



AD008

用户手册

版本号: V1.0.0.0

版权所有©

西安恩狄集成电路有限公司

本资料内容为西安恩狄集成电路有限公司在现有数据资料基础上编制而成，本资料中所记载的实例以正确的试用方法和标准操作为前提，使用方在应用该等实例时应充分考虑外部诸条件，西安恩狄集成电路有限公司不担保或确认该等实例在使用方的适用性、适当性或完整性，西安恩狄集成电路有限公司亦不对使用方使用本资料所有内容而可能或已经带来的风险或后果承担任何法律责任。文档中所有涉及到第三方软件的，请自行购买正版软件，因第三方软件版权问题涉及到的一切后果，与西安恩狄集成电路有限公司无关。基于使本资料的内容更加完善等原因，西安恩狄集成电路有限公司保留未经预告的修改权。

西安恩狄集成电路有限公司

地 址：陕西省西安市高新区高新一路 19 号思安大厦 501

地 址：深圳市龙岗区坂田国际中心 A 栋 21 层 2112 室

电 话：+ (86 29) 88322766 网 站：www.admicrochip.com

微信号：恩狄 ADUC



版本修订记录

Bin	Version	Change List	Owner	Data
1	1.0.0.0	初版	Aaron	2022.8.29

目录

版本修订记录	1
1 产品简介	9
1.1 功能特性	9
1.2 系统结构	11
1.3 引脚排列	11
1.4 引脚说明	12
1.5 烧录引脚说明	17
1.6 极限参数	17
2 中央处理器	18
2.1 指令集	18
2.2 程序存储	18
2.2.1. 程序计数器	18
2.2.2. 返回地址堆栈	18
2.2.3. 栈顶访问	18
2.2.4. 快速寄存器堆栈	20
2.2.5. 程序存储器	20
2.3 数据存储器 RAM	21
2.4 数据寻址方式	22
2.4.1. 固有和立即数寻址	22
2.4.2. 直接寻址	22
2.4.3. 间接寻址	22
2.5 配置选项	23
3 EEPROM 和 FLASH	27
3.1 EEPROM	27
3.1.1. EECON1 和 EECON2 寄存器	27
3.1.2. 读数据 EEPROM 存储器	28
3.1.3. 写数据 EEPROM 存储器	29
3.1.4. 写校验	29
3.1.5. 避免误写操作	29
3.2 FLASH	30
3.2.1. IAPTRIG 寄存器	30
3.2.2. IAPCTRL 寄存器	30
3.2.3. IAPADDRH 寄存器	31

3.2.4.	IAPADDR1 寄存器	31
3.2.5.	IAPDATA 寄存器	31
3.2.6.	IAPDATA1 寄存器	31
3.2.7.	IAPWAIT 寄存器	31
3.2.8.	SIGCON 寄存器	32
4	系统时钟源	33
4.1	系统时钟相关寄存器	33
4.1.1.	OSCCON 寄存器	33
4.1.2.	CLKCFG1 寄存器	33
4.1.3.	CLKCFG2 寄存器	34
4.2	双速时钟启动模式	35
4.3	双速启动模式配置	35
4.4	双速启动顺序	35
4.5	故障保护时钟监控器	35
4.5.1.	故障保护操作	36
4.5.2.	故障检测完成	36
5	复位和电源电压检测	37
5.1	上电复位计数器(PWRT)	37
5.2	振荡启动计数器(OST).....	40
5.3	复位顺序	40
5.4	LVD 检测	42
5.4.1.	PCON(寄存器).....	42
5.4.2.	RCON(寄存器)	43
6	I/O 端口	44
6.1	IO 工作模式.....	44
6.1.1.	PORTA (Port 寄存器)	44
6.1.2.	PORTB (Port 寄存器)	44
6.1.3.	PORTC (Port 寄存器)	44
6.1.4.	TRISA (I/O 口方向控制寄存器).....	44
6.1.5.	TRISB (I/O 口方向控制寄存器).....	45
6.1.6.	TRISC(I/O 口方向控制寄存器).....	45
6.1.7.	PINA (数据锁存寄存器)	45
6.1.8.	PINB (数据锁存寄存器)	45
6.1.9.	PINC (数据锁存寄存器)	45

6.2	下拉电阻开漏	46
6.2.1.	PAPD(I/O 下拉控制寄存器)	46
6.2.2.	PAPU (I/O 上拉控制寄存器)	46
6.2.3.	PAOD (I/O 开漏控制寄存器).....	46
6.2.4.	PBPD(I/O 下拉控制寄存器).....	46
6.2.5.	PBPU (I/O 上拉控制寄存器)	46
6.2.6.	PBOD (I/O 开漏控制寄存器).....	47
6.2.7.	PCPD(I/O 下拉控制寄存器)	47
6.2.8.	PCPU (I/O 上拉控制寄存器)	47
6.2.9.	PCOD (I/O 开漏控制寄存器).....	47
6.2.10.	PAINTMASK (Port A 端口电平变化中断掩膜位).....	47
6.2.11.	PBINTMASK (Port B 端口电平变化中断掩膜位).....	48
6.2.12.	PCINTMASK(Port C 端口电平变化中断掩膜位).....	48
6.3	模拟 IO 寄存器.....	48
6.3.1.	ANASEL0 (IO 的模拟通道寄存器).....	48
6.3.2.	ANASEL1 (IO 的模拟通道寄存器).....	48
6.3.3.	ANASEL2 (IO 的模拟通道寄存器).....	49
6.3.4.	SMTVA (PA 口施密特寄存器).....	49
6.3.5.	SMTVB (PB 口施密特寄存器)	49
6.3.6.	SMTVC (PC 口施密特寄存器)	49
6.3.7.	CURCON(寄存器读写).....	50
7	定时器	51
7.1	Timer0/Prescaler/BUZZER/PWM.....	51
7.1.1.	Timer0 计数/定时	51
7.1.2.	使用内部时钟: 定时模式	51
7.1.3.	使用外部时钟/内部 1M 时钟/: 计数模式	51
7.1.4.	Prescaler (预置器)	51
7.1.5.	BUZZER (BUZZER 输出)	51
7.1.6.	TMR0 与 4 路 PWM.....	52
7.2	TIMER1 16 位定时/计数器	54
7.2.1.	TMR1L (Timer1 16 位低 8 位寄存器).....	55
7.2.2.	TMR1H (Timer1 16 位高 8 位寄存器).....	55
7.2.3.	T1CON0 (Timer1 控制寄存器)	55
7.2.4.	T1CON1(Timer1 控制寄存器)	56
7.3	TIMER2 定时器.....	57

7.3.1.	PR2L (timer2 的周期寄存器)	58
7.3.2.	PR2H timer2 的周期寄存器)	58
7.3.3.	TMR2L (Timer2 的低八位寄存器)	58
7.3.4.	TMR2H (Timer2 的高八位寄存器)	58
7.3.5.	T2CON0(timer2 控制寄存器)	59
7.4	TIMER3 定时器和 4 路 12 位 PWM	59
7.4.1.	TIMER3 定时器	59
7.4.2.	4 路 12 位 PWM	61
7.4.3.	4 路 PWM 的输出	62
7.4.4.	4 路 PWM 的周期	63
7.4.5.	PWM 的占空比	63
7.4.6.	PWM 的分辨率	63
7.4.7.	PWM 的工作设置	63
7.5	看门狗定时器 (WDT)	66
7.5.1.	WDTCON(看门狗的控制寄存器)	67
8	捕获/比较/PWM 模块	68
8.1	捕捉模式	69
8.2	比较模式	71
8.2.1.	PWM 中心对齐模式	72
8.3	3 组互补 PWM 输出	73
8.3.1.	PWM1CON1(PWM 控制寄存器)	73
8.3.2.	PWM1CON0(PWM 的控制寄存器)	74
8.3.3.	PMS 寄存器	75
8.3.4.	PXC(PWM 控制输出寄存器)	76
8.3.5.	DTC (死区时间控制寄存器)	76
8.3.6.	POLS (极性选择寄存器)	77
8.3.7.	PME 寄存器	77
8.3.8.	PWM 调制	78
8.3.9.	死区时间	79
8.3.10.	互补式输出控制防呆电路	79
8.3.11.	增强型六路 PWM	82
9	通用同步/异步收发器 (USART)	89
9.1	波特率发生器	91
9.2	异步发送器	92
9.3	异步接收	93

9.4	同步模式	95
9.4.1.	同步主机发送	95
9.4.2.	同步主机接收	96
9.5	同步从机模式	97
9.5.1.	同步从机发送	97
9.5.2.	同步从机接收	98
10	I2C 控制器	99
10.1	I ² C 相关寄存器:	99
10.1.1.	I2CACKDLY (I2C 的延时控制寄存器)	99
10.1.2.	I2CTX (I2C 的发送寄存器)	100
10.1.3.	I2CRC (I2C 的接收寄存器)	100
10.1.4.	I2CADD0 (I2C 的地址寄存器)	100
10.1.5.	I2CADD1 (I2C 的地址寄存器)	100
10.1.6.	I2CMASK (I2C 的寄存控制器)	100
10.1.7.	I2CSTAT (I2C 的控制寄存器)	101
10.1.8.	I2CCON0 (I2C 的控制寄存器)	101
10.1.9.	I2CCON1 (I2C 的控制寄存器)	102
10.1.10.	I2CIE 寄存器	102
10.1.11.	I2CIF 寄存器	103
10.1.12.	I2CIP 寄存器	104
10.2	从机模式	104
11	中断	107
11.1	外部中断	107
11.2	Timer0 中断	108
11.3	Timer1 中断	108
11.4	Timer2 中断	108
11.5	Timer3 中断	108
11.6	PortA 输入改变中断	108
11.7	PortB 输入改变中断	108
11.8	PortC 输入改变中断	108
11.9	低电压、高电压中断	109
11.10	运放中断	109
11.11	比较器中断	110
11.12	ADC 中断	110
11.13	中断的相关寄存器	110

11.13.1.	INTCON 寄存器.....	110
11.13.2.	IPR1 寄存器.....	110
11.13.3.	PIR1 寄存器.....	111
11.13.4.	PIE1 寄存器.....	111
11.13.5.	IPR2 寄存器.....	112
11.13.6.	PIR2 寄存器.....	113
11.13.7.	PIE2 寄存器.....	113
11.13.8.	IPR3 寄存器.....	114
11.13.9.	PIR3 寄存器.....	115
11.13.10.	PIE3 寄存器.....	115
12	省电模式 (SLEEP).....	117
12.1	睡眠唤醒.....	117
12.1.1.	SMCR (状态控制寄存器).....	119
13	固定参考电压 (FVR).....	121
13.1	FVRCON 寄存器.....	121
14	多通道输入模拟缓冲器 (BUFFER UNIT).....	122
14.1	BUCON 寄存器.....	122
15	模数转换器 (ADC).....	124
15.1	ADCMP0H (AD 比较寄存器 0 高字节).....	125
15.2	ADCMP1H(AD 比较寄存器 1 高字节).....	125
15.3	ADCMP01L(AD 比较寄存器 0 和 1 低字节).....	126
15.4	ADRESH (AD 转换结果的高四位).....	126
15.5	ADRESL(AD 转换结果的低四位).....	126
15.6	ADCON0 (ADC 控制寄存器 0).....	126
15.7	ADCON1 (ADC 控制寄存器 1).....	127
15.8	ADCON2 (ADC 控制寄存器 2).....	128
16	数模转换器(DAC0 和 DAC1).....	129
16.1	DAC 参考电压选择寄存器.....	130
16.2	DAC0 控制寄存器.....	131
16.3	DAC1 控制寄存器.....	131
17	运放 (OP0 和 OP1).....	132
17.1	运放 OP0.....	132
17.1.1.	OPOCON0(FOP 控制寄存器).....	132
17.1.2.	OPOCON1(OP 控制寄存器).....	133
17.1.3.	OPOCON2(OP 控制寄存器).....	134

17.1.4.	OPOCON3(OP 控制寄存器)	134
17.2	运放 OP1.....	135
17.2.1.	OP1CON0(OP 控制寄存器)	135
17.2.2.	OP1CON1(OP 控制寄存器)	136
17.2.3.	OP1CON2(OP 控制寄存器)	137
17.2.4.	OP1CON3(OP 控制寄存器)	137
17.3	运放 OP 电流偏置控制寄存器.....	138
17.3.1.	OPCON(OP 控制寄存器)	138
17.4	OP0 与 OP1 的输入失调校准	139
17.4.1.	OP0OFFSET 校准.....	139
17.4.2.	OP1OFFSET 校准:.....	140
18	比较器 (CMP0 和 CMP1)	141
18.1	比较器 CMP0.....	141
18.1.1.	CMP0CON(比较器控制寄存器)	141
18.2	比较器 CMP1.....	142
18.2.1.	CMP1CON(比较器控制寄存器)	142
18.3	比较器输出控制寄存器	143
18.3.1.	CMPCON(比较器控制寄存器)	143
19	LCD 驱动模块	145
19.1	LCD 管脚设置	145
19.2	LCD 相关寄存器	145
19.2.1.	LCDCON(LCD 使能寄存器)	145
19.2.2.	LCDCON1(比较器控制寄存器).....	145
19.2.3.	LCDCON2(比较器控制寄存器).....	146
19.2.4.	LCDCON3(比较器控制寄存器).....	146
20	8X8 硬件乘法器	148
21	电气特性	149
22	封装尺寸	154

1 产品简介

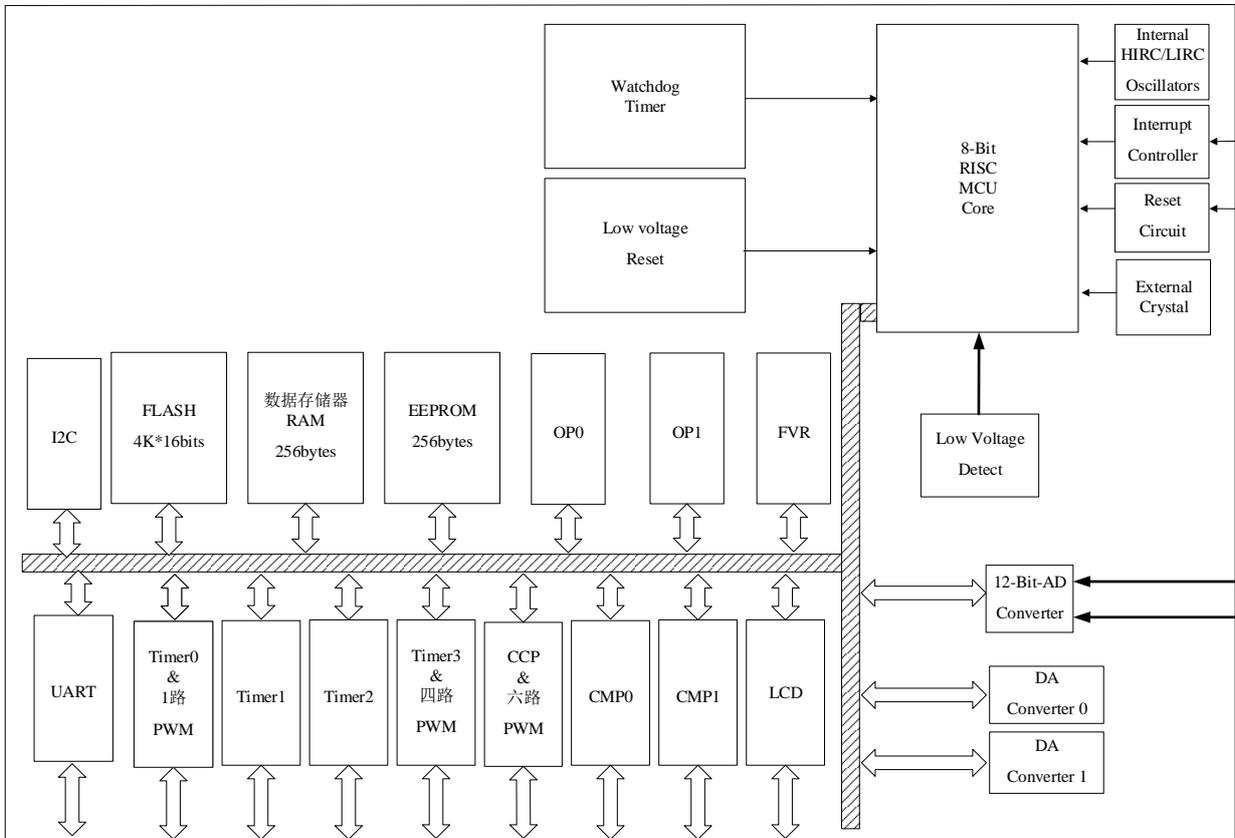
1.1 功能特性

- 8位 RISC CPU 内核，FOSC=16MHz，支持 2T、4T CPU 运算模式
- 支持 8 位硬件乘法器
- 支持在线调试，4 个硬件断点
- 内置 4K*16 bit FLASH(8K Bytes)可编程存储器，256 Byte EEPROM，256 Byte SRAM，8 级深度硬件堆栈；支持 IAP 功能；程序存储空间的 Checksum 自动校验；
- CPU 双时钟模式，高速可选 FCPU、FCPU/2、FCPU/4、FCPU/8、FCPU/16、FCPU/32；低速时钟可选 1M/512K/256K/128K/64K/32K/16K，并可以作为 FCPU 时钟；通过寄存器可实时切换 CPU 时钟；
- CPU 有 IDLE, PWSAVE, DEEPPWSAVE, PWOFF 四种工作模式；
- TIMER0，8 位定时/计数器，可选 FCPU、OSC32K(32K 时钟)、T0CKI、运放输出作为 TIMER0 时钟，同时支持 BUZZER 输出模式，支持 1 路 8 位硬件 PWM，可同时输出到 4 个不同 IO；
- TIMER1，16 位定时/计数器，有多个时钟输入选择，可以工作在异步时钟工作模式，同时支持 IO 管脚输入捕捉功能；可以做长定时，支持 1 秒定时；能定时启动 ADC 采集；
- TIMER2，12 位定时器，时钟源为 Fosc；
- TIMER3，12 位定时器，时钟源为 Fosc；支持 4 路独立的 12 位 PWM 输出；
- CCP，与 TIMER1 构成 16 位精度的一路捕获，支持多种捕获定义，并可同时记录上升、下降沿捕获记录；与 TIMER1 构成 16 位精度的一路数字比较，可以控制 IO 输出或者启动 ADC 采集；与 Timer2 构成 6 路 12 位精度的三路互补 PWM 输出半桥控制；支持 OP、CMP、IO 关闭和启动 PWM；
- UART：同步、异步、半双工和单线串口，支持 1bit、2bit、3bit 停止位
- 硬件从 I2C 接口，支持 100K，400K，1MBPS，支持两个 7 位从地址应答寄存器，可设定是否产生应答
- 内置高精度电压源，提供电压 0.3V、1.0V、1.1V、1.2V、2.0V、2.2V、2.4V；误差小于 1%；
- 内置多通道模拟输入缓冲器 BUFFER，可以输入温度传感器（NTC）值；可以输入 1/4VCC 电压，用于系统电压采集；
- 12bit 高精度的 ADC，支持多个 IO 引脚采集，同时支持内部模拟信号采集；支持自动采集比较模式，可用于低功耗下信号采集；支持 2 个电压采集比较值，实现高精度比较器功能；
- LVR 提供 14 种低电压选择，支持低电压中断，同时也支持高电压中断，高电压中断可选择自动强制 PB0 输出指定电平，可实现电源电压采集；
- 2 组低失调和高增益运算放大器，灵敏度高并能有效抑制误触发；支持 IO、OP、FVR、ADC BUFER 输入；运放可以输出到多个管脚；支持单独配置功能；支持双沿产生中断；支持输入失调电压校准；
- 2 路 6 位精度的 DAC；可以输出到 OP、CMP、ADC、ADC BUFFER；支持单独配置使能功能；DAC 基准可以分开控制；

- 2 组比较器，用于比较输入端电压的大小，输入电压比较范围为 0~VCC/2；CMP 输出可以输出 IO；比较器输入支持 IO、DAC0、DAC1、OP0、OP1、FVR、ADC BUFFER；支持单独配置使能功能；支持双沿产生中断；
- 支持 1/2 偏压 LCD 驱动；
- 超强程序加密算法，保证芯片程序内容唯一性；
- 运行速度：支持最高 16 MHz CPU 运行工作频率；
- 上电复位计数器（PWRT）和振荡启动计数器（Oscillator Start-up Timer OST）
- 内部振荡器集成了一个看门狗保证了可靠的操作同时软件使能看门狗操作
- IO 口的 SINK 电流支持 2 档选择。其中 PB2 引脚支持 70mA SINK 驱动电流和 50mA 的 SOURCE 驱动电流；
- 通过编程控制 I/O 端口的上拉/下拉、开漏等状态；上拉电阻支持两类选择；
- 中断：
 - ◇ 三个内部计数/定时器中断源；
 - ◇ 三个外部 IO 管脚中断源：INT 管脚；
 - ◇ PA、PB、PC 的输入改变中断源
 - ◇ CCP 中断源；
 - ◇ 低电压、高电压 LVD 中断源；
 - ◇ 运放输出电压比较中断源；
 - ◇ ADC 中断；
 - ◇ OP 中断
 - ◇ CMP 中断
 - ◇ 时钟监测中断
 - ◇ EEPROM 写中断
- 振荡器
 - ◇ 内部 1.024MHz RC 振荡器，分频到 32KHZ 用于看门狗计数
 - ◇ 内部 16MHZ RC 振荡器，用于系统时钟
 - ◇ 外部 32768 低频晶体振荡器，用于系统时钟或 RTC
 - ◇ 外部高频晶体振荡器 412KHz-16MHz
- 通过外部中断、PortA 中断、PortB 中断、PortC 中断、LVD 中断、TIMER 中断、CCP 比较模式、WDT 溢出和外部复位实现睡眠模式唤醒。
- 内部有 16MHz RC 振荡器（1%误差），以及 1.024MHz RC 振荡器（5%误差）；
- 有可靠的保证使得程序代码不被读出；
- VCC 工作电压范围：2.2V - 5.5V，CPU 工作的最高频率为 16MHz；
- 封装：SOP16，TSSOP20，QFN20

1.2 系统结构

注：AD008 EEPROM 功能暂未开放



1.3 引脚排列

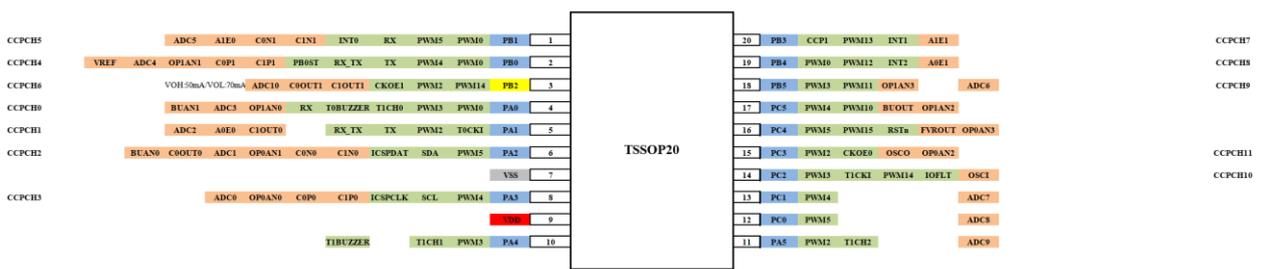


图1. TSSOP20 引脚排列

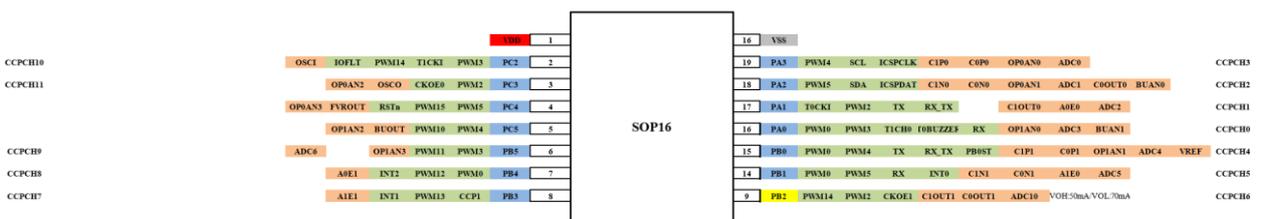


图2. SOP16 引脚排列

1.4 引脚说明

管脚名	功能名	输入类型	输出类型	具体描述	支持开漏	上下拉
PA0	PA0	SMT	CMOS	GPIO	Y	Y
	PWM0	-	CMOS	PWM0 的输出		
	PWM3	-	CMOS	PWM3 的输出		
	T1CH0	SMT	-	TIMER1 的脉宽输入通道		
	TOBUZZER	-	CMOS	TIMERO 的 BUZZ 输出		
	RX	SMT	-	UART 的接收 IO		
	OP1AN0	AN	-	OP1 通道 0 输入		
	ADC3	AN	-	ADC 通道 3 输入		
	BUAN1	AN	-	BUFF 的输入通道		
	COM0	AN	-	LCD COM 输出		
PA1	PA1	SMT	CMOS	GPIO	Y	Y
	TOCKI	SMT	-	Timer0 时钟源输入		
	TX	SMT	-	UART 的发送 IO		
	RX_TX	SMT	-	UART 的单线半双工通信 IO		
	PWM2	-	CMOS	PWM2 的输出		
	C1OUT0	-	CMOS	CMP1 输出引脚		
	AOEO	AN	-	OPO 输出引脚		
	ADC2	AN	-	ADC 通道 2 输入		
	COM1	AN	-	LCD COM 输出		
PA2	PA2	SMT	CMOS	GPIO	Y	Y
	PWM5	-	CMOS	PWM 的输出		
	ICSPDAT	-	CMOS	芯片烧录的数据引脚		
	BUAN0	AN	-	BUFF 的输入通道		
	COOUT0	-	CMOS	CMP0 的输出引脚		
	C1N0	AN	-	CMP1 负端输入通道 0		
	CON0	AN	-	CMP0 负端输入通道 0		
	OPOAN1	AN	-	OPO 通道 1 输入		
	ADC1	AN	-	ADC 通道 1 输入		

	SDA	-	CMOS	I2C 的数据引脚 IO		
	COM2	AN	-	LCD COM 输出		
PA3	PA3	SMT	CMOS	GPIO	Y	Y
	PWM4	-	CMOS	PWM4 的输出		
	C1P0	AN	-	CMP1 正端输入通道 0		
	COPO	AN	-	CMPO 正端输入通道 0		
	OPOANO	AN	-	OPO 通道 0 输入		
	ADCO	AN	-	ADC 通道 0 输入		
	ICSPCLK	SMT	-	芯片烧录的时钟引脚		
	SCL	SMT	-	I2C 的时钟引脚 IO		
	COM3	AN	-	LCD COM 输出		
PA4	PA4	SMT	CMOS	GPIO	Y	Y
	PWM3	-	CMOS	PWM3 的输出		
	T1BUZZER	-	CMOS	TIMER1 的 BUZZ 输出		
	T1CH1	SMT	-	TIMER1 的脉宽输入通道		
	COM4	AN	-	LCD COM 输出		
PA5	PA5	SMT	CMOS	GPIO	Y	Y
	PWM2	-	CMOS	PWM2 的输出		
	T1CH2	SMT	-	TIMER1 的脉宽检测输入引脚		
	ADC9	AN	-	ADC 通道 9 输入		
	COM5	AN	-	LCD COM 输出		
PB0	PB0	SMT	CMOS	GPIO	Y	Y
	PWM0	-	CMOS	PWM0 的输出		
	PWM4	-	CMOS	PWM4 的输出		
	TX	-	-	UART 的发送 IO		
	RX_TX	SMT	CMOS	UART 的单线半双工通信 IO		
	PBOST	-	CMOS	LVR 中电源电压高于阈值电压, 强制输出引脚		
	C1P1	AN	-	CMP1 正端输入通道 1		
	COPI	AN	-	CMPO 正端输入通道 1		
	OP1AN1	AN	-	OP1 通道 1 输入		

	ADC4	AN	-	ADC 通道 4 输入		
	VREF	AN	-	ADC 的外部参考电压输入		
	COM6	AN	-	LCD COM 输出		
PB1	PB1	SMT	CMOS	GPIO	Y	Y
	PWM0	-	CMOS	PWM0 的输出		
	PWM5	-	CMOS	PWM5 的输出		
	RX	SMT	-	UART 的接收 IO		
	C1N1	AN	-	CMP1 负端输入通道 1		
	CON1	AN	-	CMPO 负端输入通道 1		
	A1E0	AN	-	OP1 输出引脚		
	ADC5	AN	-	ADC 通道 5 输入		
	INT0	SMT	-	外部中断 0		
	COM7	AN	-	LCD COM 输出		
PB2	PB2	SMT	CMOS	GPIO	Y	Y
	PWM14	-	CMOS	PWM14 的输出		
	PWM2	-	CMOS	PWM2 的输出		
	C1OUT1	AN	-	CMP1 输出引脚		
	COOUT1	AN	-	CMPO 输出引脚		
	CKOE1	-	CMOS	系统时钟的输出 IO		
	ADC10	AN	-	ADC 的输入端		
	COM8	AN	-	LCD COM 输出		
PB3	PB3	SMT	CMOS	GPIO	Y	Y
	CCP1	-	CMOS	CCP 的输出引脚		
	PWM13	-	CMOS	PWM13 的输出		
	INT1	SMT	-	外部中断 1		
	A1E1	AN	-	OP1 输出引脚		
	COM9	AN	-	LCD COM 输出		
PB4	PB4	SMT	CMOS	GPIO	Y	Y
	PWM0	-	CMOS	PWM0 的输出		
	PWM12	-	CMOS	PWM12 的输出		

	INT2	SMT	-	外部中断 2		
	AOE1	AN	-	OPO 输出通道 1		
	COM10	AN	-	LCD COM 输出		
PB5	PB5	SMT	CMOS	GPIO	Y	Y
	PWM3	-	CMOS	PWM3 的输出		
	PWM11	-	CMOS	PWM11 的输出		
	ADC6	AN	-	ADC 通道 6 输入		
	OP1AN3	AN	-	OP1 通道 3 输入		
	COM11	AN	-	LCD COM 输出		
PC0	PC0	SMT	CMOS	GPIO	Y	Y
	PWM5	-	CMOS	PWM5 的输出		
	ADC8	AN	-	ADC 通道 8 输入		
	COM12	AN	-	LCD COM 输出		
PC1	PC1	SMT	CMOS	GPIO	Y	Y
	PWM4	-	CMOS	PWM4 的输出		
	ADC7	AN	-	ADC 通道 7 输入		
	COM13	AN	-	LCD COM 输出		
PC2	PC2	SMT	CMOS	GPIO	Y	Y
	PWM3	-	CMOS	PWM3 的输出		
	PWM14	-	CMOS	PWM14 的输出		
	IOFLT	-	CMOS	故障检测输入引脚		
	T1CKI	SMT	-	Timer1 时钟源输入		
	OSCI	SMT	-	外部晶振的输入引脚		
	COM14	AN	-	LCD COM 输出		
PC3	PC3	SMT	CMOS	GPIO	Y	Y
	PWM2	-	CMOS	PWM2 的输出		
	CKOE0	-	CMOS	系统时钟的输出引脚		
	OSCO	-	CMOS	外部晶振的输出引脚		
	OPOAN2	AN	-	OPO 通道 2 输入		
	COM15	AN	-	LCD COM 输出		

PC4	PC4	SMT	CMOS	GPIO	Y	Y
	PWM5	-	CMOS	PWM5 的输出		
	PWM15	-	CMOS	PWM15 的输出		
	RSTn	SMT	-	芯片的复位引脚		
	FVR0UT	AN	-	FVR 的输出引脚		
	OPOAN3	AN	-	OPO 通道 3 输入		
	COM16	AN	-	LCD COM 输出		
PC5	PC5	SMT	CMOS	GPIO	Y	Y
	PWM4	-	CMOS	PWM4 的输出		
	PWM10	-	CMOS	PWM10 的输出		
	BUOUT		CMOS	BUFF 的输出引脚		
	OP1AN2	AN	-	OP1 通道 2 输入		
	COM17	AN	-	LCD COM 输出		

注：使用外部晶振时，若需要同时使用 PWM 输出，使用其他 IO 对应的相同的 PWM 输出；

1.5 烧录引脚说明

AD008烧录引脚说明			
序号	烧录引脚	烧录信号	在板说明
1	VCC	VCC	VCC Pin 和 Vss Pin 在烧录时，电压会变化，因此必须由烧录器 VCC 直接连接到芯片 VCC。在烧录时，烧录器会断电再上电做检查，因此并联的电容建议不要超过470uF电容，电容越大，烧录时间越长。
2	VSS	GND	
3	PA2	DAT	CLK, DAT, 为通信线，不允许0.1uF以上的电容，不允许强上下拉电阻。
4	PA3	CLK	
5	PC4	CKOE	CKOE可以是任意脚位（不包括VCC VSS PA2 PA3）。校准芯片时会产生1KHZ方波，电容会影响波形，因此禁止电容并联，也禁止上下拉电阻。

注：PC4：烧录空片时要用来校准 FVR，此时必须连接。已做 FT 的芯片烧录时不用连接此脚。

PB4：CKOE 可以选择任意脚位（不含 VCC、VSS、PC4、PA2、PA3）。已做 FT 的芯片烧录时不用连接此脚。

1.6 极限参数

电源供应电压： $V_{SS}-0.3V \sim 6V$

输入电压： $V_{SS}-0.3V \sim V_{DD}+0.3V$

储存温度： $-50^{\circ}C \sim 125^{\circ}C$

工作温度： $-40^{\circ}C \sim 85^{\circ}C$

IOL 总电流：100mA

IOH 总电流：-100mA

总功耗：500mW

注：这里只强调额定工作功率，超过极限参数所规定的范围将对芯片造成损害，无法预期芯片在上述参数范围外的工作状态，若长期在标示范围外条件下工作，可能影响芯片可靠性。

2 中央处理器

2.1 指令集

AD008 具有一个支持 83 条内核指令的标准指令集。其中包含 8 条针对优化递归和软件堆栈代码的扩展指令。

2.2 程序存储

AD008 的 FLASH 为 4K*16 bit。

AD008 有两个中断向量。复位向量地址为 0000h,中断向量地址为 0008h 和 0018h。

2.2.1. 程序计数器

程序计数器 (Program Counter,PC), 指定要取出执行的指令地址。PC 内的地址为 14 位二进制数, 并且保存在 2 个独立的 8 位寄存器中。其中的低字节称为 PCL 寄存器, 该寄存器可读写。高字节, 即 PCH 寄存器, 存储 PC<14: 8>位, 不可直接读写。可以通过 PCLATH 寄存器更新 PCH 寄存器。

通过执行写 PCL 的操作, 可以将 PCLATH。类似的, 通过执行读 PCL 的操作, 可以将程序计数器的两个高字节传送到 PCLATH。

CALL, RCALL, GOTO 和程序转移指令直接写入程序计数器。对于这些指令, PCLATH 的内容将不会被传送到程序计数器。

2.2.2. 返回地址堆栈

用于存放返回地址的堆栈允许保存最多 8 个程序调用地址和中断向量。当执行 CALL 或 RCALL 指令或响应中断时, PC 值被压入堆栈。而执行 RETURN、RETLW 或 RETFIE 指令时, PC 值从堆栈弹出。PCLATH 不受 RETURN 或 CALL 指令的影响。

通过 21 位的 RAM 和 4 位的堆栈指针 (STKPTR) 来实现 16 级的堆栈操作。堆栈既不占用程序存储空间也不占用数据存储空间。堆栈指针可以读写, 并且通过栈顶的特殊文件寄存器可以读写栈顶地址。也可使用这些寄存器将数据压入堆栈, 或将数据从堆栈弹出。

执行 CALL 类型指令引起进栈操作: 堆栈指针首先加 1, 并且将 PC 的内容写入堆栈指针指向的地址单元 (PC 已经指向 CALL 的下一条指令)。执行 RETURN 类型指令时, 引起出栈操作: STKPTR 寄存器所指向的地址单元的内容被传送给 PC, 然后堆栈指针减 1。

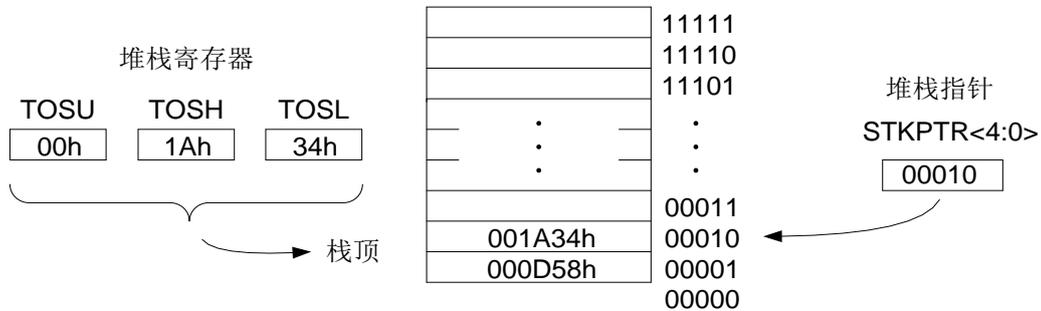
所有复位后, 堆栈指针被初始化 00000。堆栈指针值 00000 不指向任何 RAM 单元, 它只是一个复位值。状态表明堆栈是满、上溢还是下溢。

2.2.3. 栈顶访问

只有栈顶 (Top-of-Stack,TOS) 是可读写的。有 2 个寄存器 TOSH:TOSL 用于保存 STKPTR 寄存器所

指向的堆栈单元的内容。这可以让用户在必要时实现软件堆栈。在 CALL、RCALL 或中断后，软件可以通过读取 TOSH:TOSL 寄存器来读取进栈值。这些值可以被置入用户定义的软件堆栈。返回时，软件将这些值存回 TOSH:TOSL 并执行返回。

为防止对堆栈的意外操作，访问堆栈时用户必须禁止全局中断使能位。



2.2.3.1. 返回堆栈指针 (STKPTR)

STKPTR 寄存器包含堆栈指针值、STKFUL (堆栈满) 状态位和 STKUNF (堆栈下溢) 状态位。堆栈指针值可为 0 到 7 之间的整数。向堆栈压入值前，堆栈指针加 1；而从堆栈弹出值后，堆栈指针减 1。复位时，堆栈指针值为 0。用户可以读写堆栈指针的值。实时操作系统 (Real-Time Operating System, RTOS) 可以利用此特性对返回堆栈进行维护。

当向堆栈压入 PC 值 8 次 (且没有值从堆栈弹出) 后，STKFUL 位就会置 1。通过软件或 POR 使 STKFUL 位清零。

堆栈满时执行的操作由 STVREN (堆栈上溢复位使能) 配置位的状态决定。如果 STVREN 位已经置 1 (默认)，第 15 次进栈将把 (PC+2) 值压入堆栈，将 STKFUL 位置 1，并复位器件。STKFUL 位将保持置 1，而堆栈指针将被清零。如果 STVREN 位被清零，第 15 次进栈时 STKFUL 位会被置 1，堆栈指针则加 1 变为 7。任何其他进栈操作都不会覆盖第 15 次进栈的值，并且 STKPTR 将保持 7。当堆栈弹出次数足够卸空堆栈时，下一次出栈会向 PC 返回一个零值，并将 STKUNF 位置 1，而堆栈指针则保持为 0。STKUNF 位将保持置 1，直到被软件清零或发生 POR。

STKPTR 寄存器

地址:0XFFC

Bit	Name	Description	Attribute	Reset
7	STKFUL	堆栈满标志位 1: 堆栈满或上溢 0: 堆栈未满或未上溢	R/W	0
6	STKUNF	堆栈下溢标志位 1: 发生堆栈下溢 0: 未发生堆栈下溢	R/W	0
5	Reserved			
4:0	SP[4:0]	堆栈指针地址位	R/W	0

2.2.4. 快速寄存器堆栈

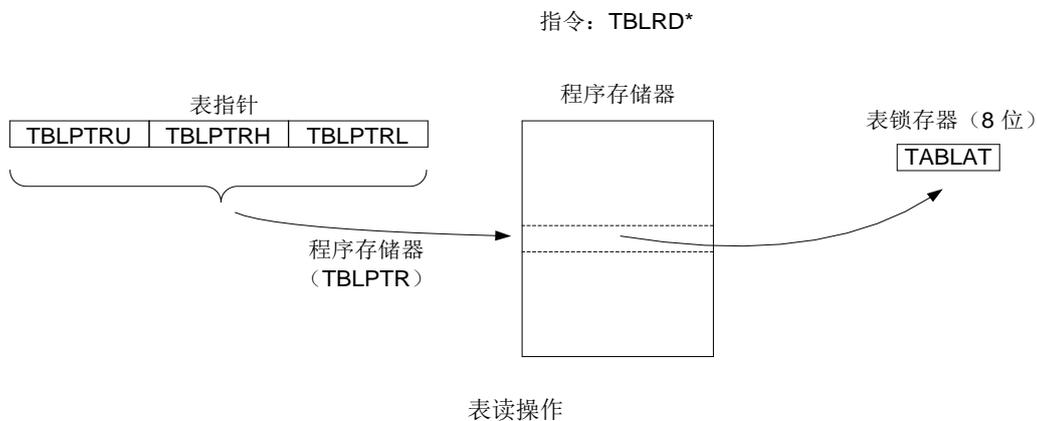
为 Status、WREG 和 BSR 寄存器提供的快速寄存器堆栈具有从中断“快速返回”的功能。每个寄存器堆栈的深度仅为 1 级，并且不可读写。当处理器转入中断向量处执行指令时，此堆栈装入对应寄存器的当前值。所有中断源都会将值压入堆栈寄存器。如果使用 RETFIE, FAST 指令从中断返回，这些寄存器中的值会被重新装回对应的寄存器。

如果同时使能了低优先级中断和高优先级中断，从低优先级中断返回时，无法可靠地使用堆栈寄存器。如果在为低优先级中断提供服务时，发生了高优先级中断，则低优先级中断存储在堆栈寄存器中的值将被覆盖。在这种情况下，用户必须在低优先级中断期间用软件保存关键寄存器的值。

如果未使用中断优先级，所有中断都可以使用快速寄存器堆栈从中断返回。如果没有使用中断，快速寄存器堆栈可以用于在子程序调用结束后恢复 Status、WREG 和 BSR 寄存器。要将快速寄存器堆栈用于子程序调用，必须执行 CALL lable,FAST 指令将 Status、WREG 和 BSR 寄存器的内容存入快速寄存器堆栈。在调用结束后执行 RETURN,FAST 指令，从快速寄存器堆栈中弹出并恢复这些寄存器的值。

2.2.5. 程序存储器

为了读取程序存储器，AD008 支持表读操作（TBLRD）。程序存储空间为 16 位宽，而数据 RAM 空间为 8 位宽。表读通过一个 8 位寄存器（TABLAT）在这两个存储空间之间移动数据。



表锁存器（Table Latch, TABLAT）是映射到 SFR 空间的一个 8 位寄存器。表锁存器用于在程序存储器和数据 RAM 之间传输数据时保存 8 位数据。

表指针（Table Pointer, TBLPTR）在程序存储器中寻址字节。TBLPTR 由 3 个 SFR 寄存器组成：表指针最高字节、表指针次高字节和表指针低字节（TBLPTRH:TBLPTRL）。这 3 个寄存器合起来组成一个 21 位宽的指针。

TBLRD 指令使用表指针寄存器 TBLPTR。利用表操作的四种方法之一，这些指令可以更新 TBLPTR。下表列出了这些操作。

示例	表指针操作
TBLRD*	不修改 TBLPTR

TBLRD*+	TBLPTR 在读后递增
TBLRD*-	TBLPTR 在读后递减
TBLRD+*	TBLPTR 在读前递增

使用 TBLRD 指令执行表指针操作

读取一个闪存程序存储器字:

```

MOV LW    CODE_ADDR_UPPER    ;Load TBLPR with the base
MOV WF    TBLPTRU             ;address of the word
MOV LW    CODE_ADDR_HIGH
MOV WF    TBLPTRH
MOV LW    CODE_ADDR_LOW
MOV WF    TBLPTRL
    
```

READ_WORD

```

TBLRD*+           ;read into TABLAT and increment
MOV F    TABLAT, W    ;get data
MOV WF   WORD_EVEN
TBLRD*+
MOV FW   TABLAT, W    ;get data
MOV F    WORD_ODD
    
```

2.3 数据存储器 RAM

AD008 的数据存储器是用静态 RAM 实现的。存储空间包含 256 个字节；

数据存储器由特殊功能寄存器（SFR）和通用寄存器（General Purpose Register,GPR）组成。SFR 用于单片机和外设功能模块的控制和状态显示，而 GPR 则用于在用户应用程序中存储数据和高速暂存操作。任何未使用单元的读取值均为 0。

数据存储器映射



SRAM 数据存储器映射图

2.4 数据寻址方式

AD008 支持 4 种寻址方式：

- 固有寻址
- 立即数寻址
- 直接寻址
- 间接寻址

2.4.1. 固有和立即数寻址

AD008 的很多控制指令根本不需要任何参数，执行这些指令要么对整个器件造成影响，要么仅针对一个寄存器进行操作。这种寻址模式就是固有寻址。例如 SLEEP、RESET 和 DAW 指令。

其他指令的工作方式与此类似但需要操作码中有直接的参数。由于需要一些立即数作为参数，这种寻址模式被称为立即数寻址。例如 ADDLW 和 MOVLW，它们分别将立即数移入 W 寄存器或从中移出。其他的立即数寻址指令，例如 CALL 和 GOTO，包括一个 15 位的程序存储器地址。

2.4.2. 直接寻址

直接寻址在操作码中指定操作的全部或部分源地址和/或目标地址。此选项由指令附带的参数指定。

2.4.3. 间接寻址

间接寻址允许用户访问数据存储器中的单元而不需要在指令给出一个固定的地址。这是通过使用文件选择寄存器（File Select Register, FSR）指向被读取或写入的单元实现的。由于 FSR 本身作为特殊功能寄存器位于 RAM 中，所以也可以在程序控制下直接对它们进行操作。这使得 FSR 对于在数据存储器中实现诸如表和数组等数据结构非常有用。

也可以使用间接指针操作（Indirect File Operand, INDF）进行间接寻址。这种操作允许自动递增、递减或偏移指针，从而自动控制指针的值。它通过循环提高代码执行效率。

2.4.3.1. FSR 寄存器和 INDF 操作数

间接寻址的核心是三组寄存器：FSR0、FSR1 和 FSR2。每组寄存器都含有一对 8 位寄存器，FSRnH 和 FSRnL。FSRnH 寄存器的高四位未使用，所以每对 FSR 只保存一个 12 位二进制数，从而可以线性寻址整个数据存储空间。因此，FSR 寄存器对被用作数据存储器的地址指针。

间接寻址是通过一组间接指针操作数（从 INDF0 到 INDF2）完成的。这些操作数可以被看作“虚拟”寄存器：它们被映射到 SFR 空间中而不是通过物理方式实现的。对特定的 INDF 寄存器执行读或写操作实际上访问的是相应的 FSR 寄存器对。例如，读 INDF1 就是读 FSR1H:FSR1L 指向的地址单元中的数据。使用 INDF 寄存器作为操作数的指令实际上使用的是相应的 FSR 的内容，该内容为指向目标地址的指针。INDF 操作数只是使用指针的一种较方便的方法。

由于间接寻址使用完整的 12 位地址，因此没有必要进行数据 RAM 分区。因此 BSR 的当前内容和快

速操作 RAM 位对于确定目标地址没有影响。

2.4.3.2. FSR 寄存器和 POSTINC、POSTDEC、PREINC 以及 PLUSW

除了 INDF 操作数之外，每对 FSR 寄存器还有四个额外的间接操作数。和 INDF 一样，它们也是不能直接读写的“虚拟”寄存器。访问这些寄存器其实就是访问相关的 FSR 寄存器对，也是在其存储的数据所指向的地址单元上进行特定的操作。

- POSTDEC: 访问 FSR 值，然后自动将它减 1
- POSTINC: 访问 FSR 值，然后自动将它加 1
- PREINC: 将 FSR 的值加 1，然后在操作中使用该值
- PLUSW: 将 W 寄存器中带符号的值（从-127 到 128）与 FSR 寄存器中带符号的值相加，并在操作中使用得到的新值

在应用中使用 FSR 寄存器中的值（不会更改此值）访问 INDF 寄存器。同样，访问 PLUSW 寄存器是将 W 寄存器中的值作为 FSR 值的偏移量，该操作不会改变这两个寄存器中的值。访问其他虚拟寄存器会更改 FSR 寄存器的值。

用 POSTDEC、POSTINC 和 PREINC 对 FSR 进行操作会影响整对寄存器，也就是 FSRnL 寄存器从 FFh 到 00h 溢出并向 FSRnH 寄存器进位。但这些操作的结果不会更改 Status 寄存器中的标志位（如 Z、N 和 OV 等）。

PLUSW 寄存器可以用于在数据存储空间实现变址寻址。通过控制 W 寄存器中的值，用户可以访问相对当前指针地址有固定偏移量的地址单元。在某些应用中，该功能可以被用于在程序存储器内部实现某些非常有用的程序控制结构，如软件堆栈。

2.4.3.3. 通过 FSR 对其他 FSR 进行操作

在某些特殊情况下，间接寻址操作以其他 FSR 或虚拟寄存器作为寻址目标。例如，使用 FSR 指向一个虚拟寄存器会导致操作不成功。假设如下特殊情况：FSR0H:FSR0L 保存的是 INDF1 的地址 FE7h。尝试使用 INDF0 作为操作数读取 INDF1 的值，将返回 00h。尝试使用 INDF0 作为操作数写入 INDF1，将会导致执行一条 NOP。

另一方面，使用虚拟寄存器对一对 FSR 寄存器进行写操作可能会产生与预期不同的结果。在这些情形下，会将写入一对 FSR 寄存器，但 FSR 中的值不会有任何递增或递减。因此，写入 INDF2 或 POSTDEC2 时会把同样的值写入 FSR2H:FSR2L。

由于 FSR 是在 SFR 空间中映射的物理寄存器，所以可以通过直接寻址对它们进行操作。用户在使用这些寄存器时应该特别小心，尤其是在代码使用间接寻址的情况。

同样，通常允许通过间接寻址对所有其他的 SFR 进行操作。用户在进行此类操作时应该特别小心，以免更改设置从而影响器件操作。

2.5 配置选项

配置字 0

位	名称	说明
3:0	LVD[3:0]	LVR 电压选择 0: 保留, 勿配置 1: 保留, 勿配置 2: 2.2V 3: 2.4V (为确保稳定运行, 推荐设定值为 2.4V 及以上) 4: 2.6V 5: 2.7V 6: 2.9V 7: 3.0V 8: 3.1V 9: 3.3V 10: 3.6V 11: 3.7V 12: 3.8V 13: 4.1V 14 : 4.2V 15 : 4.3V
4	LVREN	LVR 使能控制位 1: 使能 0: 禁止 (默认)
7:5	FINTOSC[2:0]	内部 RC 振荡器频率选择 111: 1:1 分频 110: 1:2 分频 101: 1:4 分频 100: 1:8 分频 011: 1:16 分频(默认)

配置字 1

位	名称	说明
0	STVEN	堆栈溢出复位 1: 允许使能 (默认) 0: 禁止使能
1	RSEL	端口上/下拉电阻选择(需要确认) 1: 上拉电阻: 30K 下拉电阻: 30K 0: 上拉电阻: 190K 下拉电阻: 300K(默认)
2	RESETE	外部复位使能 1: 使能外部复位功能 0: 屏蔽外部复位功能 (默认)
3	CBP	代码保护选择位 1: 代码加密 0: 代码不加密(默认)

4	CKSUM	校验使能位 1: 使能校验 0: 关闭校验(默认)
5	IDIS	模拟 IO 打开, 数据读取控制位。 1: 模拟 IO 打开, 能正常读取 IO 输入(默认); 0: 模拟 IO 打开, 读取 IO 数据始终为 0。 注: 模拟 IO 关闭, IDIS 位对读取 IO 输入无影响。
6	CRYHEN	外部晶体振荡器高速选择 1: 使能外部晶体高频振荡器 0: 禁止外部晶体高频振荡器(默认)
7	CRYLEN	外部晶体振荡器低速选择 1: 使能外部晶体低频振荡器 0: 禁止外部晶体低频振荡器(默认)
8	FLASH_CPO	0~1K*16 FLASH 保护位(CBP 为 1 时有效), 保护后不可擦写 1: 使能 FLASH 保护 0: 禁止 FLASH 保护(默认)
9	FLASH_CP1	1K*16~2K*16 FLASH 保护位(CBP 为 1 时有效), 保护后不可擦写 1: 使能 FLASH 保护 0: 禁止 FLASH 保护(默认)
10	FLASH_CP2	2K*16~3K*16 FLASH 保护位(CBP 为 1 时有效), 保护后不可擦写 1: 使能 FLASH 保护 0: 禁止 FLASH 保护(默认)
11	FLASH_CP3	3K*16~4K*16 FLASH 保护位(CBP 为 1 时有效), 保护后不可擦写 1: 使能 FLASH 保护 0: 禁止 FLASH 保护(默认)
12	DBG	DEBUG 模式使能位, 使能后将固定 PA2 和 PA3 为 DEBUG IO 1: 使能 0: 禁止(默认)

配置字 2

位	名称	说明
2:0	TWDT[2:0]	看门狗溢出时间及上电复位时间选择 111: TWDT (no Prescaler) = 8.5s 110: TWDT (no Prescaler) = 4.31s 101: TWDT (no Prescaler) = 1.92s 100: TWDT (no Prescaler) = 1.1s 011: TWDT (no Prescaler) = 653ms(默认) 010: TWDT (no Prescaler) = 391ms 001: TWDT (no Prescaler) = 260ms 000: TWDT (no Prescaler) = 128.4ms
3	WDTE	WDT 使能 1: 使能 WDT(默认)

		0: 关闭 WDT
4	FCPUS	指令周期选择 1: 1 个指令周期为 4 个机器周期（默认） 0: 1 个指令周期为 2 个机器周期
7:5	PSUT[2:0]	上电复位时间选择 111: PWRT= 20.4ms 110: PWRT= 11.9ms 101: PWRT= 309.6ms 100: PWRT = 79.2ms 011: PWRT = 2.57ms（默认） 010: PWRT = 1.55ms 001: PWRT = 1.43ms 000: PWRT = 1.55ms

配置字 3

位	名称	说明
0	ADCSP	ADC 采集速度选择位 1: 高速（默认） 0: 低速
1	IESO	双速启动模式设置位 1: 使能双速启动模式，仅在系统时钟为外部晶振模式才起作用 0: 禁止双速启动模式（默认）
2	FCMEN	系统时钟失效监测使能位 1: 使能系统时钟失效监测，仅在系统时钟为外部晶振模式才起作用 0: 禁止系统时钟失效监测（默认）
7~6	OSCM	内部芯片工作时钟选择 00: 选择内部低速时钟 32KHz 01: 选择外部高速时钟 16MHz 10: 选择内部低速时钟 1.024MHz 11: 选择内部高速时钟 16MHz（默认）

3 EEPROM 和 FLASH

3.1 EEPROM

注：AD008 暂未开放此功能，请勿配置使用；

数据 EEPROM 在正常工作期间（整个 VDD 范围）是可读写的。该存储器并不直接映射到文件寄存器空间。而是通过特殊功能寄存器来间接寻址。有以下 4 个 SFR 用于读写该存储器：

- EECON1
- EECON2（可写不可读）
- EEDATA
- EEADR

EEDATA 寄存器存放 8 位要读写的数据，而 EEADR 寄存器存放要被访问的 EEPROM 单元的地址。AD008 具有 256 字节的数据 EEPROM，地址范围从 00H 到 FFH。

EEPROM 数据存储器允许以字节为单位进行读写。字节写操作将自动擦除目标存储单元并写入新数据（即先擦后写）EEPROM 数据存储器可以反复擦写很多次。写入时间由片上定时器控制。写入时间会因为电压、温度及芯片的不同而发生变化。写 EEPROM 时需要在 55°C 以下进行。

如果数据存储器有代码保护，CPU 仍可继续对数据 EEPROM 存储器进行读写操作。器件编程器无法再访问数据 EEPROM 的数据，EEPROM 单元读为零。

EEDATA 寄存器

地址：0XF5C

Bit	Name	Description	Attribute	Reset
7:0	EEDATA[7:0]	要从数据 EEPROM 读取或者向数据 EEPROM 写入的字节值	R/W	0

EEADR 寄存器

地址：0XF5B

Bit	Name	Description	Attribute	Reset
7:0	EEADR[7:0]	指定在 256 个字节中进行读写操作的 8 位地址字节	R/W	0

3.1.1. EECON1 和 EECON2 寄存器

EECON1 是控制寄存器，它的低 4 位是物理实现的，高 4 位未实现，读为 0。

控制位 RD 和 WR 分别启动读操作和写操作。用软件只能将这些位置 1 而无法清零。当读操作或写操作完成后，由硬件将其清零。由于不能用软件将 WR 位清零，可有效防止写操作意外或提前终止。

将 WREN 位置 1 将允许一次写操作。上电时将清零 WREN 位。当正常的写操作被 MCLR 复位或 WDT 复位中断时，WRERR 将被置 1。在这些情况下，用户可以在复位后检查 WRERR 位，将其清零并重写相应的单元。数据和地址将被清空。因此，需要重新对 EEDATA 和 EEADR 寄存器进行初始化。

写操作结束时，PIFB1 寄存器的中断标志位 EEIF 将被置 1。此标志位必须用软件清零。

EECON2 不是物理寄存器。读 EECON2 得到的是不确定值。EECON2 寄存器仅在数据 EEPROM 写入过程中使用。

注：当对数据 EEPROM 进行写操作（WR=1）时，不应修改 EECON1、EEDATA 和 EEADRH 和 EEADRL 寄存器。

EECON1 寄存器

地址：0XF5A

Bit	Name	Description	Attribute	Reset
7:4	Reserved			
3	WRERR	EEPROM 错误标志位 1: 写操作提前终止(正常工作期间发生的任何 RST 复位、WDT 复位和欠压检测) 0: 写操作完成	R/W	0
2	WREN	EEPROM 写使能位 1: 允许写周期 0: 禁止写周期	R/W	0
1	WR	写控制位 1: 启动写周期(一旦完成写操作，硬件会将该位清零。软件只能将 WR 位置 1 而不清零) 0: 写入数据 EEPROM 的周期完成	R/W	0
0	RD	读控制位 1: 启动 EEPROM 读操作(读取需要一个周期，RD 位由硬件清零。软件只能将 RD 位置 1 而不清零) 0: 不启动 EEPROM 读操作	R/W	0

EECON2 寄存器

地址：0XF59

Bit	Name	Description	Attribute	Reset
7:0	EECON2	写入数据 0X18 和 0XF2，允许 EEPROM 写操作	W	0

3.1.2. 读数据 EEPROM 存储器

要读取数据存储单元，用户必须将地址写入 EEADRH 和 EEADRL 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，EEDATA 寄存器就有数据了。因此该数据可由下一条指令读取。EEDATA 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）。

例：读数据 EEPROM

EEADR = data_addr;	写 EEPROM 数据地址
EECON1 = 0X01;	使能读操作

3.1.3. 写数据 EEPROM 存储器

要写入 EEPROM 数据存储单元，用户必须先把该单元的地址写 EEADR 寄存器并且把数据写入 EEDATA 寄存器。然后用户必须按特定顺序开始写入每个字节。

例：写数据 EEPROM

EEADR = data_addr;	写 EEPROM 数据地址
EEDAT = data;	写 EEPROM 数据
EEIF = 0;	清中断标志
EECON1 = 0X04;	使能 WREN
EECON2 = 0X18;	
EECON2 = 0XF2;	
EECON1 = 0X06;	使能 WR
while (EEIF==0)	等待写成功中断标志

如果没有完全按照以上顺序逐字节写入，写操作将不会开始。在这个代码段执行过程中，固定写法的四行语句必须禁止中断，否则可能带来不可预测结果。在必需序列执行过程中会进行周期计数。当计数值与执行必需序列所需的周期不等时，数据无法写入EEPROM。

此外，必须将 EECON1 中的WREN 位置1 以使能写操作。这种机制可防止由于代码执行错误（异常）（即程序失控）导致误写数据 EEPROM。除非更新 EEPROM，用户应始终保持 WREN 位清零。

3.1.4. 写校验

根据应用情况，将写入数据 EEPROM 的实际值与要写入的值进行核对是一种很好的编程习惯。

EEADR = data_addr;	写 EEPROM 数据地址
EECON1 = 0X01;	读 EEPROM 数据
if (EEDAT != data)	核对写入的数据
check error;	（伪代码）
else	
check ok	（伪代码）

3.1.5. 避免误写操作

有些情况下，用户可能不希望向数据 EEPROM 存储器写入数据。EEPROM 存储器有各种机制以防对 EEPROM 误写。上电时，WREN 被清零。而且，上电延时定时器也可以防止误写 EEPROM。

写操作启动序列和 WREN 位可以共同预防在以下情况下发生误写：

- 欠压
- 电源毛刺
- 软件故障

3.2 FLASH

FLASH 可以自编程的地址空间从 0000H 到 1FFFH，IAPADDR[12:0]（由 IAPADDRH 高 5 位和 IAPADDRL 组成）对应 FLASH 的地址。IAPDATH[15:8]和 IAPDATL[7:0]组成 16 位数据。写 FLASH 时最高温度不能超过 125°C。

FLASH 写入数据操作步骤：

- (1) 写入电压为芯片工作电压
- (2) 对 IAPTRIG 连着写入 0xad,0x18,0xf2，进入到 IAP mode。**注意：如果往 IAPTRIG 写入其他任何值，则进入 IAPLOCK 状态。进行重新上电来解除锁定，解除后需要重新开始进行 IAP 写操作。**
- (3) 设置 IAPADDRH（高位地址）和 IAPADDRL（低位地址），设置数据要写入 FLASH 对应 0000H-1FFFH 地址内的某个地址。
- (4) 设置 IAPDATH（高位数据）和 IAPDATL（低位数据），即把需要写入数据寄存器的数据暂放在 IAPDAT 寄存器中。
- (5) CPU 将 IAPDAT[15: 0]数据写入 IAPADDRH 和 IAPADDRL 对应的 FLASH 地址中。
- (6) 如果要继续写，则回到步骤（3）开始。如果要退出写入操作，则到步骤（7）。
- (7) 对 IAPTRIG 写入 0，便可以退出写操作。

3.2.1. IAPTRIG 寄存器

地址：0XF6C

Bit	Name	Description	Attribute	Reset
7:0	IAPTRIG[7:0]	IAP 的触发寄存器	R/W	0x00

3.2.2. IAPCTRL 寄存器

地址：0XF6B

Bit	Name	Description	Attribute	Reset
7:4	Reserve			
3	ERR	FLASH 的写错误标志位 1: 写操作提前终止(正常工作期间发生的任何 RST 复位、WDT 复位和欠压检测) 0: 写操作完成	R/W	0
2	LOCK	IAP 锁定状态位 1: 锁定	R/W	0

		0: 未锁定		
1	ER	FLASH 页擦除 (Fsys 为 16Mhz 时需要耗时 2.1ms, 无需手动延时) 1: 使能擦除 0: 未擦除	R/W	0
0	PG	写入 1: 写入 (Fsys 为 16Mhz 需要耗时 21us, 无需手动延时) 0: 未写入	R/W	0

3.2.3. IAPADDRH 寄存器

地址: 0XF6A

Bit	Name	Description	Attribute	Reset
7:0	IAPADDRH[15:8]	IAP 地址高八位	R/W	0

3.2.4. IAPADDRL 寄存器

地址: 0XF69

Bit	Name	Description	Attribute	Reset
7:0	IAPADDRL[7:0]	IAP 地址低八位	R/W	0

3.2.5. IAPDATH 寄存器

地址: 0XF68

Bit	Name	Description	Attribute	Reset
7:0	IAPDATH[15:8]	IAP 数据高八位	R/W	0

3.2.6. IAPDATL 寄存器

地址: 0XF67

Bit	Name	Description	Attribute	Reset
7:0	IAPDATL[7:0]	IAP 数据低八位	R/W	0

3.2.7. IAPWAIT 寄存器

地址: 0XF66

Bit	Name	Description	Attribute	Reset
7:0	IAPWAIT[7:0]	IAP 延时位 操作 FLASH 时, 该寄存器需设置为 0x40	R/W	0

		操作 EEPROM 时，该寄存器需设置为 0x04		
--	--	---------------------------	--	--

3.2.8. SIGCON 寄存器

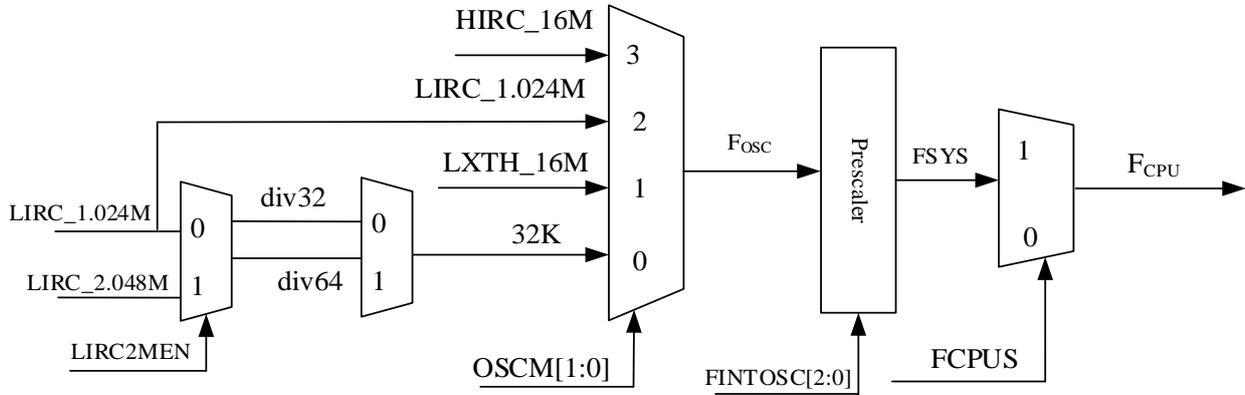
地址：0XF61

Bit	Name	Description	Attribute	Reset
7:3	Reserved			
2	ICE_TMREN	SWD TIMER 计数使能位（用于仿真器断点工作时控制 TIMER 是否计数） 1：允许使能 0：禁止使能	R/W	0
1	UART_MUX	UART 的 IO 复用 1：PB0(TX 和 RX_TX) ;PB1(RX) 0：PA1(TX 和 RX_TX) ;PA0(RX)	R/W	0
0	SIG	读取配置字中的校准信息 1：读取配置字中的校准信息 0：禁止读取配置字中的校准信息	R/W	0

注：当对 FLASH 进行 IAP 操作时，不应使能配置字中的 CKSUM 功能。

4 系统时钟源

4.1 系统时钟相关寄存器



4.1.1. OSCCON 寄存器

地址: 0XF64

Bit	Name	Description	Attribute	Reset
7	Reserved			
6:4	FINTOSC	内部振荡器频率选择位 111: 1:1 分频 110: 1:2 分频 101: 1:4 分频 100: 1:8 分频 011: 1:16 分频	R/W	011
3	LIRC2MEN	低速时钟的倍频使能 1: 使能, LIRC 输出为 2.048M 0: 未使能, LIRC 输出 1.024M	R/W	0
2:0	Reserved			

4.1.2. CLKCFG1 寄存器

地址: 0XF63

Bit	Name	Description	Attribute	Reset
7	OPCLKEN	OP 的时钟使能, 使能后 OP 才可以工作 1: 允许使能 0: 禁止使能	R/W	1
6	UARTCLKEN	UART 的时钟使能, 使能后 UART 才可以工作 1: 允许使能 0: 禁止使能	R/W	0

5	CKOE1	PB2 输出当前系统时钟 1: 使能系统时钟输出 0: 禁止系统时钟输出	R/W	0
4	CKOE0	PC3 输出当前系统时钟 1: 使能系统时钟输出 0: 禁止系统时钟输出	R/W	0
3:2	OSCF[1:0]	工作时钟状态位 00: 表示内部低速时钟工作 (32KHz) 01: 表示外部高速时钟工作 (16MHz) 10: 表示内部低速时钟工作 (1.024MHz) 11: 表示内部高速时钟工作 (16MHz)	R	11
1:0	OSCM[1:0]	系统时钟选择 00: 选择内部低速时钟 32KHz 01: 选择外部高速时钟 16MHz 10: 选择内部低速时钟 1.024MHz 11: 选择内部高速时钟 16MHz	R/W	11

4.1.3. CLKCFG2 寄存器

地址: 0XF62

Bit	Name	Description	Attribute	Reset
7	CMPCLKEN	CMP 的时钟使能, 使能后 CMP 才可以工作 1: 允许使能 0: 禁止使能	R/W	1
6	DACCLKEN	DAC 的时钟使能, 使能后 DAC 才可以工作 1: 允许使能 0: 禁止使能	R/W	1
5	ADCCLKEN	ADC 的时钟使能, 使能后 ADC 才可以工作 1: 允许使能 0: 禁止使能	R/W	1
4	CCPCLKEN	CCP 的时钟使能, 使能后 CCP 才可以工作 1: 允许使能 0: 禁止使能	R/W	1
3	T3CLKEN	TIMER3 的时钟使能, 使能后 TIMER3 才可以工作 1: 允许使能 0: 禁止使能	R/W	1
2	T2CLKEN	TIMER2 的时钟使能, 使能后 TIMER2 才可以工作 1: 允许使能 0: 禁止使能	R/W	1
1	T1CLKEN	TIMER1 的时钟使能, 使能后 TIMER1 才可以工作 1: 允许使能 0: 禁止使能	R/W	1

0	TOCLKEN	TIMERO 的时钟使能，使能后 TIMERO 才可以工作 1: 允许使能 0: 禁止使能	R/W	1
---	---------	---	-----	---

4.2 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 FINTOSC 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。

4.3 双速启动模式配置

通过以下设定来配置双速启动模式：

- 配置字选项3中的位IESO = 1，使能双速启动模式；
- 配置字选项3中的位OSCM = 2'b01，选择芯片工作时钟为外部高速时钟；
- 配置字选项1中的位CRYHEN = 1，使能外部晶体高频振荡器。

在下列操作之后，进入双速启动模式：

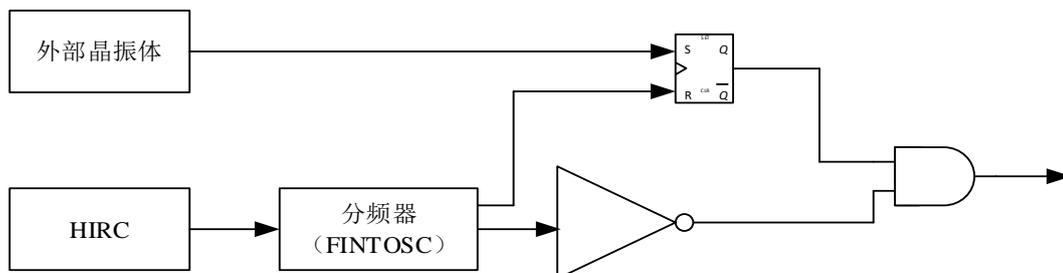
- 从休眠状态唤醒。

4.4 双速启动顺序

1. 从休眠中唤醒；
2. 系统时钟切换到外部时钟源。

4.5 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 能在振荡器起振一段时间后的任一时刻检测振荡器故障。FSCM 通过将配置字选项 3 中的位 FCMEN = 1，使能系统时钟失效监测。



FSCM 原理图

4.5.1. 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 PIR3 寄存器的 OSFIF 标志位置 1。如果在 PIE3 寄存器 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时钟。

FSCM 所选的内部时钟源由配置字选项 3 中的 OSCM 位或者 CLKCFG1 寄存器 OSCM 决定，内部振荡器应该在故障发生前配置。

4.5.2. 故障检测完成

FSCM 设计为能在振荡器起振一段时间后任一时刻检测振荡器故障。一旦检测到故障完成，FSCM 就处于激活状态。

注：经过一段时间后，读 CLKCFG1 寄存器的 OSCF 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。

5 复位和电源电压检测

AD008 单片机能通过以下方式复位：

- 上电复位(POR)
- 掉电复位(Brown-out Reset BOR)
- RSTB 管脚复位
- 看门狗 WDT 溢出复位

一些寄存器在一些复位条件下没有影响，在上电和其他一些复位情况下它们的状态是未知的。大多数寄存器先进入到复位状态然后再上电复位，例如 RSTB 管脚复位，看门狗 WDT 溢出复位。通过 Vdd 上升信号检测告知芯片是否加上上电复位脉冲信号。要使用这个特点，用户需要把 RSTB 管脚连接到 Vdd。掉电复位作为一种典型应用主要用在 AC 或重载交换的应用上。RSTB 或 WDT 睡眠唤醒也导致芯片复位。根据不同的状态设置对 TO(RCON< 3 >)和 PD 位(RCON< 4 >)置 1 或清零。

5.1 上电复位计数器 (PWRT)

上电复位计数器提供一个 1.43/2.57/11.9/20.4/79.2/309.6ms 延迟时间 (该延迟时间由 配置字 PSUT<7:5> 设置) (基于不同的振荡源和复位条件) 在 Power-on Reset (POR), Brown-out Reset (BOR), RSTB Reset 或看门狗溢出复位。只要 PWRT 在运行，设备就一直保持的复位状态。Vdd、温度和其他变化会影响其控制的设备延迟时间。

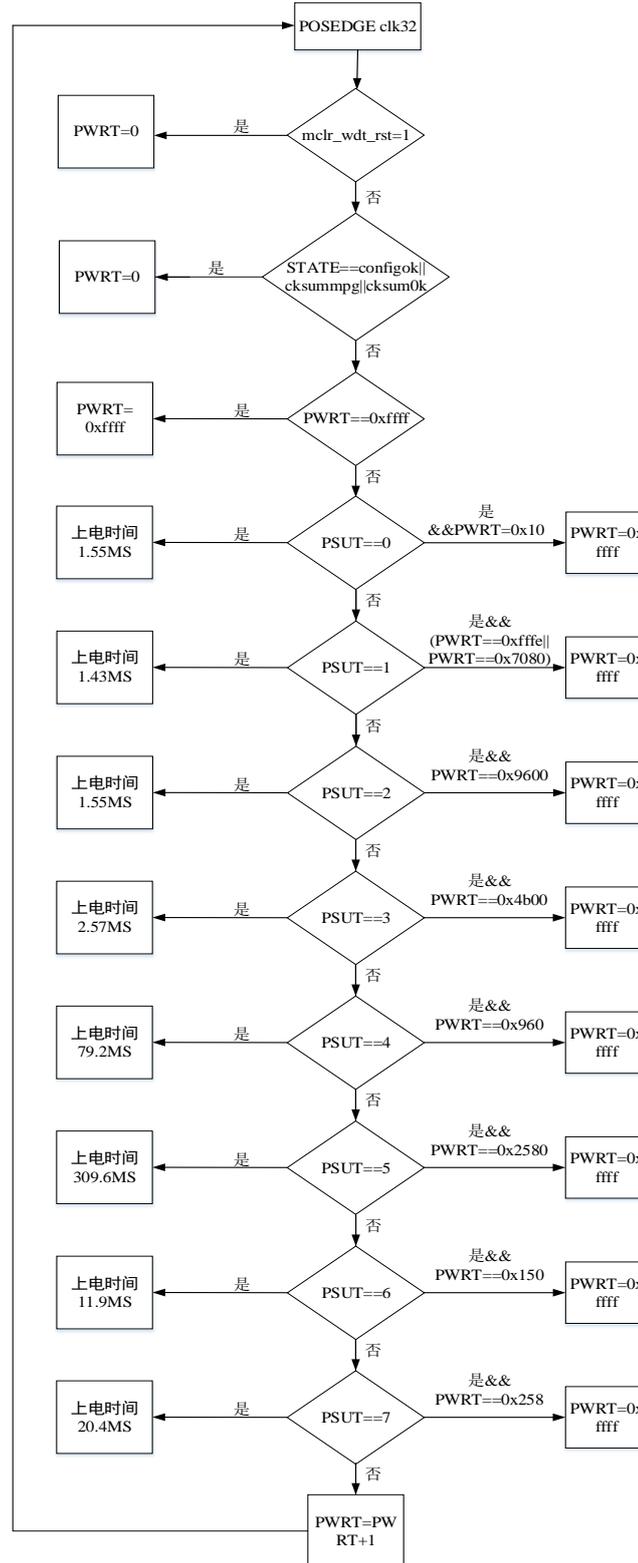
模拟模块在 POR 上电初始工作状态表：

模块	POR 上电
BOR	√
ADC	
LVR	
CMPO	
CMP1	
DACO	
DAC1	
FVR	
LCD	
OP0	
OP1	
HIRC16M	√
OSC1. 024M	√

OSC32K	√
LDO	√
BGR	√
BUFFER	
NTC	
BIAS	

上电复位时间选择流程图：

上电之前先是产生 HIRC16M,依次产生 OSC1.024M,OSC32K。



上电复位时间流程图

上电中和上电完成后，内部低速时钟 32KHz、内部低速时钟 1.024MHz、内部高速 16MHz 均正常工作，并且在配置字配置之前，系统时钟(SYSCLK)是按内部低速时钟 1MHz 工作。当配置字或 CLKCFG1[1:0] 完成系统时钟选择，此时 SYSCLK 按照选择时钟工作。

5.2 振荡启动计数器 (OST)

在 HF 或 LF 或 IRC_RTC 振荡模式下在 PWRT 延迟 (1.43/2.57/11.9/20.4/79.2/309.6ms) 之后振荡启动计数器会再提供一个 64 个 clock 的延迟。这种延迟晶体谐振器能提供稳定的振荡源, 这段时间内只要 OST 在工作, 设备就一直保持的复位状态。

5.3 复位顺序

AD008 复位时序如下:

- 复位锁存器置 1, PWRT & OST 清零。
- 当内部的 POR, BOR, RSTB 复位或 WDT 溢出复位脉冲加载完成后, PWRT 开始计数。
- PWRT 溢出以后, OST 开始计数延迟。
- OST 延迟完成以后, 复位锁存器清零最后芯片得到一个复位信号。

在高频或低频振荡模式机器复位延迟时间为 1.43/2.57/11.9/20.4/79.2/309.6ms 加上 64 个振荡周期, 在 IRC/ERIC, ERC 振荡模式单片机会在 Power-on Reset (POR), Brown-out Reset (BOR), 或 RSTB 复位以后在延迟 640us, 看门狗溢出复位后再延迟 1.43/2.57/11.9/20.4/79.2/309.6ms 的时间。

图: 复位电路结构图

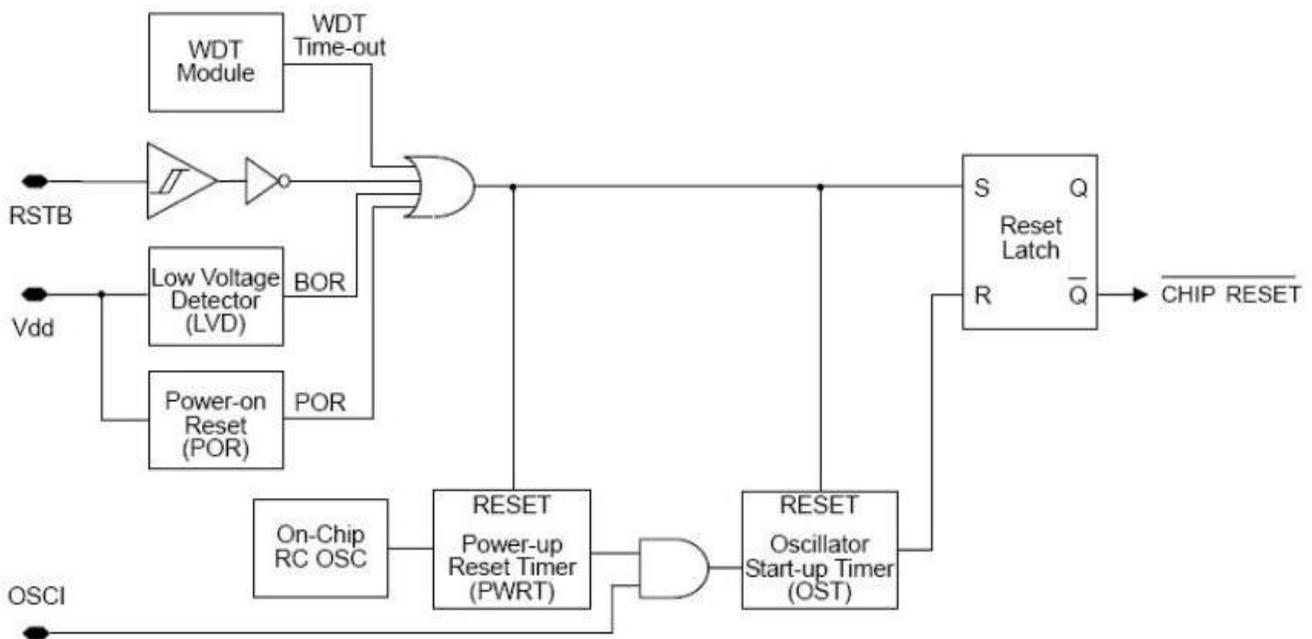


表: RST/ TO / PD 复位和唤醒后的状态

RST	/TO	/PD	复位方式
0	1	1	Power-on Reset
0	1	1	Brown-out reset

0	u	u	RSTB Reset during normal operation
0	1	0	RSTB Reset during SLEEP
0	0	1	WDT Reset during normal operation
0	0	0	WDT Wake-up during SLEEP
1	1	0	Wake-up on pin change during SLEEP

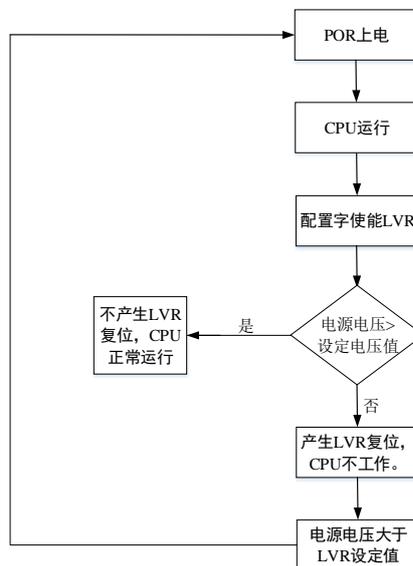
Legend: u =不变

表: /TO /PD 状态位影响事件

事件	/TO	/PD
Power-on	1	1
WDT Time-Out	0	u
SLEEP instruction	1	0
CLRWDT instruction	1	1

Legend: u =不变

LVR 复位工作流程:



LVR 复位工作流程图

不管在 LVR 复位期间, 还是 LVR 不复位时, HIRC16M/OSC1.024M/OSC32K 均能正常工作。

5.4 LVD 检测

LVD 可通过 PCON[3:0]配置电压阈值。电压检测电路有一定的回滞特性,通常回滞电压为 0.05V 左右。例如,如果选择了 3.6V 的 LVD 电压,则当电源电压下降到约 3.6V 复位有效,而电压需要上升到约 3.65V 时 LVD 复位才会解除。

5.4.1. PCON(寄存器)

地址: 0XFF1

Bit	Name	Description	Attribute	Reset
7	Reserved			
6	PBOST	LVDM=11 时, PBO 输出值	R	0
5:4	LVDM[1:0]	电压比较中断 00: 禁止电压比较器 01: VCC 低于阈值电压产生中断 10: VCC 高于阈值电压产生中断 11: VCC 高于阈值电压产生中断, 且强制 PBO 输出为 PBOST 值	R/W	00
3:0	LVD[3:0]	LVD3~0 VCC 电压阈值选择 0: 保留, 勿配置 1: 保留, 勿配置 2: 2.2V 3: 2.4V 4: 2.6V 5: 2.7V 6: 2.9V 7: 3.0V 8: 3.1V 9: 3.3V 10: 3.6V 11: 3.7V 12: 3.8V 13: 4.1V 14 : 4.2V 15 : 4.3V	R/W	0000

注: PCON 寄存器中的 LVD 电压值的选择用于产生中断, 配置字中的 LVR 电压选择用于产生复位, 两者相互独立, 复位的优先级高于中断。

5.4.2. RCON(寄存器)

地址: 0XFD2

Bit	Name	Description	Attribute	Reset
7	IPEN	中断高低优先级使能位 1: 允许中断高低优先级 0: 只允许高优先级中断	R/W	1
6	Reserved			
5	LVDST	LVR 比较器输出值 1: 当 VCC 高于 LVD 设定值, 输出为 1 0: 当 VCC 低于 LVD 设定值, 输出为 0 注: 与 LVD 模块相关	R/W	0
4	RI	RESET 指令标志位 1: 未执行 RESET 指令 (只能由固件置 1) 0: 已执行 RESET 指令, 导致器件复位 (必须在欠压复位发生之后用软件置 1)	R/W	1
3	TO	看门狗超时标志位 1: 通过上电、CLRWDT 指令或 SLEEP 指令置 1 0: 发生了 WDT 超时	R/W	1
2	PD	掉电检测标志位 1: 通过上电或 CLRWDT 指令置 1 0: 通过执行 SLEEP 指令置 0	R/W	1
1	POR	上电复位状态位 1: 未发生上电复位 (只能用固件置 1) 0: 已发生上电复位 (必须在发生上电复位后由软件置 1)	R/W	0
0	BOR	欠压复位状态位 1: 未发生欠压 0: 已发生欠压复位 (必须在欠压复位后由软件置 1)	R/W	0

6 I/O 端口

PortA、PortB、PortC 为双向三态 I/O 口。所有的 I/O 的输入/输出方式由 I/O 控制寄存器(TRISA、TRISB,TRISC)设置。PA、PB 和 PC 有相应的上拉控制位(上拉寄存器)来设置使能内部上拉, 如果设置为输出模式, 内部上拉功能会自动关闭。PA、PB 和 PC 有相应的下拉控制位(下拉寄存器)来设置使能内部下拉。如果设置为输出模式, 内部下拉功能不会自动关闭, 需要自行关闭。PA、PB 和 PC 有相应的开漏控制位(开漏寄存器)来设置使能开漏来设置输出为开漏输出。

6.1 IO 工作模式

6.1.1. PORTA (Port 寄存器)

地址: 0XF80

Bit	Name	Description	Attribute	Reset
7:6	Reserved			
5:0	PORTA[5:0]	输入数据 (读), 输出数据 (写)	R/W	0x00

6.1.2. PORTB (Port 寄存器)

地址: 0XF81

Bit	Name	Description	Attribute	Reset
7:6	Reserved			
5:0	PORTB[5:0]	输入数据 (读), 输出数据 (写)	R/W	0x00

6.1.3. PORTC (Port 寄存器)

地址: 0XF82

Bit	Name	Description	Attribute	Reset
7:6	Reserved			
5:0	PORTC[5:0]	输入数据 (读), 输出数据 (写)	R/W	0x00

6.1.4. TRISA (I/O 口方向控制寄存器)

地址: 0XF90

Bit	Name	Description	Attribute	Reset
7:6	Reserved			
5:0	TRISA[5:0]	1:IO 的输入模式	R/W	0x3F

		0: IO 的输出模式		
--	--	-------------	--	--

6.1.5. TRISB (I/O 口方向控制寄存器)

地址: 0XF91

Bit	Name	Description	Attribute	Reset
7:6	Reserved			
5:0	TRISB[5:0]	1: IO 的输入模式 0: IO 的输出模式	R/W	0x3F

6.1.6. TRISC (I/O 口方向控制寄存器)

地址: 0XF92

Bit	Name	Description	Attribute	Reset
7:6	Reserved			
5:0	TRISC[5:0]	1: IO 的输入模式 0: IO 的输出模式	R/W	0x3F

6.1.7. PINA (数据锁存寄存器)

地址: 0XF88

Bit	Name	Description	Attribute	Reset
7:6	Reserved			
5:0	PINA[5:0]	读: 读取当前 IO 状态; 写: 写入 PORTA 寄存器;	R/W	0x00

6.1.8. PINB (数据锁存寄存器)

地址: 0XF89

Bit	Name	Description	Attribute	Reset
7:6	Reserved			
5:0	PINB[5:0]	读: 读取当前 IO 状态; 写: 写入 PORTB 寄存器;	R/W	0x00

6.1.9. PINC (数据锁存寄存器)

地址: 0XF8A

Bit	Name	Description	Attribute	Reset
7:6	Reserved			

5:0	PINC[5:0]	读：读取当前 IO 状态； 写：写入 PORTC 寄存器；	R/W	0x00
-----	-----------	----------------------------------	-----	------

6.2 下拉电阻开漏

6.2.1. PAPD (I/O 下拉控制寄存器)

地址：0XF8C

Bit	Name	Description	Attribute	Reset
7:6	Reserved			
5:0	PAnPD[5:0]	1: 关闭内部下拉 0: 使能内部下拉	R/W	0x3F

6.2.2. PAPU (I/O 上拉控制寄存器)

地址：0XF84

Bit	Name	Description	Attribute	Reset
7:6	Reserved			
5:0	PAnPU[5:0]	1: 关闭内部上拉 0: 使能内部上拉	R/W	0x3F

6.2.3. PAOD (I/O 开漏控制寄存器)

地址：0XF94

Bit	Name	Description	Attribute	Reset
7:6	Reserved			
5:0	PAnOD[5:0]	1: 使能内部开漏 0: 关闭内部开漏	R/W	0X00

6.2.4. PBPD (I/O 下拉控制寄存器)

地址：0XF8D

Bit	Name	Description	Attribute	Reset
5:0	PBnPD[5:0]	1: 关闭内部下拉 0: 使能内部下拉	R/W	0X3F

6.2.5. PBPU (I/O 上拉控制寄存器)

地址：0XF85

Bit	Name	Description	Attribute	Reset
7:6	Reserved			

5:0	PBnPU[5:0]	1:关闭内部上拉 0:使能内部上拉	R/W	0x3F
-----	------------	------------------------	-----	------

6.2.6. PBOD (I/O 开漏控制寄存器)

地址: 0XF95

Bit	Name	Description	Attribute	Reset
7:6	Reserved			
5:0	PBnOD[5:0]	1: 使能内部开漏 0: 关闭内部开漏	R/W	0x00

6.2.7. PCPD (I/O 下拉控制寄存器)

地址: 0XF8E

Bit	Name	Description	Attribute	Reset
7:6	Reserved			
5:0	PCnPD[5:0]	1: 关闭内部下拉 0: 使能内部下拉	R/W	0X3F

6.2.8. PCPU (I/O 上拉控制寄存器)

地址: 0XF86

Bit	Name	Description	Attribute	Reset
7:6	Reserved			
5:0	PCnPU[5:0]	1:关闭内部上拉 0:使能内部上拉	R/W	0X3F

6.2.9. PCOD (I/O 开漏控制寄存器)

地址: 0XF96

Bit	Name	Description	Attribute	Reset
7:6	Reserved			7
5:0	PcnOD[5:0]	1: 使能内部开漏 0: 关闭内部开漏	R/W	0X00

6.2.10. PAINTMASK (Port A 端口电平变化中断掩膜位)

地址: 0XF60

Bit	Name	Description	Attribute	Reset
7:6	Reserved			

5:0	PINTMASKA[5:0]	1: 使能 PortA 端口电平变化中断允许位 0: 禁止 PortA 端口电平变化中断允许位	R/W	0x00
-----	----------------	--	-----	------

6.2.11. PBINTMASK (Port B 端口电平变化中断掩膜位)

地址: 0XF5F

Bit	Name	Description	Attribute	Reset
7:6	Reserved			
5:0	PINTMASKB[5:0]	1: 使能 PortB 端口电平变化中断允许位 0: 禁止 PortB 端口电平变化中断允许位	R/W	0x00

6.2.12. PCINTMASK (Port C 端口电平变化中断掩膜位)

地址: 0XF5E

Bit	Name	Description	Attribute	Reset
7:6	Reserved			
5:0	PINTMASKC[5:0]	1: 使能 PortC 端口电平变化中断允许位 0: 禁止 PortC 端口电平变化中断允许位	R/W	0x00

6.3 模拟 IO 寄存器

6.3.1. ANASEL0 (IO 的模拟通道寄存器)

地址: 0XF54

Bit	Name	Description	Attribute	Reset
7:6	Reserved			
5:0	PAnAEN[5:0]	1: PA 口打开模拟通道 0: PA 口关闭模拟通道	R/W	0X00

6.3.2. ANASEL1 (IO 的模拟通道寄存器)

地址: 0XF53

Bit	Name	Description	Attribute	Reset
7:6	Reserved			
5:0	PBnAEN[5:0]	1: PB 口打开模拟通道 0: PB 口关闭模拟通道	R/W	0X00

6.3.3. ANASEL2 (IO 的模拟通道寄存器)

地址: 0XF52

Bit	Name	Description	Attribute	Reset
7:6	Reserved			
5:0	PCnAEN[5:0]	1: PC 口打开模拟通道 0: PC 口关闭模拟通道	R/W	0X00

6.3.4. SMTVA (PA 口施密特寄存器)

地址: 0XF98

Bit	Name	Description	Attribute	Reset
7:6	Reserved			
5:0	SMTVA[5:0]	施密特寄存器 1:0.3VDD/0.7VDD (IO 低电平变为高电平时为 0.7VDD, 高电平变为低电平时为 0.3VDD) 0:0.2VDD/0.4VDD (IO 低电平变为高电平时为 0.4VDD, 高电平变为低电平时为 0.2VDD)	R/W	0X3F

6.3.5. SMTVB (PB 口施密特寄存器)

地址: 0XF99

Bit	Name	Description	Attribute	Reset
7:6	Reserved			
5:0	SMTVB[5:0]	施密特寄存器 1:0.3VDD/0.7VDD (IO 低电平变为高电平时为 0.7VDD, 高电平变为低电平时为 0.3VDD) 0:0.2VDD/0.4VDD (IO 低电平变为高电平时为 0.4VDD, 高电平变为低电平时为 0.2VDD)	R/W	0X3F

6.3.6. SMTVC (PC 口施密特寄存器)

地址: 0XF9A

Bit	Name	Description	Attribute	Reset
7:6	Reserved			
5:0	SMTVC[5:0]	施密特寄存器 1:0.3VDD/0.7VDD (IO 低电平变为高电平时为 0.7VDD, 高电平变为低电平时为 0.3VDD) 0:0.2VDD/0.4VDD (IO 低电平变为高电平时为 0.4VDD, 高电平变为低电平时为 0.2VDD)	R/W	0X3F

6.3.7. CURCON (寄存器读写)

地址: 0XF55

Bit	Name	Description	Attribute	Reset
7:6	Reserved			
5:3	PB2CUR	PB2 电流驱动 (VDD=5V 单位:mA) VOH VOL 000: 13 16 001: 24 31 010: 13 16 011: 35 45 100: 13 16 101: 25 30 110: 13 16 111: 45 55	R/W	000
2	CURC	供电为 5V 时, PC 口的电流驱动 1: 允许灌电流中大电流驱动 0: 允许灌电流中小电流驱动	R/W	0
1	CURB	供电为 5V 时, PB 口的电流驱动 1: 允许灌电流中大电流驱动 0: 允许灌电流中小电流驱动	R/W	0
0	CURA	供电为 5V 时, PA 口的电流驱动 1: 允许灌电流中大电流驱动 0: 允许灌电流中小电流驱动	R/W	0

7 定时器

7.1 Timer0/Prescaler/BUZZER/PWM

7.1.1. Timer0 计数/定时

TMR0 是一个 8 位定时/计数器寄存器，Timer0 的时钟源可以取值于指令周期、外部实时钟（T0CKI pin）、内部 1.024M 时钟源、比较器的输出，使用外部时钟需要设置 T0CON 的 T0CS0、T0CS1 共同决定

7.1.2. 使用内部时钟：定时模式

定时模式在没有预置器的情况下，定时寄存器每个指令周期自动加 1，设置 TMR0 以后，定时器将在两个时钟周期以后开始自增。

7.1.3. 使用外部时钟/内部 1M 时钟/：计数模式

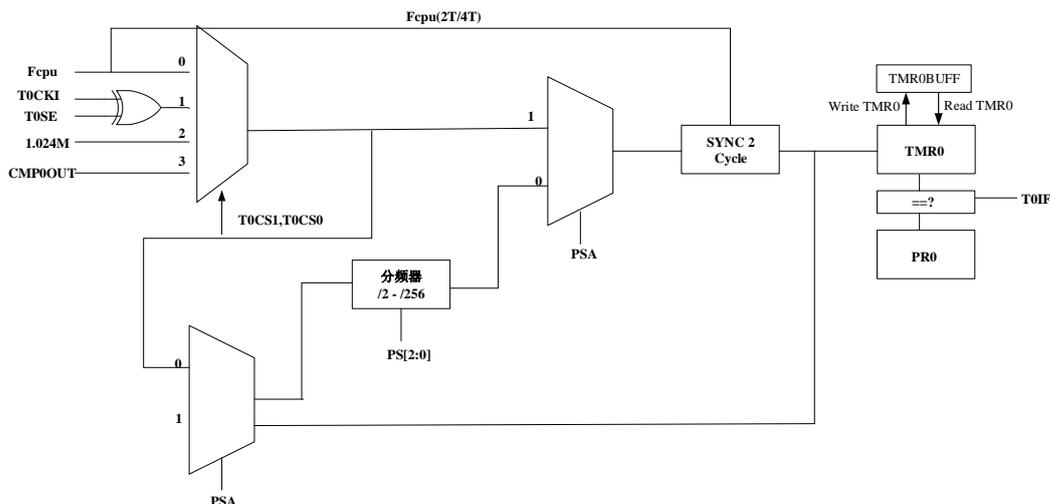
通过 T0CKI 上升或下降沿、1M 时钟，触发 Timer0 寄存器的增加，T0CKI 由 T0SE 位(T0CON<4>)决定上升下降触发。

在没有预置器的情况下，外部时钟输入同样也可以作为预置器输出；T0CKI 与内部时钟同步时能方便处理在 T2 和 T4 周期上的预分频。因此 T0CKI 为高或低电平必须要保持两个以上时钟周期才有效。

7.1.4. Prescaler（预置器）

有一个 8 位的向上计数器作为 Timer 的预置器。注意该预置器只能分配给 Timer0 使用。当作为 Timer0 的预置器的时候，TMR0 会被预置器清零。

Timer0 结构图：



7.1.5. BUZZER（BUZZER 输出）

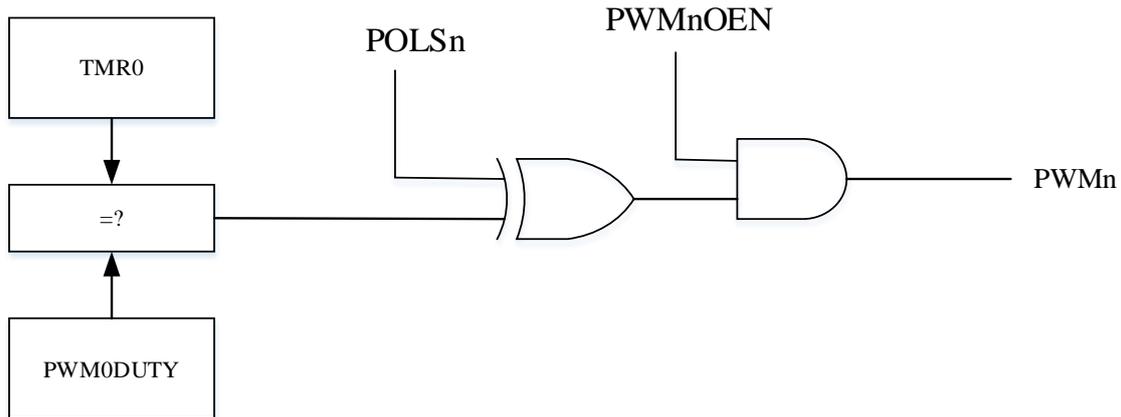
Buzzer 输出是一个简单的 1/2 占空比信号输出，由 TIMER0 产生。当 TMR0 溢出时，Buzzer 开始输出

一个方波，中断间隔时间频率 2 分频后作为 Buzzer 输出的频率。

TMR0 溢出后，Buzzer 输出时，TOIF 有效，且当 TOIE=1 时，使能 TIMER0 中断功能。Buzzer 输出引脚与 GPIO 引脚共用，T0OUT=1 时，该引脚自动设为 Buzzer 输出引脚。如清 T0OUT 位以禁止 Buzzer 输出后，该引脚自动返回到最后一个 GPIO 模式。

7.1.6. TMR0 与 4 路 PWM

当 TMR0 的设置与 PWM0DUTY 寄存器的设置相等时，PWM 输出，如下图所示：



PWM 原理框图 (n=0)

注：PWM0 的输出复用的 IO 有：PA0、PB0、PB1、PB4

PWM0DUTY 寄存器

地址：0XFD4

Bit	Name	Description	Attribute	Reset
7:0	PWM0DUTY[7:0]	PWM 的占空比控制	R/W	0XFF

PR0 (TIMER0 的周期寄存器)

地址：0XFD6

Bit	Name	Description	Attribute	Reset
7:0	PR0[7:0]	TMR0 周期寄存器	R/W	0XFF

TMR0 (定时/计数器 Time lock/Counter register)

地址：0XFD5

Bit	Name	Description	Attribute	Reset
7:0	TMR0[7:0]	8 位定时/计数器	R/W	00000000

T0CON Register (TMR0 控制寄存器)

地址: 0XFD7

Bit	Name	Description	Attribute	Reset
7	TOOUT	PA0 溢出输出 BUZZEE 0: 禁止 BUZZEE 模式 1: 启动 BUZZEE 模式	R/W	0
6:5	TOCS[1:0]	TMRO 的时钟源选择 00: TMRO 时钟源为 CPU 运行时钟 01: TMRO 时钟源为 TOCKI (PA1) 10: TMRO 时钟源为内部 32K 11: TMRO 时钟源是 CMPOOUT 输出	R/W	11
4	TOSE	TMRO 触发方式控制位 1: TOCKI 脚下降沿触发计数 0: TOCKI 脚上升沿触发计数	R/W	1
3	PSA	TMRO 的时钟分频使能 1: 使能 TMRO 的时钟且不分频; 0: 使能 TMRO 的时钟源分频;	R/W	1
2:0	PS[2:0]	分频率选择控制位 000 1:2 001 1:4 010 1:8 011 1:16 100 1:32 101 1:64 110 1:128 111 1:256	R/W	111

在 4T 和 2T 模式下, TOCKI 的时钟选择必须小于 $F_{CPU}/8$;

PWM0CON(PWM 的控制寄存器)

地址: 0XFD3

Bit	Name	Description	Attribute	Reset
7	POLS03	PWM0 通过 PB4 输出的极性 1: 反向输出 0: 正向输出	R/W	0
6	POLS02	PWM0 通过 PB1 输出的极性 1: 反向输出 0: 正向输出	R/W	0
5	POLS01	PWM0 通过 PBO 输出的极性 1: 反向输出 0: 正向输出	R/W	0

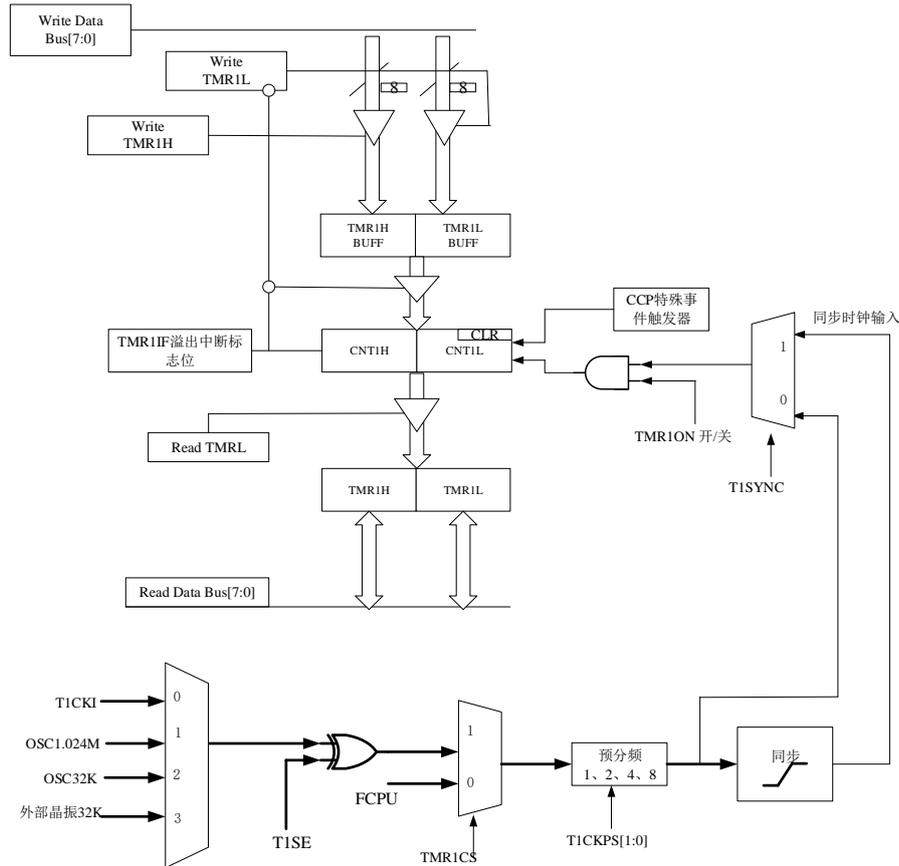
4	POLS00	PWMO 通过 PA0 输出的极性 1: 反向输出 0: 正向输出	R/W	0
3	PWM03OEN	PWMO 输出使能, 通过 PB4 输出 1: 使能 PWMO 输出 0: 禁止 PWMO 输出	R/W	0
2	PWM02OEN	PWMO 输出使能, 通过 PB1 输出 1: 使能 PWMO 输出 0: 禁止 PWMO 输出	R/W	0
1	PWM01OEN	PWMO 输出使能, 通过 PBO 输出 1: 使能 PWMO 输出 0: 禁止 PWMO 输出	R/W	0
0	PWM00OEN	PWMO 输出使能, 通过 PA0 输出 1: 使能 PWMO 输出 0: 禁止 PWMO 输出	R/W	0

7.2 TIMER1 16 位定时/计数器

TIMER1 支持定时/计算器工作模式, 支持 IO 脉宽测量模式、支持定时启动 ADC 工作模式。

TIMER1 为 16 位定时/计数器, 内部拥有写入缓冲区, 当溢出时, 自动从写入缓冲区装载到 TIMER1 计数器作为初值, 然后每个 TIMER1 时钟周期, 计数递增。当 TIMER1 由计数值从 0XFFFF 变为 0X0000 时产生时钟溢出中断, 并自动重载写入缓冲区数值。TIMER1 的时钟源可以是内部 1M 时钟、外部时钟源 (TICKI pin)、外部 32K 时钟、CMP 输出等。

写入 16 位初值时, 首先写入 TMR1H 寄存器, 然后写入 TMR1L, 此时硬件自动将写入缓冲区值重载到 TIMER1 的计数器中; 当读取 TMR1L 时, 自动将当前 TIMER1[15:8]的计数值锁存到 TMR1H 寄存器, 保证读取时间准确。



注：CNT1H 和 CNT1L 是 TIMER1 的内部寄存器。

7.2.1. TMR1L (Timer1 16 位低 8 位寄存器)

地址：0XFD0

Bit	Name	Description	Attribute	Reset
7:0	TMR1L[7:0]	16 位定时/计数器低 8 位	R/W	000000

7.2.2. TMR1H (Timer1 16 位高 8 位寄存器)

地址：0XFD1

Bit	Name	Description	Attribute	Reset
7:0	TMR1H[7:0]	16 位定时/计数器高 8 位	R/W	000000

7.2.3. T1CON0 (Timer1 控制寄存器)

地址：0XFCF

Bit	Name	Description	Attribute	Reset
7:6	T1CK[1:0]	T1 的时钟选择 00: T1CKI 作为时钟 (PC2) 01: 内部 1.024M 作为时钟	R/W	00

		10: 内部 32K 作为时钟 11: 外部晶振 32K 作为时钟		
5:4	T1CKPS[1:0]	T1 的输入时钟分频 00: TIMER1 输入时钟 1:1 分频 01: TIMER1 输入时钟 1:2 分频 10: TIMER1 输入时钟 1:4 分频 11: TIMER1 输入时钟 1:8 分频	R/W	00
3	T1SE	TIMER1 的时钟异或输入选项	R/W	0
2	T1SYNC	1: 使用 FCPU 同步分频后时钟作为 TIMER1 时钟(注意不支持选择内部时钟, 同时 1:1 分频, 并且 T1SYNC 为 1) 0: 使用选择分频时钟作为 TIMER1 时钟	R/W	0
1	TMR1CS	1: 选择 T1CK[1:0]作为外设 TIMER1 的时钟 0: 选择 FCPU 时钟外设 TIMER1 的时钟	R/W	0
0	TMR1ON	1: 使能 Timer1 定时计数器 0: 关闭 Timer1 定时计数器	R/W	0

注:

- 1、在 4T 和 2T 模式下, T1CKI 的时钟选择必须小于 $F_{CPU}/8$;
- 2、在测试模式 (SWD) 下, T1CK[1:0]=11 时, 选择内部高速时钟作为 TIMER1 的计数时钟, 通过 PA4 输出, 在 TIM[10]=11 时且不使能 ADC 模块, 将通过 CCP 的第 16 个通道进行捕捉;
- 3、TMR1 寄存器在写寄存器时, 必须先写高字节在写低字节;

7.2.4. T1CON1 (Timer1 控制寄存器)

地址: 0XFCE

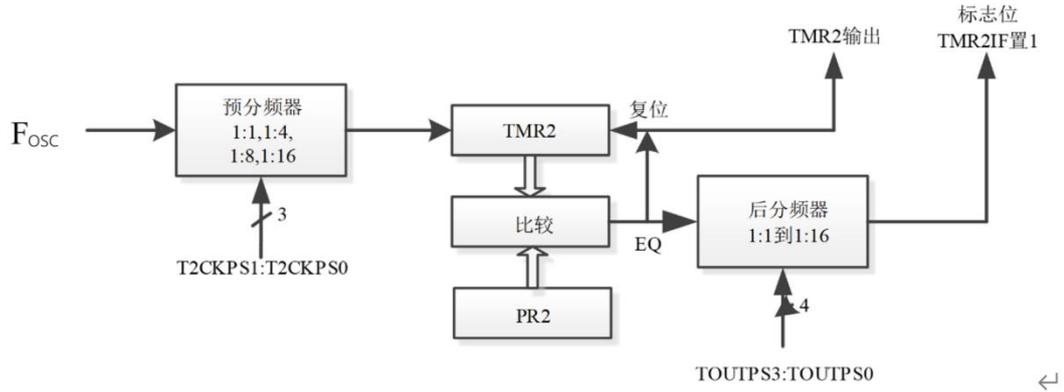
Bit	Name	Description	Attribute	Reset
7	INT2EDGE	外部中断边沿选择位 0: 上升沿触发中断 1: 下降沿触发中断	R/W	0
6	INT1EDGE	外部中断边沿选择位 0: 上升沿触发中断 1: 下降沿触发中断	R/W	0
5	INT0EDGE	外部中断边沿选择位 0: 上升沿触发中断 1: 下降沿触发中断	R/W	0
4	T1EDEG	在 T1M[1:0]在脉宽测量模式时 0: 在下降沿启动计数, 上升沿停止计数 1: 在上升沿启动计数, 下降沿停止计数	R/W	0
3:2	T1CH1:T1CHO	TIMER1 脉宽信号输入选择 00: T1CHO 作为脉宽检测输入信号 PA0) 01: T1CH1 作为脉宽检测输入信号 (PA4)	R/W	00

		10: T1CH2 作为脉宽检测输入信号 (PA5) 11: TMR2CLK 作为脉宽检测输入信号 (配置 T2CON0<2>=1)		
1:0	T1M[1:0]	<p>TIMER1 工作模式</p> <p>00: TIMER1 工作在普通模式, 溢出时间产生中断标志位。</p> <p>01: 脉宽测量模式 0, 测量上升沿与下降沿之间的时间, T1EDGE=1 时, 在脉冲的上升沿开始计数, 脉冲下降沿停止计数, 并在脉冲下降沿沿触中断。T1EDGE=0 时, 在脉冲的下降沿开始计数, 脉冲的上升沿停止计数, 在脉冲上升沿触发中断。产生中断标志后, 可以直接读取 TIMER1 内容, 就可以得到测量值。</p> <p>10: 脉宽测量模式 1, 测量上升沿与下降沿之间的时间, T1EDGE=1 时在脉冲的上升沿 TIMER1 数据寄存器发生复位从 0x0000开始计数, 并在下降沿沿触中断。T1EDGE=0 时, 在脉冲的下降沿 TIMER1 数据寄存器发生复位从 0x0000开始计数, 在脉冲的上升沿触发中断。产生中断标志后, 可以直接读取 TIMER1 内容, 就可以得到测量值。</p> <p>11: 模式 2, TIMER1 工作在定时模式, 在定时产生中断时, 启动 ADC 采集。此模式使用时, 首先要将 ADC 配置完成, 才能使用。</p>	R/W	00

7.3 TIMER2 定时器

Timer2 定时器模块具有以下特征:

- 12 位定时器和周期寄存器 (分别为 TMR2L、TMR2H 和 PR2L、PR2H)
- 可读写 (以上四个寄存器)
- 可软件编程的预分频器 (分频比为 1:1、1:4、1:8、1:16, 与 TMR2 计数有关)
- 可软件编程的后分频器 (分频比为 1:1 至 1:16, 与 TMR2 计数无关)
- 当 TMR2 (TMR2H, TMR2L) 与 PR2 (PR2H, PR2L) 匹配时产生中断
- 采用系统时钟 Fosc 控制



T2 结构图←

7.3.1. PR2L (timer2 的周期寄存器)

地址: 0XFCEB

Bit	Name	Description	Attribute	Reset
7:0	PR2[7:0]	周期寄存器的低 8 位	R/W	0XFF

7.3.2. PR2H timer2 的周期寄存器)

地址: 0XFCA

Bit	Name	Description	Attribute	Reset
7:4	Reserved			
3:0	PR2[11:8]	周期寄存器的高 4 位	R/W	0X1F

7.3.3. TMR2L (Timer2 的低八位寄存器)

地址: 0XFCD

Bit	Name	Description	Attribute	Reset
7:0	TMR2L[7:0]	Timer2 定时/计数器的低 8 位	R/W	0X00

7.3.4. TMR2H (Timer2 的高八位寄存器)

地址: 0XFCC

Bit	Name	Description	Attribute	Reset
7:4	Reserved			
3:0	TMR2H[11:8]	Timer2 定时/计数器的高 4 位	R/W	0X00

注： TMR2 寄存器和 PR2 寄存器在写寄存器时，必须先写高字节在写低字节；

7.3.5. T2CON0 (timer2 控制寄存器)

地址: 0XFC9

Bit	Name	Description	Attribute	Reset
7	PWM2CAEN	Timer2 自减模式使能位 1: 使能 Timer2 自减模式。 0: 禁止 Timer2 自减模式。	R/W	0
6:3	T2OUTPS[3:0]	Timer2 输出后分频选择位 0000: 1:1 后分频值 0001: 1:2 后分频值 0010: 1:3 后分频值 0011: 1:4 后分频值 0100: 1:5 后分频值 0101: 1:6 后分频值 0110: 1:7 后分频值 0111: 1:8 后分频值 1000: 1:9 后分频值 1001: 1:10 后分频值 1010: 1:11 后分频值 1011: 1:12 后分频值 1100: 1:13 后分频值 1101: 1:14 后分频值 1110: 1:15 后分频值 1111: 1:16 后分频值	R/W	0000
2	TMR2ON	Timer2 使能位 1: 使能 Timer2 0: 禁止 Timer2	R/W	0
1:0	T2CKPS[1:0]	Timer2 时钟预分频选择位 00: 预分频值为 1 01: 预分频值为 4 10: 预分频值为 8 11: 预分频值为 16	R/W	0

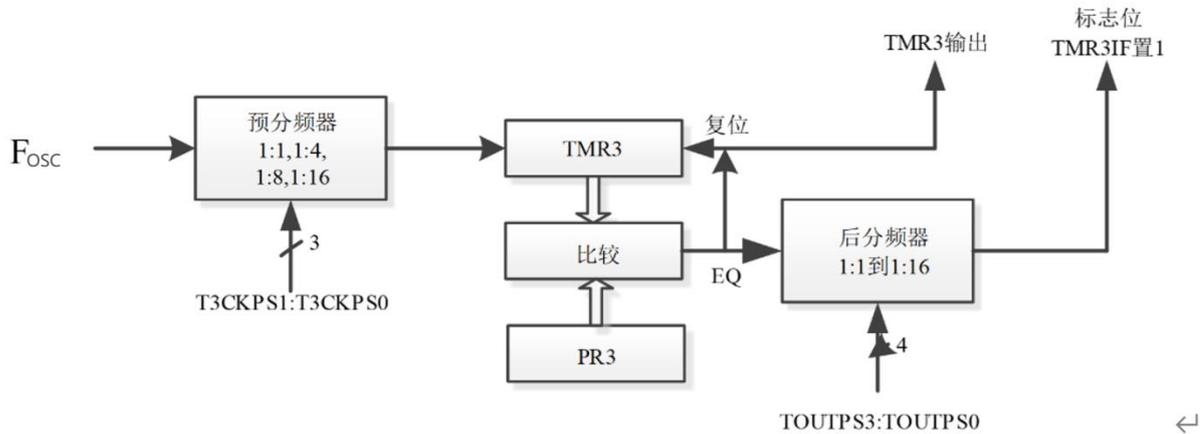
7.4 TIMER3 定时器和 4 路 12 位 PWM

7.4.1. TIMER3 定时器

Timer3 定时器模块具有以下特征:

- 12 位定时器和周期寄存器 (分别为 TMR3L、TMR3H 和 PR3L、PR3H)
- 可读写 (以上四个寄存器)
- 可软件编程的预分频器 (分频比为 1:1、1:4、1:8、1:16)

- 可软件编程的后分频器（分频比为 1:1 至 1:16）
- 当 TMR3（TMR3H, TMR3L）与 PR3（PR3H, PR3L）匹配时产生中断
- 采用系统时钟 Fosc 控制



T3 结构图

PR3L（timer3 的周期寄存器）

地址：0XFB8

Bit	Name	Description	Attribute	Reset
7:0	PR[7:0]	周期寄存器的低 8 位	R/W	0XFF

PR3H (timer3 的周期寄存器)

地址：0XFB9

Bit	Name	Description	Attribute	Reset
7:4	Reserved			
3:0	PR[3:0]	周期寄存器的高 4 位	R/W	0X0F

TMR3L（Timer3 的低八位寄存器）

地址:0XFBA

Bit	Name	Description	Attribute	Reset
7:0	TMR3L[7:0]	Timer3 定时/计数器的低 8 位	R/W	0X00

TMR3H（Timer3 的高八位寄存器）

地址：0XFBB

Bit	Name	Description	Attribute	Reset
7:0	TMR3H[11:8]	Timer3 定时/计数器的高 4 位	R/W	0X00

- 注：1、当 TMR3 运行时，TMR3 寄存器和 PR3 寄存器在写寄存器时，必须先写高字节再写低字节；
 2、当 TMR3 运行时，TMR3 寄存器读取时，必须先读低字节再读高字节；

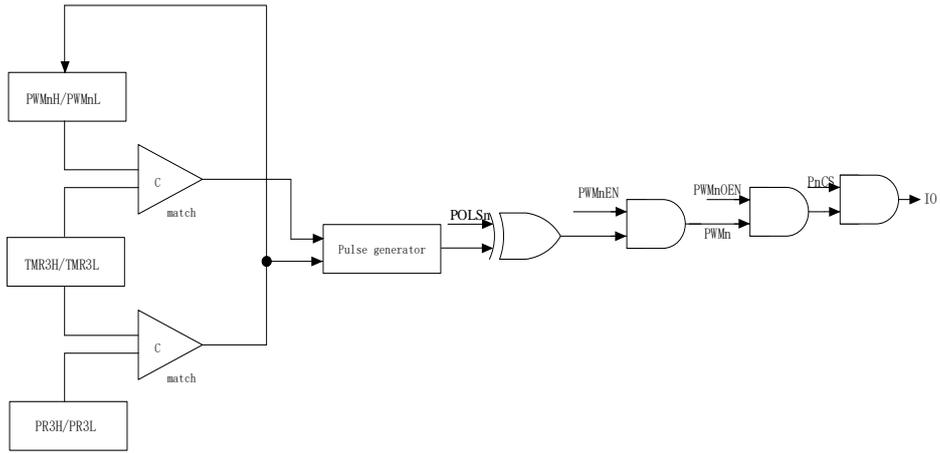
T3CON0(timer3 控制寄存器)

地址：0XFBC

Bit	Name	Description	Attribute	Reset
7	PWM3CAEN	Timer3 自减模式使能位 1: 使能 Timer3 自减模式。 0: 禁止 Timer3 自减模式。	R/W	0
6:3	T3OUTPS[3:0]	Timer3 输出后分频选择位 0000: 1:1 后分频值 0001: 1:2 后分频值 0010: 1:3 后分频值 0011: 1:4 后分频值 0100: 1:5 后分频值 0101: 1:6 后分频值 0110: 1:7 后分频值 0111: 1:8 后分频值 1000: 1:9 后分频值 1001: 1:10 后分频值 1010: 1:11 后分频值 1011: 1:12 后分频值 1100: 1:13 后分频值 1101: 1:14 后分频值 1110: 1:15 后分频值 1111: 1:16 后分频值	R/W	0
2	TMR3ON	Timer3 使能位 1: 使能 Timer3 0: 禁止 Timer3	R/W	0
1:0	T3CKPS[1:0]	Timer3 时钟预分频选择位 00: 预分频值为 1 01: 预分频值为 4 10: 预分频值为 8 11: 预分频值为 16	R/W	0

7.4.2. 4 路 12 位 PWM

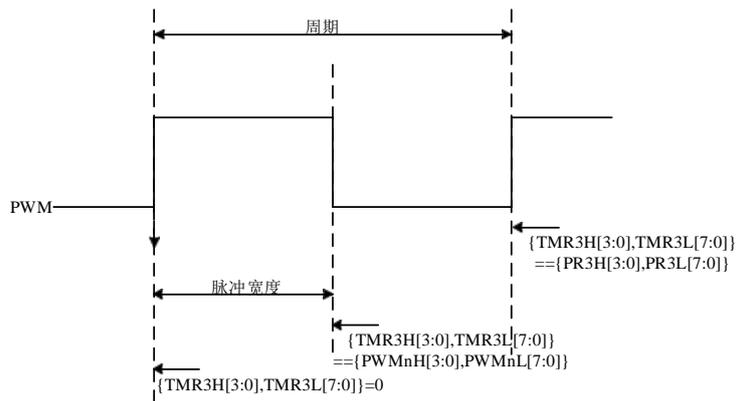
PWM 原理框图如下图所示。



PWM 原理框图 (n=2/3/4/5)

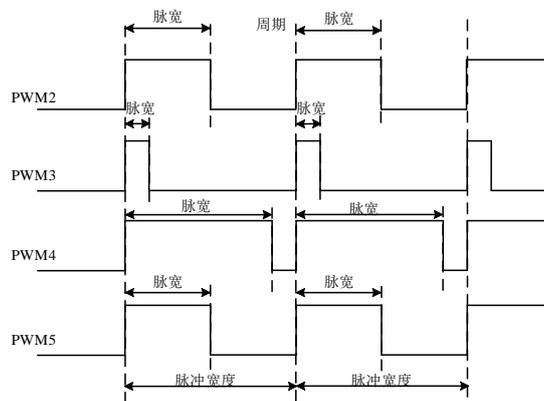
7.4.3. 4 路 PWM 的输出

如下图所示是 PWM 输出原理图，由周期和一段输出保持为高电平（占空比）的时间组成。



PWM 输出原理图

下图为 4 路 PWM 输出图。



4 路 PWM 输出框图

7.4.4. 4 路 PWM 的周期

PWM 周期由 Timer3 的 {PR3H,PR3L} 寄存器指定。PWM 周期公式

PWM 周期为 $\{ (PR3H,PR3L) + 1 \} * F_{sys}$ (系统工作时钟) * (TMR3 的预分频值)

当 {TMR3H,TMR3L} 等于 { PR3H,PR3L } 时, 下一次递增周期将发生以下事件:

1. TMR3H,TMR3L 清零;
2. PWM 输出引脚置 1 (例外: 若 PWM 占空比 = 0%, 引脚不置 1)。

7.4.5. PWM 的占空比

通过给 PWMnH 和 PWMnL 寄存器写入 12 位值可指定 PWM 占空比。

PWM 脉冲宽度 = $\{ (PWMnH,PWMnL) + 1 \} * F_{sys}$ (系统工作时钟) * (TMR3 的预分频值)

$$PWM \text{ 占空比} = \frac{\{PWMnH,PWMnL\}}{\{PR3H,PR3L\}}$$

注: n=2/3/4/5。

7.4.6. PWM 的分辨率

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 12 位分辨率则有 4096 个分立的占空比。

$$\text{分辨率} = \text{Log}_2 \{ (PR3H,PR3L) + 1 \} \text{位}$$

7.4.7. PWM 的工作设置

PWM 频率来自系统工作频率。系统工作频率的任何改变将使 PWM 频率的改变。

应按照以下步骤配置 PWM 工作:

- 1、将相关 IO 配置为输出状态;
- 2、配置 PR3H,PR3L 寄存器设置 PWM 周期;
- 3、配置 TMR3H,TMR3L 寄存器设置计数初值;
- 4、配置 PWMnH,PWMnL 寄存器设置 PWM 占空比;
- 5、配置 T3CON0 寄存器, 使能 Timer3。

注: n=2/3/4/5。

PWM2CON0(PWM 的控制寄存器)

地址: 0XFB1

Bit	Name	Description	Attribute	Reset
7	PWM50EN	PWM5 映射的 IO 输出使能 1: 使能 PWM5 映射的 IO 输出 0: 禁止 PWM5 映射的 IO 输出	R/W	0

6	PWM4OEN	PWM4 映射的 IO 输出使能 1: 使能 PWM4 映射的 IO 输出 0: 禁止 PWM4 映射的 IO 输出	R/W	0
5	PWM3OEN	PWM3 映射的 IO 输出使能 1: 使能 PWM3 映射的 IO 输出 0: 禁止 PWM3 映射的 IO 输出	R/W	0
4	PWM2OEN	PWM2 映射的 IO 输出使能 1: 使能 PWM2 映射的 IO 输出 0: 禁止 PWM2 映射的 IO 输出	R/W	0
3	PWM5EN	PWM5 功能使能 1: 使能 PWM5 功能 0: 禁止 PWM5 功能	R/W	0
2	PWM4EN	PWM4 功能使能 1: 使能 PWM4 功能 0: 禁止 PWM4 功能	R/W	0
1	PWM3EN	PWM3 功能使能 1: 使能 PWM3 功能 0: 禁止 PWM3 功能	R/W	0
0	PWM2EN	PWM2 功能使能 1: 使能 PWM2 功能 0: 禁止 PWM2 功能	R/W	0

PWM2CON1(PWM 的控制寄存器)

地址: 0XFB0

Bit	Name	Description	Attribute	Reset
7: 5	Reserved			
3	POLS5	PWM5 输出极性 1: 反向输出 0: 正向输出	R/W	0
2	POLS4	PWM4 输出极性 1: 反向输出 0: 正向输出	R/W	0
1	POLS3	PWM3 输出极性 1: 反向输出 0: 正向输出	R/W	0
0	POLS2	PWM2 输出极性 1: 反向输出 0: 正向输出	R/W	0

PWM2CON3(PWM 的控制寄存器)

地址: 0XFAE

Bit	Name	Description	Attribute	Reset
7: 6	P5CS[1:0]	PWM5 映射的 IO 输出 00:PA2 输出 01:PB1 输出 10:PC0 输出 11:PC4 输出	R/W	0
5:4	P4CS[1:0]	PWM4 对映射的 IO 输出 00:PA3 输出 01:PB0 输出 10:PC1 输出 11:PC5 输出	R/W	0
3:2	P3CS[1:0]	PWM3 映射的 IO 输出 00:PA0 输出 01:PA4 输出 10:PB5 输出 11:PC2 输出	R/W	0
1:0	P2CS[1:0]	PWM2 映射的 IO 输出 00:PA1 输出 01:PA5 输出 10:PB2 输出 11:PC3 输出	R/W	0

PWM2L(PWM2L 占空比寄存器)

地址: 0XFB0

Bit	Name	Description	Attribute	Reset
7:0	PWM2L[7:0]	PWM2 低位占空比控制	R/W	0X00

PWM2H (PWM2H 占空比寄存器)

地址: 0XFB1

Bit	Name	Description	Attribute	Reset
7:4	Reserved			
3:0	PWM2H[3:0]	PWM2 高位占空比控制	R/W	0X0

PWM3L(PWM3L 占空比寄存器)

地址: 0XFB2

Bit	Name	Description	Attribute	Reset
7:0	PWM3L[7:0]	PWM3 低位占空比控制	R/W	0X00

PWM3H (PWM3H 占空比寄存器)

地址: 0XFB3

Bit	Name	Description	Attribute	Reset
7:4	Reserved			
3:0	PWM3H[3:0]	PWM3 高位占空比控制	R/W	0X0

PWM4L(PWM4L 占空比寄存器)

地址: 0XFB4

Bit	Name	Description	Attribute	Reset
7:0	PWM4L[7:0]	PWM4 低位占空比控制	R/W	0X00

PWM4H (PWM4H 占空比寄存器)

地址: 0XFB5

Bit	Name	Description	Attribute	Reset
7:4	Reserved			
3:0	PWM4H[3:0]	PWM4 高位占空比控制	R/W	0X0

PWM5L(PWM5L 占空比寄存器)

地址: 0XFB6

Bit	Name	Description	Attribute	Reset
7:0	PWM5L[7:0]	PWM5 低位占空比控制	R/W	0X00

PWM5H (PWM5H 占空比寄存器)

地址: 0XFB7

Bit	Name	Description	Attribute	Reset
7:4	Reserved			
3:0	PWM5H[3:0]	PWM5 高位占空比控制	R/W	0X0

7.5 看门狗定时器 (WDT)

看门狗定时器 (WDT) 的运行依赖于芯片里的 RC 振荡器, 无需任何额外电路即能工作。如在睡眠模式。在一般操作或睡眠模式情况下, 看门狗定时器的溢出都会导致 MCU 复位同时 TO (RCON<3>)位被清零。

配置字 WDTE 位(配置选项 2<3>)与 WDTEN 位(WDTON<0>)都可以单独控制看门狗定时器。

如 WDTEN 位(WDTON<0>)与配置字 WDTE 位(配置选项 2<3>)都清零,看门狗定时器不能工作。

在没有预置器时看门狗的溢出约为 128.4/260/391/653ms(或 1.1/1.92/4.31/8.5s),这个时间可以通过配置字 TWDT 位(配置选项 2<2:0>) 设置。

需要看门狗的溢出周期变长可以通过设置 WDTCON 寄存器的 PREDIV 位(WDTON<3:1>)进行分频,因此最长的看门狗溢出周期为 68 秒。

CLRWDWT 指令能使 WDT 和预置器清零,启用看门狗可以防止超时,如果超时 MCU 能复位。

SLEEP 指令重置 WDT 和预置器,启用看门狗就给机器分派了一个最大睡眠时间。

芯片处于调试模式中, WDT 被禁止使用。

7.5.1. WDTCON(看门狗的控制寄存器)

地址: 0XF58

Bit	Name	Description	Attribute	Reset
7:4	Reserved			
3:1	PREDIV[2:0]	WDT 预分频 000: 1:1 001: 1:2 010: 1:3 011: 1:4 100: 1:5 101: 1:6 110: 1:7 111: 1:8	R/W	0
0	WDTEN	WDT 使能 1: 使能 WDT 0: 禁止 WDT	R/W	0

8 捕获/比较/PWM 模块

捕捉/比较/PWM (CCP) 模块包含一个 1 个 16 位寄存器，它可被用作：1 个 16 位捕捉寄存器、1 个 16 位比较寄存器、1 个 12 位 PWM 主/从占空比寄存器。捕捉/比较的寄存器 (CCPR1) 由两个 8 位寄存器组成：CCPR1L (低字节) 和 CCPR1H (高字节)。CCPCON 寄存器控制 CCP1 的操作。捕捉和比较均是和 TMR1 相关，比较匹配将产生特殊事件触发信号，该信号会使 TMR1H 和 TMR1L 寄存器清零。PWM 和 TMR2 相关。Timer3 中的 PWM 和 CCP 中 PWM 可以映射到 CPP 的 HBRIDGE 单元。

CCPR1H 寄存器

地址：0XFC8

Bit	Name	Description	Attribute	Reset
7:0	CCPR1H[7:0]	CCPR1 寄存器高字节，用于捕获、比较。	R/W	0X00

CCPR1L 寄存器

地址：0XFC7

Bit	Name	Description	Attribute	Reset
7:0	CCPR1L[7:0]	CCPR1 寄存器低字节，用于捕获、比较、PWM 的占空比的低 8 位。	R/W	0X00

CCPR1LH 寄存器

地址：0XFC6

Bit	Name	Description	Attribute	Reset
7:4	Reserved			
3:0	CCPR1LH[11:8]	CCPR1 寄存器低字节中的高 4 字节，用于 PWM 的高 4 位	R/W	0X00

CCPCON 寄存器

地址：0XFBB

Bit	Name	Description	Attribute	Reset
7:4	CCPR1CH[3:0]	CCP1 捕获触发信号选择 0000: 外部 CCPCH0 引脚输入信号 (PA0) 0001: 外部 CCPCH1 引脚输入信号 (PA1) 0010: 外部 CCPCH2 引脚输入信号 (PA2) 0011: 外部 CCPCH3 引脚输入信号 (PA3) 0100: OP10UT 输出信号 0101: 内部 OSC32K 输入信号 0110: 外部 32K 晶振输入信号	R/W	0X00

		0111: CMP1OUT 输出信号 1000: 外部 CCPCH4 引脚输入信号 (PB0) 1001: 外部 CCPCH5 引脚输入信号 (PB1) 1010: 外部 CCPCH6 引脚输入信号 (PB2) 1011: 外部 CCPCH7 引脚输入信号 (PB3) 1100: 外部 CCPCH8 引脚输入信号 (PB4) 1101: 外部 CCPCH9 引脚输入信号 (PB5) 1110: 外部 CCPCH10 引脚输入信号 (PC2) 1111: 外部 CCPCH11 引脚输入信号 (PC3)		
3:0	CCPR1M[3:0]	CCP1 模式选择位 0000: 禁止捕捉/比较/PWM 0100: 比较模式, 选择 CCPR1 匹配时将输出置为高电平 (CCPIF 位置 1), 输出引脚为 PB3, TIMER1 溢出时, PWM 为低电平 0101: 比较模式, 选择 CCPR1 匹配时将输出置为低电平 (CCPIF 位置 1), 输出引脚为 PB3, TIMER1 溢出时, PB3 为高电平 0110: 比较模式, 选择 CCPR1 匹配时将产生软件中断 (CCPIF 位置 1, 而 PB3 引脚不受影响); 并启动 ADC 采集 (如果 ADCON=1) 0111: 比较模式, 选择 CCPR1 触发特殊事件 (CCPIF 位置 1, PB3 引脚不受影响); CCP1 清零 Timer1; 并启动 ADC 采集 (如果 ADCON=1) 1000: 捕捉模式, 在每个下降沿发生, 捕捉值存入 CCPR1 寄存器, 并产生中断标志 CCPIF 1001: 捕捉模式, 在每个上升沿发生, 捕捉值存入 CCPR1 寄存器, 并产生中断标志 CCPIF 1010: 捕捉模式, 在每 4 个上升沿发生一次, 捕捉值存入 CCPR1 寄存器, 并产生中断标志 CCPIF 1011: 捕捉模式, 在每 16 个上升沿发生一次, 捕捉值存入 CCPR1 寄存器, 并产生中断标志 CCPIF 1100: 捕捉模式, 下降沿, 复位 TIMER1, 捕捉值存入 CCPR1 寄存器, 并产生中断标志 CCPIF 1101: 捕捉模式, 上升沿, 复位 TIMER1, 捕捉值存入 CCPR1 寄存器, 并产生中断标志 CCPIF 1110: 捕捉模式, 下降沿, 捕捉值存入 CCPR1 寄存器, 不产生中断标志 CCPIF 1111: 捕捉模式, 上升沿, 捕捉值存入 CCPR1 寄存器, 不产生中断标志 CCPIF 001X: PWM 模式, 启动 CCP1 PWM 模式	R/W	0000

8.1 捕捉模式

在捕捉模式下, 当引脚 CCP1 发生事件时, CCPR1H:CCPR1L 将捕捉 TMR1 寄存器的 16 位值。事件

定义如下，由 CCPCON[3:0]进行配置：

1000：捕捉模式，在每个下降沿发生，捕捉值存入 CCPR1 寄存器，并产生中断标志 CCPIF

1001：捕捉模式，在每个上升沿发生，捕捉值存入 CCPR1 寄存器，并产生中断标志 CCPIF

1010：捕捉模式，在每 4 个上升沿发生一次，捕捉值存入 CCPR1 寄存器，并产生中断标志 CCPIF

1011：捕捉模式，在每 16 个上升沿发生一次，捕捉值存入 CCPR1 寄存器，并产生中断标志 CCPIF

1100：捕捉模式，下降沿，复位 TIMER1，捕捉值存入 CCPR1 寄存器，并产生中断标志 CCPIF

1101：捕捉模式，上升沿，复位 TIMER1，捕捉值存入 CCPR1 寄存器，并产生中断标志 CCPIF

1110, 1111：捕捉模式，上升沿和下降沿，捕捉值存入 CCPR1 寄存器，不产生中断标志 CCPIF

进行捕捉后，中断请求标志位 CCP1IF 被置 1。该中断标志位必须用软件清零。如果在 CCPR1H 和 CCPR1L 寄存器对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。输入捕获通道有 CCP1CH[3:0]进行配置：

0000：外部 CCPCH0 引脚输入信号(PA0)

0001：外部 CCPCH1 引脚输入信号(PA1)

0010：外部 CCPCH2 引脚输入信号(PA2)

0011：外部 CCPCH3 引脚输入信号(PA3)

0100：OP1OUT 输出信号

0101：内部 OSC32K 输入信号

0110：外部 32K 晶振输入信号

0111：CMP1OUT 输出信号

1000：外部 CCPCH4 引脚输入信号(PB0)

1001：外部 CCPCH5 引脚输入信号(PB1)

1010：外部 CCPCH6 引脚输入信号(PB2)

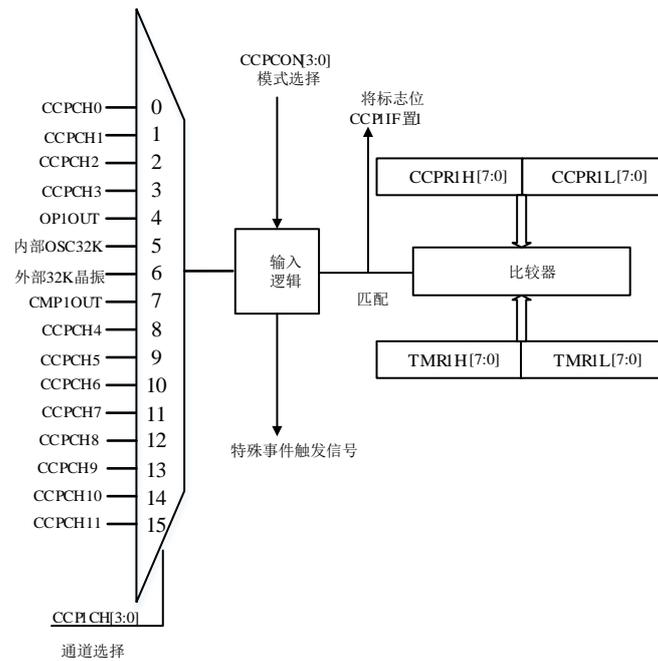
1011：外部 CCPCH7 引脚输入信号(PB3)

1100：外部 CCPCH8 引脚输入信号(PB4)

1101：外部 CCPCH9 引脚输入信号(PB5)

1110：外部 CCPCH10 引脚输入信号(PC2)

1111：外部 CCPCH11 引脚输入信号(PC3)

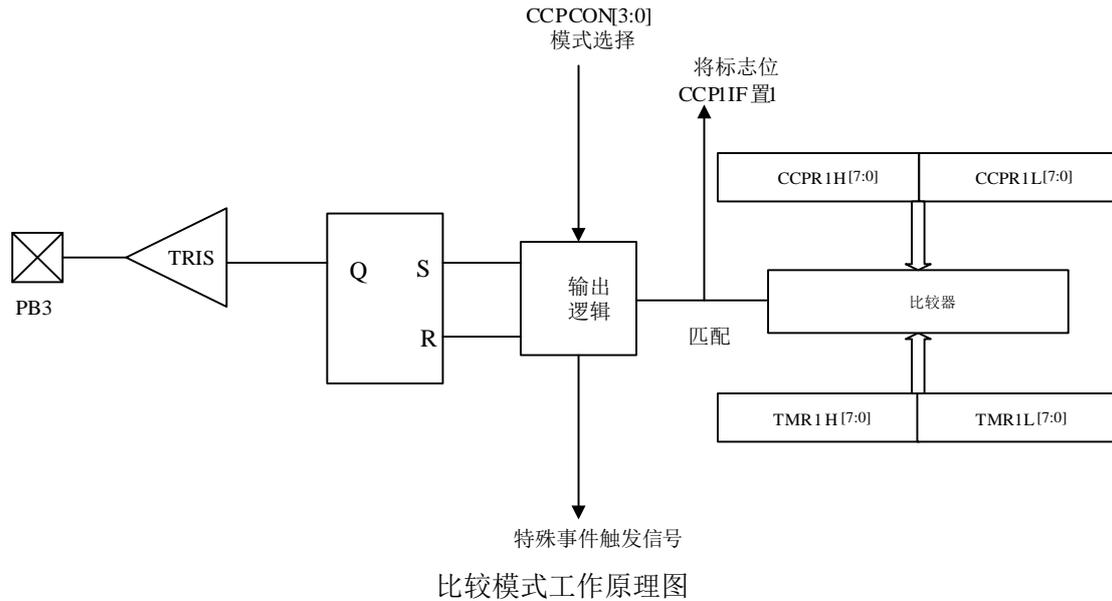


CCP1 捕捉模式工作原理图

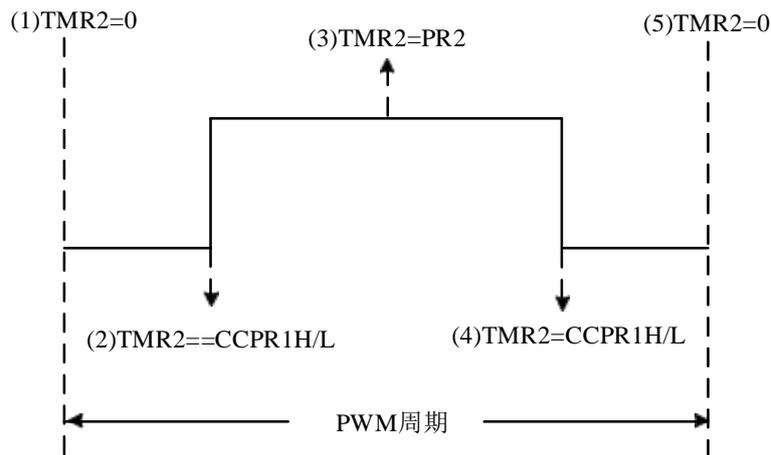
8.2 比较模式

在比较模式下，CCPR1 寄存器的 16 位值不断与 TMR1 寄存器的值进行比较。事件定义如下，由 CCPCON[3:0] 进行配置：

- ✧ 0100：比较模式，选择 CCPR1 匹配时将输出置为高电平（CCPIF 位置 1），输出引脚为 PB3；TIMER1 溢出时，PB3 为低电平；可以实现 16 位 PWM 控制；
- ✧ 0101：比较模式，选择 CCPR1 匹配时将输出置为低电平（CCPIF 位置 1），输出引脚为 PB3；TIMER1 溢出时，PB3 为高电平；可以实现 16 位 PWM 控制；
- ✧ 0110：比较模式，选择 CCPR1 匹配时将产生软件中断（CCPIF 位置 1，而 PB3 引脚不受影响）；并启动 ADC 采集（如果 ADON=1）
- ✧ 0111：比较模式，选择 CCPR1 触发特殊事件（CCPIF 位置 1，PB3 引脚不受影响）；CCP1 清零 TIMER1；并启动 ADC 采集（如果 ADON=1）



8.2.1. PWM 中心对齐模式



PWM 中心对齐示意图

当 PWM2CAEN (T2CON<7>) 开启时, PWM 使能中心对齐模式, 此时 PWM 工作过程如下:

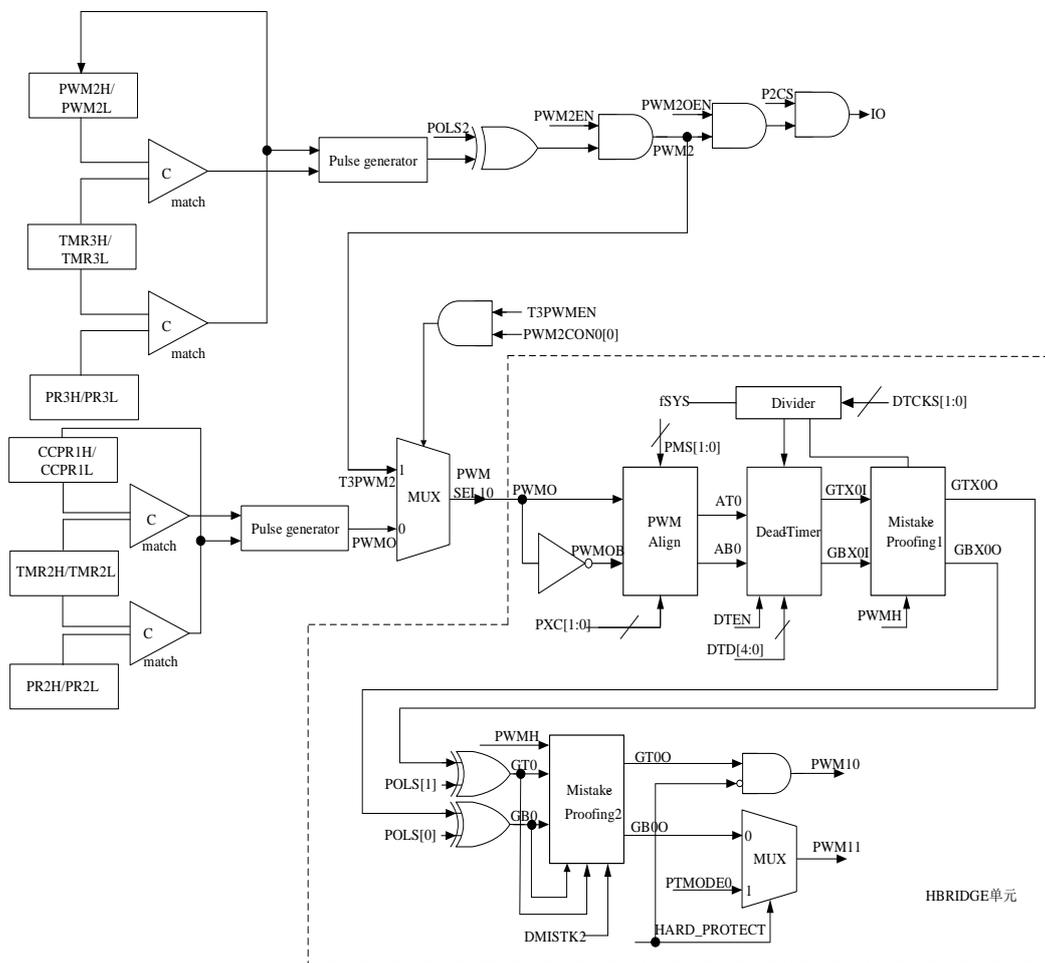
- (1) PWM周期开始, TMR2开始从0递增;
- (2) 当TMR2与CCPR1H/L相等时, PWM开始第一次高低电平变化, TMR2继续递增;
- (3) 当TMR2与PR2相等时, TMR2开始自减;
- (4) 当TMR2再次与CCPR1H/L相等时, PWM再次变化高低电平;
- (5) 当TMR2自减为0时, 此时PWM周期结束, 开始下一个PWM周期。

此时, 实际上 PWM 周期为 $2 \cdot \{ (PR2H, PR2L) + 1 \} \cdot F_{sys}(\text{系统工作时钟}) \cdot (\text{TMR2 的预分频值})$ 。

注意: 普通 PWM 模式切换中心对齐模式需要将 PWM 关闭后重新打开。

8.3 3组互补 PWM 输出

互补式输出控制是通过一系列寄存器实现的。这些寄存器可用于选择 PWM 调制模式、死区时间设置以及输出极性控制等。通过选择可以将 Timer2 或 Timer3 的 PWM 映射到 CPP 的 HBRIDGE 单元。CCP 和 Timer2 构成一路 PWM0 信号，同时输出到 3 个 HBRIDGE 单元，可控制 3 组互补 PWM 输出。Timer3 的 PWM2, PWM3, PWM4 的三路 PWM 信号，分别输出到 3 个 HBRIDGE 单元，可以构成 3 组独立占空比可调的互补 PWM 输出，用于复杂 PWM 互补输出控制。当 Timer3 的 PWMEN 寄存器使能时，PWM2 输出控制 PWM10 和 PWM11，PWM3 输出控制 PWM12 和 PWM13，PWM4 输出控制 PWM14 和 PWM15。PWM10 和 PWM11、PWM12 和 PWM13、PWM14 和 PWM15 分别是三组输出，其中 PWM10 和 PWM11 的原理图如下图，PWM12 和 PWM13、PWM14 和 PWM15 原理相同。



PWM 输出原理图

注：上臂对应 PWM10、PWM12、PWM14

下臂对应 PWM11、PWM13、PWM15

8.3.1. PWM1CON1 (PWM 控制寄存器)

地址：0XFBD

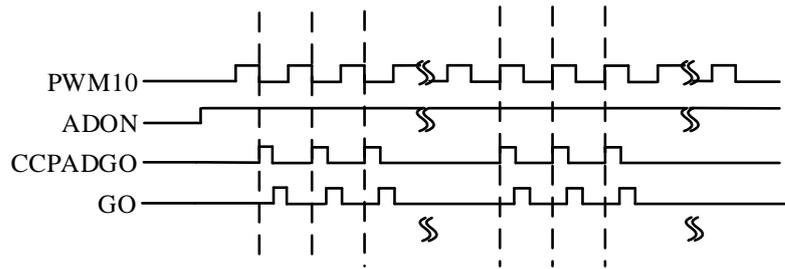
Bit	Name	Description	Attribute	Reset
7:0	PWMADDLY[7:0]	ADC 采集延时时间: $T = \{PWMADDLY[7:0]\} * F_{sys}$	R/W	0

8.3.2. PWM1CON0 (PWM 的控制寄存器)

地址: 0XFBE

Bit	Name	Description	Attribute	Reset
7	Reserve			
6	CMPFLT	CMP1OUT 为故障输入, 当从 0 变为 1 后, 禁止 PWM 输出 1: 使能 CMP1OUT 作为故障输入 0: 禁止 CMP1OUT 作为故障输入	R/W	0
5	ASTART	发生故障后 (ALOSE=1), 当使能的故障都解除后, 自动启动 PWM 输出 1: 使能启动 PWM 输出 0: 禁止启动 PWM 输出	R/W	0
4	ALOSE	发生故障后, 自动关闭 PWM 输出 1: 使能关闭 PWM 输出 0: 禁止关闭 PWM 输出	R/W	0
3	OPFLT	OP1OUT 为故障输入, 当从 0 变为 1 后, 禁止 PWM 输出 1: 使能 OP1OUT 作为故障输入 0: 禁止 OP1OUT 作为故障输入	R/W	0
2	IOFLT	IO 为故障输入, 当从 0 变为 1 后, 禁止 PWM 输出 1: 使能 IO 作为故障输入 0: 禁止 IO 作为故障输入 注: 故障 IO 映射为 PC2;	R/W	0
1	PWMADPOS	PWM 跳变沿使能 ADC 采集 1: 使能 PWM10 由 0 变为 1 采集 0: 使能 PWM10 由 1 变为 0 采集	R/W	0
0	PWM10ADEN	使能 PWM10 输出跳变时, 自动启动 ADC 采集功能 1: 使能 ADC 采集 0: 禁止 ADC 采集	R/W	0

下图为 PWM10 输出跳变时, ADC 采集时序图。



ADC 采集时序图

8.3.3. PMS 寄存器

地址：0XFC5

Bit	Name	Description	Attribute	Reset
7	DMISTK2	二级防呆逻辑 1: BYPASS 二级防呆; 0: 使能二级防呆。	R/W	0
6	PWMH	用于二级防呆上臂逻辑值设定(上下臂同相时强制输出相应电平) 1: 防呆输出 1 0: 防呆输出 0	R/W	0
5:4	PMS2[1:0]	PWM14、PWM15 高压电平转换驱动器的调制模式选择位 00: 互补式控制 01: 非互补式上臂调制 10: 非互补式下臂调制 11: PWM14、PWM15 控制 (由 PXC20 和 PXC21 位分别控制上/下臂输出)	R/W	00
3:2	PMS1[1:0]	PWM12、PWM13 高压电平转换驱动器的调制模式选择位 00: 互补式控制 01: 非互补式上臂调制 10: 非互补式下臂调制 11: PWM12、PWM13 控制 (由 PXC11 和 PXC10 位分别控制上/下臂输出)	R/W	00
1:0	PMS0[1:0]	PWM10、PWM11 高压电平转换驱动器的调制模式选择位 00: 互补式控制 01: 非互补式上臂调制 10: 非互补式下臂调制 11: PWM10、PWM11 控制 (由 PXC00 和 PXC01 位分别控制上/下臂输出)	R/W	00

8.3.4. PXC (PWM 控制输出寄存器)

地址: 0XFC4

Bit	Name	Description	Attribute	Reset
7	T3PWMMEN	使能 TIME3 PWM2, PWM3, PWM4 控制 CCP 的 HBRIDGE 单元 1: 允许使能 0: 禁止使能	R/W	0
6	PWM14SEL	PWM14 映射的 IO 输出 1: PC2 输出 0: PB2 输出	R/W	0
5:4	PXC2[1:0]	PMS2[1:0]==2' b11 时, PWM14、PWM15 高压电平转换驱动器的上/下臂输出选择位 00: 上/下臂输出都关闭 01: 上臂输出关闭/下臂输出导通 10: 上臂输出导通/下臂输出关闭 11: 上/下臂输出都关闭 (防止上/下臂同时导通)	R/W	00
3:2	PXC1[1:0]	PMS1[1:0]==2' b11 时, PWM12、PWM13 高压电平转换驱动器的上/下臂输出选择位 00: 上/下臂输出都关闭 01: 上臂输出关闭/下臂输出导通 10: 上臂输出导通/下臂输出关闭 11: 上/下臂输出都关闭 (防止上/下臂同时导通)	R/W	00
1:0	PXC0[1:0]	PMS0[1:0]==2' b11 时, PWM10、PWM11 高压电平转换驱动器的上/下臂输出选择位 00: 上/下臂输出都关闭 01: 上臂输出关闭/下臂输出导通 10: 上臂输出导通/下臂输出关闭 11: 上/下臂输出都关闭 (防止上/下臂同时导通)	R/W	00

8.3.5. DTC (死区时间控制寄存器)

地址: 0XFC3

Bit	Name	Description	Attribute	Reset
7:6	DTCKS[1:0]	选择死区时间时钟源 f_{DT} 00: $F_{DT}=F_{sys}$ 01: $F_{DT}=F_{sys}/2$ 10: $F_{DT}=F_{sys}/4$ 11: $F_{DT}=F_{sys}/8$	R/W	0
5	DTEN	死区时间使能 1: 允许使能 0: 禁止使能	R/W	0

4:0	DTD[4:0]	死区时间计数器死区时间 = (DTD[4:0]+1)/F _{DT}	R/W	0
-----	----------	--	-----	---

8.3.6. POLS（极性选择寄存器）

地址：0XFC2

Bit	Name	Description	Attribute	Reset
7	Reserved			
6	PTMODE2	保护 PWM15 的输出 1: PWM15 输出 1 0: PWM15 输出 0	R/W	0
5	POLS15	高压电平转换驱动器上臂输出极性控制 1: 反相输出 0: 同相输出	R/W	0
4	POLS14	高压电平转换驱动器下臂输出极性控制 1: 反相输出 0: 同相输出	R/W	0
3	POLS13	高压电平转换驱动器上臂输出极性控制 1: 反相输出 0: 同相输出	R/W	0
2	POLS12	高压电平转换驱动器下臂输出极性控制 1: 反相输出 0: 同相输出	R/W	0
1	POLS11	高压电平转换驱动器上臂输出极性控制 1: 反相输出 0: 同相输出	R/W	0
0	POLS10	高压电平转换驱动器下臂输出极性控制 1: 反相输出 0: 同相输出	R/W	0

8.3.7. PME 寄存器

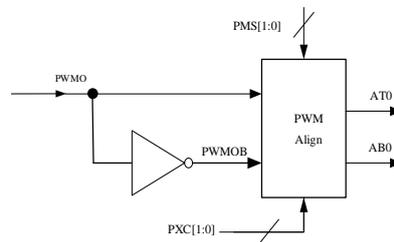
地址：0XFC1

Bit	Name	Description	Attribute	Reset
7	PT1MODE	保护 PWM13 的输出 1: PWM13 输出 1 0: PWM13 输出 0	R/W	0
6	PTOMODE	保护 PWM11 的输出 1: PWM11 输出 1 0: PWM11 输出 0	R/W	0
5	PWM15EN	PWM15 映射的 IO 输出使能 0: 禁止 PC4 输出 1: 使能 PC4 输出	R/W	0

4	PWM14EN	PWM14 映射的 IO 输出使能 0: 禁止 PC2 和 PB2 输出 1: 使能 PC2 和 PB2 输出 注: 选择 PB2 或 PC2 见 PXC[6]。	R/W	0
3	PWM13EN	PWM13 映射的 IO 输出使能 0: 禁止 PB3 输出 1: 使能 PB3 输出	R/W	0
2	PWM12EN	PWM12 映射的 IO 输出使能 0: 禁止 PB4 输出 1: 使能 PB4 输出	R/W	0
1	PWM11EN	PWM11 映射的 IO 输出使能 0: 禁止 PB5 输出 1: 使能 PB5 输出	R/W	0
0	PWM10EN	PWM10 映射的 IO 输出使能 0: 禁止 PC5 输出 1: 使能 PC5 输出	R/W	0

8.3.8. PWM 调制

用户可选择由单臂 PWM 信号、互补式 PWM 信号或软件设置来驱动 PWM，通过 PMS 和 PXC 寄存器的相关位控制，如下图所示。



校准框图

PWM0	PWMOB
0	1
1	0

校准电路输出表 1

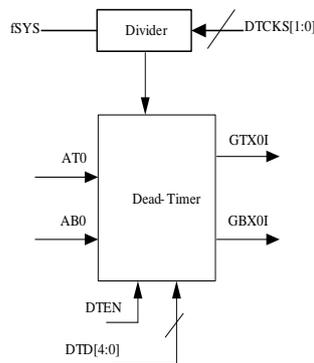
PMS[1:0]	PWM 调制模式	ATO	ABO
00	互补式控制	PWM0	PWMOB
01	非互补式上臂调制	PWM0	0
10	非互补式下臂调制	0	PWM0
11	由 PXC 寄存器控制上下臂输出	PXC[1]	PXC[0]

校准电路输出表 2

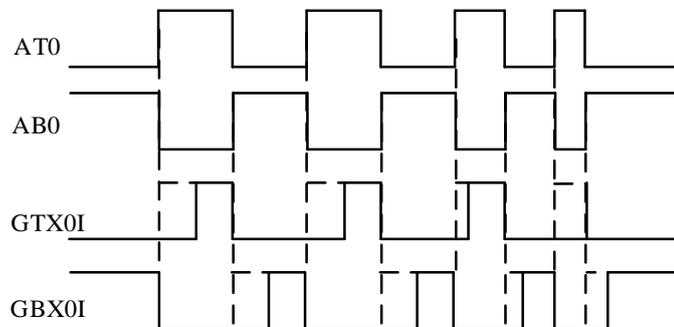
8.3.9. 死区时间

死区时间电路设计的目的是，插入死区时间可确保外部驱动电路晶体管对的上下臂在转态时不会瞬间导通（上下臂 MOS 皆开启）而产生短路电流。为了消除这种危险，设计了一段死区时间，确保输出转态的过程中，两个晶体管处于不会同时导通的状态。死区时间插入使能或除能由 DTC 寄存器的 DTEN 位控制。死区时间要控制在 $0.3\mu\text{s}\sim 5\mu\text{s}$ 左右，可通过 DTCKS1~DTCKS0 位选择死区时钟源，并通过 DTD4~DTD0 位对插入的死区时间进行调整。

下图为死区时间方框图和插入死区时间时序图。需注意的是，若开启死区时间功能，只有在上升沿时插入死区时间，下降沿不变化。



死区时间方框图



死区时间时序图 ($n=0$)

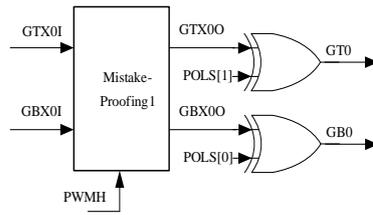
DTEN	GTX0I	GBX0I
1	ATO&DTD	ABO&DTD
0	ATO	ABO

死区时间电路输出表 1

8.3.10. 互补式输出控制防呆电路

此防呆电路设计的目的为，当软件有误写动作发生，或是因外力因素如 ESD 发生时，导致方向控制的寄存器被打乱，造成外部驱动晶体管对上臂与下臂的输出 MOS 皆为开启的状态，此时防呆电路则强迫

输出 MOS 皆为关闭，以保护马达。



一级防呆电路

GTX0I	GBX0I	GTX00	GBX00
0	0	PWMH	0
0	1	0	1
1	0	1	0
1	1	PWMH	0

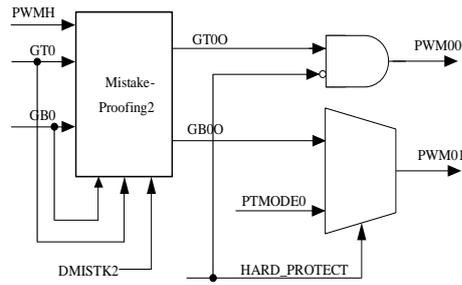
一级防呆电路输出表 1

GTX00	POLS[1]	GTO
0	0	0
0	1	1
1	0	1
1	1	0

一级防呆电路输出表 2

GBX00	POLS[0]	GB0
0	1	1
1	1	0
0	0	0
1	0	1

一级防呆电路输出表 3



二级防呆电路

GTO	GB0	DMISK2	GT00	GB00
0	0	0	PWMH	0
0	1	0	0	1
1	0	0	1	0
1	1	0	PWMH	0

GTO	GB0	DMISK2	GT00	GB00
0	0	1	0	0
0	1	1	1	1
1	0	1	1	0
1	1	1	1	1

二级防呆电路输出表 1

GT00	GB00	HARD_PROTECT	PWM00	PWM01
0	0	0	0	0
0	1	0	0	1
1	0	0	1	0
1	0	0	1	0

PTMOME0	HARD_PROTECT	PWM00	PWM01
0	1	0	0
1	1	0	1

二级防呆电路输出表 2

注：表格中的 0 表示 MOS 关闭，1 表示 MOS 开启。PWMH 为 PMS 寄存器第 6 位。

例子：

```

CCPIE=0;
CCPIF=0;
CCPIP=0;
TMR2IE=0; //中断相关配置

PR2H=0xf;
PR2L=0xff; //周期配置
TMR2H= 0x00;
TMR2H= 0x00; //计数值初始值

CCPR1H=0x8;
CCPR1L=0x00; //占空比配置

PXC=0x40; //上下臂及PWM12IO映射选择
PMS=0x00; //防呆及互补配置
POLS=0x00; //PWM14保护选择及极性选择
PME=0x3f; //PWM映射IO使能及PWM10/12保护选择
CCPCON=0x02;
T2CON=0x04;
    
```

8.3.11. 增强型六路 PWM

增强型 PWM 模式可在最多六个输出引脚上产生 PWM 信号。可以通过四种 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

PWM 输出与 I/O 引脚复用，并被指定为 PWM10/11/12/13/14/15 依次复用 PC5、PB5、PB2、PB3、PC2、PC4。PWM 引脚的极性可配置，可通过配置 POLS 寄存器选择极性。每两个 I/O 为一组半桥。全桥为二组半桥。

注意：

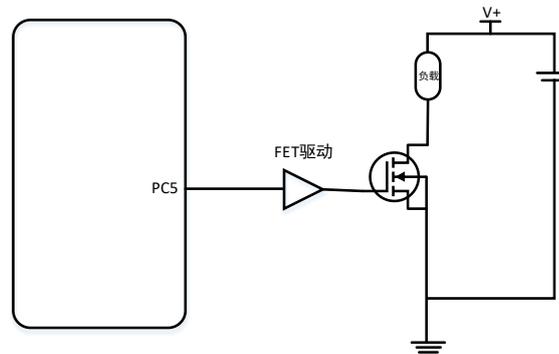
- 必须正确配置每个 PWM 输出的 TRIS 寄存器值；
- 清零 CCPCON 寄存器将放弃所有 PWM 输出引脚；
- 当 PWM_nOE 不使能时，增强型 PWM 模式所不使用的任何引脚均可用于其他引脚功能。
(n=10/11/12/13/14/15)

8.3.11.1. 单 PWM 模式

在单 PWM 模式下，有一个引脚用作输出。通过配置 PMS 寄存器高压电平转换驱动器的调制模式选择位为：非互补式上臂调制或非互补式上臂调制下臂为单桥模式。



单 PWM 模式输出图



单 PWM 电路

例子

```

CCPIE=0;
CCPIF=0;
CCPIP=0;
TMR2IE=0; //中断相关配置

PR2H=0xf;
PR2L=0xff; //周期配置
TMR2H= 0x00;
TMR2L= 0x00; //计数值初始值

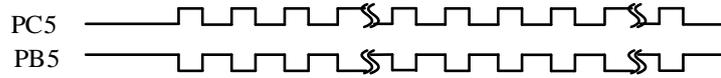
CCPR1H=0x8;
CCPR1L=0x00; //占空比配置

PMS=0x3d; //单桥输出
PXC=0x40; //上下臂及PWM12IO映射选择及单桥输出
POL=0x00; //PWM14保护选择及极性选择
PME=0x3f; //PWM映射IO使能及PWM10/12保护选择
CCPCON=0x02; //使能PWM模式
T2CON=0x04; //使能 TMR2
    
```

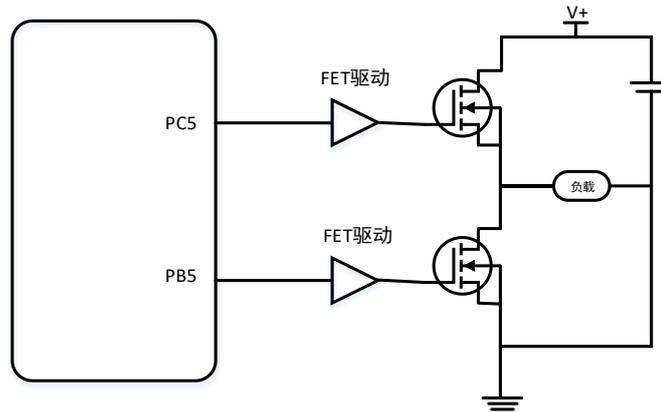
8.3.11.2. 半桥模式

在半桥模式下，有两个引脚用作输出以驱动负载。PC5 和 PB5 为一组半桥，PB2 和 PB3 为一组半桥，PC2 和 PC4 为一组半桥。PWM 输出信号被输出到 PC5、PB2、PC2 引脚，而互补 PWM 输出信号被输

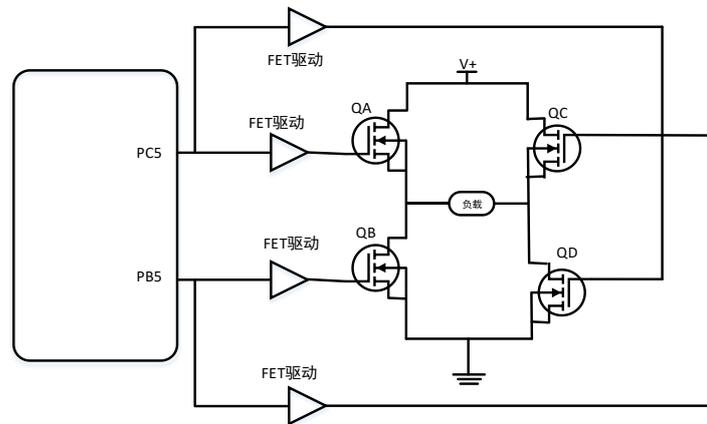
出到 PB5、PB3、PC4。下图是一组半桥输出图，其它两组半桥同理。半桥模式可以运用为全桥模式。



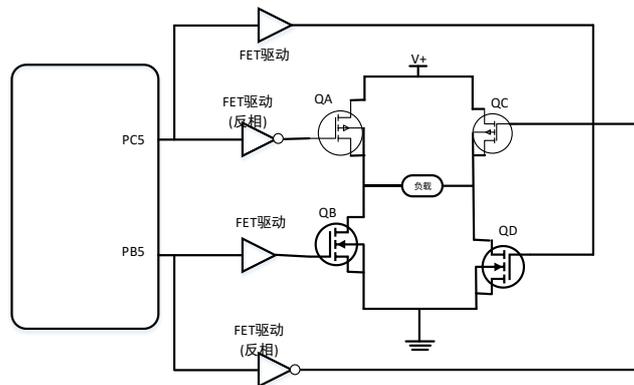
半桥模式输出图



标准半桥电路图



半桥输出驱动全桥电路（4NMOS）注：PB2 和 PB3,PC2 和 PC4 同图上一样



半桥输出驱动全桥电路（2PMOS+2NMOS）注：PB2 和 PB3,PC2 和 PC4 同图上一样

标准半桥例子

```

CCPIE=0;
CCPIF=0;
CCPIP=0;
TMR2IE=0; //中断相关配置

PR2H=0xf;
PR2L=0xff; //周期配置
TMR2H= 0x00;
TMR2L= 0x00; //计数值初始值

CCPR1H=0x8;
CCPR1L=0x00; //占空比配置

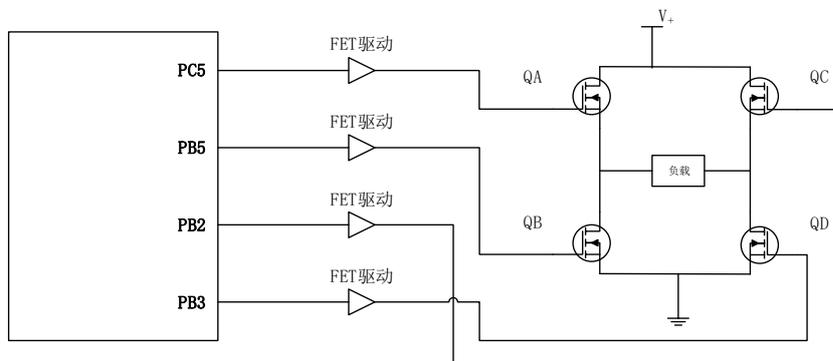
PMS=0x00; //互补输出
PXC=0x40; //上下臂及PWM12IO映射选择
POLS=0x00; //PWM14保护选择及极性选择
PME=0x3f; //PWM映射IO使能及PWM10/12保护选择
CCPCON=0x02; //使能PWM模式
T2CON=0x04; //使能 TMR2
    
```

8.3.11.3. 全桥模式

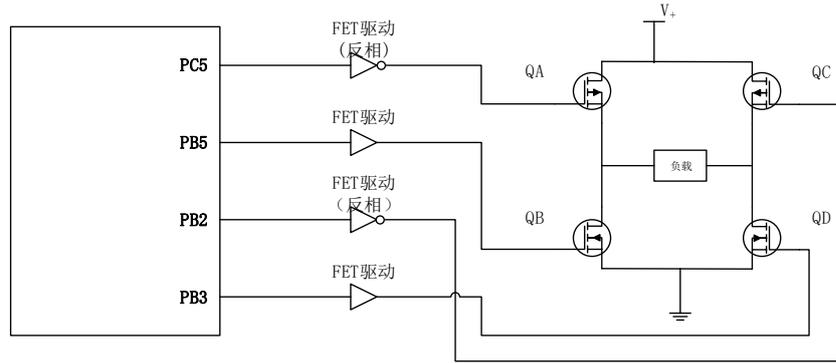
在全桥模式下，所有四个引脚用作输出。这里以第一组半桥 PC5 和 PB5，第二组半桥 PB2 和 PB3 为例。三组半桥可以自由两两组合为全桥。

全桥应用示例 a 所示为一个全桥应用示例，使用 4 个 NMOS。

全桥应用示例 b 另外的一个全桥应用，使用 2 个 PMOS 和 2 个 NMOS。

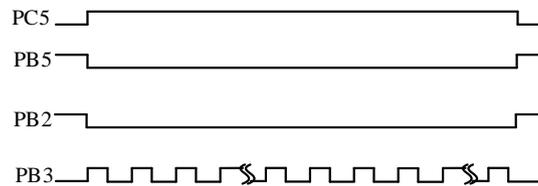


全桥应用示例 a



全桥应用示例 b

使用 4 个 NMOS 时，在全桥正向模式下，PC5 引脚驱动为有效状态，PB3 引脚为调制输出，而 PB5 和 PB2 则被驱动为无效状态，如全桥 PWM 输出示例 a 所示。需要配置 PMS 寄存器第二组半桥为非互补式下臂调制，第一组半桥为 PXC 控制，并且在 PXC 寄存器第一组桥为上臂导通，下臂关闭。



全桥 PWM 输出示例 a

全桥 PWM 正向输出例子：

```

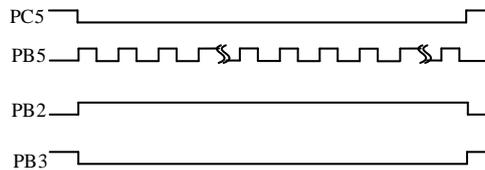
CCPIE=0;
CCPIF=0;
CCPIP=0;
TMR2IE=1; //中断相关配置

PR2H=0xf;
PR2L=0xff; //周期配置
TMR2H= 0x00;
TMR2L= 0x00; //计数值初始值

CCPR1H=0x8;
CCPR1L=0x00; //占空比配置
PMS =0x0b; //第一组PWMPXC控制输出，第二组PWM非互补式下臂调制
PXC=0x02; //第一组PWM上臂导通下臂关闭及PWM12IO映射选择
POLS=0x00; //PWM14保护选择及极性选择
PME=0x3f; //PWM映射IO使能及PWM10/12保护选择
CCPCON=0x02; //使能PWM模式
T2CON=0x04; //使能 TMR2
    
```

使用 4 个 NMOS 时，在全桥反向模式下，PB2 驱动为有效状态，PB5 引脚为调制输出，而 PC5 和 PB3

则被驱动为无效状态，如全桥 PWM 输出示例 b 所示。需要配置在 PMS 寄存器配置第一组桥为非互补式下臂调制，第二组桥为 PXC 控制，并且 PXC 寄存器第二组桥为上臂导通，下臂关闭。



全桥 PWM 输出示例 b

全桥 PWM 输出示例 b 例子:

```

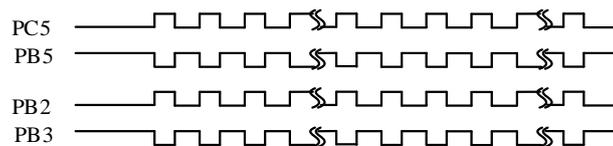
CCPIE=0;
CCPIF=0;
CCPIP=0;
TMR2IE=0; //中断相关配置

PR2H=0xf;
PR2L=0xff; //周期配置
TMR2H= 0x00;
TMR2L= 0x00; //计数值初始值

CCPR1H=0x8;
CCPR1L=0x00; //占空比配置

PMS=0x0e; //第一组非互补式下臂调制，第二组PWMPXC控制输出
PXC=0x08; //第二组PWM上臂导通下臂关闭及PWM12IO映射选择
POLLS=0x00; //PWM14保护选择及极性选择
PME=0x3f; //PWM映射IO使能及PWM10/12保护选择
CCPCON=0x02; //使能PWM模式
T2CON=0x04; //使能 TMR2
    
```

全桥 PWM 输出示例 c 是使用 2 个 PMOS 和 2 个 NMOS 的 PWM 输出波形。PMS 寄存器中两组桥均配置为互补式控制。



全桥 PWM 输出示例 c

全桥 PWM 输出示例 c 例子:

```

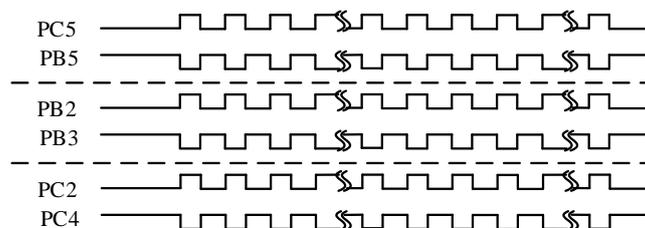
CCPIE=0;
CCPIF=0;
CCPIP=0;
TMR2IE=0; //中断相关配置

PR2H=0xf;
PR2L=0xff; //周期配置
TMR2H= 0x00;
TMR2L= 0x00; //计数值初始值

CCPR1H=0x8;
CCPR1L=0x00; //占空比配置

PMS=0x00; //互补式输出
PXC=0x00; // PWM12IO映射选择
POLS=0x00; //PWM14保护选择及极性选择
PME=0x3f; //PWM映射IO使能及PWM10/12保护选择
CCPCON=0x02; //使能PWM模式
T2CON=0x04; //使能 TMR2
    
```

8.3.11.4. 3 组 PWM 输出



3 组 PWM 输出

如上图所示，PC5 和 PB5 是第一组半桥 PWM 输出，PB2 和 PB3、PC2 和 PC4 分别为第二组和第三组，他们的波形和第一组是一样的。

例子同全桥 PWM 输出示例 c 例子

9 通用同步/异步收发器 (USART)

通用同步/异步收发器 (Universal Synchronous Asynchronous Receiver Transmitter, USART) 模块是两个串行 I/O 模块之一。(通常, USART 也被称为串行通信接口或 SCI)。USART 支持全双工异步通信。

IO 状态

工作模式	异步模式		同步模式		BCLK(外部时钟) PA5
	UART_MUX=1	UART_MUX=0	UART_MUX=1	UART_MUX=0	
IO 功能	RX/TX	RX/TX	DT/CK	DT/CK	DUAL(单线半双工)
映射 IO	PB1/PB0	PA0/PA1	PB1/PB0	PA0/PA1	PA1/PB0

TXSTA

地址: 0XF4E

Bit	Name	Description	Attribute	Reset
7	CSRC	时钟源选择位 异步模式: 忽略。 同步模式: 1: 主机模式 (时钟来自内部 BRG) 0: 从机模式	R/W	0
6	TX9	9 位发送使能位 1: 选择 9 位发送 0: 选择 8 位发送	R/W	0
5	TXEN	发送使能位 1: 使能发送 0: 禁止发送	R/W	0
4	SYNC	USART 模式选择位 1: 同步模式 0: 异步模式	R/W	0
3	TXPARIT	使能硬件自动奇偶校验 1: 自动填充发送第 9 位校验位 0: 校验位由 TXD9 填充	R/W	0
2	BRGH	高波特率选择位 异步模式: 1: 高速 0: 低速 同步模式: 在此模式下未使用。	R/W	0
1	TRMT	发送移位寄存器状态位 1: TSR 空 0: TSR 满	R/W	1

0	TX9D	发送数据的第 9 位 该位可以是地址/数据位或奇偶校验位。	R/W	0
---	------	----------------------------------	-----	---

RCSTA

地址: 0XF4D

Bit	Name	Description	Attribute	Reset
7	SPEN	串口使能位 1: 使能串口 (将 RX/DT 和 TX/CK 引脚配置为串口引脚) 0: 禁止串口 (保持在复位状态)	R/W	0
6	RX9	9 位接收使能位 1: 选择 9 位接收 0: 选择 8 位接收	R/W	0
5	SREN	单字节接收使能位 异步模式: 忽略。 同步主机模式: 1: 使能单字节接收 0: 禁止单字节接收 此位在接收完成后清零。 同步从机模式: 忽略。	R/W	0
4	CREN	连续接收使能位 1: 使能接收器 0: 禁止接收器	R/W	0
3	BRGM	与 BRGH 组成 UART 时钟选择位	R/W	0
2	FERR	帧错误位 1: 帧错误 (可以通过读 RCREG 寄存器刷新并接收下一个有效字节) 0: 无帧错误	R/W	0
1	OERR	溢出错误位 1: 溢出错误 (可以通过清零 CREN 位清除) 0: 无溢出错误	R/W	0
0	RX9D	接收数据的第 9 位 该位可以是地址/数据位或奇偶校验位, 必须由用户固件计算得到	R/W	0

UARTCON

地址: 0XF4C

Bit	Name	Description	Attribute	Reset
-----	------	-------------	-----------	-------

7	INVMODE	数据电平反向接受发送使能位 异步： 1: 使能 0: 禁止 同步： 忽略	R/W	0
6	PARITY	接收奇偶校验位 1: 表示有奇数个 1 (奇校验) 0: 表示有偶数个 1 (偶校验)	R/W	0
5	ODD_EVEN	设定发送、接收奇偶校验位功能 1: 奇校验 0: 偶校验	R/W	0
4	BCLK	波特率发生器时钟源选择位 1: 外部时钟 (PA5) 0: 内部系统时钟	R/W	0
3	HALF_DUPLEX	单线半双工使能位 1: 使能 0: 禁止	R/W	0
2	DLSB	数据位高低位选择位 1: 高位在前 (MSB) 0: 低位在前 (LSB)	R/W	0
1:0	STOP_BIT[1:0]	停止位个数设置位 00: 1 位停止位 01: 2 位停止位 10: 3 位停止位	R/W	00

9.1 波特率发生器

BRG 是一个 8 位的发生器，受 BRGH(TXSTA<2>)位控制。波特率计算公式如下：

BCLK	BRGM	BRGH	波特率公式
0	1	0	$FOSC/[4*(SPBRG)]$
0	1	1	$FOSC/[2*(SPBRG)]$
0	0	0	$FOSC/[64*(SPBRG)]$
0	0	1	$FOSC/[16*(SPBRG)]$
1	1	0	$BCLK/[4*(SPBRG)]$
1	1	1	$BCLK/[2*(SPBRG)]$
1	0	0	$BCLK/[64*(SPBRG)]$
1	0	1	$BCLK/[16*(SPBRG)]$

针对工作在异步模式下，工作频率 F_{osc} 为 16MHz，采用 8 位 BRG，目标波特率为 9600bps 的器件：

$$\text{目标波特率} = F_{osc}/(64([\text{SPBRG}]))$$

求解 SPBRG：

$$\begin{aligned} X &= ((F_{osc}/\text{目标波特率})/64) \\ &= ((16000000)/9600/64) \\ &= [26.042]=26 \end{aligned}$$

$$\begin{aligned} \text{计算得到的波特率} &= 16000000/(64 \times 26) \\ &= 9615 \end{aligned}$$

$$\begin{aligned} \text{误差} &= (\text{波特率计算结果}-\text{目标波特率})/\text{目标波特率} \\ &= (9615-9600)/9600=0.16\% \text{误差} \end{aligned}$$

针对工作在同步模式下，工作频率为 BCLK，波特率计算与异步模式相同。

9.2 异步发送器

发送器的核心是发送（串行）移位寄存器（Transmit Shift Register, TSR）。移位寄存器从读/写发送缓冲寄存器 TXREG 中获取数据。TXREG 寄存器中的数据由软件装入。直到前一次装入的停止位已被发送，才会向 TSR 寄存器装入新数据。一旦停止位发送完毕，TXREG 寄存器中的新数据（如果有的话）就会被装入 TSR。

一旦 TXREG 寄存器向 TSR 寄存器传输了数据（在 1 个 T_{CY} 内发生），TXREG 寄存器就为空，同时标志位 TXIF (PIR1<5>) 置 1。可以通过将中断使能位 TXIE (PIE1<5>) 置 1 或清零来使能/禁止该中断。不管 TXIE 的状态如何，只要中断发生，TXIF 就会置 1 并且不能用软件清零。TXIF 不会在 TXREG 装入新数据时立即被清零，而是在装入指令后的第二个指令周期复位。因此在 TXREG 装入新数据后立即查询 TXIF，会返回无效结果。标志位 TXIF 指示的是 TXREG 寄存器的状态，而另一个位 TRMT (TXSTA<1>) 则指示 TSR 寄存器的状态。TRMT 是只读位，它在 TSR 寄存器为空时被置 1。TRMT 位与任何中断均无关联，因此要确定 TSR 寄存器是否为空，用户只能对此位进行轮询。

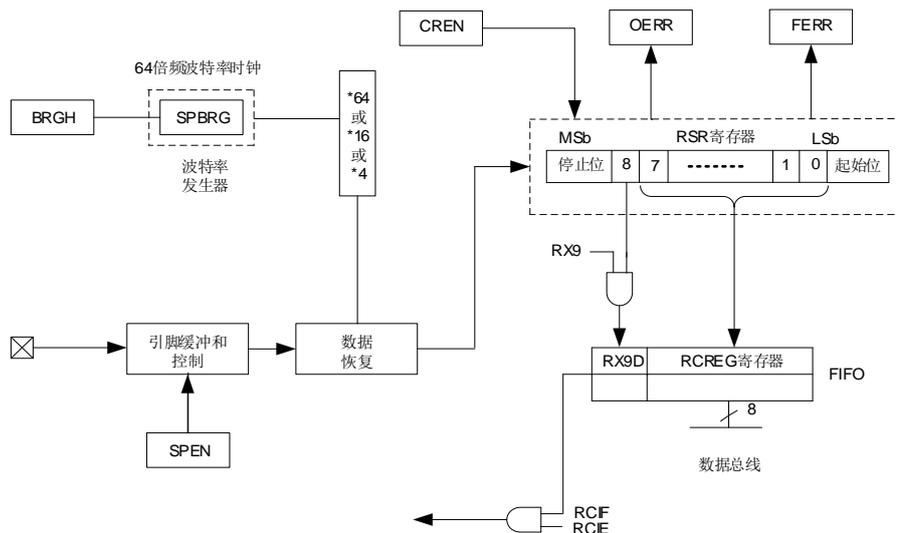
设置异步发送的操作步骤如下：

- 初始化 SPBRG，设置合适的波特率。按需要将 BRGH 位置 1 或清零，以获得目标波特率。
- 如果需要中断，将使能位 TXIE 置位。
- 若需要发送 9 位数据，将发送位 TX9 置 1。发送的第 9 位可以是地址位也可以是数据位。
- 通过将 TXEN 位置 1 使能发送，此操作同时也会将 TXIF 位置 1。
- 如果选择发送 9 位数据，应该将第 9 位数据装入 TX9D 位。
- 将数据装入 TXREG 寄存器（启动发送）。
- 若想使用中断，请确保将 INTCON 寄存器中的 GIEH (INTCON<7>) 或 GIEL (INTCON<6>) 置 1。

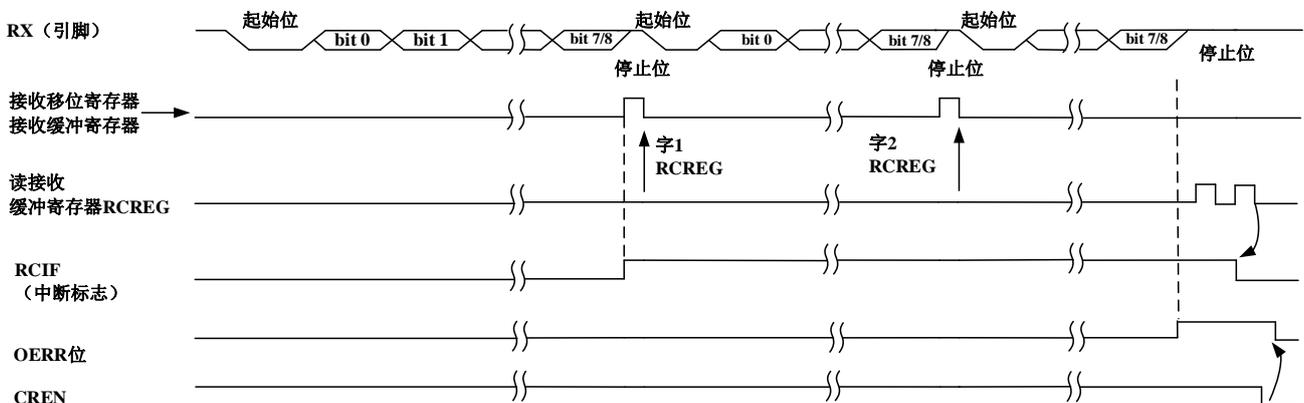
通常用于 RS-232 系统。

设置异步接收的操作步骤如下：

- 初始化 SPBRG，设置合适的波特率。按需要将 BRGH 位置 1 或清零，以获得目标波特率。
- 如果需要中断，将使能位 RCIE 置位。
- 若需要发送 9 位数据，将发送位 RX9 置 1。
- 通过将 CREN 位置 1，使能接收。
- 当接收完成时标志位 RCIF 将被置 1，此时如果使能位 RCIE 已值 1，还将产生一个中断。
- 读 RCSTA 寄存器以获取第 9 位数据（如果已使能），并判断是否在接收过程中是否发生了错误。
- 通过读 RCREG 寄存器来读取接收到的 8 位数据。
- 如果发生错误，通过将使能位 CREN 清零来清除错误。
- 若想使用中断，请确保将 PIE0 寄存器中的 GIEH (INTCON<7>) 或 GIEL (INTCON<6>) 置 1。



UART 接收原理图



异步接收时序

9.4 同步模式

将CSRC 位 (TXSTA<7>) 置1 可以进入同步主机模式。在此模式中, 数据以半双工方式 (即发送和接收不同时进行) 发送。发送数据时, 禁止接收, 反之亦然。将SYNC 位 (TXSTA<4>) 置1 可进入同步模式。此外, 应将使能位SPEN (RCSTA<7>) 置1, 分别把TX和RX 引脚配置为CK (时钟) 和DT (数据) 线。主机模式意味着处理器在CK 时钟线上发送主机时钟信号。

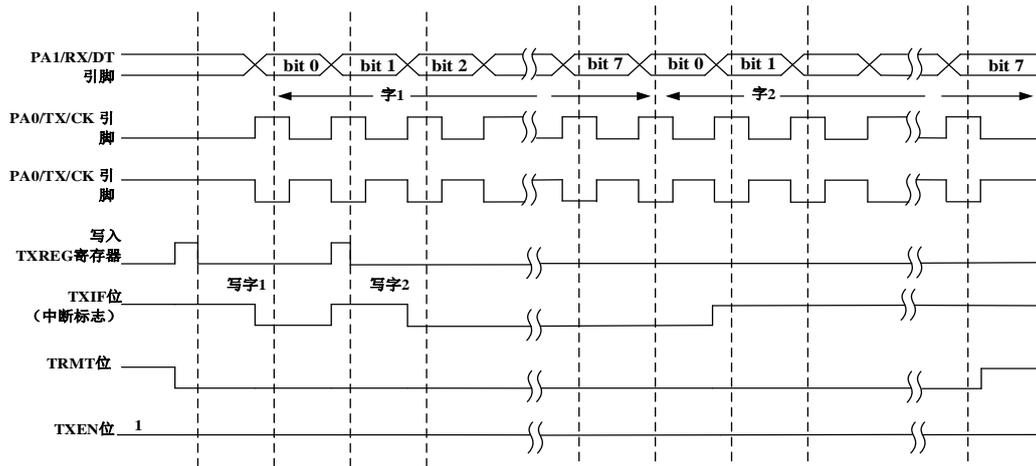
9.4.1. 同步主机发送

发送器的核心是发送 (串行) 移位寄存器 (TSR)。移位寄存器从读/写发送缓冲寄存器TXREG中获取数据, 而TXREG寄存器中的数据由软件装入。在前一次装入数据的最后一位发送完成后, 才会向TSR 寄存器装入新数据。一旦最后一位发送完成, 就会将TXREG 寄存器中的新数据 (如果有的话) 装入TSR。

一旦TXREG 寄存器向TSR 寄存器传输了数据 (在1个TCY 内发生), TXREG 寄存器就为空, 同时标志位TXIF (PIR1<5>) 被置1。可以通过将中断使能位TXIE (PIE1<5>) 置1 或清零来使能/ 禁止该中断。TXIF 的设置与使能位TXIE 的状态无关, 且不能用软件清零。只有在新数据写入TXREG 寄存器时, TXIF 才会复位。TXIF 表示的是TXREG 寄存器的状态, 而另一个标志位TRMT (TXSTA<1>) 则表示TSR 寄存器的状态。TRMT是只读位, 它在TSR 寄存器为空时被置1。TRMT 位与任何中断均无关联, 因此要判断TSR 寄存器是否为空, 用户只能对此位进行轮询。TSR 寄存器并未映射到数据存储寄存器中, 因此用户不能直接访问它。

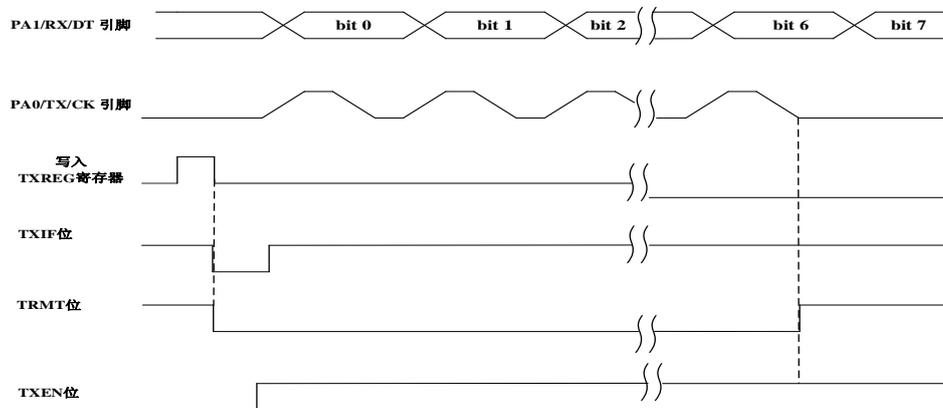
设置同步主机发送操作的步骤如下:

- 初始化SPBRG 寄存器, 设置合适的波特率。
- 通过将SYNC、SPEN 和CSRC 位置1, 使能同步主机串口。
- 若需要中断, 将中断使能位TXIE 置1。
- 若需要发送9 位数据, 将TX9 位置1。
- 将TXEN 位置1, 使能发送。
- 如果选择发送9位数据, 将第9位数据装入TX9D位。
- 将数据装入TXREG 寄存器, 启动发送。
- 若想使用中断, 请确保将 PIE0 寄存器中的 GIEH (INTCON<7>) 或 GIEL (INTCON<6>) 置 1。



注：同步主控模式，SPBRG=0，连续发送两个8位字节

同步发送时序



同步发送时序（有TXEN位控制）

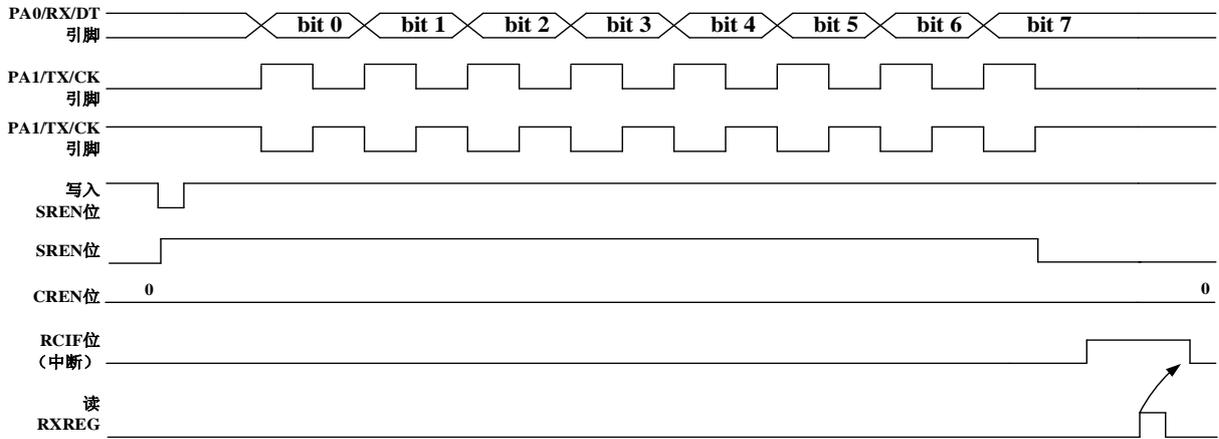
9.4.2. 同步主机接收

一旦选择了同步模式，只要将单字节接收使能位SREN（RCSTA<5>）或连续接收使能位CREN（RCSTA<4>）置1，即可使能接收。在时钟的下降沿采样RX引脚上的数据。如果将使能位SREN置1，则只接收单个字。如果将使能位CREN置1，则会连续接收数据，直到将CREN位清零。如果两个位均被置1，则CREN具有优先权。

设置同步主机接收操作的步骤如下：

- 初始化SPBRG寄存器，设置合适的波特率。
- 通过将SYNC、SPEN和CSRC位置1，使能同步主机串口。
- 确保将CREN和SREN位清零。
- 若需要中断，将中断使能位RCIE置1。
- 若需要接收9位数据，将RX9位置1。
- 若需要单字节接收，将SREN位置1；若需要连续接收，将CREN位置1。
- 当接收完成时中断标志位RCIF将置1，此时如果中断使能位RCIE已置1，还将产生一个中断。

- 读RCSTA 寄存器获取第9 位数据（如果已使能），并判断在接收过程中是否发生了错误。
- 通过读RCREG寄存器来读取接收到的8位数据。
- 如果发生错误，将CREN 位清零以清除错误。
- 若想使用中断，请确保将 PIE0 寄存器中的 GIEH (INTCON<7>) 或 GIEL (INTCON<6>) 置 1。



主机模式同步接收的时序（由SREN位控制）

9.5 同步从机模式

将CSRC (TXSTA<7>) 位清零可进入同步从机模式。此模式与同步主机模式的区别在于移位时钟由 CK 引脚上的外部时钟提供（主机模式中由内部时钟提供）。这使得器件能在任何低功耗模式下发送或接收数据。

9.5.1. 同步从机发送

除了休眠模式以外，同步主机、从机模式的工作方式是完全相同的。如果向TXREG 写入两个字，然后执行SLEEP 指令，则将发生以下事件：

1. 第一个字立即传送到TSR 寄存器进行发送。
2. 第二个字仍保留在TXREG 寄存器中。
3. 不会将标志位TXIF 置1。
4. 当第一个字移出TSR 后，TXREG 寄存器将把第二个字送入TSR，同时将标志位TXIF 置1。
5. 如果中断使能位TXIE 已置1，中断将把器件从休眠状态唤醒。如果使能了全局中断，程序则会跳转到中断向量处执行。

设置同步从机发送操作的步骤如下：

1. 通过将SYNC 和SPEN 位置1 并将CSRC 位清零使能同步从机串口。
2. 将CREN 和SREN 位清零。
3. 若需要中断，将中断使能位TXIE 置1。
4. 若需要发送9 位数据，将TX9 位置1。
5. 将使能位TXEN 置1 使能发送。

6. 如果选择发送9位数据，将第9位数据装入TX9D位。
7. 将数据装入TXREG 寄存器，启动发送。
8. 若想使用中断，请确保将 PIE0 寄存器中的 GIEH (INTCON<7>) 或 GIEL (INTCON<6>) 置 1。

9.5.2. 同步从机接收

除了休眠模式、空闲模式以及在从机模式下忽略SREN位以外，同步主机和同步从机模式的工作方式完全相同。如果在进入休眠或空闲模式前将CREN 位置1，使能接收，那么在该低功耗模式下可以接收到一个数据字。接收到该字后，RSR 寄存器将把数据传输到RCREG寄存器，如果中断使能位RCIE 已置1，产生的中断将把器件从低功耗模式唤醒。如果使能了全局中断，程序则会跳转到中断向量处执行。

设置同步从机接收操作的步骤如下：

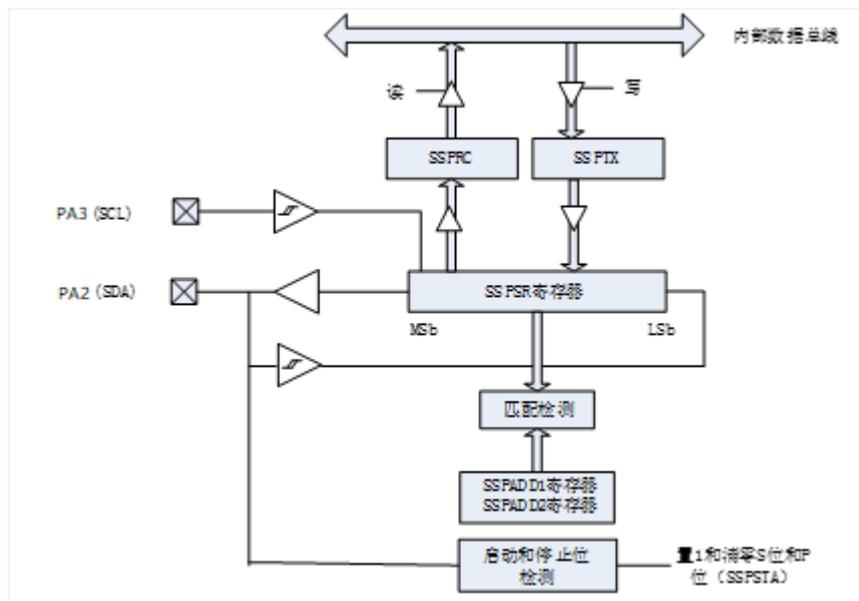
1. 通过将SYNC 和SPEN 位置1 并将CSRC 位清零使能同步从机串口。
2. 若需要中断，将中断使能位RCIE 置1。
3. 若需要接收9 位数据，将RX9 位置1。
4. 将使能位CREN 置1，使能接收。
5. 当接收完成时，RCIF 位将被置1。如果使能位RCIE 已置1，还将产生一个中断。
6. 读RCSTA 寄存器获取第9 位数据（如果已使能），并判断在接收过程中是否发生了错误。
7. 通过读RCREG寄存器来读取接收到的8位数据。
8. 如果发生错误，将CREN 位清零以清除错误。
9. 若想使用中断，请确保将 PIE0 寄存器中的 GIEH (INTCON<7>) 或 GIEL (INTCON<6>) 置 1。

10 I2C 控制器

注：1、使用 I2C 时，请将 CURA 寄存器使能

I2C 模块仅支持从机模式，有两个引脚用于数据传输：

- 串行时钟 (SCL) -- PA3
- 串行数据 (SDA) -- PA2



I²C 模式框图

I2CCON0、I2CCON1 和 I2CSTAT 是 I2C 模式的控制寄存器和状态寄存器。I2CCON0 和 I2CCON1 寄存器是可读写的。I2CSTAT 的低六位是只读的，而高两位是可读写的。

SSPSR 是用来将数据移入或移出的移位寄存器。SSPRC 是接收缓冲寄存器，用于数据字节的读出，I2CTX 是发送缓冲寄存器，用于数据字节的写入。在 I2C 从机模式下配置 SSP 时，SSPAD 寄存器将保存器件的地址。SSPAD 寄存器将保存从器件的地址。在主机模式下配置 I2C 时，I2CADD 的低 7 位保存波特率发生器的重载值。

接收数据时，SSPSR 和 I2CRC、SSPTX 共同构成一个双重缓冲接收器。当 SSPSR 接收到一个完整的字节后，该字节被送入 SSPRC 寄存器，同时将中断标志位 I2C_RCIF 置 1；同时如果 SSPTX 有数据，则可以立即进入发送数据模式，同时将中断标志位 I2C_TXIF 置 1。在发送过程中，SSPTX 与 SSPSR 构成双重缓冲发送。

10.1 I²C 相关寄存器：

10.1.1. I2CACKDLY (I2C 的延时控制寄存器)

地址：0XF4B

Bit	Name	Description	Attribute	Reset
-----	------	-------------	-----------	-------

7	ACKDLYON	1: 允许使能, SCL 输出 0 0: 禁止使能, 软件将 ACKDLY 清零	R/W	0
6:0	ACKDLY[6:0]	I2CACK 延时时间 $T_{ACKDLY} = ACKDLY * 8 / F_{osc}$	R/W	0

10.1.2. I2CTX (I2C 的发送寄存器)

地址: 0XF4A

Bit	Name	Description	Attribute	Reset
7:0	I2CTX[7:0]	与 SSPSR 构成双发送 buffer	R/W	0

10.1.3. I2CRC (I2C 的接收寄存器)

地址: 0XF49

Bit	Name	Description	Attribute	Reset
7:0	I2CRC[7:0]	与 SSPSR 构成双接收 buffer	R	0

10.1.4. I2CADD0 (I2C 的地址寄存器)

地址: 0XF48

Bit	Name	Description	Attribute	Reset
7	ADDOEN	I2C 器件地址 0 的使能 1: 允许使能 0: 禁止使能	R/W	0
6:0	I2CADD0[6:0]	I2C 的器件地址 0	R/W	0

10.1.5. I2CADD1 (I2C 的地址寄存器)

地址: 0XF47

Bit	Name	Description	Attribute	Reset
7	ADD1EN	I2C 器件地址 1 的使能 1: 允许使能 0: 禁止使能	R/W	0
6:0	I2CADD1[6:0]	I2C 的器件地址 1	R/W	0

10.1.6. I2CMASK (I2C 的寄存控制器)

地址: 0XF46

Bit	Name	Description	Attribute	Reset
7:0	I2CMASK[7:0]	I2CADDRMASK[7] 是 I2C 地址掩码的使能 1: I2C 的 7 位器件地址位与 I2CADDRMASK[6:0]	R/W	0

		进行比对 0: I2C 的 7 位器件地址位与 I2CADDRMASK[6:0] 不进行比对 在从机模式下: I2C 的 7 位器件地址位与 I2CADDRMASK[6:0]进 行匹配		
--	--	---	--	--

10.1.7. I2CSTAT (I2C 的控制寄存器)

地址: 0XF45

Bit	Name	Description	Attribute	Reset
7:6	SMP[1:0]	变化率控制位 <u>在主机或从机模式下:</u> 11: 标准速度模式下禁止变化率控制 (100kHz 和 1MHz) 00: 高速模式下使能变化率控制 (400kHz)	R/W	0
5	D/A	数据/地址标志位 1: 表示上一个接收或发送的字节是数据 0: 表示上一个接收或发送的字节是地址	R	0
4	P	停止位 1: 表示上一次检测到停止位 0: 表示上一次未检测到停止位 注: 当复位或 SSPEN 被清零时, 该位被清零。	R	0
3	S	启动位 1: 表示上一次检测到启动位 0: 表示上一次未检测到启动位	R	0
2	R/W	读/写信息位 (仅用于 I ² C 模式) 1: 读 0: 写 注: 该位用来保存在最近一次地址匹配后的 R/W 位信息。 该位仅在从地址匹配开始到下一个启动位、停止位或非 ACK 位之间有效。	R	0
1	BPRX	接收缓冲器满状态位 1: SSPSR 已满 (不包括 ACK 位和停止位) 0: SSPSR 为空 (不包括 ACK 位和停止位)	R	0
0	BPTX	发送缓冲器满状态位 1: SSPBUF 已满 0: SSPBUF 为空	R	0

10.1.8. I2CCON0 (I2C 的控制寄存器)

地址: 0XF44

Bit	Name	Description	Attribute	Reset
7	WCOL	写冲突检测位 1: 正在发送前一个字时, 又有数据写入 I2CTX 寄存器 (必须用软件清零)	R/W	0

		0: 未发生冲突 在接收模式（主机或从机模式）下：在此忽略。		
6	I2COV	接收溢出指示位 在接收模式下： 1: I2CRC 寄存器仍在保存前一个字节时，又接收到一个新的字节（必须用软件清零） 0: 无溢出 在发送模式下：此位被忽略。	R/W	0
5	I2CEN	同步串口使能位 1: 使能串口并将 SDA 和 SCL 引脚配置为串口引脚 0: 禁止串口并将上述引脚配置为 I/O 端口引脚 注：当使能该位时，必须将 SDA 和 SCL 引脚正确地配置为输入引脚或输出引脚	R/W	0
4	I2CSTEN	发送时是否要发 START 信号 1: 不发 START 信号，发送与接收的地址也可以匹配 0: 必须发 START 信号，发送与接收的地址才可以匹配	R/W	0
3:0	Reserved			

10.1.9. I2CCON1（I2C 的控制寄存器）

地址：0XF43

Bit	Name	Description	Attribute	Reset
7	GCEN	广播呼叫使能位（仅用于从机模式） 1: 当 SSPSR 接收到广播呼叫地址（0000h）时允许中断 0: 禁止广播呼叫地址	R/W	0
6	ACKSTAT	从器件发送状态时，接收到的主机的应答状态位 1: 收到来自主器件的 NACK 应答 0: 收到来自主器件的 ACK 应答	R/W	0
5	ACKGET	从器件发送状态时，表示接收到的主机的应答状态位； ACKSTAT 表示接收的具体应答 1: 未收到来自主器件的应答，需要软件清 0 0: 收到来自主器件的应答	R/W	0
4	ACKEN	从器件接收状态时，返回主机应答状态位 1: 表示下一个 I2CRX 接收后，从器件发送应答位 NACK 0: 表示下一个 I2CRX 接收后，从器件发送应答 ACK	R/W	0
3:0	Reserved		R/W	0

注：对于 ACKEN、RCEN、PEN、RSEN 和 SEN 位来说，如果 I²C 模块不处于空闲模式，这些位不能被置 1（或支持后台操作），并且不能对 SSPBUF 进行写操作（或者禁止写 SSPBUF）。

10.1.10. I2CIE 寄存器

地址：0XF42

Bit	Name	Description	Attribute	Reset
7:5	Reserved			
4	I2C_FLOATIE	I2C 总线低时间超过 25ms，则产生中断 1: 允许使能中断 0: 禁止使能中断	R/W	0
3	I2C_STOPIE	I2C STOP 信号产生中断 1: 允许使能中断 0: 禁止使能中断	R/W	0
2	I2C_STARTIE	I2C START 信号产生中断 1: 允许使能中断 0: 禁止使能中断	R/W	0
1	I2C_TXIE	I2C 当 I2CTX 寄存器写入到 SSPSR 时，产生发送中断 1: 允许使能中断 0: 禁止使能中断	R/W	0
0	I2C_RCIE	I2C 当 SSPSR 写入 I2CRC 寄存器时，产生接收中断 1: 允许使能中断 0: 禁止使能中断	R/W	0

10.1.11. I2CIF 寄存器

地址：0XF41

Bit	Name	Description	Attribute	Reset
7:5	Reserved			
4	I2C_FLOATIF	I2C 总线浮空标志 1: 产生中断，软件清 0 0: 没有中断	R/W	0
3	I2C_STOPIF	I2C STOP 中断标志 1: 产生中断，软件清 0 0: 没有中断	R/W	0
2	I2C_STARTIF	I2C START 中断标志 1: 产生中断，软件清 0 0: 没有中断	R/W	0
1	I2C_TXIF	I2C 当 I2CTX 发送中断标志 1: 中断，写 I2CTX 数据，自动清 0 0: 没有中断	R/W	0
0	I2C_RCIF	I2C 接收中断标志 1: 中断，读取 I2CRC 寄存器，自动清 0 0: 没有中断	R/W	0

注：总线浮空中断产生时间为 31.965ms。

总线浮空由内部 FLOAT 检测机制上电计数至 3FFH 后若总线仍拉低则再等待 FLOAT 检测机制计数至 7FFH 后产生总线浮空中断 I2C_FLOATIF，在此期间若总线随时被拉高，则 FLOAT 检测机制重新启动。

FLOAT 检测机制的时钟来源为 OSC32K。

10.1.12. I2CIP 寄存器

地址：0XF40

Bit	Name	Description	Attribute	Reset
7:5	Reserved			
4	I2C_FLOATIP	I2C 总线浮空中断优先级 1: 高优先级 0: 低优先级	R/W	0
3	I2C_STOPIP	I2C STOP 中断优先级 1: 高优先级 0: 低优先级	R/W	0
2	I2C_STARTIP	I2C START 中断优先级 1: 高优先级 0: 低优先级	R/W	0
1	I2C_TXIP	I2C 当 I2CTX 发送中断优先级 1: 高优先级 0: 低优先级	R/W	0
0	I2C_RCIP	I2C 接收中断优先级 1: 高优先级 0: 低优先级	R/W	0

10.2 从机模式

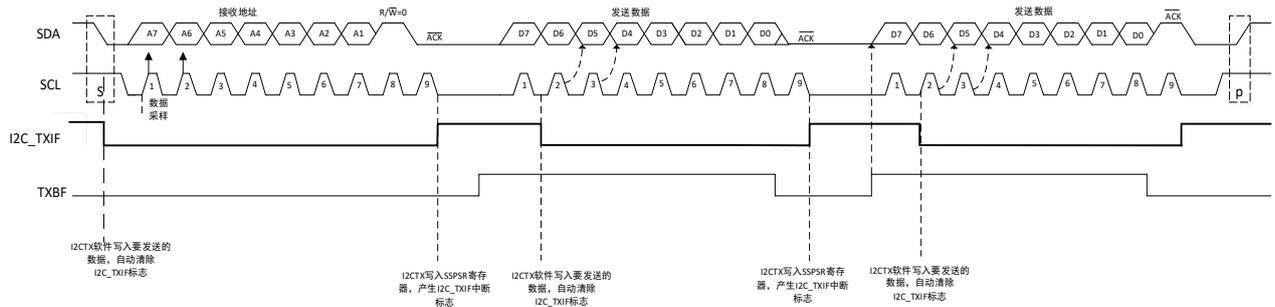
在从机模式下，一旦使能了 I2C 从机模块，它就会等待启动条件出现。启动条件出现后，就会向 SSPSR 寄存器移入 8 位数据。在时钟信号（SCL）的上升沿采样所有的输入位。在第 8 个时钟（SCL）脉冲的下降沿，寄存器 SSPSR<7:1>的值会和 I2CADD0 地址寄存器的值进行比较。如果地址匹配，并且 RCBF 位和 I2COV 位都被清零，会发生下列事件：

- 1) SSPSR 寄存器的值被载入 I2CRC 寄存器。
- 2) 缓冲器满标志位 RCBF 置 1。
- 3) 产生 ACK 脉冲。
- 4) 在第 9 个 SCL 脉冲下降沿，I2C_RCIF 被置 1（如果允许中断，则产生中断）。
- 5) 当地址字节的 R/W 位为 0 且地址匹配时，SSPSTAT 寄存器的 R/W 位清零。接收的地址被装入 I2CRC 寄存器，且 SDA 信号保持低电平（ACK）。

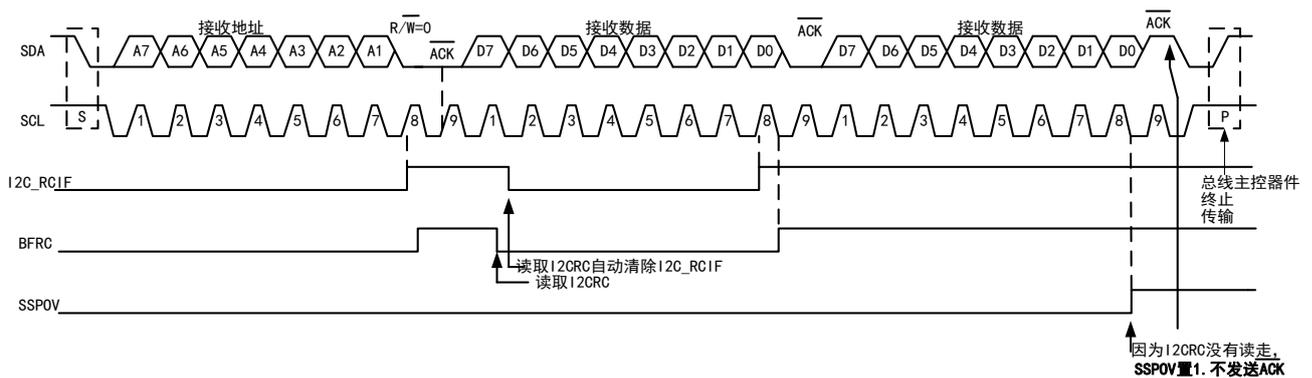
当接收的地址字节的 R/W 位为 1 且地址匹配时，SSPSTAT 寄存器的 R/W 位置 1。接收到的地址被装入 I2CRC 寄存器。ACK 脉冲在第 9 位上发送。从器件可以采用 SSPTX 提前准备好发送数据，并进行发送（可以在 I2C 配置时，写入一个 SSPTX 寄存器），当发送时被装入了 SSPSR 寄存器。8 个数据位在 SCL 时钟输入的下降沿被移出。

来自主接收器的 ACK 脉冲将在第 9 个 SCL 输入脉冲的上升沿被锁存。如果 SDA 数据信号线为高电平（无 ACK），那么表示数据传输已完成。在这种情况下，如果从器件锁存了 ACK，将复位从机逻辑（复位 SSPSTAT 寄存器），同时从器件监视下一个启动位的出现。如果 SDA 线为低电平（ACK），则必须将下一个要发送的数据装入 I2CTX 寄存器。

每个发送数据都会产生一个 I2C_TXIF 中断，并且。I2C_TXIF 位必须用软件清零，I2CSTAT 寄存器用于确定字节的状态。I2C_TXIF 位在第 8 个时钟脉冲的下降沿被置 1。



I2C 从机模式的发送时序 (1) (SEN=0)



I2C 从机模式的接收时序

支持广播呼叫地址

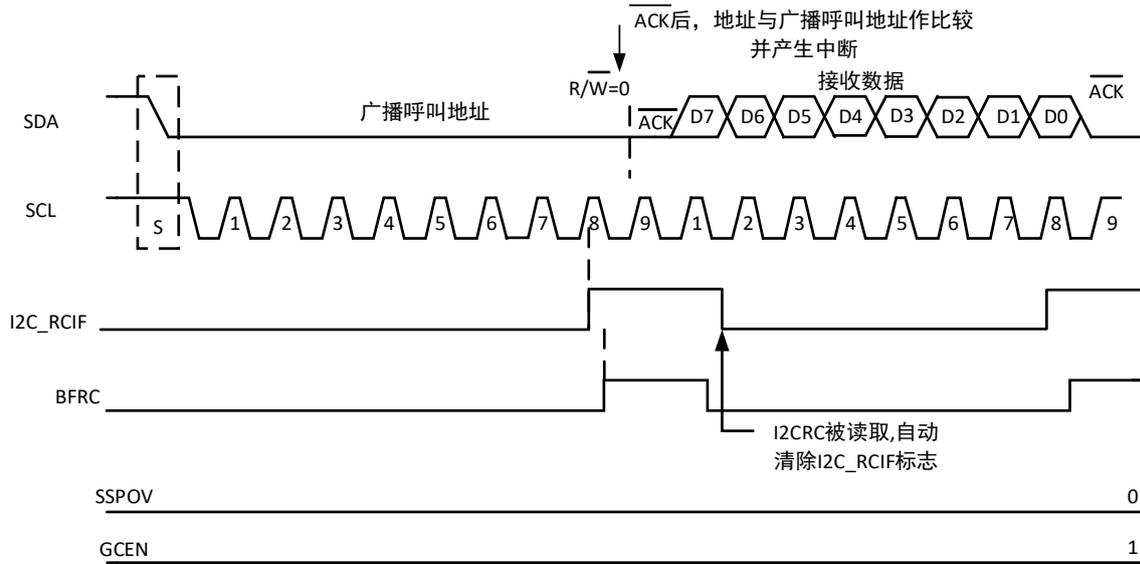
在 I²C 总线的寻址过程中，通常由启动条件后的第一个字节决定主器件将寻址哪个从器件。但广播呼叫地址例外，它能寻址所有器件。当使用这个地址时，理论上所有的器件都应该发送一个应答信号来响应。同时支持两个地址选择，并支持 MASK 屏蔽选择。

广播呼叫地址是由 I²C 协议为特定目的保留的 8 个地址之一。它由全 0 组成，且 R/W=0。

当使能广播呼叫使能位 (GCEN) (I2CCON1<7>置 1) 时，即可识别广播呼叫地址。检测到启动位后，8 位数据会被移入 I2CRC，同时将该地址与 I2C_ADDR0 进行比较。它还会与广播呼叫地址进行比较并用硬件设定。

如果与广播呼叫地址匹配，I2CTX 的值将被传输到 SSPBUF，BF 标志位(第 8 位)置 1，并且在第 8 位 (ACK 位) 的下降沿 I2C_TXIF 中断标志位置 1。

当响应中断时，可以通过读取 I2CRC 的内容来检查中断源。该值可以用于判断是特定器件的地址还是一个广播呼叫地址。



从机模式广播呼叫地址时序

11 中断

系统具备以下中断源:

- INT 管脚的外部中断
- TMR0 溢出中断
- TMR1 溢出中断
- TMR2 溢出中断
- TMR3 溢出中断
- PORTA、PORTB、PORTC 输入改变中断
- CCP 中断
- OP 中断
- ADC 中断
- UART 中断
- CMP 中断

中断允许高优先级总控位 GIEH (INTCON<7>) 和中断允许低优先级总控位 GIEL (INTCON<6>), 能使所有高低优先级的中断被开放 (GIEH=1 或 GIEL=1) 或屏蔽所有中断(GIEH=0 或 GIEL=0), 中断能否启用取决于 IPR 寄存器与 PIE 寄存器, 同时保证 GIEH=1 或者 GIEL=1。

中断发生时 GIEH(GIEL) 位 (在中断发生前 GIEH(GIEL) 位和该中断相关的中断屏蔽位置 1) 被硬件清零从而禁止进一步中断 (区分中断优先级别), 中断标志位在中断允许总控位 GIEH(GIEL)重新置 1 的时候需要被软件清零以防止重复中断。一个中断标志位 (PBIF 除外的) 会被它的中断事件置 1, 而不管与它相关的中断屏蔽位是否启用。通过 IPR, PIR 和 PIE 的对应位来判断中断优先级, 是否发生中断以及中断类型。

11.1 外部中断

外部中断 INT0 管脚上升沿还是下降沿触发由 INT0EDG 位 (T1CON1 寄存器)决定, 当一个有效的跳变发生时标志位 INT0IF 置 1, 如 INT0IE 位清零, 该中断被屏蔽。

在睡眠之前 INT0IE 位已被置 1, INT0 管脚可以作为系统睡眠唤醒条件。在睡眠之前 GIEH(GIEL)位被置 1, CPU 唤醒以后会执行中断服务程序, 否则会运行睡眠以后的下一条指令。

外部中断 INT1 管脚上升沿还是下降沿触发由 INT1EDG 位 (T1CON1 寄存器)决定, 当一个有效的跳变发生时标志位 INT1IF 置 1, 如 INT1IE 位清零, 该中断被屏蔽。

在睡眠之前 INT1IE 位已被置 1, INT1 管脚可以作为系统睡眠唤醒条件。在睡眠之前 GIEH(GIEL)位被置 1, CPU 唤醒以后会执行中断服务程序, 否则会运行睡眠以后的下一条指令。

外部中断 INT2 管脚上升沿还是下降沿触发由 INT2EDG 位 (T1CON1 寄存器)决定, 当一个有效的跳变发生时标志位 INT2IF 置 1, 如 INT2IE 位清零, 该中断被屏蔽。

在睡眠之前 INT2IE 位已被置 1, INT2 管脚可以作为系统睡眠唤醒条件。在睡眠之前 GIEH(GIEL)位

被置 1，CPU 唤醒以后会执行中断服务程序，否则会运行睡眠以后的下一条指令。

11.2 Timer0 中断

TMR0 发生溢出 TMR0=PR0 时 T0IF 标志位置 1, T0IE 位清零，该中断被屏蔽。

11.3 Timer1 中断

当 TMR1ON=1 时, TIMER1 定时器开始从 TMR1H[15:8]与 TMR1L[7:0]组成的 16 位预设值开始计数，在计数的过程中计数值到 0xFFFF 变为 0x0000 时， T1IF 标志位置 1, T1IE 位清零，该中断被屏蔽。

11.4 Timer2 中断

当 TMR2ON=1 时，TIMER2 定时器开始从零计数，在计数的过程中 PR2H[3:0]和 PR2L[7:0]组成的 12 位数值与[TIMER2H:TIMER2L]寄存器的值相等时，TIMER2 定时器也清零。T2IF 标志位置 1， T2IE 位清零，该中断被屏蔽。

11.5 Timer3 中断

当 TMR3ON=1 时，TIMER3 定时器开始从零计数，在计数的过程中 PR3H[3:0]和 PR3L[7:0]组成的 12 位数值与[TIMER3H:TIMER3L]寄存器的值相等时，TIMER3 定时器也清零。T3IF 标志位置 1， T3IE 位清零，该中断被屏蔽。

11.6 PortA 输入改变中断

输入改变中断触发时 PA<5:0> PAIF 标志位置 1 (PIR2<5>). PAIE 位(PIE2<5>)清零，该中断被屏蔽。PAIE 在睡眠之前置 1， Port A 输入脚改变中断也可以作为睡眠唤醒条件。在睡眠之前 GIE 位已被置 1 机器唤醒以后会执行中断服务程序，否则会运行睡眠以后的下一条指令。

11.7 PortB 输入改变中断

输入改变中断触发时 PB<5:0> PBIF 标志位置 1 (PIR2<6>). PBIE 位(PIE2<6>)清零，该中断被屏蔽。PBIE 在睡眠之前置 1， Port B 输入脚改变中断也可以作为睡眠唤醒条件。在睡眠之前 GIE 位已被置 1 机器唤醒以后会执行中断服务程序，否则会运行睡眠以后的下一条指令。

11.8 PortC 输入改变中断

输入改变中断触发时 PC<5:0> PCIF 标志位置 1 (PIR2<7>). PCIE 位(PIE2<7>)清零，该中断被屏蔽。PCIE 在睡眠之前置 1， PortC 输入脚改变中断也可以作为睡眠唤醒条件。在睡眠之前 GIE 位已被置 1 机器唤醒以后会执行中断服务程序，否则会运行睡眠以后的下一条指令。

11.9 低电压、高电压中断

提供 14 组电压选择，如下所示。当 $LVD\overline{M}[1:0]=2'b01$ 时，系统 VCC 电压低于设定的 LVDV 电压值，LVDIF 位置为 1。LVDIE 位($PIE1<5>$)清零，该中断被屏蔽。

PCON 寄存器

地址：0XFF1

Bit	Name	Description	Attribute	Reset
7	Reserved			
6	PBOST	LVD \overline{M} =11 时，PBO 输出值	R/W	0
5:4	LVD $\overline{M}[1:0]$	电压比较中断 00: 禁止电压比较器 01: VCC 低于阈值电压产生中断 10: VCC 高于阈值电压产生中断 11: VCC 高于阈值电压产生中断，并强制 PBO 输出为 PBOST 值	R/W	00
3:0	LVD[3:0]	LVD3:0 VCC 电压阈值选择 0: 保留，勿配置 1: 保留，勿配置 2: 2.2V 3: 2.4V 4: 2.6V 5: 2.7V 6: 2.9V 7: 3.0V 8: 3.1V 9: 3.3V 10: 3.6V 11: 3.7V 12: 3.8V 13: 4.1V 14 : 4.2V 15 : 4.3V	R/W	0000

11.10 运放中断

当 OP0OUT 从 0 变为 1 时，OP0IF 置位($OP0IF=1$)。

当 OP0OUT 从 1 变为 0 时，OP0IF 置位($OP0IF=1$)，需要额外配置 OP0POS 置位($OP0POS=1$)。

读取 OP0CON1 寄存器后，当 OP0OUT 输出发生改变，OP0IF 置位($OP0IF=1$)。OP0IE 位($PIE3<3>$)清零，OP0 中断被屏蔽。

当 OP1OUT 从 0 变为 1 时，OP1IF 置位($OP1IF=1$)。

当 OP1OUT 从 1 变为 0 时, OP1IF 置位(OP1IF==1), 需要额外配置 OP1POS 置位(OP1POS==1)。

读取 OP1CON1 寄存器后, 当 OP1OUT 输出发生改变, OP1IF 置位(OP1IF==1)。OP1IE 位(PIE3<4>)清零, OP1 中断被屏蔽。

11.11 比较器中断

当 CMP0OUT 从 0 变为 1 时, CMP0IF 置位(CMP0IF==1)。

当 CMP0OUT 从 1 变为 0 时, CMP0IF 置位(CMP0IF==1), 需要额外配置 CMP0POS 置位(CMP0POS==1)。

读取 CMP0CON 寄存器后, 当 CMP0OUT 输出发生改变, CMP0IF 置位(CMP0IF==1)。CMP0IE 位(PIE2<0>)清零, CMP0 中断被屏蔽。

当 CMP1OUT 从 0 变为 1 时, CMP1IF 置位(CMP1IF==1)。

当 CMP1OUT 从 1 变为 0 时, CMP1IF 置位(CMP1IF==1), 需要额外配置 CMP1POS 置位(CMP1POS==1)。

读取 CMP1CON 寄存器后, 当 CMP1OUT 输出发生改变, CMP1IF 置位(CMP1IF==1)。CMP1IE 位(PIE2<1>)清零, CMP1 中断被屏蔽。

11.12 ADC 中断

当 GO 从 1 变为 0 时, ADIF 置位为 1。ADIE 位(PIE0<4>)清零, 该中断被屏蔽

11.13 中断的相关寄存器

11.13.1. INTCON 寄存器

地址:0XFF2

Bit	Name	Description	Attribute	Reset
7	GIE/GIEH	全局中断使能位 当 IPEN=1 时: 1: 允许所有高优先级的外设中断 0: 禁止所有中断	R/W	0
6	PEIE/GIEL	外设中断使能位 当 IPEN=1 时: 1: 允许所有低优先级的外设中断 0: 禁止所有低优先级的外设中断	R/W	0
5:0	Reserved			

11.13.2. IPR1 寄存器

地址:0XFA3

Bit	Name	Description	Attribute	Reset
7	Reserved			

6	RCIP	UART 接收中断优先级 1: 高优先级 0: 低优先级	R/W	1
5	TXIP	UART 发送中断优先级 1: 高优先级 0: 低优先级	R/W	1
4	CCPIP	CCP 中断优先级 1: 高优先级 0: 低优先级	R/W	1
3	TMR3 IP	TMR3 中断优先级 1: 高优先级 0: 低优先级	R/W	1
2	TMR2IP	TMR2 中断优先级 1: 高优先级 0: 低优先级	R/W	1
1	TMR1IP	TMR1 中断优先级 1: 高优先级 0: 低优先级	R/W	1
0	TMROIP	TMRO 中断优先级 1: 高优先级 0: 低优先级	R/W	1

11.13.3. PIR1 寄存器

地址:0XFA2

Bit	Name	Description	Attribute	Reset
7	Reserved			
6	RCIF	UART 接收中断标志, 发生中断置 1, 软件设置清零	R/W	0
5	TXIF	UART 发送中断标志, 发生中断置 1, 软件设置清零	R/W	0
4	CCPIF	CCP 中断标志, 发生中断置 1, 软件设置清零	R/W	0
3	TMR3IF	溢出中断标志, 发生 Timer3 溢出中断置 1, 软件设置清零	R/W	0
2	TMR2IF	溢出中断标志, 发生 Timer2 溢出中断置 1, 软件设置清零	R/W	0
1	TMR1IF	溢出中断标志, 发生 Timer1 溢出中断置 1, 软件设置清零	R/W	0
0	TMROIF	溢出中断标志, 发生 Timer0 溢出中断置 1, 软件设置清零	R/W	0

11.13.4. PIE1 寄存器

地址:0XFA1

Bit	Name	Description	Attribute	Reset
7	Reserved			
6	RCIE	UART 接收中断允许位 1: 使能外部中断 0: 禁止外部中断.	R/W	0
5	TXIE	UART 发送中断允许位 1: 使能外部中断 0: 禁止外部中断.	R/W	0

4	CCPIE	CCP 中断允许位。 1: 使能外部中断 0: 禁止外部中断。	R/W	0
3	TMR3IE	Timer3 溢出中断允许位。 1: 使能 Time3 溢出中断 0: 禁止 Timer3 溢出中断	R/W	0
2	TMR2IE	Timer2 溢出中断允许位。 1: 使能 Timer2 溢出中断 0: 禁止 Timer2 溢出中断	R/W	0
1	TMR1IE	Timer1 溢出中断允许位。 1: 使能 Timer1 溢出中断 0: 禁止 Timer1 溢出中断	R/W	0
0	TMROIE	Timer0 溢出中断允许位。 1: 使能 Timer0 溢出中断 0: 禁止 Timer0 溢出中断	R/W	0

11.13.5. IPR2 寄存器

地址:0XFA0

Bit	Name	Description	Attribute	Reset
7	PCIP	PortC 输入改变中断优先级 1: 高优先级 0: 低优先级	R/W	1
6	PBIP	PortB 输入改变中断优先级 1: 高优先级 0: 低优先级	R/W	1
5	PAIP	PortA 输入改变中断优先级 1: 高优先级 0: 低优先级	R/W	1
4	INT2IP	外部中断 2 优先级 1: 高优先级 0: 低优先级	R/W	1
3	INT1IP	外部中断 1 优先级 1: 高优先级 0: 低优先级	R/W	1
2	INT0IP	外部中断 0 优先级 1: 高优先级 0: 低优先级	R/W	1
1	CMP1IP	CMP1 中断优先级 1: 高优先级 0: 低优先级	R/W	1
0	CMP0IP	CMP0 中断优先级 1: 高优先级	R/W	1

		0: 低优先级		
--	--	---------	--	--

11.13.6. PIR2 寄存器

地址:0XF9F

Bit	Name	Description	Attribute	Reset
7	PCIF	PortC 输入改变中断标志, PortC 输入改变时置 1, 软件设置清零	R/W	0
6	PBIF	PortB 输入改变中断标志, PortB 输入改变时置 1, 软件设置清零	R/W	0
5	PAIF	PortA 输入改变中断标志, PortA 输入改变时置 1, 软件设置清零	R/W	0
4	INT2IF	INT2 外部中断标志位 1: 产生 INT2 外部中断标准位 0: 未产生 INT2 外部中断标志位	R/W	0
3	INT1IF	INT1 外部中断标志位 1: 产生 INT1 外部中断标准位 0: 未产生 INT1 外部中断标志位	R/W	0
2	INT0IF	INT0 外部中断标志位 1: 产生 INT0 外部中断标准位 0: 未产生 INT0 外部中断标志位	R/W	0
1	CMP1IF	CMP1 中断标志位 1: 产生 CMP1 中断标志位 0: 未产生 CMP1 中断标志位	R/W	0
0	CMP0IF	CMP0 中断标志位 1: 产生 CMP0 中断标志位 0: 未产生 CMP0 中断标志位	R/W	0

11.13.7. PIE2 寄存器

地址:0XF9E

Bit	Name	Description	Attribute	Reset
7	PCIE	PortC 输入改变中断允许位 1: 使能 PortC 输入改变中断 0: 禁止 PortC 输入改变中断	R/W	0
6	PBIE	PortB 输入改变中断允许位 1: 使能 PortB 输入改变中断 0: 禁止 PortB 输入改变中断	R/W	0
5	PAIE	PortA 输入改变中断允许位 1: 使能 PortA 输入改变中断 0: 禁止 PortA 输入改变中断	R/W	0
4	INT2IE	外部中断 2 允许位	R/W	0

		1: 使能 INT2 中断 0: 禁止 INT2 中断		
3	INT1IE	外部中断 1 允许位 1: 使能 INT1 中断 0: 禁止 INT1 中断	R/W	0
2	INT0IE	外部中断 0 允许位 1: 使能 INT0 中断 0: 禁止 INT0 中断	R/W	0
1	CMP1IE	CMP1 中断允许位 1: 使能 CMP1 中断 0: 禁止 CMP1 中断	R/W	0
0	CMP0IE	CMP0 中断允许位 1: 使能 CMP0 中断 0: 禁止 CMP0 中断	R/W	0

11.13.8. IPR3 寄存器

地址:0XF9D

Bit	Name	Description	Attribute	Reset
7	ADCMP1IP	ADCMP1 中断优先级 1: 高优先级 0: 低优先级	R/W	1
6	ADCMP0IP	ADCMP0 中断优先级 1: 高优先级 0: 低优先级	R/W	1
5	ADIP	ADC 中断优先级 1: 高优先级 0: 低优先级	R/W	1
4	OP1IP	OP1 中断优先级 1: 高优先级 0: 低优先级	R/W	1
3	OPOIP	OPO 中断优先级 1: 高优先级 0: 低优先级	R/W	1
2	EEIP	EEPROM 中断优先级 1: 高优先级 0: 低优先级	R/W	1
1	OSFIP	OSF 中断优先级 1: 高优先级 0: 低优先级	R/W	1
0	LVDIP	LVD 中断优先级 1: 高优先级 0: 低优先级	R/W	1

11.13.9. PIR3 寄存器

地址:0XF9C

Bit	Name	Description	Attribute	Reset
7	ADCMP1IF	ADC 采集值比较器 1 中断标志位, 软件设置清零	R/W	0
6	ADCMPOIF	ADC 采集值比较器 0 中断标志位, 软件设置清零	R/W	0
5	ADIF	A/D 转换产生中断标志, 软件设置清零	R/W	0
4	OP1IF	运放中断标志, 当 OP1OUT 从 0 变为 1 时, 标志位置 1, 软件设置清零	R/W	0
3	OPOIF	运放中断标志, 当 OPOUT 从 0 变为 1 时, 标志位置 1, 软件设置清零	R/W	0
2	EEIF	EEPROM 写操作中断标志, 软件设置清零	R/W	0
1	OSFIF	时钟检测中断标志, 软件设置清零	R/W	0
0	LVDIF	电压检测中断标志, 标志位置 1, 软件设置清零;	R/W	0

11.13.10. PIE3 寄存器

地址:0XF9B

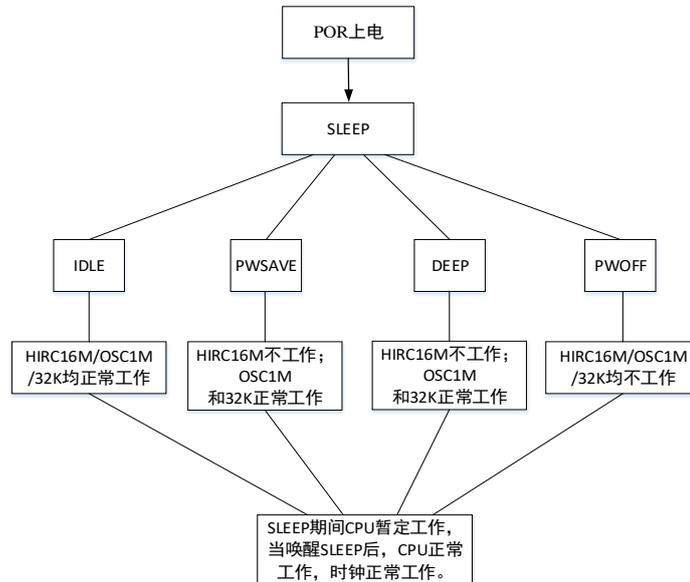
Bit	Name	Description	Attribute	Reset
7	ADCMP1IE	ADCMP1 中断允许位 1: 使能 ADCMP1 中断 0: 禁止 ADCMP1 中断	W	0
6	ADCMPOIE	ADCMPO 中断允许位 1: 使能 ADCMPO 中断 0: 禁止 ADCMPO 中断	W	0
5	ADIE	ADC 中断允许位 1: 使能 ADC 中断 0: 禁止 ADC 中断	R/W	0
4	OP1IE	OP1 中断允许位 1: 使能 OP1 中断 0: 禁止 OP1 中断	R/W	0
3	OPOIE	OPO 中断允许位 1: 使能 OPO 中断 0: 禁止 OPO 中断	R/W	0
2	EEIE	EEPROM 写数据中断允许位 1: 使能 EEPROM 中断 0: 禁止 EEPROM 中断	R/W	0
1	OSFIE	OSF 时钟检测中断允许位 1: 使能 OSF 中断 0: 禁止 OSF 中断	R/W	0

0	LVDIE	LVD 电压检测中断允许位 1: 使能 LVD 中断 0: 禁止 LVD 中断	R/W	0
---	-------	---	-----	---

12 省电模式 (SLEEP)

拥有四种睡眠模式：(IDLE、PWSAVE、DEEPPWSAVE、PWOFF)

- 000: IDLE 模式，CPU 进入睡眠，外设继续运行，保存 SRAM 数据，唤醒后打开中断会进入中断，否则顺序执行；
- 001: PWSAVE 模式，进入睡眠，保存 SRAM 数据，唤醒后打开中断会进入中断，否则顺序执行；
- 010: DEEPPWSAVE 模式，进入睡眠，保存 SRAM 数据，唤醒后打开中断会进入中断，否则顺序执行
- 011: PWOFF 模式，进入深度睡眠，保存 SRAM 数据，唤醒后打开中断会进入中断，否则顺序执行。
- SLEEP 模式下时钟工作图：



SLEEP模式下时钟工作图

- 注：1.SLEEP 语句之后需加一条 NOP 指令；
- 2.进入睡眠之前，请将唤醒源再次配置使能。
- 3.进入睡眠前，请定义全局变量读取所有 I/O 口进行锁存。

12.1 睡眠唤醒

在睡眠状态下，四种模式，单片机能通过以下方式唤醒：如下图所示

模块	睡眠模式			
	IDLE	PWSAVE	PWOFF	DEEPPWSAVE
CCP	√			

TIMER0	√			
TIMER1	√	√		√
TIMER2	√			
TIMER3	√			
WDT	√	√		√
RST	√	√	√	√
INT	√	√	√	√
IO	√	√	√	√
LVD	√	√		√

四种模式下唤醒后，HIRC16M/OSC1.024M/OSC32K 均能正常工作。在 SLEEP 模式期间，CPU 暂停工作，唤醒后 CPU 正常工作。**注：√表示可唤醒的方式**

在睡眠状态下，四种模式可以工作的模块如下图所示

模块					
	ACTIVE	IDLE	POWER SAVE	POWER OFF	DEEPPower SAVE
OSC16M	√	√			
OSC1.024M	√	√	√		√
CPU	√				
SRAM	√	√	√	√	√
Timer0/2/3	√	√			
Timer1	√	√	√		√
CCP	√	√			
WDT	√	√	√		√
External Interrupt	√	√	√	√	√
PAIF/PBIF/PCIF	√	√	√	√	√
BGR	√	√	√		√
LVD	√	√	√		√
ADC	√	√	√OSC1.024M		√OSC1.024M
DAC	√	√			
POR	√	√	√	√	√
IO	√	√	√	√	√

RESET	√	√	√	√	√
-------	---	---	---	---	---

外部的 RSTB 管脚和看门狗溢出都能使机器复位. 通过查看 /PD 和/TO 位可以检测机器是哪种复位, /PD 位置 1 为上电复位, 置 0 为执行 SLEEP, /TO 位置 0 为看门狗溢出复位. 机器通过中断唤醒, 该中断屏蔽位置 1, 中断唤醒不管 GIE 是否置 1. 当 GIE 位被清零, 机器唤醒以后执行 SLEEP 指令以后的指令; 当 GIE 位被置 1, 机器唤醒以后跳转到中断复位地址 (008h)。

12. 1. 1. SMCR (状态控制寄存器)

地址: 0XF65

Bit	Name	Description	Attribute	Reset
7:4	Reserved			
3:1	SM[2:0]	休眠模式选择 000: IDLE 模式, 空闲模式 001: POWER SAVE 模式, 低功耗模式 010: DEEPPPOWER 模式, 深度睡眠模式, 可以保持 MCU 和 SRAM 数据, 唤醒后继续运行 011: POWER OFF 模式, 电源关闭模式, 保持 MCU 和 SRAM 数据, 唤醒后继续运行	R/W	000
0	SE	休眠模式使能位 1: 使能休眠模式, 硬件自动清零 0: 禁止休眠模式	R/W	0

实例代码:

```
#include "AD008.h"
__CONFIG(1, _FINTOSC_DIV1_1L & _LVREN_ON_1L &
_LVD20V_1L
);
__CONFIG(2, _CRYLEN_OFF_2L & _CRYHEN_OFF_2L & _IDIS_ON_2L & _CKSUM_OFF_2L &
_CBP_OFF_2L & _RESETE_OFF_2L & _RSEL_LOW_2L & _STV_ON_2L );
__CONFIG(3, _PSUT_1_3L & _FCPU_2T_3L & _WDTEN_OFF_3L &
_TWDT_000_3L
);
__CONFIG(4, _OSCM_HIRC_16M_4L & _FCMEN_OFF_4L & _IESO_OFF_4L &
_ADSP_OFF_4L
);

#define SLEEP_PWIDLE() SMCR = 0X01; SLEEP();NOP()
#define SLEEP_PWSAVE() SMCR = 0X03; SLEEP();NOP()
#define SLEEP_DEEPPWSAVE() SMCR = 0X05; SLEEP();NOP()
#define SLEEP_PWOFF() SMCR = 0X07; SLEEP();NOP()
unsigned char t0;
void init()
{
    TMR1H = 0xff;
}
```

```
TMR1L = 0x55;
T1CON0 = 0x83; //T1 32k 1:1
TMR1IE = 1;
TMR1IF = 0;
}

void main(void)
{
    init();
    t0 = 0;
    GIE = 1;
    while(1){
        SLEEP_DEEPPWSAVE();
        if(t0 == 12)
            PORTB = 0X7F;
    }
    while(1);
}

void interrupt ISR_h(void)
{
    if(TMR1IF){
        TMR1IF = 0;
        t0++;
    }
}
```

13 固定参考电压 (FVR)

固定参考电压或 FVR 是稳定的参考电压，独立于 VDD，可选 0.3V、1.0V、1.1V、1.2V 或者 2.0V、2.2 V、2.4V；。可配置 FVR 的输出为以下各项提供参考电压：

- ADC 参考电压和通道输入电压
- 比较器 CMP0 和 CMP1 正端参考电压
- 运放 OP0 和运放 OP1 的输入源
- DAC 参考电压

注：本章涉及到的具体电压均为内部参考电压产生，无需单独开关；

13.1 FVRCON 寄存器

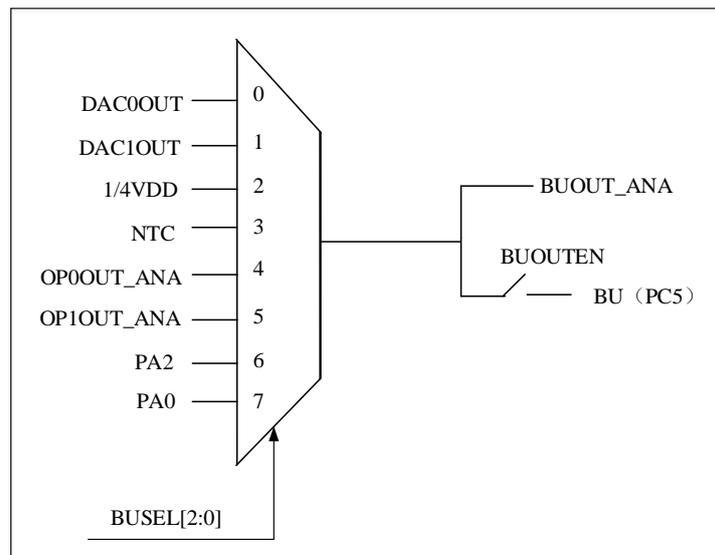
地址：0XF72

Bit	Name	Description	Attribute	Reset
7	FVROUTEN	FVR 输出 IO 使能. 通过 PC4 输出 1: 使能 FVR 输出 0: 禁止 FVR 输出	R/W	0
6	FVRPGA	FVR 电压放大 1 倍选择位 1: 使能 FVR 电压放大 0: 禁止 FVR 电压放大	R/W	0
5	FVREN	FVR 使能 1: 使能 FVR 0: 禁止 FVR	R/W	0
4:0	FVR_SEL[3:0]	FVR 电压选择位 00000: 1.0V 00001: VBG (1.25V) 00010: NTC 00011: OPOUT_ANA 00100: OP1OUT_ANA 00101: DAC0OUT 00110: DAC1OUT 00111: OLDO 01000: LIRC_LDO_OUT 01001: PA4 01010: PB2 01011: PC2 01100: 0.4V 01101: 0.6V 01110: 0.8V 01111: 1.1V	R/W	00000

14 多通道输入模拟缓冲器 (BUFFER UNIT)

多输入模拟通道 BUFFER UNIT (BU) 是稳定的参考电压, 可配置 BU 的输出为以下各项提供参考电压:

- ADC 通道输入电压
- 比较器 CMP0 和 CMP1 正端参考电压
- DAC0 和 DAC1 参考电压



注: 当通道选择 1/4VDD 时, 须同时使能 LVR(_LVREN_ON_1L) 和 LVD (PCON 中 LVDM[1:0]), 此操作不影响 LVD 和 LVR 相关功能。

14.1 BUCON 寄存器

地址: 0XF6D

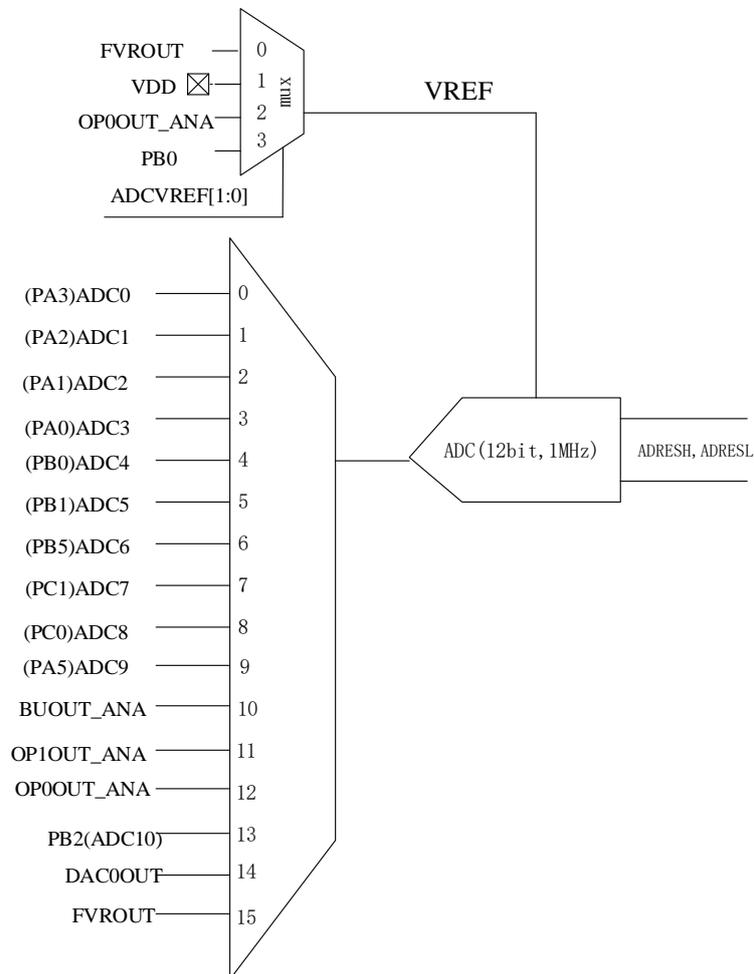
Bit	Name	Description	Attribute	Reset
7:6	Reserved			
5	BUOUTEN	BU 输出 IO 使能 1: 使能 BU 输出 0: 禁止 BU 输出	R/W	0
4	BUEN	BU 使能 1: 使能 BU 0: 禁止 BU	R/W	0
3	Reserved			
2:0	BUSEL[2:0]	BU 输入通道选择位 000: DAC0OUT 001: DAC1OUT	R/W	0

		010: 1/4VDD 011: NTC 100: OPOOUT_ANA 101: OP1OUT_ANA 110: PA2 111: PA0		
--	--	---	--	--

15 模数转换器 (ADC)

AD008 包含一个 16 通道输入的 12 位 ADC，能够将一个模拟输入转换成 12 位数字信号。在根据需要配置好 A/D 模块之后，必须在转换开始之前对选定的通道进行采样。当采集启动延时 ACQT 计数完成后，硬件启动 A/D 转换。A/D 转换完成之后，转换结果被装入 ADRESH:ADRESL 寄存器对，GO/DONE 位被硬件清零且 A/D 中断标志位 ADIF 被置 1。

处理器进入低功耗时间，可以使能 ADC 自动采集功能。当 ACEN 使能时，ADC 使能自动采集功能，ADCMP0H:ADCMP0L 和 ADCMP1H:ADCMP1L 组成两个比较值，根据 ADCMPMODE 可以进行采样值比较功能。ADCMPMODE 为 0 时，当 ADRESH:ADRESL 大于 ADCMP1H:ADCMP1L 或者小于 ADCMP0H:ADCMP0L 时，且连续计数次数达到预设值，则产生触发中断标志 ADCMP0IF，如果 ADCMP0IE 打开，会唤醒处理器。



A/D 结构图

执行 A/D 转换步骤：

配置 A/D 模块

- 选择参考电压（通过 ADCON1[7:6]寄存器）
- 选择 A/D 输入通道（通过 ADCON0[5:2]寄存器）

- 选择 A/D 采集时间（通过 ADCON1[5:3]寄存器）
 - 选择 A/D 转换时间（通过 ADCON1[2:0]寄存器）
 - 使能 A/D 模块（通过 ADCON0[0]寄存器）
- 1) 需要时，配置 A/D 中断
 - 清零 ADIF 位
 - 将 ADIE 位置 1
 - 将 GIE 位置 1
 - 2) 如果需要，需等待所需的采集时间。
 - 3) 启动转换：
 - 将 GO/DONE 位置 1（ADCON0[1]）
 - 4) 等待 A/D 转换完成，通过以下两种方式之一判断转换是否完成：
 - 查询 GO/DONE 位是否被清零
 - 等待 A/D 中断
 - 5) 读取 A/D 结果寄存器（ADRESH:ADRESL），需要时将 ADIF 位清零。
 - 6) 如需再次进行 A/D 转换，返回步骤 1 或者步骤 2。

注：当等待 AD 转换时，建议增加超时处理，如下图：

```

/*****
* Description   : ADC采集
* Input        : None
* Output       : None
* Return       : None
* Attention    : 默认采集FVR 1.0V ->反推VCC电压
*****/
void ADC_Exce(void)
{
    unsigned short timeout;

    GO = 1;      //start ADC

    //等待AD转换结束
    timeout = 0;
    while(GO)
    {
        CLRMDT();

        timeout++;
        if(timeout > 2000)      //超时时间需按照ADC采集时间适当更改
        {
            ADON = 0;
            GO = 0;
            ADON = 1;
            GO = 1;
            break;
        }
    }
}
    
```

15.1 ADCMP0H（AD 比较寄存器 0 高字节）

地址：0XFAB

Bit	Name	Description	Attribute	Reset
7:0	ADCMP0H	AD 比较寄存器 0 的高 8 位	R/W	0

15.2 ADCMP1H(AD 比较寄存器 1 高字节)

地址：0XFAA

Bit	Name	Description	Attribute	Reset
7:0	ADCMP1H	AD 比较寄存器 1 的高 8 位	R/W	0

15.3 ADCMP01L (AD 比较寄存器 0 和 1 低字节)

地址: 0XFA9

Bit	Name	Description	Attribute	Reset
7:4	ADCMP1L	AD 比较寄存器 1 的低 4 位	R/W	0
3:0	ADCMP0L	AD 比较寄存器 0 的低 4 位	R/W	0

15.4 ADRESH (AD 转换结果的高四位)

地址: 0XFA8

Bit	Name	Description	Attribute	Reset
7:4	Reserved			
3:0	ADRESH	AD 转换结果的高四位	R/W	0

15.5 ADRESL (AD 转换结果的低四位)

地址: 0XFA7

Bit	Name	Description	Attribute	Reset
7:0	ADRESL	AD 转换结果的低 8 位	R/W	0

15.6 ADCON0 (ADC 控制寄存器 0)

地址: 0XFA6

Bit	Name	Description	Attribute	Reset
7	ADFM	AD 结果对齐标志 1: 左对齐 ADRESH = adc 【11:4】 左对齐 ADRESL = {adc 【3:0】, 4' b0} 0: 右对齐 ADRESH = {4' b0, adc 【11:8】 } 右对齐 ADRESL = adc 【7:0】	R/W	0
6	ADCMPMODE	AD 采集值的比较模式; 当 ACEN 为 0 时, 支持采集电压比较功能, 并可以产生如下描述的对应中断标志; 当 ACEN 为 1 时, 只生成 ADCMPOIF 中断标志; 0: 采集值大于 ADCMP1H 产生中断 ADCMP1IF 或者小于 ADCMPOH 产生中断 ADCMPOIF; 1: 采集值小于 ADCMP1H 且大于 ADCMPOH, 产生中断 ADCMPIF0;	R/W	0

5:2	CHS[3:0]	CHS3:CHS0 - 模拟通道选择位 0000: ADC 通道 0 (PA3) 0001: ADC 通道 1 (PA2) 0010: ADC 通道 2 (PA1) 0011: ADC 通道 3 (PA0) 0100: ADC 通道 4 (PBO) 0101: ADC 通道 5 (PB1) 0110: ADC 通道 6 (PB5) 0111: ADC 通道 7 (PC1) 1000: ADC 通道 8 (PC0) 1001: ADC 通道 9 (PA5) 1010: ADC 通道 10 (BUOUT_ANA) 1011: ADC 通道 11 (OP1OUT_ANA) 1100: ADC 通道 12 (OPOOUT_ANA) 1101: ADC 通道 13 (PB2) 1110: ADC 通道 14 (DAC0OUT) 1111: ADC 通道 15 (FVROUT)	R/W	0
1	GO/DONE	GO/DONG - A/D 转换状态位 当 ADON=1 时: 1: A/D 转换正在进行 0: A/D 空闲	R/W	0
0	ADON	ADON - A/D 模拟使能位 1: 使能 A/D 转换器模块 0: 禁止 A/D 转换器模块	R/W	0

15.7 ADCON1 (ADC 控制寄存器 1)

地址: 0XFA5

Bit	Name	Description	Attribute	Reset
7:6	ADVREF[1:0]	A/D 参考电压选择位 00: FVROUT 01: VCC 10: OPOOUT_ANA 11: PBO	R/W	0
5:3	ACQT[2:0]	A/D 延时采集时间选择 111: 15 TAD 110: 13 TAD 101: 11TAD 100: 9 TAD 011: 7 TAD 010: 5TAD 001: 3 TAD 000: 1TAD	R/W	0

2:0	ADCS[2:0]	A/D 转换时钟选择位 111: 内部 1.024M OSC 110: FOSC/512(ADSP=0), FOSC/64(ADSP=1) 101: FOSC/128(ADSP=0), FOSC/16(ADSP=1) 100: FOSC/32(ADSP=0), FOSC/4(ADSP=1) 011: 内部 32K OSC 010: FOSC/256(ADSP=0), FOSC/32(ADSP=1) 001: FOSC/64(ADSP=0), FOSC/8(ADSP=1) 000: FOSC/16(ADSP=0), FOSC/2(ADSP=1)	R/W	0
-----	-----------	--	-----	---

15.8 ADCON2 (ADC 控制寄存器 2)

地址: 0XFA4

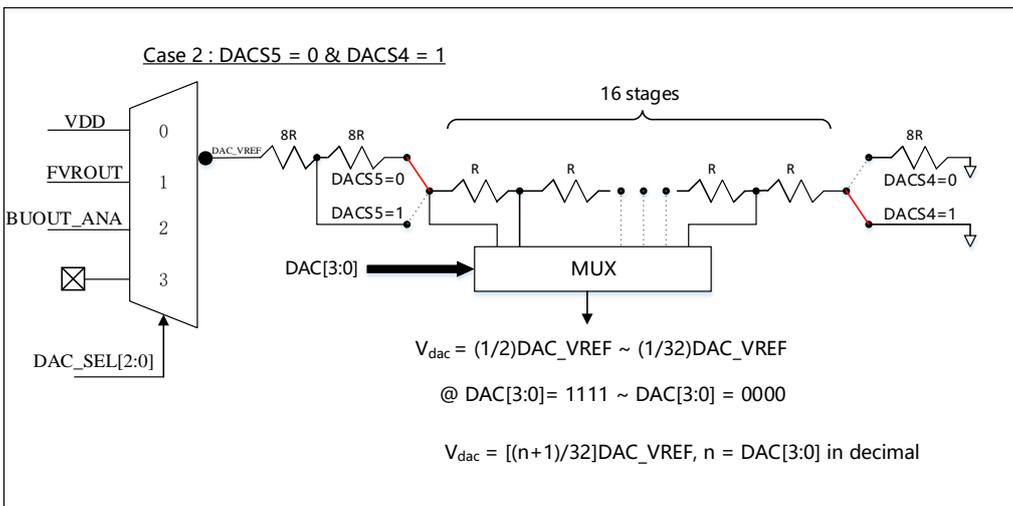
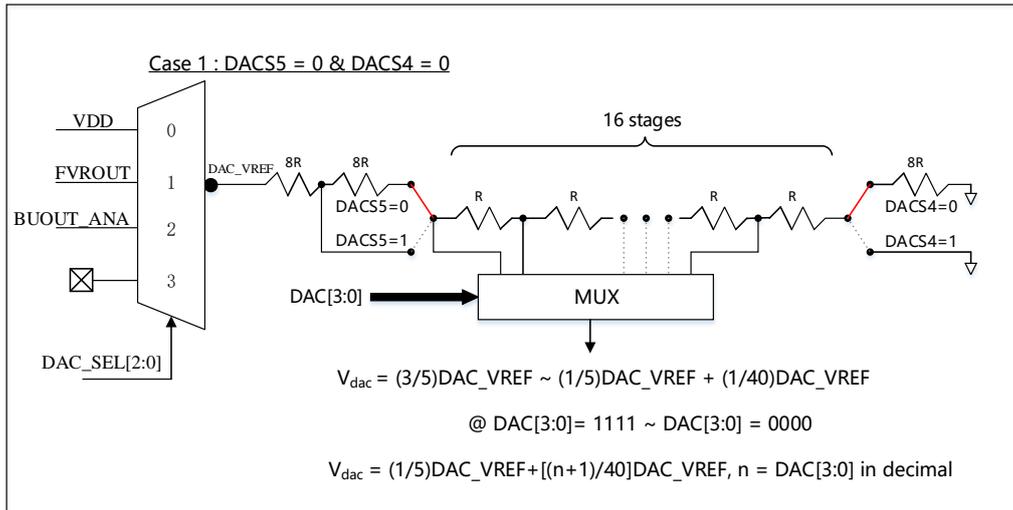
Bit	Name	Description	Attribute	Reset
7:4	ACQT[6:3]	当 ACEN 使能时, 与 ADCON1 的 ACQT[2:0] 合并, 共同扩展采集延迟选择, 延时启动 ADC 采集计算公式如下: $T = ACQT[6:0] * ADC_CLK$	R/W	0000
3:2	ACN[1:0]	根据 ADCON1 的 ADCMPMODE 设定, 当 ADC 采样值符合条件时, ACN 计数器加一, 当内部计数器值与 ACN[1:0] 相等时间, 触发比较中断标志 ADCMPOIF 00: 1 次 01: 2 次 10: 4 次 11: 8 次	R/W	0
1	ACFVR	当 ACEN 使能时间, 并且有事件触发 ADC 采集, 同时会使能 FVR 启动, FVR 启动稳定时间需要 100uS, 当 ADC 采集完成后, 硬件自动关闭 FVR, 请设定好 ACQT[6:0] 延时值, 配合 ADC 正确采集 1: 使能 FVR 自动工作 0: 禁止 FVR 自动工作	R/W	0
0	ACEN	自动采集使能 1: 使能自动采集 0: 禁止自动采集	R/W	0

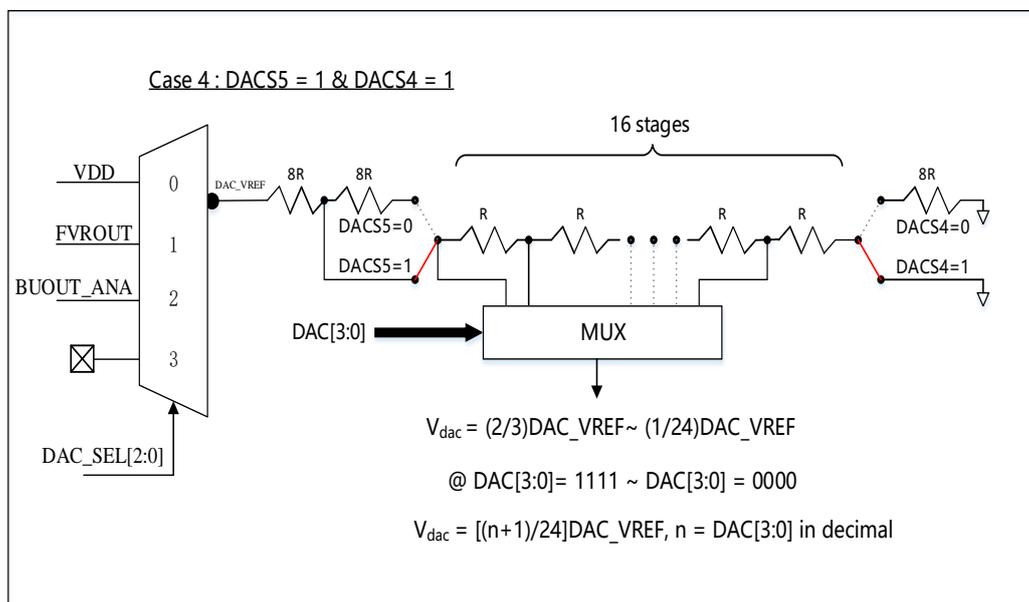
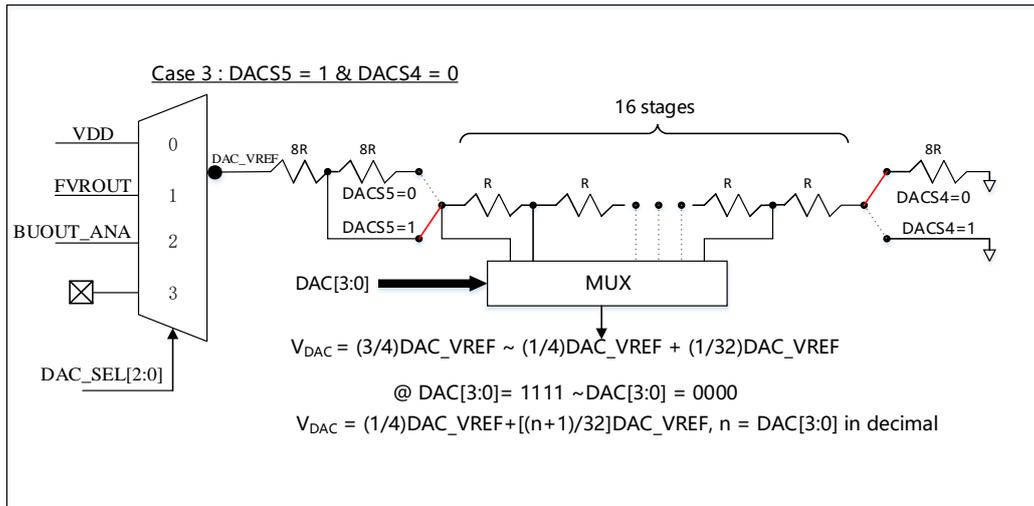
注意:

- 1、使用 ADC 自动采集功能时必须设置 TMR1, 自动采集时间间隔由 TMR1 定时时间确定。
- 2、在 SLEEP_DEEPPWSAVE、SLEEP_PWSAVE 模式下, ADCS 无法设置为内部 1.024M OSC
- 3、ADC 的转换时间=ACQT(延时采集时间)+13TAD
- 4、完成一位的转换的时间定义为 TAD

16 数模转换器 (DAC0 和 DAC1)

AD008 包含两个数模转换器 DAC0 和 DAC1。DAC 是由一串电阻所组成，可以产生不同层次的参考电压，DACON 寄存器的 4 和 5 位用来选择电阻串的最高和最低值；DAC[3:0] 用于选择所要的电压值，该值由 DACS5, DACS4 来决定。下图显示了四个不同选择时，内部参考电压值的计算。DAC 输出电压范围可以从 $(1/32) * VDD$ 到 $(3/4) * VDD$ 。





16.1 DAC 参考电压选择寄存器

DACCON

地址: 0XF75

Bit	Name	Description	Attribute	Reset
7:4	Reserved			
3:2	DAC1VREF[1:0]	DAC1 参考电压选择位 00: VDD 01: FVR0UT 10: BUOUT_ANA 11: 保留	R/W	00
1:0	DAC0VREF[1:0]	DAC0 参考电压选择位 00: VDD	R/W	00

		01: FVROUT 10: BUOUT_ANA 11: 保留		
--	--	---------------------------------------	--	--

16.2 DAC0 控制寄存器

DAC0CON

地址: 0XF74

Bit	Name	Description	Attribute	Reset
7	Reserved			
6	DACOEN	使能 DAC0 1: 使能 DAC0 0: 禁止 DAC0	R/W	0
5	DACOS5	DAC0 正端电阻抽头选择	R/W	0
4	DACOS4	DAC0 负端电阻抽头选择	R/W	0
3:0	DACO[3:0]	DAC0 输出选择	R/W	0

16.3 DAC1 控制寄存器

DAC1CON

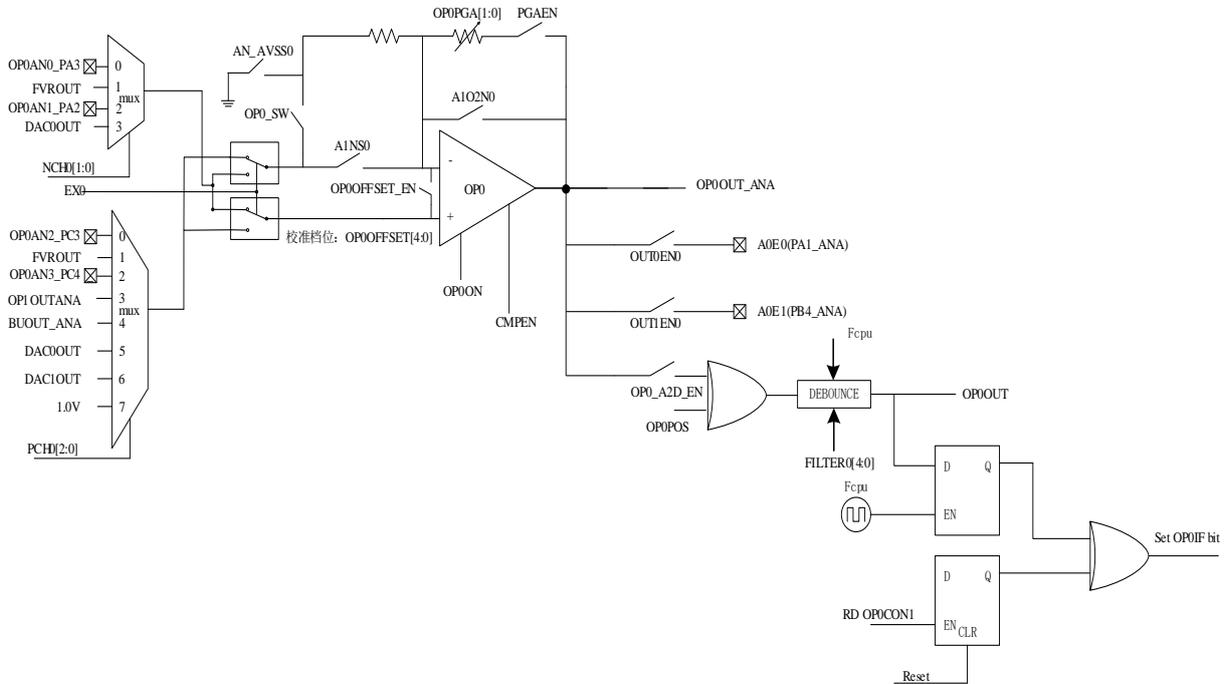
地址: 0XF73

Bit	Name	Description	Attribute	Reset
7	Reserved			
6	DAC1EN	使能 DAC1 1: 使能 DAC1 0: 禁止 DAC1	R/W	0
5	DAC1S5	DAC1 正端电阻抽头选择	R/W	0
4	DAC1S4	DAC1 负端电阻抽头选择	R/W	0
3:0	DAC1[3:0]	DAC1 输出选择	R/W	0

17 运放 (OP0 和 OP1)

AD008 提供两个轨到轨运放, 可以选取多个输入作为运放输入源。

17.1 运放 OP0



OP0 电路图

17.1.1. OPOCON0 (FOP 控制寄存器)

地址: 0XF7F

Bit	Name	Description	Attribute	Reset
7	OPOON	运放使能 1: 使能运放 0: 关闭运放	R/W	0
6	EX0	运放正负端交换 1: 关闭交换 0: 使能交换	R/W	0
5	A1NS0	运放连接 1: 负端与 NCH 选择连接 0: 断开	R/W	0
4:2	PCH0[1:0]	运放 P 输入选择: 000: OPOAN2_PC3 001: FVROUT 010: OPOAN3_PC4	R/W	000

		011: OP1OUTANA 100: BUOUT_ANA 101: DACOOUT 110: DAC1OUT 111: 1.0V (BGR)		
1:0	NCH0[1:0]	运放 N 输入选择 00: OPOANO_PA3 01: FVROUT 10: OPOAN1_PA2 11: DACOOUT	R/W	00

注 1: 当 EX=1 时, N 端接负端, P 端接正端; EX=0 时, N 端接正端, P 端接负端。

17.1.2. OPOCON1 (OP 控制寄存器)

地址: 0XF7E

Bit	Name	Description	Attribute	Reset
7	OPOOUT	运放输出信号 1: 运放输出为 0 0: 运放输出为 1	R	0
6	OUT1ENO	OPO 输出信号映射使能 IO 1: 允许使能输出到 IO 0: 禁止使能输出到 IO	R/W	0
5	AN_AVSS0	是否接地 1: 接地 0: 不接地	R/W	0
4	A102N0	运放 buffer 模式 1: 负端与输出短接, 形成 BUFFER 0: 禁止	R/W	00
3	PGAENO	内部使能放大 1: 使能 0: 禁止	R/W	0
2	OPOPOS	运放输出信号是否取反 1: 取反 0: 同向	R/W	0
1	CMPMODE0	运放切换为比较器模式 1: 切换为比较器模式 0: 切换为运放模式	R/W	0
0	OUT0ENO	OPO 输出信号映射使能 IO 1: 允许使能输出到 IO 0: 禁止使能输出到 IO	R/W	0

17.1.3. OPOCON2 (OP 控制寄存器)

地址: 0XF7D

Bit	Name	Description	Attribute	Reset
7:3	FILTER0[4:0]	OP 的滤波时间 T=FILTER0[4:0] / Fcpu 有效滤除 OPOOUT 在 T 时间内的毛刺 FILTER0[4:0]==0x00 时没有滤波	R/W	0
2	OPO_A2D_EN	运放数字输出使能	R/W	0
1:0	OPOPGA[1:0]	内部放大倍数选择 00: 2 倍 01: 10 倍 10: 20 倍 11: 100 倍	R/W	0

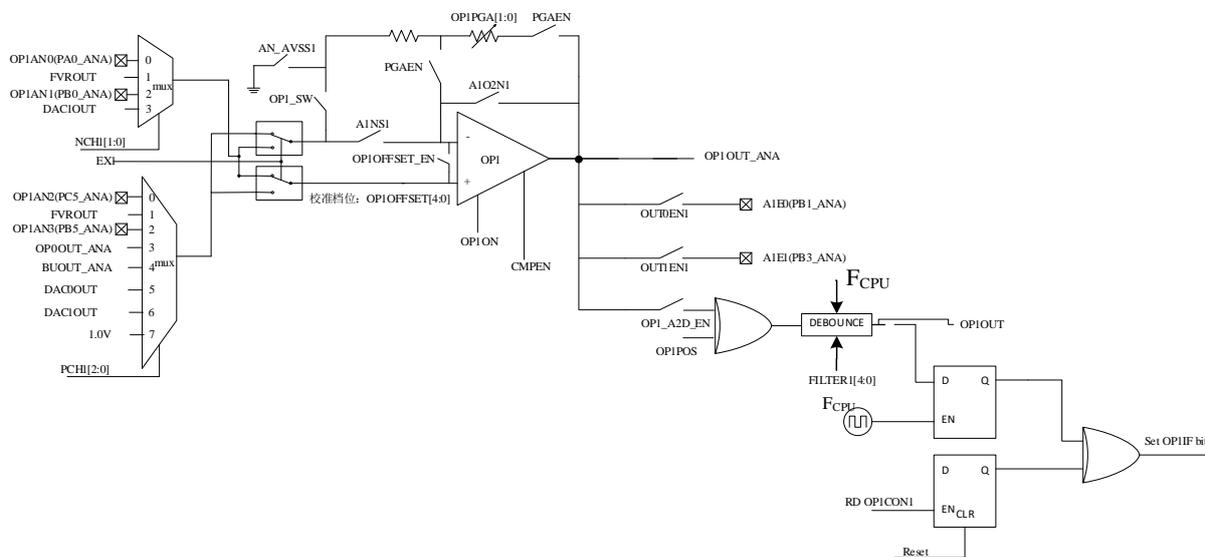
17.1.4. OPOCON3 (OP 控制寄存器)

地址: 0XF57

Bit	Name	Description	Attribute	Reset
7	Reserved			
6	OPO_SW	OPO 开关 1: 允许使能 0: 禁止使能	R/W	0
5	OPOOFFSET_EN	OPO 失调电压校准模式使能位 1: 允许使能 0: 禁止使能	R/W	0
4:0	OPOOFFSET[4:0]	OPO 失调电压校准档 单位为:mV 00000: 0 00001: 1.6 00010: 3.2 00011: 5.0 00100: 6.8 00101: 8.6 00110: 10.6 00111: 12.6 01000: 14.7 01001: 16.9 01010: 19.2 01011: 21.7 01100: 24.2 01101: 26.9	R/W	00000

	01110: 29.8		
	01111: 32.8		
	10000: 0		
	10001: -1.6		
	10010: -3.2		
	10011: -5.0		
	10100: -6.8		
	10101: -8.6		
	10110: -10.6		
	10111: -12.6		
	11000: -14.7		
	11001: -16.9		
	11010: -19.2		
	11011: -21.7		
	11100: -24.2		
	11101: -26.9		
	11110: -29.8		
	11111: -32.8		

17.2 运放 OP1



OP1 电路图

17.2.1. OP1CON0 (OP 控制寄存器)

地址: 0XF7C

Bit	Name	Description	Attribute	Reset
7	OP1ON	运放使能 1: 使能运放	R/W	0

		0: 关闭运放		
6	EX1	运放正负端交换 1: 关闭交换 0: 使能交换	R/W	0
5	A1NS1	运放连接 1: 负端与 NCH 选择连接 0: 断开	R/W	0
4:2	PCH1[1:0]	运放 P 输入选择: 000: OP1AN2_PC5 001: FVROUT 010: OP1AN3_PB5 011: OPOOUT_ANA 100: BUOUT_ANA 101: DACOOUT 110: DAC1OUT 111: 1.0V	R/W	000
1:0	NCH1[1:0]	运放 N 输入选择 00: OP1ANO_PA0 01: FVROUT 10: OP1A1N_PB0 11: DAC1OUT	R/W	00

注 1: 当 EX=1 时, N 端接负端, P 端接正端; EX=0 时, N 端接正端, P 端接负端。

17.2.2. OP1CON1 (OP 控制寄存器)

地址: 0XF7B

Bit	Name	Description	Attribute	Reset
7	OP1OUT	运放输出信号 1: 运放输出为 0 0: 运放输出为 1	R	0
6	OUT1EN1	OP1 输出信号映射使能 IO 1: 允许使能输出到 IO 0: 禁止使能输出到 IO	R/W	0
5	AN_AVSS1	是否接地 1: 接地 0: 不接地	R/W	0
4	A1O2N1	运放 buffer 模式 1: 负端与输出短接, 形成 BUFFER 0: 禁止	R/W	00
3	PGAEN1	内部使能放大 1: 使能 0: 禁止	R/W	0

2	OP1POS	运放输出信号是否取反 1: 取反 0: 同向	R/W	0
1	CMPMODE1	运放切换为比较器模式 1: 切换为比较器模式 0: 切换为运放模式	R/W	0
0	OUTOEN1	OP1 输出信号映射使能 IO 1: 允许使能输出到 IO 0: 禁止使能输出到 IO	R/W	0

17.2.3. OP1CON2 (OP 控制寄存器)

地址: 0XF7A

Bit	Name	Description	Attribute	Reset
7:3	FILTER1[4:0]	OP 的滤波时间 $T = \text{FILTER1}[4:0] / F_{\text{cpu}}$ 有效滤除 OP1OUT 在 T 时间内的毛刺 FILTER1[4:0]==0x00 时没有滤波	R/W	00000
2	OP1_A2D_EN	运放数字输出使能	R/W	0
1:0	OP1APGA[1:0]	内部放大倍数选择 00: 2 倍 01: 10 倍 10: 20 倍 11: 100 倍	R/W	0

17.2.4. OP1CON3 (OP 控制寄存器)

地址: 0XF56

Bit	Name	Description	Attribute	Reset
7	Reserved			
6	OP1_SW	OP1 开关 1: 允许使能 0: 禁止使能	R/W	0
5	OP1OFFSET_EN	OP1 失调电压校准模式使能位 1: 允许使能 0: 禁止使能	R/W	0
4:0	OP1OFFSET[4:0]	OP1 失调电压校准档 单位: mV 00000: 0 00001: 1.6 00010: 3.2 00011: 5.0 00100: 6.8	R/W	00000

		00101: 8.6		
		00110: 10.6		
		00111: 12.6		
		01000: 14.7		
		01001: 16.9		
		01010: 19.2		
		01011: 21.7		
		01100: 24.2		
		01101: 26.9		
		01110: 29.8		
		01111: 32.8		
		10000: 0		
		10001: -1.6		
		10010: -3.2		
		10011: -5.0		
		10100: -6.8		
		10101: -8.6		
		10110: -10.6		
		10111: -12.6		
		11000: -14.7		
		11001: -16.9		
		11010: -19.2		
		11011: -21.7		
		11100: -24.2		
		11101: -26.9		
		11110: -29.8		
		11111: -32.8		

17.3 运放 OP 电流偏置控制寄存器

17.3.1. OPCON(OP 控制寄存器)

地址: 0XF79

Bit	Name	Description	Attribute	Reset
7:4	OP1CSEL	OP1 电流偏置选择位(步进电流:0.125uA) 0000: 1uA 0001: 1.125uA 0010: 1.25uA 0011: 1.375uA 0100: 1.5uA 0101: 1.625uA 0110: 1.75uA 0111: 1.875uA	R/W	0000

		1000: 0nA 1001: 125nA 1010: 250nA 1111: 875nA		
3:0	OPOCSEL	OPO 电流偏置选择位(步进电流:0.125uA) 0000: 1uA 0001: 1.125uA 0010: 1.25uA 0011: 1.375uA 0100: 1.5uA 0101: 1.625uA 0110: 1.75uA 0111: 1.875uA 1000: 0nA 1001: 125nA 1010: 250nA 1111: 875nA	R/W	0000

17.4 OP0 与 OP1 的输入失调校准

17.4.1. OP0OFFSET 校准

步骤 1. 设置 OPOFFSET_EN=1 和 OPOON=1, OP0_SW=0, CMPMODE0=0, PGAEN0=0, A102N0=0, AN_AVSS0=0, A1NS0=0, EX0=1, OP0 进入失调电压校准模式。为确保校准后 VOOS 降到最小值, 校准模式下的输入参考电压大小必须与运算放大器模式下输入的直流工作电压相同。

步骤 2. 设置 OPOFFSET[4:0]=00000, 判断 OP0OUT 的值, 如果是低电平“0”设置 OPOFFSET[4]=0; 如果是高电平“1”设置 OPOFFSET[4]=1。

步骤 3. 设置 OPOFFSET[3:0]=0000, 读取 OP0OUT 的值。

步骤 4. 将 OPOFFSET[3:0]的值加 1, 然后再读取 OP0OUT 引脚值。

如果 OP0OUT 引脚值没有改变, 重复步骤 4 直到 OP0OUT 引脚值发生改变;

如果 OP0OUT 引脚值发生改变, 记录此时 OPOFFSET[3:0]值为 VOOS1, 转到执行步骤 5。

步骤 5. 设置 OPOFFSET[3:0]= 1111, 读取 OP0OUT 的值。

步骤 6. 将 OPOFFSET[3:0]的值减 1, 然后再读取 OP0OUT 的值。

如果 OP0OUT 引脚值没有改变, 重复步骤 6 直到 OP0OUT 引脚值发生改变;

如果 OP0OUT 引脚值发生改变, 记录此时 OPOFFSET[3:0]值为 VOOS2, 转到执行步骤 7。

步骤 7. 重新存储 OPOFFSET[3:0]=VOOS=(VOOS1+VOOS2)/2, 校准完成。若 (VOOS1+VOOS2)/2 非整数, 则只取整数部分, 完成后需关闭 OPOFFSET_EN。

17.4.2. OP10FFSET 校准:

步骤 1. 设置 $OP10FFSET_EN=1$ 和 $OP10N=1$, $OP1_SW=0$, $CMPMODE1=0$, $PGAEN1=0$, $A102N1=0$, $AN_AVSS1=0$, $A1NS1=0$, $EX1=1$, OP1 进入失调电压校准模式。为确保校准后 V_{OOS} 降到最小值, 校准模式下的输入参考电压大小必须与运算放大器模式下输入的直流工作电压相同。

步骤 2. 设置 $OP10FFSET[4:0]=00000$, 判断 $OP10OUT$ 的值, 如果是低电平“0”设置 $OP10FFSET[4]=0$; 如果是高电平“1”设置 $OP10FFSET[4]=1$ 。

步骤 3. 设置 $OP10FFSET[3:0]=0000$, 读取 $OP10OUT$ 的值。

步骤 4. 将 $OP10FFSET[3:0]$ 的值加 1, 然后再读取 $OP10OUT$ 引脚值。

如果 $OP10OUT$ 引脚值没有改变, 重复步骤 4 直到 $OP10OUT$ 引脚值发生改变;

如果 $OP10OUT$ 引脚值发生改变, 记录此时 $OP10FFSET[3:0]$ 值为 V_{OOS1} , 转到执行步骤 5。

步骤 5. 设置 $OP10FFSET[3:0]=1111$, 读取 $OP10OUT$ 的值。

步骤 6. 将 $OP10FFSET[3:0]$ 的值减 1, 然后再读取 $OP10OUT$ 的值。

如果 $OP10OUT$ 引脚值没有改变, 重复步骤 6 直到 $OP10OUT$ 引脚值发生改变;

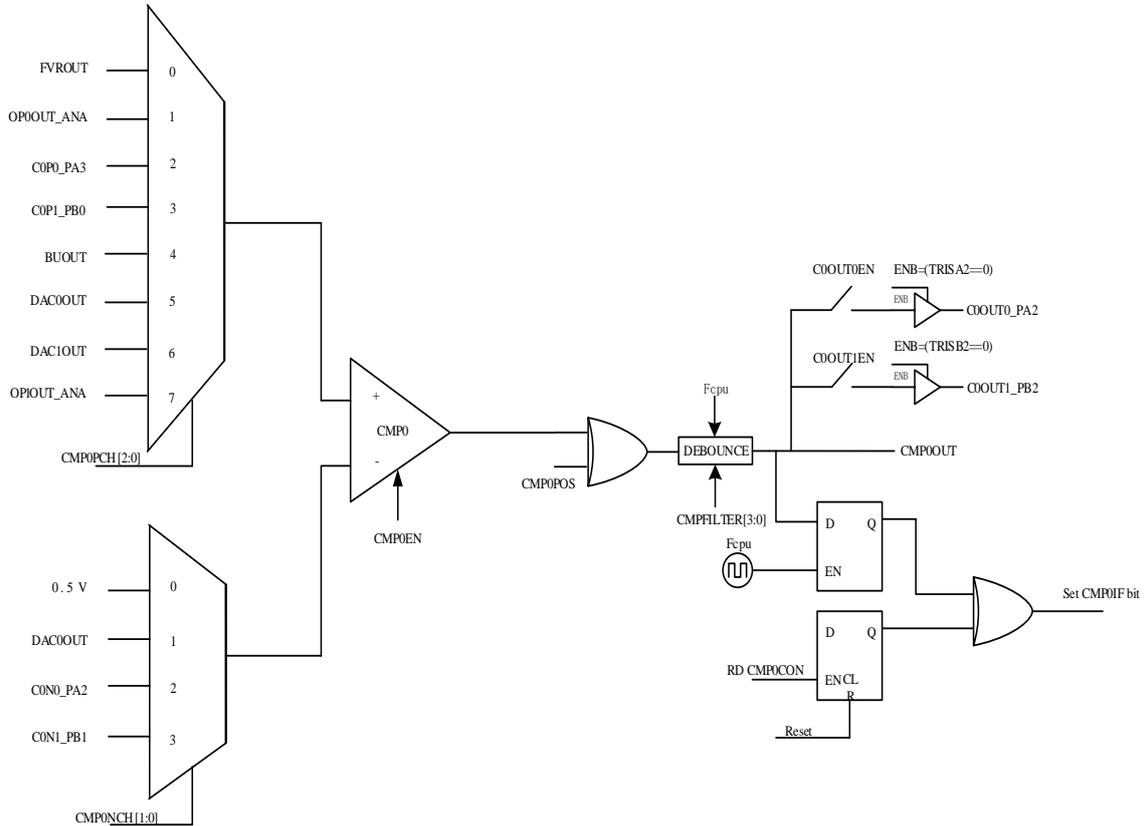
如果 $OP10OUT$ 引脚值发生改变, 记录此时 $OP10FFSET[3:0]$ 值为 V_{OOS2} , 转到执行步骤 7。

步骤 7. 重新存储 $OP10FFSET[3:0]=V_{OOS}=(V_{OOS1}+V_{OOS2})/2$, 校准完成。若 $(V_{OOS1}+V_{OOS2})/2$ 非整数, 则只取整数部分, 完成后需关闭 $OP10FFSET_EN$ 。

18 比较器 (CMP0 和 CMP1)

AD008 提供两个比较器，可以选取多个输入作为比较器输入源。

18.1 比较器 CMP0



CMP0 电路图

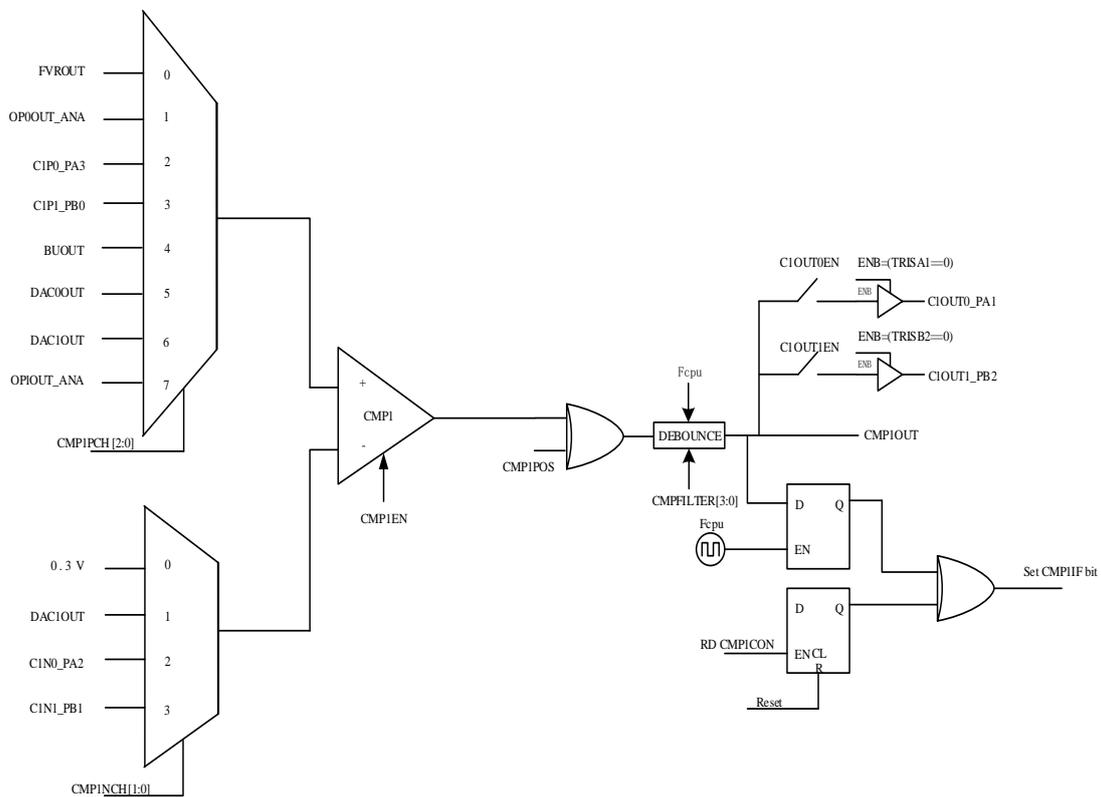
18.1.1. CMP0CON (比较器控制寄存器)

地址: 0XF78

Bit	Name	Description	Attribute	Reset
7	CMPOEN	比较器的使能 1: 使能 CMP 0: 禁止 CMP	R/W	0
6	CMPOOUT	CMP0 比较器的输出	R	0
5	CMPOOS	比较器输出信号是否取反 1: 取反 0: 同向	R/W	0
4:2	CMPOPCH[2:0]	比较器的正端输入 000: FVROUT 001: OPOOUT_ANA	R/W	000

		010: COPO_PA3 011: COP1_PB0 100: BUOUT_ANA 101: DAC0OUT 110: DAC1OUT 111: OP1OUT_ANA		
1:0	CMPONCH[1:0]	比较器的负端输入 00: 0.5V 01: DAC0OUT 10: CON0_PA2 11: CON1_PB1	R/W	00

18.2 比较器 CMP1



CMP1 电路图

18.2.1. CMP1CON (比较器控制寄存器)

地址: 0XF77

Bit	Name	Description	Attribute	Reset
7	CMP1EN	比较器的使能 1: 使能 CMP	R/W	0

		0: 禁止 CMP		
6	CMP1OUT	CMP1 比较器的输出	R	0
5	CMP1POS	比较器输出信号是否取反 1: 取反 0: 同向	R/W	0
4:2	CMP1PCH[2:0]	比较器的正端输入 000: FVROUT 001: OPOOUT_ANA 010: C1P0_PA3 011: C1P1_PB0 100: BUOUT_ANA 101: DACOOUT 110: DAC1OUT 111: OP1OUT_ANA	R/W	000
1:0	CMP1NCH[1:0]	比较器的负端输入 00: 0.3V 01: DAC1OUT 10: C1N0_PA2 11: C1N1_PB1	R/W	00

18.3 比较器输出控制寄存器

18.3.1. CMPCON(比较器控制寄存器)

地址: 0XF76

Bit	Name	Description	Attribute	Reset
7:4	CMPFILTER[3:0]	CMP 的滤波时间 $T = (\text{CMPFILTER}[3:0] \ll 4 + 0X0F) / F_{\text{cpu}}$ 有效滤除 CMP0OUT 与 CMP1OUT 在 T 时间内的毛刺 CMPFILTER[3:0]==0x00 时没有滤波	R/W	0000
3	C1OUT1EN	使能比较器 CMP1OUT 由 PB2 输出, 此时使能 PB2 IO 状态为输出 1: 使能 0: 禁止	R/W	0
2	C1OUT0EN	使能比较器 CMP1OUT 由 PA1 输出, 此时使能 PA1 IO 状态为输出 1: 使能 0: 禁止	R/W	0
1	COOUT1EN	使能比较器 CMP0OUT 由 PB2 输出, 此时使能 PB2 IO 状态为输出	R/W	0

		1: 使能 0: 禁止		
0	COOUTOEN	使能比较器 CMP0OUT 由 PA2 输出, 此时使能 PA2 IO 状态为输出 1: 使能 0: 禁止	R/W	0

19 LCD 驱动模块

AD008 可驱动 1/2Bias 的 LCD，使能 LCD 控制位后，芯片需程序控制输出驱动 LCD。

19.1 LCD 管脚设置

若使能 LCD 模块并使能 COM 口功能，相应的 I/O 口将被强制作为输入态，无需考虑相应 TRIS 位的状态。

19.2 LCD 相关寄存器

19.2.1. LCDCON (LCD 使能寄存器)

地址:0XF71

Bit	Name	Description	Attribute	Reset
7	LCDEN	LCD 使能寄存器 1: 使能 0: 禁止	R/W	0
6:0	Reserved			

19.2.2. LCDCON1 (比较器控制寄存器)

地址: 0XF70

Bit	Name	Description	Attribute	Reset
7	COM7EN	COM7 口使能 (PB1) 1: 使能 0: 禁止	R/W	0
6	COM6EN	COM6 口使能 (PB0) 1: 使能 0: 禁止	R/W	0
5	COM5EN	COM5 口使能 (PA5) 1: 使能 0: 禁止	R/W	0
4	COM4EN	COM4 口使能 (PA4) 1: 使能 0: 禁止	R/W	0
3	COM3EN	COM3 口使能 (PA3) 1: 使能 0: 禁止	R/W	0
2	COM2EN	COM2 口使能 (PA2) 1: 使能	R/W	0

		0: 禁止		
1	COM1EN	COM1 口使能 (PA1) 1: 使能 0: 禁止	R/W	0
0	COM0EN	COM0 口使能 (PA0) 1: 使能 0: 禁止	R/W	0

19.2.3. LCDCON2 (比较器控制寄存器)

地址: 0XF6F

Bit	Name	Description	Attribute	Reset
7:6	LCDISEL	LCD 输出电流选择位 00: 100uA@5V 01: 200uA@5V 10: 400uA@5V 11: 800uA@5V	R/W	00
5:2	Reserved			
1	COM9EN	COM9 口使能 (PB3) 1: 使能 0: 禁止	R/W	0
0	COM8EN	COM8 口使能 (PB2) 1: 使能 0: 禁止	R/W	0

19.2.4. LCDCON3 (比较器控制寄存器)

地址: 0XF6E

Bit	Name	Description	Attribute	Reset
7	COM17EN	COM17 口使能 (PC5) 1: 使能 0: 禁止	R/W	0
6	COM16EN	COM16 口使能 (PC4) 1: 使能 0: 禁止	R/W	0
5	COM15EN	COM15 口使能 (PC3) 1: 使能 0: 禁止	R/W	0
4	COM14EN	COM14 口使能 (PC2) 1: 使能 0: 禁止	R/W	0

3	COM13EN	COM13 口使能 (PC1) 1: 使能 0: 禁止	R/W	0
2	COM12EN	COM12 口使能 (PC0) 1: 使能 0: 禁止	R/W	0
1	COM11EN	COM11 口使能 (PB5) 1: 使能 0: 禁止	R/W	0
0	COM10EN	COM10 口使能 (PB4) 1: 使能 0: 禁止	R/W	0

20 8X8 硬件乘法器

AD008 包含一个 8x8 硬件乘法器。该乘法器可执行无符号运算并产生一个 16 位运算结果，该结果存储在—对乘积寄存器 PRODH:PRODL 中。该乘法器执行的运算不会影响状态寄存器中的任何标志。

通过硬件执行乘法运算只需要 1 个指令周期。硬件乘法器具有更高的计算吞吐量并减少了乘法算法的代码长度，从而可在许多先前仅能使用数字信号处理器的应用中使用 AD008 器件。

21 电气特性

HIRC

Parameter	Symbol	Test Condition	Min	Typ	Max	Unit
16MHz HIRC 频率	F _{osc}	3.3V/5.0V, 25°C	-1%	16	+1%	MHz
	F _{osc}	-40°C~85°C, 2.5V~5.5V	-2.5%	16	+2.5%	MHz

LIRC

Parameter	Symbol	Test Condition	Min	Typ	Max	Unit
1.024MHz LIRC 频率	F _{osc}	3.3V/5.0V, 25°C	-1%	1.024	+1%	MHz
	F _{osc}	-40°C~85°C, 2.2V~5.5V	-5%	1.024	+5%	MHz

CMP

Parameter	Symbol	Test Condition	Min	Typ	Max	Unit
CMP	f _{max}	VCC=5.0V	—	900	—	KHz
	VCC	—	2.2	5	5.5	V
	V _{CM}	5V, TT, 25°C	VSS+0.5	—	VCC-0.6	V
	Response time		—	0.25	—	us

DAC

Parameter	Symbol	Test Condition	Min	Typ	Max	Unit
DAC	VCC	—	2.2	—	5.5	V
	V _{DACO}	VCC=5.0V	(1/32) *DAC_VREF	—	(3/4) *DAC_VREF	V
	DNL	2.2V ≤ VCC ≤ 5.0V	—	±0.05	—	LSB
	INL		—	±0.15	—	LSB

ADC

Parameter	Symbol	Test Condition	Min	Typ	Max	Unit
ADC	VCC	—	2.7	5	5.5	V
	INL	TT, 5V, 25°C, VREF=5V	—	±5	—	LSB
	DNL		—	±1.5	—	LSB
	ENOB	TT, 5V, 25°C,	—	—	12	Bit

		$f_{in}=11.71875\text{KHz}$				
	V_{AD1} (A/D 输入电压)	—	0	—	V_{REF}	V
	V_{REF} (A/D 参考电压)	—	—	—	VCC	V
	t_{on2st} (On-to-start 时间)	—	—	10	—	μs
	t_{ADC} (转换时间)	—	—	13	—	t_{ADCK}
	t_{ADCK} (转换时钟周期)	—	0.125	—	31.25	μs

I/O 特性

Parameter	Symbol	Test Condition	Min	Typ	Max	Unit
I/O 口驱动能力	I_{OH}	VCC = 5V、VCC-0.7	—	10	—	mA
	I_{OL}	CUR=0、VCC = 5V、GND+0.6	—	3	—	mA
	I_{OL}	CUR=1、VCC = 5V、GND+0.6	—	20	—	mA
	I_{OH}	PB2、VCC = 5V、VCC-0.7	—	50	—	mA
	I_{OL}	PB2、VCC = 5V、GND+0.6	—	60	—	mA

WDT

Parameter	Symbol	Test Condition	Min	Typ	Max	Unit
WDT TIME	t_{RSTD}	TWDT_000	—	128	—	ms
		TWDT_001	—	256	—	ms
		TWDT_010	—	384	—	ms
		TWDT_011	—	512	—	ms
		TWDT_100	—	1.024	—	s
		TWDT_101	—	2.048	—	s
		TWDT_110	—	4.096	—	s
		TWDT_111	—	8.192	—	s

OP

Parameter	Symbol	Test Condition	Min	Typ	Max	Unit
OP	VCC	—	2.2	—	5.5	V
	I_Q	$I_0 = 0\text{ mA}$	—	31.5	—	μA
		$T_A = 0^\circ\text{C to } +85^\circ\text{C}$	—	105	—	μA

	V_{OS}	VCC=4.5V, $T_A = 25^\circ\text{C}$	0.5	1	3	mv
	V_{CM}	VCC=5.0V	VSS+0.05	—	VCC-0.05	V
	BW (Gain bandwidth product)	TT, 5V, 25°C, 单位增益 (仿真数据)	—	3.73	—	MHz
	Gain	VCC=5.0V, 单位增益 (仿真数据)	—	132.5	—	dB
	SR+	VCC=5.0V (仿真数据)	—	1.98	—	V/us
	SR-		—	-2.21	—	V/us
	PSRR		—	95.4	—	dB

FVR

Parameter	Symbol	Test Condition	Min	Typ	Max	Unit
FVR	FVR1V0	VCC=3.3V	-0.5%	1	+0.5%	V
	FVR1V1			1.1		V
	FVR2V0			2		V
	FVR2V2			2.2		V

SMT

Parameter	Symbol	Test Condition	Min	Typ	Max	Unit
Schmidt trigger	V_{OH}	VCC = 5V	—	0.68VCC	—	V
	V_{OL}			0.35VCC		
	V_{OH}	VCC = 5V		0.4VCC		
	V_{OL}			0.25VCC		
	V_{OH}	VCC = 3.3V		0.68VCC		
	V_{OL}			0.38VCC		
	V_{OH}	VCC = 3.3V		0.45VCC		
	V_{OL}			0.27VCC		
	V_{OH}	VCC = 2.2V		0.65VCC		
	V_{OL}			0.35VCC		
	V_{OH}	VCC = 2.2V		0.5VCC		

	V_{OL}			0.28VCC		
--	----------	--	--	---------	--	--

LVD

Parameter	Symbol	Test Condition	Min	Typ	Max	Unit
LVD	V_{LVD}	LVD 使能, 电压选择 2.2V	-5%	2.2	+5%	V
		LVD 使能, 电压选择 2.4V		2.4		V
		LVD 使能, 电压选择 2.6V		2.6		V
		LVD 使能, 电压选择 2.7V		2.7		V
		LVD 使能, 电压选择 2.9V		2.9		V
		LVD 使能, 电压选择 3.0V		3		V
		LVD 使能, 电压选择 3.1V		3.1		V
		LVD 使能, 电压选择 3.3V		3.3		V
		LVD 使能, 电压选择 3.6V		3.6		V
		LVD 使能, 电压选择 3.7V		3.7		V
		LVD 使能, 电压选择 3.8V		3.8		V
		LVD 使能, 电压选择 4.1V		4.1		V
		LVD 使能, 电压选择 4.2V		4.2		V
		LVD 使能, 电压选择 4.3V		4.3		V

I/O 上下拉电阻

Parameter	Symbol	Test Condition	Min	Typ	Max	Unit
Pull up resistance/Pull down resistance	R_{PH}	VCC = 5V、_RSEL_HIGH_2L	—	180	—	K Ω
	R_{DH}		—	310	—	K Ω
	R_{PH}	VCC = 5V、_RSEL_LOW_2L	—	25	—	K Ω
	R_{DH}		—	30	—	K Ω
	R_{PH}	VCC = 3.3V、_RSEL_HIGH_2L	—	300	—	K Ω
	R_{DH}		—	540	—	K Ω
	R_{PH}	VCC = 3.3V、_RSEL_LOW_2L	—	40	—	K Ω
	R_{DH}		—	45	—	K Ω
	R_{PH}	VCC = 2.2V、_RSEL_HIGH_2L	—	594	—	K Ω
	R_{DH}		—	1157	—	K Ω

	R _{PH}	VCC = 2.2V、_RSEL_LOW_2L	—	70	—	KΩ
	R _{DH}		—	85	—	KΩ

BUFFER UNIT 误差

Parameter	Symbol	Test Condition	Min	Typ	Max	Unit
BUFFER UNIT ERR	BU ERR	0.15V ≤ BUFFER_IN ≤ 4.60V	—	3	—	mV
		0.045V ≤ BUFFER_IN ≤ 0.1V 4.65V ≤ BUFFER_IN ≤ 4.70	—	9	—	mV
		BUFFER_IN ≤ 0.04V BUFFER_IN ≥ 4.75V	—	25	—	mV

SLEEP

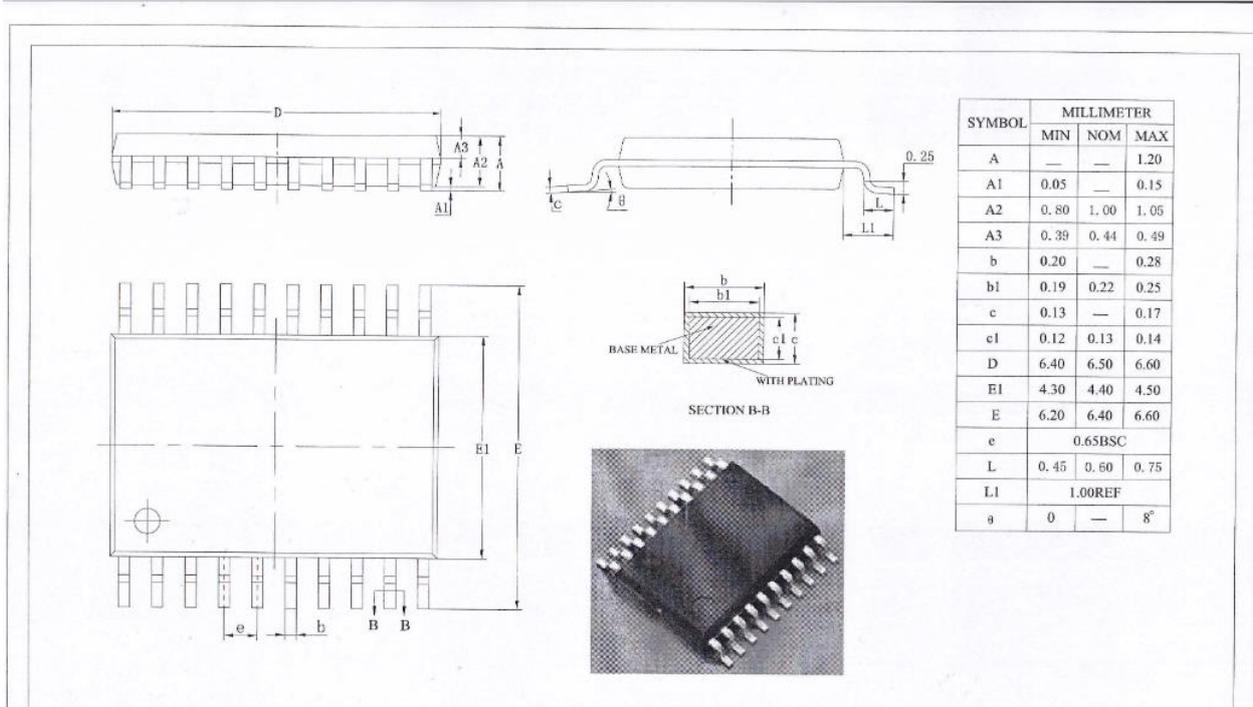
Parameter	Mode	Type of work	Voltage/V							Unit
			2.0	2.5	3.0	3.3	4.0	5.0	5.5	
SLEEP	NORMAL_2T	UART	1.87	2.56	3.04	3.3	3.9	4.75	5.33	mA
	NORMAL_4T	UART	1.46	2	2.34	2.54	2.98	3.6	3.92	mA
	PWIDLE	输出 0	0.74	0.92	1.1	1.2	1.46	1.84	2.04	mA
	PWSAVE	输出 0	32.1	35	36	36.7	38.3	40.9	42.4	μA
	DEEP PWSAVE	输出 0	10.2	11.6	12.6	13.3	14.9	17.3	18.8	μA
	PWOFF	输出 0	0.5	0.5	0.6	0.6	0.6	0.8	0.9	μA

OSC1M SLEEP

Parameter	Mode	Type of work	Voltage/V			Unit
			2.2	3.3	5.0	
OSC1M_SLEEP	正常工作	输出 0	277.2	420	539	μA
	PWIDLE	输出 0	115.5	159.1	230.5	μA
	PWSAVE	输出 0	34.2	39.6	45.6	μA
	DEEP PWSAVE	输出 0	11.5	15.6	19.1	μA
	PWOFF	输出 0	0.3	0.5	0.8	μA

22 封装尺寸

TSSOP20



SOP16

