



# WF8M165S16M 2KFE DATASHEET V3.0



## 1 特性

- 自主知识产权精简指令集
- 8 层硬件堆栈 x11bit
- 2T 或 4T 指令周期
- 2Kx14b 程序存储空间
  - ◇ 程序存储空间的 checksum 自动校验
  - ◇ 可配置，User Option
- 256x8b 数据 EEPROM
  - ◇ 数据 EEPROM 在应用编程
- 128x8b SRAM
- 1 x 带 8 位预分频的定时器 0
- 1x 带 3 位预分频的 16 位定时器 1
- 1 x 带 8 位预分频的定时器 2
- 增强性捕捉、比较和可编程“死区”时间的 PWM 模块
  - ◇ 时钟源可选：系统时钟或者是内部 32MHz 时钟
  - ◇ 单次脉冲模式
  - ◇ 最多 3 对带“死区”的 PWM 输出
- 3x12bit Timer，3x12bit PWM，支持 BUZZER 模式
- 带 7 位预分频的 WDT，溢出频率约为 16ms~2048ms
- 上电延迟计数器 PWRT
- 低功耗模式 SLEEP
- 多个唤醒源，INT、端口变化中断、WDT 和数据 EEPROM 写完成，等等
- 内置高速 16M RC 振荡器
- 内置低速 32K RC 振荡器
- 支持外部晶振 16M 或 32K，以及外部时钟模式
  - ◇ 时钟缺失检测
  - ◇ 双速启动模式
- 内置 10 位的 ADC，支持 8 个通道（7 个外部通道 + 1 个内部 1/4VDD 通道）
  - ◇ 参考电压可选：外部 Vref，VDD，内部 2V/3V
- 内置 2 个高速高精度比较器
  - ◇ 可编程的参考电压
  - ◇ 比较结果可直接输出
- 低电压复位 LVR：1.8V/2.0V/2.2V/2.8V
- 低电压检测 LVD：1.6V/1.8V/2.0V/2.4V/2.8V/3.0V/3.6V/4.2V
- 两路稳压输出：
  - ◇ 每路分别可输出多达 32 档电压
- 最多 14 个通用 IO，16 根芯片管脚
  - ◇ 14 个 IO 带独立上拉控制
  - ◇ 4 个 IO 带独立下拉控制
- 端口变化中断，RA0~RA7
- 支持在系统编程 ICSP



- 支持在线调试，3 个硬件断点
- 程序空间保护
- 工作电压范围：1.8V~5.5V
- 最大时钟工作频率：16MHz
  - ◇  $F_{SYS}=8\text{MHz}$ : 1.8V~5.5V
  - ◇  $F_{SYS}=16\text{MHz}$ : 2.7V~5.5V
- 封装类型：SOP8，SOP14，SOP16

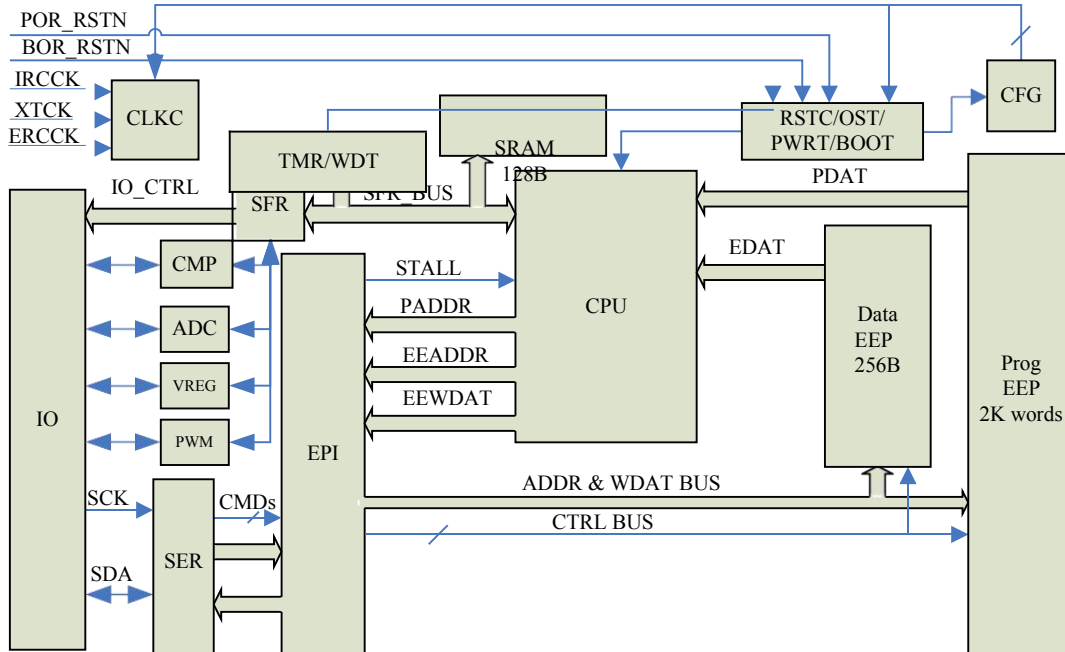


图 1.1 系统功能框图

## 1.1 程序存储器

地址寄存器为 13 位 (0x0000 ~ 0x1FFF)，最多支持 8K 地址空间。程序存储器一共有 2K 个字 (0x0000 ~ 0x07FF)，加上 6 个额外的用户配置、工厂配置区，总共有 2K+64 个字，它们由 EEPROM 构成。一个 PAGE 是 16 个字，整个 PROM 一共有 134 个 PAGE。

其中 0~0x07FF 对主程序区访问，未实现部分 0x800~0x1FFF 保留。用户和工厂配置信息区从 0x2000 开始，到 0x205F 结束。

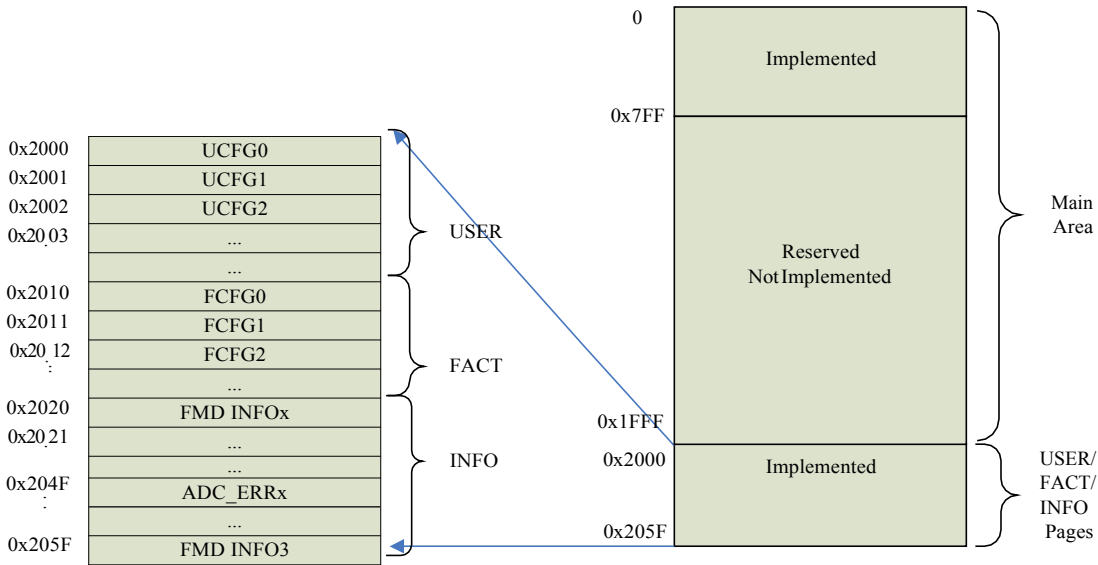


图 1.2 程序空间地址映射

## 1.2 脚位图

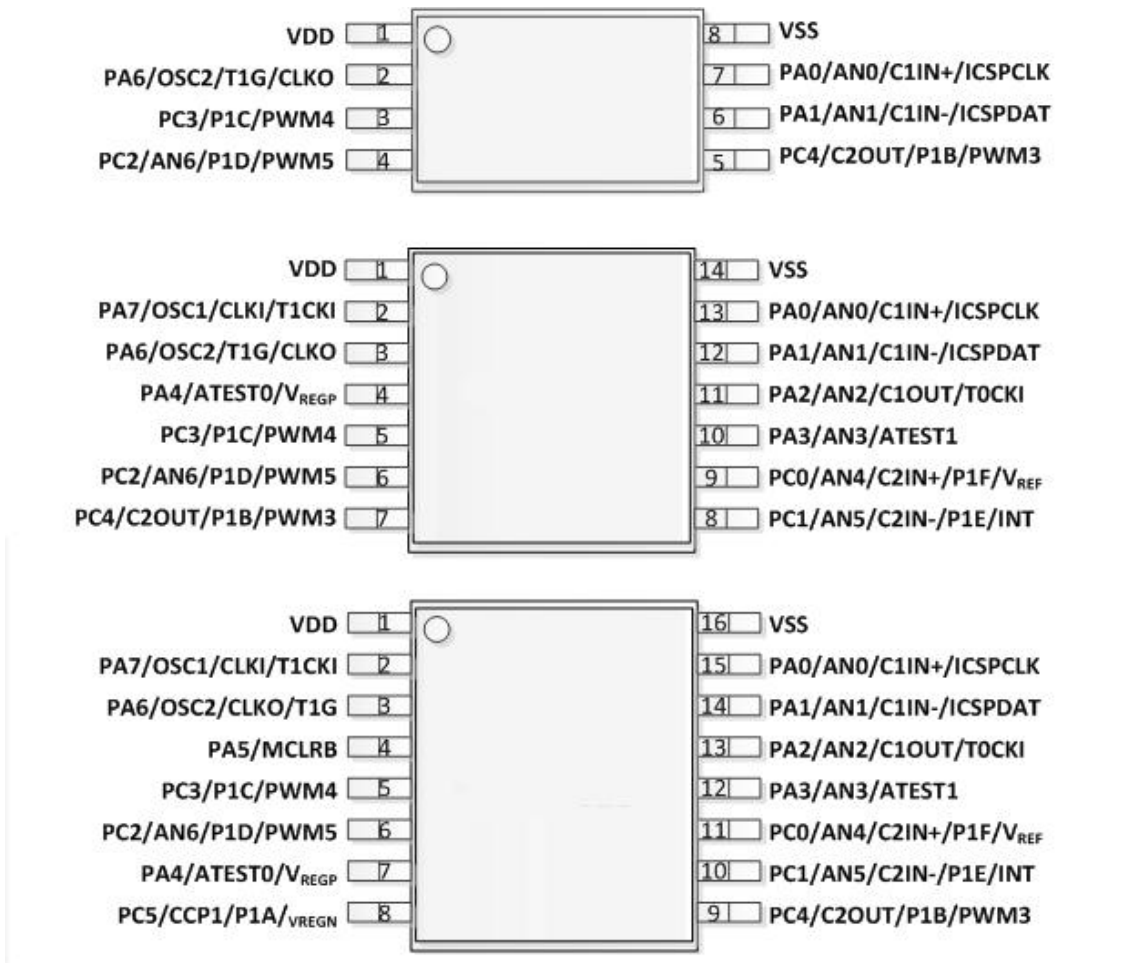


图 1.3.1 WF8M165 所规划脚位图：SOP8，SOP14，SOP16



以下为芯片管脚的详细描述：

管脚名	功能名	输入信号类型	输出信号类型	具体描述	上下拉
PA0/AN0/C1IN+/ICSPCLK	PA0	TTL	CMOS	GPIO with IOC and WPU	可配置上拉
	AN0	AN	—	A/D 通道 0 输入	
	ICSPCLK	ST	—	Debug/烧录模式串口 clock 信号 (Fmax=6MHz)	
	C1IN+	AN	—	Comparator1 non-inverting input	
PA1/AN1/C1IN-/ICSPDAT	PA1	TTL	CMOS	GPIO with IOC and WPU	可配置上拉
	AN1	AN	—	A/D 通道 1 输入	
	C1IN-	AN	—	Comparator1 inverting input	
	ICSPDAT	TTL	CMOS	Debug/烧录模式串口 data 信号 (Fmax=6MHz)	
PA2/AN2/C1OUT/T0CKI	PA2	TTL	CMOS	GPIO with IOC and WPU	可配置上拉
	AN2	AN	—	A/D 通道 2 输入	
	C1OUT	—	CMOS	Comparator1 output	
	T0CKI	TTL	—	Timer 0 源头时钟输入 (Fmax=4MHz)	
PA3/AN3/ATEST1	PA3	TTL	—	GPIO with IOC and WPU	可配置上拉
	AN3	AN	—	A/D 通道 3 输入	
	ATEST1	AN	AN	模拟测试管脚 1	内部测试用
PA4/ATEST0/VREGP	PA4	TTL	CMOS	GPIO with IOC and WPU	可配置上拉/下拉
	ATEST0	AN	AN	模拟测试管脚	内部测试用
	VREGP	—	AN	高压档稳压输出	
PA5/MCLRB	PA5	TTL	CMOS	GPIO with IOC and WPU(or input only)	可配置上拉
	MCLRB	TTL	—	外部复位输入	MCLRB
PA6/OSC2/T1G/CLKO	PA6	TTL	CMOS	GPIO with IOC and WPU	可配置上拉
	OSC2	—	XTAL	Crystal/Resonator	OSC2
	T1G	ST	—	Timer1 门控输入	



	CLKO	—	CMOS	测试时钟输出 (Fmax=10MHz)	CLKO
PA7/OSC1/CLKI/T1CKI	PA7	TTL	CMOS	GPIO with IOC and WPU	可配置上拉
	CLKI	ST	—	External clock input/RC oscillator connection	
	OSC1	XTAL	—	Crystal/Resonator	
	T1CKI	ST	—	Timer1 外部时钟	
PC0/AN4/C2IN+/P1F/VREF	PC0	TTL	CMOS	PORTC I/O	可配置上拉
	AN4	AN	—	A/D 通道 4 输入	
	C2IN+	AN	—	Comparator2 non-inverting input	
	P1F	—	CMOS	增强型 PWM 输出	
	VREF	AN	—	A/D 外部参考电压输入	
PC1/AN5/C2IN-/P1E/INT	PC1	TTL	CMOS	PORTC I/O	可配置上拉/下拉
	AN5	AN	—	A/D 通道 5 输入	
	C1IN-	AN	—	Comparator2 inverting input	
	P1E	—	CMOS	增强型 PWM 输出	
	INT	TTL	—	外部中断输入	
PC2/AN6/P1D/PWM5	PC2	TTL	CMOS	PORTC I/O	可配置上拉/下拉
	AN6	AN	—	A/D 通道 6 输入	
	P1D	—	CMOS	增强型 PWM 输出	
	PWM5	—	CMOS	PWM5 输出	
PC3/P1C/PWM4	PC3	TTL	CMOS	PORTC I/O	可配置上拉/下拉
	P1C	—	CMOS	增强型 PWM 输出	
	PWM4	—	CMOS	PWM4 输出	
PC4/C2OUT/P1B/PWM3	PC4	TTL	CMOS	PORTC I/O	可配置上拉
	C2OUT	—	CMOS	Comparator2 output	
	P1B	—	CMOS	增强型 PWM 输出	
	PWM3	—	CMOS	PWM3 输出	
PC5/CCP1/P1A/VREGN	PC5	TTL	CMOS	PORTC I/O	可配置上拉
	CCP1	ST	CMOS	捕捉输入/比较输出	
	P1A	—	CMOS	增强型 PWM 输出	
	VREGN	—	AN	低压档稳压输出	

注意：

1. IOC : Interrupt on change · 通用 IO
2. WPU : Weak pullup
3. ST : 带 CMOS 电平的施密特触发器输入
4. AN : 模拟输入或输出



## 2 特殊功能寄存器

### 2.1 地址映射

#### 2.1.1 SFR · BANK0

ADDR	Name	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	POR reset	
BANK0											
0	INDF	使用 FSR 的内容对数据存储器进行访问（非物理寄存器）								xxx x xxxx	
1	TMR0	TIMER0 计数器								xxx x xxxx	
2	PCL	程序计数器低 8 位								0000 0000	
3	STATUS	FSR_B8	PAGE[1:0]		/TF	/PF	Z	HC	C	-- 01 1xxx	
4	FSR	间接寻址指针寄存器								xxx x xxxx	
5	PORTA	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	xxx 0 0000	
6	-	-								---- ----	
7	PORTC	-	-	PC5	PC4	PC3	PC2	PC1	PC0	-- xx x 000	
8	-	-								---- ----	
9	-	-								1111 1111	
A	PCLATH	-	-	-	程序计数器高 5 位锁存器					---0 0000	
B	INTCON	GIE	PEIE	T0IE	INTE	PAIE	T0IF	INTF	PAIF	0000 0000	
C	PIR1	EEIF	CKMEA IF	-	C2IF	C1IF	OSFIF	TMR2IF	TMR1IF	0000 0000	
D	PIR2	-	-	-	-	-	-	ADIF	CCP1IF	---- --00	
E	TMR1L	16 位 TIMER1 低字节保持寄存器								xxx x xxxx	
F	TMR1H	16 位 TIMER1 高字节保持寄存器								xxx x xxxx	
10	T1CON	T1GIN V	TMR1G E	T1CKPS 1	T1CKPS 0	T1OSC EN	T1SYN C	TMR1C S	TMR1O N	0000 0000	
11	TMR2	TIMER2[7:0] timer2 module register								0000 0000	
12	T2CON	-	TOUTPS[3:0]				TMR2O N	T2CKPS[1:0]		-000 0000	
13	CCPR1L	捕捉、比较、PWM 寄存器 1 的低字节								xxx x xxxx	
14	CCPR1H	捕捉、比较、PWM 寄存器 1 的高字节								xxx x xxxx	
15	CCP1CO N	P1M[1:0]		DC1B[1:0]		CCP1M[3:0]				0000 0000	
16	PWM1CO N	PRSEN	PDC[6:0]								0000 0000
17	ECCPAS	ECCPA SE	ECCPAS[2:0]			PSSAC[1:0]		PSSBD[1:0]		0000 0000	
18	WDTCON	-	-	-	WDTPS[3:0]				SWDTE N	---0 1000	
19	CMCON0	C2OUT	C1OUT	C2INV	C1INV	CIS	CM[2:0]			0000 0000	



1A	CMCON1	-	-	-	-	-	-	T1GSS	C2SYN C	---- --10
1B	MSCKCON	-	VREG_ OE	T2CKS RC	SLVRE N	-	CKMAV G	CKCNTI		0000 -00-
1C	SOSCPRL	SOSCPRL [7:0]								1111 1111
1D	SOSCPRH	-	-	-	-	SOSCPRH [11:8]			---- 1111	
1E	ADRESH	左对齐格式下 AD 结果的高 8 位或者右对齐格式下的高 2 位								xxx x xxxx
1F	ADCON0	ADFM	VCFG1	VCFG0	CHS[2:0]			GO/DO NE	ADON	0000 0000
20~7F	SRAM BANK0, (96Bytes)									xxx x xxxx

## 2.1.2 SFR · BANK1

ADDR	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	POR reset
BANK1										
80	INDF	使用 SFR 的内容对数据存储器进行访问（非物理寄存器）								xx x x xx xx
81	OPTION	/PAPU	INTED G	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111
82	PCL	程序计数器低 8 位								0000 0000
83	STATUS	FSR_B8	PAGE[1:0]		/TF	/PF	Z	HC	C	-- 01 1x x x
84	FSR	间接寻址指针寄存器								xx x x xx x x
85	TRISA	TRISA[7:6]		TRISA[5 ]	TRISA[4:0]					1111 1111
86	-	-								---- ----
87	TRISC	-	-	TRISC[5:0]						-- 11 1111
88	WPUC	-	-	WPUC[5:0]						-- 00 0000
89	WPD	-	-	-	WPDA4	WPDC1	WPDC2	WPDC3	-	--- 0 000 -
8A	PCLATH	-	-	-	程序计数器高 5 位锁存器				-	--- 0 0000
8B	INTCON	GIE	PEIE	T0IE	INTE	PAIE	T0IF	INTF	PAIF	0000 0000
8C	PIE1	EEIE	CKMIE IE	-	C2IE	C1IE	OSFIE	TMR2IE	TMR1IE	0000 0000
8D	PIE2	-	-	-	-	-	-	ADIE	CCP1IE	---- -- 00
8E	PCON	VREF_ OE	LVDL[2:0]			LVDEN	LVDW	/POR	/BOR	0000 00qq
8F	OSCCON	LFOMOD	IRCF[2:0]			OSTS	HTS	LTS	SCS	0101 x 000
90	PWM1AUX	AUX1EN	P1OS	P1FOE	P1EOE	P1DOE	P1COE	P1BOE	P1AOE	0 -- 0 0000
91	ANSEL	ANSEL[7:0]								1111 1111
92	PR2	PR2[7:0] timer2 period register								1111 1111
93	-	-								0000 0000
94	-	-								0000 0000
95	WPUA	WPUA[7:6]		WPUA[	WPUA[4:0]					1111 1111





				5]						
96	IOCA	IOCA[7:0]								0000 0000
97	-	-								0000 0000
98	-	-								0000 0000
99	VRCON	VREN	-	VRR	-	VR[3:0]			0-0- 0000	
9A	EEDAT	EEDAT[7:0]								0000 0000
9B	EEADR	EEADR[7:0]								0000 0000
9C	EECON1	-	-	WREN3	WREN2	WRERR	WREN1	-	RD	--00 x 0 - 0
9D	EECON2	-	-	-	-	-	-	-	WR	---- ----0
9E	ADRESL	左对齐格式下 AD 结果的低 2 位或者右对齐下结果的低 8 位								xx x x xx x x
9F	ADCON1	DIVS	ADCS[2:0]		-	-	-	-		0000 ----
A0~BF	SRAM BANK1 (32Bytes)									xx x x xx x x
C0~EF	-									---- ----
F0~FF	SRAM, 访问 BANK0 的 0x70~0x7F									xx x x xx x x

### 2.1.3 SFR · BANK2

ADDR	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	POR reset
BANK3										
100	INDF	使用 SFR 的内容对数据存储器进行访问（非物理寄存器）								xxxx xxxx
101	-	-								---- ----
102	PCL	程序计数器低 8 位								0000 0000
103	STATUS	FSR_B8	PAGE[1:0]		/TF	/P F	Z	HC	C	- 001 1xxx
104	FSR	间接寻址指针寄存器								xxxx xxxx
105	-	-								---- ----
106	-	-								---- ----
107	-	-								---- ----
108	VCON1	-	VREGM[1:0]		VREGHB[4:0]					-000 0000
109	VCON2	-	-	-	VREGLB[4:0]					---0 0000
10A	PCLATH	-	-	-	程序计数器高 5 位锁存器					---0 0000
10B	INTCON	GIE	PEIE	TOIE	INTE	PAIE	TOIF	INTF	PAIF	0000 0000
10C	TMR3L	12 位定时器 3 低 8 位								xxxx xxxx
10D	TMR3H	12 位定时器 3 高 4 位				PWM3 匹配寄存器 PR3 高 4 位				x xxx 1111
10E	PR3L	PWM3 匹配寄存器 PR3 低 8 位								1111 1111
10F	PWM3CR0	P3INTS	P3PER[2:0]			P3CKSRC[2:0]		P3BZR		0000 0000
110	PWM3CR1	P3EN	P3POL	TMR3PS[2:0]		TMR3ON	TMR3IE	TMR3IF		0000 0000
111	T3CKDIV	定时器 3 时钟分频寄存器								0000 0000
112	TMR4L	12 位定时器 4 低 8 位								xxxx xxxx
113	TMR4H	12 位定时器 4 高 4 位				PWM4 匹配寄存器 PR4 高 4 位				x xxx 1111
114	PR4L	PWM4 匹配寄存器 PR4 低 8 位								1111 1111
115	PWM4CR0	P4INTS	P4PER[2:0]			P4CKSRC[2:0]		P4BZR		0000 0000



116	PWM4CR1	P4EN	P4POL	TMR4PS[2:0]	TMR4ON	TMR4IE	TMR4IF	0000 0000
117	T4CKDIV	定时器 4 时钟分频寄存器						0000 0000
118	TMR5L	12 位定时器 5 低 8 位						XXXX XXXX
119	TMR5H	12 位定时器 5 高 4 位			PWM5 匹配寄存器 PR5 高 4 位			x xxx 1111
11A	PR5L	PWM5 匹配寄存器 PR5 低 8 位						1111 1111
11B	PWM5CR0	P5INTS	P5PER[2:0]	P5CKSRC[2:0]		P5BZR		0000 0000
11C	PWM5CR1	P5EN	P5POL	TMR5PS[2:0]	TMR5ON	TMR5IE	TMR5IF	0000 0000
11D	T5CKDIV	定时器 5 时钟分频寄存器						0000 0000
11E	-	-						---- ----
11F	-	-						---- ----
120~ 16F	-						---- ----	
170~ 17F	SRAM, 访问 BANK0 的 0x70~0x7F						XXXX XXXX	

注意：

1. INDF 不是物理寄存器；
2. 灰色部分表示没有实现；
3. 未实现的寄存器位不要写 1，以后芯片升级可能会用到；
4. ANSEL 复位值为 0xFF，即相关 IO 为模拟管脚，故任何复位之后去读 PORTA 或者 PORTC，PORTA[3:0] 和 PORTC[2:0] 返回的值为 0，因为此时它们为模拟管脚，这与它们的数据寄存器内容无关。

### 2.1.4 STATUS 寄存器，地址 0x03，0x83

Bit	7	6	5	4	3	2	1	0
Name	FSR_B8	PAGE[1:0]		/TF	/PF	Z	HC	C
Reset	0	00		1	1	x	x	x
Type	RW	RW		RO	RO	RW	RW	RW

Bit	Name	Function
7	FSR_B8	FSR 寄存器第 8 位，与 FSR 组成一个 9 位的寄存器，在间接寻址时使用 详见 <a href="#">INDF 和 FSR 寄存器</a> 一节
6:5	PAGE[1:0]	<b>PAGE:</b> Register Bank Select bit (used for direct addressing) 00 = Bank0 (0x00~ 0x7F) 01 = Bank 1 (0x80~ 0xFF) 1x = Bank2 (0x100~ 0x17F)
4	/TF	<b>TF:</b> Time-out bit 1 = After power-up，CLRWDT instruction or SLEEP instruction 0 = A WDT time-out occurred
3	/PF	<b>PF:</b> Power-down bit 1 = After power-up or by the CLRWDT instruction 0 = By execution of the SLEEP instruction
2	Z	<b>Z:</b> Zero bit 1 = The result of an arithmetic or logic operation is zero



		0 = The result of an arithmetic or logic operation is not zero
1	HC	<b>HC:</b> Digit Carry/Borrow bit (ADDWR, ADDWI, SUBWI, SUBWR instructions), For Borrow, the polarity is reversed. 1 = A carry-out from the 4 <sup>th</sup> low-order bit of the result occurred 0 = No carry-out from the 4 <sup>th</sup> low-order bit of the result
0	C	<b>C:</b> Carry/Borrow bit( <b>1</b> ) (ADDWR, ADDWI, SUBWI, SUBWR instructions) 1 = A carry-out from the Most Significant bit of the result occurred 0 = No carry-out from the Most Significant bit of the result occurred

/TF	/PF	条件
1	1	上电或者低电压复位
0	U	WDT 复位
0	0	WDT 唤醒
U	U	正常运行下发生 MCLR 复位
1	0	睡眠状态下发生 MCLR 复位

注意：

1. 和其它寄存器一样，状态寄存器也可以作为任何指令的目标寄存器。如果一条指令影响 Z、HC 或 C 位的指令以状态寄存器作为目标寄存器，将禁止对这三位的写操作，它们只受逻辑结果影响，被置 1 或清 0。因此，当执行一条把状态寄存器作为目标寄存器的指令后，STATUS 内容可能和预想的不一致；
2. 建议只使用 BCR、BSR、SWAPR 和 STR 指令来改变状态寄存器。

## 2.1.6 配置寄存器 UCFGx

软件不能访问 UCFG0、UCFG1、UCFG2，它们只在上电过程由硬件写入。

UCFG0，PROM 地址 0x2000。

Bit	7	6	5	4	3	2	1	0
Name	-	CPB	MCLRE	PWRTEB	WDTE	FOSC<2:0>		

位	名称	描述
Bit[7]	NA	保留位，读 0
Bit[6]	CPB	1：Flash 内容不保护 0：启动 Flash 内容保护，MCU 能读，串口不能读 注意： 此位只能由 1 改写为 0，而不能由 0 改写为 1。由 0 改写成 1 的唯一方法是进行一次包括 USER_OPT 在内的片擦操作，并且重新上电后 CPB 才变为 1
Bit[5]	MCLRE	1：PA5/MCLR 脚执行 MCLR 功能，是复位脚 0：PA5/MCLR 脚执行 GPIO 功能
Bit[4]	PWRTEB	1：PWRT 禁止 0：PWRT 使能
Bit[3]	WDTE	1：WDT 使能，程序不能禁止



		0 :WDT 禁止，但程序可通过设置 WDTCON 的 SWDTEN 位将 WDT 使能
Bit[2:0]	FOSC[2:0]	000 : 32K 晶振模式，PA6/PA7 接低速晶体 001 : 20M 晶振模式，PA6/PA7 接高速晶体 010 : 外部时钟模式，PA6 为 IO 功能，PA7 接时钟输入 011 : INTOSC 模式，PA6 输出系统时钟的 2 分频，PA7 为 IO 引脚 1xx : INTOSCIO 模式，PA6 为 IO 引脚，PA7 为 IO 引脚

UCFG1，PROM 地址 0x2001。

Bit	7	6	5	4	3	2	1	0
Name	-	CSUME NB	TSEL	FCMEN	IESO	RD_CTR L	LVREN1	LVREN 0

位	名称	描述
Bit[7]	-	保留
Bit[6]	CSUMENB	Checksum enable 1 : 禁止 checksum 功能 0 : 使能 checksum 功能，boot 完成后，硬件自动把 2K PROM 空间所有单元内容相加，结果与 0x2007 单元内容比较，如果相等说明校验成功；校验失败时 CPU 将不执行程序
Bit[5]	TSEL	指令周期选择位 1 : 指令周期为 2T 0 : 指令周期为 4T
Bit[4]	FCMEN	时钟故障监视使能 1 : 使能时钟故障监视 0 : 禁止时钟故障监视
Bit[3]	IESO	双速时钟使能 1 : 使能双速时钟模式 0 : 禁止双速时钟模式
Bit[2]	RD_CTRL	输出模式时读端口控制 1 : 读数据端口返回的 PAD 上的值 0 : 读数据端口返回的 Latch 上的值
BIT[1:0]	LVREN[1:0]	低电压复位选择 00 : 使能低电压复位 01 : LVR 由 MSCKCON 的 SLVREN 决定 10 : MCU 正常模式时开启 LVR，睡眠模式时关闭 LVR，跟 SLVREN 位无关 11 : 禁止低电压复位

UCFG2，PROM 地址 0x2002。

Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	LVRS[3:0]			

位	名称	描述
Bit[7:4]	-	保留位
Bit[3:0]	LVRS[3:0]	低电压复位阈值选择

	数值	电压
	0010	1.8V
	0011	2.0V
	0100	2.2V
	0110	2.8V
	其它	保留

## 2.1.7 PCL 和 PCLATH

程序计数器（PC）为 11 位宽。其低 8 位来自可读写的 PCL 寄存器，高 3 位（PC<10:8>）来自 PCLATH，不能直接读写。只要发生复位，PC 就将被清 0。下图显示了装载 PC 值的两种情形。注意图右边的 LCALL 和 LJUMP 指令，由于指令中的操作码为 11 位，而芯片的 PC 刚好是 11 位，所以这时 PCLATH 并不需要用到。

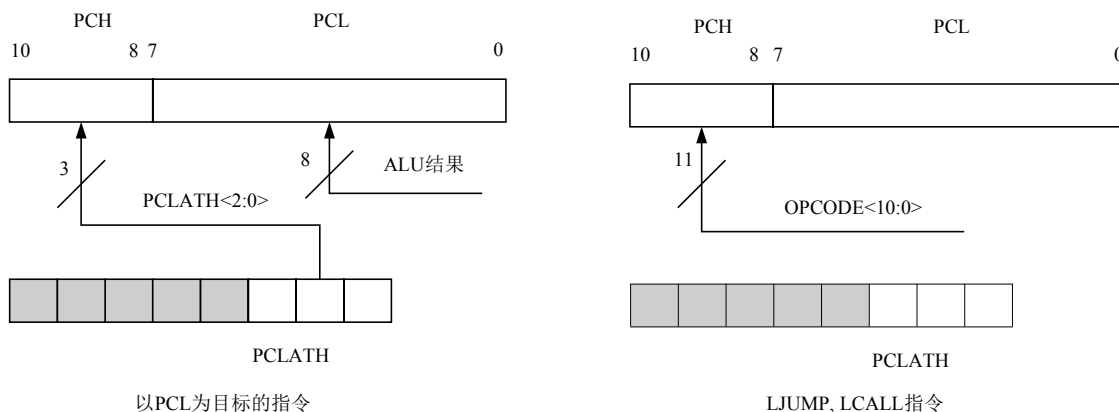


图 2.1.1 不同情况下 PC 的加载

### 修改 PCL

执行任何以 PCL 寄存器为目标寄存器的指令将同时使程序计数器 PC<10:8>位被 PCLATH 内容取代。这样可通过将所需的高 3 位写入 PCLATH 寄存器来改变程序计数器的所有内容。

计算 LJUMP 指令是通过向程序计数器加入偏移量（ADDWR PCL）来实现的。通过修改 PCL 寄存器跳转到查找表或程序分支表（计算 LJUMP）时应特别谨慎。假定 PCLATH 设置为表的起始地址，如果表长度大于 255 条指令，或如果存储器地址的低 8 位在表的中间从 0xFF 计满返回到 0x00，那么在每次表起始地址与表内的目标地址之间发生计满返回时，PCLATH 必须递增。

## 2.1.8 INDF 和 FSR 寄存器

INDF 不是物理存在的寄存器，对 INDF 进行寻址将产生间接寻址，可寻址范围为 0~511，但由于 BANK3 没有实现，所以实现可寻址的寄存器个数为 384。

任何使用 INDF 寄存器的指令，实际上是对文件选择寄存器{FSR\_B8,FSR}所指向的单元进行存取。需要注意的是，当 FSR\_B8 为 1 且使用间接寻址访问 SFR 空间时，{FSR\_B8,FSR}将指向 BANK2，如图 2.1.2 所示。间接对 INDF 进行读操作将返回 0。间接对 INDF 进行写将导致空操作（可能会影响状态标志位）。

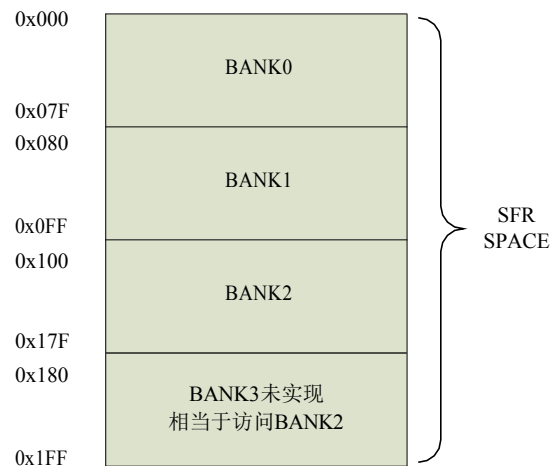


图 2.1.2 间接寻址

## 2.1.9 关于寄存器保留位

如 2.1.1/2.1.2/2.1.3 章节的表格所示，SFR 空间有部分寄存器或者寄存器位没有实现，未实现的寄存器位是**保留位**，软件读返回的是 0，写无效。

不建议程序对这些保留位写 1，这有可能会给以后程序的移植带来问题，因为后续芯片产品可能会用到这此位。

### 3 系统时钟源

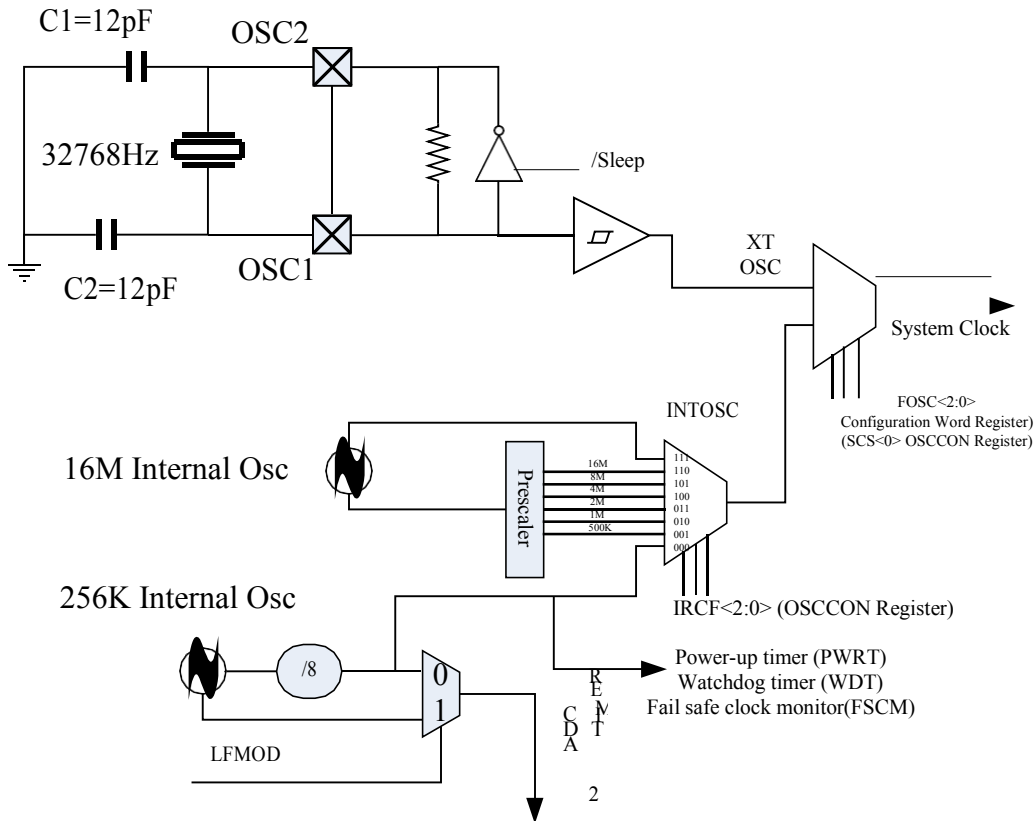


图 3.1 系统时钟源框图

本芯片包含 4 个时钟源：2 个内置振荡器作为各种时钟源，1 个外部晶体振荡器，1 个外部时钟灌入源。内置振荡器包括 1 个内部 16M 高速精准振荡器(HFINTOSC)，1 个内部 32K/256K(LFINTOSC)低速低功耗振荡器。这些时钟或振荡器结合预分频器可以给系统提供各种频率的时钟源。

系统时钟源的预分频器比例由 OPTION 寄存器里的 IRCF<2:0>位控制。

**注意：**

内部慢时钟的 256KHz 模式只供 ADC (当 ADCS 为 x11 时) 使用，看门狗、系统时钟源 (IRCF=000) 以及 PWRT 统一使用 8 分频之后的输出，即 32KHz，而不管 LFMOD 为何值。

#### 3.1 时钟源模式

时钟源模式分为外部和内部模式。

- 外部时钟模式依靠外部电路提供时钟源，比如外部时钟 EC 模式，晶体谐振器 XT、LP 模式。
- 内部时钟模式内置于振荡器模块中，振荡器模块有 16MHz 高频振荡器和 32KHz 低频振荡器。可通过 OSCCON 寄存器的系统时钟选择位 (SCS) 来选择内部或者外部时钟源。

## 3.2 外部时钟模式

### 3.2.1 振荡器起振定时器 ( OST )

如果振荡器模块配置为 LP、XT 模式，振荡器起振定时器 (OST) 将对来自 OSC1 的振荡计数 1024 次。这发生在上电复位 (POR) 之后以及上电延时定时器 (PWRT) 延时结束 (如果被使能) 时，或从休眠中唤醒后。在此期间，程序计数器不递增，程序执行暂停。OST 确保使用石英晶体谐振器或陶瓷谐振器的振荡器电路已经启动并向振荡器模块提供稳定的系统时钟信号。当在时钟源之间切换时，需要一定的延时以使新时钟稳定。

**注意：**OST 复用了 WDT 定时器，故在 OST 对晶体时钟计数时，WDT 功能被屏蔽，待 OST 发生溢出后，WDT 功能才恢复 (如果此前 WDT 被使能的话)。

### 3.2.2 EC 模式

外部时钟模式允许外部产生的逻辑电平作为系统时钟源。工作在此模式下时，外部时钟源连接到 OSC1 输入，OSC2 引脚可用作通用 I/O。

当选取 EC 模式时，振荡器起振定时器 (OST) 被禁止。因此，上电复位 (POR) 后或者从休眠中唤醒后的操作不存在延时。MCU 被唤醒后再次启动外部时钟，器件恢复工作，就好像没有停止过一样。

### 3.2.3 LP 和 XT 模式

LP 和 XT 模式支持连接到 OSC1 和 OSC2 的石英晶体谐振器或陶瓷谐振器的使用。

模式选择内部反相放大器的低或高增益设定，以支持各种谐振器类型及速度。

LP 振荡器模式选择内部反相放大器的最低增益设定。

LP 模式的电流消耗在两种模式中最小。该模式设计仅用于驱动 32.768 kHz 音叉式晶振 (钟表晶振)。

XT 振荡器模式选择内部反相放大器的高增益设定。

## 3.3 内部时钟模式

振荡器模块有两个独立的内部振荡器，可配置或选取为系统时钟源。

1. HFINTOSC (高频内部振荡器) 出厂时已校准，工作频率为 16MHz。
2. LFINTOSC (低频内部振荡器) 未经校准，工作频率为 32 kHz。软件对 OSCCON 寄存器的内部振荡器频率选择位 IRCF<2:0> 进行操作，可选择系统时钟速度。

可通过 OSCCON 寄存器的系统时钟选择 (SCS) 位，在外部或内部时钟源之间选择系统时钟。

**注意：**OSCCON 寄存器的 LFMOD 可以选择 LFINTOSC 是 32KHz 或者 256KHz，但看门狗固定使用 32KHz，不管 LFMOD 为何值。

### 3.3.1 频率选择位 ( IRCF )

16MHz HFINTOSC 和 32KHz LFINTOSC 的输出连接到预分频器和多路复用器 (见图 3.1)。OSCCON 寄存器的内





部振荡器频率选择位 IRCF<2:0>用于选择内部振荡器的频率输出。可通过软件选择以下 8 个频率之一：

- 16MHz
- 8MHz
- 4MHz（复位后的缺省值）
- 2MHz
- 1MHz
- 500 kHz
- 250 kHz
- 32 kHz

### 3.3.2 HFINTOSC 和 LFINTOSC 时钟切换时序

当在 LFINTOSC 和 HFINTOSC 之间切换时，新的振荡器可能为了省电已经关闭（见图 3.2 和图 3.3）。在这种情况下，OSCCON 寄存器的 IRCF 位被修改之后、频率选择生效之前，存在一个延时。OSCCON 寄存器的 LTS 和 HTS 位将反映 LFINTOSC 和 HFINTOSC 振荡器的当前活动状态。频率选择时序如下：

1. OSCCON 寄存器的 IRCF<2:0>位被修改。
2. 如果新时钟是关闭的，开始一个时钟启动延时。
3. 时钟切换电路等待当前时钟下降沿的到来。
4. CLKOUT 保持为低，时钟切换电路等待两个新时钟下降沿的到来。
5. 现在 CLKOUT 连接到新时钟。OSCCON 寄存器的 HTS 和 LTS 位按要求被更新。
6. 时钟切换完成。

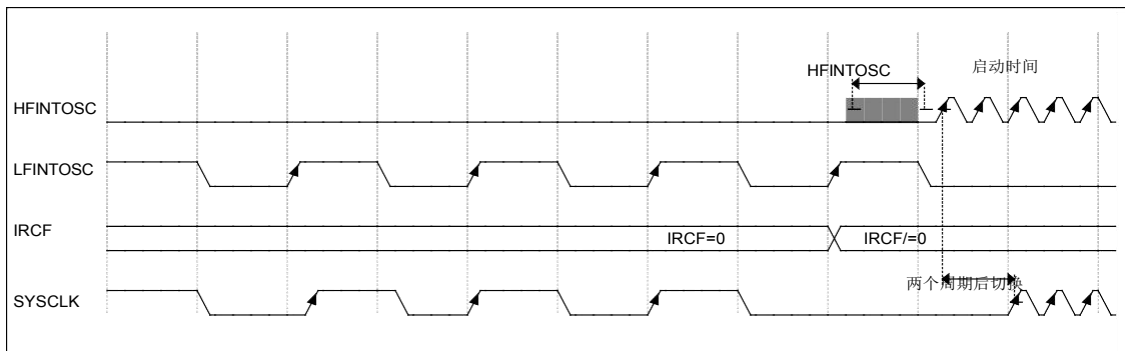


图 3.2 由慢时钟切换到快时钟

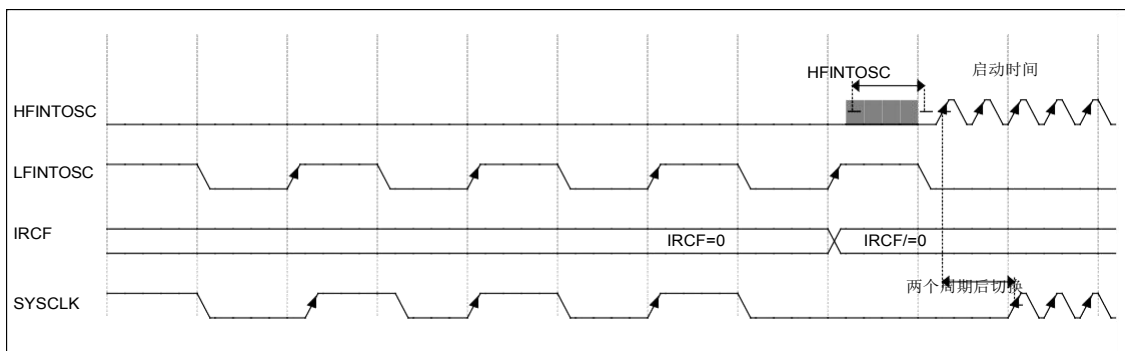


图 3.3 由快时钟切换到慢时钟

## 3.4 时钟切换

通过软件对 OSCCON 寄存器的系统时钟选择 (SCS) 位进行操作, 可将系统时钟源在外部和内部时钟源之间切换。

### 3.4.1 系统时钟选择 (SCS) 位

OSCCON 寄存器的系统时钟选择 (SCS) 位选择用于 CPU 和外设的系统时钟源。

- OSCCON 寄存器的位 SCS = 0 时, 系统时钟源由配置字寄存器 (UCFG0) 中 FOSC<2:0>位的配置决定。
- OSCCON 寄存器的位 SCS = 1 时, 根据 OSCCON 寄存器的 IRCF<2:0>位所选的内部振荡器频率选取系统时钟源。复位后, OSCCON 寄存器的 SCS 总是被清零。

**注:** 任何由硬件引起的时钟切换 (可能产生自双速启动或故障保护时钟监控器) 都不会更新 OSCCON 寄存器的 SCS 位。用户应该监控 OSCCON 寄存器的 OSTS 位以确定当前的系统时钟源。

### 3.4.2 振荡器起振超时状态 (OSTS) 位

OSCCON 寄存器的振荡器起振超时状态 (OSTS) 位用于指示系统时钟是来自外部时钟源, 还是来自内部时钟源。外部时钟源由配置字寄存器 (UCFG0) 的 FOSC<2:0>定义。OSTS 还特别指明在 LP 或 XT 模式下, 振荡器起振定时器 (OST) 是否已超时。

## 3.5 双速时钟启动模式

双速启动模式通过最大限度地缩短外部振荡器起振与代码执行之间的延时, 进一步节省了功耗。对于频繁使用休眠模式的应用, 双速启动模式将在器件唤醒后除去外部振荡器的起振时间, 从而可降低器件的总体功耗。该模式使得应用能够从休眠中唤醒, 将 INTOSC 用作时钟源执行数条指令, 然后再返回休眠状态而无需等待主振荡器的稳定。

**注:** 执行 SLEEP 指令将中止振荡器起振时间, 并使 OSCCON 寄存器的 OSTS 位保持清零。

当振荡器模块配置为 LP 或 XT 模式时, 振荡器起振定时器 (OST) 使能 (见第 3.2.1 节“振荡器起振定时器”)。OST 将暂停程序执行, 直到完成 1024 次振荡计数。双速启动模式在 OST 计数时使用内部振荡器进行工作, 使代码执行的延时最大限度地缩短。当 OST 计数到 1024 且 OSCCON 寄存器的 OSTS 位置 1 时, 程序执行切换至外部振荡器。

### 3.5.1 双速启动模式配置

通过以下设定来配置双速启动模式:

- 配置字寄存器 (UCFG1) 中的位 IESO = 1; 内部/外部切换位 (使能双速启动模式)。
- OSCCON 寄存器的位 SCS = 0。
- 配置字寄存器 (CONFIG) 中的 FOSC<2:0>配置为 LP 或 XT 模式。

在下列操作之后, 进入双速启动模式:



- 上电复位（POR）且上电延时定时器（PWRT）
- 延时结束（使能时）后，或者从休眠状态唤醒。

如果外部时钟振荡器配置为除 LP 或 XT 模式以外的任一模式，那么双速启动将被禁止。这是因为 POR 后或从休眠中退出时，外部时钟振荡器不需要稳定时间。

### 3.5.2 双速启动顺序

1. 从上电复位或休眠中唤醒。
2. 使用内部振荡器以 OSCCON 寄存器的 IRCF<2:0>位设置的频率开始执行指令。
3. OST 使能，计数 1024 个时钟周期。
4. OST 超时，等待内部振荡器下降沿的到来。
5. OSTS 置 1。
6. 系统时钟保持为低，直到新时钟下一个下降沿的到来（LP 或 XT 模式）。
7. 系统时钟切换到外部时钟源。

### 3.6 故障保护时钟监控器

故障保护时钟监控器（FSCM）使得器件在出现外部振荡器故障时仍能继续工作。FSCM 能在振荡器起振延时定时器（OST）到期后的任一时刻检测振荡器故障。FSCM 通过将配置字寄存器（UCFG1）中的 FCMEN 位置 1 来使能。FSCM 可用于所有外部振荡模式（LP、XT 和 EC）。

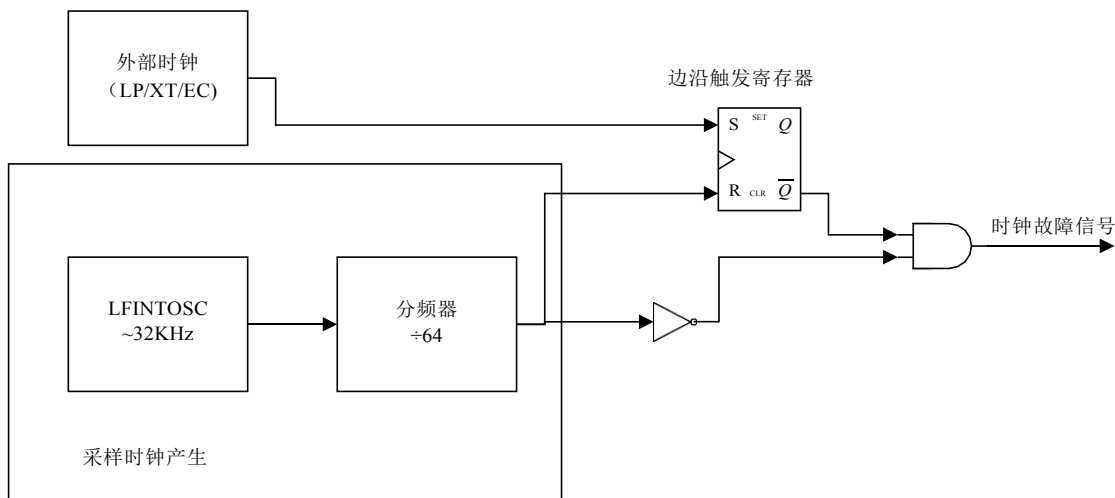


图 3.4 FSCM 原理框图

#### 3.6.1 故障保护检测

FSCM 模块通过将外部振荡器与 FSCM 采样时钟比较来检测振荡器故障。LFINTOSC 除以 64，就产生了采样时钟。请参见图 3.4。故障检测器内部有一个锁存器。在外部时钟的每个下降沿，锁存器被置 1。在采样时钟的每个上升沿，锁存器被清零。如果采样时钟的整个半周期流逝而主时钟依然未进入低电平，就检测到故障。

### 3.6.2 故障保护操作

当外部时钟出现故障时，FSCM 将器件时钟切换到内部时钟源，并将 PIR1 寄存器的 OSFIF 标志位置 1。如果在 PIR1 寄存器的 OSFIE 位置 1 的同时将该标志位置 1，将产生中断。器件固件随后会采取措施减轻可能由故障时钟所产生的问题。系统时钟将继续来自内部时钟源，直到器件固件成功重启外部振荡器并切换回外部操作。

FSCM 所选的内部时钟源由 OSCCON 寄存器的 IRCF<2:0>位决定。这使内部振荡器可以在故障发生前就得以配置。

### 3.6.3 故障保护条件清除

复位、执行 SLEEP 指令或翻转 OSCCON 寄存器的 SCS 位后，故障保护条件被清除。OSCCON 寄存器的 SCS 位被修改后，OST 将重新启动。OST 运行时，器件继续从 OSCCON 中选定的 INTOSC 进行操作。OST 超时后，故障保护条件被清除，器件将从外部时钟源进行操作。必须先清除故障保护条件，才能清零 OSFIF 标志位。

### 3.6.4 复位或从休眠中唤醒

FSCM 设计为能在振荡器起振延时定时器（OST）到期后的任一时刻检测振荡器故障。OST 的使用场合为从休眠状态唤醒后以及任何类型的复位后。OST 不能在 EC 时钟模式下使用，所以一旦复位或唤醒完成，FSCM 就处于激活状态。当 FSCM 被使能时，双速启动也被使能。因此，当 OST 运行时，器件总是处于代码执行阶段。

注：由于振荡器起振时间的范围变化较大，在振荡器起振期间（从复位或休眠中退出时），故障保护电路不处于激活状态。经过一段适当的时间后，用户应检查 OSCCON 寄存器的 OSTS 位，以验证振荡器是否已成功起振以及系统时钟是否切换成功。

## 3.7 与时钟源相关寄存器汇总

名称	地址	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	复位值
UCFG0	0x2000	-	CPB	MCLRE	PWRTEB	WDTE	FOSC2	FOSC1	FOSC0	---- ----
OSCCON	0x8F	LFMOD	IRCF2	IRCF1	IRCF0	OSTS	HTS	LTS	SCS	0101 x000

### 3.7.1 OSCCON 寄存器，地址 0x8F

Bit	7	6	5	4	3	2	1	0
Name	LFOMOD	IRCF[2:0]			OSTS	HTS	LTS	SCS
Reset	0	3' b101			1	0	0	0
TYPE	RW	RW			RO	RO	RO	RW

Bit	Name	Function
7	LFOMOD	低频内振模式： 1 = 256K 振荡频率模式 0 = 32K 振荡频率模式



6:4	IRCF[2:0]	内部振荡器频率选择：		
		值	2T 模式	4T 模式
		111	16MHz	8MHz
		110	8MHz	4MHz
		101	4MHz(default)	2MHz(default)
		100	2MHz	1MHz
		011	1MHz	500kHz
		010	500kHz	250kHz
		001	250kHz	125kHz
000	32kHz(LFINTOSC)	16kHz(LFINTOSC/2)		
3	OSTS	振荡器起振超时状态位 1 = 器件运行在 FOSC<2:0>指定的外部时钟之下 0 = 器件运行在内部振荡器之下		
2	HTS	高速内部时钟状态 1 = HFINTOSC is ready 0 = HFINTOSC is not ready		
1	LTS	低速内部时钟状态 1 = LFINTOSC is ready 0 = LFINTOSC is not ready		
0	SCS	系统时钟选择位 1 = 系统时钟选择为内部振荡器 0 = 时钟源由FOSC<2:0>决定		



## 4 复位时序

WF8M165 有以下几种不同的复位：

- A) 上电复位 POR
- B) WDT(看门狗)复位 - 在常规运行期间
- C) WDT(看门狗)唤醒 - 在睡眠期间
- D) /MCLR 管脚复位 - 在常规运行期间
- E) /MCLR 管脚复位 - 在睡眠期间
- F) 低电压 (BOR/LVR) 复位
- G) 指令错误复位

有些寄存器是不被任何复位影响的；这些寄存器的状态在上电复位时是未知的，也不受复位事件影响。大多数其它寄存器都会在以下复位事件时恢复到其‘复位状态’：

- 上电复位 POR
- WDT(看门狗)复位 - 在常规运行期间
- WDT(看门狗)复位 - 在睡眠期间
- /MCLR 管脚复位 - 在常规运行期间
- 低电压 (BOR) 复位
- 错误指令复位

WDT(看门狗)睡眠唤醒不会造成和在常规运行状态下 WDT(看门狗)超时所造成的复位。因为睡眠唤醒本身就是一种继续的意思，而不是复位/TO 和/PD 位的设置和清零在不同复位条件下的动作是不同的。具体可参考表 4.1 和 4.2。

/MCLR 管脚背后的电路带有防抖功能，能够滤除一些干扰造成的尖细脉冲信号。

下图为复位电路的总体概述框图。

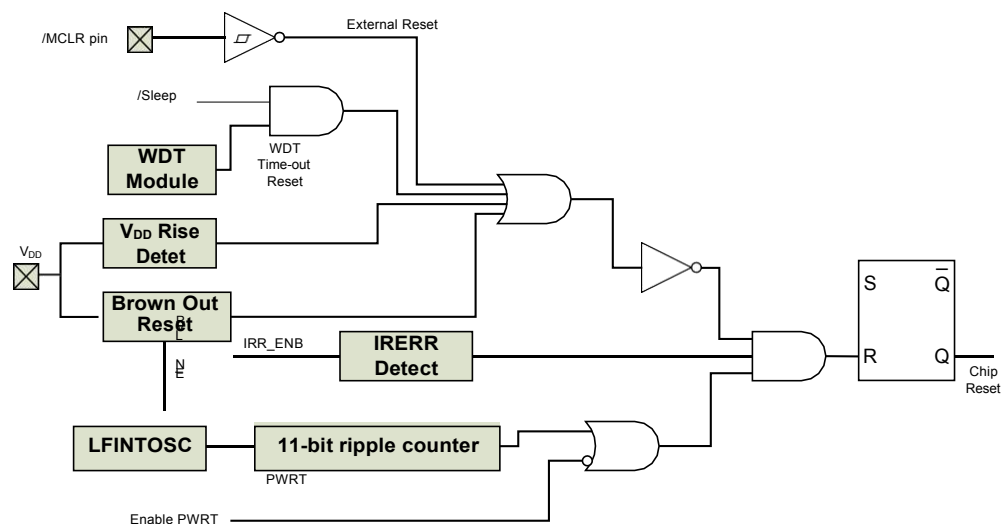


图 4.1 复位功能框图



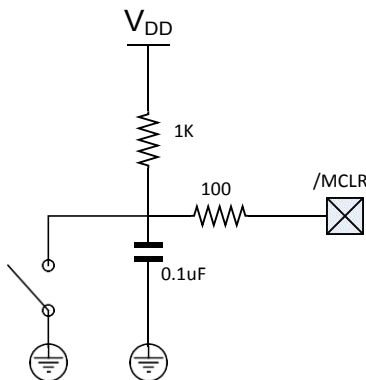
## 4.1 POR 上电复位

片上的 POR 电路会将芯片保持在复位状态直到 VDD 电源电压达到足够高，为充分利用片内复位电路的功能，用户可以简单地直接在 VDD 和/MCLR 之间结一个电阻。这样外部就无需任何 RC 复位电路。不过这要求 VDD 电压上升时间为最大。

上电完成后，系统复位不会立即释放，还要等一个约 4ms 的延时，期间数字电路保持在复位状态。

## 4.2 外部复位 MCLR

需要注意的是，WDT 复位不会把/MCLR 管脚拉低。在/MCLR 管脚上施加超过指标的电压（例如 ESD 事件）会造成/MCLR 复位，而且在管脚上产生超标的大电流，因此我们推荐用户不再直接用一个电阻将/MCLR 和 VDD 连接起来，而是采用以下电路。



在芯片的 CONFIG OPTION 寄存器（UCFG0）中有一个 MCLRE 使能位，将此位清零会使得复位信号由芯片内部产生。当此位为 1 时，芯片的 PA5/MCLR 脚成为外部复位脚。在这个模式下，/MCLR 脚上有个对 VDD 的弱上拉。

## 4.3 PWRT ( 上电计时器 )

PWRT 为上电复位，低电压复位提供一个固定的 64ms（正常情况下）的定时。这个定时器由内部慢时钟驱动。芯片在定时器超时之前都是被保持在复位状态。这段时间能保证 VDD 上升到足够高的电压使得系统能正常工作。PWRT 也可以通过系统 CONFIG 寄存器（UCFG0）来使能。在开启低电压复位功能时，用户应该也打开 PWRT。PWRT 定时是由 VDD 电压超过 VBOR 门限事件启动的。另外需要注意的是，由于由内部慢时钟驱动，定时的实际时间长度是随温度，电压等条件变化而变化的。这个时间不是一个精准参数。

## 4.4 BOR 低电压复位

低电压复位由 UCFG1<1:0>位和 MSCKCON 的 SLVREN 位来控制。低电压复位就是指当电源电压低于 VBOR 门限电压时所产生的复位。不过当 VDD 电压低于 VBOR 不超过 TBOR 时间时，低电压复位可能不会发生。

VBOR 电压在芯片出货之前需要校准，校准可通过串口写入内部校准寄存器来完成。

如果 BOR（低电压复位）是使能（UCFG1<1:0>=00）的，那么最大 VDD 电压上升时间的要求就不存在。BOR 电



路会将芯片控制在复位状态，一直到 VDD 电压达到 VBOR 门限电压以上。  
需要注意的是，当 VDD 低于系统能正常工作的门限时，POR 电路并不会产生复位信号。

当 UCFG1<1:0>=10 时，BOR 电路关闭将由 CPU 的运行状态决定：CPU 正常工作时 BOR 电路工作，CPU 处于睡眠模式时 BOR 电路关闭，这样可以方便的使系统功耗降至更低水平。

## 4.5 LVD 低电压侦测

除了低电压复位功能外，芯片还内置有低电压侦测功能。当电源电压低于设置的电压档位（由 PCON 的 LVDL<2:0> 选择）超过 TBOR（4 到 5 个慢时钟周期）以上时，标志位 LVDW 将会被置 1。软件可以利用此位来监控电源电压。

## 4.6 错误指令复位

当 CPU 的指令寄存器取指到非法指令时，系统将进行复位，利用此功能可增加系统的抗干扰能力。

## 4.7 超时动作

在上电过程中，芯片内部的超时动作顺序按以下流程执行：

POR 结束后启动 PWRT 计时

由于计时是由 POR 脉冲结束后启动的，如果/MCLR 在低电平状态下保持足够长的时间，超时事件就会发生。那么将/MCLR 拉高会让 CPU 立即开始执行。这在测试或者需要多个 MCU 同步的情况下会很有用。

Power Control Register（PCON）

PCON 寄存器里有 2 位指示哪一种复位发生了。Bit0 是/BOR 指示位，其在上电复位是未知态，软件必须将其置 1，然后检查其是否为 0。Bit1 是/POR 指示位，其在上电复位后为 0，软件必须将其置 1。

## 4.8 上电配置过程

发生上电复位或者低电压复位后，除了固有的 4ms 复位延时外，还有一个初始化配置寄存器 UCFGx 的动作。该动作从 PROM 的保留地址读取内容写到 UCFGx，待所有配置地址读取完成后，才可以释放系统复位，如图 4.2 和图 4.3 的所示，该过程大概需要 17us。

## 4.9 上电校验过程

如果 UCFG1.6(CSUMENB)为 0，则上电配置过程结束后，CPU 不会立即执行程序，而是会启动 PROM 的内部校验过程。硬件会把 PROM 主程序区内容读来自加，2K 字全部做完累加后和存储在 0x2007 的值相比较，如果相等说明校验成功，CPU 会从 0 地址执行程序，如图 4.4 所示；如果不相等说明校验失败，CPU 将不会执行程序。校验过程大概需要耗时 1ms。



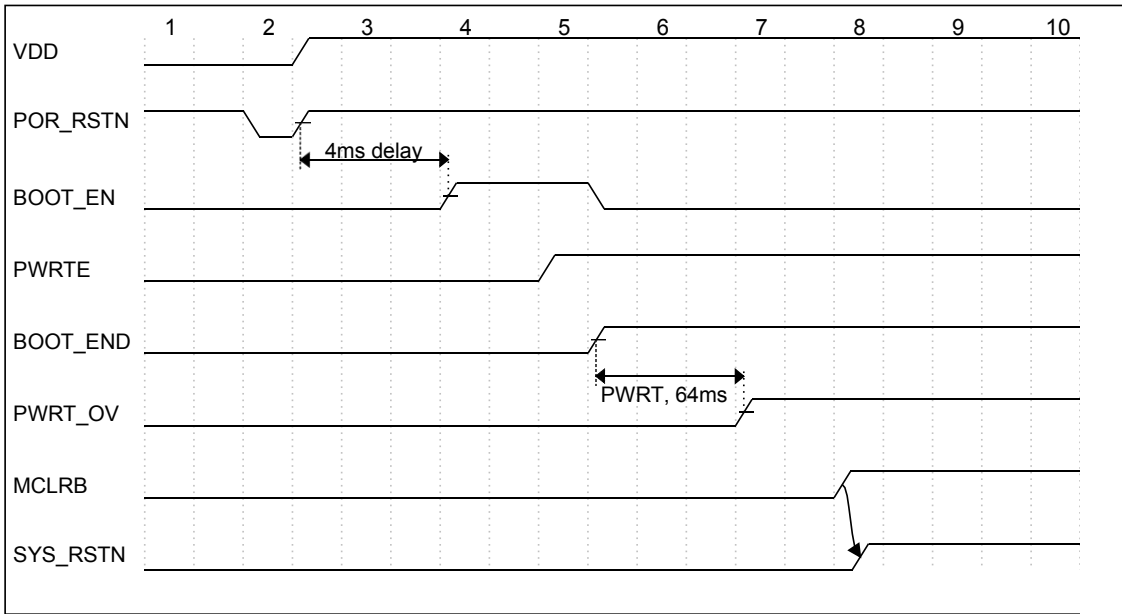


图 4.2 上电复位，使用了 MCLR

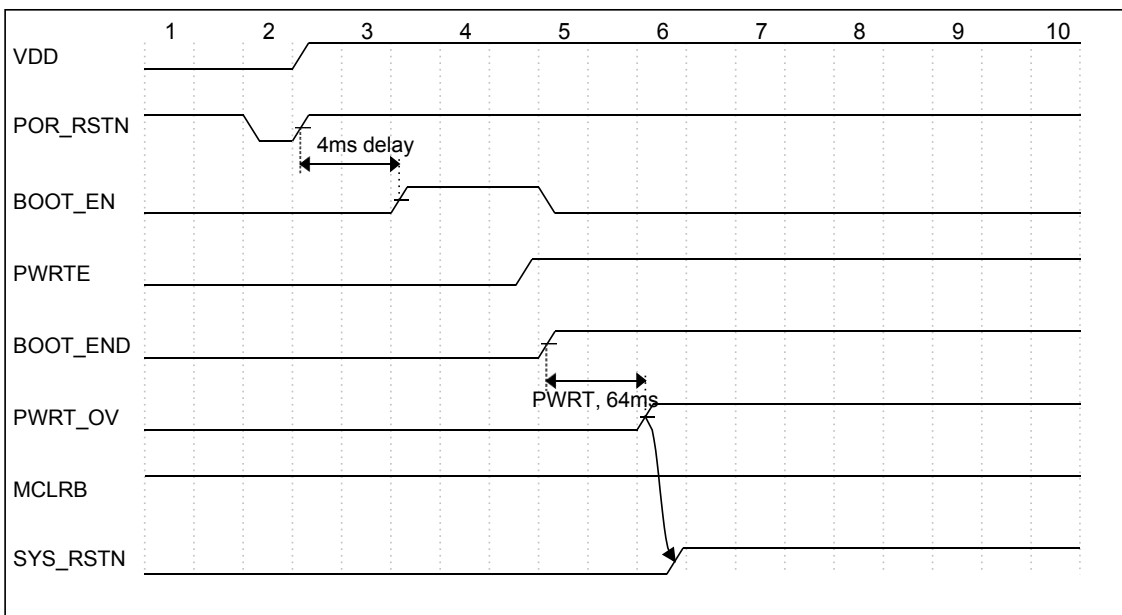


图 4.3 上电复位，没使用 MCLR

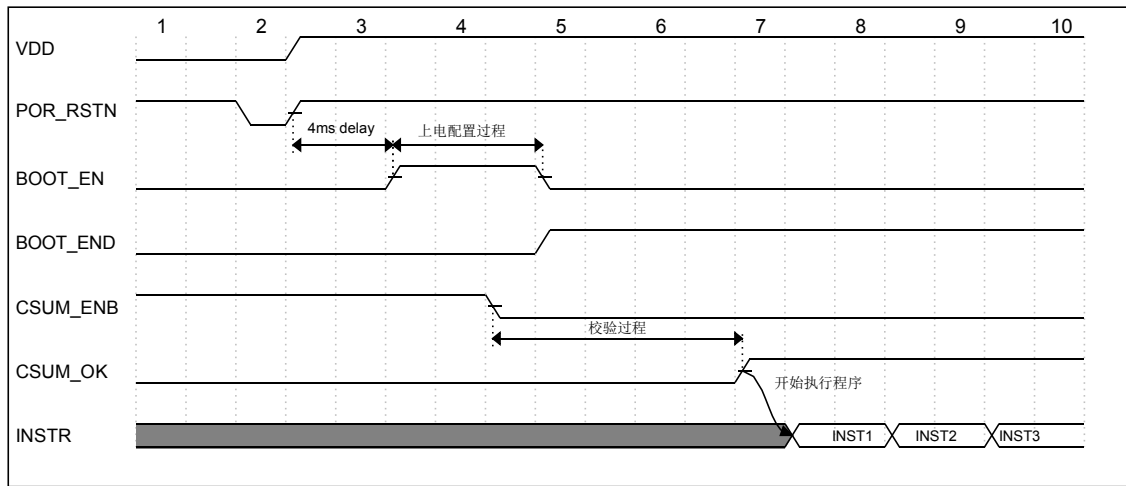


图 4.4 校验过程

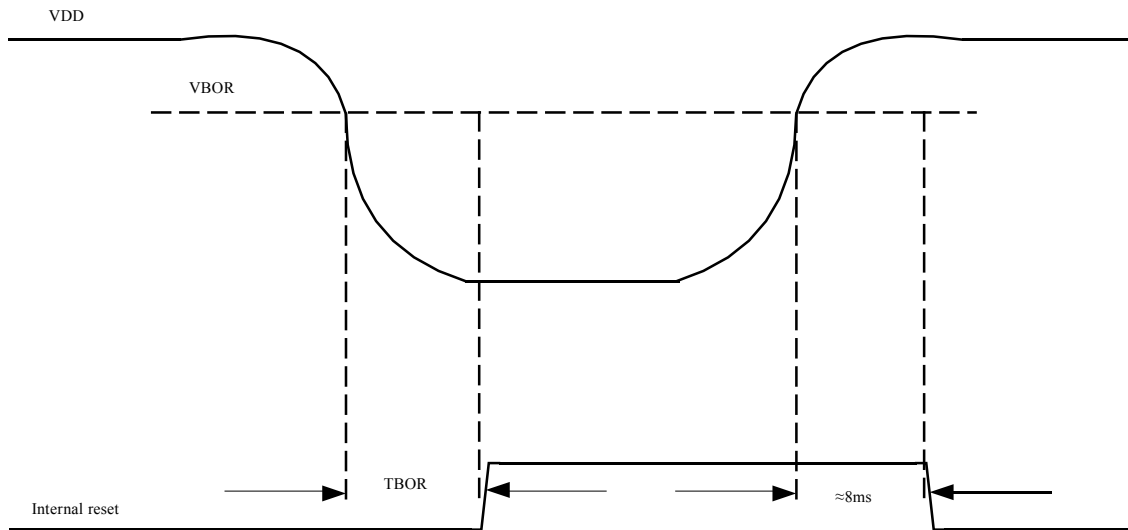


图 4.5 BOR 复位

注意：

1. 上电复位或低电压复位后，并且在 PWRTEB (UCFG0.4) 为低时，PWRT 有效。它是 2048 个内部慢时钟周期，约 64ms；
2. TBOR 时间约为 122~152us；
3. 电压恢复正常之后，内部复位不会立即释放，而是要等约为 4ms 的时间。

振荡器配置	上电复位		低电压复位		睡眠醒来
	/PWRTEB=0	/PWRTEB=1	/PWRTEB=0	/PWRTEB=1	
INTOSC	TPWRT	-	TPWRT	-	-

表 4.1 各种情况下的超时

/POR	/BOR	/TO	/PD	条件
0	x	1	1	POR
u	0	1	1	BOR
u	u	0	u	WDT 复位
u	u	0	0	WDT 唤醒
u	u	u	u	常规运行下/MCLR 复位
u	u	1	0	睡眠下/MCLR 复位

表 4.2 STATUS/PCON 位及其意义 (u-没变化 x-未知)

## 4.10 PCON 寄存器，地址 0x8E

Bit	7	6	5	4	3	2	1	0
Name	-	LVDL[2:0]			LV DEN	LV DW	/POR	/BOR
Reset	0	0	0	0	0	-	q	q
Type	RW	RW	RW	RW	RW	RO	RW	RW

Bit	Name	Function	
7	-	<b>保留位</b>	
6:4	LV DL	低电压侦测选择位， <b>这些位不受低电压复位的影响</b>	
		值	检测电压
		000	1.6V
		001	1.8V
		010	2.0V
		011	2.4V
		100	2.8V
		101	3.0V
		110	3.6V
111	4.2V		
3	LV DEN	低电压侦测使能， <b>该位不受低电压复位的影响</b> 1：开启 LVD 侦测功能 0：关闭 LVD 侦测功能	
2	LV DW	低电压标志位，只读 1：VDD 掉到了 LV DL[2:0]所设置的电压超过 100us 以上 0：VDD 正常，高于 LV DL[2:0]所设置的电压	
1	/POR	上电复位标志，低有效 0：发生了上电复位 1：没发生上电复位或者由软件置 1 /POR 在上电复位后值为 0，此后软件应该将其置 1	
0	/BOR	低电压复位标志，低有效 0：发生了低电压复位 1：没发生低电压复位或者由软件置 1 /BOR 在上电复位后其值不确定，必须由软件置 1。发生后续复位后，通过查询此位来确定是否低电压复位	

## 5 看门狗定时器

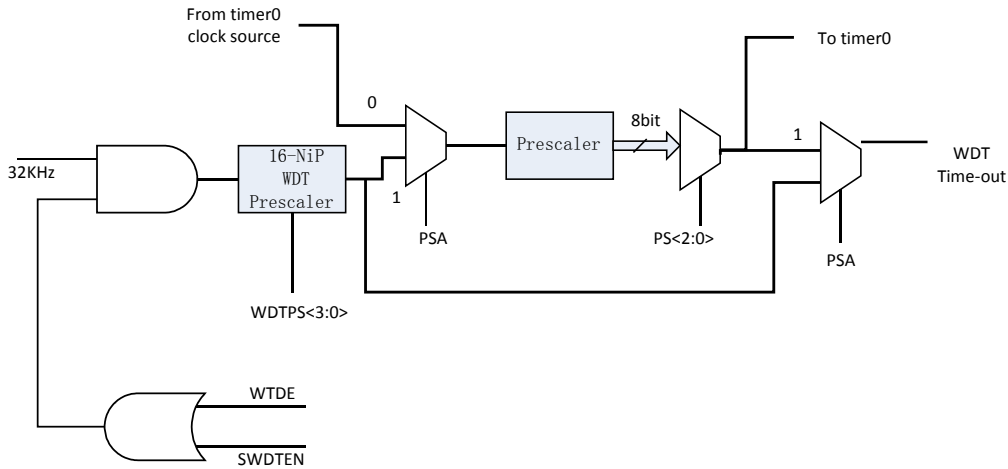


图 5.1 看门狗和定时器 0 框图

### 5.1 看门狗

看门狗的时钟源为内部慢时钟（32KHz），它是一个 16 位的计数器，和定时器 0 共用一个 8 位的预分频器，使能位位于配置寄存器 UCFG0 的第 3 位，WDTEN。为 1 时表示使能看门狗，为 0 时禁止，由上电启动过程 BOOT 决定，或者可通过外部串口写入。

清看门狗指令 CLRWDT、SLEEP 会清除看门狗计数器。

在使能了看门狗的情况下，MCU 睡眠时看门狗溢出事件可以作为一个唤醒源，而 MCU 正常工作时作为一个复位源。

条件	看门狗状态
WDTEN 和 SWDTEN 同时为 0	清零
CLRWDT 指令	
进入 SLEEP、退出 SLEEP 时刻	

注意：

1. 如果内部慢时钟从 32K 切换到 256K 模式(或反之从 256K 切换到 32K 模式)，都不影响看门狗计时，因为 WDT 固定使用 32K 时钟源，见 3.1 小节的时钟框图。

### 5.2 与看门狗相关寄存器汇总

名称	地址	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	复位值
WDTCON	0x18	-	-	-	WDTPS[3:0]				SWDTEN	- - - 0 1 0 0 0



UCFG0	0x2000	-	CPB	MCLRE	PWRTEB	WDTE	FOSC2	FOSC1	FOSC0	---- ----
OPTION	0x81	/PAPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1 1 1 1 1 1 1 1

## 5.2.1 WDTCON 寄存器 · 地址 0x18

Bit	7	6	5	4	3	2	1	0
Name	-	-	-	WDTPS3	WDTPS2	WDTPS1	WDTPS0	SWDTEN
Reset	-	-	-	0	1	0	0	0
Type	-	-	-	RW	RW	RW	RW	RW

Bit	Name	Function
7:5	N/A	-
4:1	WDTPS[3:0]	<p><b>WDTPS&lt;3:0&gt;</b>: 看门狗定时器周期选择：</p> <p>Bit Value = 看门狗定时器驱动时钟之预分频比值</p> <p>0000 = 1:32</p> <p>0001 = 1:64</p> <p>0010 = 1:128</p> <p>0011 = 1:256</p> <p>0100 = 1:512 (复位值)</p> <p>0101 = 1:1024</p> <p>0110 = 1:2048</p> <p>0111 = 1:4096</p> <p>1000 = 1:8192</p> <p>1001 = 1:16384</p> <p>1010 = 1:32768</p> <p>1011 = 1:65536</p> <p>1100 = 1:65536</p> <p>1101 = 1:65536</p> <p>1110 = 1:65536</p> <p>1111 = 1:65536</p>
0	SWDTEN	<p>软件打开/关闭看门狗定时器：</p> <p>1 = 打开</p> <p>0 = 关闭</p>

## 6 定时器 0

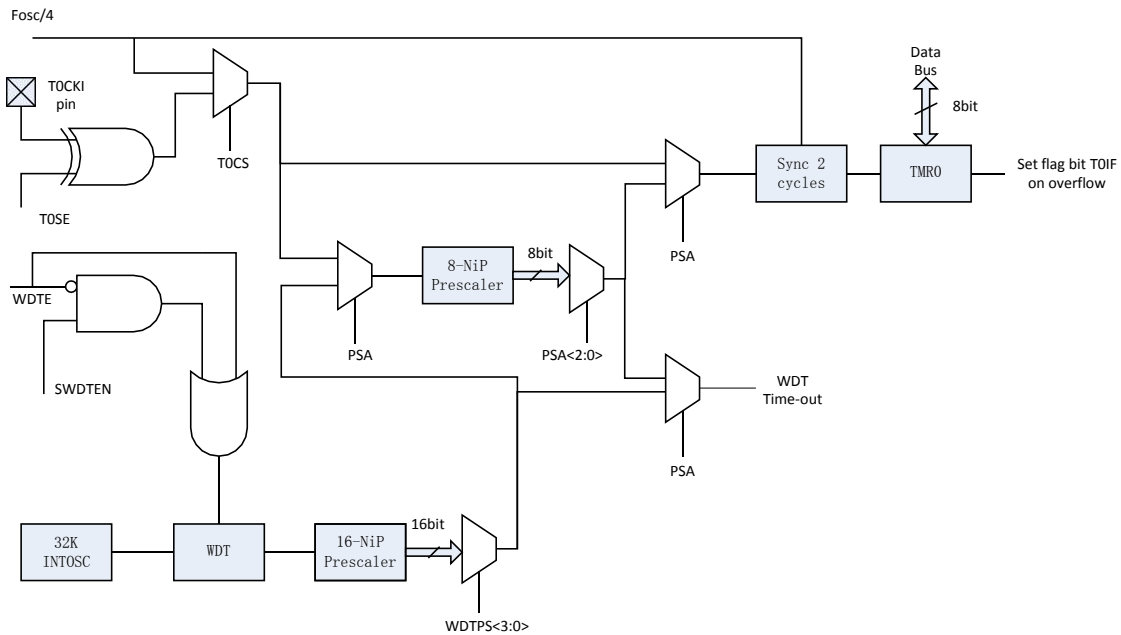


图 6.1 看门狗和定时器 0 框图

### 6.1 Timer 0

定时器 0 为 8 位，可配置为计数器或定时器使用，当作为外部事件（TOCKI）计数器时，可以配置为上升沿或者下降沿计数。作为定时器时，其计数时钟为系统时钟的 2 分频，即每一指令周期递增一次。有一个与 WDT 共用的 8 位预分频器，PSA 为 0 时该预分频器分配给定时器 0 使用。

注意：

1. 当改变 PSA 的值时，硬件会自动把预分频器清 0。

### 6.2 Timer 0 定时器模式

该模式下，定时器 0 在每个指令周期加 1(不带预分频)。软件可以清零 OPTION 寄存器里的 TOCS 位以进入定时器模式。当软件对 TMRO 进行写操作时，定时器在写后面 2 个周期内不会递增。

### 6.3 Timer 0 计数器模式

该模式下，定时器 0 由每个 TOCKI 管脚的上升沿或下降沿触发加 1(不带预分频)。具体哪一种沿触发由 OPTION 寄存器里的 TOSE 位决定。软件可以将 OPTION 寄存器里的 TOCS 位置 1 以进入计数器模式。

## 6.3.1 软件可配置预分频电路

芯片在定时器 0 和 watchdog 定时器前面有一个分频电路，可以分配给 Timer0 或者 watchdog 定时器用，但二者不能同时使用这个预分频电路。具体分配给 Timer0 还是 watchdog 由 OPTION 寄存器里的 PSA 位决定，PSA 为 0 时，预分频分配给 Timer0 使用。在 Timer0 预分频模式下，总共有 8 个预分频比（1:2 到 1:256）可以通过 OPTION 寄存器里的 PS[2:0]位设置。

预分频电路既不可读也不可写。任何对 TMR0 寄存器的写动作会清零预分频电路。

当预分频电路分配给 watchdog 时，1 条 CLRWDT 指令会清零预分频电路。

### 6.3.1.1 在定时器和watchdog 之间切换预分频电路

由于分频电路可以分配给 Timer0 或者 watchdog 定时器用，在二者之间切换预分频器是有可能导致误复位。

在将预分频电路从分配给 TMR0 切换到分配给 watchdog 时，请务必执行以下指令顺序：

```
BANKSEL TMR0  
CLRWDT ;Clear WDT  
CLRR TMR0 ;Clear TMR0 and prescaler  
BANKSEL OPTION_REG  
BSR OPTION_REG,PSA ;Select WDT  
CLRWDT  
  
LDWI b'11111000 ;Mask prescaler bits  
ANDWR OPTION_REG,W  
IORWI b'00000101' ;Set WDT prescaler bits to 1:32  
LDWI OPTION_REG
```

在将预分频电路从分配给 watchdog 切换到分配给 TMR0 时，请务必执行以下指令顺序：

```
CLRWDT ;Clear WDT and prescaler  
BANKSEL OPTION_REG  
LDWI b'11110000' ;Mask TMR0 select and prescaler bits  
ANDWR OPTION_REG,W  
IORWI b'00000011' ;Set prescale to 1:16  
STR OPTION_REG
```

## 6.3.2 定时器 0 中断

芯片在定时器 0 从 0xFF 溢出到 0x00 时会置起 TOIF 标志，并产生中断（如果使能了的话）。注意，timer0 中断无法唤醒 CPU 因为在睡眠状态下，定时器是被冻结的。

### 6.3.3 用外部时钟驱动定时器 0

在计数其模式下，T0CKI 管脚输入和 Timer0 寄存器之间的同步是由在 Q1，Q2 内部时钟相位采样实现的，所以外部时钟源周期的高电平时间和低电平时间必须满足相关时序要求。

## 6.4 与 TIMER0 相关寄存器汇总

名称	地址	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	复位值
TMRO	0x01	TIMER0 计数值寄存器								xxxx xxxx
INTCON	0x0B/8B	GIE	PEIE	TOIE	INTE	PAIE	TOIF	INTF	PAIF	0000 0000
OPTION	0x81	/PAPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111
TRISA	0x85	TRISA[7:0]，PORTA 方向控制								1111 1111

### 6.4.1 OPTION 寄存器，地址 0x81

Bit	7	6	5	4	3	2	1	0
Name	/PAPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
Reset	1	1	1	1	1	1	1	1
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function															
7	/PAPU	<b>PAPU:</b> PORTA Pull-up Enable bit 1 = PORTA pull-ups are disabled 0 = PORTA pull-ups are enabled by individual PORT latch values															
6	INTEDG	<b>INTEDG:</b> Interrupt Edge Select bit 1 = Interrupt on rising edge of PC1/INT pin 0 = Interrupt on falling edge of PC1/INT pin															
5	T0CS	<b>T0CS:</b> Timer0 Clock Source Select bit 1 = Transition on PA2/T0CKI pin 0 = Internal instruction cycle clock															
4	T0SE	<b>T0SE:</b> Timer0 Source Edge Select bit 1 = Increment on high-to-low transition on PA2/T0CKI pin 0 = Increment on low-to-high transition on PA2/T0CKI pin															
3	PSA	<b>PSA:</b> Prescaler Assignment bit 1 = Prescaler is assigned to the WDT 0 = Prescaler is assigned to the Timer0 module															
2	PS2	<b>PS&lt;2:0&gt;:</b> Prescaler Rate Select bits <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>BIT VALUE</th> <th>TIMER0 RATE</th> <th>WDTRATE</th> </tr> </thead> <tbody> <tr> <td>000</td> <td>1 : 2</td> <td>1 : 1</td> </tr> <tr> <td>001</td> <td>1 : 4</td> <td>1 : 2</td> </tr> <tr> <td>010</td> <td>1 : 8</td> <td>1 : 4</td> </tr> <tr> <td>011</td> <td>1 : 16</td> <td>1 : 8</td> </tr> </tbody> </table>	BIT VALUE	TIMER0 RATE	WDTRATE	000	1 : 2	1 : 1	001	1 : 4	1 : 2	010	1 : 8	1 : 4	011	1 : 16	1 : 8
BIT VALUE	TIMER0 RATE	WDTRATE															
000	1 : 2	1 : 1															
001	1 : 4	1 : 2															
010	1 : 8	1 : 4															
011	1 : 16	1 : 8															





		100	1 : 32	1 : 16
		101	1 : 64	1 : 32
		110	1 : 128	1 : 64
		111	1 : 256	1 : 128

## 6.4.2 TMR0 · 地址 0x01

Bit	7	6	5	4	3	2	1	0
Name	TMR0[7:0]							
Reset	xxxx xxxx							
Type	RW							

Bit	Name	Function
7:0	TMR0[7:0]	Timer 0 计数结果寄存器

## 7 带门控的TIMER1 模块

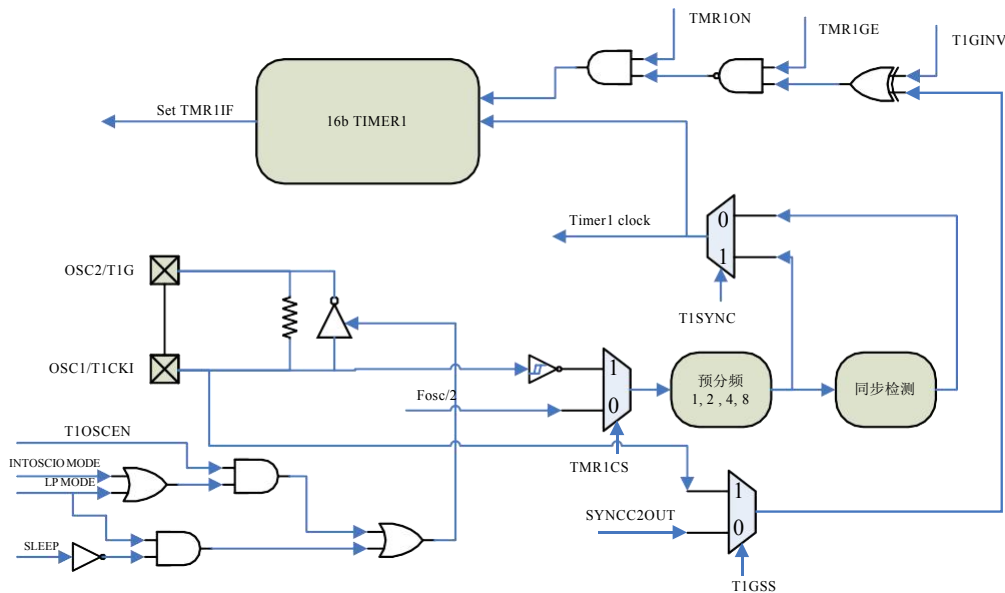


图 7.1 TIMER1 原理框图

TIMER1 是一个 16 位的定时器、计数器，有以下特性：

- 一对 16 位定时器/计数器寄存器 (TMR1H:TMR1L)
- 可编程内部或外部时钟源
- 3 位预分频器
- 可选 LP 振荡器
- 同步或异步操作
- 通过比较器或 T1G 引脚的 Timer1 门控 (计数使能)
- 溢出中断
- 溢出时唤醒 (仅限外部时钟且异步模式时)
- 捕捉/比较功能的时基
- 特殊事件触发 (带 ECCP)
- 比较器输出与 Timer1 时钟同步

### 7.1 Timer1 工作原理

Timer1 模块是 16 位递增计数器，通过一对寄存器 TMR1H:TMR1L 访问。对 TMR1H 或 TMR1L 的写操作将直接更新计数器。

与内部时钟源配合使用时，该模块为定时器。与外部时钟源配合使用时，该模块可用作定时器或计数器。

### 7.2 选择时钟源

T1CON 寄存器的 TMR1CS 位用于选择时钟源。当 TMR1CS=0 时，时钟源为  $F_{osc}/2$ ，即指令时钟 (2T 模式下)。

当 TMR1CS=1 时，时钟源由外部提供（T1CKI 管脚）。

## 7.2.1 内部时钟源

选择内部时钟源时，TMR1H:TMR1L 这对寄存器将在 FOSC 的某个倍数递增，倍数由 Timer1 预分频器决定。

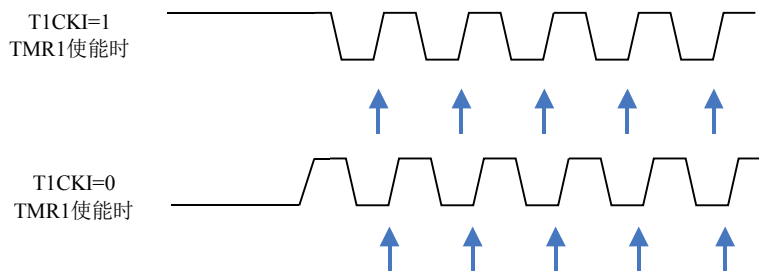
## 7.2.2 外部时钟源

选择外部时钟源时，Timer1 模块可作为定时器，也可作为计数器工作。计数时，Timer1 在外部时钟输入 T1CKI 的上升沿递增。此外，计数器模式时钟可同步到单片机系统时钟，也可异步运行。

若需外部时钟振荡器（且单片机使用不带 CLKOUT 的 INTOSC），则 Timer1 可使用 LP 振荡器作为时钟源。

**注意**，在计数器模式下，发生以下任何一个或多个情况后，计数器在首个上升沿递增前，必须先经过一个下降沿：

- 在计数器模式下，发生以下任何一个或多个情况后，计数器在首个上升沿递增前，必须先经过一个下降沿：
- POR 复位后使能 Timer1
- 写入 TMR1H 或 TMR1L
- Timer1 被禁止
- T1CKI 为高电平时 Timer1 被禁止（TMR1ON=0），然后在 T1CKI 为低电平时 Timer1 被使能（TMR1ON=1）。



注意：

1. 箭头所指边沿为计数器递增；
2. 计数器模式下，计数器递增之前必须先经过一个下降沿。

图 7.2 TIMER1 边沿递增示意

## 7.3 Timer1 预分频器

Timer1 有四种预分频选项，可对时钟输入进行 1、2、4 或 8 分频。T1CON 寄存器的 T1CKPS 位控制预分频计数器。预分频计数器不可直接读写；但当发生对 TMR1H 或 TMR1L 的写操作时，或者 TIMER1 被关闭时（TMR1ON 为 0），预分频计数器被清零。

## 7.4 Timer1 振荡器

OSC1（输入）引脚与 OSC2（放大器输出）引脚之间内置有一个低功耗 32.768kHz 晶振。将 T1CON 寄存器的 T1OSCEN 控制位置 1 使能振荡器。休眠时振荡器继续工作。

Timer1 振荡器与系统 LP 振荡器共用。这样，Timer1 就只能在主系统时钟来自内部振荡器或振荡器处于 LP 模式时使用此模式。用户必须提供软件延时以确保振荡器正常起振。

Timer1 振荡器被使能时，PORTA[7]、PORTA[6]的输出驱动被禁止，且 PA7 和 PA6 位读为 0。但 TRISA7、TRISA6 保持原来的值。

**注意：**

1. 振荡器在使用前需要一段起振和稳定时间。所以，应将 T1OSCEN 置 1 并在使能 Timer1 之前应有一段适当的延时；
2. 配置为振荡器模式时，T1G 固定输出 1，固不能用它来门控 TIMER1。

## 7.5 Timer1 工作于异步计数器模式下

若 T1CON 寄存器的控制位 T1SYNC 置 1，则外部时钟输入不同步。定时器与内部相位时钟异步递增。若选择了外部时钟源，则定时器将在休眠时继续运行，并可在溢出时产生中断，唤醒处理器。然而，读写定时器时应特别谨慎（见第 7.5.1 节“在异步计数器模式下读写 Timer1”）。

**注意：**

1. 从同步操作切换为异步操作时，有可能错过一次递增从异步操作切换为同步操作时，有可能多产生一次递增。

### 7.5.1 在异步计数器模式下读写TIMER1

定时器运行于外部异步时钟时，读取 TMR1H 或 TMR1L 将确保读操作有效（由硬件负责）。然而，应该注意的是，用两个 8 位值来读取 16 位定时器本身就会产生某些问题，这是因为定时器可能在两次读操作之间产生溢出。

对于写操作，建议用户直接停止计数器，然后写入所期望的值。如果寄存器正进行递增计数，对定时器寄存器进行写操作，可能会导致写入竞争，从而可能在 TMR1H:TMR1L 这对寄存器中产生不可预测的值。

## 7.6 TIMER1 门控

Timer1 门控源可软件配置为 T1G 引脚或比较器 C2 的输出。这使器件可直接使用 T1G 为外部事件定时，或使用比较器 C2 对模拟事件定时。Timer1 门控源的选择请参见 CMCON1 寄存器。这个特性可以简化  $\Delta$ - $\Sigma$  A/D 转换器和许多其他应用的程序。

使用 T1CON 寄存器的 T1GINV 位可翻转 Timer1 门控，不论其来自 T1G 引脚还是比较器 C2 的输出。这将配置 Timer1 以确保事件之间存在低电平有效或高电平有效的有效时间。

## 7.7 Timer1 中断

Timer1 的一对寄存器（TMR1H:TMR1L）递增至 FFFFh 后返回 0000h。Timer1 计满返回时，PIR1 寄存器的 Timer1



中断标志位被置 1。为确保计满返回时产生中断，必须将以下位置 1：

- T1CON 寄存器的 TMR1ON 位
- PIE1 寄存器的 TMR1IE 位
- INTCON 寄存器的 PEIE 位
- INTCON 寄存器的 GIE 位

在中断服务程序中将 TMR1IF 位清零将清除中断。

## 7.8 Timer1 在休眠模式下的工作原理

只有在设定异步计数器模式时，Timer1 才能在休眠模式下工作。在该模式下，可使用外部晶振或时钟源信号使计数器递增。要做以下设置定时器以唤醒器件：

- 必须将 T1CON 寄存器的 TMR1ON 位置 1
- 必须将 PIE1 寄存器的 TMR1IE 位置 1
- 必须将 INTCON 寄存器的 PEIE 位置 1
- 必须将 T1CON 寄存器的 T1SYNC 位置 1
- 必须将 T1CON 寄存器的 TMR1CS 位置 1
- 可将 T1CON 寄存器的 T1OSCEN 位置 1

溢出时器件将被唤醒并执行下一条指令。若 INTCON 寄存器的 GIE 位置 1，器件将调用中断服务程序（0004h）。

## 7.9 ECCP 捕捉/比较时基

工作于捕捉或比较模式时，ECCP 模块使用一对 TMR1H:TMR1L 寄存器作为时基。在捕捉模式下，TMR1H:TMR1L 这对寄存器的值在发生某个配置好的事件时被复制到 CCPR1H:CCPR1L 这对寄存器中。

在比较模式下，当 CCPR1H:CCPR1L 这对寄存器的值与 TMR1H:TMR1L 的值匹配时，将触发一个事件。该事件可以是特殊事件触发。

更多信息请参见第 9 节“增强型捕捉/比较/PWM+（带自动关闭和死区）模块”。

## 7.10 ECCP 特殊事件触发器

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。该特殊事件不会产生 Timer1 中断。ECCP 模块仍可配置为产生 ECCP 中断。在此工作模式下，CCPR1H:CCPR1L 这对寄存器成为了 Timer1 的周期寄存器。

- 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

更多信息请参见第 9.2.4 节“特殊事件触发器”。



## 7.11 与 TIMER1 相关寄存器汇总

名称	地址	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	复位值
TMR1L	0x0E	16 位 TIMER1 计数值低 8 位								XXXX XXXX
TMR1H	0x0F	16 位 TIMER1 计数值高 8 位								XXXX XXXX
INTCON	0x0B/8B	GIE	PEIE	TOIE	INTE	PAIE	TOIF	INTF	PAIF	0000 0000
PIE1	0x8C	EEIE	CKMEAIE	-	C2IE	C1IE	OSFIE	TMR2IE	TMR1IE	0000 0000
PIR1	0x0C	EEIF	CKMEAIF	-	C2IF	C1IF	OSFIF	TMR2IF	TMR1IF	0000 0000
T1CON	0x10	T1GINV	TMR1GE	T1CKPS[1:0]		T1OSCEN	T1SYNC	TMR1CS	TMR1ON	0000 0000
CMCON1	0x1A	-	-	-	-	-	-	T1GSS	C2SYNC	---- --10

### 7.11.1 T1CON 寄存器 · 地址 0x10

Bit	7	6	5	4	3	2	1	0
Name	T1GINV	TMR1GE	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7	T1GINV	TIMER1 门控翻转位 1=Timer1 门控为高电平有效（门控为高电平时 Timer1 计数） 0=Timer1 门控为低电平有效（门控为低电平时 Timer1 计数）
6	TMR1GE	Timer1 门控使能位(2) 如果 TMR1ON = 0： 此位被忽略 如果 TMR1ON = 1： 1 = Timer1 在 Timer1 门控不活动时打开 0 = Timer1 打开
5:4	T1CKPS[1:0]	Timer1 输入时钟预分频比选择位 11=1:8 预分频比 10=1:4 预分频比 01=1:2 预分频比 00=1:1 预分频比
3	T1OSCEN	LP 振荡器使能控制位 如果无 CLKOUT 振荡器的 INTOSC 处于激活状态： 1 = LP 振荡器被使能用于 Timer1 时钟 0 = LP 振荡器关闭 否则： 此位被忽略
2	T1SYNC	Timer1 外部时钟输入同步控制位 TMR1CS = 1：



		1 = 不同步外部时钟输入 0 = 同步外部时钟输入 TMR1CS = 0 : 此位被忽略。Timer1 使用内部时钟
1	TMR1CS	Timer1 时钟源选择位 1 = 来自 T1CK1 引脚（上升沿）的外部时钟 0 = 内部时钟（指令时钟）
0	TMR1ON	Timer1 使能位 1 = 使能 Timer1 0 = 停止 Timer1

### 7.11.2 TMR1L 寄存器，地址 0x0E

Bit	7	6	5	4	3	2	1	0
Name	TMR1L[7:0]							
Reset	x	x	x	x	x	x	x	x
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:0	TMR1L[7:0]	16 位 TIMER1 定时器、计数器计数值的低 8 位

### 7.11.3 TMR1H 寄存器，地址 0x0F

Bit	7	6	5	4	3	2	1	0
Name	TMR1H[7:0]							
Reset	x	x	x	x	x	x	x	x
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:0	TMR1H[7:0]	16 位 TIMER1 定时器、计数器计数值的高 8 位

## 8 定时器 2

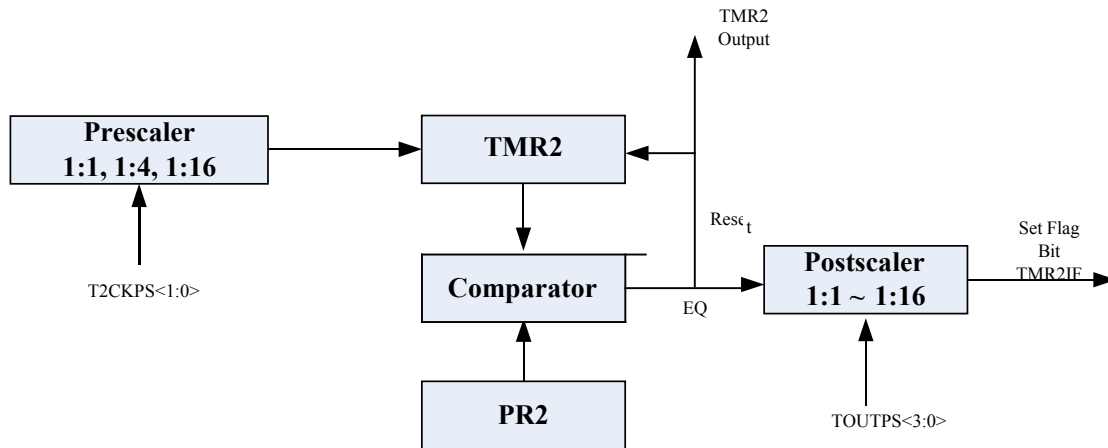


图 8.1 定时器 2 框图

定时器 2 为 8 位定时器包含以下功能：

- 8 位计数寄存器
- 8 位周期寄存器
- TMR2 值等同 PR2 时产生中断
- 1:1, 1:4, 1:16 预分频比
- 1:1~1:16 后分频比
- 时钟源可选系统时钟或者内部 32MHz 时钟（由 HFINTOSC 的 2 倍频得到）

图 8.1 为 Timer2 的整体框图。

### 8.1 Timer2 工作原理

Timer2 模块的时钟输入是系统指令时钟。该时钟送入 Timer2 预分频器，其预分频比有 1:1、1:4 或 1:16 三种选择。随后预分频器的输出被用于递增 TMR2 寄存器。

TMR2 和 PR2 的值被不断比较以确定何时匹配。TMR2 将从 00h 开始递增直到与 PR2 的值相同。匹配时将发生以下两种情况：

- TMR2 在下一递增周期复位为 00h
- Timer2 后分频比递增

Timer2/PR2 比较器的匹配输出送入 Timer2 后分频器。后分频器的选项范围为 1:1 至 1:16。Timer2 后分频器的输出用于将 PIR1 寄存器的 TMR2IF 中断标志置 1。

TMR2 和 PR2 都是可读写寄存器。在复位时，他们的值分别是 0 和 0xFF。

将 T2CON 寄存器中的 TMR2ON 位置 1 可打开 Timer2，反之将 TMR2ON 位清零关闭 Timer2。



Timer2 预分频器由 T2CON 寄存器的 T2CKPS 位控制。

Timer2 后分频器由 T2CON 寄存器的 TOUTPS 位控制。

预分频和后分频计数器会在写以下寄存器时清零：

- 写 TMR2
- 写 T2CON
- 任何 reset 动作

注：

1. 写 T2CON 并不会清零 TMR2 寄存器；
2. TIMER2 的时钟源由 MSCKCON.5 控制，当 T2CKSRC 为 1 时表示选择内部 32MHz 时钟，与当前运行的系统时钟无关。32MHz 时钟是由内部 HFINTOSC 倍频得到，所以一旦 TIMER2 选择 32MHz 时钟源并且 TMR2ON=1 时，即使系统时钟选择的是内部慢时钟或者外部晶体时钟，HFINTOSC 是不会关闭的，除非进入了睡眠模式。

## 8.2 与 TIMER2 相关寄存器汇总

名称	地址	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	复位值	
TMR2	0x11	TIMER2 计数值寄存器								xxxxxxx	
INTCON	0x0B/8B	GIE	PEIE	TOIE	INTE	PAIE	T0IF	INTF	PAIF	00000000	
PIE1	0x8C	EEIE	CKMEAIE	-	C2IE	C1IE	OSFIE	TMR2IE	TMR1IE	00000000	
PIR1	0x0C	EEIF	CKMEAIF	-	C2IF	C1IF	OSFIF	TMR2IF	TMR1IF	00000000	
MSCKCON	0x1B	-	VREG_OE	T2CKSRC	SLVREN	-	CKMAVG	CKCNT1	-	0000-00-	
PR2	0x92	TIMER2 周期寄存器								11111111	
T2CON	0x12	-	TOUTPS[3:0]				-	TMR2ON	T2CKPS[1:0]		-0000000

### 8.2.1 PR2 寄存器 · 地址 0x92

Bit	7	6	5	4	3	2	1	0
Name	PR2[7:0]							
Reset	1	1	1	1	1	1	1	1
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:0	PR2[7:0]	Timer 2 周期(比较)寄存器 (详见 Timer2 描述章节)

## 8.2.2 TMR2 寄存器，地址 0x11

Bit	7	6	5	4	3	2	1	0
Name	TMR2[7:0]							
Reset	0000 0000							
Type	RW							

Bit	Name	Function
7:0	TMR2[7:0]	Timer 2 计数结果寄存器

## 8.2.3 T2CON 寄存器，地址 0x12

Bit	7	6	5	4	3	2	1	0
Name	-	TOUTPS[3:0]				TMR2ON	T2CKPS[1:0]	
Reset	-	0000				0	00	
Type	-	RW				RW	RW	

Bit	Name	Function
7	-	未实现，读 0
6:3	TOUTPS[3:0]	<b>TOUTPS&lt;3:0&gt;</b> : Timer2 Output Postscaler Select bits 定时器2输出后分频比选择 0000 = 1:1 后分频比 0001 = 1:2 后分频比 0010 = 1:3 后分频比 0011 = 1:4 后分频比 0100 = 1:5 后分频比 0101 = 1:6 后分频比 0110 = 1:7 后分频比 0111 = 1:8 后分频比 1000 = 1:9 后分频比 1001 = 1:10 后分频比 1010 = 1:11 后分频比 1011 = 1:12 后分频比 1100 = 1:13 后分频比 1101 = 1:14 后分频比 1110 = 1:15 后分频比 1111 = 1:16 后分频比
2	TMR2ON	<b>TMR2ON</b> : Timer2 On bit 打开定时器2 1 = Timer2打开 0 = Timer2 关闭
1:0	T2CKPS[1:0]	<b>T2CKPS&lt;1:0&gt;</b> : Timer2 Clock Prescale Select bits 定时器2驱动时钟预分频比选择



		00 = Prescaler is 1 01 = Prescaler is 4 1x = Prescaler is 16
--	--	--

## 9 增强型捕捉/比较/PWM 模块

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 9.1 ECCP 各模式所需的定时器资源

### 9.1 捕捉模式

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 9.1）。

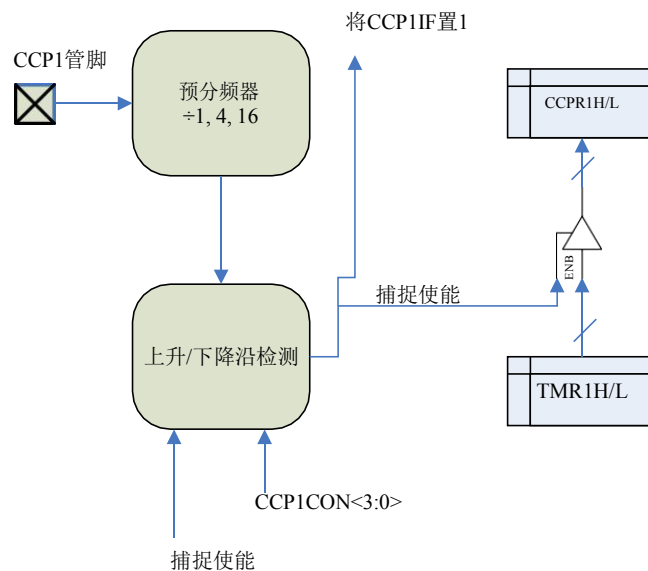


图 9.1 捕捉模式原理框图

### 9.1.1 CCP1 引脚配置

在捕捉模式下，应该通过将相关的 TRISC 控制位置 1 将 CCP1 引脚配置为输入。

**注意：**如果 CCP1 引脚配置为输出，则写端口将产生一次捕捉条件。

### 9.1.2 TIMER1 模式选择

为使 CCP 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。

### 9.1.3 软件中断

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 PIE1 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。

### 9.1.4 CCP 预分频器

CCP1CON 寄存器的 CCP1M<3:0>位指定了 4 种不同的预分频比。每当关闭 CCP 模块，或 CCP 模块不在捕捉模式时，预分频计数器都将被清零。任何复位都会将预分频计数器清零。

从一个捕捉预分频比切换到另一个不会清零预分频器并可能产生一次中断。要避免意外操作，在改变预分频比前应将 CCP1CON 寄存器清零，以关闭 CCP 模块（见例 9.1）。

```
BANKSEL CCP1CON          ;Set Bank bits to point to CCP1CON  
CLRR CCP1CON             ;Turn CCP module off  
LDW1NEW_CAPT_PS         ;Load the W reg with the new prescaler  
STR CCP1CON              ;Load CCP1CON with this value
```

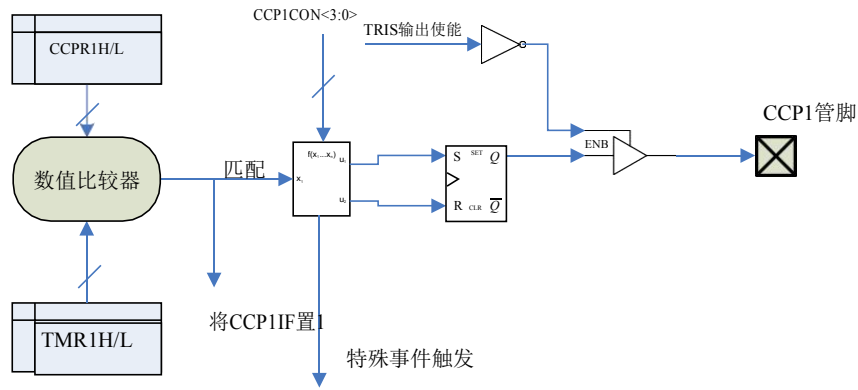
例 9.1 切换预分频比

## 9.2 比较模式

在比较模式下，16 位 CCPR1 寄存器值被不断与 TMR1 的一对寄存器的值相比较。发生匹配时，CCP 模块可能：

- 翻转 CCP1 输出
- 将 CCP1 输出置 1
- 将 CCP1 输出清零
- 触发特殊事件
- 产生软件中断

引脚上的动作取决于 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。



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 9.2 比较模式原理框图

### 9.2.1 CCP1 引脚的配置

用户必须通过清零相关的 TRIS 位将 CCP1 引脚配置为输出。

### 9.2.2 TIMER1 模式选择

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。比较操作在异步计数器模式下可能不能正常工作。

### 9.2.3 软件中断模式

当选择产生软件中断模式 ( $CCP1M\langle 3:0 \rangle = 1010$ ) 时, CCP 模块并不得到 CCP1 引脚的控制权 (见 CCP1CON 寄存器)。

### 9.2.4 特殊事件触发器

当选定了特殊事件触发模式 ( $CCP1M\langle 3:0 \rangle = 1011$ ) 时, CCP 模式将会:

- 复位 Timer1
- 若 ADC 使能, 则启动一次 ADC 转换

在此模式下, CCP 模块不得到 CCP1 引脚的控制权 (见 CCP1CON 寄存器)。

一旦 TMR1H:TMR1L 这对寄存器和 CCPR1H:CCPR1L 这对寄存器之间发生匹配, 便会发生 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。这使 CCPR1H:CCPR1L 寄存器可作为 Timer1 的 16 位可编程周期寄存器。

注意：

1. CCP 模块的特殊事件触发不会将 PIR1 寄存器的中断标志位 TMR1IF 置 1；
2. 在生成特殊事件触发的时钟边沿和复位 Timer1 的时钟边沿之间，通过更改 CCPR1H 和 CCPR1L 这对寄存器的内容来移除匹配条件，可以预防复位的发生。

### 9.3 PWM 模式

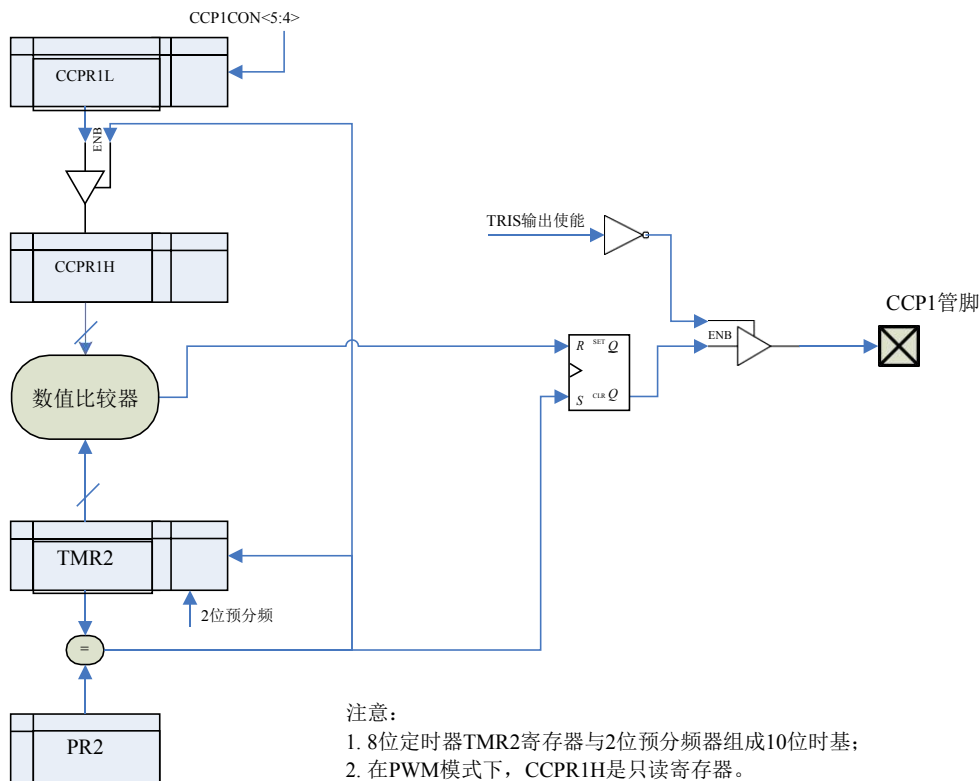


图 9.3 PWM 原理框图

PWM 模式将在 CCP1 引脚产生脉宽调制信号。其占空比，周期和分辨率由以下寄存器决定：

- PR2
- T2CON
- CCPR1L
- CCP1CON

在脉宽调制（Pulse-Width-Modulation，PWM）模式下，CCP 模块在 CCP1 引脚上产生高达 10 位分辨率的 PWM 输出。由于 CCP1 引脚与 PORT 数据锁存器复用，必须将该引脚的 TRIS 清零以使能 CCP1 引脚输出驱动器。

注意：

1. 将 CCP1CON 寄存器清零将放弃对 CCP1 引脚的控制；
2. 由于慢时钟测量也占用 TIMER2，所以想使用 PWM 功能时不能启动慢时钟测量。

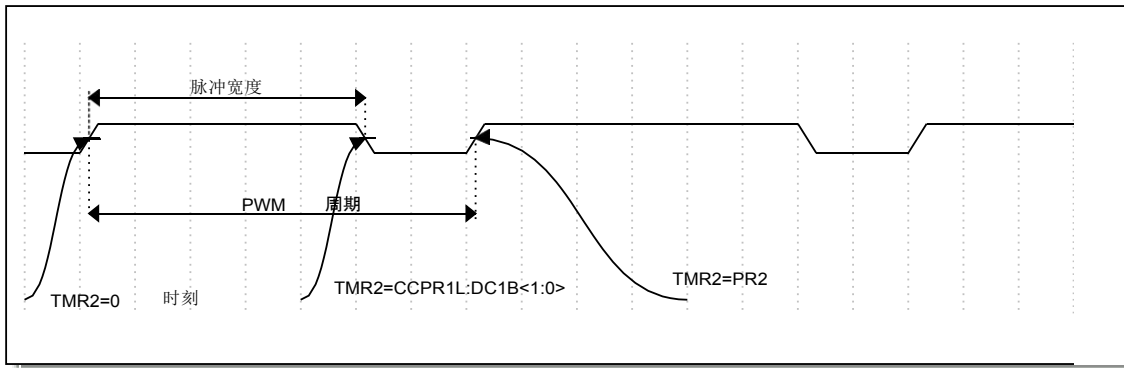


图 9.4 PWM 输出波形

### 9.3.1 PWM 周期

PWM 周期由 Timer2 的 PR2 寄存器指定。用公式 11.1 可计算 PWM 周期。

PWM 周期 =  $(PR2 + 1) * 4 * T_{sys} * (TMR2 \text{ 预分频值})$

公式 9.1

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. 确定 PWM 频率时不使用 Timer2 的后分频器。见第 8 节的“定时器 2”相关工作原理描述；
3. 当 MSCKCON.5 为 1 时，Timer2 的时钟源为 32MHz。

### 9.3.2 PWM 占空比

通过对以下几个寄存器写入 10 位值可指定 PWM 占空比：CCPR1L 寄存器和 CCP1CON 寄存器的 DC1B<1:0>位。CCPR1L 包含八个 MSb，CCP1CON 寄存器的 DC1B<1:0>位包含两个 LSb。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 寄存器是只读的。

公式 9.2 用于计算 PWM 脉宽。

公式 9.3 用于计算 PWM 占空比。

脉冲宽度 =  $(CCPR1L:CCP1CON<5:4>) * T_{sys} * (TMR2 \text{ 预分频值})$

公式 9.2

占空比 =  $(CCPR1L:CCP1CON<5:4>) / (4 * (PR2 + 1))$

公式 9.3

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。如果 Timer2 预分频器置为 1:1，则使用系统时钟。

当 10 位时基与 CCP1H 及 2 位的锁存器匹配时，CCP1 引脚被清零（见图 9.3）。

### 9.3.3 PWM 分辨率

分辨率决定某个周期的有效占空比。例如，10 位分辨率有 1024 个分立的占空比，而 8 位分辨率则有 256 个分立的占空比。

PR2 为 255 时产生 10 位的最大 PWM 分辨率。分辨率是 PR2 寄存器值的函数，如公式 9.4 所示。

$$\text{分辨率} = \log[4(\text{PR2}+1)] \div \log(2) \text{ 位} \quad \text{公式 9.4}$$

**注意：**如果脉冲宽度大于周期，那么分配的 PWM 引脚将保持不变。

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

表 9.1 PWM 频率和分辨率示例（Fosc=20MHz）

### 9.3.4 休眠模式下的工作

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。

### 9.3.5 系统时钟频率的改变

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。

**注意：**

1. 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。

### 9.3.6 复位的影响

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。

### 9.3.7 PWM 工作的设置

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 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

#### 9.4 PWM (增强模式)

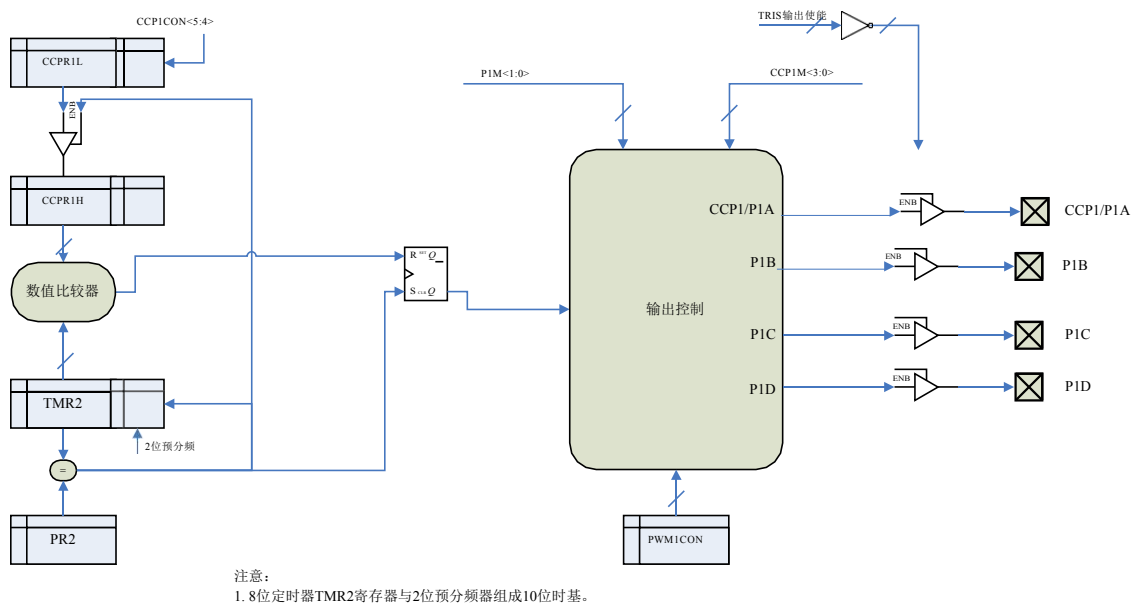


图 9.5 增强型 PWM 模式原理框图

增强型 PWM 模式可在最多四个输出引脚上产生高达 10 位分辨率的 PWM 信号。它通过四种 PWM 输出模式做到：

- 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 模式所不使用的任何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 9.2 不同 PWM 增强模式的引脚分配示例

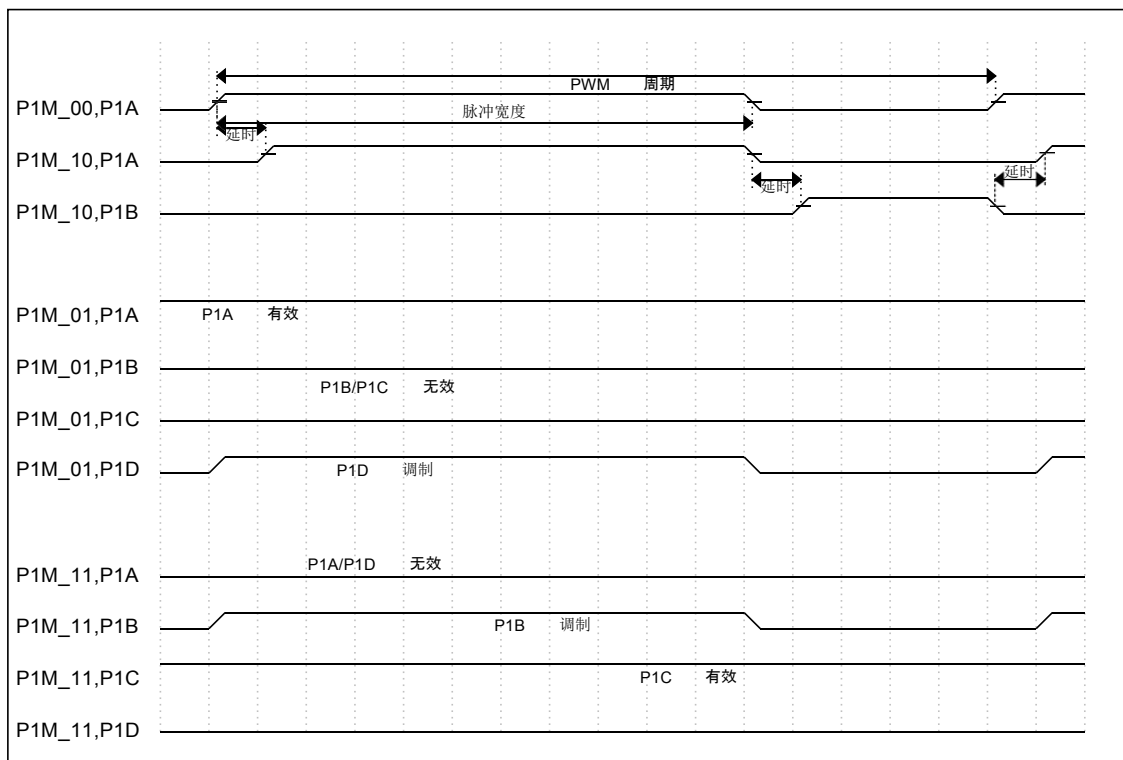


图 9.6 PWM 输出关系示意图（高电平有效）

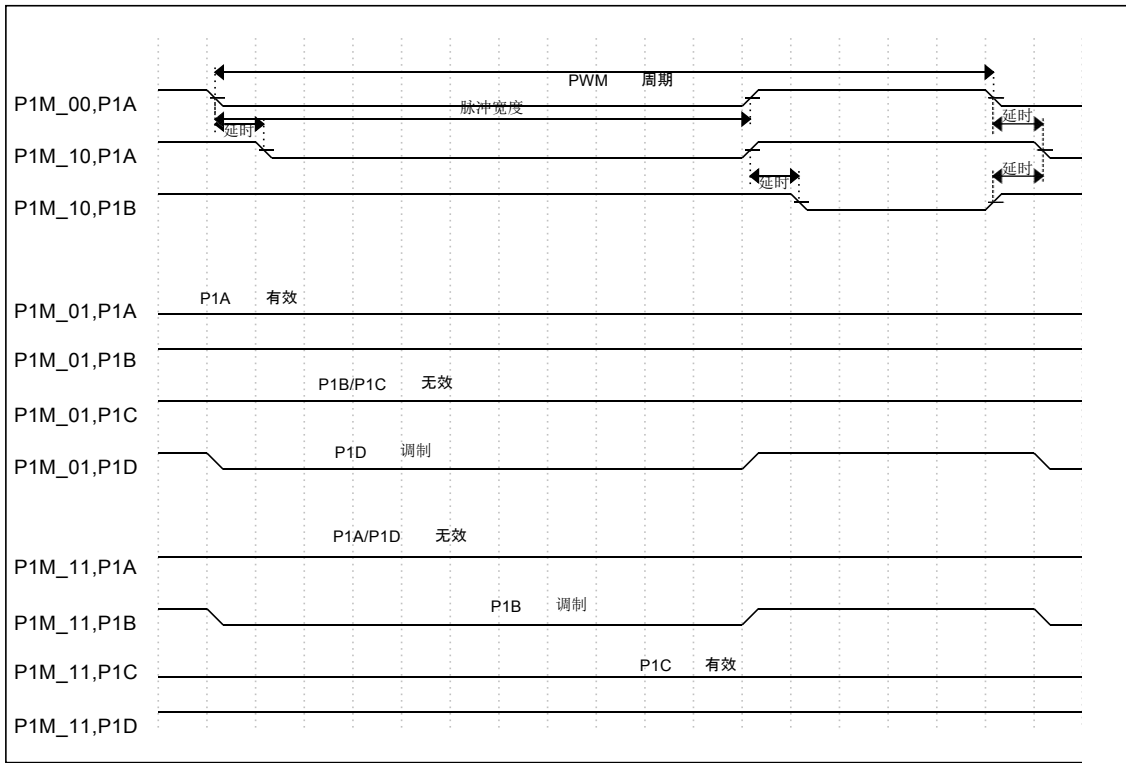


图 9.7 PWM 输出关系示意图（低电平有效）

#### 9.4.1 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 9.6）。此模式可用于半桥应用，或用于全桥应用，此时用两个 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 9.4.6 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。

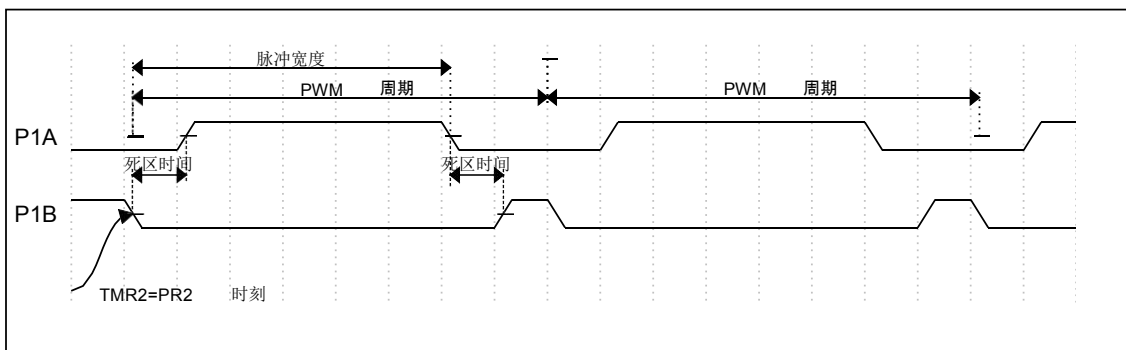


图 9.8 半桥 PWM 输出

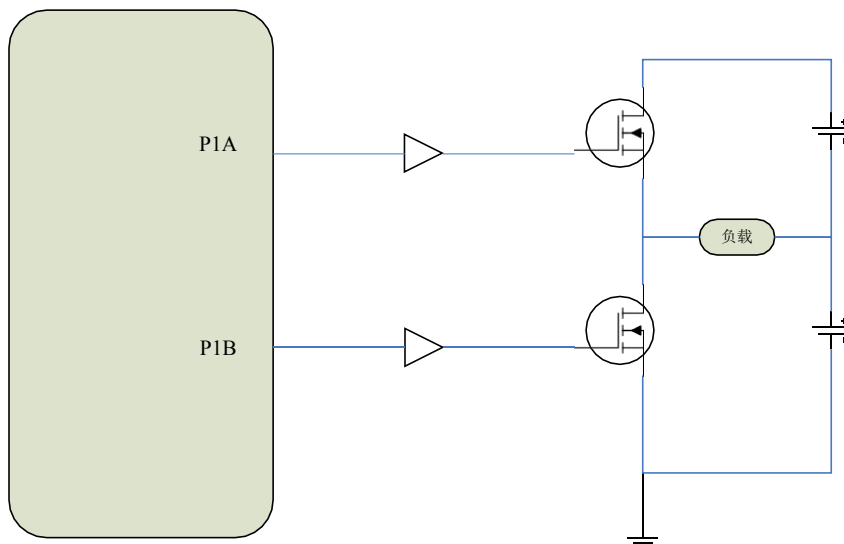


图 9.9 标准半桥电路（推挽）

#### 9.4.2 全桥模式

在全桥模式下，所有四个引脚均用作输出。图 9.10 所示为一个全桥应用示例。

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 则被驱动为无效状态，如图 9.11a 所示。

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 则被驱动为无效状态，如图 9.11b 所示。

P1A、P1B、P1C 和 P1D 输出与 PORT 数据锁存器复用。必须清零相关 TRIS 位以将 P1A、P1B、P1C 和 P1D 引脚配置为输出。

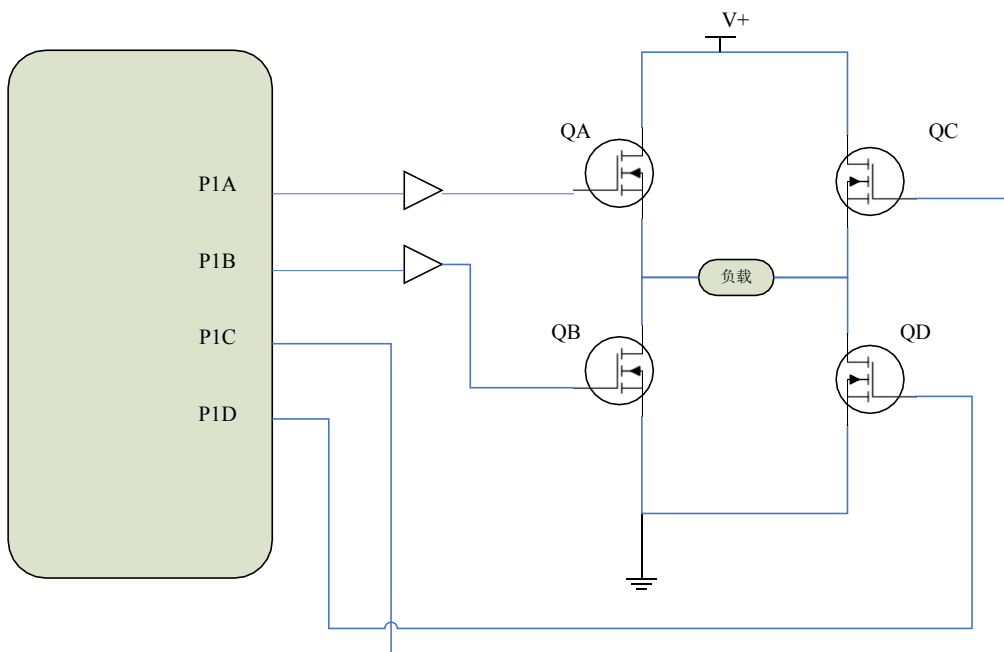


图 9.10 全桥应用示例

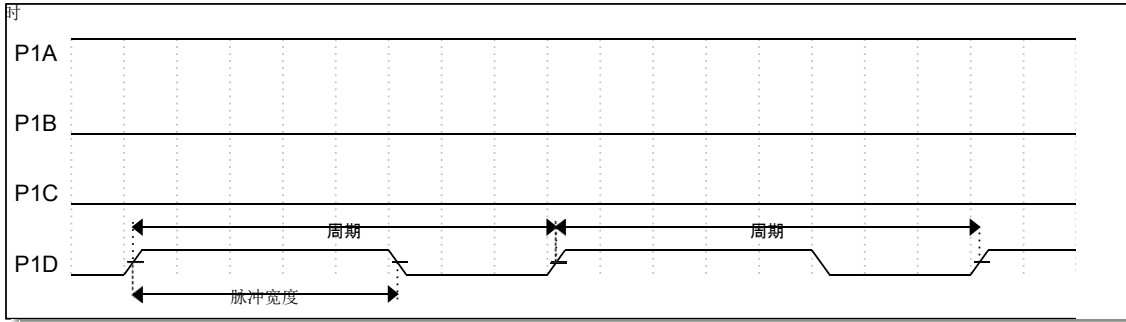


图 9.11a 全桥 PWM 输出示例 a

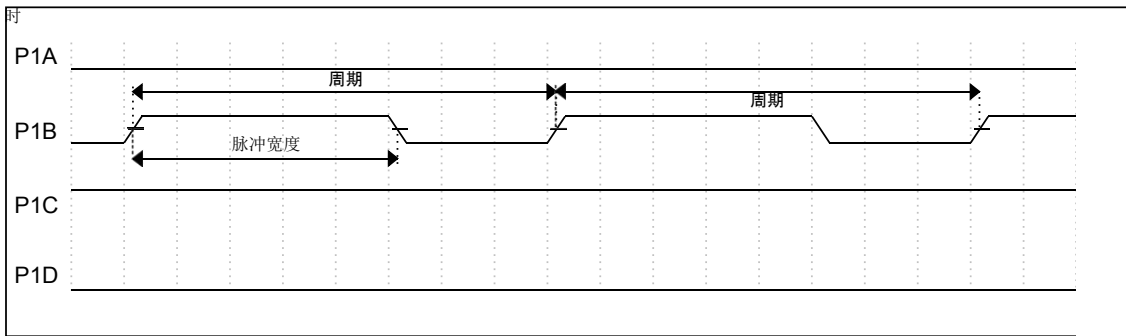


图 9.11b 全桥 PWM 输出示例 b

#### 9.4.2.1 在全桥模式下改变方向

在全桥模式下，CCP1CON 寄存器的 P1M1 位可供用户控制正/反方向。当应用固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：

- 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

此时序的图示请参见图 9.12。

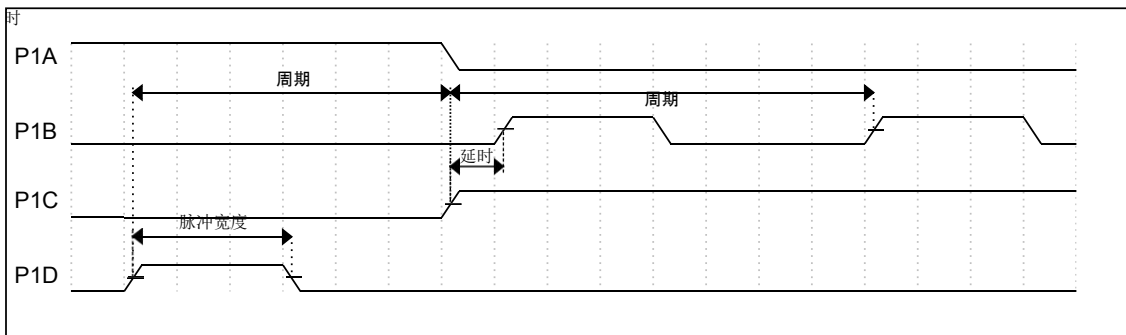


图 9.12 PWM 方向改变示例



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。此时长为 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 9.13 所示为点空比接近 100% 时，PWM 方向从正向变为反向的示例。此示例中，在时间  $t_1$  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 9.10）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。

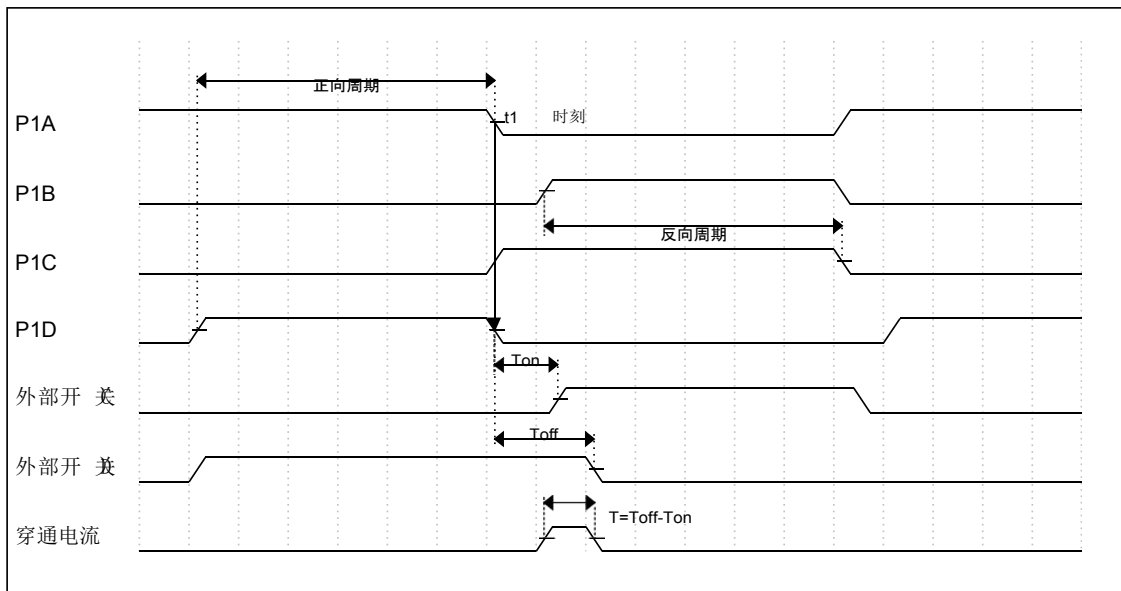


图 9.13 占空比接近 100 %时 PWM 方向变化示例

注意：

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

### 9.4.3 启动考虑事项

使用任何 PWM 模式时，应用硬件中必须在 PWM 输出引脚上使用适当的外部上拉/下拉电阻。

注意：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一整个 PWM 周期。一整个 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 查看。

#### 9.4.4 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
- 在固件中将 ECCPASE 位置 1

关闭状态由 ECCPAS 寄存器的 ECCPASE（自动关闭事件状态）位指示。如果该位为 0，PWM 引脚正常工作。如果该位为 1，PWM 输出处于关闭状态。

发生关闭事件时，将出现两个状况：

1. ECCPASE 位置 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 9.4.5 节“自动重启模式”）；
2. 使能的 PWM 引脚被异步置于其关闭状态。PWM 输出引脚被分为两对[P1A/P1C]和[P1B/P1D]。两对引脚的状态由 ECCPAS 寄存器的 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）

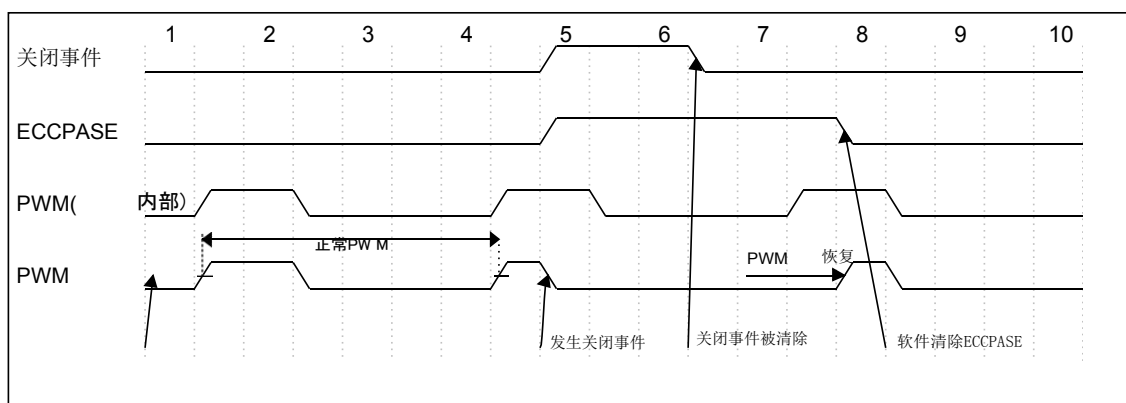


图 9.14 非自动重启时的 PWM 自动关闭

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。

### 9.4.5 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。通过将 PWM1CON 寄存器中的 PRSEN 位置 1 可启用自动重启。

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。

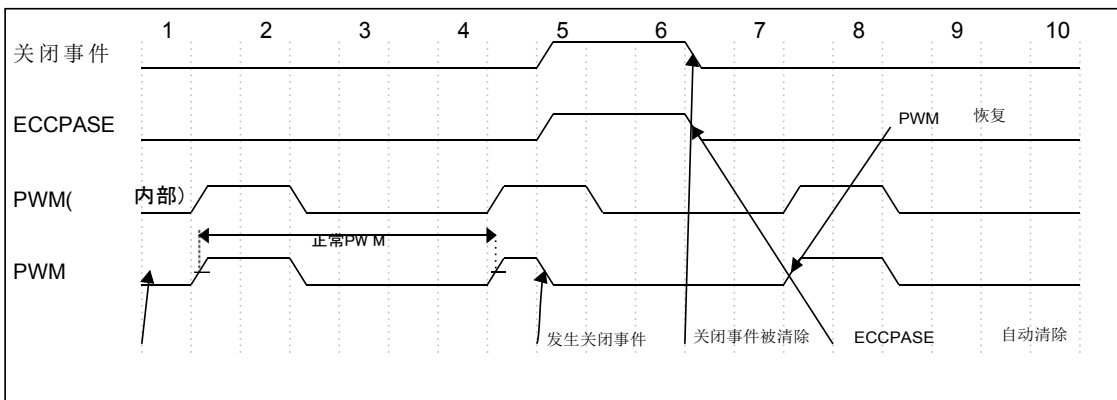


图 9.15 自动重启时的 PWM 自动关闭

### 9.4.6 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 9.16 中的图解说明。相关 PWM1CON 寄存器的低 7 位以单片机的指令周期（TCY，即 2 个系统时钟周期）为单位设置延时期限。

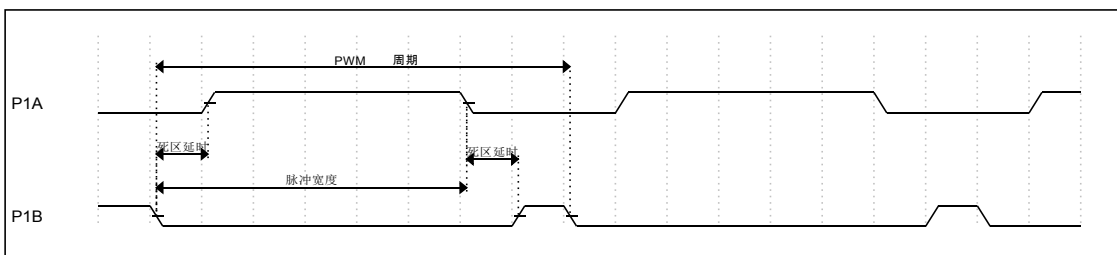


图 9.16 半桥 PWM 输出示例

## 9.5 PWM 的辅助功能

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 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]）

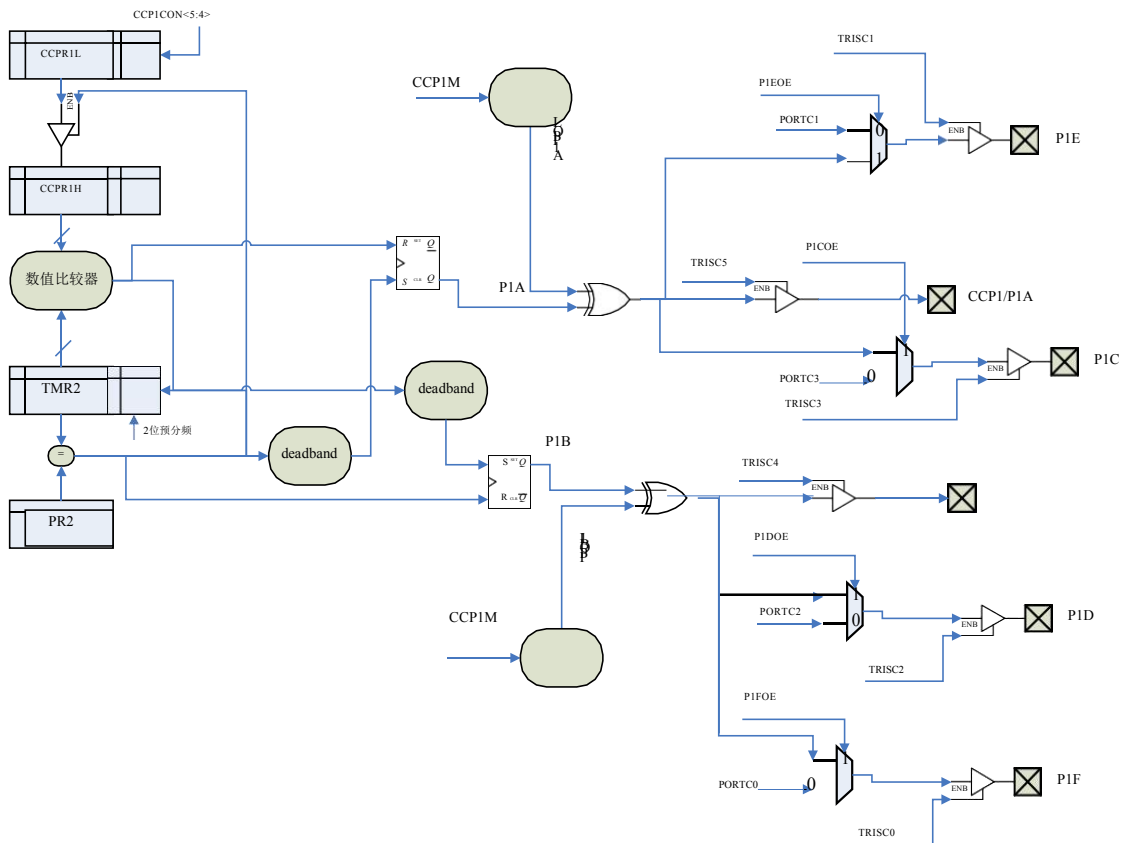


图 9.17 PWM 的辅助功能原理框图

注意：

1. 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  $P1M<1:0>=10$ 。

### 9.5.1 一次性脉冲模式

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 为一次性脉冲模式。

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。

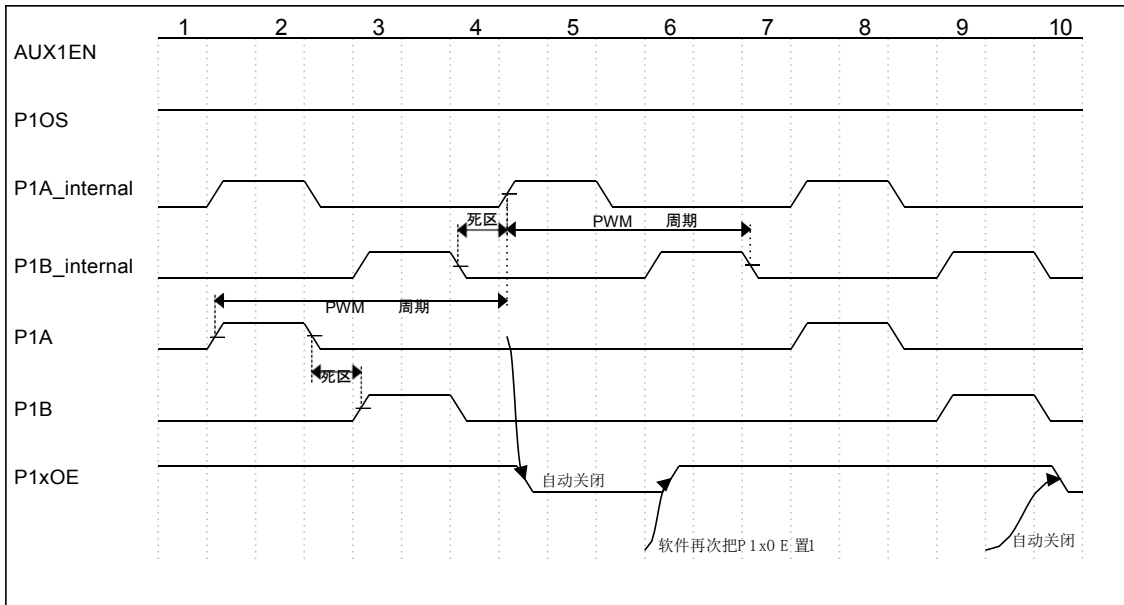


图 9.18 PWM 的辅助功能原理框图

### 9.5.2 3 对 PWM 信号输出

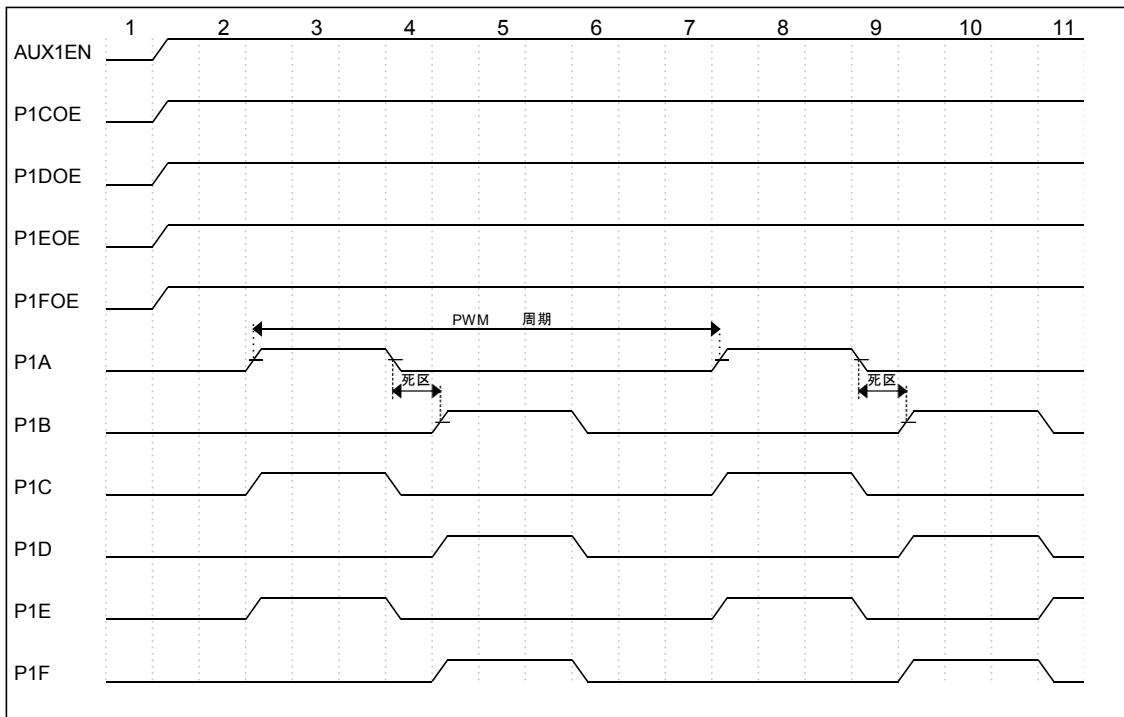


图 9.18 PWM 的辅助功能原理框图

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。



### 9.5.3 PWM 辅助功能的使用

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

### 9.6 与 ECCP 相关寄存器汇总

名称	地址	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	复位值	
TMR2	0x11	TIMER2 计数值寄存器								xxxx xxxx	
PR2	0x92	TIMER2 周期寄存器								1111 1111	
INTCON	0x0B/8B	GIE	PEIE	T0IE	INTE	PAIE	T0IF	INTF	PAIF	0000 0000	
PIE1	0x8C	EEIE	CKMEAIE	-	C2IE	C1IE	OSFIE	TMR2IE	TMR1IE	0000 0000	
PIR1	0x0C	EEIF	CKMEAIF	-	C2IF	C1IF	OSFIF	TMR2IF	TMR1IF	0000 0000	
PIE2	0x8D	-	-	-	-	-	-	ADIE	CCP1IE	---- --00	
PIR2	0x0C	-	-	-	-	-	-	ADIF	CCP1IF	---- --00	
PWM1AUX	0x90	AUX1EN	P1OS	P1FOE	P1EOE	P1DOE	P1COE	P1BOE	P1AOE	0000 0000	
T1CON	0x10	T1GINV	TMR1GE	T1CKPS[1:0]		T1OSCEN	T1SYNC	TMR1CS	TMR1ON	0000 0000	
T2CON	0x12	-	TOUTPS[3:0]			-	TMR2ON	T2CKPS[1:0]		-000 0000	
TMR1L	0x0E	16 位 TIMER1 计数值低 8 位								xxxx xxxx	
TMR1H	0x0F	16 位 TIMER1 计数值高 8 位								xxxx xxxx	
TRISA	0x85	TRISA[7:0]，PORTA 方向控制								1111 1111	
TRISC	0x87	-	-	TRISC[5:0]，PORTC 方向控制						--11 1111	
CCPR1L	0x13	ECCP1 寄存低 8 位								xxxx xxxx	
CCPR1H	0x14	ECCP1 寄存高 8 位								xxxx xxxx	
CCP1CON	0x15	P1M[1:0]			DC1B[1:0]		CCP1M[3:0]			0000 0000	
PWM1CON	0x16	PRSEN	PDC[6:0]								0000 0000
ECCPAS	0x17	ECCPASE	ECCPAS[2:0]			PSSAC[1:0]		PSSBD[1:0]		0000 0000	



### 9.6.1 CCPR1L 寄存器 · 地址 0x13

Bit	7	6	5	4	3	2	1	0
Name	CCPR1L							
Reset	x	x	x	x	x	x	x	x
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:0	CCPR1L[7:0]	捕捉、比较、PWM 寄存器 1 的低字节

### 9.6.2 CCPR1H 寄存器 · 地址 0x14

Bit	7	6	5	4	3	2	1	0
Name	CCPR1H							
Reset	x	x	x	x	x	x	x	x
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:0	CCPR1H[7:0]	捕捉、比较、PWM 寄存器 1 的高字节

### 9.6.3 CCP1CON 寄存器 · 地址 0x15

Bit	7	6	5	4	3	2	1	0
Name	P1M1	P1M0	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:6	P1M[1:0]	<p>PWM 输出配置位</p> <p>如果 CCP1M&lt;3:2&gt; = 00、01 或 10：</p> <p>xx =P1A 配置为捕捉/比较输入；P1B、P1C 和 P1D 配置为端口引脚</p> <p>如果 CCP1M&lt;3:2&gt; = 11：</p> <p>00 = 单输出；P1A 调制；P1B、P1C 和 P1D 配置为端口引脚</p> <p>01 = 全桥正向输出；P1D 调制；P1A 有效；P1B 和 P1C 无效</p> <p>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</p> <p>11 = 全桥反向输出；P1B 调制；P1C 有效；P1A 和 P1D 无效</p>
5:4	DC1B[1:0]	<p>PWM 占空比最低有效位</p> <p>捕捉模式：</p>



		未使用 比较模式： 未使用 PWM 模式： 这些位是 PWM 占空比的低 2 位。高 8 位在 CCP1L 中
3:0	CCP1M[3:0]	ECCP 模式选择位 0000 = 捕捉/比较/PWM 关闭（复位 ECCP 模块） 0001 = 未使用（保留） 0010 = 比较模式，匹配时翻转输出（CCP1IF 位置 1） 0011 = 未使用（保留） 0100 = 捕捉模式，每个下降沿（CCP1IF 位置 1） 0101 = 捕捉模式，每个上升沿（CCP1IF 位置 1） 0110 = 捕捉模式，每 4 个上升沿（CCP1IF 位置 1） 0111 = 捕捉模式，每 16 个上升沿（CCP1IF 位置 1） 1000 = 比较模式，匹配时输出置 1（CCP1IF 位置 1） 1001 = 比较模式，匹配时输出清零（CCP1IF 位置 1） 1010 = 比较模式，匹配时产生软件中断（CCP1IF 位置 1，CCP1 引脚不受影响） 1011 = 比较模式，触发特殊事件（CCP1IF 位置 1，CCP1 复位 TMR1，且如果 A/D 模块被使能，启动一次 A/D 转换） 1100 = PWM 模式；P1A 和 P1C 高电平有效；P1B 和 P1D 高电平有效 1101 = PWM 模式；P1A 和 P1C 高电平有效；P1B 和 P1D 低电平有效 1110 = PWM 模式；P1A 和 P1C 低电平有效；P1B 和 P1D 高电平有效 1111 = PWM 模式；P1A 和 P1C 低电平有效；P1B 和 P1D 低电平有效

### 9.6.4 PWM1CON 寄存器，地址 0x16

Bit	7	6	5	4	3	2	1	0
Name	PRSEN	PDC6	PDC5	PDC4	PDC3	PDC2	PDC1	PDC0
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7	PRSEN	PWM 重启使能位 1 = 自动关闭时，ECCPASE 位在退出关闭事件时自动清零）PWM 自动重启 0 = 自动关闭时，必须用软件将 ECCPASE 清零以重启 PWM
6:0	PDC[6:0]	PWM 延时计数位 PDCn = 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



### 9.6.5 ECCPAS 寄存器，地址 0x17

Bit	7	6	5	4	3	2	1	0
Name	ECCPASE	ECCPAS2	ECCPAS1	ECCPAS0	PSSAC1	PSSAC0	PSSBD1	PSSBD0
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7	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
6:4	ECCPAS[2:0]	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
3:2	PSSAC[1:0]	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
1:0	PSSBD[1:0]	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

### 9.6.6 PWM1AUX 寄存器，地址 0x90

Bit	7	6	5	4	3	2	1	0
Name	AUX1EN	P1AINV	P1BINV	P1IOS	P1FOE	P1EOE	P1DOE	P1COE
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7	AUX1EN	PWM1 辅助功能使能位 1 = 使能 PWM1 辅助功能 0 = 禁止 PWM1 辅助功能
6	P1IOS	当 AUX1EN=1 且 ECCP 处于半桥 PWM 模式时，



		1 = PWM 输出一次脉冲后自动停止，P1A/P1B/P1C/P1D/P1E/P1F 变为普通 IO 0 = PWM 连续输出 当 AUX1EN=0 时，该位无意义
5	P1FOE	当 AUX1EN=1 且 ECCP 处于半桥 PWM 模式时， 1 = 引脚 P1F 变为 PWM 输出 0 = 引脚 P1F 变为 IO 当 AUX1EN=0 时，该位无意义
4	P1EOE	当 AUX1EN=1 且 ECCP 处于半桥 PWM 模式时， 1 = 引脚 P1E 变为 PWM 输出 0 = 引脚 P1E 变为 IO 当 AUX1EN=0 时，该位无意义
3	P1DOE	当 AUX1EN=1 且 ECCP 处于半桥 PWM 模式时， 1 = 引脚 P1D 变为 PWM 输出 0 = 引脚 P1D 变为 IO 当 AUX1EN=0 时，该位无意义
2	P1COE	当 AUX1EN=1 且 ECCP 处于半桥 PWM 模式时， 1 = 引脚 P1C 变为 PWM 输出 0 = 引脚 P1C 变为 IO 当 AUX1EN=0 时，该位无意义
1	P1BOE	当 AUX1EN=1 且 ECCP 处于半桥 PWM 模式时， 1 = 引脚 P1B 变为 PWM 输出 0 = 引脚 P1B 变为 IO 当 AUX1EN=0 时，该位无意义
0	P1AOE	当 AUX1EN=1 且 ECCP 处于半桥 PWM 模式时， 1 = 引脚 P1A 变为 PWM 输出 0 = 引脚 P1A 变为 IO 当 AUX1EN=0 时，该位无意义

**注意：**

1. 如果 PWM 工作在半桥模式且 AUX1EN 和 P1OS 同时为 1 时，P1AOE~P1FOE 这 6 位将在下一个 PWM 周期到来后自动清 0；
2. 当使用 PWM 的辅助功能，即 AUX1EN 为 1 且 ECCP 处于半桥模式时，P1A~P1F 是否输出 PWM 波形还要取决于 P1AOE~P1FOE 的值，只有当它们为 1 时才作为 PWM 功能管脚，否则是 GPIO。



## 10 比较器

片内集成 2 个模拟比较器可以用来比较 2 个模拟电压值而产生数字结果反映模拟电压值的高低关系。本芯片中所包含的比较器有以下特性：

- 比较结果可输出到片外
- 输出极性可编程
- 输出可作为中断源
- 输出可作为唤醒源
- 输入可挂内部参考电压
- 双比较器
- 多种比较器输入输出连接设定模式
- 可编程参考电压

### 10.1 比较器概述

模拟比较器的输入输出信号关系以及其在电路图中的标志如图 10.1 所示。当比较器的模拟输入电压  $V_{in+}$  小于其模拟输入电压  $V_{in-}$  时，输出为逻辑低电平，反之则输出逻辑高电平。

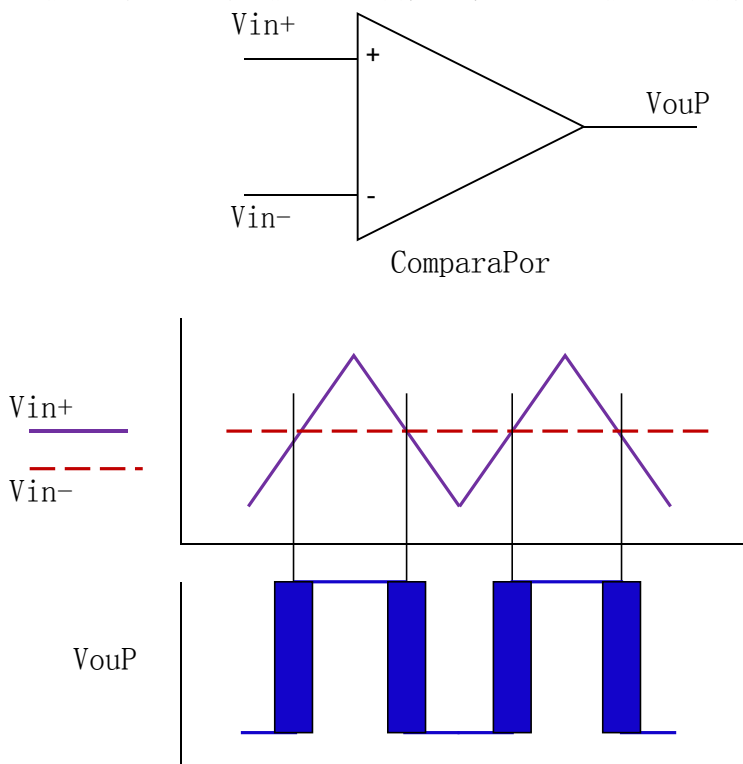
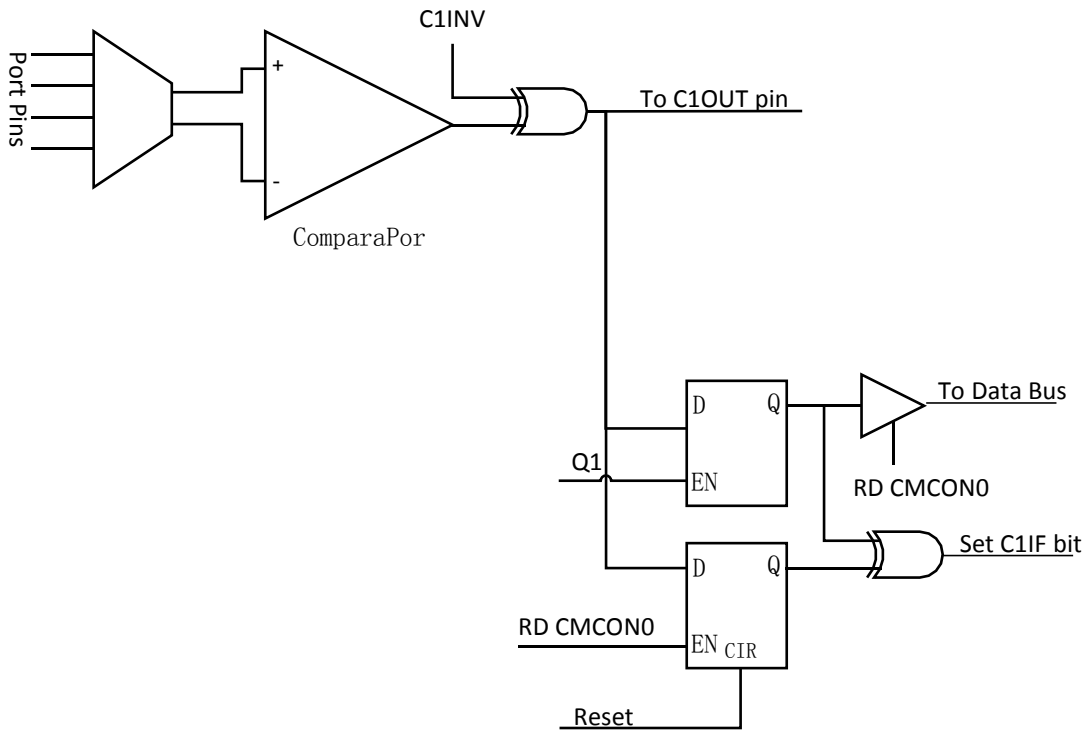


图 10.1 比较器的输入，输出信号特性

当比较器的输入和输出接近相等时，其输出为不定态，因此，为了避免这种不定态，或再某个输入端有微弱噪声存在，造成比较输入出现不稳定态，一般比较器的设计会保护一个所谓迟滞电压，是比较器的输出态变化会延后输入变化一定的距离，这样基本比较电平稍有波动也不会造成杂散输出信号。

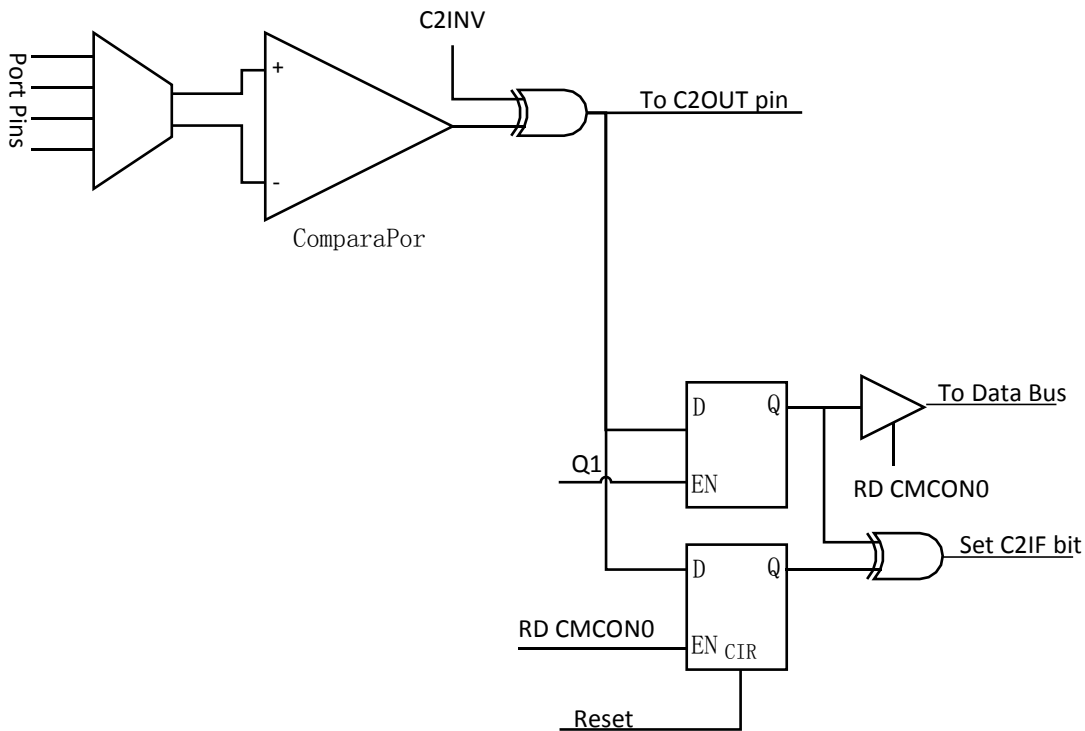
本芯片包含了如下图所示的 2 个比较器，但这 2 个比较器的配置不是独立分开的。



Q1为系统时钟相位.

软件在改变比较器的输入，输出控制寄存器时最好关闭比较器，避免比较器结果错误.

图 10.2 比较器 1 的周边电路



Q1为系统时钟相位.

软件在改变比较器的输入，输出控制寄存器时最好关闭比较器，避免比较器结果错误.

图 10.3 比较器 2 的周边电路

### 10.1.1 模拟输入端的连接

比较器的模拟输入端和同样连接该端口的数字输入共享管脚，需要注意的是在这些管脚上都有对 VDD 和 VSS 的反向保护二极管。如果输入电压偏离这个保护范围以外 0.6V，二极管就会导通到时间锁效应。我们推荐的最大输入源信号的源阻抗不超过 10K 欧姆。另外，如果管脚上外挂电容或齐纳二极管之类的元件，其不应该产生漏电流，不然可能会造成结果不精确。

需要注意的是，当读一个端口寄存器时，如果改管脚被配置为模拟信号管脚，软件会读出 0 值。当管脚被设置为数字输入管脚时，比较器仍然会以为该管脚输入一个模拟信号，并输出相应结果。如果一个管脚被设置为数字输入，而这个管脚上面的实际电压又是一个模拟电平，这可能造成输入缓存电路消耗比说明书上标的更大的电流。

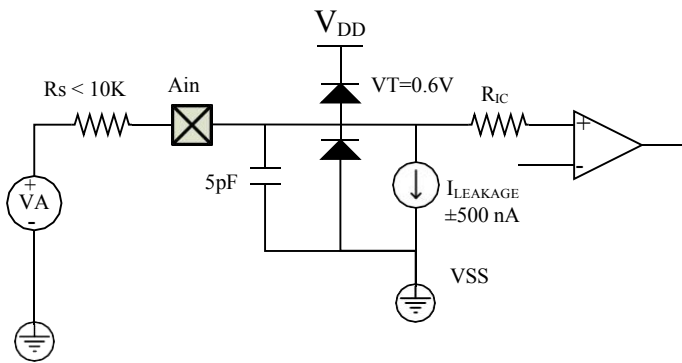


图 10.4 比较器模拟输入模型

### 10.2 比较器配置

模拟比较器总共有 8 种配置模式。由 CMCON0 寄存器的 CM<2:0>这 3 位来选择。图 10.5 详细描述了这 8 种选择的具体内容。I/O 信号线的功能也随着这 8 中不同的配置而改变，具体如下：

- 模拟功能（A）：数字输入缓存被屏蔽
- 数字功能（D）：比较器数字输出会覆盖管脚上的其他功能
- 正常端口功能（I/O）：独立于比较器

当端口上标注“A”字样，读的时候无论当前管脚上的状态或 I/O 控制寄存器 TRIS 位的状态都会返回 0 值。用户应该把与被用作模拟输入的管脚相对应的 TRIS 位置为 1 来关闭其数字输出驱动电路。当端口上标注“D”字样，用户应该将其相应的 TRIS 位职位 0 来打开数字输出驱动电路。

另外，比较器配置切换是应该屏蔽比较器中断以避免不必要的误触发事件。

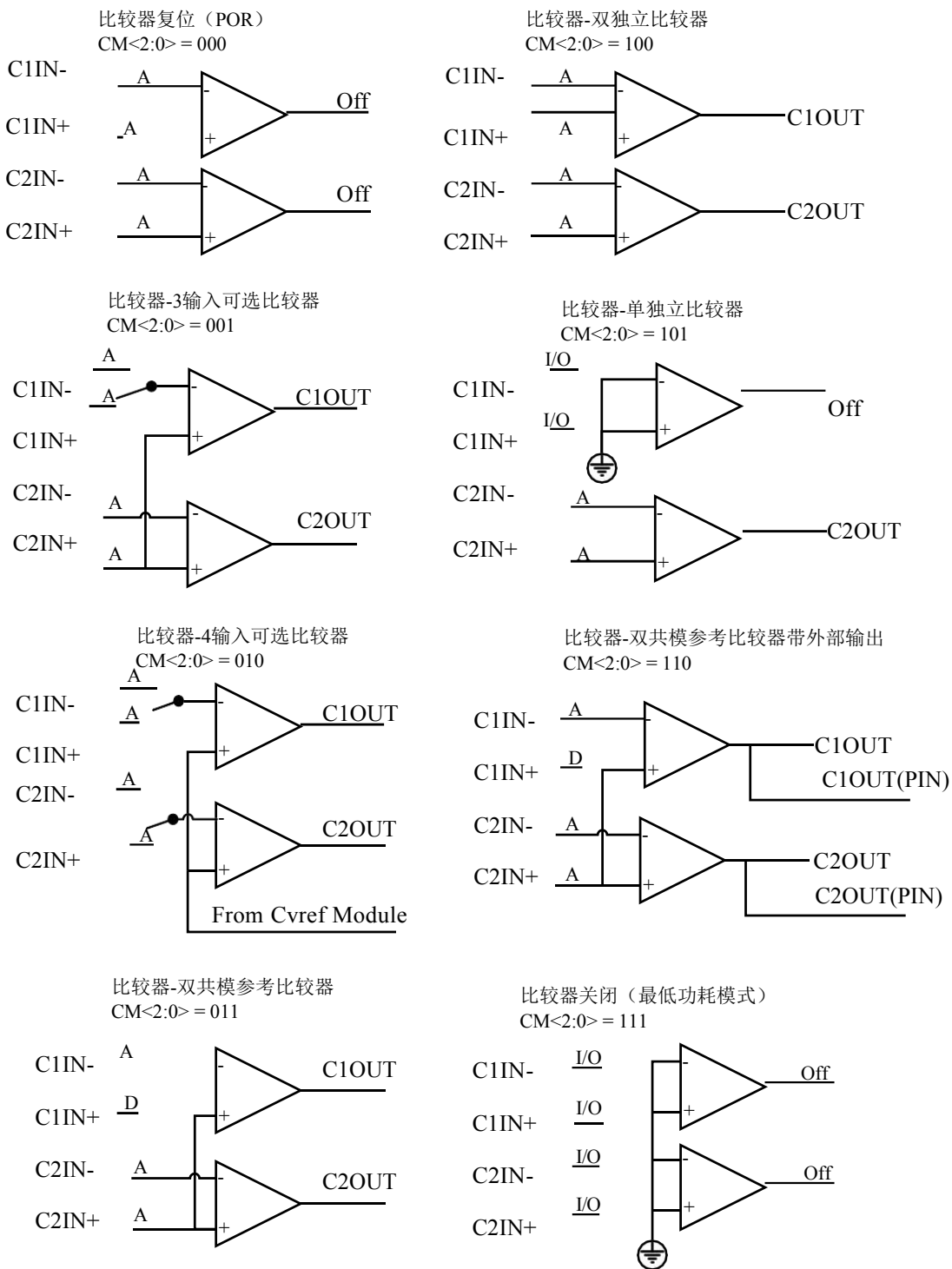


图 10.5 比较器输入，输出配置模式

### 10.3 比较器控制

CMCON0 寄存器提供了以下比较器特性的控制功能：

- 模式选择
- 输出状态
- 输出极性
- 输入开关

### 10.3.1 模拟输出状态

每个比较器的状态都能通过 CMCON0 寄存器的 CxOUT 位从内部读出。当 CM<2:0>=110 时比较器输出会被连到 CxOUT 管脚。当选择这个模式时，CxOUT 相应的 TRIS 位必须被清零来使能输出驱动电路。

### 10.3.2 模拟输出极性

将比较器输出反向就等效于将比较器的输入端交换。比较器输出极性可由 CMCON0 寄存器里的 CxINV 位来控制。将 CxINV 位清零代表比较器输出未被反向。完整的输出，输入条件和极性如下表所列：

Input Condition	CxINV	CxOUT
Vin- > Vin+	0	0
Vin- < Vin+	0	1
Vin- > Vin+	1	1
Vin- < Vin+	1	0

CxOUT 包括寄存器位和实际输出管脚。

### 10.3.3 模拟输入开关

比较器的模拟输入负端在以下模式可被切换到 2 个模拟管脚。

- CM<2:0> = 001 (仅比较器 1)
- CM<2:0> = 010 (比较器 1 和 2)

在以上模式下，无论哪根管脚被选做输入，这 2 个管脚都处在模拟模式。CMCON0 寄存器里的 CIS 位控制比较器的输入切换开关。

## 10.4 比较器反应时间

模拟比较器输出在输入变化或输入新的参考电压后若干时间后才会变化，这个时间叫做比较器反应时间。反应时间和参考电压的稳定时间还不是同一个概念。这两者加起来才是比较器的总体反应时间。具体参数，请参考本文档后面的电气参数部分。

## 10.5 比较器中断

一旦使能，模拟比较器输出状态的变化就会触发中断。输出状态变化的捕捉用两个锁门器和一个异或门完成。(见图 10.2, 10.3)。当读取 CMCON0 寄存器时，一个锁门器由比较器输出更新。该锁门器保持着状态直到 CMCON0 下一次被读或复位。另一个锁门器在系统时钟的 Q1 期更新。状态失配状态会被一直保存，并使 PIR1 寄存器的 CxIF 位为 1，直到 CMCON0 被读或比较器输出返回到以前的状态。(注，对 CMCON0 寄存器的写动作也会清除失配条件，应为所有的写动作实际上在写之前都包含着一个读动作)。

软件需要自身保存比较器的输出状态来决定实际上变化是否有发生。PIR1 寄存器里的 CxIF 位是比较器的中断指示位。这 1 位必须有软件来清零。由于软件可以将这 1 位写成 1，因此软件可以做模拟比较器变化的动作。

比较器的中断发生还需要 INTCON 寄存器中的 PEIE 位和 GIE 位联合搭配工作。如果这其中任何 1 位没有置 1，哪怕 CxIF 位变成 1 了，中断都不会发生。用户可以通过以下动作清除中断：

1. 对 CMCON0 进行读或写操作，这会结束失配条件

## 2. 清除 CxIF 位

持续的失配条件会屏蔽住 CxIF 标志位被清，因此有必要先读一下 CMCON0 寄存器先清除失配条件。

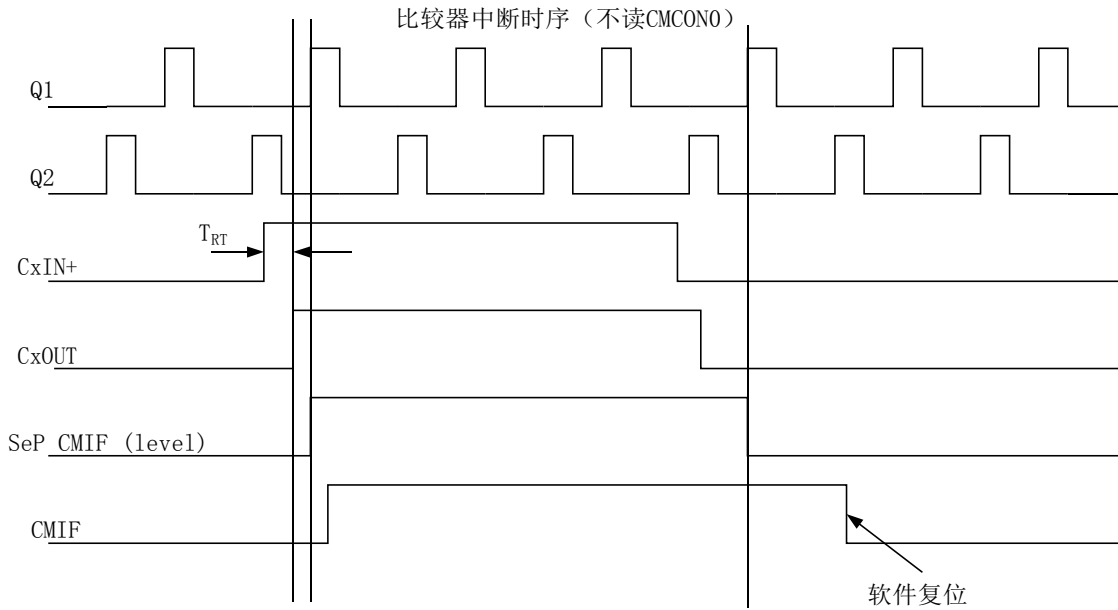


图 10.6 比较器中断时序 1

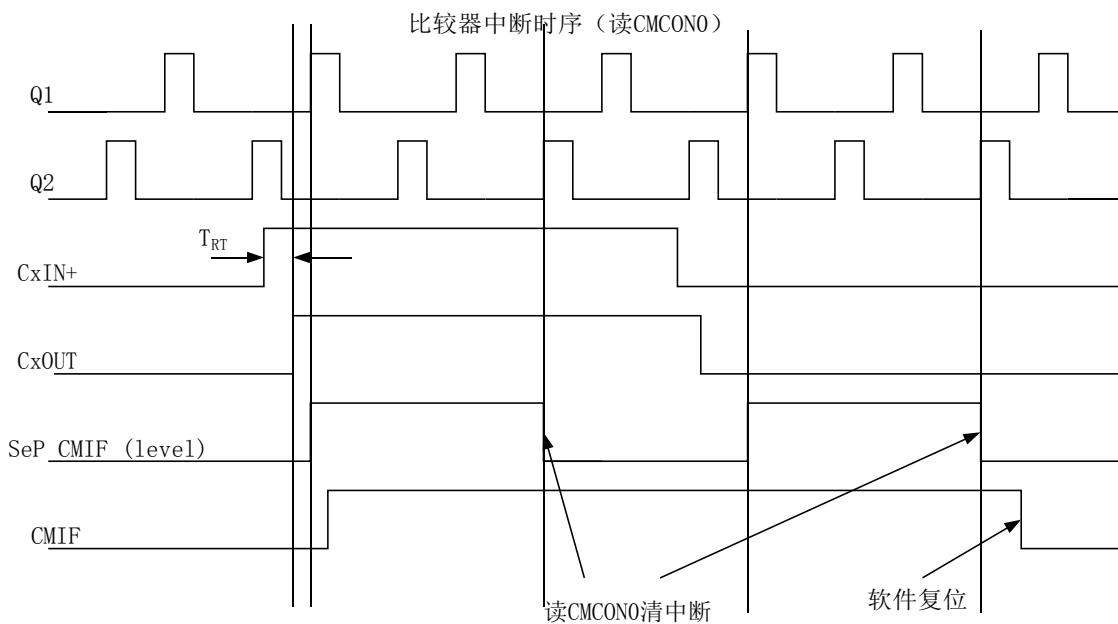


图 10.7 比较器中断时序 2

需要注意的是如果在读 CMCON0 的时候（在系统时钟的 Q2 期），CxOUT 刚好发生变化，那么 CxIF 标志位有可能不会被置高。

另外，比较器电路被启动时，其偏置电路需要 1uS 的时间稳定，在此期间，比较器的输出是无效的，软件应该在此期间关闭中断以免产生不必要的误触发。

## 10.6 比较器在睡眠状态下的工作

如果用户在使芯片进入睡眠之前使能比较器，那么比较器会在睡眠状态下继续工作。但此时芯片的功能会相应增加。如果用户并不使用比较器输出来唤醒芯片，用户可以在睡眠前关闭比较器。关闭比较器只要在 CMCON0 寄存器中将 CM<2:0>置为 000 或 111 即可。

如上所述，比较器的输出可以用来唤醒芯片。如果需要用比较器输出唤醒芯片，需要在 PIE1 寄存器中将 CxIE 位置 1，以及 INTCON 寄存器下的 PEIE 位置 1。如果 INTCON 寄存器的 GIE 位被置 1，器件就会执行中断服务程序。

## 10.7 比较器在复位状态下的工作

芯片复位会强制 CMCON0 和 CMCON1 寄存器进入复位状态，也就是 CM<2:0>=000 模式。此时，所有比较器输入都为模拟输入，比较器本身被关闭以节省功耗。

## 10.8 比较器 5 门控 Timer1

可使用此功能为模拟事件的持续时间或间隔定时。将 CMCON1 寄存器的 T1GSS 位清零将使 Timer1 根据比较器 5 的输出递增。这要求 Timer1 打开且门控使能。

详情请参见第 7 节“带门控的 Timer1 模块”。

当比较器用作 Timer1 门控源时，建议通过将 C2SYNC 位置 1 将比较器 5 与 Timer1 同步。这将确保 Timer1 在递增时若比较器变化，Timer1 不会错过递增。

## 10.9 比较器 5 输出与 Timer1 同步

通过将 CMCON1 寄存器的 C2SYNC 位置 1 可将比较器 5 的输出与 Timer1 同步。使能时，比较器输出在 Timer1 时钟源的下降沿被锁存。Timer1 使用预分频器时，比较器输出在预分频后被锁存。为了防止竞争情况，比较器输出在 Timer1 时钟源下降沿被锁存，而 Timer1 在时钟源的上升沿递增。更多信息，请参见比较器框图（图 10.6）以及 Timer1 框图（图 7.1）。

## 10.10 比较器的参考电压

模拟比较器的输入之一可以被设置成为内部参考电压，该参考电压都有以下特性：

- 独立于比较器的运行
- 两个 16 级电压范围
- 输出低钳位到 VSS
- 输出电压与 VDD 成比例

参考电压的控制由 VRCON 寄存器来控制，如图 10.8。

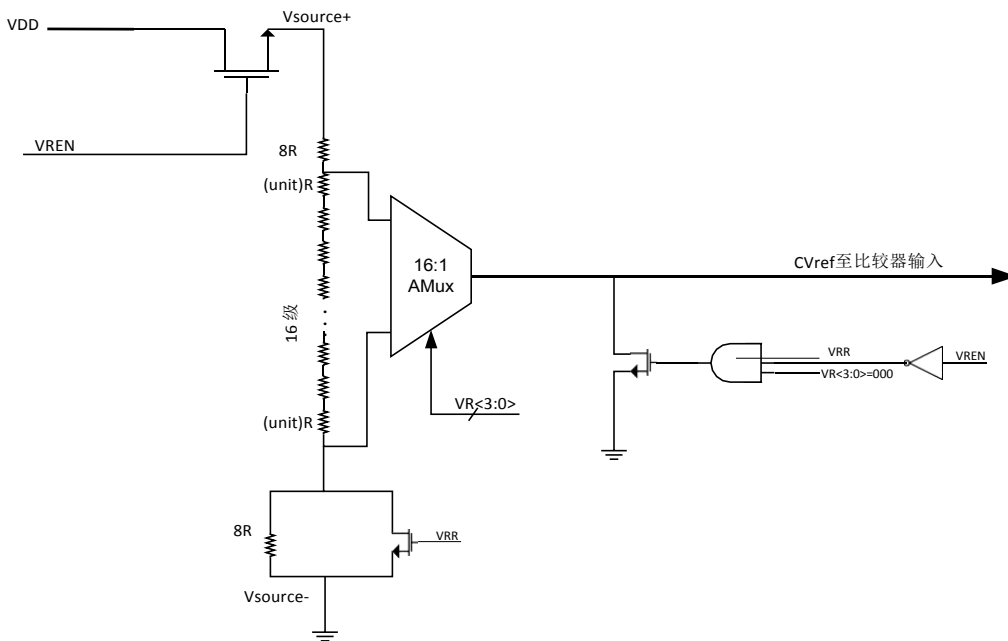


图 10.8 比较器参考电压框图

### 10.10.1 独立工作

比较器参考电压与比较器配置不相关。将 VRCON 寄存器的 VREN 位将使能参考电压。

### 10.10.2 输出电压选择

CVREF 参考电压有两种范围，每种均为 16 个的选择由 VRCON 寄存器的 VRR 位控制。这由 VRCON 寄存器的 VR<3:0>位设置。

CVREF 输出电压由以下右边公式确定：

**VRR = 1 (低电压范围):**  
 $CVREF = (VR<3:0>/24) \times VDD$   
**VRR = 0 (高电压范围):**  
 $CVREF = (VDD/4) + (VR<3:0> \times VDD/32)$

电平 范围  
16 个电平

如图 10.8 所示，由于模块的构造所限，无法实现 VSS 至 VDD 的满量程。

### 10.10.3 输出钳位至VSS

通过将 VRCON 按如下配置可将 CVREF 输出电压设置为 VSS，从而不消耗功率：

- VREN = 0
- VRR = 1
- VR<3:0> = 0000

这使比较器可进行过零检测而不消耗额外的 CVREF 模块电流。





## 10.11 与比较器相关寄存器汇总

名称	地址	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	复位值
ANSEL	0x91	ANSEL[7:0]，数模管脚选择								1111 1111
INTCON	0x0B/8B	GIE	PEIE	T0IE	INTE	PAIE	T0IF	INTF	PAIF	0000 0000
CMCON0	0x19	C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0	0000 0000
CMCON1	0x1A	-	-	-	-	-	-	T1GSS	C2SYNC	---- -- 10
VRCON	0x99	VREN	-	VRR	-	VR3	VR2	VR1	VR0	0-0- 0000
PIE1	0x8C	EEIE	CKMEAIE	-	C2IE	C1IE	OSFIE	TMR2IE	TMR1IE	0000 0000
PIR1	0x0C	EEIF	CKMEAIF	-	C2IF	C1IF	OSFIF	TMR2IF	TMR1IF	0000 0000
TRISA	0x85	TRISA[7:0]，PORTA 方向控制								1111 1111
PORTA	0x05	PORTA[7:0]，PORTA 数据寄存器								X X X X X X X X
TRISC	0x87	-	-	TRISC[5:0]，PORTC 方向控制						-- 11 1111
PORTC	0x07	-	-	PORTC[5:0]，PORTC 数据寄存器						-- X X X X X X

### 10.11.1 CMCON0 寄存器，地址 0x19

Bit	7	6	5	4	3	2	1	0
Name	C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0
Reset	0	0	0	0	1	0	0	0
Type	RO	RO	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7	C2OUT	比较器 2 输出 bit 当 C2INV=0 1: C2 V <sub>IN+</sub> > C2 V <sub>IN-</sub> 0: C2 V <sub>IN+</sub> < C2 V <sub>IN-</sub> 当 C2INV=1 1: C2 V <sub>IN+</sub> < C2 V <sub>IN-</sub> 0: C2 V <sub>IN+</sub> > C2 V <sub>IN-</sub>
6	C1OUT	比较器 1 输出 bit 当 C1INV=0 1: C1 V <sub>IN+</sub> > C1 V <sub>IN-</sub> 0: C1 V <sub>IN+</sub> < C1 V <sub>IN-</sub> 当 C1INV=1 1: C1 V <sub>IN+</sub> < C1 V <sub>IN-</sub> 0: C1 V <sub>IN+</sub> > C1 V <sub>IN-</sub>
5	C2INV	比较器 2 输出反向 bit 0 = 不反向 1 = 反向
4	C1INV	比较器 1 输出反向 bit 0 = 不反向



		1 = 反向
3	CIS	比较器输入切换 当 CM[2:0] = 010 时， 1 = C1 <sub>IN+</sub> 接 C1 V <sub>IN-</sub> C2 <sub>IN+</sub> 接 C2 V <sub>IN-</sub> 0 = C1 <sub>IN-</sub> 接 C1 V <sub>IN-</sub> C2 <sub>IN-</sub> 接 C2 V <sub>IN-</sub> 当 CM[2:0] = 001 时， 1 = C1 <sub>IN+</sub> 接 C1 V <sub>IN-</sub> 0 = C1 <sub>IN-</sub> 接 C1 V <sub>IN-</sub>
2-0	CM[2:0]	比较器模式位: 000 = 比较器关闭，CxIN管脚为模拟IO管脚 001 = 3个输入共用到2个比较器上 010 = 4个输入共用到2个比较器上 011 = 2个共参考比较器 100 = 2个独立比较器 101 = 1个独立比较器 110 = 2个带输出共参考比较器 111 = 比较器关闭，CxIN管脚为数字IO管脚

### 10.11.2 CMCON1 寄存器，地址 0x1A

Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	T1GSS	C2SYNC
Reset	-	-	-	-	-	-	1	0
Type	-	-	-	-	-	-	RW	RW

Bit	Name	Function
7:2	-	未实现，读 0 不要向这些位写 1
1	T1GSS	Timer1 门控源选择位 1 = Timer1 门控源为 T1G 引脚（引脚应配置为数字输入） 0 = Timer1 门控源为比较器 C2 的输出
0	C2SYNC	比较器 C2 输出同步位 1 = 输出与 Timer1 时钟的下降沿同步 0 = 异步输出

### 10.11.3 VRCON 寄存器，地址 0x99

Bit	7	6	5	4	3	2	1	0
Name	VREN	-	VRR	-	VR3	VR2	VR1	VR0



Reset	0	-	0	-	0	0	0	0
Type	RW	-	RW	-	RW	RW	RW	RW

Bit	Name	Function
7	VREN	CVref 使能位 1 = CVref 电路通电 0 = CVref 电路断电，无泄漏电流
6	-	未实现，读 0
5	VRR	CVref 范围选择位 1 = 低电平范围 0 = 高电平范围
4	-	未实现，读 0
3:0	VR<3:0>	CVref 值选择位 VRR=1 时：CVref= (VR<3:0>/24)*VDD VRR=0 时：CVref= VDD/4 + (VR<3:0>/32)*VDD

## 11 数据 EEPROM

片内集成有 256 个字节的 EEPROM，通过 EEADR 进行寻址访问。软件可通过 EECON1 和 EECON2 对 EEPROM 进行编程操作，硬件实现了擦除和编程的自定时功能，无需软件查询，节省有限的代码空间，同时利用此特性，启动编程周期之后可以进入睡眠模式，以降低功耗。

数据 EEPROM 在使用（无论是读还是写）之前必须进行以下初始化操作：在未使用到的 EEPROM 某个单元写两次 0xAA，后续程序不要对此单元操作。如：

```
SYSTEM_INIT:
.....
.....
LDWI 0x55
STR EEPROM_ADDR
LDWI 0xAA
STR EEPROM_DATA
LCALL EEPROM_write
LCALL EEPROM_write
.....
```

### 11.1 编程数据 EEPROM 步骤

- A. 把 INTCON 的 GIE 位清 0；
- B. 判断 GIE 是否为 1，是则重复 A 步骤，否则可以进行下一步；
- C. 往 EEADR 写入目标地址；
- D. 往 EEDAT 写入目标数据；
- E. 把位 WREN3/WREN2/WREN1 全部置 1；
- F. 把位 WR 置 1（EECON2.0，此后 WR 会维持高）；
- G. 写过程不能改变 WREN3/2/1 的值，否则编程终止；
- H. 等大概 2ms 之后编程自动完成，WR 自动清 0，WREN3、WREN2、WREN1 清 0；
- I. 如果想再次编程，重复步骤 C~H 即可；

例子 1：

```
BCR INTCON, GIE
BTSC INTCON, GIE
LJUMP $-2
BANKSEL EEADR
LDWI 55H
STREEADR _____:地址为 0x55
STREEDAT _____:数据为 0x55
LDWI 34H
STR EECON1 _____:WREN3/2/1 同时置 1
BSR EECON2, 0 _____:启动写
```



BSR INTCON, GIE ;把 GIE 置 1

例子 2 :

BCR INTCON, GIE

BTSC INTCON, GIE

LJUMP \$-2

BANKSEL EEADR

LDWI 55H

STR EEADR ;地址为 0x55

STR EEDAT ;数据为 0x55

LDWI 34H

STR EECON1 ;WREN3/2/1 同时置 1

NOP ;这里 NOP 可以换成其他指令

BSR EECON2, 0 ;启动写，实际上硬件不会启动编程 EEPROM 操作

BCR EECON1, WREN1 ;先清 WREN1，使得 WREN3/2/1 不同时为 1

BSR EECON1, WREN1 ;重新置位 WREN1，令 WREN3/2/1 同时为 1

BSR EECON2, 0 ;启动写，这次硬件将对 EEPROM 编程

BSR INTCON, GIE

注意：

1. 以上步骤的 E、F 两步必须是连续的两条指令周期完成，不能错开（如例子 2），否则编程操作不会启动，其中 WREN3、WREN2 和 WREN1 可以不是同一条指令置 1，比如可以用 BSR 指令分开对各位置 1；
2. 如果 E、F 两步被错开执行，要想启动下一次编程操作，必须在 E、F 之前加入一步，把 WREN3、WREN2 或者 WREN1 任意一位清 0，如例子 2；
3. 编程过程中读操作无效。

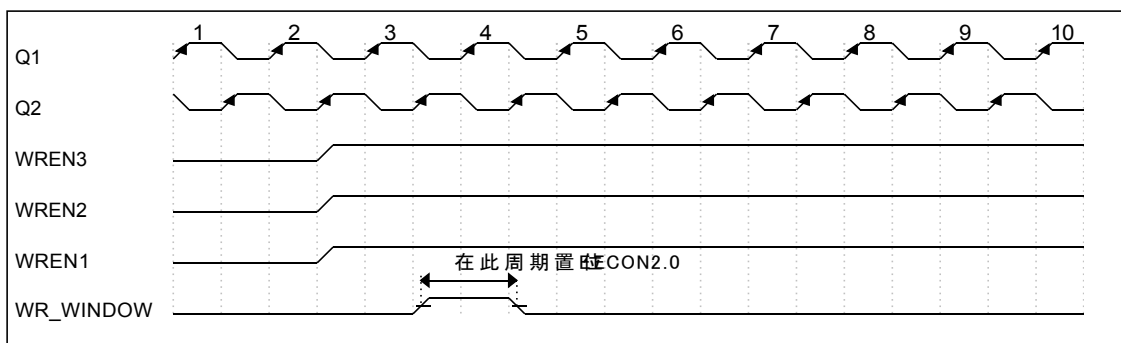


图 11.1 软件编程数据 EEPROM 时序

## 11.2 读数据 EEPROM

要读取数据存储单元，用户必须将地址写入 EEADR 寄存器，然后将 EECON1 寄存器的控制位 RD 置 1。在紧接着的下一周期，EEDAT 寄存器就被 EEPROM 数据写入。因此该数据可由下一条指令读取。EEDAT 将保持这个值直到用户下一次从该单元读取或向该单元写入数据时（在写操作过程中）。

下面是读取 EEPROM 的一段示例程序：

```

BANKSEL EEADR
LDWI dest_addr
STR EEADR
BSR EECON1, RD
LDR EEDAT, W
    
```

### 11.3 与数据 EEPROM 相关寄存器汇总

名称	地址	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	复位值
INTCON	0x0B/8B	GIE	PEIE	T0IE	INTE	PAIE	T0IF	INTF	PAIF	0000 0000
PIE1	0x8C	EEIE	CKMEAIE	-	C2IE	C1IE	OSFIE	TMR2IE	TMR1IE	0000 0000
PIR1	0x0C	EEIF	CKMEAIF	-	C2IF	C1IF	OSFIF	TMR2IF	TMR1IF	0000 0000
EEDAT	0x9A	EEDAT[7:0]								0000 0000
EEADR	0x9B	EEADR[7:0]								0000 0000
EECON1	0x9C	-	-	WREN3	WREN2	WRERR	WREN1	-	RD	--00 x 0 -0
EECON2	0x9D	-	-	-	-	-	-	-	WR	---- ---0

#### 11.3.1 EEDAT 寄存器，地址 0x9A

Bit	7	6	5	4	3	2	1	0
Name	EEDAT[7:0]							
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

#### 11.3.2 EEADR 寄存器，地址 0x9B

Bit	7	6	5	4	3	2	1	0
Name	EEADR[7:0]							
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

#### 11.3.3 EECON1 寄存器，地址 0x9C

Bit	7	6	5	4	3	2	1	0
Name	-	-	WREN3	WREN2	WRERR	WREN1	-	RD
Reset	-	-	0	0	x	0	-	0



Type	-	-	RW	RW	RW	RW	-	WO
------	---	---	----	----	----	----	---	----

Bit	Name	Function
7:6	-	保留位，读 0
5	WREN3	数据 EEPROM 写使能 3 和 WREN2、WREN1 结合使用
4	WREN2	数据 EEPROM 写使能 2 和 WREN3、WREN1 结合使用
3	WRERR	数据 EEPROM 写错误标志位 1：在 EEPROM 编程周期发生了看门狗或者外部复位，中止 0：在 EEPROM 编程周期正常完成
2	WREN1	数据 EEPROM 写使能 1 WREN3-1=111：允许软件对 EEPROM 编程，编程完成后各位自动清 0 WREN3-1=其他值：禁止软件对 EEPROM 编程
1	-	保留位，读 0
0	RD	数据 EEPROM 读控制位 此位是只写，读永远返回 0 写 1：启动一次数据 EEPROM 读周期 写 0：不启动读

### 11.3.4 EECON2 寄存器，地址 0x9D

Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	-	WR
Reset	-	-	-	-	-	-	-	0
Type	-	-	-	-	-	-	-	RW

Bit	Name	Function
7:1	-	保留位，读 0
0	WR	数据 EEPROM 写控制位 读操作： 1：数据 EEPROM 编程周期进行中 0：数据 EEPROM 不处于编程周期 写操作： 1：启动一次数据 EEPROM 编程周期 0：无意义

## 12 模数转换器 ( ADC ) 模块

模数转换器 (Analog-to-digital Converter, ADC) 可将模拟输入信号转换为相应的 10 位二进制表征值。该系列器件采用多个模拟输入复用到一个采样保持电路。采样保持电路的输出与转换器的输入相连接。转换器通过逐次逼近法产生 10 位二进制值, 并将转换结果保存在 ADC 结果寄存器 (ADRESL:ADRESH) 中。ADC 参考电压可用软件选择为 VDD 或施加在外部参考引脚上的电压。ADC 可在转换完成时产生中断。该中断可用于将器件从休眠唤醒。

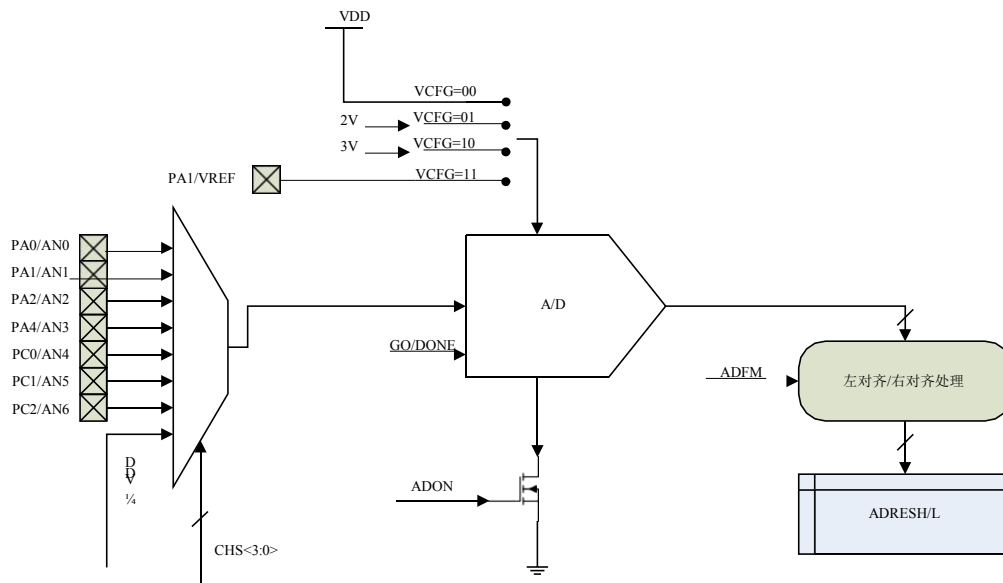


图 12.1 ADC 原理框图

### 12.1 ADC 的配置

配置和使用 ADC 时, 必须考虑以下功能:

- 端口配置
- 通道选择
- ADC 参考电压的选择
- ADC 转换时钟源
- 中断控制
- 转换结果的格式

#### 12.1.1 端口配置

ADC 可用于转换模拟和数字信号。转换模拟信号时, 应将相关的 TRIS 和 ANSEL 位置 1 将 I/O 引脚应配置为模拟功能。更多信息请参见相应的端口章节。

**注意:** 如果定义为数字输入的引脚上存在模拟电压, 可导致输入缓冲器传导过大的电流。



## 12.1.2 通道选择

ADCON0 寄存器的 CHS 位决定将哪个通道连接到采样保持电路。改变通道时，开始下一次转换前需要一个延时。更多信息请参见第 12.2 节“ADC 的工作原理”。

## 12.1.3 ADC 参考电压

ADCON0 寄存器的 VCFG 位提供对正参考电压的控制。正参考电压可以是 VDD，内部参考 2V 或者 3V，也可以是外部电压源。负参考电压始终连接到参考地。

## 12.1.4 转换时钟

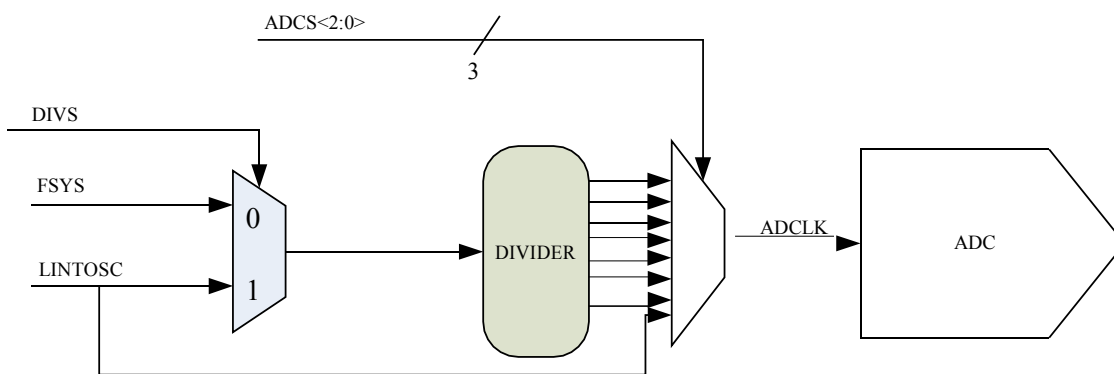


图 12.2 ADC 的时钟配置原理

转换时钟源可通过 ADCON1 寄存器的 ADCS 位用软件选择。有以下 13 种时钟选项：

- SYSCLK/2 或者 LFINTOSC/2
- SYSCLK/4 或者 LFINTOSC/4
- SYSCLK/8 或者 LFINTOSC/8
- SYSCLK/16 或者 LFINTOSC/16
- SYSCLK/32 或者 LFINTOSC/32
- SYSCLK/64 或者 LFINTOSC/64
- FRC（内部慢时钟振荡器）

完成一位（bit）的转换时间定义为 TAD。完成 10 位转换需要 11.5 个 TAD 周期（不包括采样时间），如图 12.3 和 12.5 所示。

进行正确的转换必须满足相应的 TAD 规范。更多信息请参见第 18 节“电气特性”中的 A/D 转换要求。表 12.1 所示为正确选择 ADC 时钟的示例。

**注意：**

1. 除非使用的是 FRC，否则任何系统时钟频率的变化均会改变 ADC 时钟频率，这将对 ADC 结果产生负面影响；
2. FRC 可以是 256KHz 或者是 32KHz，取决于 LFMOD 为何值。



ADC 时钟周期 ( $T_{AD}$ )		系统时钟频率 ( $F_{SYS}$ )			
ADC 时钟源	ADCS<2:0>	16MHz	8MHz	4MHz	1MHz
$F_{SYS}/2$	000	125ns	250ns	500ns	2.0us
$F_{SYS}/4$	100	250ns	500ns	1.0us	4.0us
$F_{SYS}/8$	001	500ns	1.0us	2.0us	8.0us
$F_{SYS}/16$	101	1.0us	2.0us	4.0us	16.0us
$F_{SYS}/32$	010	2.0us	4.0us	8.0us	32.0us
$F_{SYS}/64$	110	4.0us	8.0us	16.0us	64.0us
$F_{RC}$	x11	2~6us	2~6us	2~6us	2~6us

表 12.1 ADC 时钟周期和器件工作频率

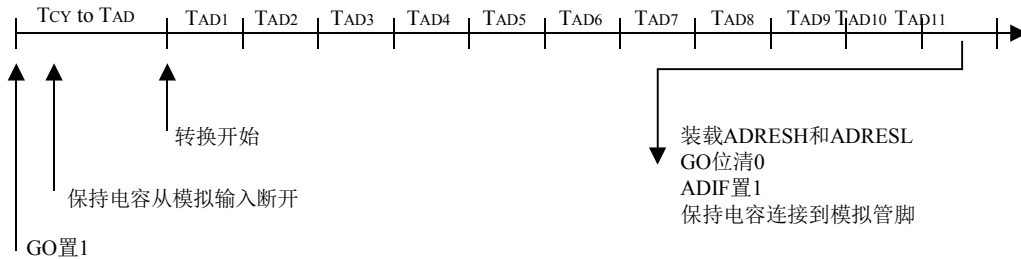


图 12.3 模数转换 TAD 周期

### 12.1.5 中断

ADC 模块可使中断在模数转换完成时产生。ADC 中断标志为 PIR1 寄存器中的 ADIF 位。ADC 中断使能为 PIE1 寄存器中的 ADIE 位。ADIF 位必须用软件清零。

**注意：**无论 ADC 中断是否被打开，ADIF 位在每次转换完成时均置 1。

器件工作或处于休眠状态时均可产生中断。如果器件处于休眠状态，中断将唤醒器件。从休眠唤醒时，始终执行 SLEEP 指令后的那条指令。如果用户试图唤醒器件并恢复顺序执行代码，必须禁止全局中断。如果允许全局中断，代码执行将转至中断服务程序。

### 12.1.6 转换结果的格式

10 位 A/D 转换结果有两种格式，即左对齐和右对齐。ADCON0 寄存器的 ADFM 位控制输出格式。

图 12.4 所示为两种输出格式。

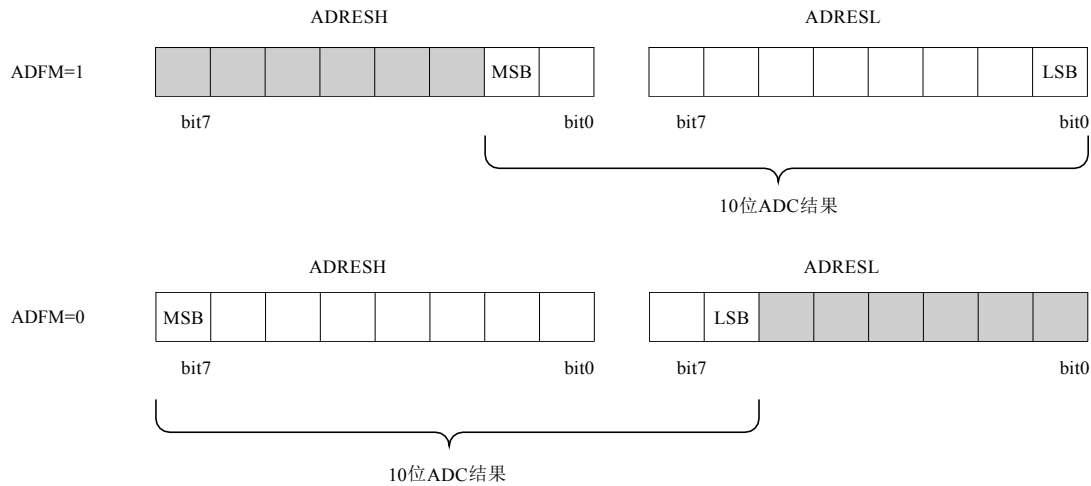


图 12.4 ADC 转换结果格式示意

## 12.2 ADC 的工作原理

### 12.2.1 启动转换

要启用 ADC 模块，必须将 ADCON0 寄存器的 ADON 位置 1。将 ADCON0 寄存器的 GO/DONE 位置 1 将启动模数转换。

**注意：**应在打开 ADC 的那条指令中将 GO/DONE 位置 1。请参见第 12.2.6 节 'A/D 转换步骤'。

### 12.2.2 转换完成

转换完成时，ADC 模块将：

- 将 GO/DONE 位清零
- 将 ADIF 标志位置 1
- 用新的转换结果更新 ADRESH:ADRESL 寄存器

### 12.2.3 终止转换

如果转换必须在完成前被终止，可用软件将 GO/DONE 清零。ADRESH:ADRESL 寄存器不会被未完成的模数转换采样更新。相反，ADRESH:ADRESL 这对寄存器将保持先前转换的值。此外，启动下一次采集前，需等待 TACQ（采样时间，图 12.4）的延时。延时结束后，所选通道的输入采集将自动启动。

**注意：**器件复位将强制所有寄存器回到其复位状态。这样，ADC 模块就被关闭，并且任何待处理的转换均被终止。

## 12.2.4 休眠模式下ADC的工作

ADC 模块可在休眠期间工作，这要求将 ADC 时钟源置于 FRC 选项。选定 FRC 时钟源后，软件可以置 MCU 于 SLEEP 模式，从而降低 ADC 转换期间的系统噪声。如果允许 ADC 中断，转换完成后器件将从休眠唤醒。如果禁止 ADC 中断，ADC 模块在转换完成后关闭，尽管 ADON 位保持置 1 状态。

如果 ADC 时钟源不是 FRC，执行一条 SLEEP 指令将使当前转换中止，并且 ADC 模块被关闭，尽管 ADON 位保持置 1 状态。

## 12.2.5 特殊事件触发器

ECCP 特殊事件触发器可在软件不干预的情况下周期性地 ADC 测量。发生触发事件时，GO/DONE 位由硬件置 1，Timer1 计数器复位为零。特殊事件触发器的使用并不确保正常 ADC 定时，用户必须确保满足 ADC 定时要求。

更多信息请参见第 9 节“增强型捕捉/比较/PWM+（带自动关闭和死区）模块”。

## 12.2.6 A/D 转换步骤

以下是使用 ADC 进行模数转换的步骤示例：

1. 配置端口：
  - 禁止引脚输出驱动器（见 TRIS 寄存器）
  - 将引脚配置为模拟
2. 配置 ADC 模块：
  - 选择 ADC 转换时钟
  - 配置参考电压
  - 选择 ADC 输入通道
  - 选择转换结果的格式
  - 打开 ADC 模块
3. 配置 ADC 中断（可选）：
  - 将 ADC 中断标志清零
  - 允许 ADC 中断
  - 允许外设中断
  - 允许全局中断
4. 等待所需稳定时间  $T_{ST}^{(1)}$ ；
5. 等待所需的采集时间  $T_{ACQ}^{(2)}$ ；
6. 将 GO/DONE 置 1 启动转换；
7. 通过以下情况之一等待 ADC 转换完成：
  - 查询 GO/DONE 位
  - 等待 ADC 中断（允许中断时）
8. 读取 ADC 结果；
9. 将 ADC 中断标志清零（在允许了中断的情况下这一步是必需的）。

以下是一段示例代码：

```

BANKSEL ADCON1 ;
LDWI B'01110000' ;ADC Frc clock
STR ADCON1 ;
BANKSEL TRISA ;
BSR TRISA,0 ;Set RA0 to input
BANKSEL ANSEL ;
BSR ANSEL,0 ;Set RA0 to analog
BANKSEL ADCON0 ;
LDWI B'10000001' ;Right justify,
STR ADCON0 ; Vdd Vref, AN0, On
LCALL StableTime ; ADC stable time
LCALL SampleTime ;Acquisiton delay
BSR ADCON0,GO ;Start conversion
BTSC ADCON0,GO ;Is conversion done?
LJUMP $-1 ;No, test again
BANKSEL ADRESH ;
LDR ADRESH,W ;Read upper 2 bits
STR RESULTHI ;store in GPR space
BANKSEL ADRESL ;
LDR ADRESL,W ;Read lower 8 bits
STR RESULTLO ;Store in GPR space
  
```

注意：

1. TST 时间是 ADC 的稳定时间，首次打开 ADC 的时候需要等待 TST+TACQ；
2. 见图 12.5，ADC 转换时序。

### 12.3 A/D 采集时间要求

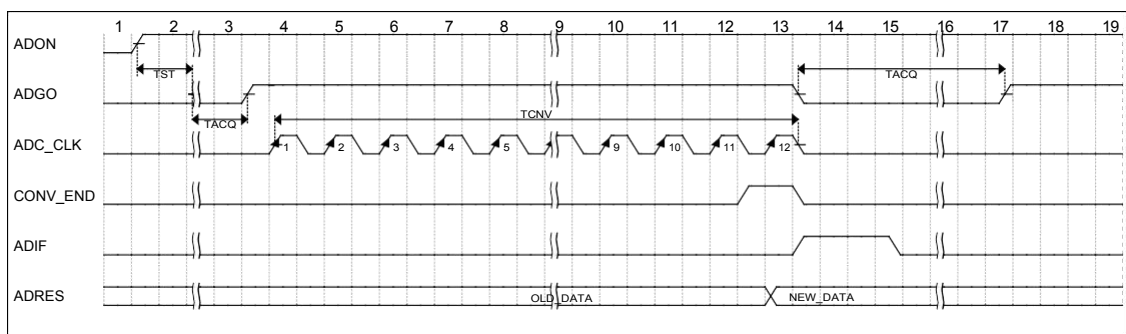
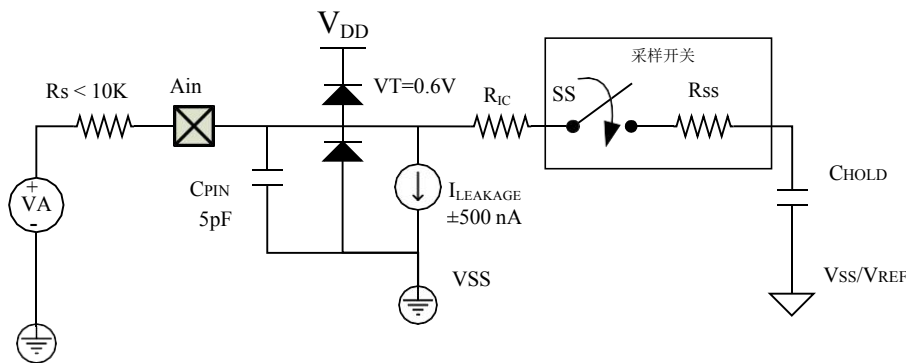


图 12.5 ADC 转换时序图

为了使 ADC 达到规定的精度，必须使充电保持电容（CHOLD）充满至输入通道的电平。模拟输入模型请参见图 12.6。源阻抗（RS）和内部采样开关（RSS）阻抗直接影响电容 CHOLD 的充电时间。采样开关（RSS）阻抗随器件电压（VDD）的变化而变化，参见图 12.6。建议模拟信号源的最大阻抗为 10kΩ。采集时间随着源阻抗的降低而缩短。在选择（或改变）模拟输入通道后，必须在开始转换前完成采集。



图注:

- CPIN = 输入电容
- VT = 门限电压
- ILEAKAGE = 结点漏电流
- RIC = 互联电阻
- SS = 采样开关
- CHOLD = 采样保持电容

图 12.6 模拟输入模型

## 12.4 与 ADC 相关寄存器汇总

名称	地址	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	复位值
ADCON0	0x1F	ADFM	VCFG1	VCFG0	CHS2	CHS1	CHS0	GO/DONE	ADON	000 0 00 00
ADCON1	0x9F	DIVS	ADCS2	ADCS1	ADCS0	-	-	-	-	0 00 0 -- --
ADRESH	0x1E	ADC 结果高 8 位								X X X X X X X X
ADRESL	0x9E	ADC 结果低 8 位								X X X X X X X X
ANSEL	0x91	ANSEL[7:0], 数模管脚选择								111 1 11 11
INTCON	0x0B/8B	GIE	PEIE	TOIE	INTE	PAIE	TOIF	INTF	PAIF	000 0 00 00
PIE1	0x8C	EEIE	CKMEAIE	-	C2IE	C1IE	OSFIE	TMR2IE	TMR1IE	000 0 00 00
PIR1	0x0C	EEIF	CKMEAIF	-	C2IF	C1IF	OSFIF	TMR2IF	TMR1IF	000 0 00 00
TRISA	0x85	TRISA[7:0], PORTA 方向控制								111 1 11 11
PORTA	0x05	PORTA[7:0], PORTA 数据寄存器								X X X X X X X X
TRISC	0x87	-	-	-	TRISC[5:0], PORTC 方向控制				-- 11 1 1 1 1	
PORTC	0x07	-	-	-	PORTC[5:0], PORTC 数据寄存器				- - X X X X X X	

### 12.4.1 ADCON0 寄存器 · 地址 0x1F

Bit	7	6	5	4	3	2	1	0
Name	ADFM	VCFG1	VCFG0	CHS2	CHS1	CHS0	GO/DONE	ADON
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7	ADFM	A/D 转换结果格式选择位 1 = 右对齐



		0 = 左对齐
6:5	VCFG[1:0]	参考电压选择位 11 = VREF 引脚 10 = 内部 3V 电压 01 = 内部 2V 电压 00 = VDD <b>注意：PC0 设置为模拟管脚才可以作为外部参考 VREF 输入</b>
4:2	CHS[2:0]	模拟通道选择位 000 = AN0 001 = AN1 010 = AN2 011 = AN3 100 = AN4 101 = AN5 110 = AN6 111 = 内部 1/4 VDD
1	GO/DONE	A/D 转换状态位 1 = A/D 转换正在进行。 将本位置 1 启动一次 A/D 转换。A/D 转换完成时此位由硬件自动清零 0 = A/D 转换完成/不在进行中
0	ADON	ADC 使能位 1 = 使能 ADC 0 = 禁止 ADC，不消耗工作电流

## 12.4.2 ADCON1 寄存器，地址 0x9F

Bit	7	6	5	4	3	2	1	0
Name	DIVS	ADCS2	ADCS1	ADCS0	-	-	-	-
Reset	0	0	0	0	-	-	-	-
Type	RW	RW	RW	RW	-	-	-	-

Bit	Name	Function
7	DIVS	A/D 分频时钟源选择位 1 = 分频时钟选择慢时钟 0 = 分频时钟选择 FOSC
6:4	ADCS[2:0]	A/D 转换时钟选择位 <b>当 DIVS 为 0 时：</b> 000 = FOSC/2 001 = FOSC/8 010 = FOSC/32 x11 = FRC (时钟来自内部振荡器，32KHz 或者 256KHz) 100 = FOSC/4 101 = FOSC/16 110 = FOSC/64



		<p>当 DIVS 为 1 时 :</p> <p>000=LFINTOSC /2</p> <p>001=LFINTOSC /8</p> <p>010 = LFINTOSC /32</p> <p>x11 = FRC (时钟来自内部振荡器, 32KHz 或者 256KHz)</p> <p>100 = LFINTOSC /4</p> <p>101=LFINTOSC /16</p> <p>110=LFINTOSC /64</p>
3:0	-	<p>未实现, 读 0</p> <p>软件不要向这些位写 1</p>

### 12.4.3 ADRESH 寄存器, 地址 0x1E

当 ADFM = 0 时 :

Bit	7	6	5	4	3	2	1	0
Name	ADRESH							
Reset	x	x	x	x	x	x	x	x
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:0	ADRES[9:2]	<p>ADC 结果寄存器位</p> <p>10 位转换结果的高 8 位</p>

当 ADFM = 1 时 :

Bit	7	6	5	4	3	2	1	0
Name	-						ADRESH	
Reset	-						x	x
Type	RO-0						RW	RW

Bit	Name	Function
7:2	-	未实现, 读 0
1:0	ADRES[9:8]	<p>ADC 结果寄存器位</p> <p>10 位转换结果的高 2 位</p>

### 12.4.4 ADRESL 寄存器, 地址 0x9E

当 ADFM = 0 时 :

Bit	7	6	5	4	3	2	1	0
Name	ADRESL		-					
Reset	x	x	-					
Type	RW	RW	RO-0					





Bit	Name	Function
7:6	ADRES[1:0]	ADC 结果寄存器位 10 位转换结果的低 2 位
5:0	-	未实现，读 0

当 ADFM = 1 时：

Bit	7	6	5	4	3	2	1	0
Name	ADRESL							
Reset	x	x	x	x	x	x	x	x
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:0	ADRES[7:0]	ADC 结果寄存器位 10 位转换结果的低 8 位

### 12.4.5 ANSEL 寄存器，地址 0x90

Bit	7	6	5	4	3	2	1	0
Name	ANSEL							
Reset	1	1	1	1	1	1	1	1
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:0	ANSEL[7:0]	模拟选择位 在 AN<7:0>引脚上分别进行模拟或数字功能的模拟选择 1 = 模拟输入，引脚被分配为模拟输入 0 = 数字 IO，引脚被分配给端口或者特殊功能 <b>注意：当 ADC 配置为采样内部 1/4 VDD 通道时，ANSEL&lt;7&gt;要设置为 1</b>

注意：将某引脚设置为模拟输入将自动禁止数字输入电路、弱上拉以及电平变化中断（如果有的话）。相应 TRIS 位必须设置为输入模式以允许对该引脚的电压进行外部控制。

## 13 慢时钟测量模式

芯片集成了两个内部 RC 振荡器，一个是经过出厂校准的高速高精度的 16M 快时钟 HFINTOSC，一个是低速低功耗的 32K 时钟 LFINTOSC，利用慢时钟测量功能可以把 LFINTOSC 的周期用系统时钟计算出来。此功能可以比较精准的测量内部慢时钟周期。

### 13.1 使用方法

在此模式下，TIMER2 的预分频、后分频配置自动变为 1:1，组成一个 12 位的定时器，TIMER2 的计数时钟为系统时钟 Fosc，而不是普通模式下的指令时钟。计数结束后结果自动存到 SOSCPR 寄存器，其单位是系统时钟 Fosc 的个数。

#### 操作步骤：

1. 为提高计量精度，建议设置 IRCF 为 111，SCS=1，选择 16M 的系统时钟；
2. 把 T2CON.2 置 1，使能 TIMER2；
3. 如果选择 4 次平均，则把 MSCKCON.2 置 1，否则把它清 0；
4. 置位 MSCKCON.1，开始测量；
5. 测量结束后 MSCKCON.1 自动清 0，中断标志置 1；
6. 可以用查询或中断的方式等待结束；
7. 当查询到中断标志为 1 时读取得到的 SOSCPR 即为最终结果。

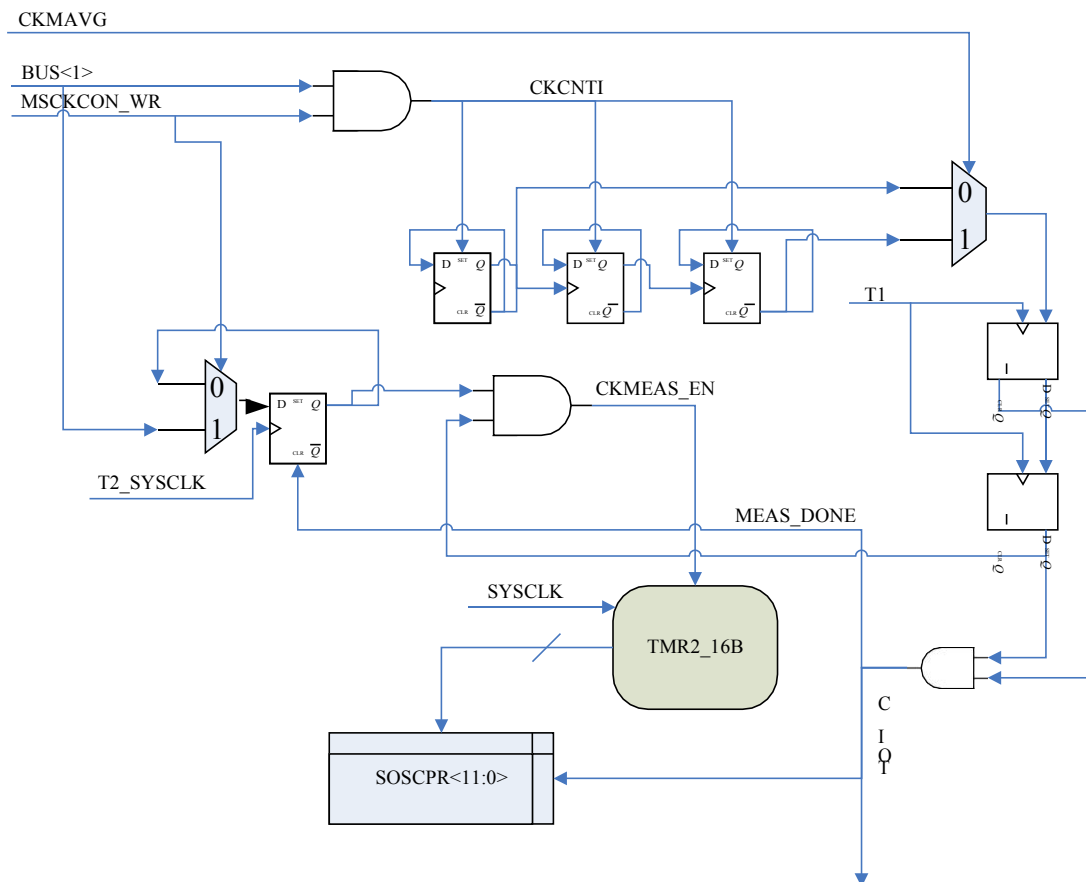


图 13.4 慢时钟测量模式原理框图

注意：

1. 在慢时钟测量过程中软件不要写 SOSCPRH/L；
2. 不要在单步调试下做慢时钟测量，因为暂停模式下 TIMER2 被停止，这样会导致测量结果不正确；

## 13.2 与 I/O 计时模式相关寄存器汇总

名称	地址	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	复位值
MSCKCON	0x1B	-	VREG_OE	T2CKSRC	SLVREN	-	CKMAVG	CKCNTI	-	0 0 0 0 - 0 0 -
SOSCPRL	0x1C	SOSCPR[7:0]								1 1 1 1 1 1 1 1
SOSCPRH	0x1D	-				SOSCPR[11:8]				- - - - 1 1 1 1
ANSEL	0x91	ANSEL[7:0]，数模管脚选择								1 1 1 1 1 1 1 1
INTCON	0x0B/8B	GIE	PEIE	TOIE	INTE	PAIE	TOIF	INTF	PAIF	0 0 0 0 0 0 0 0
PIE1	0x8C	EEIE	CKMEAIE	-	C2IE	C1IE	OSFIE	TMR2IE	TMR1IE	0 0 0 0 0 0 0 0
PIR1	0x0C	EEIF	CKMEAIF	-	C2IF	C1IF	OSFIF	TMR2IF	TMR1IF	0 0 0 0 0 0 0 0
TRISA	0x85	TRISA[7:0]，PORTA 方向控制								1 1 1 1 1 1 1 1
PORTA	0x05	PORTA[7:0]，PORTA 数据寄存器								x x x x x x x x
TRISC	0x87	-	-	TRISC[5:0]，PORTC 方向控制				-		- - 1 1 1 1 1 1
PORTC	0x07	-	-	PORTC[5:0]，PORTC 数据寄存器				-		- - x x x x x x

### 13.2.1 MSCKCON 寄存器，地址 0x1B

Bit	7	6	5	4	3	2	1	0
Name	-	VREG_OE	T2CKSRC	SLVREN	-	CKMAVG	CKCNTI	-
Reset	0	0	0	0	-	0	0	-
Type	RW	RW	RW	RW	-	RW	RW	-

Bit	Name	Function
7	-	<b>保留位</b>
6	VREG_OE	稳压器输出使能 1 = 打开片内稳压器，PA4 和 PC5 输出稳压器电压 0 = PA4 和 PC5 为普通 IO
5	T2CKSRC	TIMER2 时钟源选择 0 = TIMER2 时钟源为系统时钟 1 = TIMER2 时钟源为内部 32MHz
4	SLVREN	软件控制 LVR 使能位，当 UCFG1<1:0>为 01 时： 1 = 打开 LVR 0 = 禁止 LVR 当 UCFG1<1:0>不为 01 时，此位无实际意义



		注意：发生欠压复位时，该位不会清 0。其它任何复位都可将其清 0
3	-	保留位，不能写 1
2	CKMAVG	快时钟测量慢时钟周期的测量平均模式 1 = 打开平均模式（自动测量并累加 4 次） 0 = 关闭平均模式
1	CKCNTI	Clock Count Init -使能快时钟测量慢时钟周期 1 = 使能快时钟测量慢时钟周期 0 = 关闭快时钟测量慢时钟周期 注：这一位在测量完毕后会 自动归零
0	-	保留位，不能写 1

### 13.2.2 SOSCPR 寄存器，地址 0x1C，1D

SOSCPRL，地址 0x1C

Bit	7	6	5	4	3	2	1	0
Name	SOSCPR[7:0]							
Reset	8 hff							
Type	RW							

SOSCPRH，地址 0x1D

Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	SOSCPR[11:8]			
Reset	-	-	-	-	4' hf			
Type	-	-	-	-	RW			

Bit	Name	Function
0x1C : 7:0 0x1D : 3:0	SOSCPR[11:0]	低速振荡器周期（单位：快时钟周期数） 用于慢时钟测量功能



## 14 中断模式

WF8M165 有以下中断源：

- PC1/INT 管脚进来的外部中断
- Timer0 溢出中断
- PORTA 电平变化中断
- Timer2 比对相等中断
- EEPROM 数据写中断
- 故障保护时钟监控器中断
- 比较器中断
- TIMER3/4/5 中断
- ECCP 相关中断

中断控制寄存器（INTCON）和外围中断请求寄存器（PIR1）记录了中端标志位。INTCON 同时也包含全局中断使能位 GIE。

当中断被服务后，以下动作自动发生：

- GIE 被清零，从而关闭中断
- 返回地址被推上堆栈
- 程序指针加载 0004h 地址

中断返回指令，RETI 将退出中断函数时同时设置 GIE 位，重新使能未屏蔽的中断。需要注意的是，执行中断返回 RETI 之前应该把相关的中断标志位清 0，以免重复进入中断处理程序。

INTCON 寄存器包含以下中断标志位：

- INT 管脚中断
- PORTA 变化中断
- Timer0 溢出中断

PIR1/2 中包含着外围中断标志位，PIE1/2 中包含着其对应的中断使能位，具体请参照寄存器各位的描述。

### 14.1 INT 中断

INT 引脚上的外部中断是边沿触发的；当 OPTION 寄存器的 INTEDG 位被置 1 时在上升沿触发，而当 INTEDG 位被清零时在下降沿触发。当 INT 引脚上出现有效边沿时，INTCON 寄存器的 INTF 位置 1。可以通过将 INTCON 寄存器的 INTE 控制位清零来禁止该中断。在重新允许该中断前，必须在中断服务程序中先用软件将 INTF 位清零。如果 INTE 位在进入休眠状态前被置 1，则 INT 中断能将处理器从休眠状态唤醒。

**注意：**

1. 使用 INT 中断时，必须对 ANSEL 和 CMCON0 寄存器进行初始化，以将模拟通道配置为数字输入。否则配置为模拟输入的引脚总是读为 0。

## 14.2 PORTA 电平变化中断

PORTA 输入电平的变化会使 INTCON 寄存器的 PAIF 位置 1。可以通过置 1/清零 INTCON 寄存器的 PAIE 位来使能/禁止该中断。此外，可通过 IOCA 寄存器对该端口的各个引脚进行配置。

注意：

1. 使用 PORTA 电平变化中断时，必须对 ANSEL 和 CMCON0 寄存器进行初始化，以将模拟通道配置为数字输入。配置为模拟输入的引脚总是读为 0；
2. 初始化电平变化中断时，应先配置为数字输入 IO，把相应的 IOCA 置 1，然后读取一下该 PORTA；
3. 当 IO 电平发生变化时，PAIF 被置 1；
4. 清中断标志位之前应该读取一下 PORTA，然后再对 PAIF 清 0；

## 14.3 中断响应

外部中断包括 INT 管脚进来的或者 PORTA 变化中断的延时一般为 1 到 2 个指令周期。具体视中断发生的实际情况而定。

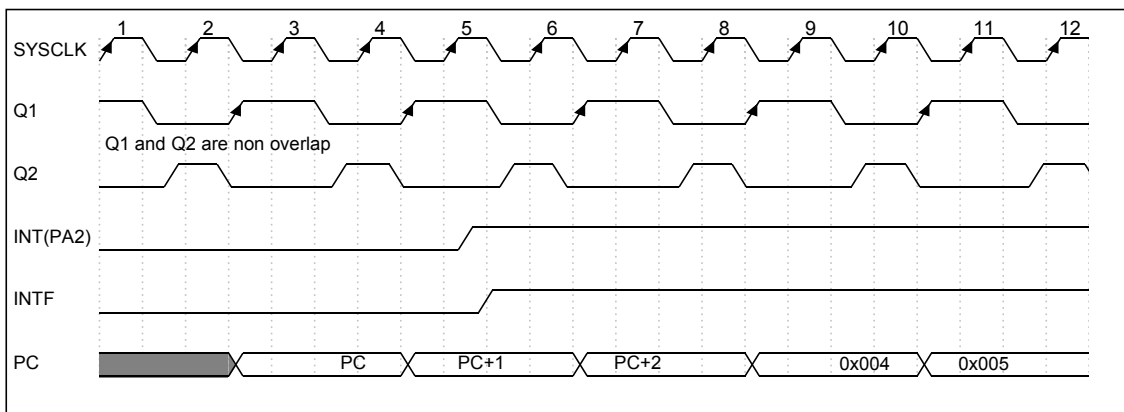


图 14.1 中断响应时序图

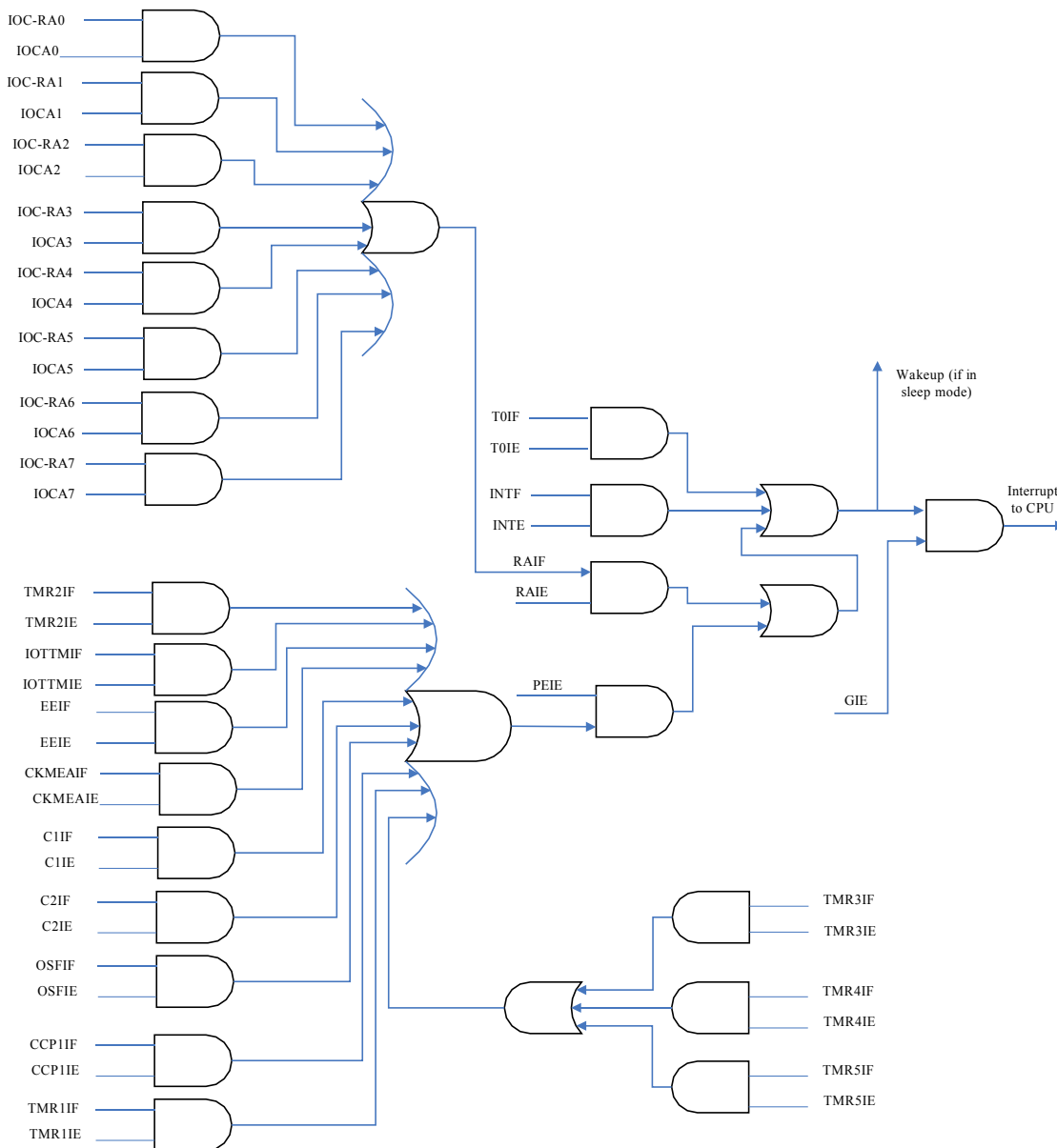


图 14.2 中断产生电路架构框图

## 14.4 中断过程中的现场保存

在中断过程中，只有返回 PC 被自动保存在堆栈上。一般来说，用户可能需要保存重要的寄存器值在堆栈上，例如 W，STATUS 寄存器等。这些必须由软件来完成。临时寄存器 W\_TEMP 和 STATUS\_TEMP 应该被放置在 GPR 的最后 16byte 里。这 16 个 GPR 落在两个页区间，因此可以稍微节省代码。

## 14.5 与中断相关寄存器汇总

名称	地址	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	复位值
INTCON	0x0B/8B	GIE	PEIE	T0IE	INTE	PAIE	T0IF	INTF	PAIF	0000 0 000
PIE1	0x8C	EEIE	CKMEAIE	-	C2IE	C1IE	OSFIE	TMR2IE	TMR1IE	0000 0 000



PIR1	0x0C	EEIF	CKMEAIF	-	C2IF	C1IF	OSFIF	TMR2IF	TMR1IF	0000 0 000
PIE2	0x8D	-	-	-	-	-	-	ADIE	CCP1IE	---- -- 00
PIR2	0x0C	-	-	-	-	-	-	ADIF	CCP1IF	---- -- 00
TRISA	0x85	TRISA[7:0]，PORTA 方向控制								1111 1 111
ANSEL	0x91	ANSEL[7:0]，数模管脚选择								1111 1 111
IOCA	0x96	IOCA[7:0]，端口中断允许位								0000 0 000

### 14.5.1 INTCON 寄存器，地址 0x0B

Bit	7	6	5	4	3	2	1	0
Name	GIE	PEIE	TOIE	INTE	PAIE	TOIF	INTF	PAIF
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7	GIE	<b>GIE:</b> Global Interrupt Enable bit 全局中断使能 1 = Enables all unmasked interrupts 0 = Disables all interrupts
6	PEIE	<b>PEIE:</b> Peripheral Interrupt Enable bit 外设中断使能 1 = Enables all unmasked peripheral interrupts 0 = Disables all peripheral interrupts
5	TOIE	<b>TOIE:</b> Timer0 Overflow Interrupt Enable bit 定时器0溢出中断使能 1 = Enables the Timer0 interrupt 0 = Disables the Timer0 interrupt
4	INTE	<b>INTE:</b> PC1/INT External Interrupt Enable bit 外部中断使能 1 = Enables the PC1/INT external interrupt 0 = Disables the PC1/INT external interrupt
3	PAIE	<b>PAIE:</b> PORTA Change Interrupt Enable bit(1) 1 = Enables the PORTA change interrupt 0 = Disables the PORTA change interrupt
2	TOIF	<b>TOIF:</b> Timer0 Overflow Interrupt Flag bit(2) 定时器0溢出中断标志位 1 = Timer0 register has overflowed (must be cleared in software) 0 = Timer0 register did not overflow
1	INTF	<b>INTF:</b> PC1/INT External Interrupt Flag bit 外部中断标志位 1 = The PC1/INT external interrupt occurred (must be cleared in software) 0 = The PC1/INT external interrupt did not occur
0	PAIF	<b>PAIF:</b> PORTA Change Interrupt Flag bit 1 = When at least one of the PORTA <5:0> pins changed state (must be cleared in software) 0 = None of the PORTA <5:0> pins have changed state





### 14.5.2 PIR1 寄存器，地址 0x0C

Bit	7	6	5	4	3	2	1	0
Name	EEIF	CKMEAIF	-	C2IF	C1IF	OSFIF	TMR2IF	TMR1IF
Reset	0	0	-	0	0	0	0	0
Type	RW	RW	R-0	RW	RW	RW	RW	RW

Bit	Name	Function
7	EEIF	<b>EEIF:</b> EE写中断标志位 1 = EE 写操作完成（必须软件清零） 0 = EE 写操作未完成
6	CKMEAIF	<b>CKMEAIF:</b> 快时钟测量慢时钟操作完成中断标志位 1 = 快时钟测量慢时钟操作完成（必须软件清零） 0 = 快时钟测量慢时钟未完成
5	-	<b>保留位</b>
4	C2IF	比较器 2 中断标志位 1 = 比较器 2 输出发生了变化 0 = 比较器 2 输出未发生改变
3	C1IF	比较器 1 中断标志位 1 = 比较器 1 输出发生了变化 0 = 比较器 1 输出未发生改变
2	OSFIF	振荡器故障中断标志位 1 = 系统振荡器发生故障，时钟输入切换为 INTOSC（必须用软件清 0） 0 = 系统时钟运行正常
1	TMR2IF	<b>TMR2IF:</b> Timer2与PR2比较相等中断标志位 1 = timer2的值等于PR2（必须软件清零） 0 = timer2 的值不等于 PR2
0	TMR1IF	TIMER1 溢出中断标志位 1 = TIMER1 发生了溢出 0 = TIMER1 未发生溢出

### 14.5.3 PIR2 寄存器，地址 0x0D

Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	ADIF	CCP1IF
Reset	-	-	-	-	-	-	0	0
Type	-	-	-	-	-	-	RW	RW

Bit	Name	Function
7:2	-	未实现，读 0 软件不要向这些位写 1
1	ADIF	AD中断标志位



		1 = AD转换完成 0 = AD 转换未完成或者没启动
0	CCP1IF	CCP1 中断标志位 捕捉模式： 1 = 发生了 TMR1 寄存器捕捉（必须用软件清 0） 0 = 未发生 TMR1 寄存器捕捉 比较模式： 1 = 发生了 TMR1 寄存器比较匹配（必须用软件清 0） 0 = 未发生 TMR1 寄存器比较匹配 PWM 模式： 此模式下该位未使用

### 14.5.4 PIE1 寄存器，地址 0x8C

Bit	7	6	5	4	3	2	1	0
Name	EEIE	CKMEAIE	-	C2IE	C1IE	OSFIE	TMR2IE	TMR1IE
Reset	0	0	-	0	0	0	0	0
TYPE	RW	RW	R-0	RW	RW	RW	RW	RW

Bit	Name	Function
7	EEIE	EE写中断使能位 1 = 使能EE 写操作完成中断 0 = 关闭 EE 写操作完成中断
6	CKMEAIE	快时钟测量慢时钟操作完成中断使能位 1 = 使能快时钟测量慢时钟操作完成中断 0 = 关闭快时钟测量慢时钟操作完成中断
5	-	<b>保留位</b>
4	C2IE	比较器 2 中断允许位 1 = 允许比较器 2 中断 0 = 禁止比较器 2 中断
3	C1IE	比较器 1 中断允许位 1 = 允许比较器 1 中断 0 = 禁止比较器 1 中断
2	OSFIE	振荡器故障中断允许位 1 = 允许振荡器故障中断 0 = 禁止振荡器故障中断
1	TMR2IE	Timer2 与PR2比较相等中断使能位 1 = 使能 timer2的值等于PR2 中断 0 = 关闭使能 timer2 的值等于 PR2 中断
0	TMR1IE	Timer1 溢出中断使能位 1 = 允许 Timer1 溢出中断 0 = 禁止 Timer1 溢出中断



### 14.5.5 PIE2 寄存器，地址 0x8D

Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	ADIE	CCP1IE
Reset	-	-	-	-	-	-	0	0
TYPE	-	-	-	-	-	-	RW	RW

Bit	Name	Function
7:2	-	未实现，读 0 软件不要向这些位写 1
1	ADIE	A/D转换器中断使能位 1 = 允许A/D转换器中断 0 = 禁止 A/D 转换器中断
0	CCP1IE	CCP1 中断使能位 1 = 允许 CCP1 溢出中断 0 = 禁止 CCP1 溢出中断

### 14.5.6 IOCA 寄存器，地址 0x96

Bit	7	6	5	4	3	2	1	0
Name	IOCA[7:0]							
Reset	§ h00							
Type	RW							

Bit	Name	Function
7:0	IOCA[7:0]	端口状态触发中断设置 1 = 使能端口状态触发中断 0 = 关闭端口状态触发中断



## 15 睡眠省电模式

芯片在执行完 SLEEP 指令后进入睡眠状态。

为了达到最低睡眠功耗，软件应该将所有 IO 置高或低，而且没有外部电路从 IO 耗电。I/O 作为输入的，外部电路应将其拉高或拉低，避免翻转耗电。/MCLR 应该在高电平。

### 15.1 唤醒模式

以下事件可以唤醒芯片：

- /MCLR 管脚上有外部复位
- WDT 超时
- PC1/INT 管脚上有中断，PORTA 变化或其他外围中断
- ADC 转换完成

清看门狗指令 CLRWDT、SLEEP（进入睡眠模式）或者从睡眠模式唤醒，都将清除看门狗计数器。

### 15.2 看门狗唤醒

看门狗工作在内部慢时钟（32KHz），它是一个 16 位的计数器，和定时器 0 共用一个 8 位的预分频器，使能位位于配置寄存器 UCFG0 的第 3 位，WDTEN，为 1 时表示使能看门狗，为 0 时将由 SWDTEN 位决定使能与否，SWDTEN 位于 WDTCN 寄存器。

清看门狗指令 CLRWDT、SLEEP 会清除看门狗计数器。

在使能了看门狗的情况下，MCU 睡眠时看门狗溢出事件可以作为一个唤醒源，而 MCU 正常工作时作为一个复位源。

## 16 I/O 端口

本芯片共包含 16 个 GPIO。这些 IO 除了作为普通输入/输出端口以外还通常具备一些与内核周边电路通讯的功能，具体见下。

### 16.1 PORTA 端口和 TRISA 寄存器

PORTA 是一个 8 位双向端口。与其相应的进出方向寄存器就是 TRISA 寄存器。反之，将某一位设置为“0”会将该对应 PORTA 端口设置为输出端口。在置为输出端口时，输出驱动电路会被打开，输出寄存器里的数据会被放置到输出端口。当 IO 处于输入状态时（TRISA=1），对 PORTA 进行读动作，PORTA 内容会是反映输入端口的状态。在 PORTA 上进行写动作时，PORTA 内容会被写入输出寄存器。所有的写操作都是“读-更改-写”这样一个微流程，即数据被读，然后更改，再写入输出寄存器的过程。

当 MCLRE 为 1 时，PORTA[5] 读的值为 0，此时它是作为外部复位管脚。

### 16.2 端口的其他功能

芯片在 PORTA 的每个端口都有一个状态变化中断选项和弱上拉选项。

#### 16.2.1 弱上拉

PORTA 和 PORTC 的每个端口（除了 PORTA[5]）都有一个可以单独设置的内部弱上拉功能。控制 WPUAx 寄存器里的位就可使能或关断这些弱上拉电路。当 GPIO 被设置为输出时，这些弱上拉电路会被自动关断。弱上拉电路在上电复位期间可以被置为关断。这是由 OPTION 寄存器中的 /PAPU 位决定的。PORTA[5] 内部也有弱上拉功能，它是在将 PORTA[5] 设置为 /MCLR 功能时自动使能的。当 PORTA[5] 被设置为 GPIO 时，该弱上拉电路由 WPUA5 控制。

#### 16.2.2 弱下拉

PORTA[4] 和 PORTC[3:1] 这 4 个 IO 配置为数字输入管脚时具有内部弱下拉功能，由寄存器 WPD 控制。当弱上拉和弱下拉同时在这些 IO 打开时，弱下拉将被禁止，弱上拉起作用。

#### 16.2.3 ANSEL 寄存器

ANSEL 寄存器用于控制 IO 的数模输入，当 ANSEL.x 为 1 时，对应的 IO 口为模拟引脚，IO 的输入上拉自动禁止，软件读该 IO 返回的是 0。

ANSEL 位对数字输出驱动没有影响，换言之，TRIS 位的优先级更高，即当 TRIS.x 为 0 时，不管 ANSEL.x 是 0 还

是 1，对应的 IO 为数字输出 IO。要想配置真正的模拟管脚，TRIS.x 要置 1，把数字输出驱动关闭。

### 16.2.4 状态变化中断

PORTA 的每个端口都可以被单独设置成一个中断源(端口状态变化触发中断)。控制 IOCAx 寄存器里的位就可使能或关闭这些端口的中断功能。端口状态变化触发中断的功能在上电复位时无效的。

当端口状态变化触发中断的功能被使能时，当前端口电平值会被与上次读动作所读取数据寄存器的旧值作对比。所有错误匹配结果会被或在一起形成 INTCON 寄存器中的 PAIF 标志位。

该中断可以将芯片从睡眠状态中唤醒。用户需要在中断服务程序中执行以下程序来清除该标志位：

- A) 对 PORTA 进行一次读或写得动作，这将结束任何错误匹配的状态。
- B) 清零 PAIF 标志位。

错误匹配的条件会一直设置 PAIF 位。对 PORTA 做一次读就可以结束任何错误匹配的状态，使得 PAIF 能被清零。数据寄存器里保持的上一次读的值不会被 MCLR 或低电压复位所影响。只要错误匹配状态存在，PAIF 位就会被置 1。

### 16.3 ECCP 和 PWM 的管脚复用

由脚位图（图 1.3）可以知道，PWM4、PWM5 分别和 PC3/P1C、PC2/P1D 复用，PWM4 和 P1C，PWM5 和 P1D 是不能同时输出的，ECCP 的优先级比 PWM4/5 高。换言之，如果 ECCP 被配置为全桥模式，即 P1C、P1D 有效，这时不管 P4EN、P5EN 为何值，PC3、PC2 将输出 P1C、P1D 脉冲。

只有当处于非全桥模式时，PWM4、PWM5 脉冲才可以输出，取决于 P4EN、P5EN 的值。

C2OUT/P1B/PWM3 这三个数字输出复用在 PC4 管脚，PC4 管脚配置如下表：

PC4 输出	条件
C2OUT	CM[2:0]=110，TRISC.4=0，P3EN=0 且： 1. ECCP 处于非 PWM 模式，或者 2. ECCP 处于 PWM 单输出模式
P1B	CM[2:0] ≠ 110，TRISC.4=0，且： 1. ECCP 处于 PWM 模式且 P1M ≠ 0
PWM3	CM[2:0] ≠ 110，TRISC.4=0，且： 1. ECCP 处于非 PWM 模式，或者 2. ECCP 处于 PWM 单输出模式

### 16.3.1 端口描述

PORTA 的每个端口都包含着不同的复用功能，其具体功能和控制在这一节里描述。

#### 16.3.1.1 PORTA[0], PORTA[1]

图 16.1 描述了此端口的内部电路结构。PA[0]/PA1 可以被配置为以下功能端口：

- GPIO
- 比较器 1 正输入
- ADC 通道输入
- 调试串口时钟 (PA0)，调试串口数据 (PA1)

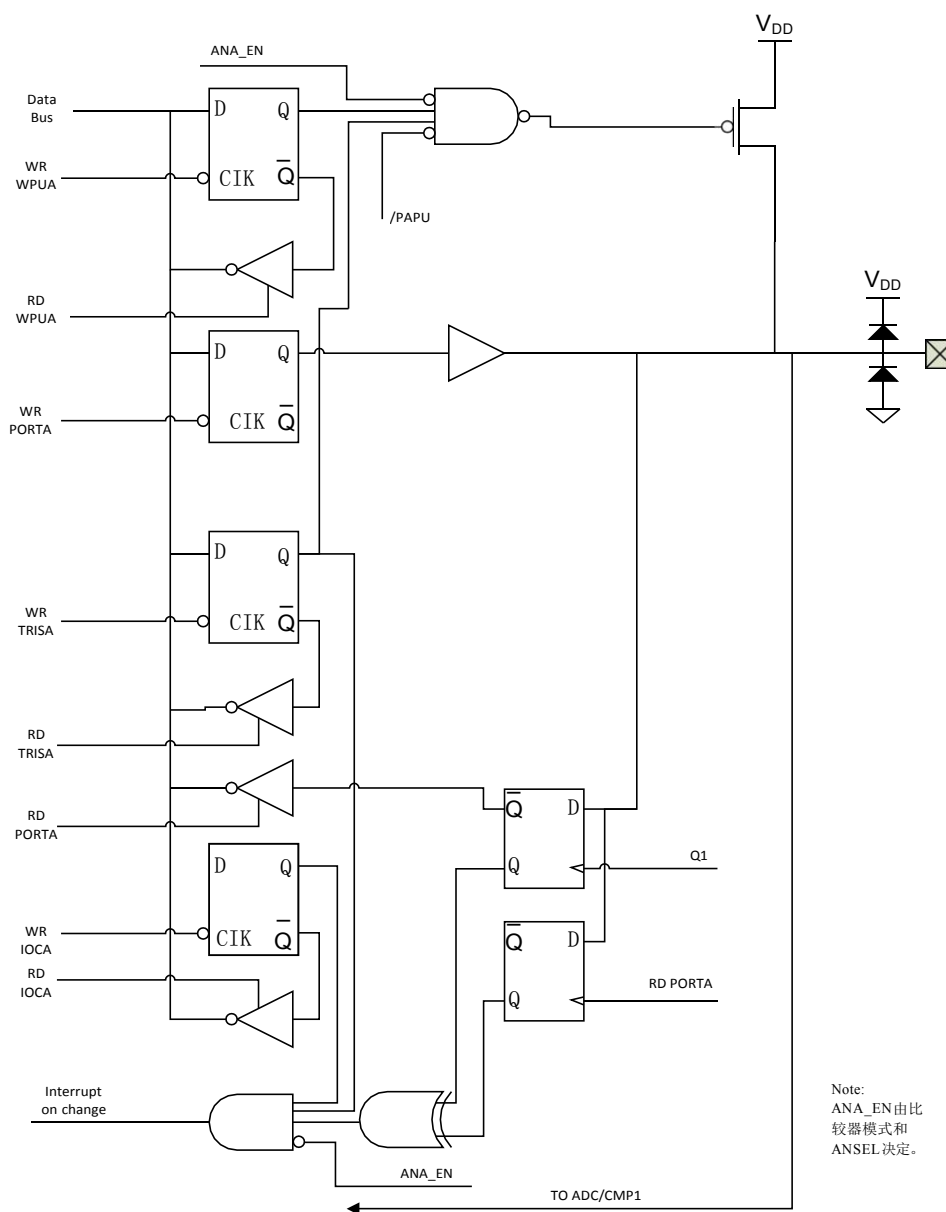


图 16.1 PA0 和 PA1 架构框图

### 16.3.1.2 PORTA[2]

图 16.2 描述了此端口的内部电路结构。PA[2]可以被配置为以下功能端口：

- GPIO
- 比较器 1 输出
- ADC 通道输入
- TIMER0 外部时钟源

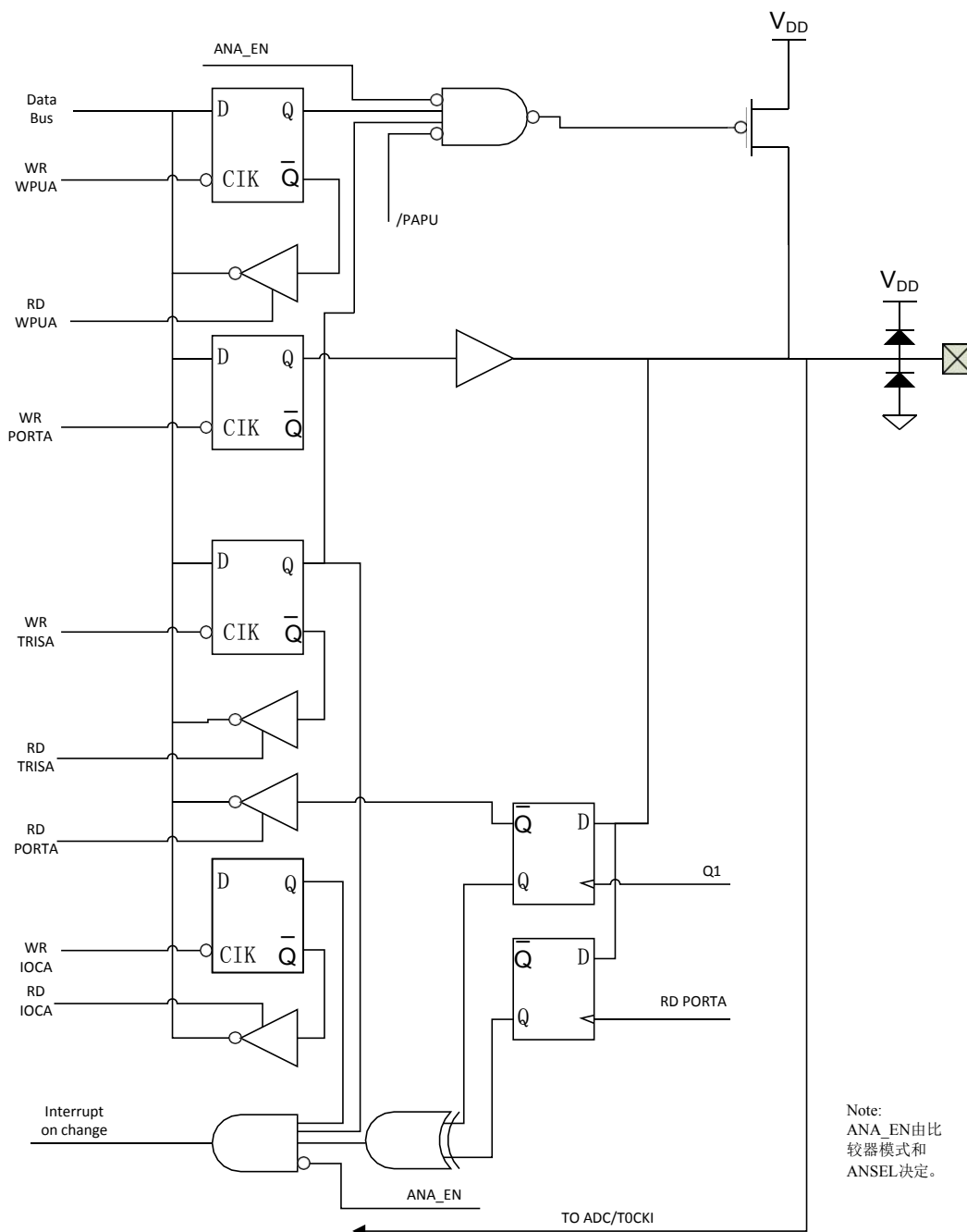


图 16.2 PA2 架构框图



### 16.3.1.3 PORTA[3]

图 16.3 描述了此端口的内部电路结构。PA[3]可以被配置为以下功能端口：

- GPIO
- ADC 通道输入
- 模拟测试输出

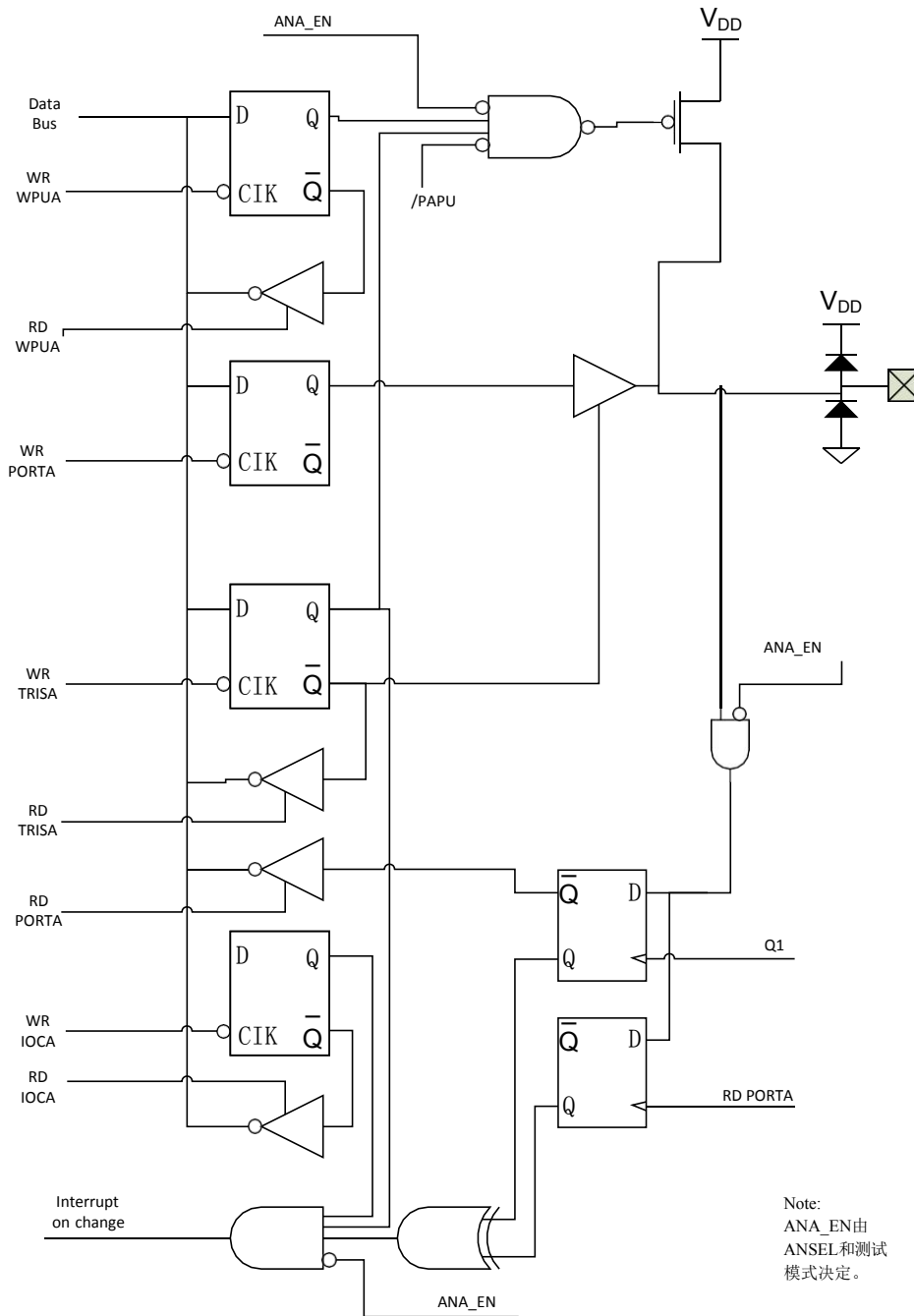


图 16.3 PA3 架构框图

### 16.3.1.4 PORTA[4]

图 16.4 描述了此端口的内部电路结构。PA[4]可以被配置为以下功能端口：

- GPIO
- 稳压器输出
- 模拟测试信号输入输出

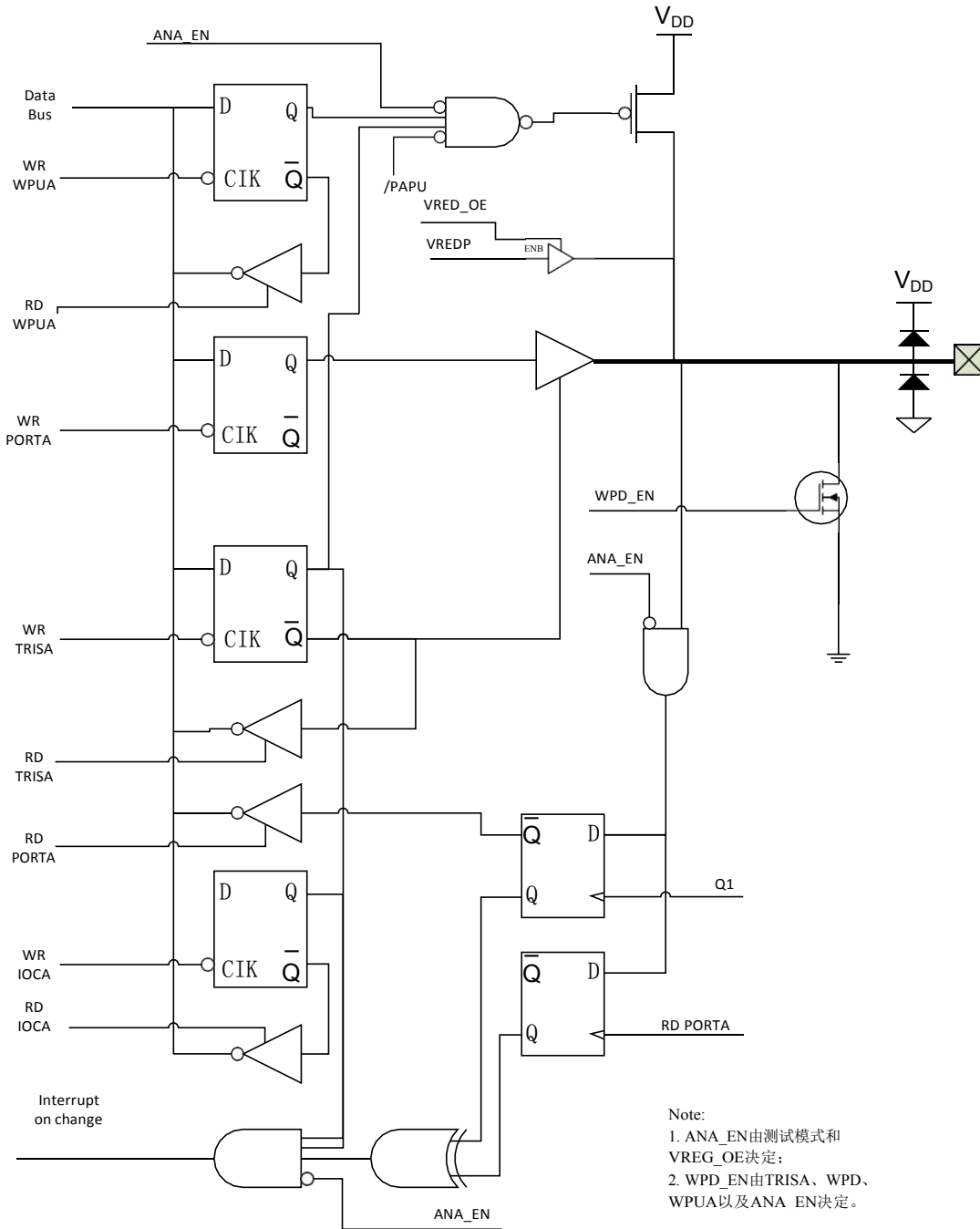


图 16.4 PA4 架构框图

### 16.3.1.5 PORTA[5]

图 16.5 描述了此端口的内部电路结构。PA[5]可以被配置为以下功能端口：

- GPIO
- 外部复位

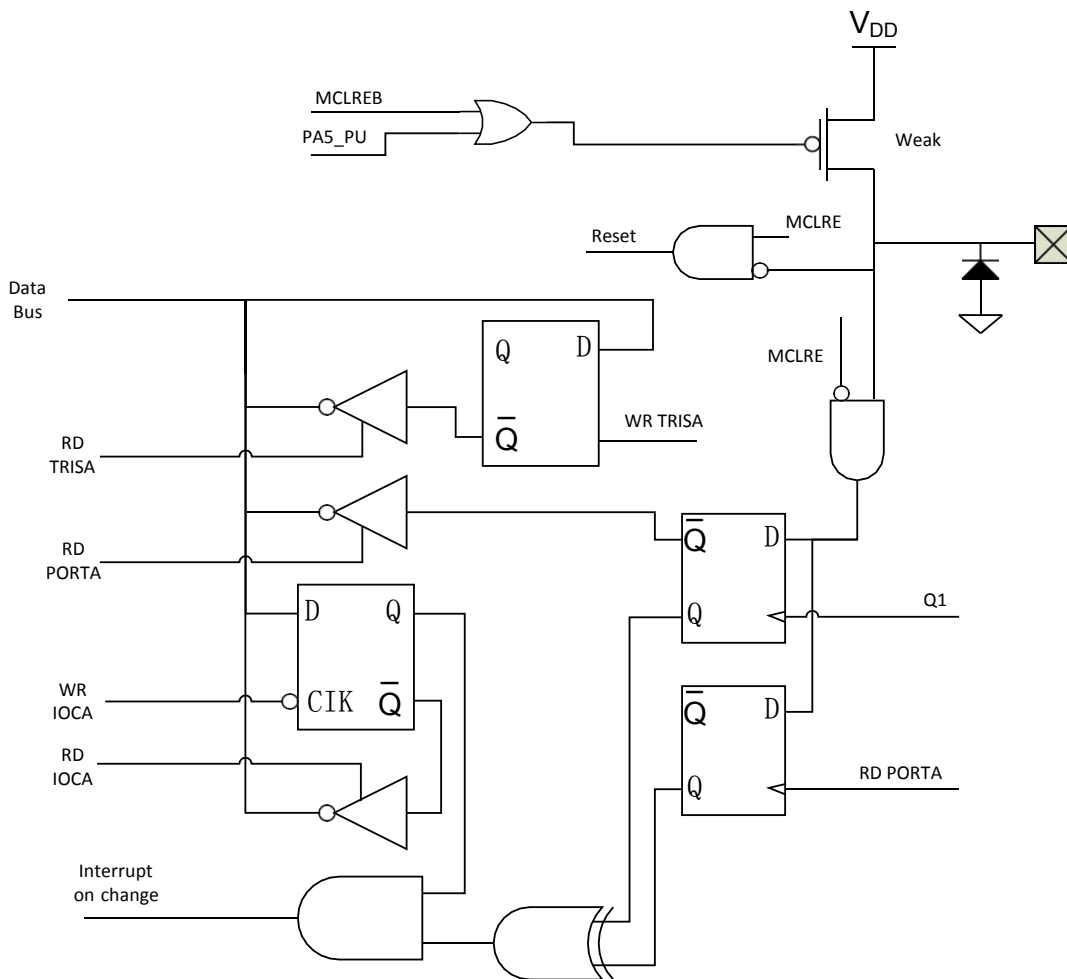


图 16.5 PA5 架构框图

### 16.3.1.6 PORTA[6]

图 16.6 描述了此端口内部电路结构。PA[6]可以配置为以下功能端口：

- GPIO
- 晶振、谐振器连接
- 时钟输出

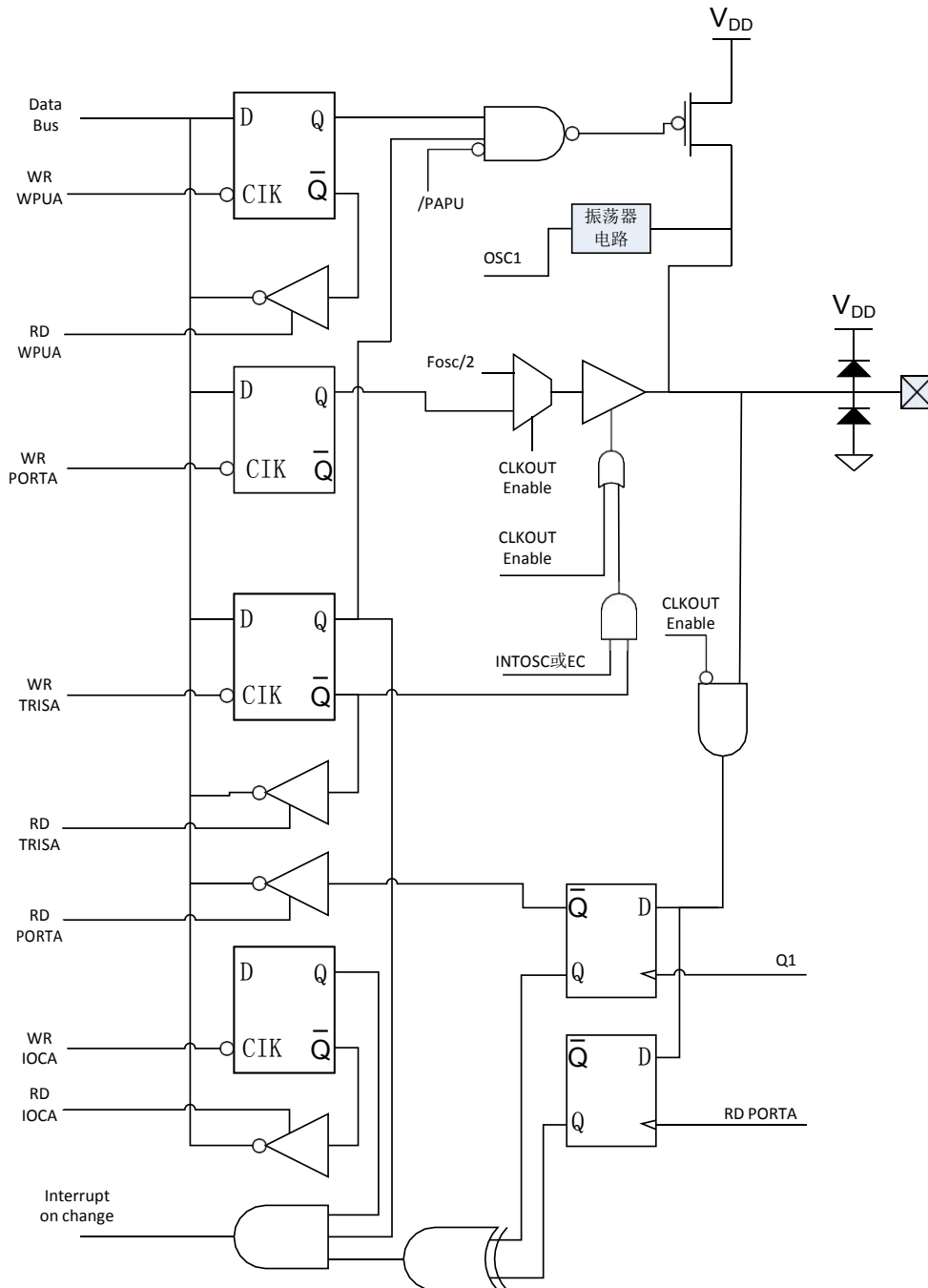


图 16.6 PA[6]架构框图

### 16.3.1.7 PORTA[7]

图 16.7 描述了此端口内部电路结构。PA[7]可以配置为以下功能端口：

- GPIO
- 晶振、谐振器连接
- 外部时钟输入
- TIMER1 外部时钟源输入

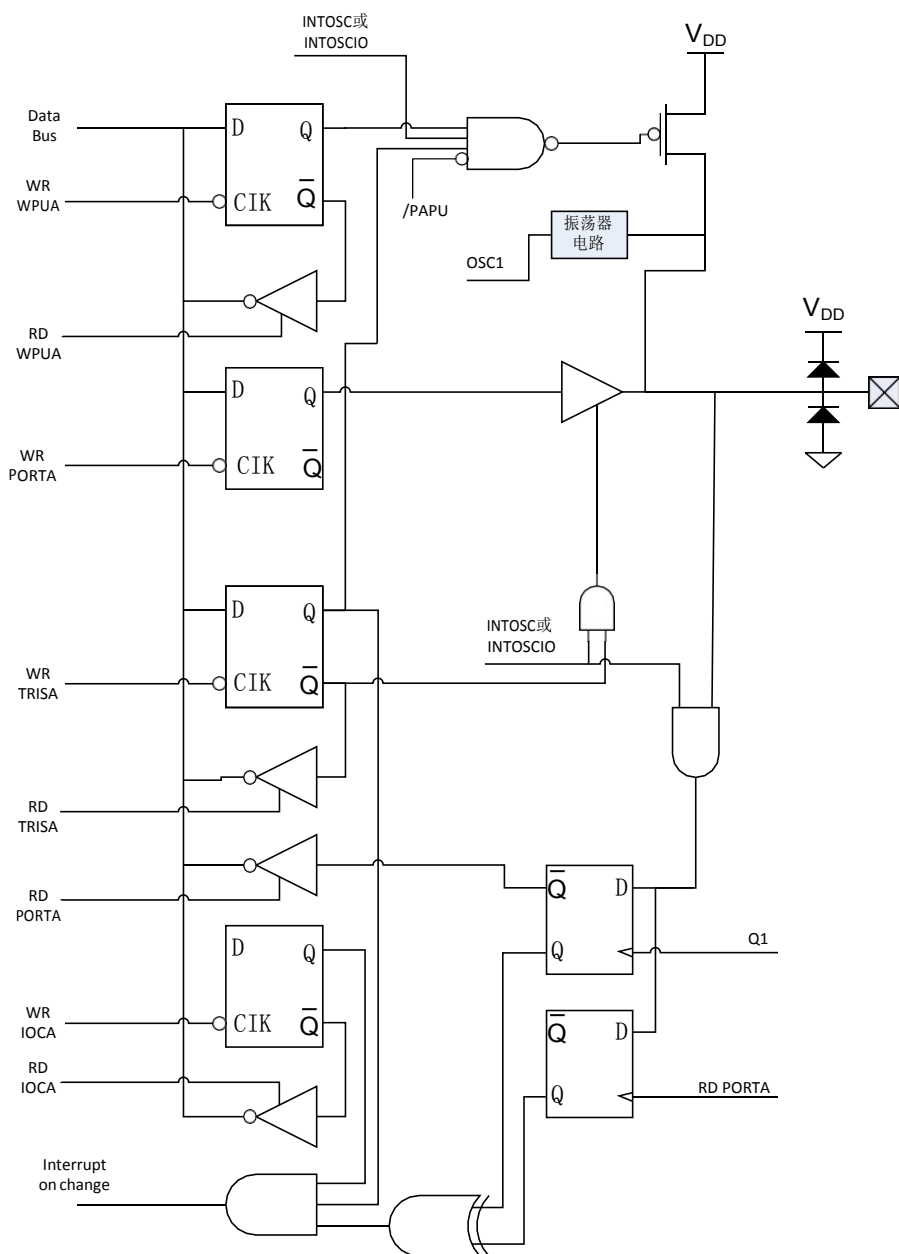


图 16.7 PA[7]架构框图

### 16.3.1.8 PORTC[0], [1], [2]

图 16.8 描述了此端口的内部电路结构。PORTC[0]/[1]/[2]可以被配置为以下功能端口：



- GPIO
- 外部中断输入 (PORTC1)
- PWM 输出
- 比较器输入 (PORTC0、PORTC1)
- ADC 输入通道

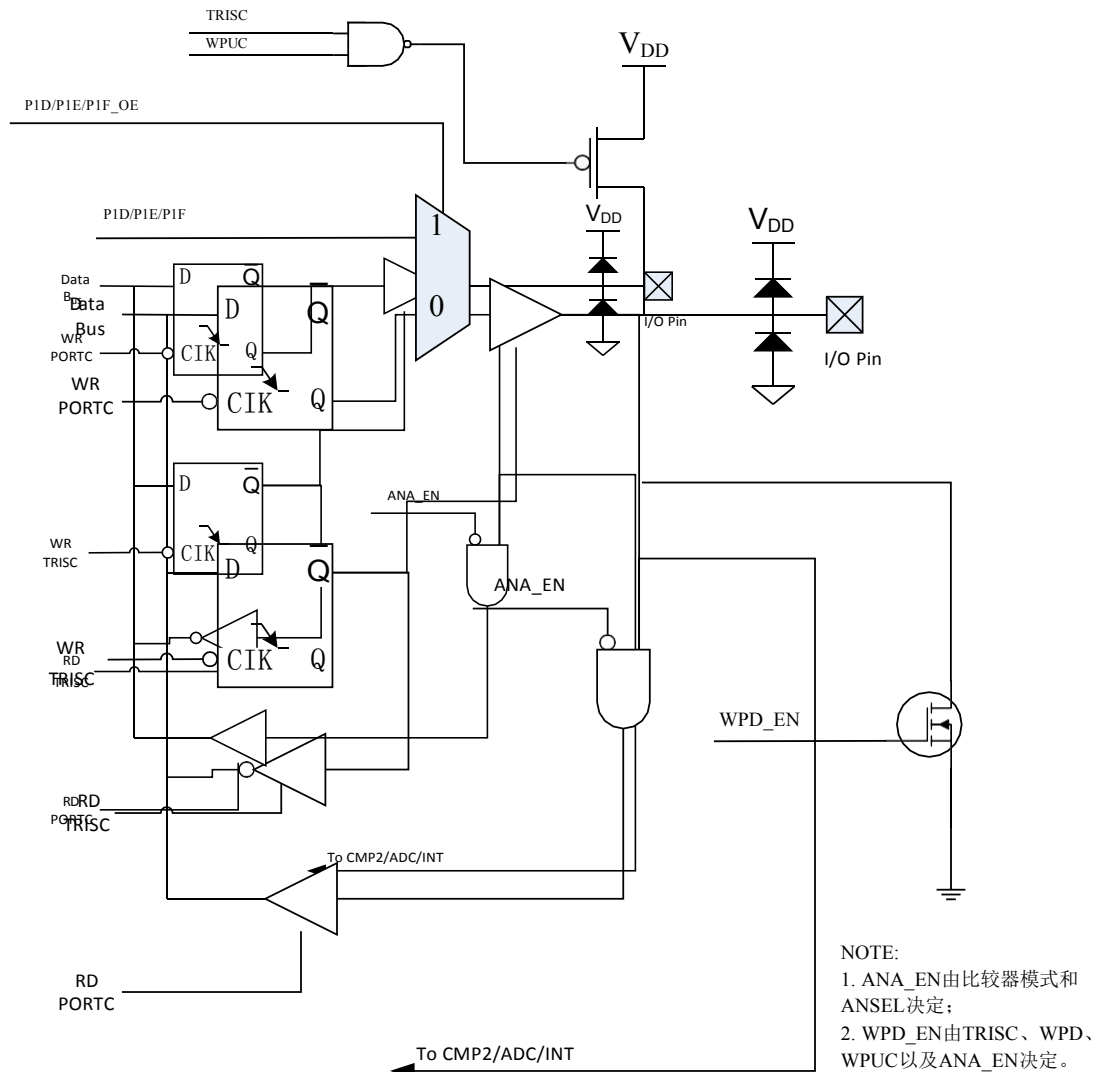


图 16.8 PC0, 1, 2 架构框图

### 16.3.1.9 PORTC[3]

图 16.9 描述了此端口的内部电路结构。PORTC[3]可以被配置为以下功能端口：

- GPIO
- 增强型 PWM 输出

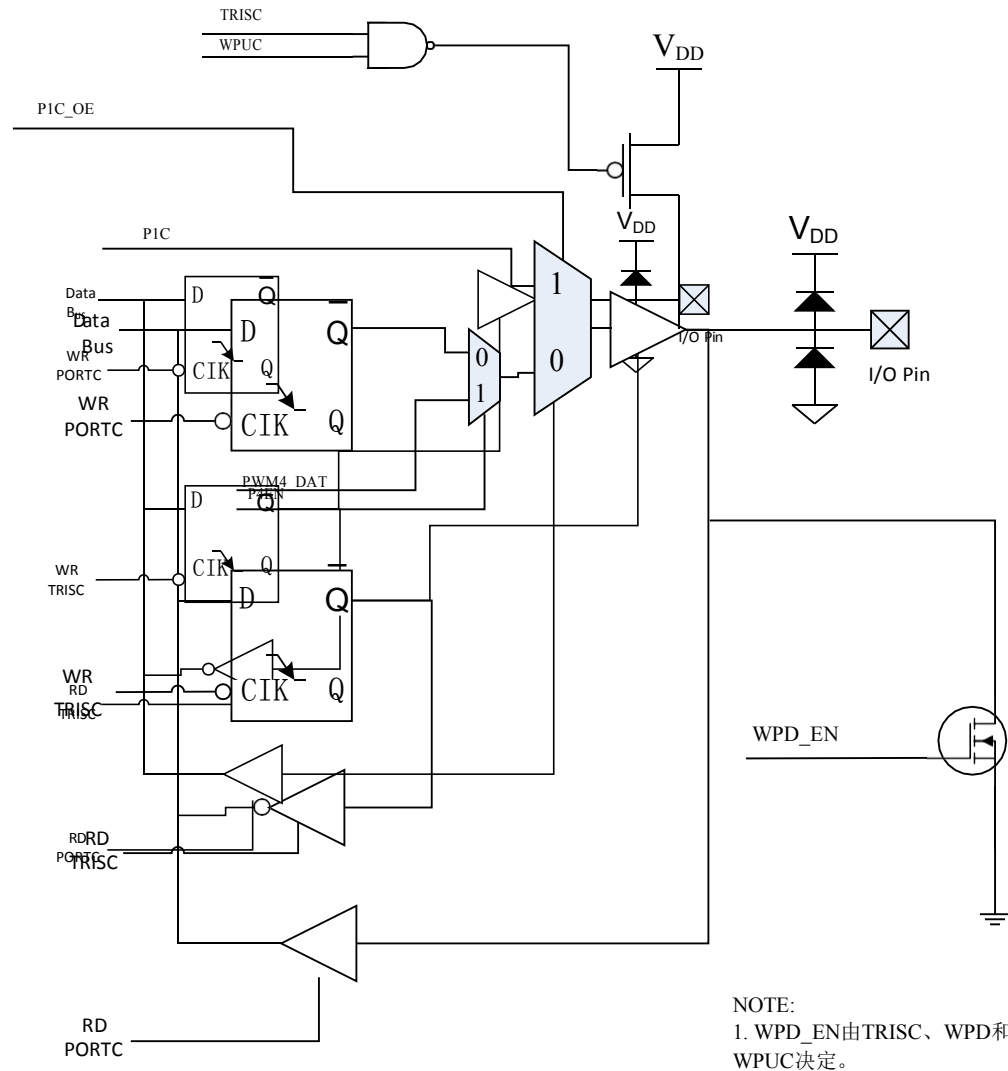


图 16.9 PC3 架构框图

### 16.3.1.10 PORTC[4]

图 16.10 描述了此端口的内部电路结构。PORTC[4]可以被配置为以下功能端口：

- GPIO
- 增强型 PWM 输出
- 比较器 2 输出

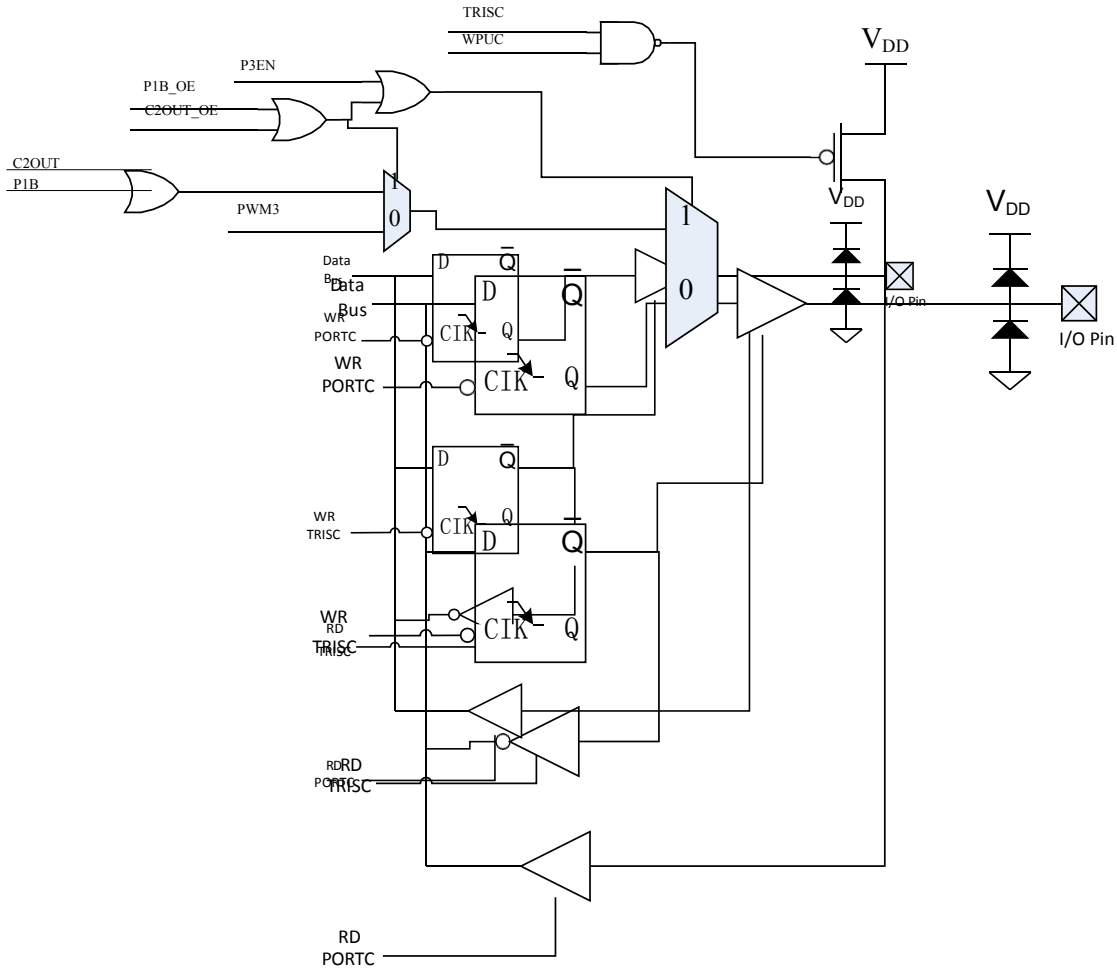


图 16.10 PC4 架构框图



### 16.3.1.11 PORTC[5]

图 16.11 描述了此端口的内部电路结构。PORTC[5]可以被配置为以下功能端口：

- GPIO
- ECCP 的捕捉输入
- 稳压器输出
- 增强型 PWM 输出

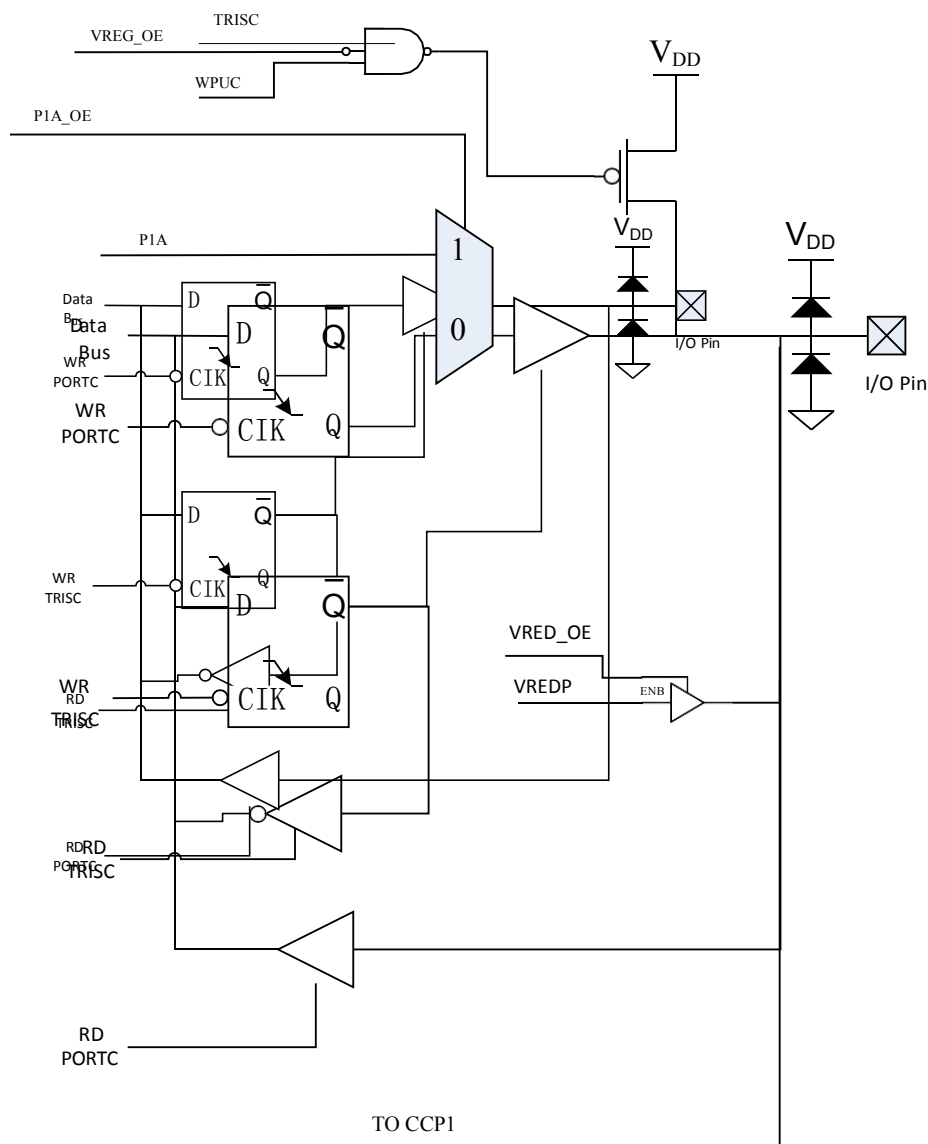


图 16.11 PC5 架构框图

## 16.4 与 GPIO 相关寄存器汇总

名称	地址	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	复位值
WPUA	0x95	PORTA 上拉控制位								1 1 1 1 1 1 1 1
TRISA	0x85	TRISA[7:0]，PORTA 方向控制								1 1 1 1 1 1 1 1
PORTA	0x05	PORTA[7:0]，PORTA 数据寄存器								x x x x x x x x
TRISC	0x87	-	-	TRISC[5:0]，PORTC 方向控制						- - 1 1 1 1 1 1
PORTC	0x07	-	-	PORTC[5:0]，PORTC 数据寄存器						- - x x x x x x
WPUC	0x88	-	-	PORTC 弱上拉控制位						- - 0 0 0 0 0 0
IOCA	0x96	IOCA[7:0]，端口中断允许位								0 0 0 0 0 0 0 0
ANSEL	0x91	ANSEL[7:0]，数模管脚选择								1 1 1 1 1 1 1 1
OPTION	0x81	/PAP U	INTED G	T0CS	T0SE	PSA	PS2	PS1	PS0	1 1 1 1 1 1 1 1
WPD	0x89	-	-	-	WPDA 4	WPD C1	WPD C2	WPD C3	-	- - - 0 0 0 0 -

### 16.4.1 PORTA 寄存器，地址 0x05

Bit	7	6	5	4	3	2	1	0
Name	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
Reset	x	x	x	x	x	x	x	x
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7	PA[7]	PORTA7 数据
6	PA[6]	PORTA6 数据
5	PA[5]	当 MCLRE 为 1 时，PA5 为外部复位管脚，否则是通用 IO PORTA5 数据
4	PA[4]	PORTA4 数据
3	PA[3]	PORTA3 数据
2	PA[2]	PORTA2 数据
1	PA[1]	PORTA1 数据
0	PA[0]	PORTA0 数据

### 16.4.2 PORTC 寄存器，地址 0x07

Bit	7	6	5	4	3	2	1	0
Name	-	-	PC5	PC4	PC3	PC2	PC1	PC0
Reset	-	-	x	x	x	x	x	x



Type	-	-	RW	RW	RW	RW	RW	RW
------	---	---	----	----	----	----	----	----

Bit	Name	Function
7:6	-	保留位
5:0	PORTC[7:0]	PORTC 数据寄存器

### 16.4.3 TRISA 寄存器 · 地址 0x85

Bit	7	6	5	4	3	2	1	0
Name	TRISA[7]	TRISA[6]	TRISA[5]	TRISA[4]	TRISA[3]	TRISA[2]	TRISA[1]	TRISA[0]
Reset	1	1	1	1	1	1	1	1
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:6	TRISA[7:6]	<b>TRISA &lt;7:6&gt;</b> : PA<7:6> Port I/O Tri-State Control bits 1 = Port output driver is disabled 0 = Port output driver is enabled
5	TRISA[5]	<b>TRISA &lt;5&gt;</b> : PA<5> Port I/O Tri-State Control bits 1 = Port output driver is disabled 0 = Port output driver is enabled
4:0	TRISA[4:0]	<b>TRISA &lt;4:0&gt;</b> : PA<4:0> Port I/O Tri-State Control bits 1 = Port output driver is disabled 0 = Port output driver is enabled

### 16.4.4 TRISC 寄存器 · 地址 0x87

Bit	7	6	5	4	3	2	1	0
Name	-	-	TRISC					
Reset	-	-	1	1	1	1	1	1
Type	-	-	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:6	-	保留位
5:0	TRISC[5:0]	PORTC I/O Tri-State Control bits 1 = Port output driver is disabled 0 = Port output driver is enabled



### 16.4.5 WPUA 寄存器，地址 0x95

Bit	7	6	5	4	3	2	1	0
Name	WPUA7	WPUA6	WPUA5	WPUA4	WPUA	WPUA2	WPUA1	WPUA0
Reset	1	1	1	1	1	1	1	1
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7,6,4,3,2,1,0	WPUAx	Port A 弱上拉使能 1 = 使能 PORT A 端口弱上拉 0 = 断开 PORT A 端口弱上拉
5	NA	MCLRE = 0 时，PORTA[5]的弱上拉功能 WPUA5 决定： WPUA5 = 1 时，使能 PA5 弱上拉 WPUA5 = 0 时，断开 PA5 弱上拉 而当 MCLRE = 1 时，PA5 弱上拉一直打开

### 16.4.6 WPUC 寄存器，地址 0x88

Bit	7	6	5	4	3	2	1	0
Name	-	-	WPUC[5:0]					
Reset	-	-	0	0	0	0	0	0
Type	R-0	R-0	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:6	-	保留位，读 0
5:0	WPUC[5:0]	Port C 弱上拉使能 1 = 使能 PORT C 端口弱上拉 0 = 断开 PORT C 端口弱上拉

### 16.4.7 WPD 寄存器，地址 0x89

Bit	7	6	5	4	3	2	1	0
Name	-	-	-	WPDA4	WPD1	WPDC2	WPDC3	-
Reset	-	-	-	0	0	0	0	-
Type	R-0	R-0	R-0	RW	RW	RW	RW	RW

Bit	Name	Function
7:5, 0	-	保留位，读 0
4	WPDA4	PORTA[4]弱下拉使能



		1 = 使能 PORTA[4]端口弱下拉 0 = 断开 PORTA[4]端口弱下拉
3	WPDC[1]	PORTC[1]弱下拉使能 1 = 使能 PORTC[1]端口弱下拉 0 = 断开 PORTC[1]端口弱下拉
2	WPDC[2]	PORTC[2]弱下拉使能 1 = 使能 PORTC[2]端口弱下拉 0 = 断开 PORTC[2]端口弱下拉
1	WPDC[3]	PORTC[3]弱下拉使能 1 = 使能 PORTC[3]端口弱下拉 0 = 断开 PORTC[3]端口弱下拉

## 17 PWMx 和 TIMERx

除了 ECCP 模块所提供的 1 路增强型 PWM 之外，片内还集成了 3 路相互独立的 12bit PWM，它们都具备以下特性：

- BUZZER 或 PWM 输出可选
- 递增计数器
- PWM/BUZZER 输出极性可选
- 8 种 PWM 分辨率
- 4 种时钟源可选
- 1~256 分频可选
- 7 位预分频
- 溢出中断或者匹配中断
- 时钟调制输出

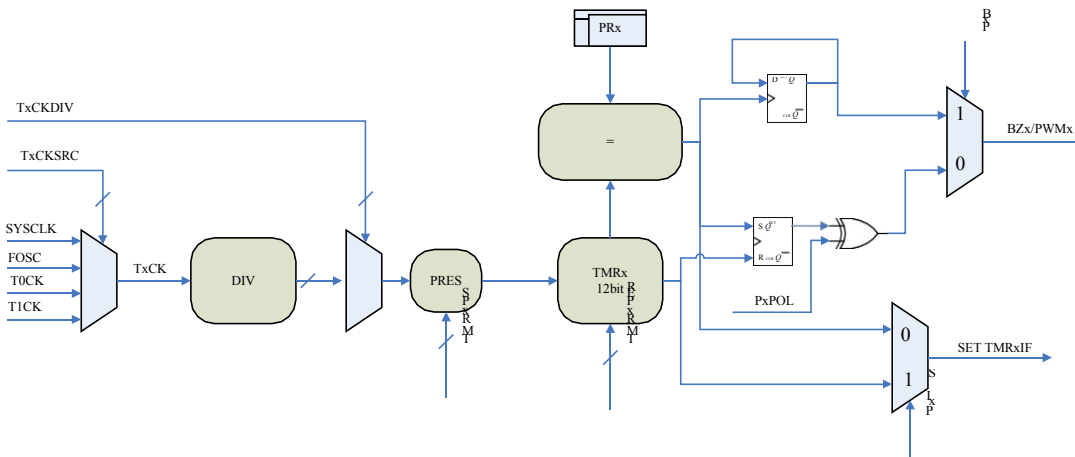


图 17.1 PWM3/4/5 原理框图

### 17.1 TIMERx 工作原理

当 PxEN (PWMxCR1.7) 为 0 时，工作在定时器模式。

TIMER3/4/5 都是 12 位的递增计数器，可以通过寄存器 TMRxH:TMRxL (x 可以是 3/4/5) 对计数值进行访问，软件对 TMRxH:TMRxL 的写操作将直接更新计数值。

如果选择内部时钟作为计数源，它们表现为定时器，如果选择外部时钟作为计数源，则表现为计数器。

#### 17.1.1 时钟源选择

PWMxCR0 的 PxCKSRC[2:0]用于选择时钟源。配置值和时钟源的关系如下表：

PxCKSRC[2:0]值	PxCK 时钟
000	系统时钟 SYSCLK/(TxCKDIV+1)
001	内部 RC 快时钟/(TxCKDIV+1)



010	T0CK/(TxCKDIV+1)
011	T1CK/(TxCKDIV+1)
100	内部 RC 快时钟/(TxCKDIV+1) 同时 PWM <sub>x</sub> 输出低电平
101	内部 RC 快时钟/(TxCKDIV+1) 同时 PWM <sub>x</sub> 输出高电平
110	内部 RC 快时钟/(TxCKDIV+1) 同时 PWM <sub>x</sub> 根据高脉冲调制 PxCK
111	内部 RC 快时钟/(TxCKDIV+1) 同时 PWM <sub>x</sub> 根据低脉冲调制 PxCK

注意：

1. 当 PxCKSRC[2:0]=001 或者 PxCKSRC[2]=1，即选择为内部 RC 快时钟时，不管系统时钟选择慢时钟或者外部时钟、晶体时钟，内部 HFINTOSC 都会打开，除非进入睡眠模式。

### 17.1.2 TIMERx 时钟分频

PWMx/TIMERx 可以对时钟源进 1~256 分频，其除数由 TxCKDIV 指定。

注意：当发生对 TxCKDIV 寄存器的写操作时，除数分频器将自动清 0。

### 17.1.3 TIMERx 中断

当 TIMERx 计数值 TMRxH:TMRxL 递增到由 PxPER 指定的最大计数值后，再过一个计数时钟便返回 0x000，此时 TIMERx 发生溢出，相关中断标志位被置 1。如果以下条件满足，则 CPU 进入中断处理程序：

- PWMxCR1 寄存器的 TMRxIE 为 1
- INTCON 寄存器的 PEIE 为 1
- INTCON 寄存器的 GIE 为 1

注意：

1. 应该在中断服务退出前把标志位 TMRxIF 清 0，以免循环进入中断。

### 17.1.4 TIMERx 预分频

TIMERx 有 8 种预分频选项，分别可以对 PxCK 进行 1、2、4、8、16、32、64、128 分频。预分频计数器不能直接读写，但当发生对 TMRxH 或者 TMRxL 的写操作时，预分频计数器会自动清 0。

### 17.1.5 TIMERx 周期

TIMERx 最大位数为 12bit，通过对 PxPER[2:0]的配置可以选择不同的溢出周期。

注意：

1. 在 BUZZER 模式下，当计数值和 PR 寄存器相等时，再来一个计数时钟 TIMERx 便会自动清 0；
2. 在 TIMERx 已经开启的情况下改写 PxPER 的值可能会把 TMRxIF 置 1，所以建议先配置好 PxPER 后才开启 TMRxON。



## 17.1.6 TIMERx 在休眠模式下工作

如果被配置使用外部时钟 T0CK/T1CK (TxCKSRC=010 或者 011)，则在睡眠模式下 TIMERx 可以作为一个定时唤醒源。软件上需要做以下配置：

- PxEN=0，工作在 TIMER 模式
- TxCKSRC=010 或者 011
- PWMxCR1 的 TMRxON 为 1
- PWMxCR1 寄存器的 TMRxIE 为 1
- INTCON 寄存器的 PEIE 为 1

当 TIMERx 定时溢出时，CPU 将被唤醒。如果此时 GIE=1，则执行 SLEEP 的下一条指令后进入中断处理，否则是顺序执行。

## 17.1.7 读写 TMRxH/L 寄存器

由图 17.1 可知，TIMERx 运行时钟相对于 2T 或者 4T 的 CPU 时钟来说是异步时钟，所以在 TIMERx 处于运行状态下软件读取一个 12bit 的计数值时会产生一些问题：有可能在读取了一个低 8bit 之后，TIMERx 产生了溢出导致计数值复位，即再读高 4bit 时已经被清 0。

而对于写操作，建议先把 TIMERx 停止 (TMRxON=0)，再把目标值写入 TMRxH/TMRxL。

## 17.2 BUZZER 工作模式

当 PxEN=1 且 PxBZR=1 时，TIMERx 就处于蜂鸣器 (BUZZER) 模式。

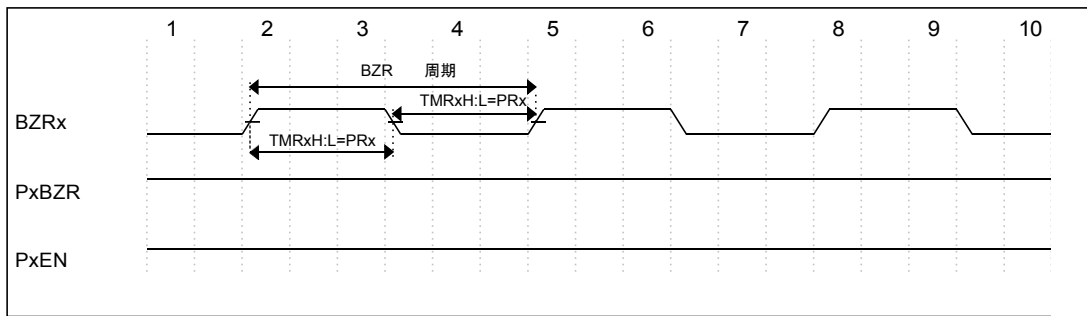


图 17.2 50 % 占空比的 BUZZER 方波

### 17.2.1 BUZZER 周期

$$T_{BUZ}=2 * 2^{TMRxPS} * PRx * T_{PxCK}$$

注意：

1. 工作在 BUZZER 模式下时，TIMERx 自动工作在 12bit 模式，而不管 PxPER 为何值；
2. 当 TMRxH:L 等于 PRx 时，TMRxH:L 自动清 0；
3. 处于 BUZZER 模式下，如果 PRx=0x000，则 BZRx 管脚固定输出 0，12bit 的 TIMERx 溢出时 TMRxIF 才会



置 1。

## 17.2.2 BUZZER 在休眠模式下工作

同 TIMER 模式一样，通过适当的配置，BUZZER 也可以工作在睡眠模式：

- PxEN=1，工作在 PWM/BUZZER 模式
- PxBZR=1，选择 BUZZER 功能
- TxCKSRC=010 或者 011
- PWMxCR1 的 TMRxON 为 1
- PWMxCR1 寄存器的 TMRxIE 为 1
- INTCON 寄存器的 PEIE 为 1

## 17.3 PWM 工作模式

当 PxEN=1 且 PxBZR=0 时，TIMERx 工作于 PWM 模式。

下图为 PWM 输出的一个示例，它由周期和一段输出保持为高（占空比）的时间组成。

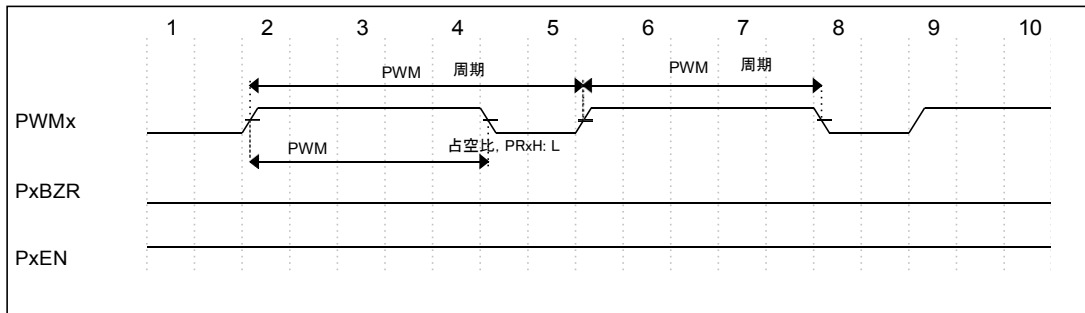


图17.3 PWM 工作模式，正向输出

PWM3/4/5 的时基和占空比相互独立，能最大限度满足各种应用的需求。

注意：

如果想同时使用 ECCP 的 PWM ( P1A 管脚输出 PWM 调制 ) 和 PWM3/4/5，则寄存器 CCP1CON 的 P1M<1:0> 要配置为 00，即单输出模式，P1A 调制，P1B/P1C/P1D 为端口配置。换句话说，ECCP 优先级比 PWM3/PWM4/PWM5 要高。

### 17.3.1 PWM 周期

PWMx 周期由 TIMERx 的预分频比设置寄存器 TMRxPS，PxPER 决定。公式如下：

$$\text{PWM 周期} = 2^{\text{TMRxPS}} * 2^{\text{Nbit}} * T_{\text{PCK}}$$

式中，Nbit 为 PxPER 设定的定时器位数。

### 17.3.2 PWM 占空比

PWM 的占空比由寄存器 PRxH:L 决定。PRxH:L 一共 12bit，软件需要分开两次写。由于没有映像寄存器，软件对



PRxH:L 的写是立刻生效，即 PRxH:L 值的改变对当前 PWM 占空比有影响，应用必须考虑这一点。

下式是计算占空比（脉冲宽度）的公式：

$$T_{pwm} = 2^{TMRxPS} * (PRx) * T_{PCK}$$

**注意：**

由于 PRx 没有对应的映像寄存器，所以 PRx 的写操作将直接影响当前占空比，建议在没启动 TIMERx 之前把目标值写入 PRx。

### 17.3.3 PWM 工作设置

建议按以下步骤设置 PWM3/4/5 工作：

1. 将相关的 TRIS 位置 1 禁止 PWMx 引脚的输出驱动器；
2. 装载 PWMxCR0 寄存器以设置 PWMx 周期，中断产生方式以及选择时钟源；
3. 用适当的值装载 PWMxCR1 寄存器配置为 PWM 模式，合适的预分频比；
4. 装载 PRx 寄存器设置 PWM 占空比；
5. 配置并启动 TIMERx：
  - 将 PWMxCR1 寄存器的 TMRXIF 中断标志位清零
  - 将 PWMxCR1 寄存器的 TMRxON 位置 1 启动 TIMERx
6. 重新开始一个 PWM 周期后，使能 PWM 输出：
  - 等待 TIMERx 溢出（PWMxCR1 寄存器的 TMRxIF 位置 1）
  - 将相关的 TRIS 位清零使能 PWMx 引脚的输出驱动器

**注意：**

- 1 如果 PWM 时钟设置为系统时钟（TxCKSRC=000），则系统时钟频率的任何改变将导致 PWM 频率的改变。

### 17.3.4 PWM 工作在休眠模式

同 TIMER 模式一样，通过适当的配置，PWM 也可以工作在睡眠模式：

- PxEN=1，工作在 PWM/BUZZER 模式
- PxBZR=0，选择 PWM 功能
- TxCKSRC=010 或者 011
- PWMxCR1 的 TMRxON 为 1
- PWMxCR1 寄存器的 TMRxIE 为 1
- INTCON 寄存器的 PEIE 为 1

## 17.4 PWMx 相关寄存器

### 17.4.1 TMR3L 寄存器，地址 0x10C

Bit	7	6	5	4	3	2	1	0
Name	TMR3L							



Reset	x	x	x	x	x	x	x	x
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:0	TMR3L[7:0]	TIMER3 计数结果低 8 位

### 17.4.2 TMR3H 寄存器，地址 0x10D

Bit	7	6	5	4	3	2	1	0
Name	TMR3H[3:0]				PR3H[3:0]			
Reset	x	x	x	x	1	1	1	1
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:4	TMR3H[7:0]	TIMER3 计数结果高 4 位
3:0	PR3H[3:0]	PWM3 匹配寄存器 PR3 高 4 位

### 17.4.3 PR3L 寄存器，地址 0x10E

Bit	7	6	5	4	3	2	1	0
Name	PR3L							
Reset	1	1	1	1	1	1	1	1
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:0	PR3L[7:0]	PWM3 匹配寄存器 PR3 低 8 位

### 17.4.4 PWM3CR0 寄存器，地址 0x10F

Bit	7	6	5	4	3	2	1	0
Name	P3INTS	P3PER[2:0]			P3CKSRC[2:0]			P3BZR
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7	P3INTS	PWM3 中断选择位 1 = TMR3 计数值和 PR3 匹配时产生中断 0 = TMR3 计数值溢出时产生中断
6:4	P3PER[2:0]	PWM3 周期选择位 P3PER[2:0]值   PWM3 周期位数



		000	4 位
		001	5 位
		010	6 位
		011	8 位
		100	9 位
		101	10 位
		110	11 位
		111	12 位
3:1	P3CKSRC[2:0]	TIMER3/PWM3 时钟选择位	
		P3CKSRC[2:0]值	P3CK 时钟
		000	系 统 时 钟 SYSCLK/(T3CKDIV+1)
		001	内部 RC 快时钟/(T3CKDIV+1)
		010	T0CK/(T3CKDIV+1)
		011	T1CK/(T3CKDIV+1)
		100	内部 RC 快时钟/(T3CKDIV+1) 同时 PWM3 输出低电平
		101	内部 RC 快时钟/(T3CKDIV+1) 同时 PWM3 输出高电平
		110	内部 RC 快时钟/(T3CKDIV+1) 同时 PWM3 根据高脉冲调制 P3CK
		111	内部 RC 快时钟/(T3CKDIV+1) 同时 PWM3 根据低脉冲调制 P3CK
0	P3BZR	PWM/BUZZER 选择 1 = 选择 BUZZER 输出 0 = 选择 PWM 输出	

### 17.4.5 PWM3CR1 寄存器 · 地址 0x110

Bit	7	6	5	4	3	2	1	0
Name	P3EN	P3POL	TMR3PS[2:0]			TMR3ON	TMR3IE	TMR3IF
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7	P3EN	TMR3/PWM3 工作模式 1 = TMR3 为 PWM/BUZZER 模式 0 = TMR3 为定时器模式
6	P3POL	PWM3 输出极性选择 1 = PWM3 为低电平有效 0 = PWM3 为高电平有效



5:3	TMR3PS[2:0]	PWM3 预分频比设置位	
		TMR3PS[2:0]值	PWM3 分频比
		000	1:1
		001	1:2
		010	1:4
		011	1:8
		100	1:16
		101	1:32
		110	1:64
		111	1:128
2	TMR3ON	定时器 3 使能位 1 = 打开定时器 3 0 = 关闭定时器 3	
1	TMR3IE	定时器 3 中断使能位 1 = 允许 TMR3 产生中断 0 = 禁止 TMR3 产生中断	
0	TMR3IF	定时器 3 中断标志位 1: P3INTS =1 时，TMR3 发生了匹配 P3INTS =0 时，TMR3 发生了溢出 0: 没有发生溢出或匹配	

### 17.4.6 T3CKDIV 寄存器，地址 0x111

Bit	7	6	5	4	3	2	1	0
Name	TMR3 时钟分频寄存器							
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:0	T3CKDIV[7:0]	TMR3 时钟频率为 $F_{T3CK}/(T3CKDIV+1)$



### 17.4.7 TMR4L 寄存器，地址 0x112

Bit	7	6	5	4	3	2	1	0
Name	TMR4L							
Reset	x	x	x	x	x	x	x	x
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:0	TMR4L[7:0]	TIMER4 计数结果低 8 位

### 17.4.8 TMR4H 寄存器，地址 0x113

Bit	7	6	5	4	3	2	1	0
Name	TMR4H[3:0]				PR4H[3:0]			
Reset	x	x	x	x	1	1	1	1
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:4	TMR4H[7:0]	TIMER4 计数结果高 4 位
3:0	PR4H[3:0]	PWM4 匹配寄存器 PR4 高 4 位

### 17.4.9 PR4L 寄存器，地址 0x114

Bit	7	6	5	4	3	2	1	0
Name	PR4L							
Reset	1	1	1	1	1	1	1	1
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:0	PR4L[7:0]	PWM4 匹配寄存器 PR4 低 8 位

### 17.4.10 PWM4CR0 寄存器，地址 0x115

Bit	7	6	5	4	3	2	1	0
Name	P4INTS	P4PER[2:0]			P4CKSRC[2:0]			P4BZR
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW



Bit	Name	Function	
7	P4INTS	PWM4 中断选择位 1 = TMR4 计数值和 PR4 匹配时产生中断 0 = TMR4 计数值溢出时产生中断	
6:4	P4PER[2:0]	PWM4 周期选择位	
		P4PER[2:0]值	PWM4 周期位数
		000	4 位
		001	5 位
		010	6 位
		011	8 位
		100	9 位
		101	10 位
		110	11 位
111	12 位		
3:1	P4CKSRC[2:0]	TIMER4/PWM4 时钟选择位	
		P4CKSRC[2:0]值	P4CK 时钟
		000	系 统 时 钟 SYSCLK/(T4CKDIV+1)
		001	内部 RC 快时钟/(T4CKDIV+1)
		010	T0CK/(T4CKDIV+1)
		011	T1CK/(T4CKDIV+1)
		100	内部 RC 快时钟/(T4CKDIV+1) 同时 PWM4 输出低电平
		101	内部 RC 快时钟/(T4CKDIV+1) 同时 PWM4 输出高电平
		110	内部 RC 快时钟/(T4CKDIV+1) 同时 PWM4 根据高脉冲调制 P4CK
111	内部 RC 快时钟/(T4CKDIV+1) 同时 PWM4 根据低脉冲调制 P4CK		
0	P4BZR	PWM/BUZZER 选择 1 = 选择 BUZZER 输出 0 = 选择 PWM 输出	

### 17.4.11 PWM4CR1 寄存器 · 地址 0x116

Bit	7	6	5	4	3	2	1	0
Name	P4EN	P4POL	TMR4PS[2:0]			TMR4ON	TMR4IE	TMR4IF
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW



Bit	Name	Function	
7	P4EN	TMR4/PWM4 工作模式 1 = TMR4 为 PWM/BUZZER 模式 0 = TMR4 为定时器模式	
6	P4POL	PWM4 输出极性选择 1 = PWM4 为低电平有效 0 = PWM4 为高电平有效	
5:3	TMR4PS[2:0]	PWM4 预分频比设置位	
		TMR4PS[2:0]值	PWM4 分频比
		000	1:1
		001	1:2
		010	1:4
		011	1:8
		100	1:16
		101	1:32
		110	1:64
	111	1:128	
2	TMR4ON	定时器 3 使能位 1 = 打开定时器 3 0 = 关闭定时器 3	
1	TMR4IE	定时器 3 中断使能位 1 = 允许 TMR4 产生中断 0 = 禁止 TMR4 产生中断	
0	TMR4IF	定时器 3 中断标志位 1: P4INTS =1 时, TMR4 发生了匹配 P4INTS =0 时, TMR4 发生了溢出 0: 没有发生溢出或匹配	

### 17.4.12 T4CKDIV 寄存器，地址 0x117

Bit	7	6	5	4	3	2	1	0
Name	TMR4 时钟分频寄存器							
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:0	T4CKDIV[7:0]	TMR4 时钟频率为 $F_{T4CK}/(T4CKDIV+1)$





### 17.4.13 TMR5L 寄存器 · 地址 0x118

Bit	7	6	5	4	3	2	1	0
Name	TMR5L							
Reset	x	x	x	x	x	x	x	x
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:0	TMR5L[7:0]	TIMER5 计数结果低 8 位

### 17.4.14 TMR5H 寄存器 · 地址 0x119

Bit	7	6	5	4	3	2	1	0
Name	TMR5H[3:0]				PR5H[3:0]			
Reset	x	x	x	x	1	1	1	1
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:4	TMR5H[7:0]	TIMER5 计数结果高 4 位
3:0	PR5H[3:0]	PWM5 匹配寄存器 PR5 高 4 位

### 17.4.15 PR5L 寄存器 · 地址 0x11A

Bit	7	6	5	4	3	2	1	0
Name	PR5L							
Reset	1	1	1	1	1	1	1	1
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:0	PR5L[7:0]	PWM5 匹配寄存器 PR5 低 8 位

### 17.4.16 PWM5CR0 寄存器 · 地址 0x11B

Bit	7	6	5	4	3	2	1	0
Name	P5INTS	P5PER[2:0]			P5CKSRC[2:0]			P5BZR
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW



Bit	Name	Function	
7	P5INTS	PWM5 中断选择位 1 = TMR5 计数值和 PR5 匹配时产生中断 0 = TMR5 计数值溢出时产生中断	
6:4	P5PER[2:0]	PWM5 周期选择位	
		P5PER[2:0]值	PWM5 周期位数
		000	4 位
		001	5 位
		010	6 位
		011	8 位
		100	9 位
		101	10 位
		110	11 位
111	12 位		
3:1	P5CKSRC[2:0]	TIMER5/PWM5 时钟选择位	
		P5CKSRC[2:0]值	P5CK 时钟
		000	系 统 时 钟 SYSCLK/(T5CKDIV+1)
		001	内部 RC 快时钟/(T5CKDIV+1)
		010	T0CK/(T5CKDIV+1)
		011	T1CK/(T5CKDIV+1)
		100	内部 RC 快时钟/(T5CKDIV+1) 同时 PWM5 输出低电平
		101	内部 RC 快时钟/(T5CKDIV+1) 同时 PWM5 输出高电平
		110	内部 RC 快时钟/(T5CKDIV+1) 同时 PWM5 根据高脉冲调制 P5CK
111	内部 RC 快时钟/(T5CKDIV+1) 同时 PWM5 根据低脉冲调制 P5CK		
0	P5BZR	PWM/BUZZER 选择 1 = 选择 BUZZER 输出 0 = 选择 PWM 输出	

### 17.4.17 PWM5CR1 寄存器 · 地址 0x11C

Bit	7	6	5	4	3	2	1	0
Name	P5EN	P5POL	TMR5PS[2:0]			TMR5ON	TMR5IE	TMR5IF
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW



Bit	Name	Function	
7	P5EN	TMR5/PWM5 工作模式 1 = TMR5 为 PWM/BUZZER 模式 0 = TMR5 为定时器模式	
6	P5POL	PWM5 输出极性选择 1 = PWM5 为低电平有效 0 = PWM5 为高电平有效	
5:3	TMR5PS[2:0]	PWM5 预分频比设置位	
		TMR5PS[2:0]值	PWM5 分频比
		000	1:1
		001	1:2
		010	1:4
		011	1:8
		100	1:16
		101	1:32
		110	1:64
111	1:128		
2	TMR5ON	定时器 3 使能位 1 = 打开定时器 3 0 = 关闭定时器 3	
1	TMR5IE	定时器 3 中断使能位 1 = 允许 TMR5 产生中断 0 = 禁止 TMR5 产生中断	
0	TMR5IF	定时器 3 中断标志位 1: P5INTS =1 时, TMR5 发生了匹配 P5INTS =0 时, TMR5 发生了溢出 0: 没有发生溢出或匹配	

### 17.4.18 T5CKDIV 寄存器，地址 0x11D

Bit	7	6	5	4	3	2	1	0
Name	TMR5 时钟分频寄存器							
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:0	T5CKDIV[7:0]	TMR5 时钟频率为 $F_{T5CK}/(T5CKDIV+1)$

## 18 稳压器输出

WF8M165 内置两路稳压器输出，每路可分别输出多达 32 档电压。当 VREG\_OE 为 1 时（寄存器位 MSCKCON.6），PA4 和 PC5 便自动禁止数字输出功能，变为模拟电压输出。

输出电压由以下公式决定：

$$V_{OUT} = V_{REG} * (VREGHB + 1) / 32 \quad (\text{公式 18.1})$$

$$V_{OUT} = V_{REG} * (VREGLB + 1) / 32 \quad (\text{公式 18.2})$$

其中公式 18.1 为管脚 V<sub>RECP</sub>（PA4）的电压输出，公式 18.2 为管脚 V<sub>RECN</sub>（PC5）的电压输出，V<sub>REG</sub> 电压可以是 2.4V、3.6V、4.8V、5.3V 中的一种，由寄存器位 VREGM[1:0] 选择。

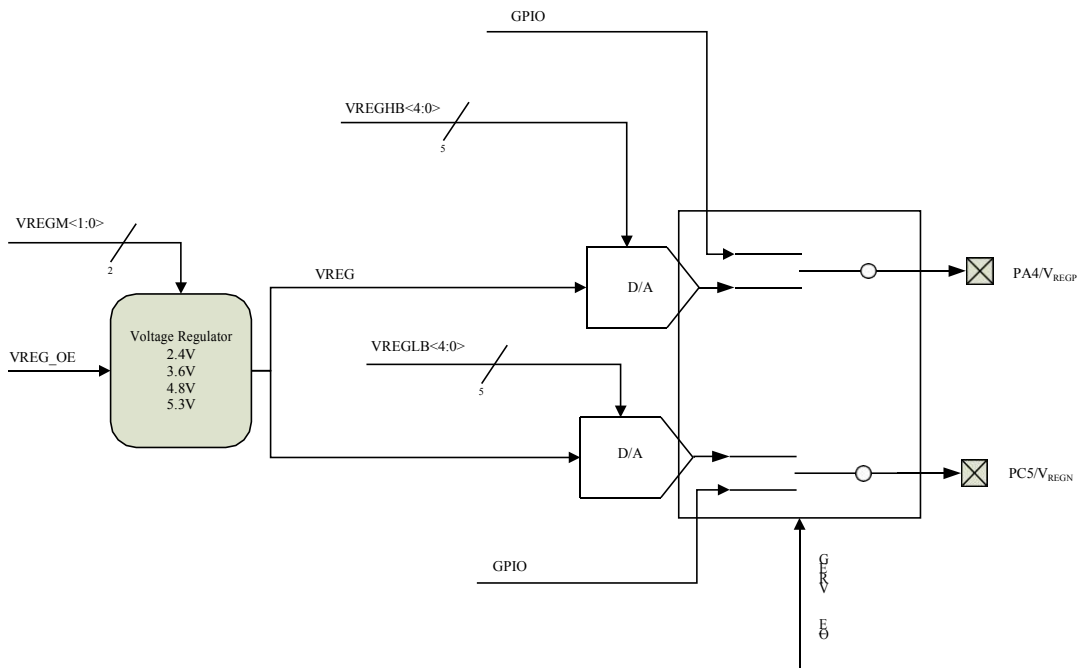


图 18.1 稳压输出原理框图

### 18.1 与稳压器输出相关寄存器汇总

名称	地址	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	复位值
MSCKCON	0x01B	-	VREG_OE	T2CKSRC	SLVREN	-	CKMAVG	CKCNTI	-	0000 0000
VCON1	0x108	-	VREGM[1:0]		VREGHB[4:0]					-000 0000
VCON2	0x109	-	-	-	VREGLB[4:0]					---0 0000

#### 18.1.1 VCON1 寄存器 · 地址 0x108

Bit	7	6	5	4	3	2	1	0
Name	-	VREGM[1:0]		VREGHB[4:0]				



Reset	-	2' b0	5' b0
Type	-	RW	RW

Bit	Name	Function	
7	-	保留位，不要写 1 读 0	
6:5	VREGM[1:0]	稳压器电压选择位	
		值	稳压器电压 $V_{REG}$
		00	2.4V
		01	3.6V
		10	4.8V
		11	5.3V
4:0	VREGHB[4:0]	当 MSCKCON 的 VREG_OE 位为 1 时，管脚 PA4 输出电压为 $V_{OUT} = V_{REG} * (VREGHB + 1) / 32$	

### 18.1.2 VCON2 寄存器，地址 0x109

Bit	7	6	5	4	3	2	1	0
Name	-			VREGLB[4:0]				
Reset	-			5' b0				
Type	-			RW				

Bit	Name	Function
7:5	-	保留位，不要写 1 读 0
4:0	VREGLB[4:0]	当 MSCKCON 的 VREG_OE 位为 1 时，管脚 PC5 输出电压为 $V_{OUT} = V_{REG} * (VREGLB + 1) / 32$



## 19 芯片的电气特性

### 19.1 极限参数

工作温度..... 0 ~ +85 °C  
 存储温度..... -40 ~ +125 °C  
 电源电压.....  $V_{SS} - 0.3V \sim V_{SS} + 6.0V$   
 端口输入电压.....  $V_{SS} - 0.3V \sim V_{DD} + 0.3V$

### 19.2 内置高频振荡器(HFINTOSC)

电气参数	最小值	典型	最大值	单位	条件/备注
Ivdd 工作电流	Osc: 12 Bgr: 6.1	Osc: 17 Bgr: 9		uA	1.6~5.5v 27C tt trim set 100000
随温度变化范围		6 /100°C			-20~80 °C, tt 3.3v
随电源电压变化范围		1.05/V			2 ~ 5.5V, tt
标准振荡频率	—	15.99	—	MHz	3.3V , 27°C , tt
启动时间	—	2.2	10	uS	
漏电流	—	0.8	2	nA	Disable fosc&bgr
校准范围	—	+/-30	—		Step 0.9375

### 19.3 内置低频振荡器(LFINTOSC)

此低频振荡器有双模模式，一种模式下振动频率为 32KHz，另一种模式下振动频率为 256KHz。振荡频率模式由 OSCCON 寄存器中的 LFOMOD 位控制，0 为 32KHz 模式，1 为 256KHz 模式

电气参数	最小值	典型	最大值	单位	条件/备注
Ivdd 工作电流	0.4	1.1	1.4	uA	1.6~5.5V, tt
随温度变化范围		2% /100°C			-20 ~ 80 °C 2.5V
随电源电压变化范围		1.5%V			2 ~ 5.5V @27°C
标准振荡频率	—	31	—	KHz	2.5V, 25°C, TT, 32K mode
	—	248	—	KHz	2.5V, 25°C, TT, 256K mode
启动时间	—	4.6	10	uS	2.5V, 25°C, TT
漏电流	—	0.15	1	nA	disable



## 19.4 带隙基准源电路(Bandgap)

带隙基准电路电气参数如下：

电气参数	最小值	典型	最大值	单位	条件/备注
Ivdd 工作电流	6.1	9		uA	1.6~5.5V, tt
随温度变化范围	—	3	47	ppm/°C	-20 ~ 85 °C
输出电压范围	1.2013	1.233	1.269	V	Corner
PSRR		70		dB	tt
启动时间	—	1.2	10	uS	tt
漏电流	—	0.8	2	nA	Disable
输出驱动电流	—	—	—		No resistor loading

## 19.5 低压差线性稳压器(LDO)

带隙基准电路电气参数如下：

电气参数	最小值	典型	最大值	单位	条件/备注
随电源电压变化范围		1.65		V	VDD>1.7V
		VDD			VDD<1.65V

## 19.6 低电压复位电路(LVR)

电气参数	最小值	典型	最大值	单位	条件/备注
		1.8			
		2.0			
		2.2			
		2.8			
LVR delay		125	157	us	

## 19.7 低电压侦测电路(LVD)

电气参数	最小值	典型	最大值	单位	条件/备注
LVD threshold 可设置范围		1.6		V	
		1.8			
		2.0			
		2.4			
		2.8			
		3.0			
		3.6			
		4.2			
LVD delay		120		us	



## 19.8 上电复位电路(POR)

电气参数	最小值	典型	最大值	单位	条件/备注
Ivdd 工作电流		50*		nA	3.3V
随温度变化范围	1.4	1.64	1.8		3.3v, -20~80°C, tt

\* 设计参考值

## 19.9 I/O PAD 电路

电气参数	最小值	典型	最大值	单位	条件/备注
VIL		0.3*Vcc		V	
VIH		0.7*Vcc		V	
源电流		19		mA	5V, 25°C
沉电流		25		mA	5V, 25°C
上拉电阻		130		kΩ	5V
		230			3.3V

## 19.10 Comparator 比较器电路

电气参数	最小值	典型	最大值	单位	条件/备注
输入共模电压	0		Vdd		1.8~5.5V, -20°C~85°C
输入偏移电压(Offset)		±5	±10	mV	同上
共模抑制比(CMRR)		55		dB	同上
迟滞(Hysteresis)		0		mV	同上
响应时间 (Response Time)		200	1000	nS	正常模式:输出低->高
		150	600	nS	正常模式:输出高->低

## 19.11 10bit ADC 电路

电气参数	最小值	典型	最大值	单位	条件/备注
模拟输入电压	VSS	—	VREF	V	
分辨率	—	—	10	位	
外置参考电压	2		5.5	V	
2V 内置参考	1.99	2	2.01	V	
3V 内置参考	2.99	3	3.01	V	
积分线性误差		±1		dB	VREF=5.12V
微分线性误差		±1		dB	VREF=5.12V
转换时钟周期 TAD		2		us	VREF>3.0V
转换时钟数	—	11.5	—	TAD	





## 19.12 稳压器输出电路

电气参数	最小值	典型	最大值	单位	条件/备注
输出电压		2.4V		V	VREGM=00, VCC=3.3~5.5V
		3.6		V	VREGM=01, VCC=3.8~5.5V
		4.8		V	VREGM=10, VCC=5~5.5V
		5.24		V	VREGM=11, VCC=5.5V
输出电流		200		uA	VCC=3.3~5.5V

## 19.13 4bit DAC 电路 ( 比较器参考电压设置 )

电气参数	最小值	典型	最大值	单位	条件/备注
相对精度	—	VDD/16	—	V	1.8~5.5V -20°C~85°C
绝对精度	—	—	1/2*	LSB	同上
单位电阻(unit resistor)	—	5000*	—		同上
稳定时间(Settle Time)	—	—	10	uS	00000->11111

\* 为设计参考值

## 19.14 总体工作电流(I<sub>VDD</sub>)

电气参数	最小值	典型	最大值	单位	条件/备注
正常模式		310		uA	3.3V, 2MHz
		130		uA	1.8V, 2MHz
		74		uA	3.3V, 32KHz
		38		uA	1.8V, 32KHz
休眠模式 ( Sleep, WDT ON)		3		uA	3.3V
休眠模式 ( Sleep, WDT OFF)		0.8		uA	3.3V
休眠模式 ( Sleep, LVR ON)		15		uA	3.3V
休眠模式 ( Sleep, LVD ON)		15		uA	3.3V

注意：

1. 电流的测试条件为 IO 处于输入模式，IO 外部下拉。

## 20 指令集列表

本芯片采用精简指令架构，一共 37 条指令，以下是各指令的描述。

汇编语法	功能	运算	状态位
BCR R, b	Bit clear	0-> R(b)	NONE
BSR R, b	Bit set	1-> R(b)	NONE
BTSC R, b	Bit test, skip if 0	Skip if R(b)=0	NONE
BTSS R, b	Bit test, skip if 1	Skip if R(b)=1	NONE
NOP	No operation	None	NONE
CLRWDT	Clear WDT	0-> WDT	/PF, /TF
SLEEP	ENTER SLEEP MODE	0-> WDT, STOP OSC	/PF, /TF
STTMD	Store W TO TMODE	W-> TMODE <sup>1</sup>	NONE
CTLIO R	Control IO direction reg	W-> IODIRr	NONE
STR R(MOVWF)	Store W to reg	W-> R	NONE
LDR R, d(MOVF)	Load reg to d	R-> d	Z
SWAPR R, d	Swap halves reg	[R(0-3)R(4-7)]-> d	NONE
INCR R, d	Increment reg	R+ 1-> d	Z
INCRSZ R, d	Increment reg, skip if 0	R+ 1-> d	NONE
ADDWR R, d	Add W and reg	W+ R-> d	C, HC, Z
SUBWR R, d	Sub W from reg	R- W-> d R+ /W+ 1-> d	C, HC, Z
DECR R, d	Decrement reg	R- 1-> d	Z
DECRSZ R, d	Decrement reg, skip if 0	R- 1-> d	NONE
ANDWR R, d	AND W and reg	R& W-> d	Z
IORWR R, d	Inclu.OR W and reg	W  R-> d	Z
XORWR R, d	Exclu.OR W and reg	W^ R-> d	Z
COMR R, d	Complement reg	/R-> d	Z
RRR R, d	Rotate right reg	R(n)-> R(n-1), C-> R(7), R(0)-> C	C
RLR R, d	Rotate left reg	R(n)-> R(n+1), C-> R(0), R(7)-> C	C
CLRW	Clear working reg	0-> W	Z
CLRR R	Clear reg	0-> R	Z
RETI	Return from interrupt	Stack-> PC, 1-> GIE	NONE
RET	Return from subroutine	Stack-> PC	NONE
LCALL N	Long CALL subroutine	N-> PC, PC+1-> Stack	NONE



LJUMP N	Long JUMP address	N-> PC	NONE
LDWI I(MOVLW)	Load immediate to W	I-> W	NONE
ANDWI I	AND W and imm	W& I-> W	Z
IORWI I	Inclu.OR W and imm	WI I-> W	Z
XORWI I	Exclu.OR W and imm	W^ I-> W	Z
RETW I	Return, place imm to W	Stack-> PC, I-> W	NONE
ADDWI I	Add imm to W	W+I-> W	C, HC, Z
SUBWI I	Subtract W from imm	I-W-> W	C, HC, Z

操作码字段说明

Field	Description
R(F)	SFR/GPR address
W	Working register
b	Bit address within the 8-bit register/ram
I/Imm(k)	Immediate data
X	Don't care, may be 0 or 1
d	Destination select
	0: store result in W
	1: store result in register/ram
N	Immediate program address
PC	Program counter
TMODE	SFR TMODE <sup>1</sup>
IODIRr	SFR IODIR, r can be A, B, C
C	Carry bit
HC	Half carry
Z	Zero flag
/PF	Power down flag
/TF	Time out flag

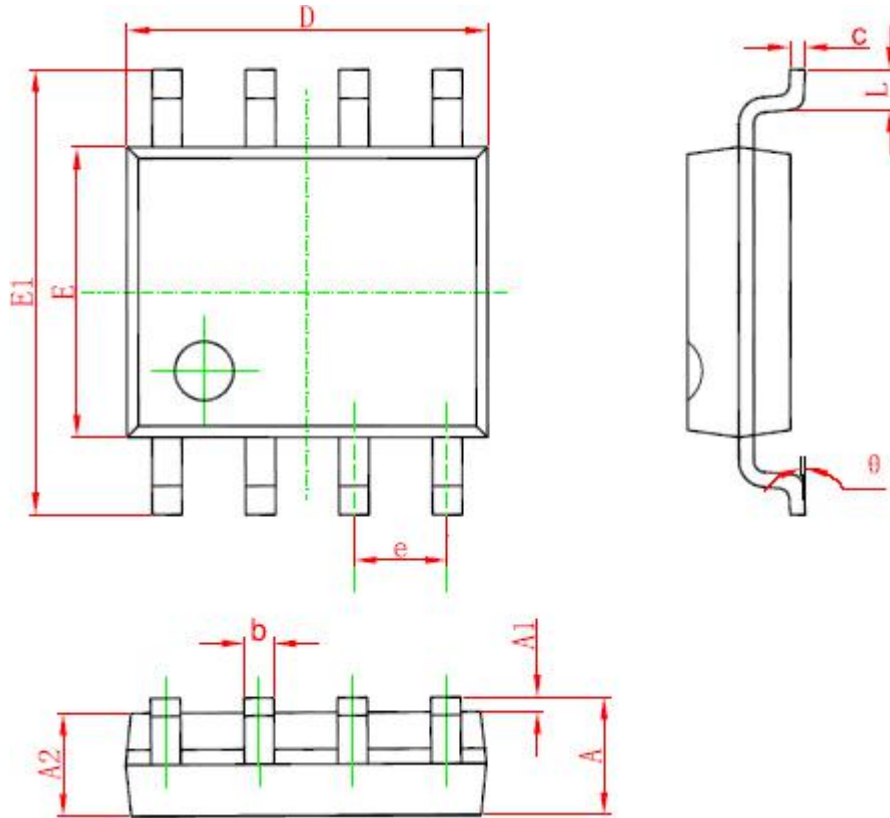
注意：

1. 在WF8M165系列芯片里，TMODE 寄存器是指 OPTION，即 STTMD 指令的操作是把 W 存到 OPTION；



## 21 芯片封装信息

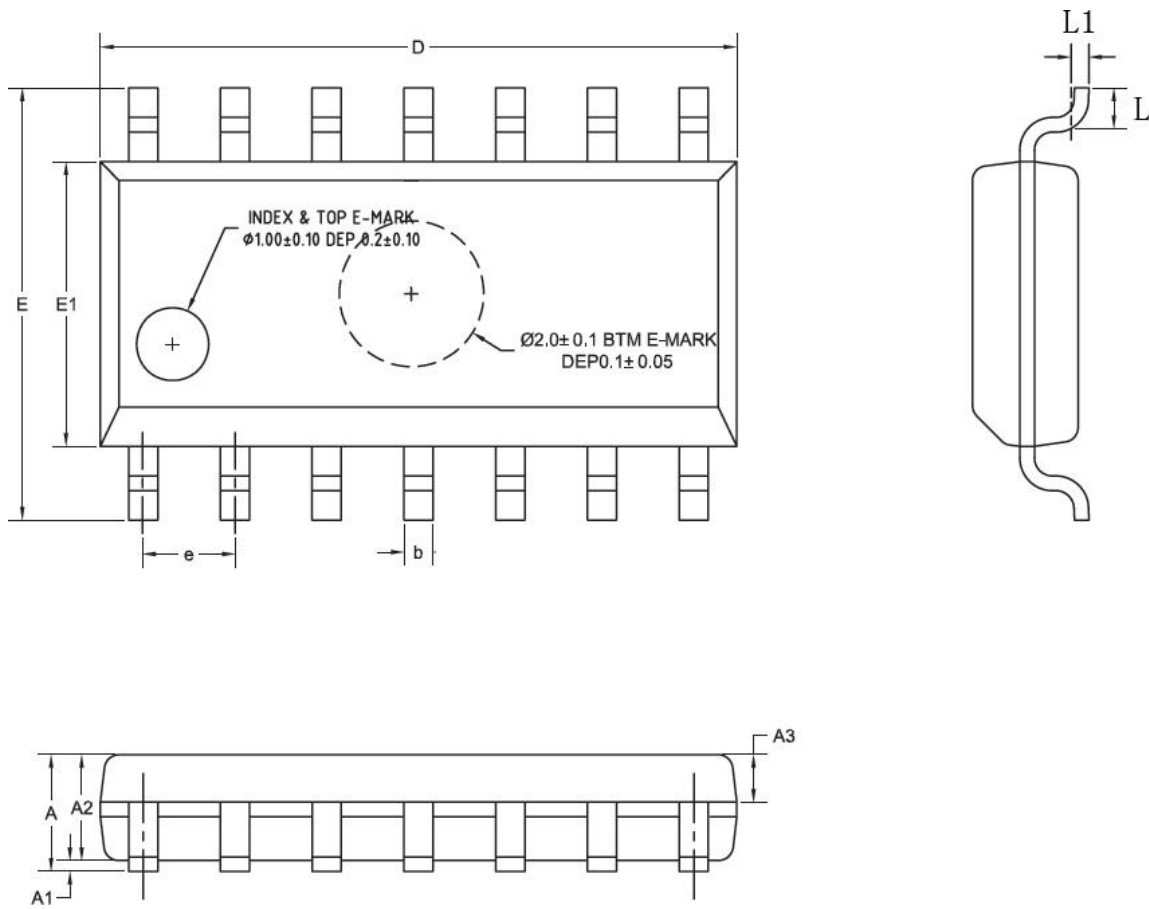
本芯片采用 SOP8、SOP14、SOP16 封装方式, 具体封装尺寸信息如下:  
SOP-8 封装尺寸如下:



Symbol	Dimensions In Millimeters		Dimensions In Inches	
	Min	Max	Min	Max
A	1.350	1.750	0.053	0.069
A1	0.100	0.250	0.004	0.010
A2	1.350	1.550	0.053	0.061
b	0.330	0.510	0.013	0.020
c	0.170	0.250	0.006	0.010
D	4.700	5.100	0.185	0.200
E	3.800	4.000	0.150	0.157
E1	5.800	6.200	0.228	0.244
e	1.270 (BSC)		0.050 (BSC)	
L	0.400	1.270	0.016	0.050
θ	0°	8°	0°	8°



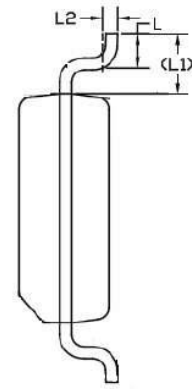
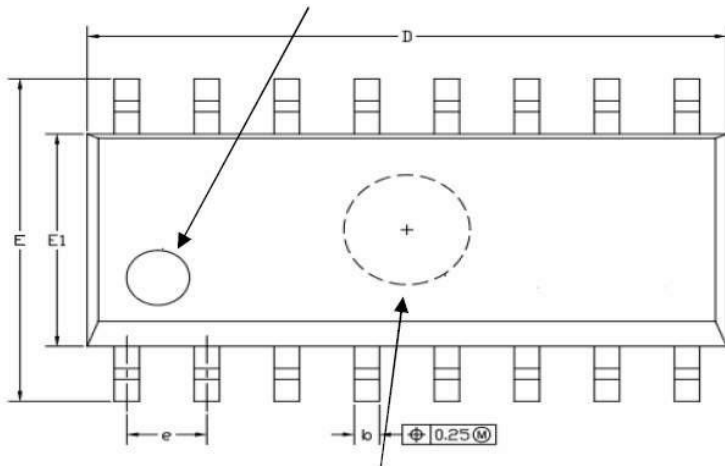
SOP-14 封装尺寸如下:



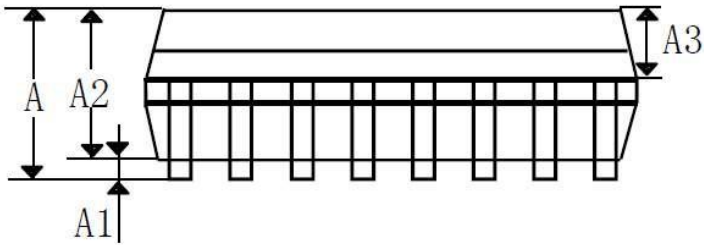
Symbol	Dimensions In Millimeters		Dimensions In Inches	
	Min	Max	Min	Max
A	-	1.700	-	0.066
A1	0.100	0.200	0.004	0.008
A2	1.400	1.500	0.054	0.059
A3	0.620	0.680	0.024	0.027
b	0.370	0.420	0.014	0.016
D	8.710	8.910	0.340	0.347
E	5.900	6.100	0.230	0.238
E1	3.800	3.950	0.148	0.154
e	1.270(BSC)		0.050(BSC)	
L	0.500	0.700	0.020	0.027
L1	0.250(BSC)		0.010(BSC)	



SOP-16 封装尺寸如下:



$\Phi 2.0 \pm 0.05$  DEP  $0.1 + 0.03 / -0.05$



Symbol	Dimensions In Millimeters		Dimensions In Inches	
	Min	Max	Min	Max
A	-	1.700	-	0.066
A1	0.100	0.200	0.039	0.008
A2	1.420	1.480	0.554	0.058
A3	0.620	0.680	0.242	0.027
D	9.960	10.160	3.884	0.396
E	5.900	6.100	2.301	0.238
E1	3.870	3.930	1.509	0.153
b	0.370	0.430	0.144	0.017
e	1.240	1.300	0.484	0.051
L	0.500	0.700	0.195	0.027
L1	1.050(REF)		0.041(REF)	
L2	0.250(BSC)		0.010(BSC)	



## 附录 1 · 文档更改历史

日期	版本	内容
2016-5-25	1.00	初版
2016-7-14	1.01	更新电流表格
2016-7-28	1.02	加入系统时钟频率对应的电压范围 F <sub>sys</sub> =8MHz: 1.8V~2.7V: F <sub>sys</sub> =16MHz: 2.7V~5.5V
2016-11-15	1.03	添加 EEPROM 的初始化操作到第 11 节，添加 PC0 作为外部参考输入的说明

## 附录 2 · 从其他器件移植

从WF8M163 移植到WF8M165 需要注意以下变化：

### 1. 寄存器地址变化

SFR \ PN	WF8M163	WF8M165
PIR2	-	0D
PIE2	-	9D

### 2. SFR 空间比较

地址 \ PN	WF8M163	WF8M165
0x0D	-	PIR2
0x0E	-	TMR1L
0x0F	-	TMR1H
0x10	-	T1CON
0x13	-	CCPR1L
0x14	-	CCPR1H
0x15	-	CCP1CON
0x16	-	PWM1CON
0x17	-	ECPAS
0x1E	-	ADRESH
0x1F	-	ADCON0
0x88	-	WPUC
0x89	-	WPD
0x8D	-	PIE2
0x90	-	PWM1AUX
0x91	-	ANSEL
0x100~0x17F	-	新增的一个 SFR bank

### 3. Status 寄存器变化

Status \ PN	WF8M163	WF8M165
bit7	- 未使用	FSR_B8，FSR 寄存器第 8 位，与 FSR 组成一个 9 位的寄存器，在间接寻址时使用，详见 INDF 和 FSR 寄存器一节
bit6:5	bit6 未使用，bit5 为 PAGE 位 1 = Bank 1 (0x80 ~ 0xFF) 0 = Bank 0 (0x00 ~ 0x7F)	PAGE[1:0]，Register Bank Select bit 00 = Bank0 (0x00~ 0x7F) 01 = Bank 1 (0x80~ 0xFF) 1x = Bank2 (0x100~ 0x17F)



4. PORTA5/TRISA5/WPUA5, PORTC[7:6]/TRISC[7:6]

SFR \ PN	WF8M163	WF8M165
PORTA[5]	未实现 只有输入功能	PA5 端口数据位 当 MCLRE 为 0 时，PA5 为 GPIO
TRISA[5]	未实现，读 1	PA5 端口方向控制位 当 MCLRE 为 0 时，控制 PA5 输入或输出
WPUA[5]	未实现，读 0	PA5 作为输入时的上位使能位
PORTC[7:6]	PORTC7/6 的数据寄存器	-
TRISC[7:6]	PORTC7/6 的输入输出控制寄存器	-
WPUC[5:0]	-	PORTC 的输入上拉控制
WPD[4:1]	-	PORTA[4]，PORTC[3:1] 的输入下拉控制

5. 编程数据 EEPROM 步骤

WF8M163	WF8M165
<p>A. 把位 WREN3/WREN2/WREN1 全部置 1；</p> <p>B. 往 EEADR 写入目标地址；</p> <p>C. 往 EEDAT 写入目标数据；</p> <p>D. 把位 WR 置 1 (EECON2.0，此后 WR 会维持高)；</p> <p>E. 写过程不能改变 WREN3/2/1 的值，否则编程终止；</p> <p>F. 等大概 2ms 之后编程自动完成，WR 自动清 0，WREN3、WREN2、WREN1 清 0；</p> <p>G. 如果想再次编程，重复步骤 A~F 即可；</p>	<p>A. 往 EEADR 写入目标地址；</p> <p>B. 往 EEDAT 写入目标数据；</p> <p>C. 把位 WREN3/WREN2/WREN1 全部置 1；</p> <p>D. 把位 WR 置 1 (EECON2.0，此后 WR 会维持高)；</p> <p>E. 写过程不能改变 WREN3/2/1 的值，否则编程终止；</p> <p>F. 最后恢复 GIE，使能中断；</p> <p>G. 等大概 2ms 之后编程自动完成，WR 自动清 0，WREN3、WREN2、WREN1 清 0；</p> <p>H. 如果想再次编程，重复步骤 A~F 即可；</p> <p><b>注意：</b></p> <ul style="list-style-type: none"> <li>● 以上步骤的 D、E 两步必须是连续的两条指令周期完成，不能错开，否则编程操作不会启动，其中 WREN3、WREN2 和 WREN1 可以不是同一条指令置 1，比如可以用 BSR 指令分开对各位置 1；</li> <li>● 如果 D、E 两步被错开执行，要想启动下一次编程操作，必须在 D、E 之前加入一步，把 WREN3、WREN2 或者 WREN1 任意一位清 0；</li> </ul>



6. UCFG1 寄存器

WF8M163		WF8M165	
Config1 寄存器		Config1 寄存器	
位	名称	位	名称
Bit[7]	-	Bit[7]	CSUMENB

7. PCON 寄存器变化

WF8M163		WF8M165	
位	名称	位	名称
Bit[7]	-	Bit[7]	VREF_OE
Bit[6:4]	-	Bit[6:4]	LVDL
Bit[3]	-	Bit[3]	LVDW
Bit[2]	-	Bit[2]	-
Bit[1]	/POR	Bit[1]	/POR
Bit[0]	/BOR	Bit[0]	/BOR

8. LVREN<1:0>变化

WF8M163		WF8M165	
LVREN[1:0]	低电压复位选择 00：使能低电压复位 其它值：禁止低电压复位	LVREN[1:0]	低电压复位选择 00：使能低电压复位 01 LVR 由 MSCKCON 的 SLVREN 决定 10：MCU 正常模式时开启 LVR，睡眠模式时关闭 LVR，跟 SLVREN 位无关 11：禁止低电压复位



9. MSCKCON 变化

WF8M163			WF8M165		
7:5	-	保留位，不能写 1	7	-	-
4	SLVREN	软件控制 LVR 使能位 1 = 当 UCFG1<1:0> 为 00 时，打开 LVR 0 = 无论 UCFG1<1:0> 为何值，禁止 LVR	6	VREG_OE	稳压器输出使能 1 = 打开片内稳压器，PA4 和 PC5 输出稳压器电压 0 = PA4 和 PC5 为普通 IO
3	-	保留位，不能写 1	5	T2CKSRC	TIMER2 时钟源选择 0 = TIMER2 时钟源为系统时钟 1 = TIMER2 时钟源为内部 32MHz
2	CKMAVG	快时钟测量慢时钟周期的测量平均模式 1 = 打开平均模式 (自动测量并累加 4 次) 0 = 关闭平均模式	4	SLVREN	软件控制 LVR 使能位，当 UCFG1<1:0> 为 01 时： 1 = 打开 LVR 0 = 禁止 LVR 当 UCFG1<1:0> 不为 01 时，此位无实际意义
1	CKCNTI	Clock Count Init -使能快时钟测量慢时钟周期 1 = 使能快时钟测量慢时钟周期 0 = 关闭快时钟测量慢时钟周期 注：这一位在测量完毕后会自动归零	3	-	保留位，不能写 1
0	-	保留位，不能写 1	2	CKMAVG	快时钟测量慢时钟周期的测量平均模式 1 = 打开平均模式 (自动测量并累加 4 次) 0 = 关闭平均模式
			1	CKCNTI	Clock Count Init -使能快时钟测量慢时钟周期 1 = 使能快时钟测量慢时钟周期 0 = 关闭快时钟测量慢时钟周期 注：这一位在测量完毕后会自动归零
			0	-	保留位，不能写 1