FT61F02X

PWM 增强型 Application note

目录

FT61F02x PWM 增强型应用

1. 增强型捕捉/比较/PWM(ECCP)

增强型捕捉/比较/PWM 模块（ECCP）是一种用户可用来对不同事件进行定时和控制的外设。在捕捉模式下，此外设可对事件的持续时间定时。比较模式使用户可在一段预定时长后触发外部事件。PWM 模式可生成频率可变的脉宽调制信号和占空比。

ECCP	定时器资源
捕捉	Timer1
比较	Timer1
PWM	Timer2

表 1-1 ECCP 各模式所需的定时器资源

1.1. ECCP 相关寄存器汇总

CCP1M[3:2]	P1M[1:0]	工作模式	PWM 输出	IO 控制
00 / 01 / 10	xx	非 PWM 模式	-	P1A 配置为捕捉/比较输入 P1B、P1C 和 P1D 配置为端口引脚
11	00	PWM 模式	单输出	P1A 调制； P1B、P1C 和 P1D 配置为端口引脚
	01		全桥正向输出	P1D 调制；P1A 有效； P1B 和 P1C 无效
	10		半桥输出	P1A 和 P1B 调制(带有死区控制)； P1C 和 P1D 配置为端口引脚
	11		全桥反向输出	P1B 调制； P1C 有效；P1A 和 P1D 无效

表 1-2 ECCP 模块 PWM 输出模式 P1M 配置

名称	状态			寄存器	地址	复位值
CCPR1L	捕捉 / 比较 / PWM寄存器低8位			CCPR1L[7:0]	0x13	RW-xxxx xxxx
CCPR1H	捕捉 / 比较 / PWM寄存器高8位			CCPR1H[7:0]	0x14	RW-xxxx xxxx
P1M	PWM输出配置位	详情请见 **表 1-2**		CCP1CON[7:6]	0x15	RW-00
DC1B	PWM 占空比低 2 位 (ECCP 模块 PWM 模式有效，高 8 位在 CCPR1L 中)			CCP1CON[5:4]		RW-00
CCP1M	ECCP模式选择			CCP1CON[3:0]		RW-0000
	值	模式	描述			
	0000	关闭	复位ECCP模块			

名称	状态			寄存器	地址	复位值
	0001	未使用	(保留)			
	0010	比较模式	匹配时翻转输出[1]			
	0011	未使用	(保留)			
	0100	捕捉模式	每个下降沿 1			
	0101		每个上升沿 1			
	0110		每 4 个上升沿 1			
	0111		每 16 个上升沿 1			
	1000	比较模式	匹配时输出置 1 1			
	1001		匹配时输出清零 1			
	1010		匹配时产生软件中断 1 [2]			
	1011		触发特殊事件 1 [3]			
	1100	PWM模式	P1A 和 P1C 高电平有效； P1B 和 P1D 高电平有效			
	1101		P1A 和 P1C 高电平有效； P1B 和 P1D 低电平有效			
	1110		P1A和P1C低电平有效； P1B和P1D高电平有效			
	1111		P1A和P1C低电平有效； P1B和P1D低电平有效			
PRSEN	PWM重启使能位 1 = 使能 (退出关闭事件时ECCPASE位自动清0) 0 = 关闭 (ECCPASE位必须软件清0) 注：自动关闭时有效			PWM1CON[7]	0x16	RW-0
PDC	PWM延时计数寄存器 预定PWM信号应转变为有效与PWM信号实际转为有效之间的指令周期数			PWM1CON[6:0]	0x16	RW-000 0000
ECCPASE	ECCP 自动关闭事件状态位 1 = 发生了自动关闭事件(ECCP 输出处于关闭状态) 0 = ECCP输出正常工作			ECCPAS[7]	0x17	RW-0

[1] CCP1IF 位置 1
[2] CCP1 引脚不受影响
[3] CCP1 复位 TMR1，且如果 A/D 模块被使能，启动一次 A/D 转换

名称	状态		寄存器	地址	复位值
ECCPAS	ECCP 自动关闭源选择位 000 = 禁止自动关闭 001 = 比较器 1 输出 C1OUT 变高 010 = 比较器 2 输出 C2OUT 变高 011 = 比较器 1 或 2 之一输出变高 100 = INT 引脚电压为 VIL 101 = INT 引脚电压为 VIL 或比较器 1 输出变高 110 = INT 引脚电压为 VIL 或比较器 2 输出变高 111 = INT引脚电压为VIL或比较器1/2之一输出变高		ECCPAS[6:4]		RW-000
PSSAC	P1A 和 P1C 引脚关闭状态控制位 00 = 驱动引脚 P1A 和 P1C 为 0 01 = 驱动引脚 P1A 和 P1C 为 1 1x = P1A和P1C引脚为三态		ECCPAS[3:2]		RW-00
PSSBD	P1B 和 P1D 引脚关闭状态控制位 00 = 驱动引脚 P1B 和 P1D 为 0 01 = 驱动引脚 P1B 和 P1D 为 1 1x = P1B和P1D引脚为三态		ECCPAS[1:0]		RW-00
AUX1EN	PWM辅助功能使能位	1 = 使能 0 = 禁止	PWM1AUX[7]	0x90	RW-0
P1OS [4]	PWM单脉冲输出使能位 1 = 使能 (输出后自动停止，P1x变为普通IO) 0 = 关闭 (PWM连续输出)		PWM1AUX[6]		RW-0
P1FOE 4 [5]	P1F 使能位	1 = 使能 (PWM输出) 0 = 禁止 (I/O)	PWM1AUX[5]		RW-0
P1EOE 4	P1E 使能位		PWM1AUX[4]		RW-0
P1DOE 4	P1D 使能位		PWM1AUX[3]		RW-0
P1COE 4	P1C 使能位		PWM1AUX[2]		RW-0
P1BOE 4	P1B 使能位		PWM1AUX[1]		RW-0
P1AOE 4	P1A 使能位		PWM1AUX[0]		RW-0

表 1-3 ECCP 相关用户控制寄存器

[4] 当 AUX1EN=1 且 ECCP 处于半桥 PWM 模式时，此位有效
[5] 当 PWM 工作在半桥模式且 AUX1EN 和 P1OS 同时为 1 时，此位将在下一个 PWM 周期到来后自动清 0。

名称	状态		寄存器	地址	复位值
GIE	全局中断	1 = 使能 (T0IE 适用) 0 = 全局关闭 (唤醒不受影响)	INTCON[7]	0x0B 0x8B 0x10B	RW-0
PEIE	外设总中断	1 = 使能 (TMR1IE 适用) 0 = 关闭 (无唤醒)	INTCON[6]		RW-0
CCP1IE	CCP1溢出 中断控制位	1 = 使能 0 = 关闭 (无唤醒)	PIE2[0]	0x8D	RW-0
CCP1IF	CCP1溢出 中断标志位	1 = 已经溢出 (锁存) 0 = 未溢出	PIR2[0]	0x0D	RW-0

表 1-4 ECCP 中断使能和状态位

1.2. 捕捉模式

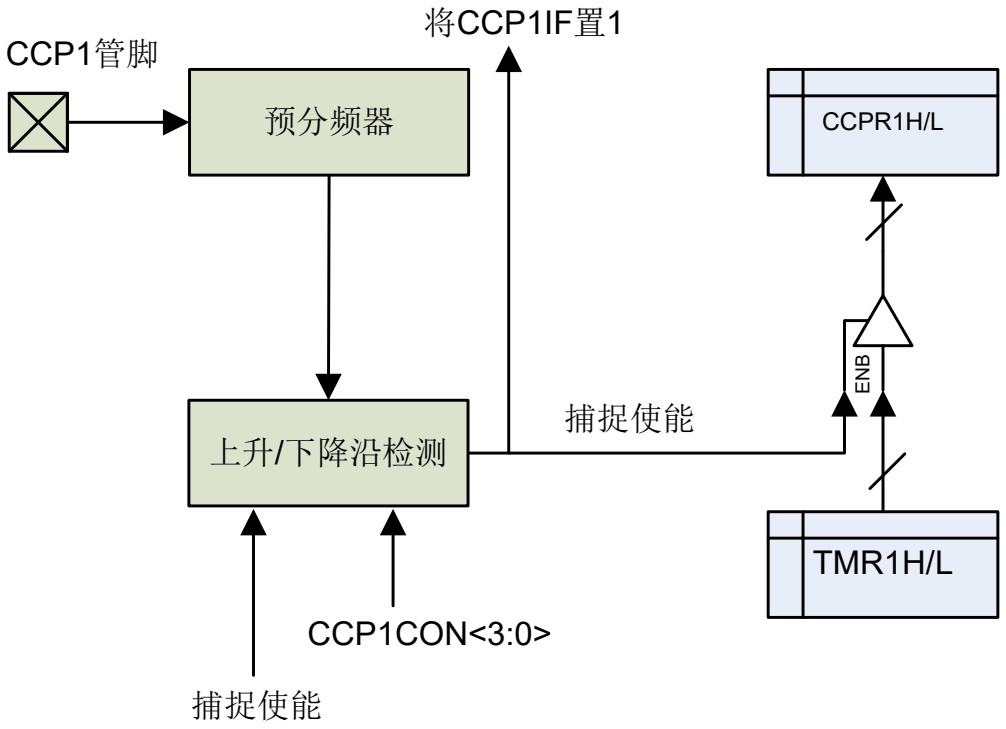


图 1-1 捕捉模式结构框图

在捕捉模式下，当在 CCP1 引脚上发生某一事件时，CCPR1H:CCPR1L 捕捉 TMR1 寄存器中的 16 位值。事件定义为以下之一，并由 CCP1CON 寄存器的 CCP1M<3:0>位进行配置：

- 每个下降沿
- 每个上升沿
- 每 4 个上升沿
- 每 16 个上升沿

进行捕捉后，PIR2 寄存器中的中断请求标志位 CCP1IF 被置 1，该位必须用软件清零。如果在 CCPR1H 和 CCPR1L 这对寄存器中的值被读出之前又发生另一次捕捉，那么原来的捕捉值会被新捕捉值覆盖 (见 **图 1-1**)。

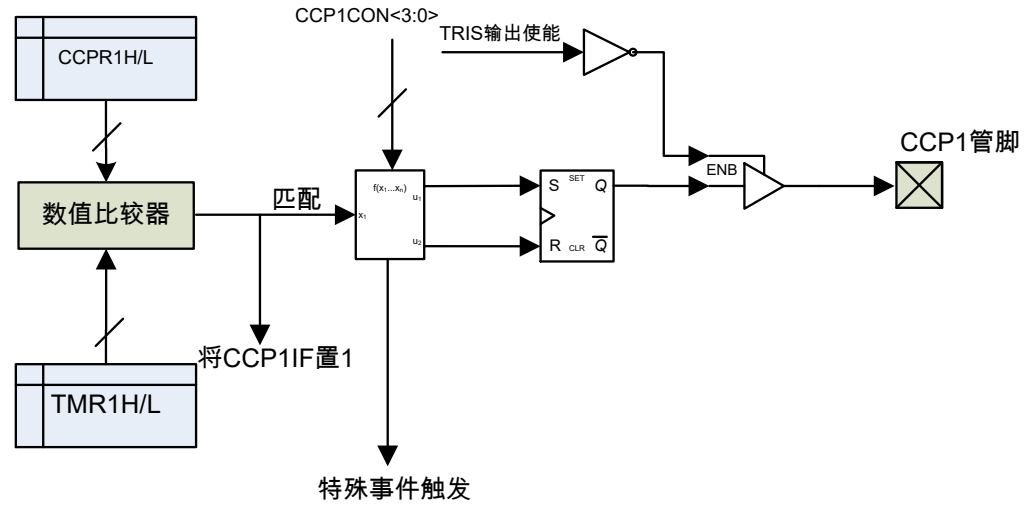
捕捉模式特性

1. 在捕捉模式下，将 CCP1 引脚(PC5)配置为输入(TRISC5 置 1)。如果 CCP1 引脚配置为输出，则写端口将产生一次捕捉条件。
2. 为使 CCP1 模块使用捕捉特性，Timer1 必须运行在定时器模式或同步计数器模式。在异步计数器模式下，捕捉操作可能无法进行。
3. 当捕捉模式改变时，可能会产生一次误捕捉中断。用户应该保持 PIE2 寄存器的 CCP1IE 位清零以避免误中断。此外，用户还应在任何这种工作模式改变之后清零 PIR2 寄存器的中断标志位 CCP1IF。
4. CCP1CON 寄存器的 CCP1M<3:0>位指定了 4 种不同的预分频比。当产生任何复位、关闭 CCP1 模块或 CCP1 模块不在捕捉模式时，预分频计数器会被清零。任何复位都会将预分频计数器清零。

   捕捉模式下，预分频比的切换不会清零预分频器，但可能产生一次中断。在改变预分频比之前可将 CCP1CON 寄存器清零（关闭 CCP1 模块），以避免意外操作。

```
BANKSEL   CCP1CON        ;Set Bank bits to point to CCP1CON
CLRR      CCP1CON        ;Turn CCP module off
LDWI      NEW_CAPT_PS    ;Load the W reg with the new prescaler
STR       CCP1CON        ;Load CCP1CON with this value
```

1.3. 比较模式



特殊事件触发包括：
1. 清零TMR1H和TMR1L寄存器；
2. 不会将PIR1寄存器相关标志位TMR1IF置1；
3. 将GO/DONE位置1启动ADC转换；
4. 发生系统复位，CCP1将被清0。

图 1-2 比较模式结构框图

在比较模式下，16 位 CCPR1 寄存器 (CCPR1H/1L) 值将持续与 TMR1 寄存器 (TMR1H/1L) 的值相比较。发生匹配时，CCP1 模块将可能产生以下事件：

- 翻转 CCP1 输出
- 触发特殊事件
- 产生软件中断

CCP1 引脚 (PC5) 上的动作取决于 CCP1CON 寄存器的 CCP1M<3:0>控制位的值。所有比较模式均可产生中断。

比较模式特性

1. 必须将 CCP1 引脚 (PC5)配置为输出 (TRISC5 = 0)。
2. 在比较模式下，Timer1 必须运行在定时器模式下或同步计数器模式下。异步计数器模式下可能工作不正常
3. 当选择产生软件中断模式 (CCP1M<3:0> = 1010) 时，CCP1 模块对 CCP1 引脚没有控制权(见 CCP1CON 寄存器) 。

1.3.1. 特殊事件触发器

当选定了特殊事件触发模式 (CCP1M<3:0> = 1011) 时，CCP 模块可能触发以下事件。在此模式下，CCP 模块对 CCP1 引脚没有控制权 (见 CCP1CON 寄存器)。

- 复位 Timer1
- 若 ADC 使能，则启动一次 ADC 转换

当 TMR1H:TMR1L 寄存器和 CCPR1H:CPR1L 寄存器之间发生匹配时，将会发生 CCP 的特殊事件触发输出。TMR1H:TMR1L 寄存器在 Timer1 时钟的下一个上升沿到来之前不会复位，因此 CCPR1H:CCPR1L 寄存器可作为 Timer1 的 16 位可编程周期寄存器。

注意：

- CCP 模块的特殊事件触发不会将中断标志位 TMR1IF 置 1 (PIR1 寄存器)，不会产生 Timer1 中断。ECCP 模块仍可配置为产生 ECCP 中断
- 在生成特殊事件触发的时钟边沿与复位 Timer1 的时钟边沿之间，更改 CCPR1H 和 CCPR1L 寄存器的内容将清除匹配条件，此操作可以预防 TMR1H/L 计数器复位 (清 0)。
- 当 ECCP 配置为触发特殊事件时，触发器会将 TMR1H:TMR1L 这对寄存器清零
- Timer1 应同步为 FOSC 以充分利用特殊事件触发器
- Timer1 异步工作可导致错过特殊事件触发器
- 当对 TMR1H 或 TMR1L 的写操作与一个 ECCP 特殊事件触发器同时发生时，写操作具有优先权

1.4. PWM 模式

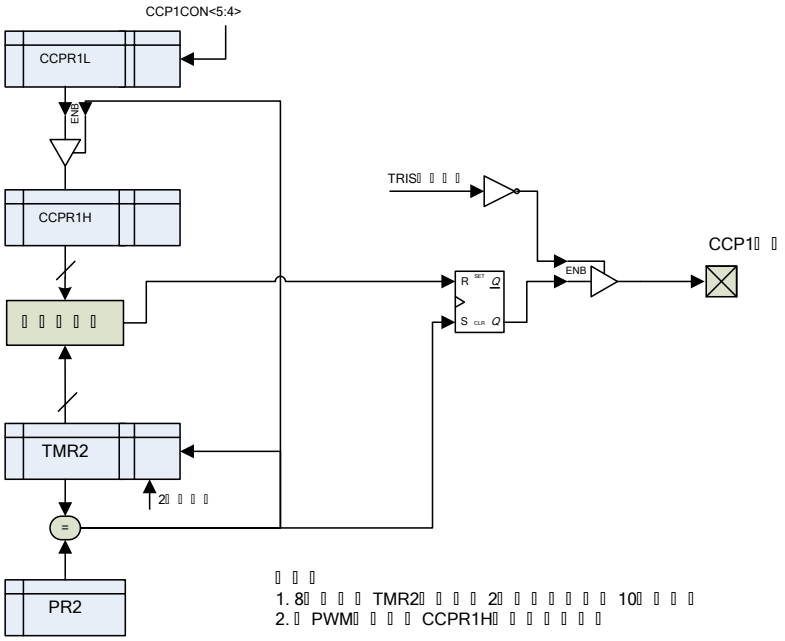


图 1-3 PWM 结构框图

PWM 模式特性：

1. CCP1 引脚将产生脉宽调制信号。其占空比，周期和分辨率由 PR2、T2CON、CCPR1L、CCP1CON 寄存器决定
2. PWM 的输出高达 10 位分辨率
3. 由于 CCP1 引脚与 PORT 数据锁存器复用，必须将 CCP1 引脚配置为输出 (TRISC5 = 0)
4. 将 CCP1CON 寄存器清零将放弃对 CCP1 引脚的控制
5. 由于慢时钟测量也占用 TIMER2，所以使用 PWM 功能时不能启动慢时钟测量

1.4.1. PWM 周期

PWM 周期由 Timer2 的 PR2 寄存器决定。用**公式 1.1** 可计算 PWM 周期。

公式 1.1 *PWM 周期 = (PR2 + 1) * 4 * Tsys * (TMR2 预分频值)*

当 TMR2 等于 PR2 时，下一次递增周期将发生以下三个事件：

- TMR2 被清零
- CCP1 引脚被置 1 (例外：若 PWM 占空比= 0%，引脚不被置 1)
- PWM 占空比从 CCPR1L 锁存到 CCPR1H

注意：

1. PWM 模式下，Timer2 的时钟源为系统时钟 Fosc，而不是指令时钟；
2. PWM 频率不使用 Timer2 的后分频器。

3. 当 MSCKCON.5 为 1 时，Timer2 的时钟源为 32MHz。

1.4.2. PWM 占空比

- CCPR1L 寄存器为高 8 位,CCP1CON 寄存器的 DC1B<1:0>为低 2 位,组成 10 位值决定 PWM 占空比。
- CCP1CON 寄存器的 DC1B<1:0>和 CCPR1L 可在任何时候被写入。
- 占空比直到周期完成时 (即 PR2 和 TMR2 寄存器发生匹配时) 才被锁存到 CCPR1H 中。
- 使用 PWM 时，CCPR1H 寄存器只读。

公式 10.2 *脉冲宽度 = (CCPR1L : CCP1CON <5:4>) * Tsys * (TMR2 预分频值)*

公式 10.3 *占空比 = (CCPR1L : CCP1CON <5:4>) ÷ (4 * (PR2+1))*

- CCPR1H 寄存器和 2 位的内部锁存器用于为 PWM 占空比提供双缓冲。双缓冲对 PWM 的无毛刺工作起着非常重要的作用。
- 8 位定时器 TMR2 寄存器与 2 位的内部系统时钟 (FOSC) 或 2 位的预分频器连接，组成 10 位时基。Timer2 预分频器置为 1:1 时，则使用系统时钟。
- 当 10 位时基与 CCPR1H 及 2 位的锁存器匹配时，CCP1 引脚被清零

1.4.3. PWM 分辨率

分辨率决定周期的有效占空比。PR2 为 255 时产生 10 位的最大 PWM 分辩率。

公式 1.4 *分辨率 = log [4 (PR2 + 1)] ÷ log (2) 位*

注意：如果脉冲宽度大于周期，则分配的 PWM 引脚将保持不变。

Timer2 预分频比	PR2	PWM 频率 (kHz)	最大分辨率
16	204	1.22	9.7
4	204	4.88	9.7
1	204	19.53	9.7
1	50	78.12	7.7
1	25	156.3	6.7
1	18	263.1	6.3

表 1-5 PWM 频率和分辨率示例 (Fosc=20MHz)

1.4.4. PWM 配置

按照以下步骤将 CCP 模块配置为 PWM 工作：

1. 将相关的 TRIS 位置 1 禁止 PWM 引脚 (CCP1) 的输出驱动器；
2. 装载 PR2 寄存器以设置 PWM 周期；
3. 用适当的值装载 CCP1CON 寄存器将 CCP 模块配置为 PWM 模式；
4. 装载 CCPR1L 寄存器和 CCP1CON 寄存器的 DC1B<1:0>设置 PWM 占空比；

5. 配置并启动 Timer2：
   - 将 PIR1 寄存器的 TMR2IF 中断标志位清零
   - 装载 T2CON 寄存器的 T2CKPS 位设置 Timer2 预分频比
   - 将 T2CON 寄存器的 TMR2ON 位置 1 使能 Timer2
   - 如果要设置高速模式，则需要把 MSCKCON.5 置 1
6. 重新开始一个 PWM 周期后，使能 PWM 输出：
   - 等待 Timer2 溢出 (PIR1 寄存器的 TMR2IF 位置 1)
   - 将相关的 TRIS 位清零使能 CCP1 引脚的输出驱动器

注：

1. 在休眠模式下，TMR2 寄存器不递增，模块的状态不变。如果 CCP1 引脚正在驱动一个值，它将继续驱动该值。器件唤醒时，TMR2 将继续先前的状态。
2. PWM 频率来自系统时钟频率，系统时钟频率的任何改变将导致 PWM 频率的改变。
3. 当 TIMER2 的时钟源选择内部 32MHz 时，系统时钟频率的改变不会影响 PWM 周期。
4. 任何复位均将强制所有端口为输入模式，并强制 CCP 寄存器为其复位状态。

1.5. PWM 增强模式

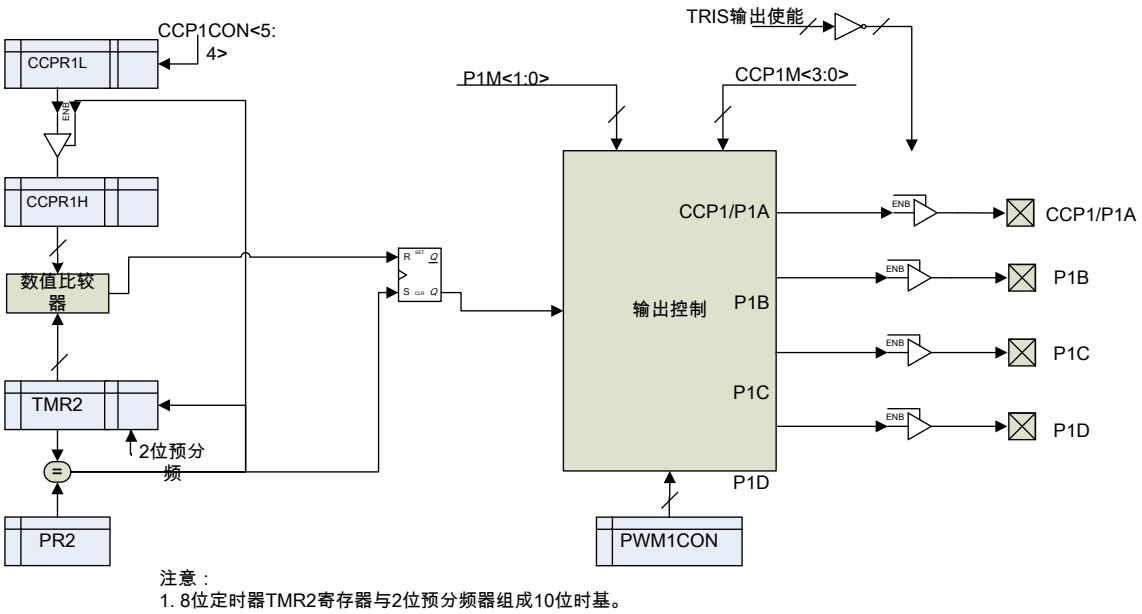


图 1-4 增强型 PWM 结构框图

增强型 PWM 模式最多可在四个输出引脚上产生高达 10 位分辨率的 PWM 信号。

四种 PWM 输出模式：

- 单 PWM
- 半桥 PWM
- 全桥 PWM，正向模式
- 全桥 PWM，反向模式

要选择增强型 PWM 模式，CCP1CON 寄存器的 P1M 位必须被正确设置。

PWM 输出与 I/O 引脚复用，并被指定为 P1A、P1B、P1C 和 P1D。PWM 引脚的极性可配置，可通过将 CCP1CON 寄存器中的 CCP1M 位进行适当置 1 选择极性。

注意：

1. 必须正确配置每个 PWM 输出的 TRIS 寄存器值；
2. 清零 CCP1CON 寄存器将放弃所有 PWM 输出引脚的 ECCP 控制权；
3. 增强型 PWM 模式未使用的引脚均可用于其他引脚功能。

ECCP	P1M<1:0>	CCP1/P1A	P1B	P1C	P1D
单 PWM	00	是	否	否	否
单桥 PWM	10	是	是	否	否
全桥，正向	01	是	是	是	是
全桥，反向	11	是	是	是	是

表 1-6 不同 PWM 增强模式的引脚分配示例

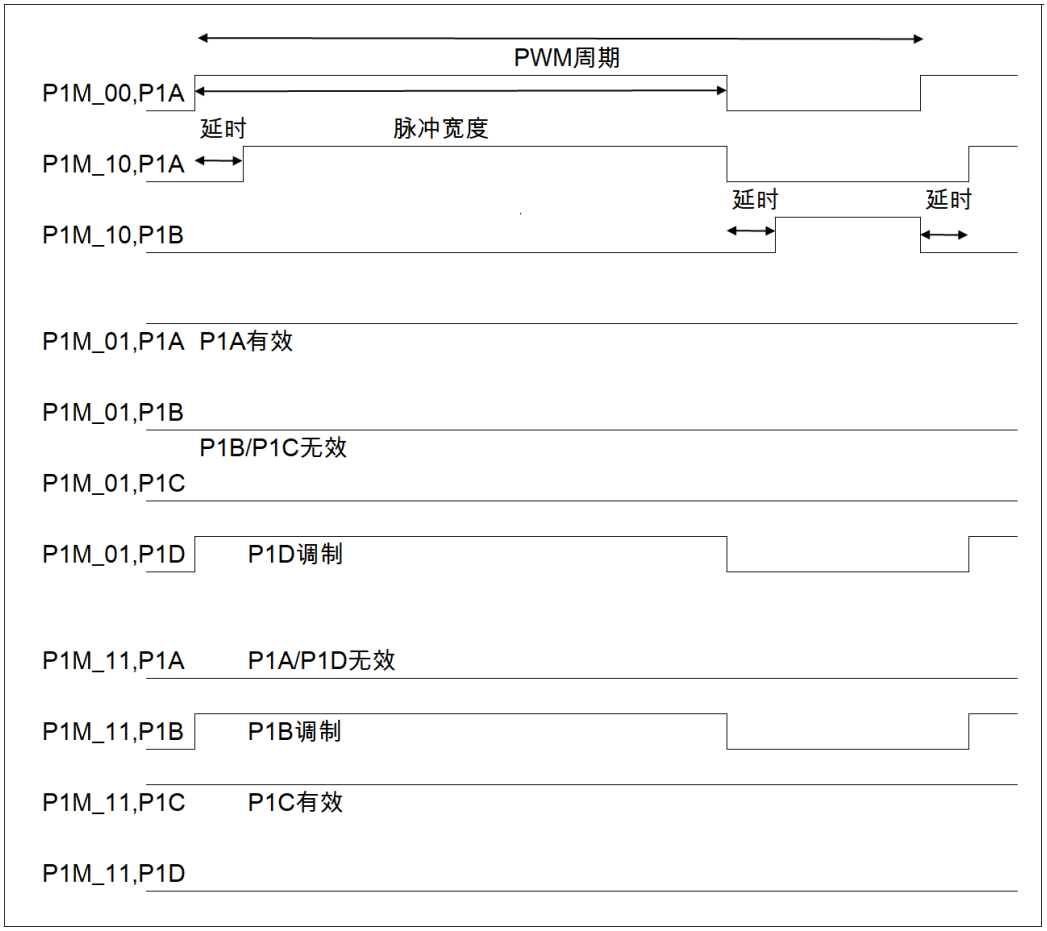


图 1-5 PWM 输出关系示意图 (高电平有效)

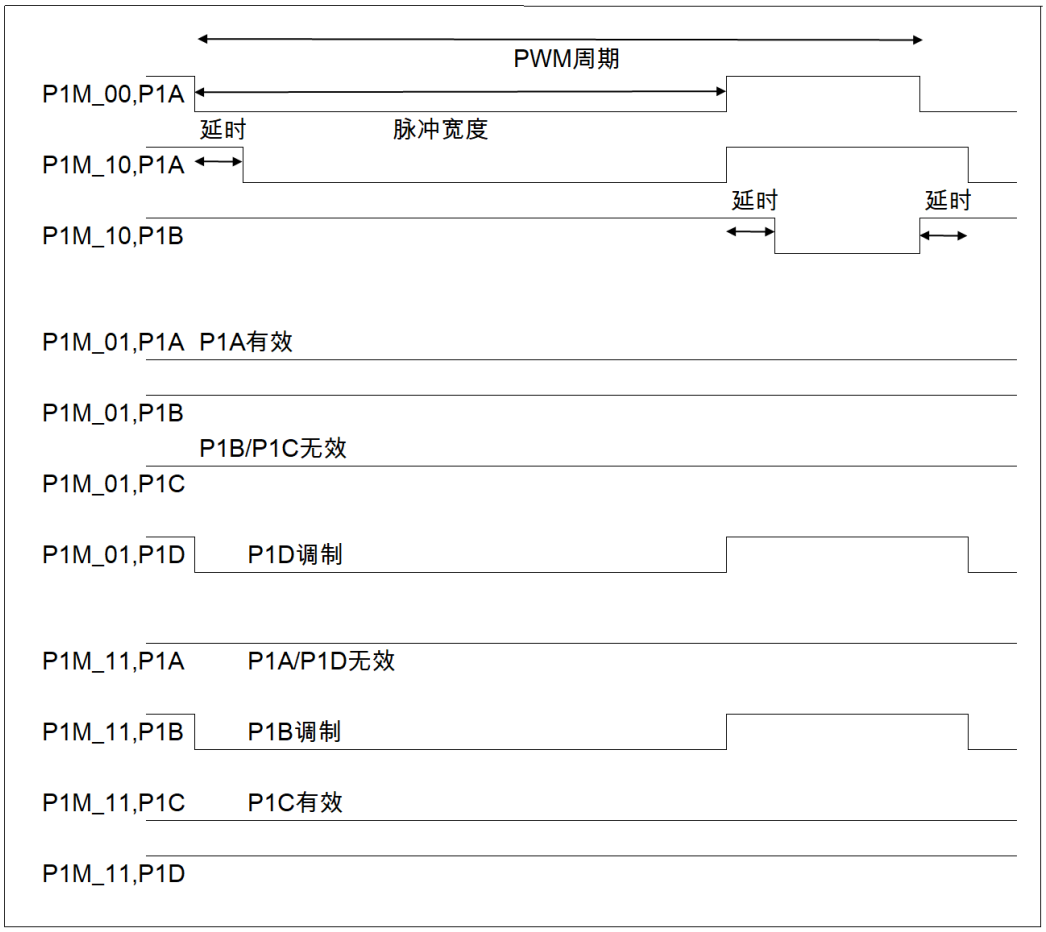


图 1-6 PWM 输出关系示意图 (低电平有效)

1.5.1. 半桥模式

在半桥模式下，有两个引脚用作输出以驱动推挽负载。PWM 输出信号被输出到 CCP1/P1A 引脚，而互补 PWM 输出信号被输出到 P1B 引脚（见**图 1-5**）。此模式可用于半桥和全桥应用，此时两个 PWM 信号调制四个功率开关。

在半桥模式下，可使用编程死区延时防止半桥功率器件中出现穿通电流。PWM1CON 寄存器的 PDC<6:0>位用于设置将输出驱动为有效前的指令数。如果该值大于占空比，则相应的输出在整个周期中将保持无效状态。死区延时操作的详情请参见第 1.5.7 节“可编程死区延时模式”。

由于 P1A 和 P1B 输出与 PORT 数据锁存器复用，必须清零相关的 TRIS 位以将 P1A 和 P1B 配置为输出。

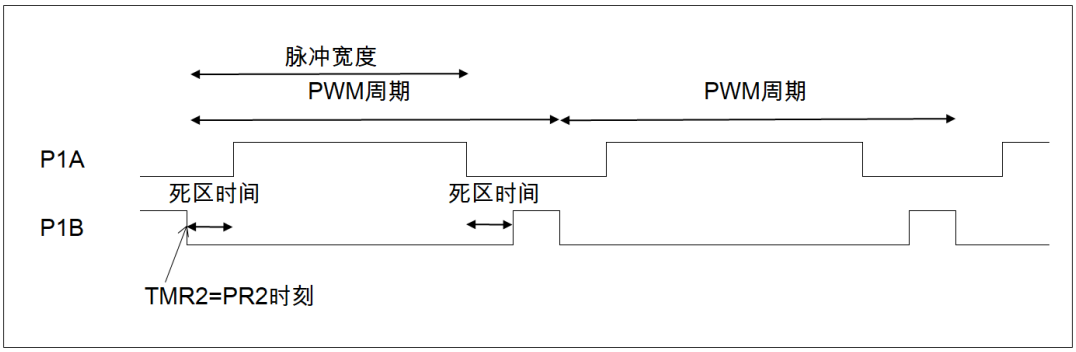


图 1-7 半桥 PWM 输出

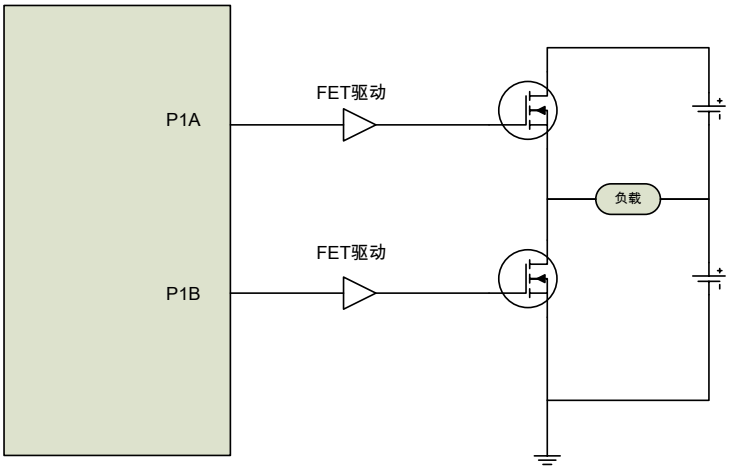


图 1-8 标准半桥电路 (推挽)

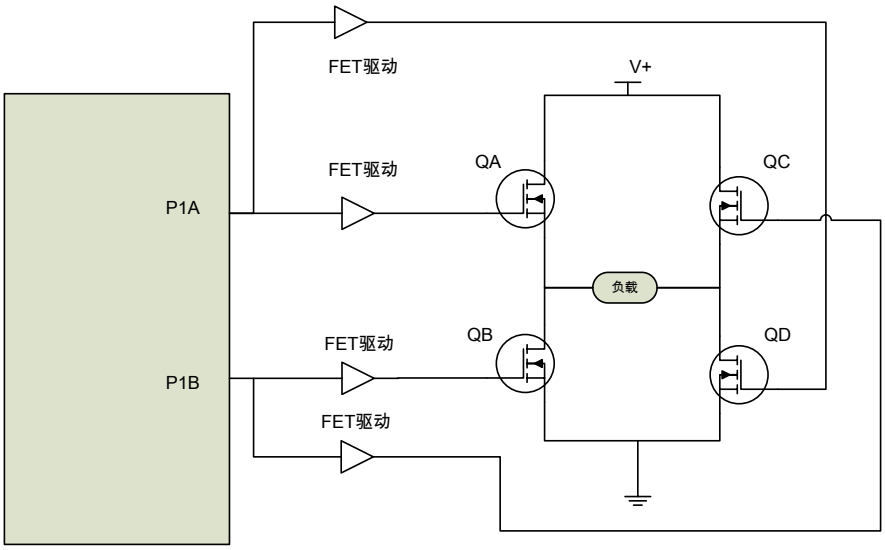


图 1-9 半桥输出驱动全桥电路 (4NMOS)

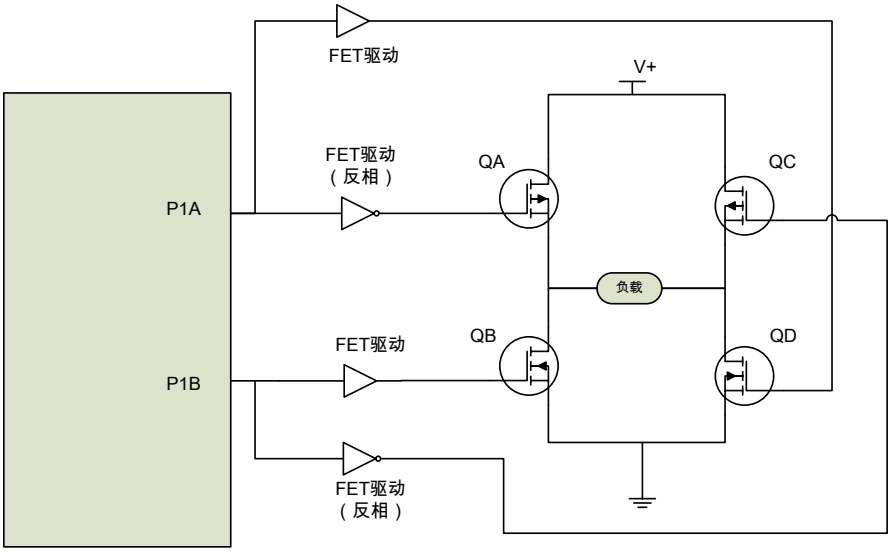


图 1-10 半桥输出驱动全桥电路 (2PMOS+2NMOS)

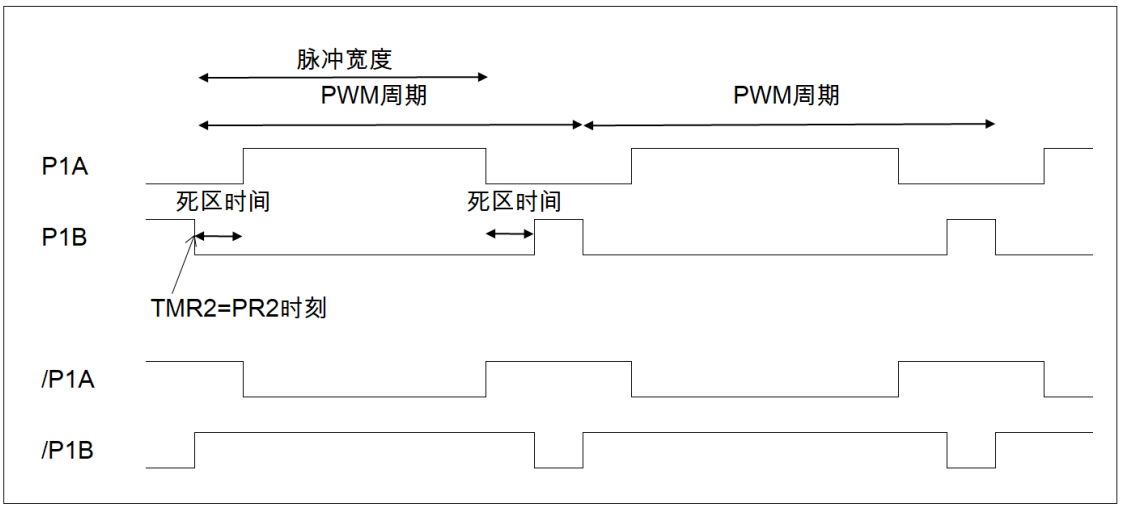


图 1-11 半桥 PWM 输出 (需要外接反相器)

1.5.2. 全桥模式

由于 P1A、P1B、P1C 和 P1D 输出与 PORT 数据锁存器复用。在全桥模式下，必须清零相关 TRIS 位将 P1A、P1B、P1C 和 P1D 引脚配置为输出。

1.5.2.1. 全桥模式 PWM 输出示例

当 CCP1CON 的 CCP1M 设置为 1100 时(即 P1A~P1D 都为高有效)，使用 4 个 NMOS 做全桥应用示例如下：

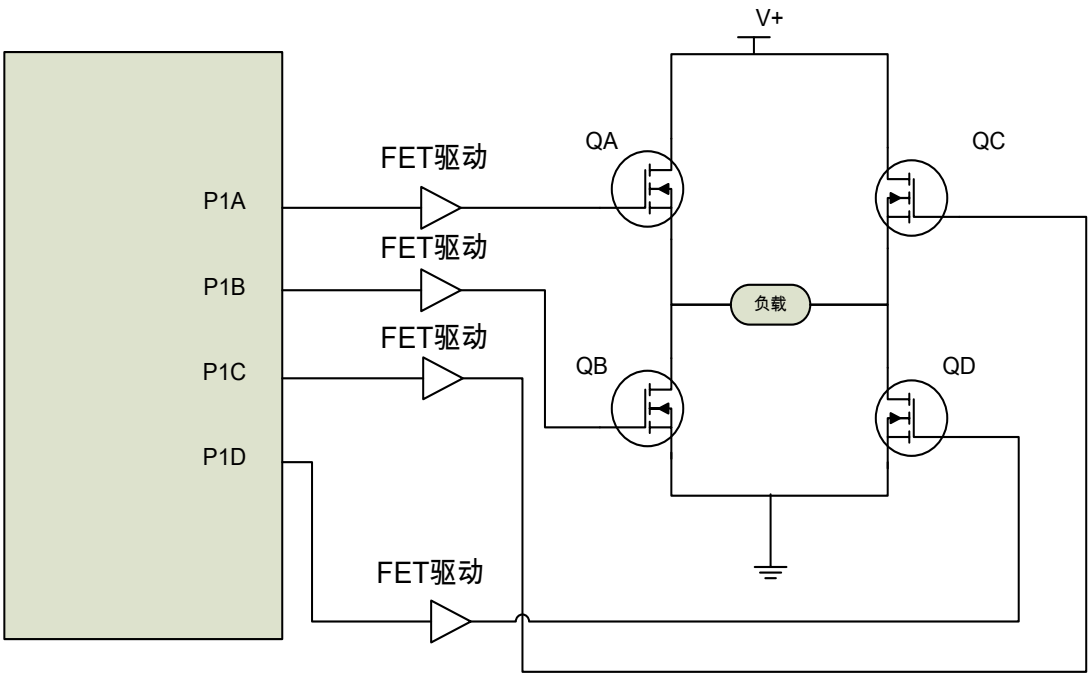


图 1-12 全桥应用示例 (4 个 NMOS)

- 在正向模式下，CCP1/P1A 引脚驱动为有效状态，P1D 引脚为调制输出，而 P1B 和 P1C 为无效状态，PWM 输出波形如下图所示：

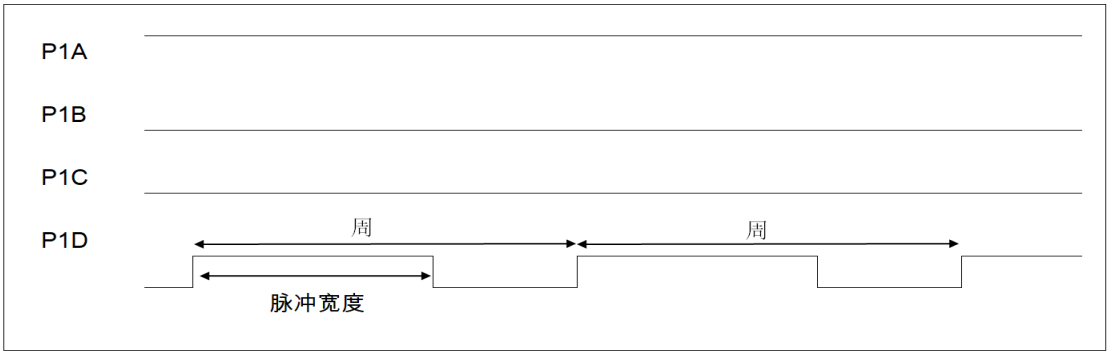


图 1-13 全桥 PWM 输出波形 a

- 在反向模式下，P1C 驱动为有效状态，P1B 引脚为调制输出，而 P1A 和 P1D 则为无效状态，PWM 输出波形如下图所示：

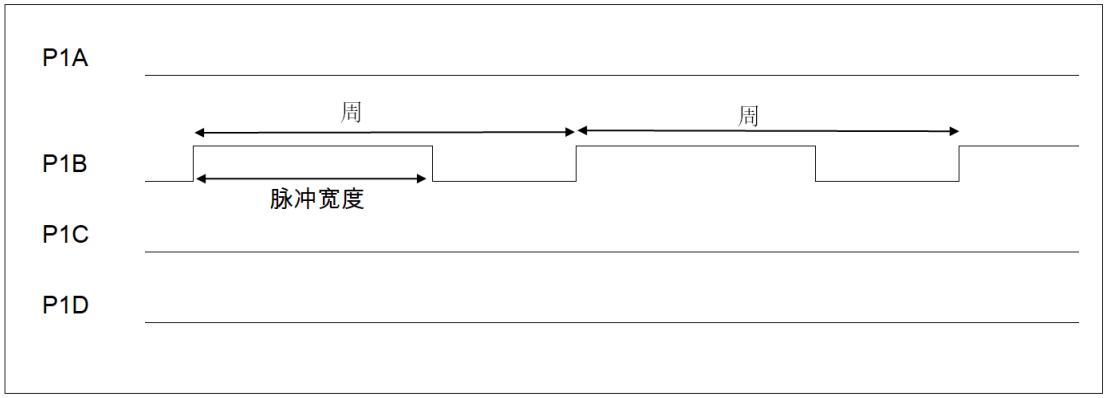


图 1-14 全桥 PWM 输出波形 b

当 CCP1CON 寄存器的 CCP1M 设置为 1110 时 (即 P1A 和 P1C 为低电平有效，P1B 和 P1D 为高电平有效)。使用 2 个 PMOS 和 2 个 NMOS 做全桥应用示例如下：

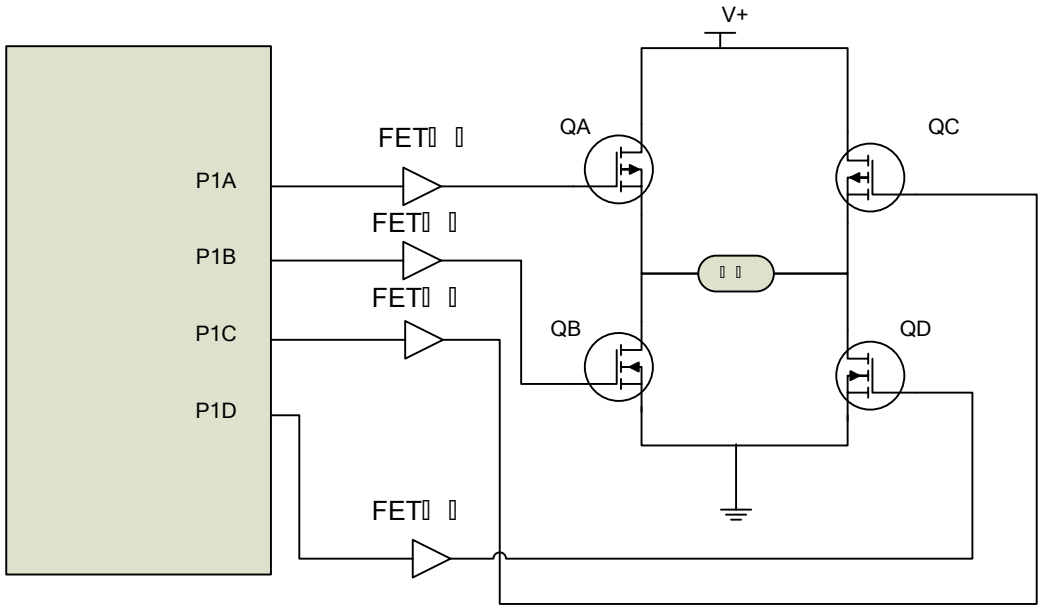


图 1-15 全桥应用示例 (2 个 PMOS 和 2 个 NMOS)

此应用下的 PWM 输出波形如下：

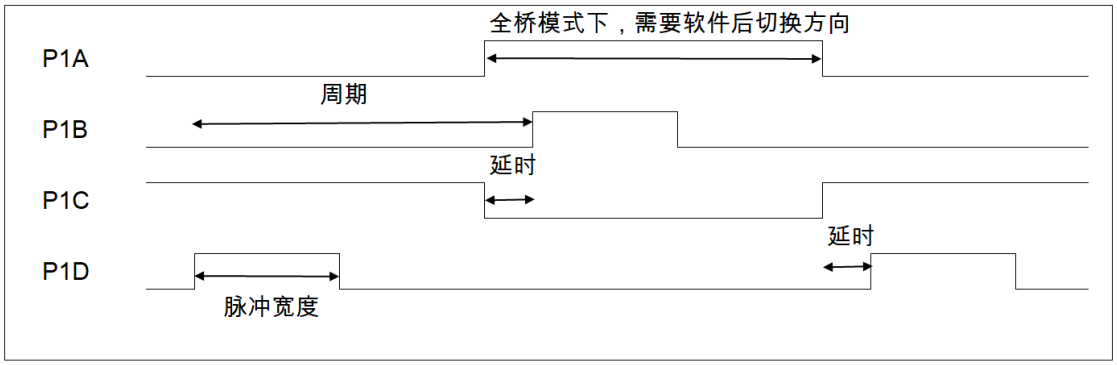


图 1-16 全桥 PWM 输出波形 c

1.5.2.2. 全桥模式下改变 PWM 周期方向

在全桥模式下，CCP1CON 寄存器的 P1M1 位可控制 PWM 周期正/反方向。当软件改变方向控制位时，模块将在下一个 PWM 周期改变方向。

用软件改变 CCP1CON 寄存器的 P1M1 位可启动方向改变。在当前 PWM 周期前的四个 Timer2 周期，发生以下时序(见**图 1-17**)：

- 调制输出 (P1B 和 P1D) 被置于无效状态
- 相关的未调制输出 (P1A 和 P1C) 切换为相反方向驱动
- 在下一个周期恢复 PWM 调制

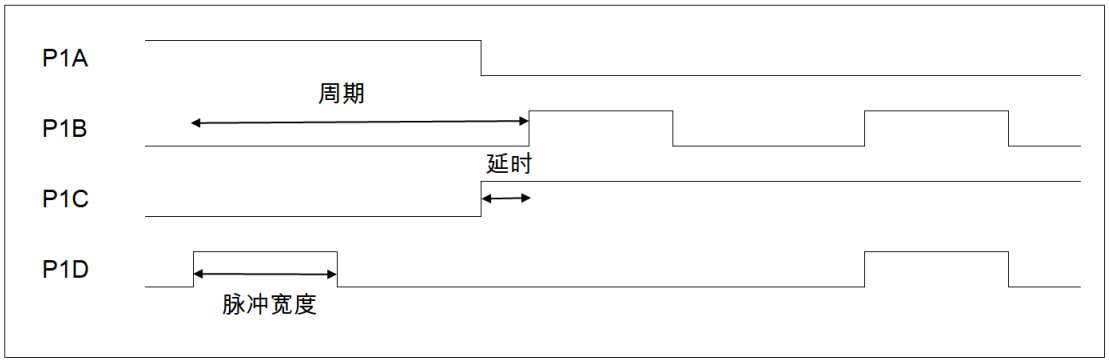


图 1-17 PWM 方向改变示例

注意：

1. CCP1CON 寄存器的方向位 P1M1 可在 PWM 周期的任何时刻被写入；
2. 改变方向时，P1A 和 P1C 信号在当前 PWM 周期结束前切换。此时 P1B 和 P1D 调制信号无效，时长为 4 次 Timer2 计数。

全桥模式不提供死区延时。在调制一个输出时，一般不需要死区延时。但有一种情况需要死区延时，当以下两个条件同时成立时即发生需要死区延时的情况：

1. 输出占空比达到或接近 100%时 PWM 输出方向改变；
2. 功率开关 (包括功率器件和驱动器电路) 的关断时间大于导通时间。

图 1-18 所示为占空比接近 100%时，PWM 方向从正向变为反向的示例。此示例中，在时间 t1 处，P1A 和 P1D 输出变为无效，而 P1C 输出变为有效。由于功率器件的关断时间大于导通时间，穿通电流将流过功率器件 QC 和 QD (见图 1-12) 并持续时间“T”。当 PWM 方向由反向变为正向时，同样的情况将发生在功率器件 QA 和 QB 上。

如果某个应用要求在占空比很高时改变 PWM 方向，以下提供了两种消除穿通电流的方法：

1. 改变方向前将减小 PWM 占空比；
2. 使用能使开关的关断时间快于导通时间的开关驱动器。

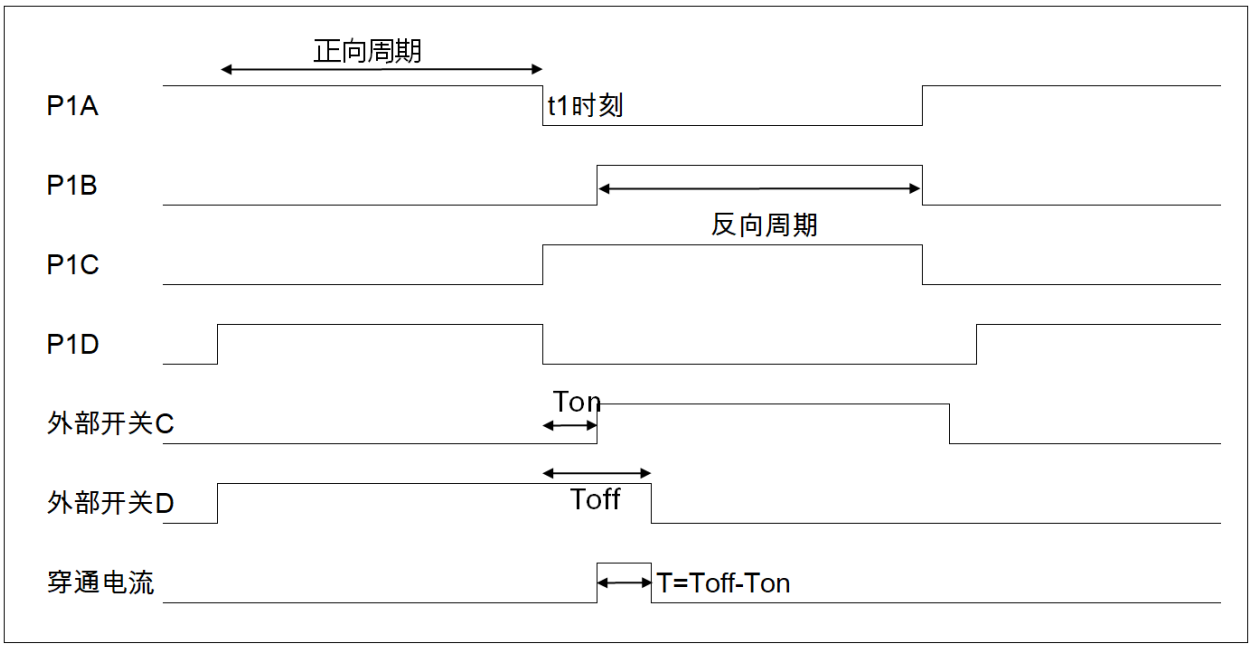


图 1-18 占空比接近 100%时 PWM 方向变化示例

注意：

1. 上图所有信号均为高电平有效；
2. Ton 为功率开关 QC 及其驱动器的导通延时；
3. Toff 为功率开关 QD 及其驱动器的判断延时。

1.5.3. 启动考虑事项

- 使用任何 PWM 模式时，硬件必须在 PWM 输出引脚上使用适当的外部上拉/下拉电阻。
- 单片机从复位退出时，所有 I/O 引脚均为高阻态。在单片机以正确的信号电平驱动 I/O 引脚或激活 PWM 输出前，外部电路必须使功率开关置于关断状态。
- CCP1CON 寄存器的 CCP1M<1:0>位可供用户选择每对 PWM 输出引脚（P1A/P1C 和 P1B/P1D）的输出信号为高电平有效还是低电平有效。PWM 输出极性必须在使能 PWM 引脚的输出驱动器前选定。不建议在 PWM 引脚的输出驱动器使能时改变极性配置，因为这可能会损坏应用电路。
- 在 PWM 模块初始化时，P1A、P1B、P1C 和 P1D 输出锁存器可能不在正确的状态下。将 PWM 引脚的输出驱动器与增强型 PWM 模式同时使能可能导致应用电路的损坏。增强型 PWM 模式必须在正确的输出模式下使能，并且在 PWM 引脚的输出驱动器被使能前完成整个 PWM 周期。PWM 周期是否完成可通过查看 PIR1 寄存器的 TMR2IF 位在第二个 PWM 周期开始时是否置 1。

1.5.4. 关闭 PWM 输出

如果要停止 PWM 输出，建议通过对写 ECCPAS 寄存器相关值，或者把相关 IO 的输出驱动关闭 (TRISC.x 置 1) ，通过外部的上下拉电阻 IO 处于确定状态，而不是简单把 CCP1CON 的 CCP1M 清 0，因为此时 IO 还处于输出状态，其输出值不确定。

下面是关闭和重启 PWM 的程序示例：

```
;关闭 PWM 输出==============
BANKSEL    ECCPAS
LDWI       0F0H
IORWR      ECCPAS, F

;重开 PWM 输出==============
BANKSEL    ECCPAS
LDWI       00FH
ANDWR      ECCPAS, F
```

1.5.5. 增强型 PWM 自动关闭模式

PWM 模块支持自动关闭模式，它会在发生外部关闭事件时禁止 PWM 输出。自动关闭模式会将 PWM 输出引脚置于预定状态。该模块用于防止 PWM 损坏应用。

使用 ECCPAS 寄存器的 ECCPASx 位可选择自动关闭源。关闭事件可由以下产生：

- INT 引脚出现逻辑 0
- 比较器 C1
- 比较器 C2
- 软件写 ECCPASE = 1

关闭状态由 ECCPAS 寄存器的 ECCPASE (自动关闭事件状态) 位标示。若 ECCPASE = 0，PWM 引脚正常工作，若 ECCPASE = 1，PWM 输出关闭。

发生关闭事件时，将出现两个状况：

1. ECCPASE 保持置 1 状态直到被固件清零或发生了自动重启 (见**第 1.5.6 节** “自动重启模式”) ；
2. 使能的 PWM 引脚被异步置于关闭状态。PWM 输出引脚被分为两对[P1A/P1C]和[P1B/P1D]。引脚状态由 PSSAC 和 PSSBD 位决定。每对引脚均可置于以下三种状态之一：

   - 驱动为逻辑 1
   - 驱动为逻辑 0
   - 三态 (高阻态)

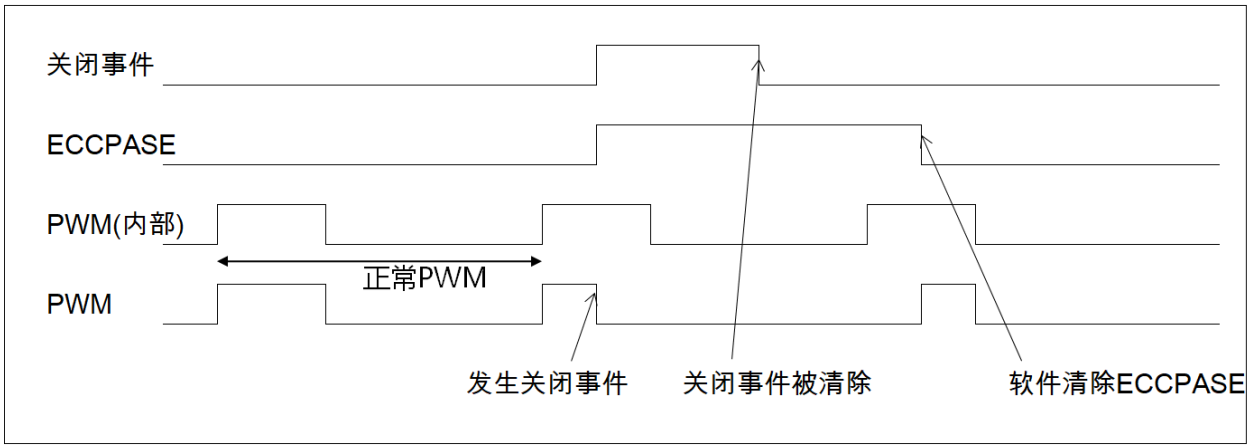


图 1-19 非自动重启时的 PWM 自动关闭

注意：

1. 自动关闭条件是基于电平的信号，而非基于边沿的信号。只要电平不变，自动关闭就不变；
2. 自动关闭条件下禁止写入 ECCPASE 位；
3. 自动关闭条件有效时，PWM 停止输出但 PWM 计数器还在继续运行，所以当自动关闭条件清除时（通过固件或自动重启），PWM 将立即恢复输出。

1.5.6. 增强型 PWM 自动重启模式

增强型 PWM 可配置为在自动关闭条件被清除时自动重启 PWM 信号。将 PWM1CON 寄存器中的 PRSEN 位置 1 可使能自动重启。

使能自动重启时，只要自动关闭条件有效，ECCPASE 位就保持置 1。当自动关闭条件被清除时，ECCPASE 位将被硬件清零，恢复正常工作。

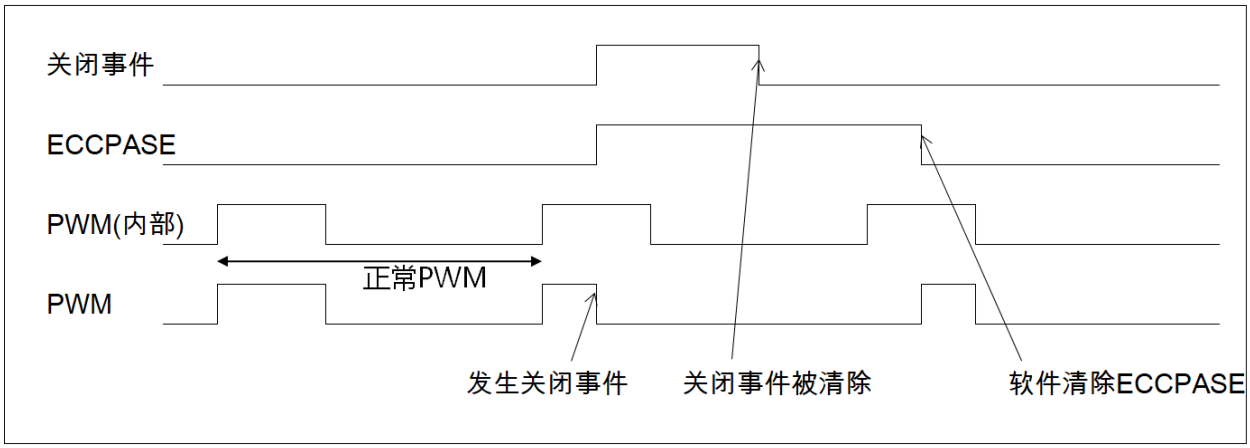


图 1-20 自动重启时的 PWM 自动关闭

1.5.7. 可编程死区延时模式

在所有功率开关均调制为 PWM 频率的半桥应用中，功率开关从关断到导通通常需要较长的时间。如果上下两个功率开关同时动作（一个导通另一个关断），在一个开关完全关断前，两个开关可能在一个很短的时间内同时导通。在这段很短的时间内，在两个功率开关中会流过极高的电流（穿通“shoot-through”

电流)，使桥路的电源短路。为避免在开关时出现这种极具破坏力的穿通电流，通常使任一功率开关的导通时间延后，以使另一个开关有时间完全关断。

在半桥模式下，使用数字可编程死区延时来避免穿通电流破坏桥路的功率开关。信号从无效状态变为有效状态时发生延时，如图 1-21 所示。相关 PWM1CON 寄存器的低 7 位以单片机的指令周期 (2 个系统时钟周期) 为单位设置延时期限。

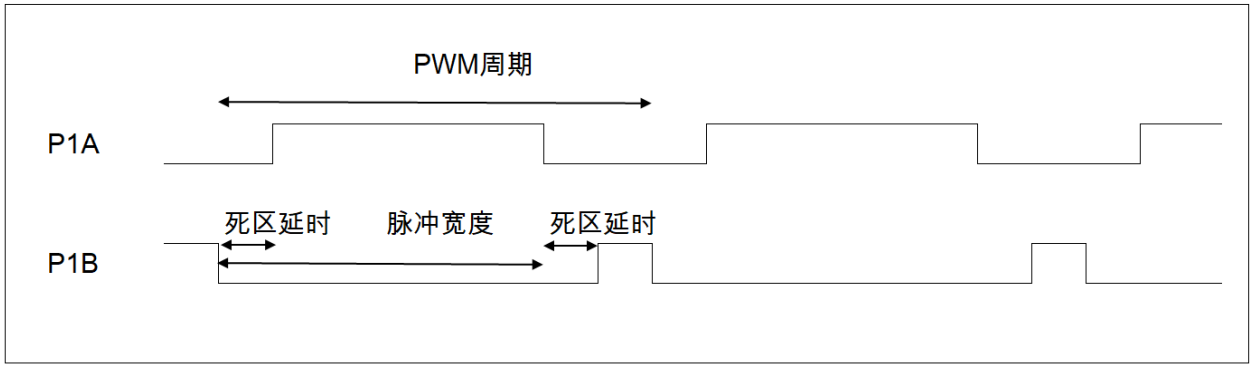


图 1-21 半桥 PWM 输出示例

1.6. PWM 的辅助功能

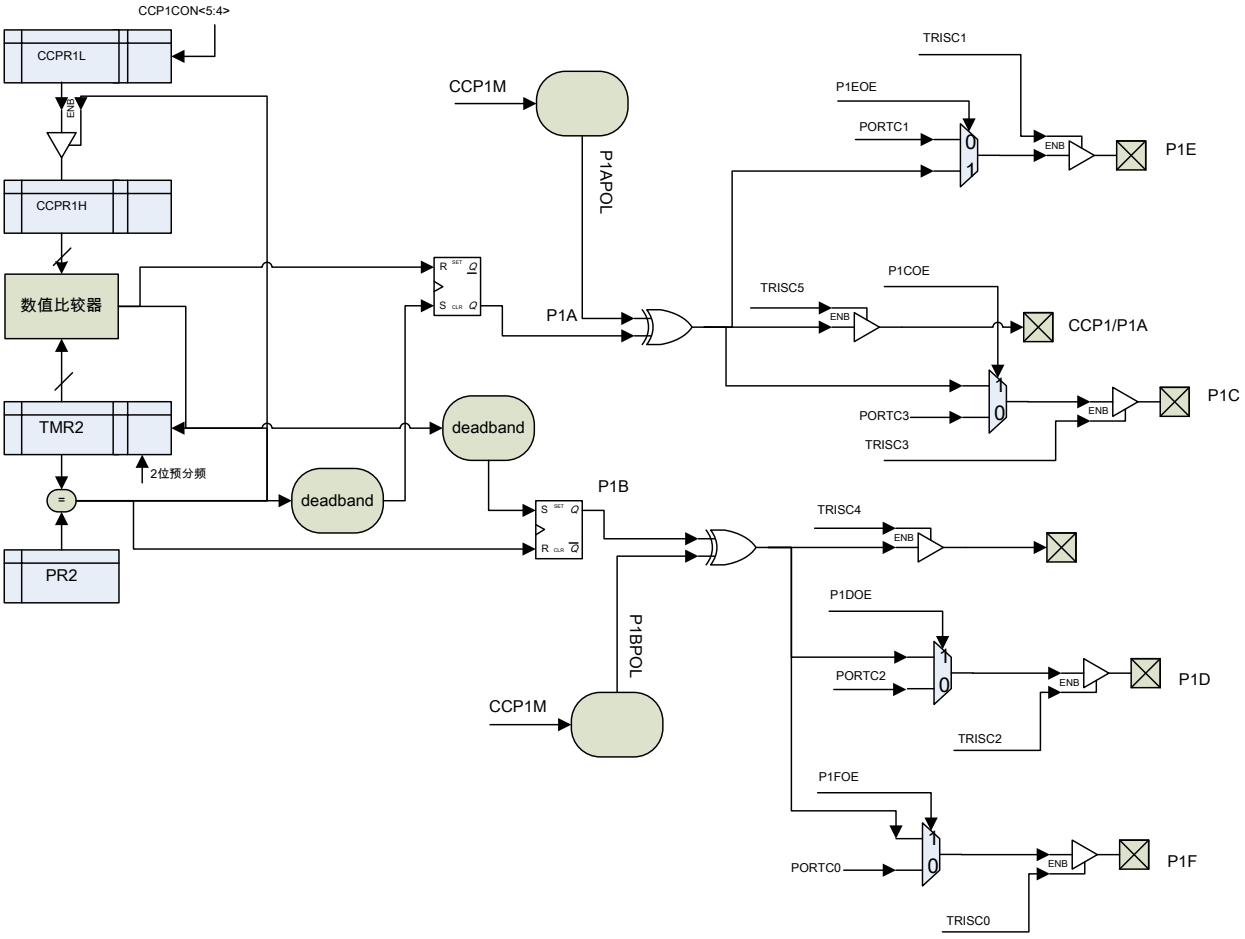


图 1-22 PWM 的辅助功能原理框图

通过适当设置寄存器 PWM1AUX，工作在**半桥模式**下的增强型 PWM 可以做到：

- 输出一次 PWM 信号后自动关闭 PWM 输出
- 最多有 3 对 6 路 PWM 信号同时输出 (当 P1xOE 全部为 1 时，这里 x 是 A~F)
- 输出极性可配置 (通过 CCP1M[3:0])

注意：
PWM 的辅助功能只对半桥模式起作用，其它的单输出或全桥模式不起作用，即 P1M<1:0>=10。

1.6.1. 单次脉冲模式

配置 CCP1CON 使 ECCP 处于 PWM 半桥模式，同时把 PWM1AUX 的 AUX1EN 置 1 和 P1OS 置 1，此时 PWM 为单次脉冲模式。

当下一个周期 PWM 到来时 (TIMER2 等于 PR2+1)，PWM 输出由硬件自动关闭，P1A~P1F 变为通用 IO。

需要注意的是，在该模式下，PWM 输出一次脉冲波形后只是把 P1xOE 关闭，里面的 PWM 计数器将保持计数，如果软件再次把 P1xOE (x 可以是 A~F) 置 1，则在下一个 PWM 周期管脚 P1x (x 可以是 A~F) 会输出一个 PWM 波形，如下图所示：

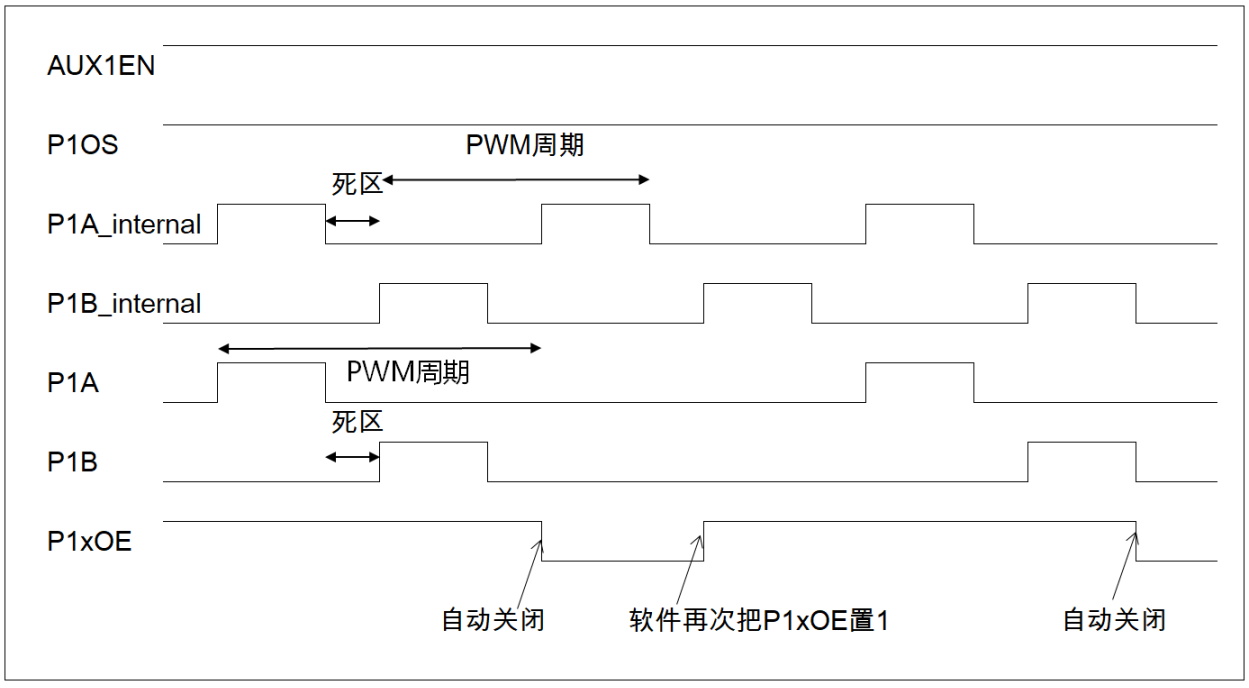


图 1-23 PWM 的辅助功能原理框图

1.6.2. 3 对 PWM 信号输出

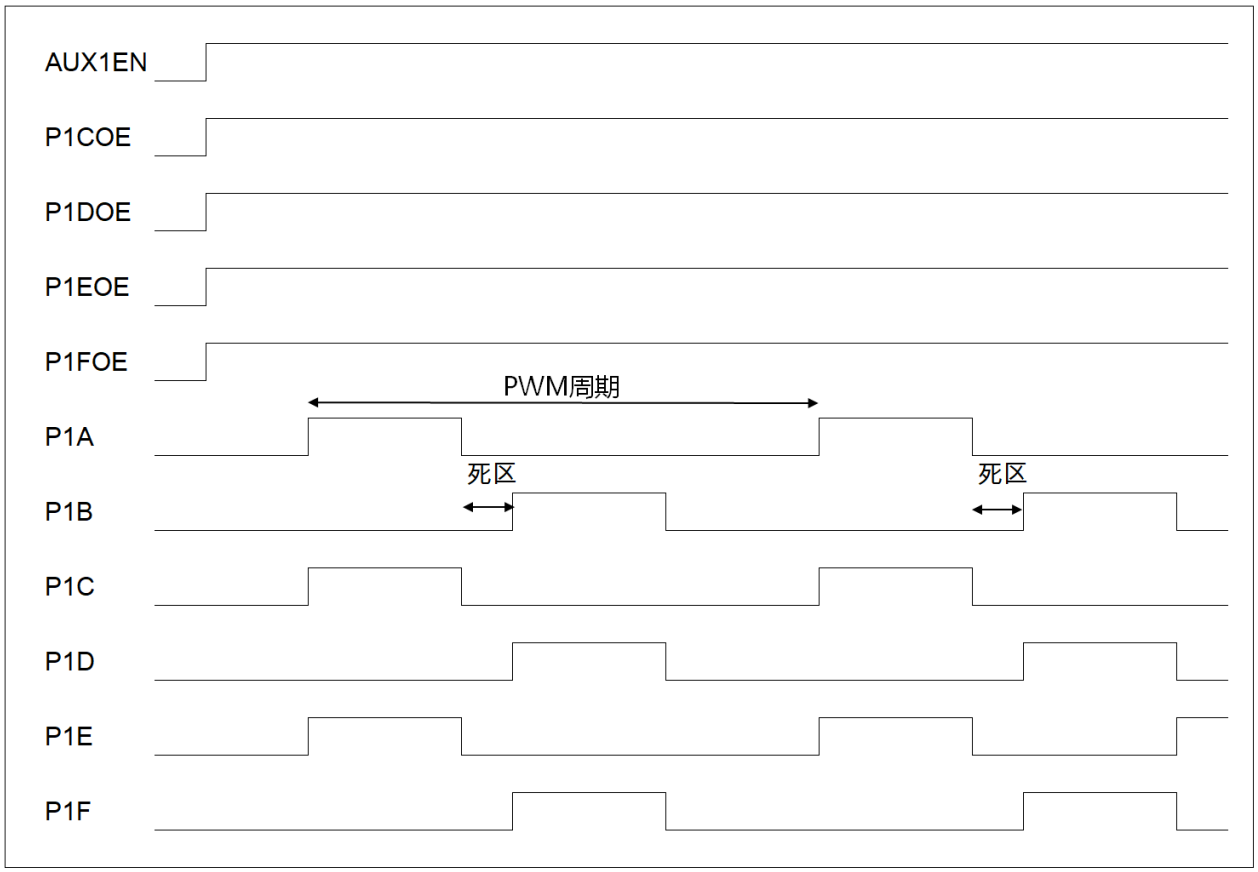


图 1-24 PWM 的辅助功能原理框图

注意：

如上图所示，P1A 和 P1B 是第一组带死区时间的半桥 PWM 输出，P1C 和 P1D，P1E 和 P1F 是第二和第三组，它们的波形和第一组是一样的。

1.6.3. PWM 辅助功能的配置

应按照以下步骤将 CCP 模块配置为 PWM 工作：

1. 将相关的 TRIS 位置 1 禁止 PWM 引脚 (CCP1) 的输出驱动器；
2. 装载 PR2 寄存器以设置 PWM 周期；
3. 用适当的值装载 CCP1CON 寄存器将 CCP 模块配置为 PWM 半桥模式；
4. 装载 CCPR1L 寄存器和 CCP1CON 寄存器的 DC1B<1:0>设置 PWM 占空比；
5. 配置并启动 Timer2：
   - 将 PIR1 寄存器的 TMR2IF 中断标志位清零
   - 装载 T2CON 寄存器的 T2CKPS 位设置 Timer2 预分频比
   - 将 T2CON 寄存器的 TMR2ON 位置 1 使能 Timer2
   - 如果要设置高速模式，则需要把 MSCKCON.5 置 1
6. 重新开始一个 PWM 周期后，使能 PWM 输出：
   - 等待 Timer2 溢出 (PIR1 寄存器的 TMR2IF 位置 1)

- 设置 PWM1AUX，把 AUX1EN 位置 1，其它各位根据应用需要设置
- 将相关的 TRIS 位清零使能 CCP1 引脚的输出驱动器

1.6.4. ECCP 与 PWM 管脚复用

管脚复用优先级：ECCP > PWM / CxOUT > GPIO。

Pin 脚	输出	条件
PC4	C2OUT	CM[2:0]=110，TRISC.4=0，P3EN=0。且 ECCP 处于非 PWM 模式或 PWM 单输出模式
	P1B	CM[2:0]≠110，TRISC.4=0，且 ECCP 处于 PWM 模式且 P1M≠0
	PWM3	CM[2:0]≠110，TRISC.4=0，且 ECCP 处于非 PWM 模式或 PWM 单输出模式
PC3	PWM4	ECCP 为非全桥模式，P4EN = 1
	P1C	ECCP 为全桥模式
PC2	PWM5	ECCP 为非全桥模式，P5EN = 1
	P1D	ECCP 为全桥模式

表 1-7 ECCP 与 PWM 管脚复用

2. 应用范例

```
/* 文件名：TEST_61F02x_PWM.C
* 功能：     FT61F02x-增强型 PWM 功能演示
*  IC：      FT61F023 SOP16
* 晶振：     16M/2T
* 说明：     芯片工作在 PWM 增强模式下,P1A(PC5)
*            引脚均输出 20kHz 占空比渐增渐减的波形
*            P1B,P1C,P1D,PWM 引脚为普通 I/O 口
*
*               FT61F023   SOP16
*                   ---------------
*  VDD--------|1(VDD) (VSS)16|--------GND
*  NC---------|2(PA7)   (PA0)15|---------NC
*  NC---------|3(PA6)   (PA1)14|---------NC
*  NC---------|4(PA5)   (PA2)13|---------NC
*  P1C---------|5(PC3)   (PA3)12|---------NC
*  P1D---------|6(PC2)   (PC0)11|---------NC
*  NC---------|7(PA4)   (PC1)10|---------NC
*  P1A---------|8(PC5)   (PC4)09|--------P1B
*                   ---------------
*/
//************************************************************
#include "SYSCFG.h"
//************************宏定义*****************************
#define   unchar  unsigned char
#define   unint    unsigned int

//PWM 引脚功能选择 P1XOE 1-PWM 输出  0-普通 I/O 口
#define   PWM_A_OE   P1AOE
#define   PWM_B_OE   P1BOE
#define   PWM_C_OE   P1COE
#define   PWM_D_OE   P1DOE

//PWM 引脚输入输出控制
#define   P1ADir       TRISC5
#define   P1BDir       TRISC4
#define   P1CDir       TRISC3
#define   P1DDir       TRISC2

volatile    unint      pwm_d=0;        //PWM 脉冲宽度值
volatile    bit        SAFlag;
```

```
/*---------------------------------------------
* 函数名：interrupt ISR
* 功能：   中断函数
* 设置中断定时时长=设定值*指令周期
*                    =16000*0.125μs=2000μs
 ---------------------------------------------*/
void interrupt ISR(void)
{
    if(TMR1IF)
    {
        TMR1IF = 0;

        //初值=65536-16000=49536=>0XC180
        TMR1L = 0X80;           //赋初值=>TMR1H=0XC1;TMR1L=0X80
        TMR1H = 0XC1;
    }
}
/*---------------------------------------------
* 函数名：POWER_INITIAL
* 功能：   上电系统初始化
* 输入：   无
* 输出：   无
 ---------------------------------------------*/
void POWER_INITIAL (void)
{
    OSCCON = 0B01110001;        //IRCF=111=16MHz/2T=8MHz,0.125μs
    INTCON = 0;                 //暂禁止所有中断

    PORTA = 0B00000000;
    TRISA = 0B00000000;         //PA 输入输出  1-输入  0-输出
    PORTC = 0B00000000;
    TRISC = 0B11111111;         //PC 输入输出  1-输入  0-输出

    WPUA = 0;                   //禁止所有 PA 口上拉
    WPUC = 0;                   //禁止所有 PC 口上拉
    OPTION = 0B00001000;        //Bit3=1,WDT MODE,PS=000=WDT RATE 1:1
    MSCKCON = 0B00000000;
    //Bit6->0,禁止 PA4，PC5 稳压输出
    //Bit5->0,TIMER2 时钟为 Fosc
    //Bit4->0,禁止 LVR
    CMCON0 = 0B00000111;        //关闭比较器，CxIN 为数字 IO 口
}
```

```
/*-----------------------------------------------
* 函数名：TIMER1_INITIAL
* 功能：    定时器 1 初始化
* 设置定时时长=(1/系统时钟周期)*指令周期*预分频值*(65536-TMR1H:TMR1L)
*                  =(1/16000000)*2*1*( 65536-46536)=2000μs
 -----------------------------------------------*/
void TIMER1_INITIAL (void)
{
    //需要在中断里重新赋初始值
    T1CON = 0B00000000;
    //Bit[5:4]:00-T2 时钟分频  1:1
    //Bit2:    0-Timer1 使用内部时钟
    //Bit1:    0-T1 时钟源选择内部时钟
    TMR1L = 0X80;                  //赋初值 TMR1H=0XC1;TMR1L=0X80
    TMR1H = 0XC1;
    TMR1IE = 1;                    //使能 TMER1 的中断
    TMR1ON = 1;                    //使能 TMER1 启动
    PEIE=1;                        //使能外设中断
    GIE = 1;                       //使能全局中断
}
/*-----------------------------------------------
* 函数名：pwm_duty_count
* 功能：    将十位 pwm_d 的值赋值给 CCPR1L:CCP1CON<5:4>
* 输入：    无
* 输出：    无
 -----------------------------------------------*/
void pwm_duty_count (void)
{
    unint   lsb22;
    unchar lsb23;
    lsb22 = (unchar) pwm_d & 0B00000011;
    CCP1CON = CCP1CON & 0B11001111;
    lsb22 <<= 4;
    lsb23 = (unchar)lsb22;
    CCP1CON = CCP1CON | lsb23;     //赋值 10Bit PWM 脉冲宽度的低 2 位 LSB
    lsb22 = pwm_d >> 2;            //赋值 10Bit PWM 脉冲宽度的高 8 位 MSB
    lsb23 = (unchar)lsb22;
    CCPR1L = lsb23;
}
/*-----------------------------------------------
* 函数名：PWM_INITIAL
* 功能：    PWM 初始化
* 设置 PWM 周期  = (PR2+1)*4*(1/Fosc)*T2 分频比值
*                  = (99+1)*4*(1/16000000)*1 = 25μs
```

```
*       PWM 频率=1/PWM 周期=1/25μs=40kHz
 ------------------------------------------------*/
void PWM_INITIAL (unchar SET_PR2)
{
    MSCKCON = 0B00000000;
    //Bit6->0,禁止 PA4，PC5 稳压输出
    //Bit5->0,TIMER2 时钟为 Fosc ; 1 为 TIMER2 时钟为 32MHz
    //Bit4->0,禁止 LVR

    T2CON = 0B00000000;              //Bit[1:0]=00, T2 分频比为 1:1
    PR2 = SET_PR2;                   //设置 PWM 的周期
    CCP1CON = 0B10001101;
    //Bit[7:6]=00;   P1A 单输出,P1B,P1C,P1D 位普通 I/O 口
    //Bit[5:4]=00;   10Bit PWM 占空比低 2 位
    //Bit[3:0]=1101;PWM 模式，P1A 高电平有效,P1B 低电平有效

    pwm_duty_count();                //PWM 占空比计算  占空比 pwm_d/((PR2+1)*4)

    PWM1CON = 0B10000001;
    //Bit7=1,自动关闭时 PWM 自动重启
    //死区时间=Bit<6:0>*(1/Fosc)*2
    //          =1*(1/16000000)*2=0.125ns
    ECCPAS = 0B00001111;
    //Bit[6:4]=000,禁止自动关闭
    //Bit[3:2]=11, P1A,P1C 关闭时为三态
    //Bit[1:0]=11, P1B,P1D 关闭时为三态

    PWM1AUX = 0B10000000;
    //Bit7=1,使能 PWM 辅助功能
    //Bit6=0,使能 PWM 连续输出
    //Bit[5:2]=0000,P1C,P1D 为半桥 PWM 输出
    //Bit[1:0]=00,P1A,P1B,P1E,P1F 引脚为 IO

    TMR2IF = 0;                      //清 T2 中断标志位
}
/*-------------------------------------------------
* 函数名：DelayUs
* 功能：   短延时函数 --16M-2T--大概快 1%左右.
* 输入：   Time 延时时间长度 延时时长 Time μs
* 输出：   无
 ------------------------------------------------*/
void DelayUs(unsigned char Time)
{
    unsigned char a;
```

```
    for(a=0;a<Time;a++)
    {
        NOP();
    }
}
/*-------------------------------------------------
* 函数名：DelayMs
* 功能：   短延时函数   快 1%
* 输入：   Time  延时时间长度  延时时长 Time ms
* 输出：   无
 -------------------------------------------------*/
void DelayMs(unsigned char Time)
{
    unsigned char a,b;
    for(a=0;a<Time;a++)
    {
        for(b=0;b<5;b++)
        {
            DelayUs(197);
        }
    }
}
/*-------------------------------------------------
* 函数名：main
* 功能：   主函数
* 输入：   无
* 输出：   无
 -------------------------------------------------*/
void main(void)
{
    POWER_INITIAL();
    TIMER1_INITIAL();
    pwm_d = 10;
    PWM_INITIAL(99);            //Tpwm = (99+1)*4*(1/16000000)*1 = 25μs = 40kHz
                                //PWM 阶数  = 4 * (99+1) = 400
    PWM_A_OE = 1;
    P1ADir = 0;

    TMR2ON = 1;                 //使能开启 T2
    SAFlag = 1;
    while(1)
    {
        if(SAFlag == 1)         //PWM 占空比增加
        {
```

```
            pwm_d++;
            if(pwm_d > 400)
            {
                SAFlag = 0;
            }
        }
        else                        //PWM 占空比减小
        {
            pwm_d--;
            if(pwm_d == 0)
            {
                SAFlag = 1;
            }
        }
        pwm_duty_count();           //载入 PWM 占空比
        DelayMs(10);                //延迟 10ms
    }
}
```

联系信息

Fremont Micro Devices Corporation

#5−8, 10/F, Changhong Building
Ke-Ji Nan 12 Road, Nanshan District,
Shenzhen, Guangdong, PRC 518057

Tel: (+86 755) 8611 7811
Fax: (+86 755) 8611 7810

Fremont Micro Devices (HK) Limited

#16, 16/F, Block B, Veristrong Industrial Centre,
34−36 Au Pui Wan Street, Fotan, Shatin, Hong Kong SAR

Tel: (+852) 2781 1186
Fax: (+852) 2781 1144

http://www.fremontmicro.com