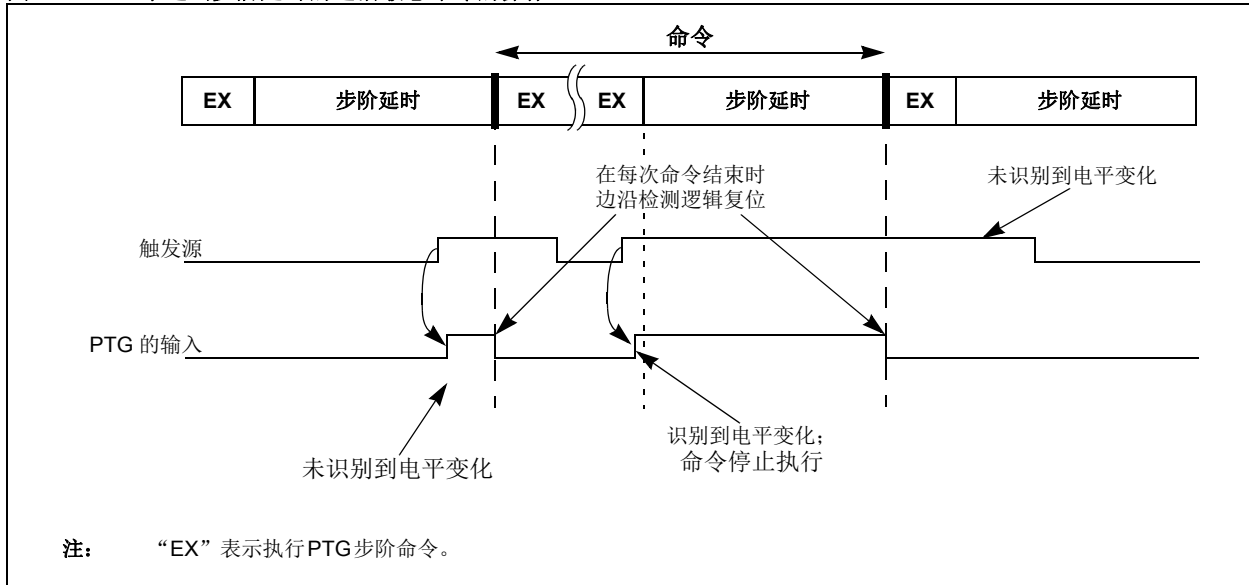
如果使能了步阶延时计数器，则会在检测到有效边沿之后和命令执行之后插入步阶延时（一次）。

如果未使能步阶延时计数器，则命令在检测到有效边沿之后即完成，并立即开始执行后续命令。

注： 边沿检测逻辑会在命令执行完成之后（即，在与命令关联的任何步阶延时之前）复位。为了检测到边沿，边沿应在PTGWHI或PTGWLO命令执行期间，或在其前一条命令的步阶延时期间出现。

图4-2给出了模式0操作的示例时序图。

图4-2: 带退出步阶延时的边沿敏感命令的操作



4.7.2.2 模式1: $PTGITM<1:0> = 0x01$ (在退出时进行不带步阶延时的连续边沿检测)

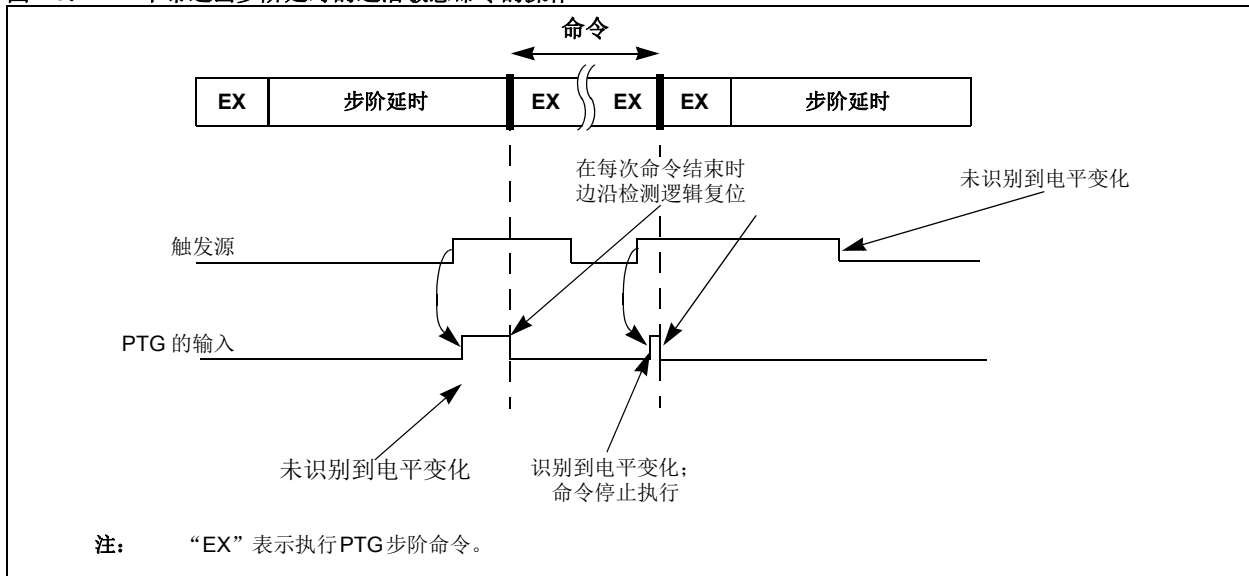
在该模式下,将在执行PTGWHI或PTGWLO命令之后立即开始对选定触发输入进行连续测试。当检测到触发边沿时,即完成命令执行。

无论是使能还是禁止步阶延时计数器,都不会在命令执行完成后插入步阶延时。

注: 边沿检测逻辑在命令执行完成之后复位。为了检测到边沿,边沿应在PTGWHI或PTGWLO命令执行期间,或在其前一条命令的步阶延时期间出现。

图4-3给出了模式1操作的示例时序图。

图4-3: 不带退出步阶延时的边沿敏感命令的操作



外设触发信号发生器 (PTG)

4.7.2.3 模式2: $PTGITM<1:0> = 0x10$ (在退出时进行带步阶延时的采样电平检测)

在该模式下,将在执行PTGWHI或PTGWLO命令之后立即对选定触发输入进行采样测试,测试所选的触发输入是否出现有效电平(每个PTG时钟一次)。

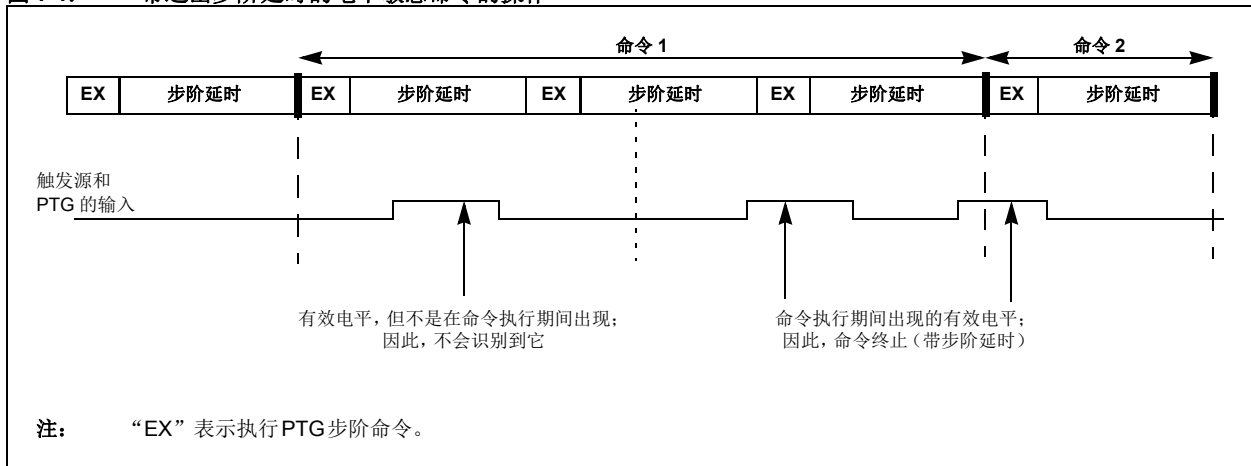
如果未出现触发信号,并且使能了步阶延时,则命令会等待步阶延时结束,然后再次对触发输入进行测试。当出现触发信号时,即完成命令执行,并且会重新插入步阶延时。

如果未出现触发信号,并且禁止了步阶延时,则命令会在下一个PTG时钟周期期间立即再次测试触发输入。当出现触发信号时,即完成命令执行,并立即开始执行后续命令。

- 注 1:** 由于该工作模式是电平敏感的,所以如果输入触发电平在开始执行PTGWHI或PTGWLO命令时即为真,则将立即满足输入测试条件。
- 2:** 输入不会进行锁存,因此它必须在命令执行时有效,这样才能识别到它。

图4-4给出了模式2操作的示例时序图。

图4-4: 带退出步阶延时的电平敏感命令的操作



4.7.2.4 模式3: $PTGITM<1:0> = 0x11$ (在退出时进行不带步阶延时的采样电平检测)

在该模式下,将在执行PTGWHI或PTGWLO命令之后立即对选定触发输入进行采样测试,测试所选的触发输入是否出现有效电平(每个PTG时钟一次)。

如果未出现触发信号,并且使能了步阶延时,则命令会等待步阶延时结束,然后再次对触发输入进行测试。当发现触发信号为真时,即完成命令执行,并立即开始执行后续命令。不插入步阶延时。

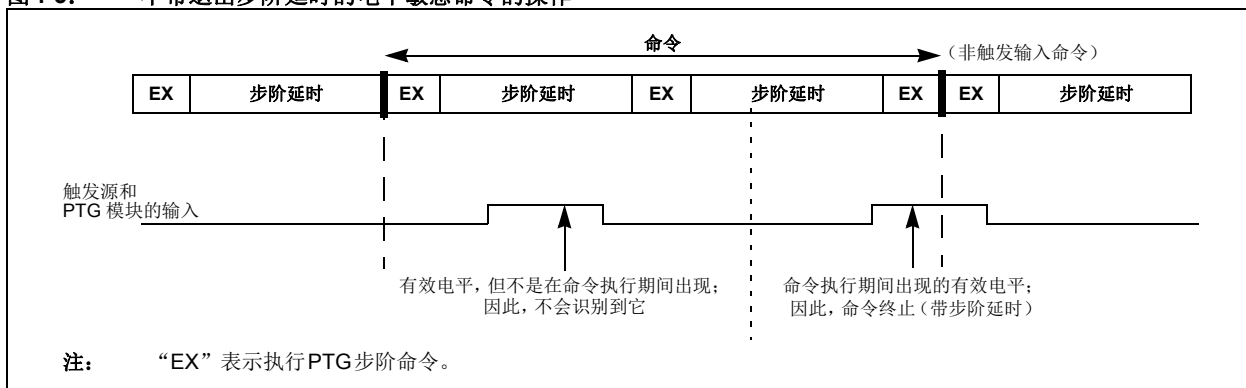
如果未出现触发信号,并且禁止了步阶延时,则命令会在下一个PTG时钟周期期间立即再次测试触发输入。当出现触发信号时,即完成命令执行,并立即开始执行后续命令。

注 1: 由于该工作模式是电平敏感的,所以如果输入触发电平在开始执行PTGWHI或PTGWLO命令时即为真,则将立即满足输入测试条件。

2: 输入不会进行锁存,因此它必须在命令执行时有效,这样才能识别到它。

图4-5给出了模式3操作的示例时序图。

图4-5: 不带退出步阶延时的电平敏感命令的操作



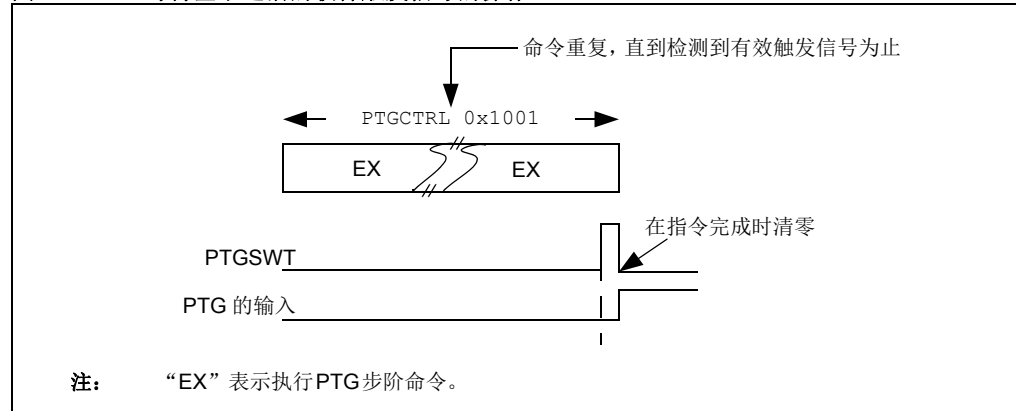
外设触发信号发生器 (PTG)

4.7.3 等待软件触发信号

用户可以通过设置PTGCTRL 0x1011 (边沿触发) 或PTGCTRL 0x1010 (电平触发) 命令来等待软件生成的触发信号。该触发信号通过将PTGSWT位 (PTGCST<10>) 置1来产生。

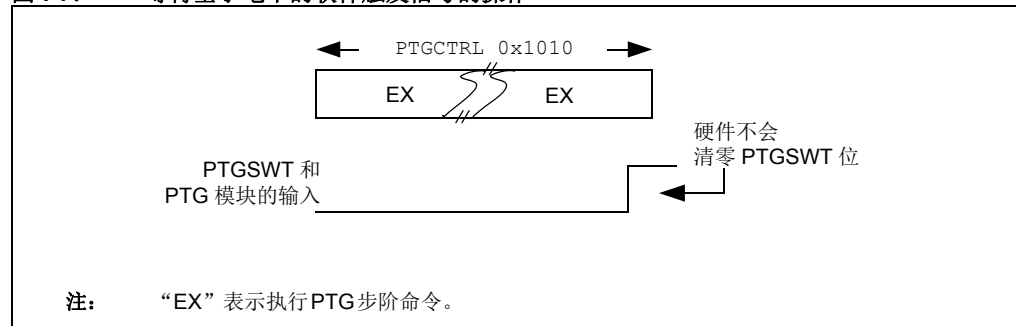
PTGCTRL 0x1011 命令只对PTGSWT位从0到1的变化敏感。此电平变化必须在命令执行期间发生; 否则, 命令将继续等待。在PTGCTRL 0x1011命令执行完成时, 硬件会自动清零PTGSWT位, 从而为下一条软件触发命令初始化该位。图4-6说明了等待基于边沿的软件触发信号的操作。

图4-6: 等待基于边沿的软件触发信号的操作



PTGCTRL 0x1010 命令对于PTGSWT位的电平敏感。该命令会等待, 直到PTGSWT = 1为止。如果在进入命令时PTGSWT = 1, 则将立即完成命令执行。PTGCTRL 0x1010命令不会自动清零PTGSWT位。如果需要, 用户应用程序可以在PTGCTRL 0x1010命令执行完毕后清零PTGSWT位。图4-7说明了等待基于电平的软件触发信号的操作。

图4-7: 等待基于电平的软件触发信号的操作



PTGSWT位与PTG步阶命令一起可以产生中断, 使用户可以协调PTG模块和应用软件之间的活动。

注: 电平敏感软件触发 (PTGCTRL 0x1010) 并非在所有器件上都可用。有关详细信息, 请参见具体器件的数据手册。

4.7.4 等待 GP 定时器

PTG 具有两个内部专用 16 位通用（General Purpose, GP）定时器（PTGT0 和 PTGT1），序列发生器可以使用它们来等待一个指定的周期。可以通过步阶命令来装入、修改或初始化 GP 定时器。

每个 GP 定时器都包含一个递增定时器（PTGTx）和一个限制寄存器（PTGTxLIM）。限制寄存器的值可以通过 CPU 写操作进行更改（禁止模块时），也可以通过 PTG 序列发生器进行更改（使能模块时）。从限制寄存器中读取的数据取决于 PTG 计数器/定时器可视性位（PTGIVIS）的状态。

在运行时，定时器将在 PTG 时钟的上升沿递增，这在 PTGCST 寄存器中定义。用户可以通过执行相应的 PTGCTRL 0x1000 或 PTGCTRL 0x1001 命令（等待选定的 GP 定时器），指定一个使用 GP 定时器的等待操作。

在等待 GP 定时器时，命令将等待，直到定时器（Timer0 或 Timer1）的值达到其相应的限制值（PTGT0LIM 或 PTGT1LIM）为止。在达到限制值时，即完成步阶命令执行，并开始执行下一条命令。此外，定时器还将清零，以备下次使用。当器件处于复位状态，或 PTG 模块被禁止（PTGEN = 0）时，所有定时器都将清零。

4.7.5 步阶命令延时

步阶延时定时器（SDLY）可以作为一种便利的方法来使每条步阶命令消耗特定的时间量。通常，用户需要指定一个等于外设功能持续时间（如 ADC 转换时间）的步阶延时。步阶延时使用户可以以控制速率生成触发输出信号，从而避免目标外设超负荷。

PTGSDLIM 寄存器用于定义每条步阶命令的额外持续时间（以 PTG 时钟数表示）。

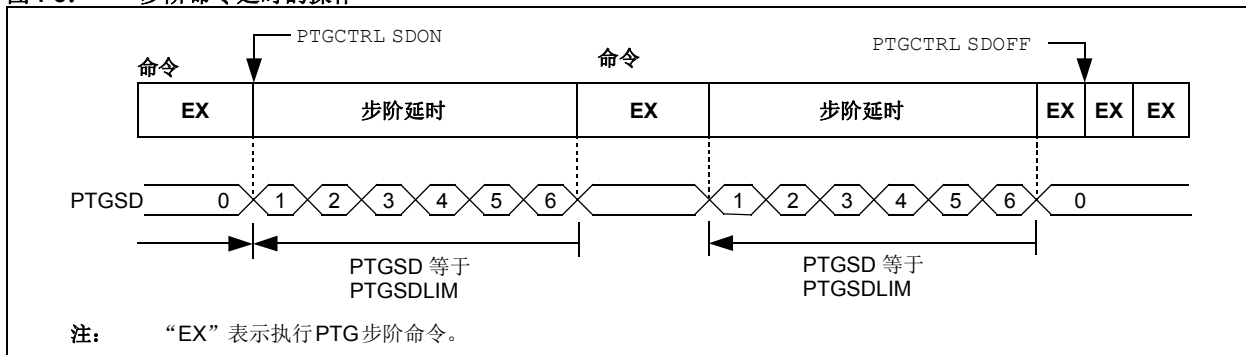
默认情况下，SDLY 会被禁止。用户可以通过 PTGCTRL 0x0110 或 PTGCTRL 0x0010 命令（命令可以放入步阶队列中）使能和禁止 SDLY。

在工作时，SDLY 将按 PTGCST 寄存器中定义的 PTG 时钟速率进行递增。PTGSDLIM 寄存器的值将作为步阶延时定时器限制值。步阶延时在执行每条命令之后插入，从而使所有步阶命令发生停顿，直到 PTGSD 定时器达到其限制值为止。在达到限制值时，即完成命令执行，并开始执行下一条命令。定时器也将在执行每条命令过程中清零，从而为执行下一条命令做好准备。

注： 在使能步阶延时定时器时，如果 PTGSDLIM 寄存器值为 0x0000，则不会插入额外的 PTG 时钟。如果 PTGSDLIM 寄存器值为 0x0001，则在使能步阶延时定时器之后，会在每条后续指令中插入单个 PTG 步阶延时（1 个 PTG 时钟）。

边沿敏感命令（PTGCTRL 0x1011 和 PTGWHI/PTGWLO，工作于边沿敏感模式时）的触发源具有一个位于序列发生器外部的额外硬件，用以识别相应的边沿变化。该硬件会在每条命令结束时复位，以维持这些输入触发的边沿敏感性质。如果在执行步阶命令之后，在已插入的步阶延时期间再次出现有效边沿，它不会被任何后续步阶命令识别到。图 4-8 说明了步阶命令延时的操作。

图 4-8: 步阶命令延时的操作



4.8 PTG看门狗定时器

由于在执行以下命令时，PTG会无限期地等待外部事件，所以需要看门狗定时器 (WDT)：

- 等待硬件触发正边沿或高电平状态 (PTGWHI)
- 等待硬件触发负边沿或低电平状态 (PTGWLO)

WDT在命令开始执行时使能并开始计数。在命令完成执行时，它会在插入任何步阶延时之前被禁止。所有其他命令均执行预定义的周期数。

注： 在执行PTGCTRL 0x1011或PTGCTRL 0x1010命令期间不需要PTG看门狗定时器。假定将通过其他方式来监视器件是否正确工作。

4.8.1 操作概述

如果预期事件未能在WDT超时周期结束之前到达，PTG模块将：

1. 中止正在进行的命令（视为失败）。
2. 暂停序列发生器 (PTGSTRT = 0)。
3. 设置PTGWDTO = 1。
4. 向处理器发出看门狗定时器错误中断。

用户可以使用看门狗定时器错误中断，也可以通过定期查询PTGWDTO位 (PTGCST<6>) 来确定是否发生WDT事件。

4.8.2 配置

WDT通过设置PTGWDT<2:0>位 (PTGCON<2:0>) 进行配置。WDT会对PTG时钟进行计数，计数数量由PTGCON寄存器中的PTGCLK<2:0>和PTGDIV<4:0>位定义。有关详细信息，请参见第4.2节“PTG时钟选择”。WDT超时计数值通过PTGWDT<2:0>位进行选择，并且当PTGWDT<2:0> = 0x000时，它将被禁止。

注 1： WDT将在插入任何步阶延时之前被禁止；因此，在计算适合的WDT超时值时，用户不需要考虑步阶延时。

2： 当PTGSTRT = 1（序列发生器正在执行命令）时，PTGCON寄存器的某些位是只读的。请参见寄存器2-2。

4.8.3 看门狗定时器事件恢复

如果发生WDT事件，用户可以选择执行必要的操作来查明并解决问题，然后再继续执行步阶命令序列，也可以简单地重新启动序列。

要清零PTGWDTO位，并从步阶队列起始处开始重新启动PTG序列发生器，可以禁止 (PTGEN = 0) 并重新使能 (PTGEN = 1) PTG模块，然后重新开始执行 (PTGSTRT = 1)。

或者，由于序列发生器只是暂停（不是复位），用户可以选择通过检查PTGQPTR寄存器来确定哪条步阶命令是问题的来源，然后可以执行修正操作。引起问题的命令会在PTGQPTR更新之前被中止。因此，在WDT事件之后，它将仍然寻址发生失败的命令。在PTGWDTO位清零之后，可以通过设置PTGSTRT = 1在同一命令处重新启动步阶队列。

注： 用户应在WDT事件之后清零PTGWDTO位。未清零该位不会影响后续的操作，但将无法通过该位查询任何未来的WDT事件。

4.9 PTG 模块输出

PTG 模块可以通过执行一些特定步阶命令来产生触发、中断和选通数据输出。

4.9.1 触发输出

PTG 模块最多可以产生 32 个独特的触发输出信号。存在两种类型的触发输出功能：

- 独立
- 广播

PTG 模块可以使用 PTGTRIG 命令在任意一个触发输出上产生独立触发输出。触发输出特定于器件。关于可用性，请参见具体器件的数据手册。

独立触发输出通常用于触发独立的 ADC 输入转换操作，但它可以分配给任意功能，包括通用 I/O 端口。对兼容外设（如 ADC 模块）使用 PTG 模块时，PTG 的独立触发输出信号将独立分配给 ADC 模块中的特定模拟输入转换控制器。

广播触发输出功能通过 PTGBTE/PTGBTEH 寄存器指定。PTGBTE/PTGBTEH 寄存器中的每个位都对应一个关联的独立触发输出。如果 PTGBTE/PTGBTEH 寄存器中的某个位置 1，并执行广播触发步阶命令（PTGCTRL 0x1111），则相应的独立触发输出会被置为有效。广播触发输出使用户可以使用单个步阶命令同时产生大量触发输出。

4.9.2 中断输出

PTG 模块最多可以产生 16 个独特的中断请求信号。这些信号对于通过与应用软件进行交互而产生更复杂功能非常有用。

PTG 模块可以通过使用 PTGIRQ 命令生成独立的中断脉冲。

4.10 选通输出

PTG 模块的选通输出可以用于从 PTG 模块输出数据。通常，此输出连接到 ADC 通道选择寄存器，从而允许 PTG 控制 ADC 通道的切换。具体器件数据手册将指示 PTG 选通输出如何连接到其他外设。

PTGCTRL 0x1110 命令会将 PTGL0 寄存器的内容写入选通输出。PTGL0 寄存器可以通过使用 PTGADD 和 PTGCOPY 命令进行修改。

PTGCTRL 0x1100 命令会将 PTGC0 寄存器的内容写入选通输出。PTGCTRL 0x1101 命令会将 PTGC1 寄存器的内容写入选通输出。

外设触发信号发生器 (PTG)

4.10.1 输出时序

当相应的步阶命令在PTG执行时钟上升沿开始执行时（即，在步阶延时定时器指定的任何额外时间之前），PTG状态机会将所有触发、中断和数据选通输出在内部置为有效。

注： 如果某条命令触发了脉宽延时计数器，计数器会与PTG时钟同步复位，从而终止脉冲（受1个PTG时钟周期最小脉宽的限制）。

在脉冲模式（PTGTOGL = 0）下，触发输出信号的宽度由PTGPWD<3:0>位（PTGCON<7:4>）决定，它可以为介于1和16个PTG时钟周期之间的任意值。默认值为1个PTG时钟周期。

工作于翻转模式（PTGTOGL = 1）时，请参见第4.10.1.2节“TRIG取反，PTGTOGL = 1时”。

在通过PTGCTRL 0x1111广播触发命令进行全局控制时，TRIG输出脉冲宽度将由PTGPWD<3:0>位（PTGCON<7:4>）决定，它可以为介于1和16个PTG时钟周期之间的任意值。默认值为1个PTG时钟周期。

注： 通过PTGCTRL 0x1111广播触发命令生成的触发信号只能工作于脉冲模式（即，PTGTOGL = “无关”）。

4.10.1.1 TRIG取反，PTGTOGL = 0时

如果在PTGTOGL位（PTGCST<12>）= 0时生成独立触发输出，或者如果生成广播触发输出，TRIG输出脉冲宽度将由PTGPWD<3:0>位决定。

4.10.1.2 TRIG取反，PTGTOGL = 1时

如果在PTGTOGL位（PTGCST<12>）= 1时生成独立触发输出，TRIG输出将保持置1，直到再次执行PTGTRIG命令为止。在PTGTRIG命令开始执行时，TRIG输出会在PTG执行时钟开始处翻转。

注： PTGTOGL位对PTGCTRL 0x1111多重触发（广播）生成命令的操作没有任何作用。例外情形如下：

- 所有广播触发的脉冲宽度总是由PTGPWD<3:0>位决定。
- 如果目标触发输出已处于逻辑1状态（因为PTGTOGL有效），PTGCTRL 0x1111命令将没有任何作用，并且触发输出将保持为逻辑1。

4.11 停止序列发生器

当PTG模块被禁止（PTGEN = 0）时，PTG时钟会被禁止（触发脉冲计数器除外），序列发生器会停止执行，并且模块进入其最低功耗状态。PTGSTRT、PTGSWT、PTGWDTO和PTGQPTR<4:0>位全都复位。所有其他位和寄存器不会被修改。当PTGEN = 0时，可以读写所有控制寄存器。

当PTGEN位清零时，如果正在进行的命令在等待以下任意操作，则会立即中止该命令：

- 来自另一个源的输入
- 定时器匹配
- 步阶延时结束（更多信息，请参见第4.7.5节“步阶命令延时”）

在禁止PTG模块之前，将允许所有其他命令完成执行。

当PTG模块暂停时，所有控制寄存器将保持它们的当前状态。PTG模块可以由用户通过清零PTGSTRT位而暂停，也可以在看门狗定时器超时（这也会清零PTGSTRT位）暂停。更多信息，请参见第4.8节“PTG看门狗定时器”。

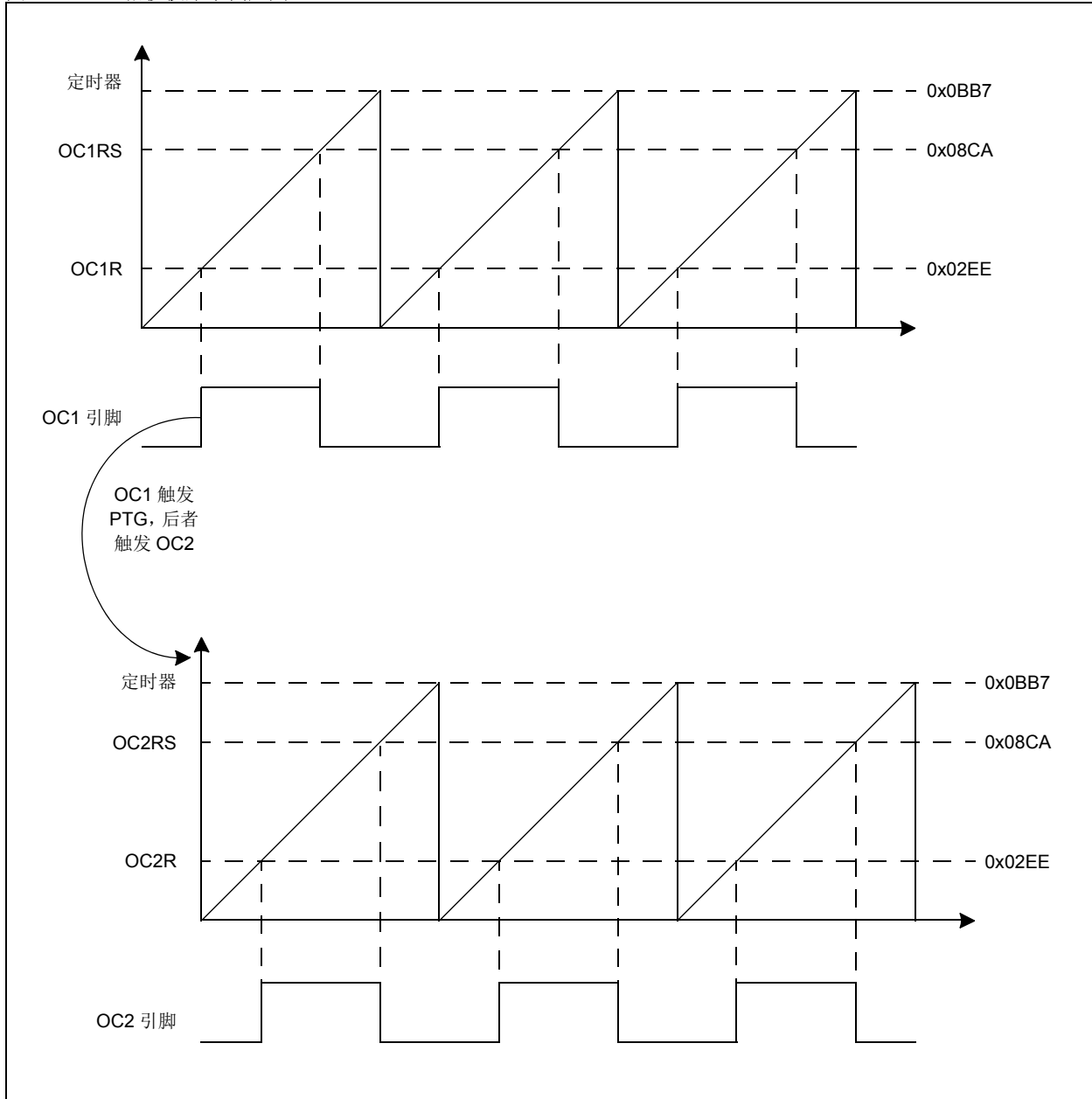
5.0 应用示例

5.1 生成相移波形

图5-1给出了一个用户需要生成相移波形的应用示例。在该示例中，输出比较1使用Timer2作为同步源来产生脉冲。脉冲的上升沿是PTG模块的触发输入。当检测到触发信号时，PTG模块将触发输出比较2，后者使用PTG模块作为同步源来生成相移波形。

注： 此应用示例仅适用于具有输出比较和Timer2外设的器件。关于可用性，请参见具体器件的数据手册。

图5-1: 相移波形示例应用



外设触发信号发生器 (PTG)

例5-1给出了用于生成相移波形的代码。

例5-1: 生成相移波形

```
#include <xc.h>

_FOSCSEL(FNOSC_FRC);
_FOSC(FCKSM_CSECMD & POSCMD_XT & OSCIOFNC_OFF & IOL1WAY_OFF);
_FWDT(FWDTEN_OFF);
_FPOR(ALT12C1_ON & ALT12C2_ON);
_FICD(ICS_PGD2 & JTAGEN_OFF);

void Init_Timer(void);
void Init_Ptg(void);
void Init_PPS(void);
void Init_OC1(void);
void Init_OC2(void);

int main(void)
{
    // Configure the device PLL to obtain 60 MIPS operation. The crystal frequency is 8 MHz.
    // Divide 8 MHz by 2, multiply by 60 and divide by 2. This results in Fosc of 120 MHz.
    // The CPU clock frequency is Fcy = Fosc/2 = 60 MHz.
    PLLFBD = 58;                /* M = 30 */
    CLKDIVbits.PLLPOST = 0;     /* N1 = 2 */
    CLKDIVbits.PLLPRE = 0;     /* N2 = 2 */
    OSCTUN = 0;

    // Initiate Clock Switch to Primary
    // Oscillator with PLL (NOSC= 0x3)
    __builtin_write_OSCCONH(0x03);
    __builtin_write_OSCCONL(0x01);
    while (OSCCONbits.COSC != 0x3);
    while ( _LOCK == 0);        /* Wait for PLL lock at 60 MIPS */

    Init_Timer();
    Init_Ptg();
    Init_PPS();
    Init_OC1();
    Init_OC2();

    PTGCSTbits.PTGEN = 1;      /* Enable the PTG */
    PTGCSTbits.PTGSTRT = 1;    /* Start the PTG */
    T2CONbits.TON = 1;         /* Start the timer */

    while(1);
}

void Init_Timer( void )
{
    // Initialize and enable Timer2
    T2CON = 0x0000;            /* Timer reset */
    TMR2 = 0x0000;            /* Clear timer register */
    PR2 = 0x0BB7;             /* Load the period value */
}

void Init_Ptg( void )
{
    PTGCST = 0; // Clear the control/status register
    PTGCON = 0; // Clear the control register

    /* Program the command sequence */
    _STEP0 = PTGWHI(0x7);     /* Wait for OC1 input trigger event */
    _STEP1 = PTGTRIG(0x1);    /* Trigger PTG02 (trig/sync for OC2) */
    _STEP2 = PTGJMP(0x0);     /* Jump to _STEP0 */
}

```

关于PTG命令定义, 请参见例3-1。关于PTGCTRL、PTGADD和PTGCOPY命令选项, 请参见例3-2和例3-3。

dsPIC33/PIC24 系列参考手册

例 5-1: 生成相移波形 (续)

```
void Init_PPS(void)
{
    _RP39R = 0x10;           // Set up the PPS for OC1
    _RP40R = 0x11;           // Set up the PPS for OC2
}

void Init_OC1(void)
{
    OC1R = 0x02EE;           // Initialize the compare register
    OC1RS = 0x08CA;          // Initialize the secondary compare register

    // Initialize Output Compare Module
    OC1CON1 = 0x0;           // Clear all control bits
    OC1CON2 = 0x0;           // Clear all control bits
    OC1CON1bits.OCTSEL = 0x7; // Select peripheral clock as clock source
    OC1CON2bits.SYNCSEL = 0xC; // Select Timer2 as sync source
    OC1CON1bits.OCM = 0x5;    // Double compare continuous pulse mode
}

void Init_OC2(void)
{
    OC2R = 0x02EE;           // Initialize the compare register
    OC2RS = 0x08CA;          // Initialize the secondary compare register

    // Initialize Output Compare Module
    OC2CON1 = 0x0;           // Clear all control bits
    OC2CON2 = 0x0;           // Clear all control bits
    OC2CON1bits.OCTSEL = 0x7; // Select peripheral clock as clock source
    OC2CON2bits.SYNCSEL = 0xA; // Select PTG as sync source
    OC2CON1bits.OCM = 0x5;    // Double compare continuous pulse mode
}
```

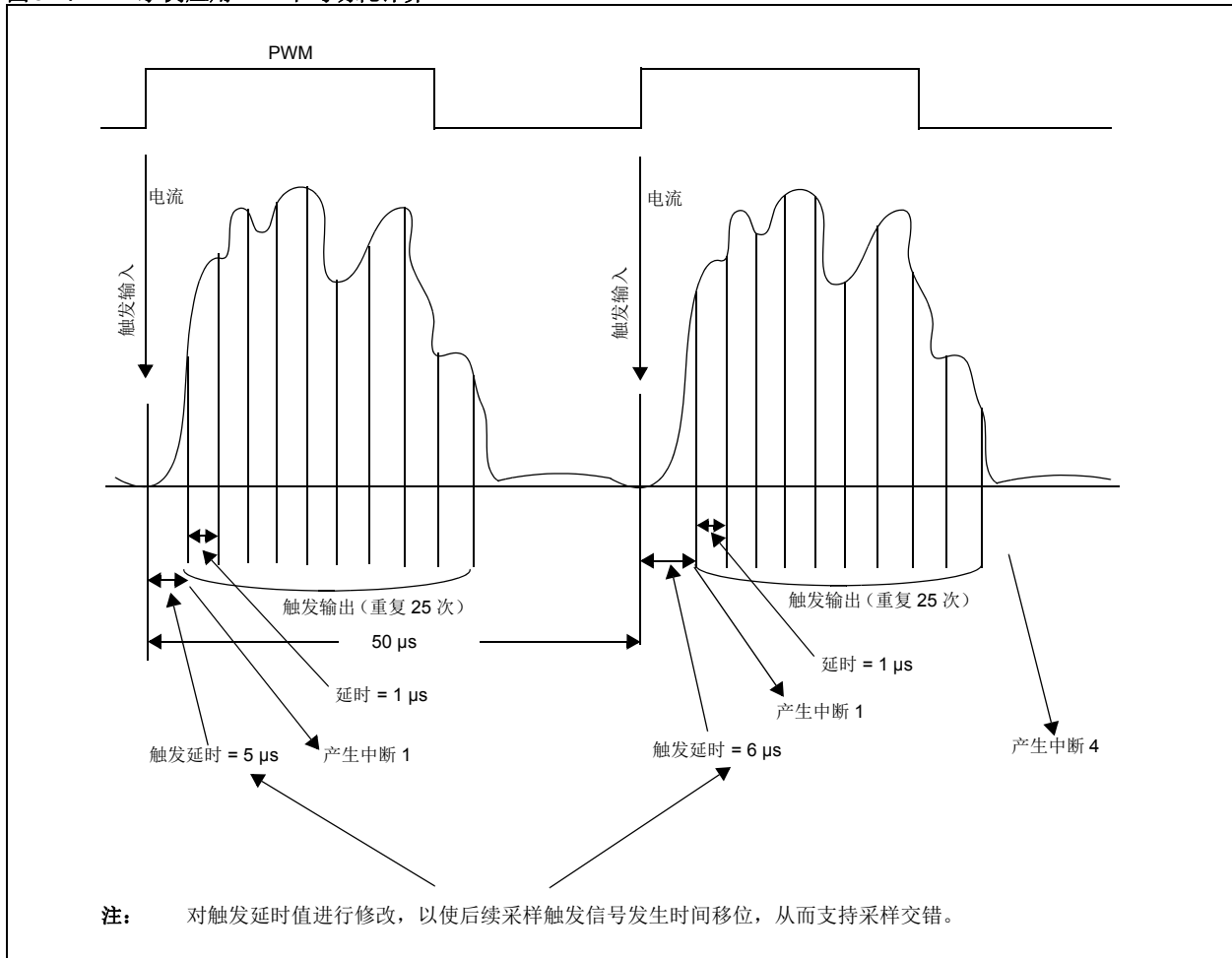
5.2 在多个周期中交错采样

图5-2显示了一个用户需要精确测量系统功耗的应用的波形，该系统中的电流负载高度依赖于温度、电压和用户应用。电流波形会因用户使用情形而发生很大变化，但若干个PWM周期的波形是相对稳定的。

该示例的目的是以交错方式采集若干个PWM周期的许多电流和/或电压读数。数据在采集过程中存储在存储器中，之后经过处理（积分）来产生一个精确的功耗值。

该示例说明了一种情形，其中通过软件精确调度ADC采样是不实际或不可行的。

图5-2: 示例应用——平均功耗计算



5.3 交错采样步阶命令编程

本节介绍实现图5-2所示时序的步阶命令编程。

作出如下假设：

1. 触发输入1连接到PWM信号。PWM信号的上升沿将启动序列。
2. 输出触发3连接到ADC模块。该信号向ADC模块发送命令，使之开始采样和转换过程。
3. 中断1用于向处理器指示有一个子序列已启动（提供状态）。
4. 中断4用于向处理器指示整个序列已经完成。
5. ADC时钟选择作为PTG时钟源。
6. ADC时钟为14 MHz。
7. 初始触发延时为5 μ s。
8. 第二个触发延时为6 μ s。
9. 在每个PWM周期中，ADC将被触发25次。
10. 基本序列将执行两次。

初始化以下控制寄存器：

- PTGT0LIM = 0x0046 (5 μ s x 14 个时钟/ μ s)
- PTGT1LIM = 0x000B ([1 μ s x 14 个时钟/ μ s] - 3 个步阶时钟)
- PTGC0LIM = 0x0018 (总共25个内循环迭代)
- PTGC1LIM = 0x0001 (总共两个外循环迭代)
- PTGHOLD = 0x0046 (5 μ s x 14 个时钟/ μ s)
- PTGADJ = 0x000E (1 μ s x 14 个时钟/ μ s)
- PTGSDLIM = 0x0000 (无步阶延时)
- PTGBTE = 0x0000 (无广播触发)
- PTGQPTR = 0x0000 (步阶队列起始位置)
- PTGCST = 0x8200 (在初始化PTGQPTR之后)

例 5-2: PTGQUEn 中的步阶命令

```
void PTG_InterleavedSamplingQueue(void)
{
    // Outer loop
    _STEP0 = PTGWHI(0x1);           // Wait for positive edge trigger 1
    _STEP1 = PTGCTRL(t0Wait);       // Start PTGT0, wait for time out
    _STEP2 = PTGIRQ(0x1);           // Generate IRQ 1
    // Inner loop
    _STEP3 = PTGTRIG(0x3);          // Generate output trigger 3
    _STEP4 = PTGCTRL(t1Wait);       // Start PTGT1, wait for time out
    _STEP5 = PTGJMPC0(0x3);         // Go to STEP3 if PTGC0 != PTGC0LIM, increment PTGC0
    // End inner loop
    _STEP6 = PTGADD(t0Limit);        // Add PTGADJ to PTGT0LIM
    _STEP7 = PTGJMPC1(0x0);
    // End outer loop

    _STEP8 = PTGIRQ(0x4);           // Generate IRQ 4
    _STEP9 = PTGCOPY(0x8);          // Copy PTGHOLD to PTGT0LIM (restore original value)
    _STEP10 = PTGJMP(0x0);          // Jump to start of queue
}
```

关于PTG命令定义，请参见例3-1。关于PTGCTRL、PTGADD和PTGCOPY命令选项，请参见例3-2和例3-3。

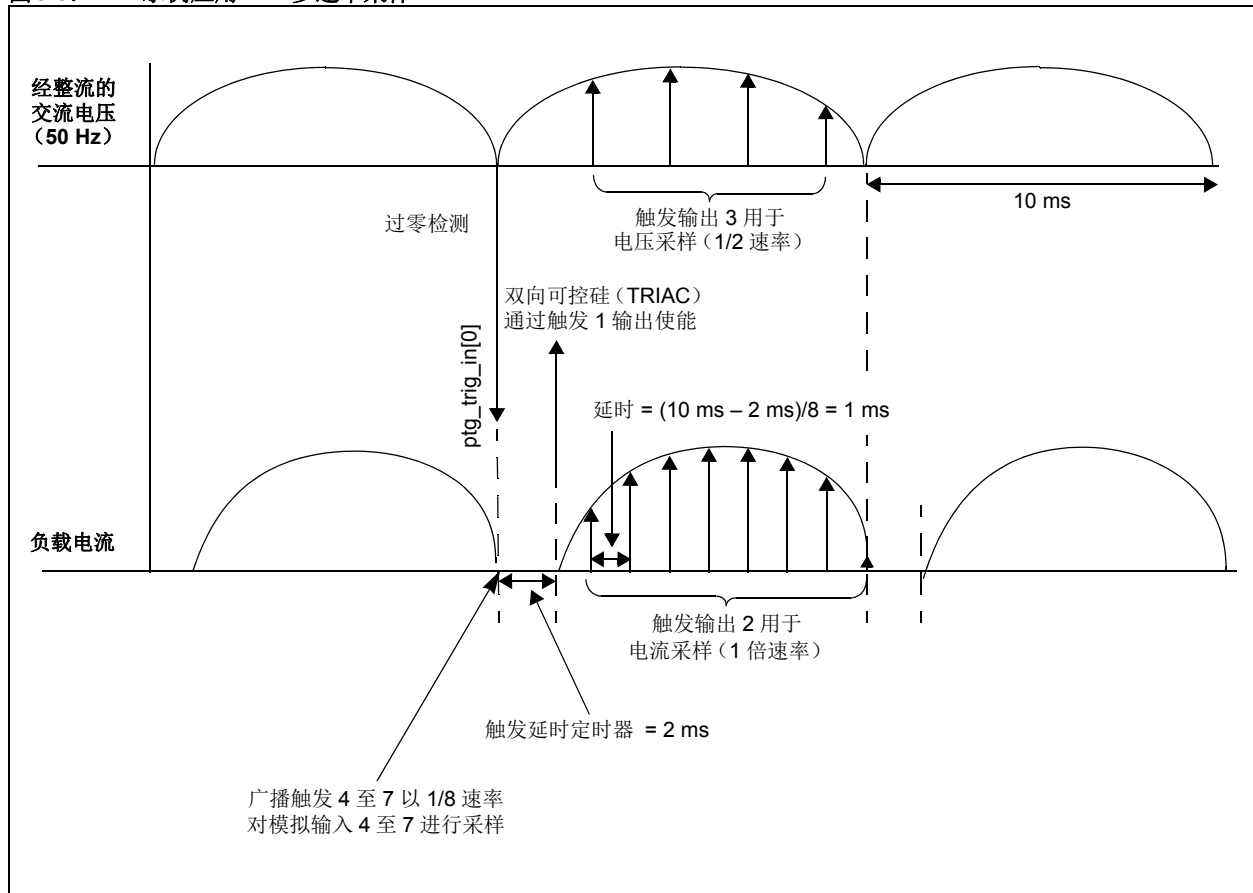
外设触发信号发生器 (PTG)

5.4 多速率采样

图5-3显示了一个应用示例，该应用的目标是以快速速率（1倍速率）对ADC输入进行采样，以较慢速率（1/2速率）对第二个模拟输入进行采样，以1/8速率对模拟输入4至7进行采样。该示例是一个电机控制应用，它使用一个在交流线过零之后的指定时间触发的可控硅整流器（Silicon Controlled Rectifier, SCR）。

虽然该示例使用简单的二进制采样比，但PTG模块可以生成范围极宽的采样比来满足应用的需求。

图5-3: 示例应用——多速率采样



5.5 多速率采样步阶命令编程

本节介绍实现图5-3所示时序的步阶命令编程。

作出如下假设：

- 触发输入0连接到过零检测。过零检测信号的上升沿将启动序列
- 从触发输入0到产生触发输出1的触发延时为2 ms
- 触发输出1用于使能应用电路中的SCR
- 触发输出2连接到ADC，从而以1 ms的时间间隔触发电流测量采样
- 触发输出3连接到ADC，从而以2 ms的时间间隔触发电源电压测量采样
- 触发输出4、5、6和7连接到ADC，每个周期对其他数据值进行一次采样
- 选择ADC时钟作为PTG时钟源
- ADC时钟为14 MHz

初始化以下控制寄存器：

- PTGT0LIM = 0x6D60 (2 ms x 14 个时钟/μs)
- PTGT1LIM = 0x36B0 (1 ms x 14 个时钟/μs)
- PTGC0LIM = 0x0018 (总共25个内循环迭代)
- PTGC1LIM = 0x0001 (总共两个外循环迭代)
- PTGHOLD = 0x0000 (不使用)
- PTGADJ = 0x0000 (不使用)
- PTGSDLIM = 0x0000 (无步阶延时)
- PTGBTE = 0x00F0 (使能广播触发4-7)
- PTGQPTR = 0x0000 (步阶队列起始位置)
- PTGCST = 0x8200 (在初始化PTGQPTR之后)

6.0 节能模式

PTG 模块支持三种节能模式：

- 禁止：在该模式下不为PTG模块提供时钟
- 空闲：处理器内核和选定外设将被关闭
- 休眠：整个器件关闭

6.1 禁止模式

当PTGEN = 1时，模块被视为处于活动模式，并完全供电和全功能工作。当PTGEN = 0时，模块将被关闭。电路的PTG时钟部分将被禁止，以最大限度地节省电流。只有控制寄存器保持正常读写功能，以允许软件更改模块的工作模式。模块序列发生器会发生复位。

6.2 空闲模式

要在PTG模块处于空闲模式时继续使模块全功能工作，必须在进入空闲模式之前清零PTGSIDL位。如果PTGSIDL = 1，模块在空闲模式下的行为将与休眠模式下相同。

6.3 休眠模式

如果PTG模块在模块使能 (PTGEN = 1) 时进入休眠模式，模块将暂停在其当前状态，直到时钟执行恢复为止。应当避免这种情况，因为它可能会导致意外的操作。建议在进入休眠模式之前，先有序地关闭所有外设。

7.0 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，但是概念是相关的，通过适当修改即可使用，但在使用中可能会受到一定限制。当前与外设触发信号发生器（PTG）模块相关的应用笔记包括：

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)。

8.0 版本历史

版本A (2009年9月)

这是本文档的初始版本。

版本B (2017年8月)

- 章节:
 - 将寄存器映射移动到文档的前面部分, 替换“**状态和控制寄存器**”部分。
 - 更新了第 1.0 节“简介”、第 3.0 节“步阶命令和格式”、第 4.0 节“模块工作原理”、第 4.1 节“PTG 说明”、第 4.2 节“PTG 时钟选择”、第 4.2.3 节“模块使能延时”、第 4.2.1 节“时钟源选择”、第 4.3 节“基本操作”、第 4.4 节“控制寄存器访问”、第 4.6 节“命令循环控制”、第 4.7 节“序列发生器操作”、第 4.7.4 节“等待 GP 定时器”、第 4.8 节“PTG 看门狗定时器”、第 4.10 节“选通输出”、第 4.10.1 节“输出时序”、第 4.10.1.2 节“TRIG 取反, PTGTOGL = 1 时”、第 4.11 节“停止序列发生器”和第 5.1 节“生成相移波形”。
 - 删除了第 4.8 节“步阶命令”。
 - 图:
 - 更新了图 1-1。
 - 增加了图 4-1。
 - 删除了图 4-9: “GP 定时器”和图 4-11: “立即数 0”。
 - 示例:
 - 更新了例 5-2。
 - 删除了例 32-1: “生成相移波形”。
 - 表:
 - 更新了表 2-1、表 3-1 和表 3-2。
 - 寄存器:
 - 更新了寄存器 2-1、寄存器 2-2、寄存器 2-3、寄存器 2-4 和寄存器 2-14。
- 对整篇文档的语法进行了少量修正。

dsPIC33/PIC24 系列参考手册

注:

请注意以下有关 Microchip 器件代码保护功能的要点：

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

代码保护功能处于持续发展中。Microchip 承诺将不断改进产品的代码保护功能。任何试图破坏 Microchip 代码保护功能的行为均可视为违反了《数字千年版权法案 (Digital Millennium Copyright Act)》。如果这种行为导致他人在未经授权的情况下，能访问您的软件或其他受版权保护的成果，您有权依据该法案提起诉讼，从而制止这种行为。

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

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

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 and 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。

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

商标

Microchip 的名称和徽标组合、Microchip 徽标、AnyRate、AVR、AVR 徽标、AVR Freaks、BitCloud、chipKIT、chipKIT 徽标、CryptoMemory、CryptoRF、dsPIC、FlashFlex、flexPWR、Heldo、JukeBlox、KeeLoq、Kleer、LANCheck、LINK MD、maXStylus、maXTouch、MediaLB、megaAVR、MOST、MOST 徽标、MPLAB、OptoLyzer、PIC、picoPower、PICSTART、PIC32 徽标、Prochip Designer、QTouch、SAM-BA、SpyNIC、SST、SST 徽标、SuperFlash、tinyAVR、UNI/O 及 XMEGA 均为 Microchip Technology Inc. 在美国和其他国家或地区的注册商标。

ClockWorks、The Embedded Control Solutions Company、EtherSynch、Hyper Speed Control、HyperLight Load、IntelliMOS、mTouch、Precision Edge 和 Quiet-Wire 均为 Microchip Technology Inc. 在美国的注册商标。

Adjacent Key Suppression、AKS、Analog-for-the-Digital Age、Any Capacitor、AnyIn、AnyOut、BodyCom、CodeGuard、CryptoAuthentication、CryptoAutomotive、CryptoCompanion、CryptoController、dsPICDEM、dsPICDEM.net、Dynamic Average Matching、DAM、ECAN、EtherGREEN、In-Circuit Serial Programming、ICSP、INICnet、Inter-Chip Connectivity、JitterBlocker、KleerNet、KleerNet 徽标、memBrain、Mindi、MiWi、motorBench、MPASM、MPF、MPLAB Certified 徽标、MPLIB、MPLINK、MultiTRAK、NetDetach、Omniscient Code Generation、PICDEM、PICDEM.net、PICkit、PICtail、PowerSmart、PureSilicon、QMatrix、REAL ICE、Ripple Blocker、SAM-ICE、Serial Quad I/O、SMART-I.S.、SQI、SuperSwitcher、SuperSwitcher II、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。

© 2011-2019, Microchip Technology Inc. 版权所有。

ISBN: 978-1-5224-4241-7



全球销售及服务中心

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

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

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

休斯敦 Houston, TX
Tel: 1-281-894-5983

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

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

罗利 Raleigh, NC
Tel: 1-919-844-7510

纽约 New York, NY
Tel: 1-631-435-6000

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

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

亚太地区

中国 - 北京
Tel: 86-10-8569-7000

中国 - 成都
Tel: 86-28-8665-5511

中国 - 重庆
Tel: 86-23-8980-9588

中国 - 东莞
Tel: 86-769-8702-9880

中国 - 广州
Tel: 86-20-8755-8029

中国 - 杭州
Tel: 86-571-8792-8115

中国 - 南京
Tel: 86-25-8473-2460

中国 - 青岛
Tel: 86-532-8502-7355

中国 - 上海
Tel: 86-21-3326-8000

中国 - 沈阳
Tel: 86-24-2334-2829

中国 - 深圳
Tel: 86-755-8864-2200

中国 - 苏州
Tel: 86-186-6233-1526

中国 - 武汉
Tel: 86-27-5980-5300

中国 - 西安
Tel: 86-29-8833-7252

中国 - 厦门
Tel: 86-592-238-8138

中国 - 香港特别行政区
Tel: 852-2943-5100

中国 - 珠海
Tel: 86-756-321-0040

台湾地区 - 高雄
Tel: 886-7-213-7830

台湾地区 - 台北
Tel: 886-2-2508-8600

台湾地区 - 新竹
Tel: 886-3-577-8366

亚太地区

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

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

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

印度 India - Pune
Tel: 91-20-4121-0141

日本 Japan - Osaka
Tel: 81-6-6152-7160

日本 Japan - Tokyo
Tel: 81-3-6880-3770

韩国 Korea - Daegu
Tel: 82-53-744-4301

韩国 Korea - Seoul
Tel: 82-2-554-7200

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

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

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

新加坡 Singapore
Tel: 65-6334-8870

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

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

欧洲

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

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

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

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

德国 Germany - Garching
Tel: 49-8931-9700
德国 Germany - Haan
Tel: 49-2129-3766400

德国 Germany - Heilbronn
Tel: 49-7131-67-3636

德国 Germany - Karlsruhe
Tel: 49-721-625370

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

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

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

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

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

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

挪威 Norway - Trondheim
Tel: 47-7288-4388

波兰 Poland - Warsaw
Tel: 48-22-3325737

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

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

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

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

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